Name: Alexzander J. Reyes	**Date Performed: 17/10/2025**
Course/Section: CPE212/ CPE31S2	**Date Submitted: 17/10/2025**
Instructor: Engr. Robin Valenzuela	**Semester and SY: 1st & 25-26**

Activity 8: Install, Configure, and Manage Availability Monitoring tools

1. Objectives

Create and design a workflow that installs, configure and manage enterprise monitoring tools using Ansible as an Infrastructure as Code (IaC) tool.

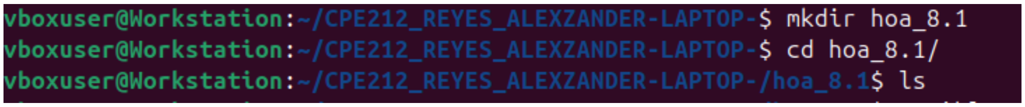
2. Discussion

Availability monitoring is a type of monitoring tool that we use if the certain workload is up or reachable on our end. Site downtime can lead to loss of revenue, reputational damage and severe distress. Availability monitoring prevents adverse situations by checking the uptime of infrastructure components such as servers and apps and notifying the webmaster of problems before they impact on business.

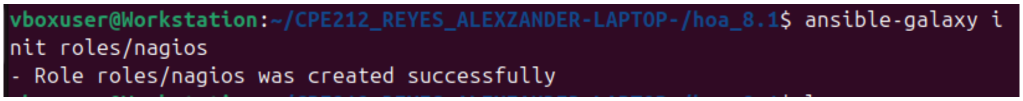
3. Tasks

1. Create a playbook that installs Nagios in both Ubuntu and CentOS. Apply the concept of creating roles.
2. Describe how you did step 1. (Provide screenshots and explanations in your report. Make your report detailed such that it will look like a manual.)
3. Show an output of the installed Nagios for both Ubuntu and CentOS.
4. Make sure to create a new repository in GitHub for this activity.

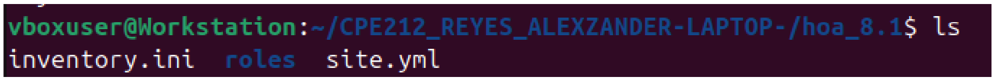
4. **Output** (screenshots and explanations)



```
vboxuser@Workstation:~/CPE212_REYES_ALEXZANDER-LAPTOP-$ mkdir hoa_8.1
vboxuser@Workstation:~/CPE212_REYES_ALEXZANDER-LAPTOP-$ cd hoa_8.1/
vboxuser@Workstation:~/CPE212_REYES_ALEXZANDER-LAPTOP-/hoa_8.1$ ls
```



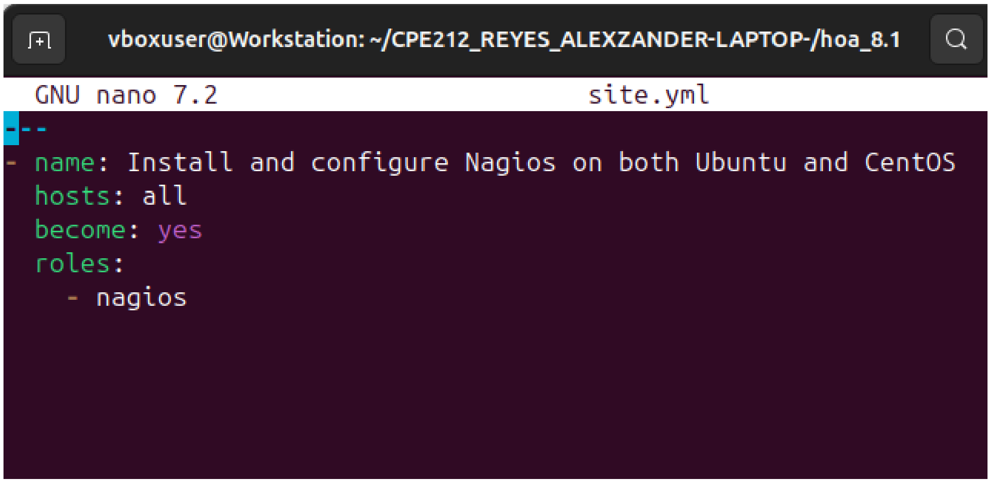
```
vboxuser@Workstation:~/CPE212_REYES_ALEXZANDER-LAPTOP-/hoa_8.1$ ansible-galaxy i
nit roles/nagios
- Role roles/nagios was created successfully
```



```
vboxuser@Workstation:~/CPE212_REYES_ALEXZANDER-LAPTOP-/hoa_8.1$ ls
inventory.ini  roles  site.yml
```

- This series of commands creates a new directory called hoa_8.1 and navigates into it. Inside, an Ansible role named nagios is initialized within the roles folder using the ansible-galaxy init command, which sets up the basic structure for the role. Finally, the contents of the hoa_8.1 directory are listed, showing the inventory.ini file, the newly created roles folder (containing nagios), and the site.yml playbook file.

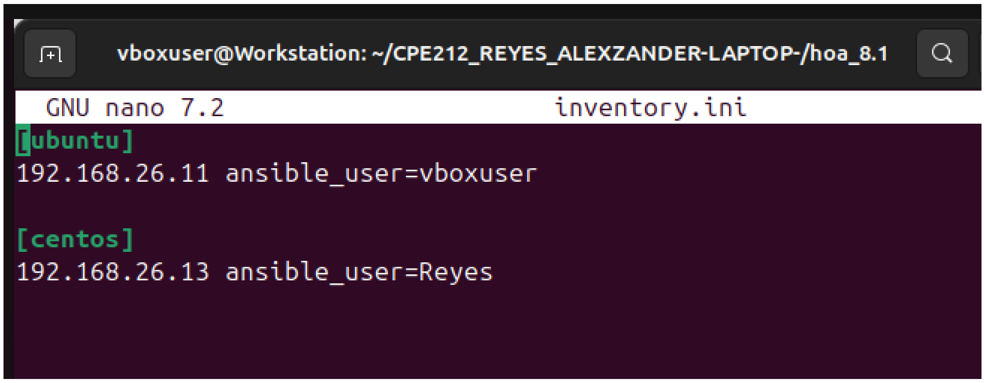
site.yml



```
vboxuser@Workstation: ~/CPE212_REYES_ALEXZANDER-LAPTOP-/hoa_8.1
  GNU nano 7.2                              site.yml
---
- name: Install and configure Nagios on both Ubuntu and CentOS
  hosts: all
  become: yes
  roles:
    - nagios
```

- This file calls the Nagios role for all hosts.

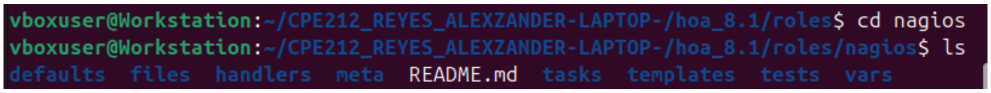
inventory.ini



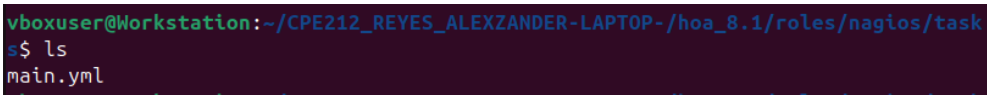
```
vboxuser@Workstation: ~/CPE212_REYES_ALEXZANDER-LAPTOP-/hoa_8.1
  GNU nano 7.2                              inventory.ini
[ubuntu]
192.168.26.11 ansible_user=vboxuser

[centos]
192.168.26.13 ansible_user=Reyes
```

- The inventory file defines our target hosts (Ubuntu and CentOS):

roles

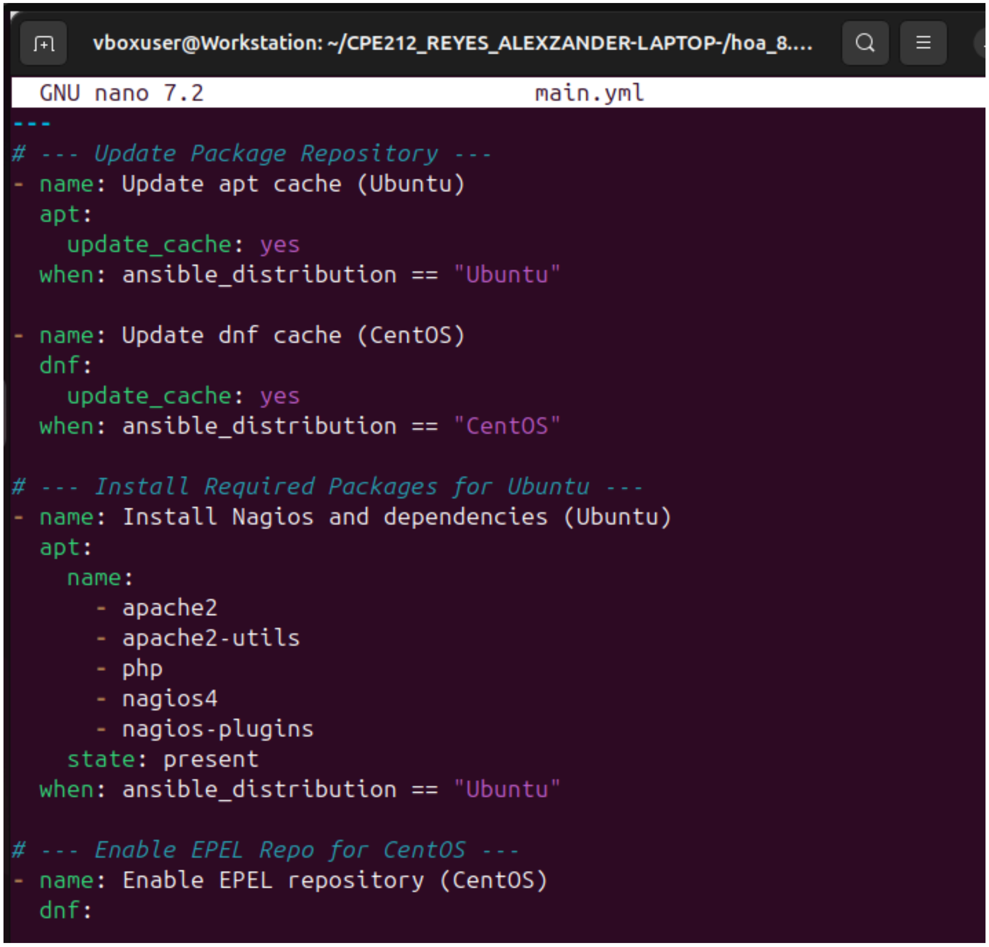


```
vboxuser@Workstation:~/CPE212_REYES_ALEXZANDER-LAPTOP-/hoa_8.1/roles$ cd nagios
vboxuser@Workstation:~/CPE212_REYES_ALEXZANDER-LAPTOP-/hoa_8.1/roles/nagios$ ls
defaults  files  handlers  meta  README.md  tasks  templates  tests  vars
```



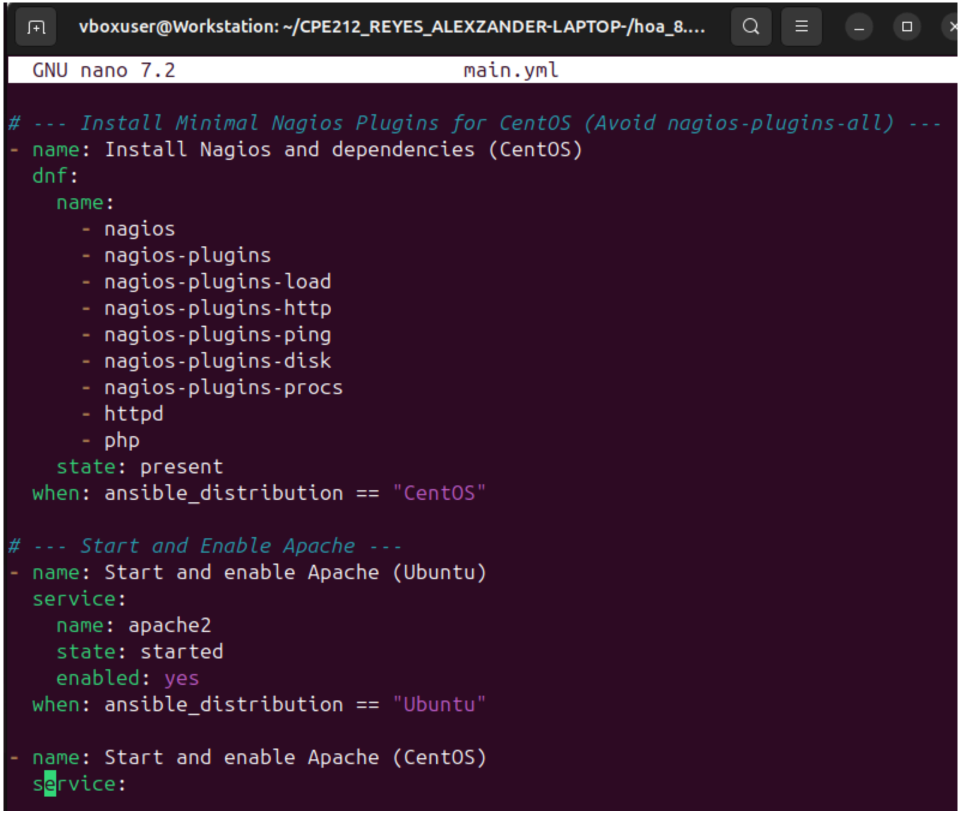
```
vboxuser@Workstation:~/CPE212_REYES_ALEXZANDER-LAPTOP-/hoa_8.1/roles/nagios/task
s$ ls
main.yml
```

tasks/main.yml

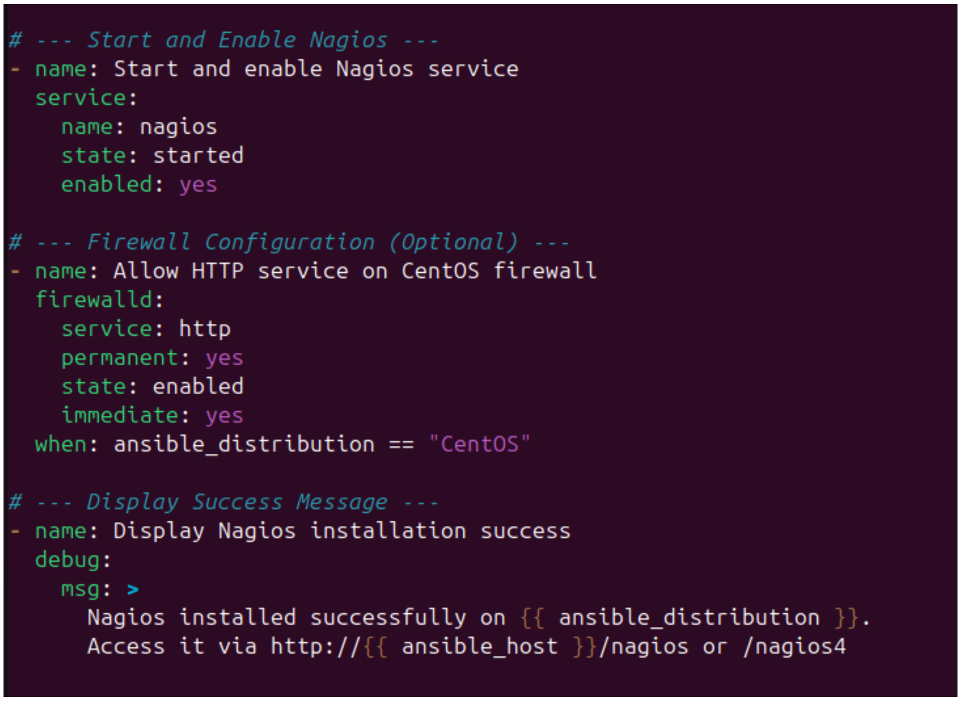

vboxuser@Workstation: ~/CPE212_REYES_ALEXZANDER-LAPTOP-/hoa_8....

GNU nano 7.2 main.yml

```
---
# --- Update Package Repository ---
- name: Update apt cache (Ubuntu)
  apt:
    update_cache: yes
  when: ansible_distribution == "Ubuntu"

- name: Update dnf cache (CentOS)
  dnf:
    update_cache: yes
  when: ansible_distribution == "CentOS"

# --- Install Required Packages for Ubuntu ---
- name: Install Nagios and dependencies (Ubuntu)
  apt:
    name:
      - apache2
      - apache2-utils
      - php
      - nagios4
      - nagios-plugins
    state: present
  when: ansible_distribution == "Ubuntu"

# --- Enable EPEL Repo for CentOS ---
- name: Enable EPEL repository (CentOS)
  dnf:
```


vboxuser@Workstation: ~/CPE212_REYES_ALEXZANDER-LAPTOP-/hoa_8....

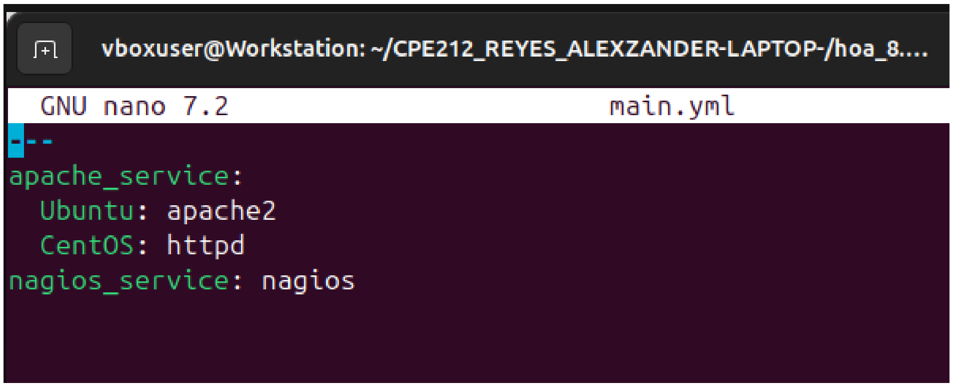
```
  GNU nano 7.2                        main.yml

# --- Install Minimal Nagios Plugins for CentOS (Avoid nagios-plugins-all) ---
- name: Install Nagios and dependencies (CentOS)
  dnf:
    name:
      - nagios
      - nagios-plugins
      - nagios-plugins-load
      - nagios-plugins-http
      - nagios-plugins-ping
      - nagios-plugins-disk
      - nagios-plugins-procs
      - httpd
      - php
    state: present
  when: ansible_distribution == "CentOS"

# --- Start and Enable Apache ---
- name: Start and enable Apache (Ubuntu)
  service:
    name: apache2
    state: started
    enabled: yes
  when: ansible_distribution == "Ubuntu"

- name: Start and enable Apache (CentOS)
  service:
```



```
# --- Start and Enable Nagios ---
- name: Start and enable Nagios service
  service:
    name: nagios
    state: started
    enabled: yes

# --- Firewall Configuration (Optional) ---
- name: Allow HTTP service on CentOS firewall
  firewalld:
    service: http
    permanent: yes
    state: enabled
    immediate: yes
  when: ansible_distribution == "CentOS"

# --- Display Success Message ---
- name: Display Nagios installation success
  debug:
    msg: >
      Nagios installed successfully on {{ ansible_distribution }}.
      Access it via http://{{ ansible_host }}/nagios or /nagios4
```

- This Ansible playbook automates the complete installation and basic configuration of the Nagios monitoring system on both Ubuntu and CentOS servers.

vars/main.yml

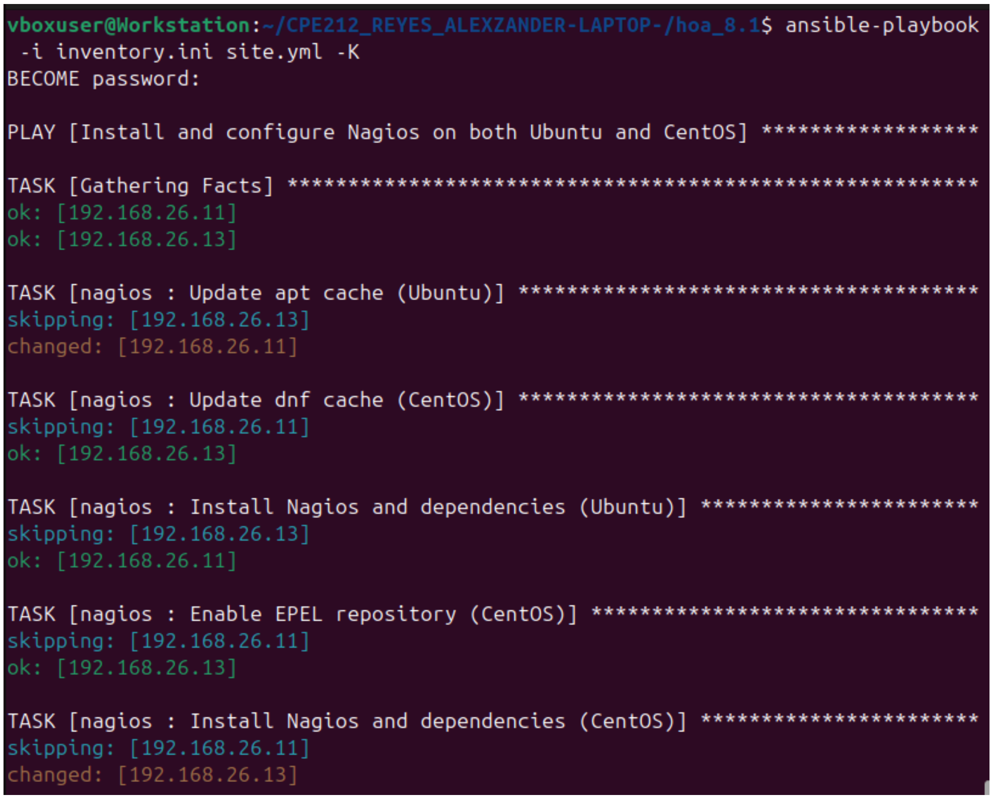


vboxuser@Workstation: ~/CPE212_REYES_ALEXZANDER-LAPTOP-/hoa_8....

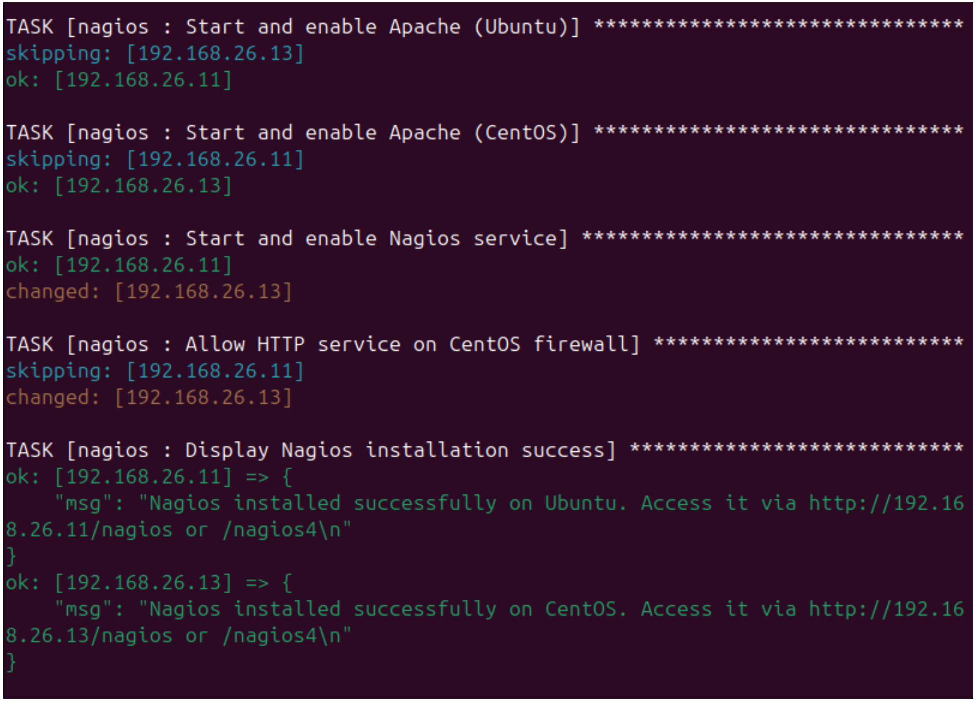
```
  GNU nano 7.2                         main.yml
---
apache_service:
  Ubuntu: apache2
  CentOS: httpd
nagios_service: nagios
```

- This image shows the content of the `vars/main.yml` file in an Ansible role. It defines variables that specify which service names to use depending on the operating system. The variable `apache_service` is set to `apache2` for Ubuntu and `httpd` for CentOS, while `nagios_service` is set to `nagios`. These variables make the playbook adaptable for different Linux distributions by automatically using the correct service names during configuration or installation.

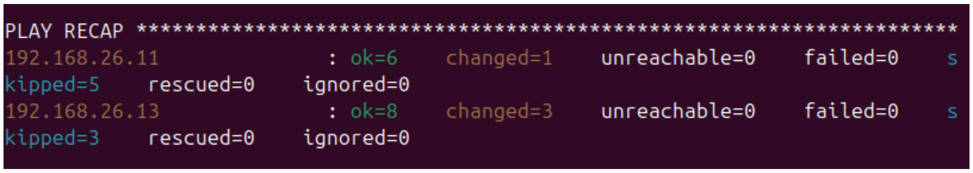
running playbook



```
vboxuser@Workstation:~/CPE212_REYES_ALEXZANDER-LAPTOP-/hoa_8.1$ ansible-playbook
 -i inventory.ini site.yml -K
BECOME password:

PLAY [Install and configure Nagios on both Ubuntu and CentOS] ******************

TASK [Gathering Facts] *********************************************************
ok: [192.168.26.11]
ok: [192.168.26.13]

TASK [nagios : Update apt cache (Ubuntu)] **************************************
skipping: [192.168.26.13]
changed: [192.168.26.11]

TASK [nagios : Update dnf cache (CentOS)] **************************************
skipping: [192.168.26.11]
ok: [192.168.26.13]

TASK [nagios : Install Nagios and dependencies (Ubuntu)] ***********************
skipping: [192.168.26.13]
ok: [192.168.26.11]

TASK [nagios : Enable EPEL repository (CentOS)] ********************************
skipping: [192.168.26.11]
ok: [192.168.26.13]

TASK [nagios : Install Nagios and dependencies (CentOS)] ***********************
skipping: [192.168.26.11]
changed: [192.168.26.13]
```

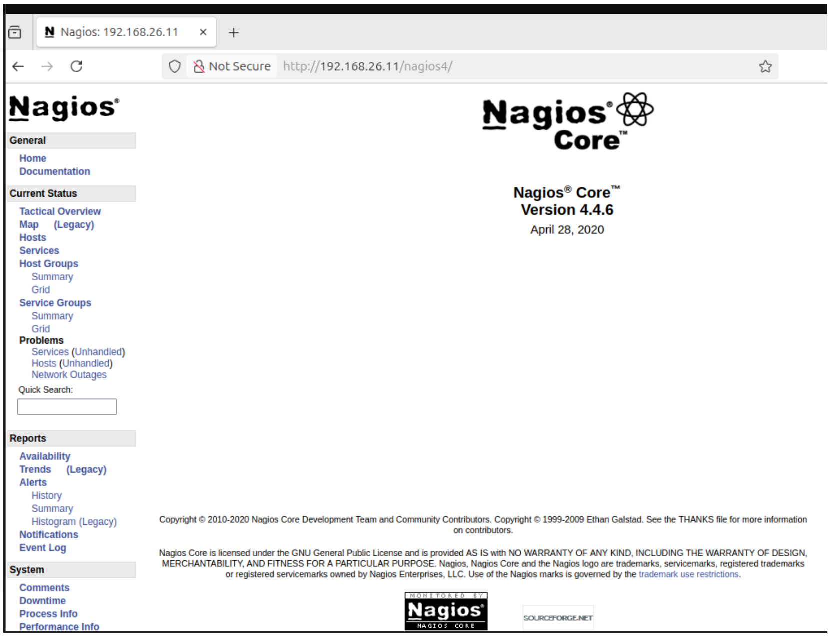


```
TASK [nagios : Start and enable Apache (Ubuntu)] *******************************
skipping: [192.168.26.13]
ok: [192.168.26.11]

TASK [nagios : Start and enable Apache (CentOS)] *******************************
skipping: [192.168.26.11]
ok: [192.168.26.13]

TASK [nagios : Start and enable Nagios service] ********************************
ok: [192.168.26.11]
changed: [192.168.26.13]

TASK [nagios : Allow HTTP service on CentOS firewall] **************************
skipping: [192.168.26.11]
changed: [192.168.26.13]

TASK [nagios : Display Nagios installation success] ****************************
ok: [192.168.26.11] => {
    "msg": "Nagios installed successfully on Ubuntu. Access it via http://192.16
8.26.11/nagios or /nagios4\n"
}
ok: [192.168.26.13] => {
    "msg": "Nagios installed successfully on CentOS. Access it via http://192.16
8.26.13/nagios or /nagios4\n"
}
```



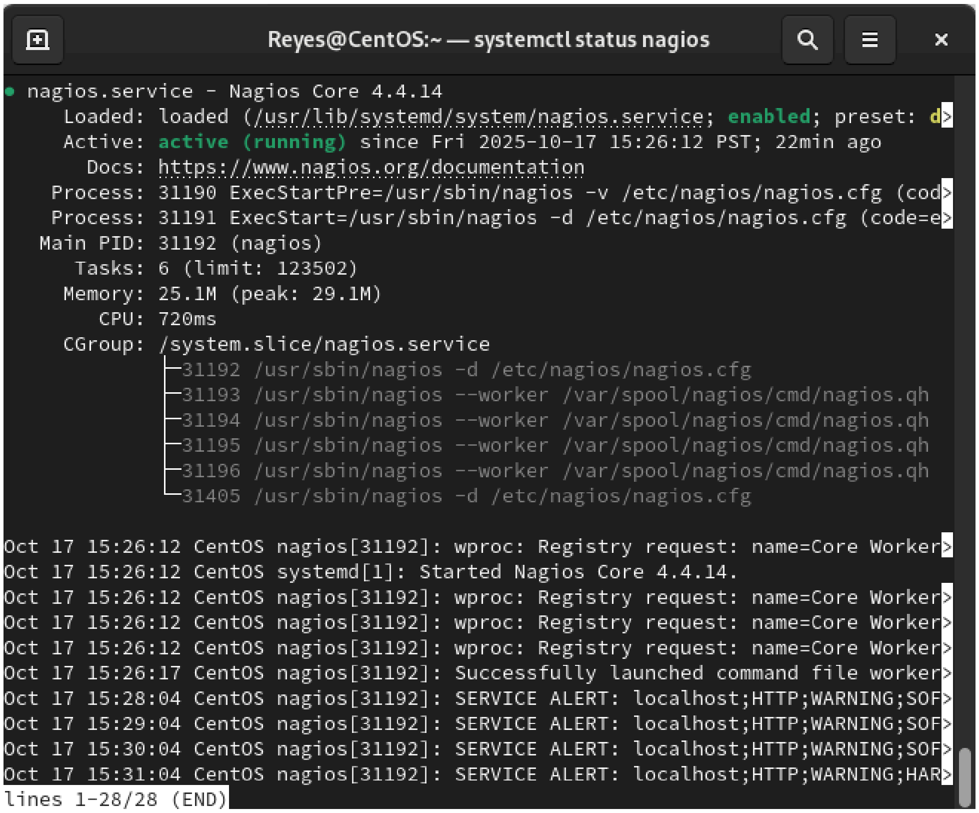
```
PLAY RECAP *********************************************************************
192.168.26.11              : ok=6    changed=1    unreachable=0    failed=0    s
kipped=5    rescued=0    ignored=0
192.168.26.13              : ok=8    changed=3    unreachable=0    failed=0    s
kipped=3    rescued=0    ignored=0
```

ubuntu (192.168.26.11)



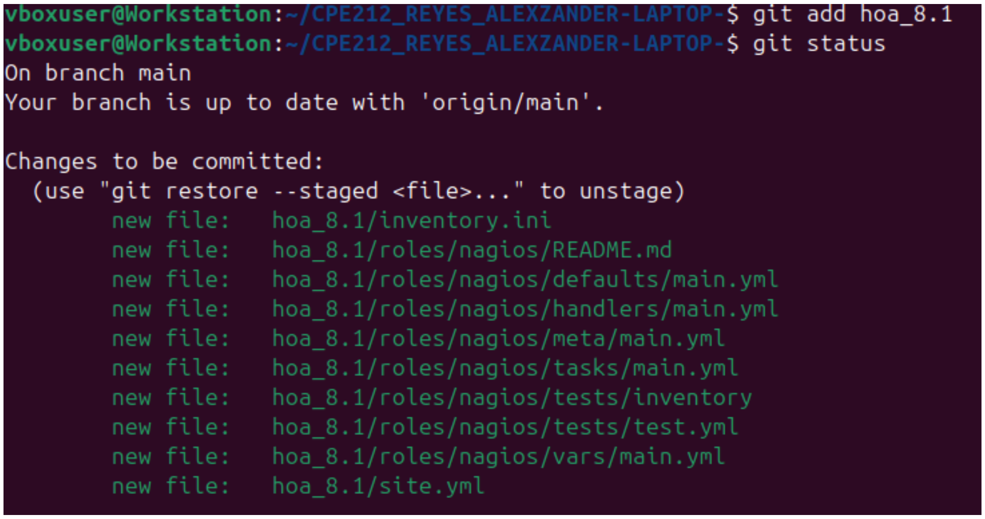
Nagios® Core™
Version 4.4.6
April 28, 2020

centos (192.168.26.13)

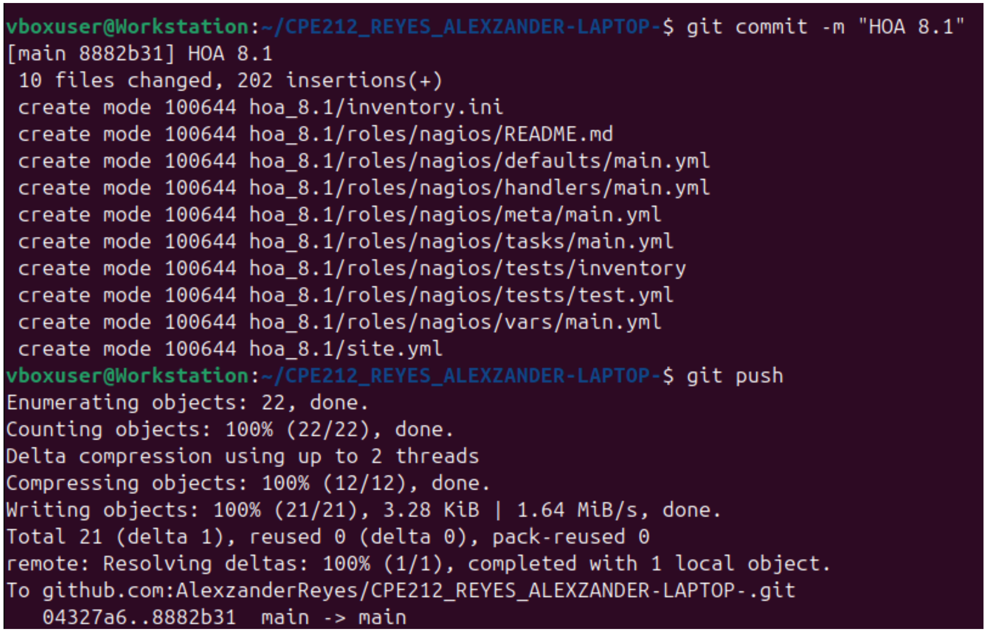


```
● nagios.service - Nagios Core 4.4.14
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: d>
     Active: active (running) since Fri 2025-10-17 15:26:12 PST; 22min ago
       Docs: https://www.nagios.org/documentation
    Process: 31190 ExecStartPre=/usr/sbin/nagios -v /etc/nagios/nagios.cfg (cod>
    Process: 31191 ExecStart=/usr/sbin/nagios -d /etc/nagios/nagios.cfg (code=e>
   Main PID: 31192 (nagios)
      Tasks: 6 (limit: 123502)
     Memory: 25.1M (peak: 29.1M)
        CPU: 720ms
     CGroup: /system.slice/nagios.service
             ├─31192 /usr/sbin/nagios -d /etc/nagios/nagios.cfg
             ├─31193 /usr/sbin/nagios --worker /var/spool/nagios/cmd/nagios.qh
             ├─31194 /usr/sbin/nagios --worker /var/spool/nagios/cmd/nagios.qh
             ├─31195 /usr/sbin/nagios --worker /var/spool/nagios/cmd/nagios.qh
             ├─31196 /usr/sbin/nagios --worker /var/spool/nagios/cmd/nagios.qh
             └─31405 /usr/sbin/nagios -d /etc/nagios/nagios.cfg

Oct 17 15:26:12 CentOS nagios[31192]: wproc: Registry request: name=Core Worker>
Oct 17 15:26:12 CentOS systemd[1]: Started Nagios Core 4.4.14.
Oct 17 15:26:12 CentOS nagios[31192]: wproc: Registry request: name=Core Worker>
Oct 17 15:26:12 CentOS nagios[31192]: wproc: Registry request: name=Core Worker>
Oct 17 15:26:12 CentOS nagios[31192]: wproc: Registry request: name=Core Worker>
Oct 17 15:26:17 CentOS nagios[31192]: Successfully launched command file worker>
Oct 17 15:28:04 CentOS nagios[31192]: SERVICE ALERT: localhost;HTTP;WARNING;SOF>
Oct 17 15:29:04 CentOS nagios[31192]: SERVICE ALERT: localhost;HTTP;WARNING;SOF>
Oct 17 15:30:04 CentOS nagios[31192]: SERVICE ALERT: localhost;HTTP;WARNING;SOF>
Oct 17 15:31:04 CentOS nagios[31192]: SERVICE ALERT: localhost;HTTP;WARNING;HAR>
lines 1-28/28 (END)
```

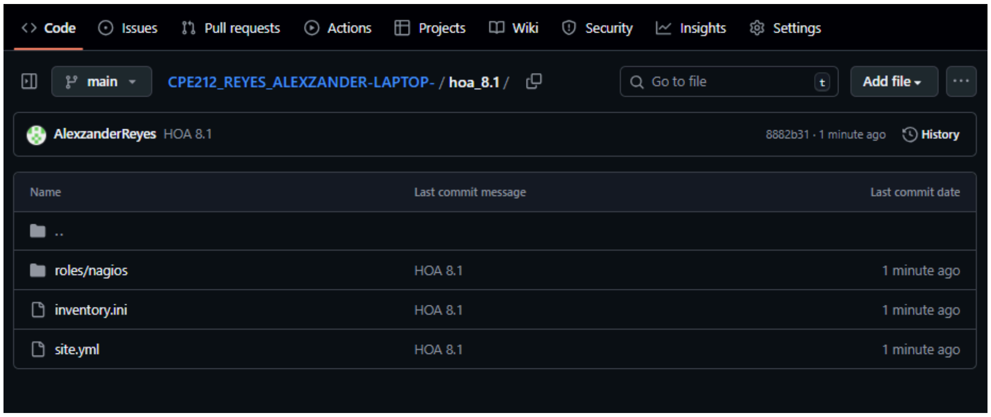
git push



```
vboxuser@Workstation:~/CPE212_REYES_ALEXZANDER-LAPTOP-$ git add hoa_8.1
vboxuser@Workstation:~/CPE212_REYES_ALEXZANDER-LAPTOP-$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   hoa_8.1/inventory.ini
        new file:   hoa_8.1/roles/nagios/README.md
        new file:   hoa_8.1/roles/nagios/defaults/main.yml
        new file:   hoa_8.1/roles/nagios/handlers/main.yml
        new file:   hoa_8.1/roles/nagios/meta/main.yml
        new file:   hoa_8.1/roles/nagios/tasks/main.yml
        new file:   hoa_8.1/roles/nagios/tests/inventory
        new file:   hoa_8.1/roles/nagios/tests/test.yml
        new file:   hoa_8.1/roles/nagios/vars/main.yml
        new file:   hoa_8.1/site.yml
```



```
vboxuser@Workstation:~/CPE212_REYES_ALEXZANDER-LAPTOP-$ git commit -m "HOA 8.1"
[main 8882b31] HOA 8.1
 10 files changed, 202 insertions(+)
 create mode 100644 hoa_8.1/inventory.ini
 create mode 100644 hoa_8.1/roles/nagios/README.md
 create mode 100644 hoa_8.1/roles/nagios/defaults/main.yml
 create mode 100644 hoa_8.1/roles/nagios/handlers/main.yml
 create mode 100644 hoa_8.1/roles/nagios/meta/main.yml
 create mode 100644 hoa_8.1/roles/nagios/tasks/main.yml
 create mode 100644 hoa_8.1/roles/nagios/tests/inventory
 create mode 100644 hoa_8.1/roles/nagios/tests/test.yml
 create mode 100644 hoa_8.1/roles/nagios/vars/main.yml
 create mode 100644 hoa_8.1/site.yml
vboxuser@Workstation:~/CPE212_REYES_ALEXZANDER-LAPTOP-$ git push
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 2 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (21/21), 3.28 KiB | 1.64 MiB/s, done.
Total 21 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:AlexzanderReyes/CPE212_REYES_ALEXZANDER-LAPTOP-.git
   04327a6..8882b31  main -> main
```



CPE212_REYES_ALEXZANDER-LAPTOP- / hoa_8.1 /

AlexzanderReyes HOA 8.1 8882b31 · 1 minute ago

Name	Last commit message	Last commit date
..		
roles/nagios	HOA 8.1	1 minute ago
inventory.ini	HOA 8.1	1 minute ago
site.yml	HOA 8.1	1 minute ago

Reflections:

Answer the following:

1. What are the benefits of having an availability monitoring tool?
   - An availability monitoring tool helps ensure the continuous operation and reliability of IT systems by automatically checking the status of servers, networks, and applications. It detects issues early, sends alerts when problems occur, and provides centralized visibility into system health. This enables faster troubleshooting, reduces downtime, supports proactive maintenance, and offers valuable performance data for capacity planning and reporting. Overall, it improves uptime, operational efficiency, and confidence that critical services are always running smoothly.

Conclusions:

- The activity involved creating and running an Ansible playbook to install and configure Nagios on both Ubuntu and CentOS systems. While doing the activity, I learned how to organize roles, define variables, and manage tasks within the playbook to ensure a smooth and automated installation process. It also helped me understand how Ansible simplifies configuration management across different operating systems. Overall, the activity demonstrated how automation tools like Ansible can efficiently deploy and manage availability monitoring tools such as Nagios, ensuring that systems are properly monitored and maintained.