| Name: Tamayo, Ray Lan A. | Date Performed: 10/18/2024 |
|---|---|
| Course/Section: CPE31S21 | Date Submitted: 10/18/2024 |
| Instructor: Engr. Robin Valenzuela | Semester and SY: First 2024-2025 |

**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.

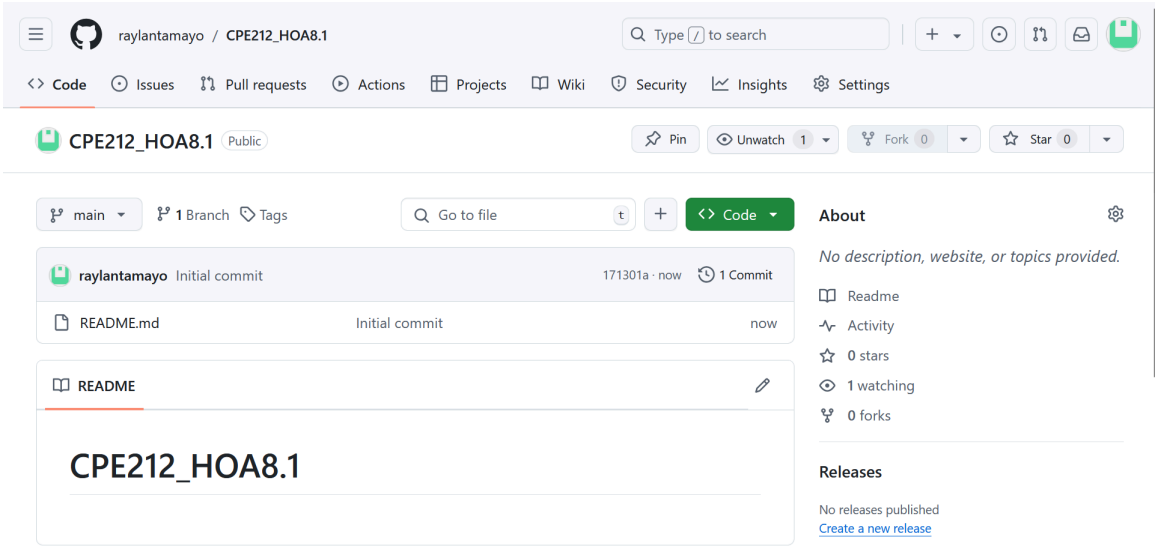
## 3. Tasks

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

### Task 1: Create a File

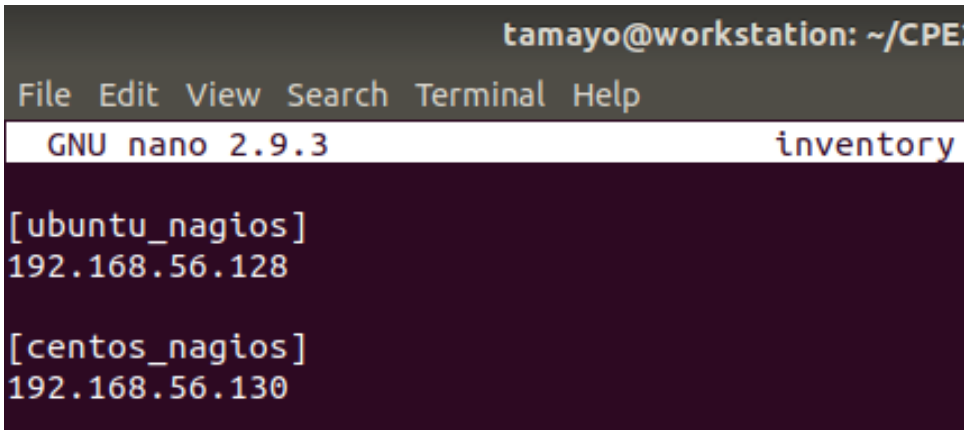
1. Create a new repository for this activity.




2. Clone the repository to the local machine.

```
tamayo@workstation:~$ git clone git@github.com:raylantamayo/CPE212_HOA8.1.git
Cloning into 'CPE212_HOA8.1'...
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:~$ ls
CPE212_HOA6.1  Documents          Music       sysad2
CPE212_HOA8.1  Downloads          Pictures    Tamayo_PrelimExam
CPE232_Tamayo  examples.desktop   Public      Templates
Desktop        HOA7               site.retry  Videos
tamayo@workstation:~$ cd CPE212_HOA8.1
tamayo@workstation:~/CPE212_HOA8.1$
```

3. Create the ansible.cfg and inventory file (must include one Ubuntu and CentOS)



```
tamayo@workstation: ~/CPE212_HOA8.1
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/
```

**Task 2: Create Playbook for Installing Nagios in Ubuntu and CentOS**

1. Create a playbook and name it install_nagios.yml.

```
tamayo@workstation: ~/CPE212_HOA8.1
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                      install_nagios.yml

---

- hosts: all
  become: true
  pre_tasks:

    - name: dnf and epel installation
      dnf:
        name:
          - epel-release
          - dnf
      when: ansible_distribution == "CentOS"

    - name: dpkg in ubuntu
      shell: |
        dpkg --configure -a
      when: ansible_distribution == "Ubuntu"

    - name: install updates (CentOS)
      dnf:
        update_cache: yes
        update_only: yes
      when: ansible_distribution == "CentOS"
```

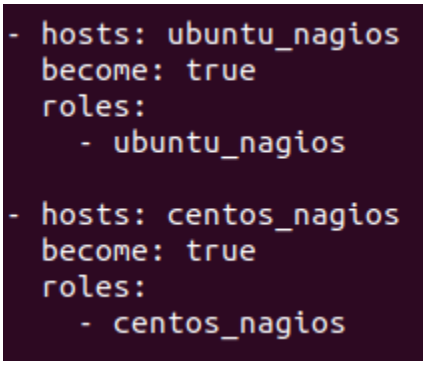
```
    - name: install updates (Ubuntu)
      apt:
        upgrade: dist
        update_cache: yes
      when: ansible_distribution == "Ubuntu"

- hosts: ubuntu_nagios
  become: true
  roles:
    - ubuntu_nagios

- hosts: centos_nagios
  become: true
  roles:
    - centos_nagios
```

**Code Explanation**

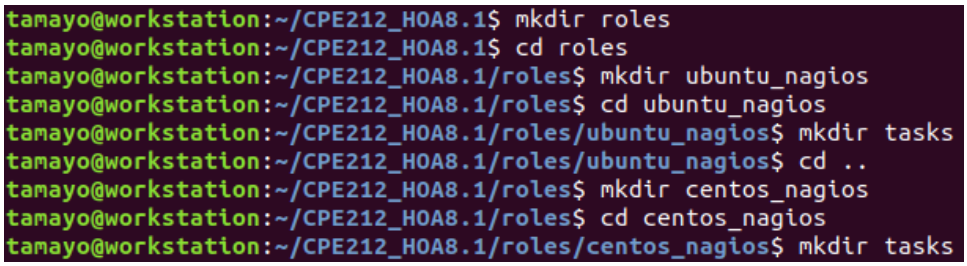
| | |
|---|---|
| It checks if the target system is CentOS. If it is, it installs two packages: "epel-release" and "dnf" using the "dnf" module. This is helpful for managing software on CentOS systems. | `- name: dnf and epel installation`<br>`  dnf:`<br>`  name:`<br>`  - epel-release`<br>`  - dnf`<br>`  when: ansible_distribution == "CentOS"` |
| The dpkg --configure -a command is used to fix any broken or unfinished package installations. It only works on Ubuntu systems, so it's helpful for keeping packages in good condition on Ubuntu servers. | `- name: dpkg in ubuntu`<br>`  shell: \|`<br>`  dpkg --configure -a`<br>`  when: ansible_distribution == "Ubuntu"` |
| It refreshes the package cache and updates only the installed packages. This task runs on CentOS systems to ensure that CentOS servers are up to date with the latest package updates. | `- name: install updates (CentOS)`<br>`  dnf:`<br>`  update_cache: yes`<br>`  update_only: yes`<br>`  when: ansible_distribution == "CentOS"` |
| It upgrades all packages to their latest versions and refreshes the package cache. This task runs only on Ubuntu | `- name: install updates (Ubuntu)`<br>`  apt:`<br>`  upgrade: dist`<br>`  update_cache: yes`<br>`  when: ansible_distribution == "Ubuntu"` |

| | |
|---|---|
| systems to ensure that Ubuntu servers stay updated with the newest package updates. | |
| It uses roles to first install on Ubuntu and then on CentOS, allowing Nagios to monitor both systems. The "become: true" option gives administrative rights to run tasks. |  |

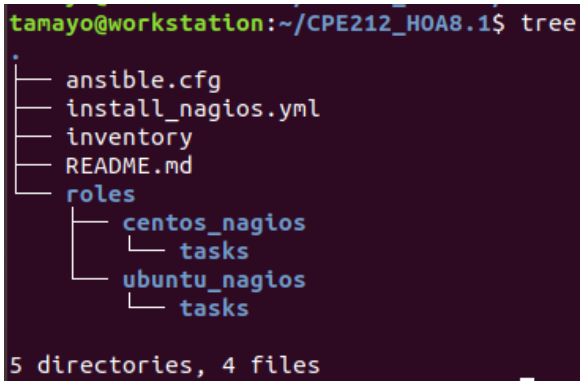


**Task 3: Create Roles**

1. Create a new directory and name its roles. Enter the roles directory and create new directories: centos_nagios and ubuntu_nagios. For each directory , create a directory and name it tasks.









2. In each of the tasks for the two directory (centos_nagios and ubuntu_nagios), create another file and name it main.yml

```
tamayo@workstation:~/CPE212_HOA8.1/roles$ cd ubuntu_nagios
tamayo@workstation:~/CPE212_HOA8.1/roles/ubuntu_nagios$ cd tasks
tamayo@workstation:~/CPE212_HOA8.1/roles/ubuntu_nagios/tasks$ touch main.yml
```

```
tamayo@workstation:~/CPE212_HOA8.1/roles$ cd centos_nagios
tamayo@workstation:~/CPE212_HOA8.1/roles/centos_nagios$ cd tasks
tamayo@workstation:~/CPE212_HOA8.1/roles/centos_nagios/tasks$ touch main.yml
```

```
tamayo@workstation:~/CPE212_HOA8.1/roles$ tree
.
├── centos_nagios
│   └── tasks
│       └── main.yml
└── ubuntu_nagios
    └── tasks
        └── main.yml

4 directories, 2 files
```

3. Copy the code to the main.yml of the Ubuntu subdirectory.

```
tamayo@workstation: ~/CPE212_HOA8.1/roles/ubuntu_nagios
File  Edit  View  Search  Terminal  Help
 GNU nano 2.9.3                    main.yml

---
- name: nagios libraries and dependencies (Ubuntu)
  tags: ubuntu, dependencies, libraries
  apt:
    name:
      - autoconf
      - libc6
      - gcc
      - make
      - wget
      - unzip
      - apache2
      - php
      - libapache2-mod-php
      - libgd-dev
      - openssl
      - libssl-dev
      - bc
      - gawk
      - dc
      - build-essential
      - snmp
      - libnet-snmp-perl
```

```
      - gettext
      - python3
      - python3-pip
    state: latest

- name: passlib package
  pip:
    name: passlib

- name: nagios directory PATH
  file:
    path: ~/nagios
```

```
    state: directory

- name: downloading nagios
  unarchive:
    src: https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.6.$
    dest: ~/nagios
    remote_src: yes
    mode: 0777
    owner: root
    group: root

- name: downloading nagios plugins
```

```
  unarchive:
    src: https://github.com/nagios-plugins/nagios-plugins/archive/release-2.3.$
    dest: ~/nagios
    remote_src: yes
    mode: 0777
    owner: root
    group: root

- name: install, compile, adding users and groups
  shell: |
    cd ~/nagios/nagioscore-*
    sudo ./configure --with-httpd-conf=/etc/apache2/sites-enabled
```

```
    sudo make all
    sudo make install-groups-users
    sudo usermod -a -G nagios www-data
    sudo make install
    sudo make install-daemoninit
    sudo make install-commandmode
    sudo make install-config
    sudo make install-webconf
    sudo a2enmod rewrite
    sudo a2enmod cgi

- name: compile and install plugins
```

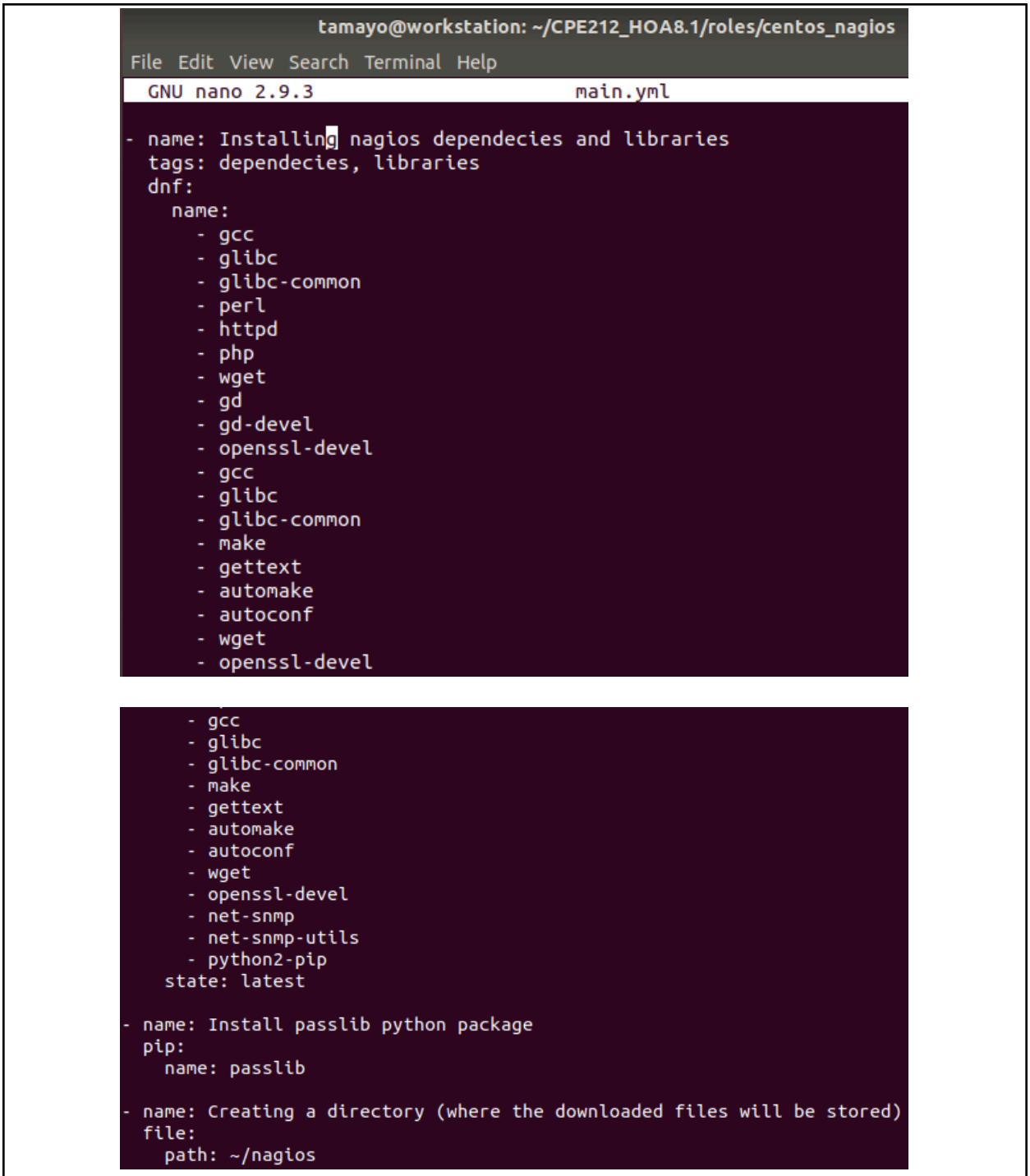
```
  shell: |
    cd ~/nagios/nagios-plugins*
    ./tools/setup
    ./configure
    make
    make install

- name: adding users to nagios
  community.general.htpasswd:
    path: /usr/local/nagios/etc/htpasswd.users
    name: admin
    password: admin
```

```
- name: Nagios Start/Enable Check
  service:
    name: nagios
    state: restarted
    enabled: true

- name: Apache/httpd Start/Enable check
  service:
    name: apache2
    state: restarted
    enabled: true
```

4. Copy the code to the main.yml of the CentOS subdirectory.


tamayo@workstation: ~/CPE212_HOA8.1/roles/centos_nagios
File Edit View Search Terminal Help
GNU nano 2.9.3 main.yml
- name: Installing nagios dependecies and libraries
tags: dependecies, libraries
dnf:
name:
- gcc
- glibc
- glibc-common
- perl
- httpd
- php
- wget
- gd
- gd-devel
- openssl-devel
- gcc
- glibc
- glibc-common
- make
- gettext
- automake
- autoconf
- wget
- openssl-devel
- gcc
- glibc
- glibc-common
- make
- gettext
- automake
- autoconf
- wget
- openssl-devel
- net-snmp
- net-snmp-utils
- python2-pip
state: latest
- name: Install passlib python package
pip:
name: passlib
- name: Creating a directory (where the downloaded files will be stored)
file:
path: ~/nagios

```
    state: directory

- name: Downloading and extracting Nagios
  unarchive:
    src: https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.6.$
    dest: ~/nagios
    remote_src: yes
    mode: 0777
    owner: root
    group: root

- name: Compiling, installing, and adding users and groups in nagios
```

```
  shell: |
    cd ~/nagios/nagioscore-**
    ./configure
    make all
    make install-groups-users
    usermod -a -G nagios apache
    make install
    make install-daemoninit
    make install-commandmode
    make install-config
    make install-webconf
- name: Downloading and extracting Nagios plugins
```

```
  unarchive:
    src: https://github.com/nagios-plugins/nagios-plugins/archive/release-2.3.$
    dest: ~/nagios
    remote_src: yes
    mode: 0777
    owner: root
    group: root

- name: Compiling and installing plugins
  shell: |
    cd ~/nagios/nagios-plugins*
    ./tools/setup
```

```
    ./configure
    make
    make install
- name: Add a user to a password file and ensure permissions are set
  community.general.htpasswd:
    path: /usr/local/nagios/etc/htpasswd.users
    name: admin
    password: admin

- name: Making sure that nagios is started and enabled
  service:
    name: nagios
```

```
    state: restarted
    enabled: true

- name: Making sure that httpd is started and enabled
  service:
    name: httpd
    state: restarted
    enabled: true
```

**Task 4: Run and Verify**

1. Run the command ansible-playbook - - ask-become-pass install_nagios.yml to completely install Nagios in both Ubuntu server and CentOS.

```
tamayo@workstation:~/CPE212_HOA8.1$ ansible-playbook --ask-become-pass install_
nagios.yml
SUDO password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.130]
ok: [192.168.56.128]

TASK [dnf and epel installation] ***********************************************
*
skipping: [192.168.56.128]
skipping: [192.168.56.130]

TASK [dpkg in ubuntu] **********************************************************
*
changed: [192.168.56.130]
changed: [192.168.56.128]

TASK [install updates (CentOS)] ************************************************
*
skipping: [192.168.56.128]
skipping: [192.168.56.130]
```

```
TASK [install updates (Ubuntu)] ************************************************
*

changed: [192.168.56.130]
changed: [192.168.56.128]

PLAY [ubuntu_nagios] ***********************************************************
*

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.128]

PLAY [centos_nagios] ***********************************************************
*

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.130]

PLAY RECAP *********************************************************************
*
192.168.56.128             : ok=4    changed=2    unreachable=0    failed=0
192.168.56.130             : ok=4    changed=2    unreachable=0    failed=0
```

2. Show the screenshot of the Nagios in both Server 2 and CentOS, by simply typing its ip address in the web browser and /nagios.

**OUTPUT:**

**SERVER2**


**CENTOS**


3. Upload it in the github.

```
tamayo@workstation:~/CPE212_HOA8.1$ git add .
tamayo@workstation:~/CPE212_HOA8.1$ git commit -m "HOA 8.1"
[main 6453e3a] HOA 8.1
 7 files changed, 239 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 install_nagios.yml
 create mode 100644 inventory
 create mode 100644 roles/centos_nagios/main.yml
 create mode 100644 roles/centos_nagios/tasks/main.yml
 create mode 100644 roles/ubuntu_nagios/main.yml
 create mode 100644 roles/ubuntu_nagios/tasks/main.yml
tamayo@workstation:~/CPE212_HOA8.1$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
tamayo@workstation:~/CPE212_HOA8.1$ git push origin
Counting objects: 12, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (10/10), done.
Writing objects: 100% (12/12), 2.47 KiB | 2.47 MiB/s, done.
Total 12 (delta 0), reused 0 (delta 0)
To github.com:raylantamayo/CPE212_HOA8.1.git
   171301a..6453e3a  main -> main
```

**GITHUB LINK:** https://github.com/raylantamayo/CPE212_HOA8.1.git

**Reflections:**

Answer the following:

1. What are the benefits of having an availability monitoring tool?
   The first benefit is that it quickly alerts users when something goes wrong, allowing them to find and fix problems before they get worse. Additionally, monitoring tools help ensure that services are running smoothly, which leads to happier users and protects the business's reputation.

**Conclusions:**

After doing this activity, I can honestly say it was the hardest one we've done so far. Even if your code is really well-written, if your computer acts up, it can make things a lot harder and test your patience while you try to fix it. However, I also realized that installing Nagios on CentOS and Ubuntu has many benefits. It provides detailed monitoring of systems and gives real-time updates on network performance and application availability. This helps administrators spot issues early, avoid downtime, and ensure a smooth experience for users. Nagios is an essential tool for businesses and organizations because it works well with all Linux systems. In the end, this activity was tough, but the sense of achievement I felt when I finished it without any errors was amazing.