Name: Ilagan, Carlo Hideki D.	Date Performed:10-14-2024
Course/Section: CPE31S2	Date Submitted:10-16-2024
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st sem / 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.

   Playbook code:

```
---

 - name: Install required dependencies on Ubuntu
     apt:
     name:
     - gcc
     - libc6
     - make
     - wget
     - unzip
     - apache2
     - php
     - libgd-dev
     - openssl
     - libssl-dev
     - autoconf
     - bc
     - gawk
     - dc
     - build-essential
```

```
      - snmp
      - libnet-snmp-perl
      - gettext
      state: present
      when: ansible_distribution == "Ubuntu"

  - name: Install required dependencies on CentOS
      yum:
      name:
      - gcc
      - glibc
      - glibc-common
      - wget
      - unzip
      - httpd
      - php
      - gd
      - gd-devel
      - perl
      - postfix
      - openssl
      - openssl-devel
      - make
      - autoconf
      state: present
      when: ansible_distribution == "CentOS"

  - name: Download Nagios Core source code
      get_url:
      url:
"https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.6.tar.gz"
      dest: /tmp/nagios-4.5.6.tar.gz

  - name: Extract Nagios source code
      unarchive:
      src: /tmp/nagios-4.5.6.tar.gz
      dest: /tmp
      remote_src: yes

  - name: Download Nagios Plugins
      get_url:
      url: "https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz"
      dest: /tmp/nagios-plugins-2.4.11.tar.gz

  - name: Extract Nagios Plugins
```

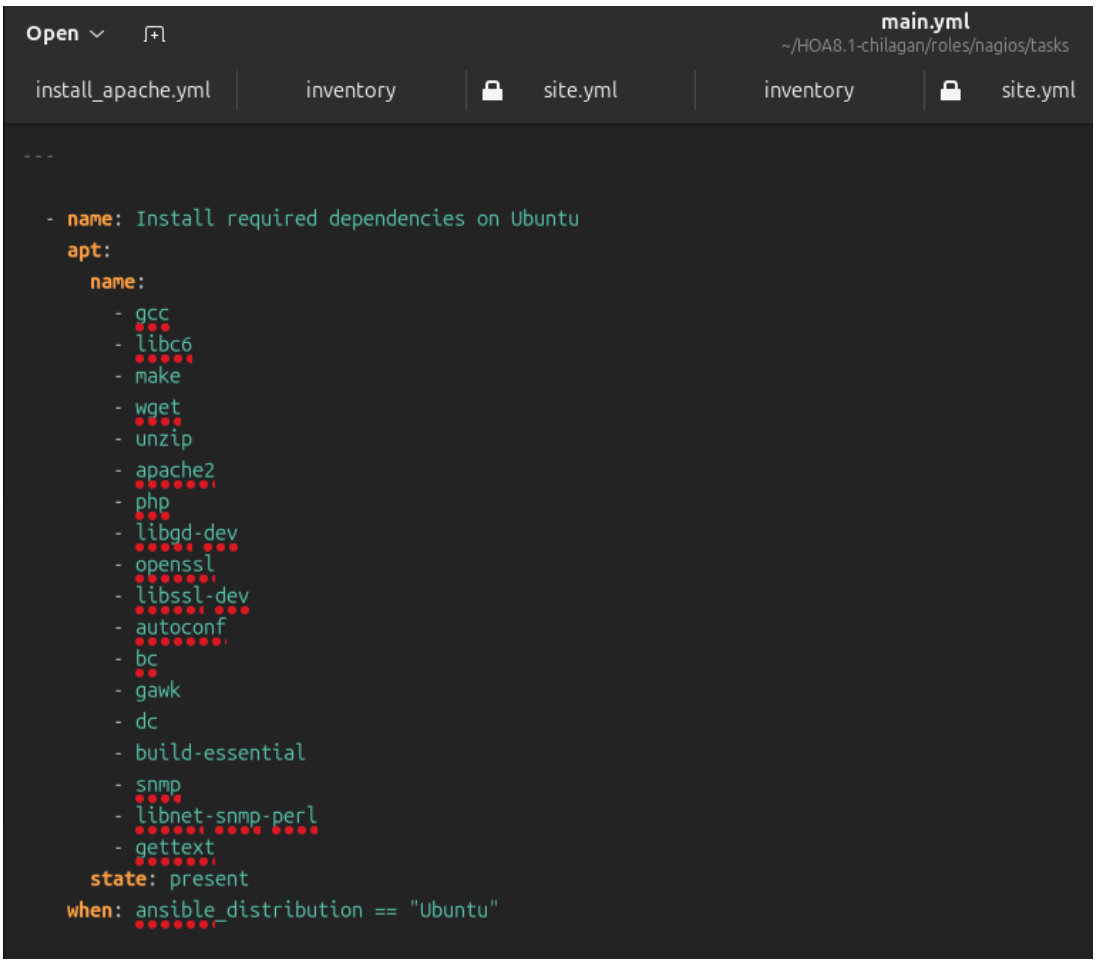
```
    unarchive:
    src: /tmp/nagios-plugins-2.4.11.tar.gz
    dest: /tmp
    remote_src: yes

- name: Create Nagios group
    group:
    name: nagios

- name: Create Nagios user and group
    user:
    name: nagios
    group: nagios

- name: Create nagcmd group
    group:
    name: nagcmd

- name: Add nagios and apache/httpd users to nagcmd group
    user:
    name: "{{ item }}"
    groups: nagcmd
    append: yes
    loop:
    - nagios
    - "{{ 'www-data' if ansible_os_family == 'Debian' else 'apache' }}"

- name: Compile and install Nagios Core
    shell: |
    cd /tmp/nagios-4.5.6
    ./configure --with-command-group=nagcmd
    make all
    make install
    make install-init
    make install-commandmode
    make install-config
    make install-webconf
    args:
    creates: /usr/local/nagios/bin/nagios

- name: Install Nagios Plugins
    shell: |
    cd /tmp/nagios-plugins-2.4.11
    ./configure --with-nagios-user=nagios --with-nagios-group=nagios
    make
```

```
    make install
    args:
    creates: /usr/local/nagios/libexec/check_http

  - name: Set Nagios admin password
    command: htpasswd -b -c /usr/local/nagios/etc/htpasswd.users
nagios_admin "123qweasdzxc"

  - name: Enable and start Apache/Httpd service on Ubuntu
    service:
    name: apache2
    enabled: yes
    state: started
    when: ansible_distribution == "Ubuntu"

  - name: Enable and start Apache/Httpd service on CentOS
    service:
    name: httpd
    enabled: yes
    state: started
    when: ansible_distribution == "CentOS"

  - name: Enable and start Nagios service
    service:
    name: nagios
    enabled: yes
    state: started

  - name: Enable external command execution in Nagios
    lineinfile:
    path: /usr/local/nagios/etc/nagios.cfg
    regexp: '^#?check_external_commands='
    line: 'check_external_commands=1'

  - name: Restart Nagios service to apply changes
    service:
    name: nagios
    state: restarted

  - name: Restart Apache/Httpd to apply changes on Ubuntu
    service:
    name: apache2
    state: restarted
    when: ansible_distribution == "Ubuntu"
```

```
- name: Restart Apache/Httpd to apply changes on CentOS
    service:
    name: httpd
    state: restarted
    when: ansible_distribution == "CentOS"
```

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.)

Install required dependencies for both CentOS and Ubuntu in order to run Nagios



```yaml
- name: Install required dependencies on CentOS
  yum:
    name:
      - gcc
      - glibc
      - glibc-common
      - wget
      - unzip
      - httpd
      - php
      - gd
      - gd-devel
      - perl
      - postfix
      - openssl
      - openssl-devel
      - make
      - autoconf
    state: present
  when: ansible_distribution == "CentOS"
```

Download and Extract Nagios source code and plugins (Make sure that version will match the destination)

```yaml
- name: Download Nagios Core source code
  get_url:
    url: "https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.6.tar.gz"
    dest: /tmp/nagios-4.5.6.tar.gz

- name: Extract Nagios source code
  unarchive:
    src: /tmp/nagios-4.5.6.tar.gz
    dest: /tmp
    remote_src: yes

- name: Download Nagios Plugins
  get_url:
    url: "https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz"
    dest: /tmp/nagios-plugins-2.4.11.tar.gz

- name: Extract Nagios Plugins
  unarchive:
    src: /tmp/nagios-plugins-2.4.11.tar.gz
    dest: /tmp
    remote_src: yes
```

Create a user and groups for Nagios and nagcmd:

```
- name: Create Nagios group
  group:
    name: nagios

- name: Create Nagios user and group
  user:
    name: nagios
    group: nagios

- name: Create nagcmd group
  group:
    name: nagcmd
```

Add nagios to nagcmd group

```
- name: Add nagios and apache/httpd users to nagcmd group
  user:
    name: "{{ item }}"
    groups: nagcmd
    append: yes
  loop:
    - nagios
    - "{{ 'www-data' if ansible_os_family == 'Debian' else 'apache' }}"
```

```
- name: Compile and install Nagios Core
  shell: |
    cd /tmp/nagios-4.5.6
    ./configure --with-command-group=nagcmd
    make all
    make install
    make install-init
    make install-commandmode
    make install-config
    make install-webconf
  args:
    creates: /usr/local/nagios/bin/nagios

- name: Install Nagios Plugins
  shell: |
    cd /tmp/nagios-plugins-2.4.11
    ./configure --with-nagios-user=nagios --with-nagios-group=nagios
    make
    make install
  args:
    creates: /usr/local/nagios/libexec/check_http
```

Set admin and password for nagios:

```
  - name: Set Nagios admin password
    command: htpasswd -b -c /usr/local/nagios/etc/htpasswd.users nagios_admin "123qweasdzxc"
```

Start the apache service on both ubuntu and CentOS

```
- name: Enable and start Apache/Httpd service on Ubuntu
  service:
    name: apache2
    enabled: yes
    state: started
  when: ansible_distribution == "Ubuntu"

- name: Enable and start Apache/Httpd service on CentOS
  service:
    name: httpd
    enabled: yes
    state: started
  when: ansible_distribution == "CentOS"

- name: Enable and start Nagios service
  service:
    name: nagios
    enabled: yes
    state: started
```

Make sure to allow external command execution for nagios:

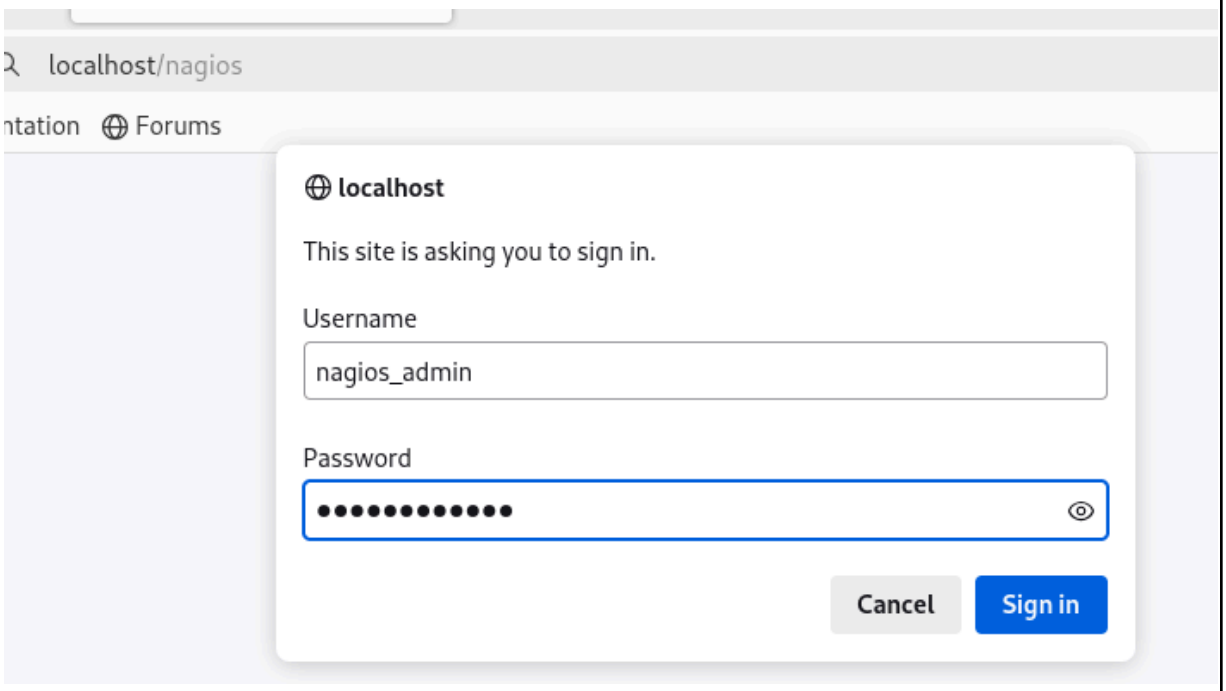
```
- name: Enable external command execution in Nagios
  lineinfile:
    path: /usr/local/nagios/etc/nagios.cfg
    regexp: '^#?check_external_commands='
    line: 'check_external_commands=1'
```

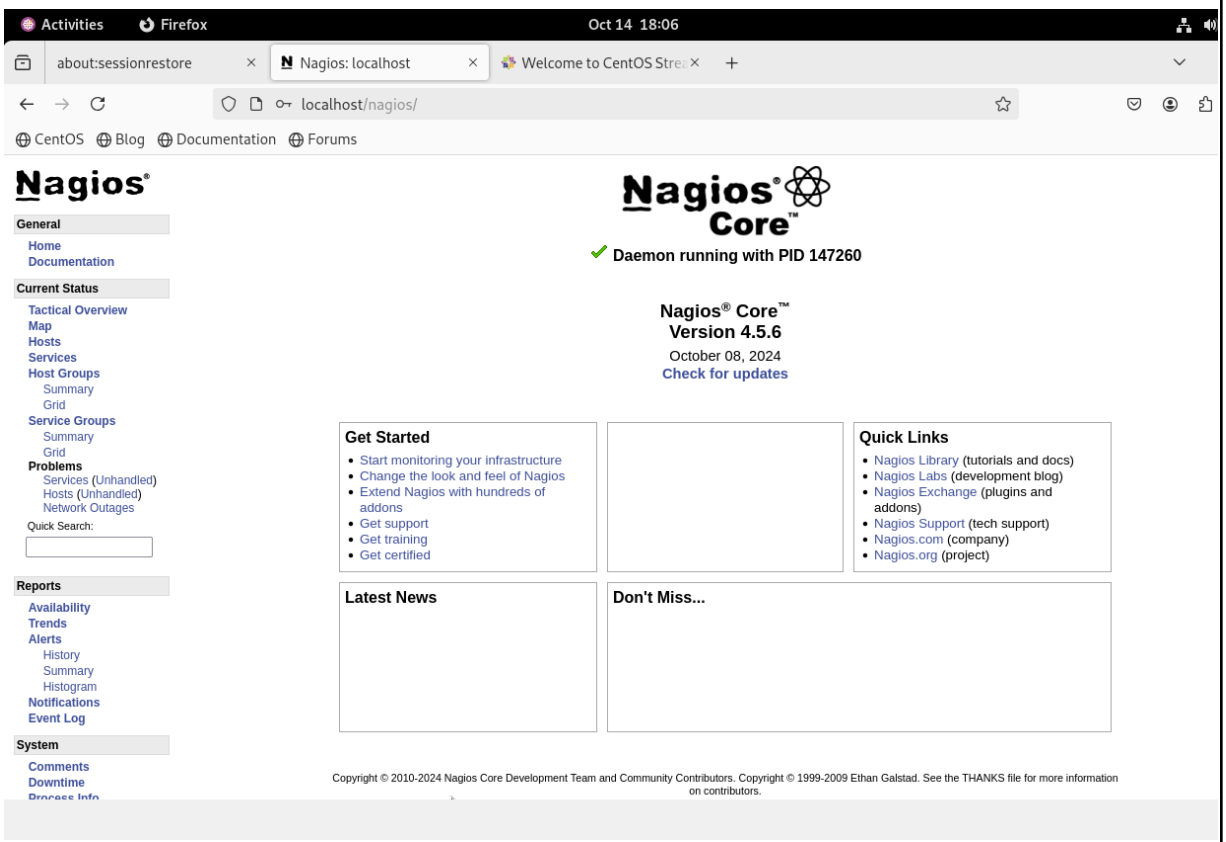
Restart services for apache and httpd to make sure that the nagios will run properly:

```
- name: Restart Nagios service to apply changes
  service:
    name: nagios
    state: restarted

- name: Restart Apache/Httpd to apply changes on Ubuntu
  service:
    name: apache2
    state: restarted
  when: ansible_distribution == "Ubuntu"

- name: Restart Apache/Httpd to apply changes on CentOS
  service:
    name: httpd
    state: restarted
  when: ansible_distribution == "CentOS"
```

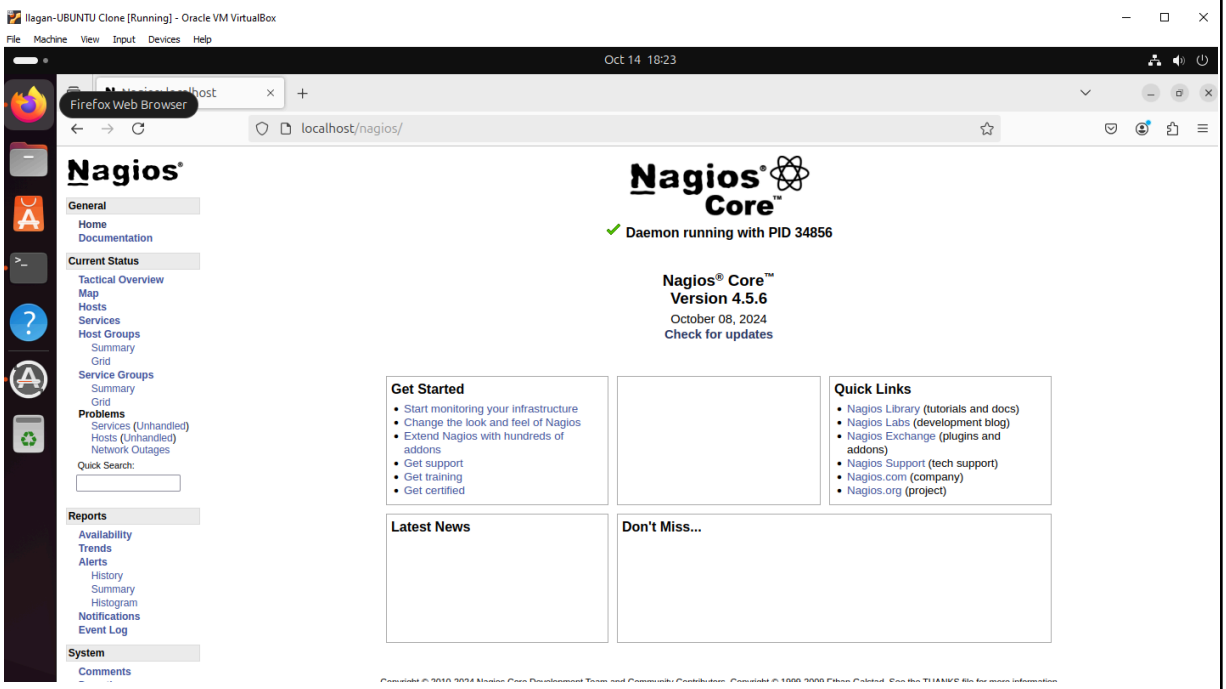
3. Show an output of the installed Nagios for both Ubuntu and CentOS.

CentOS web interface:

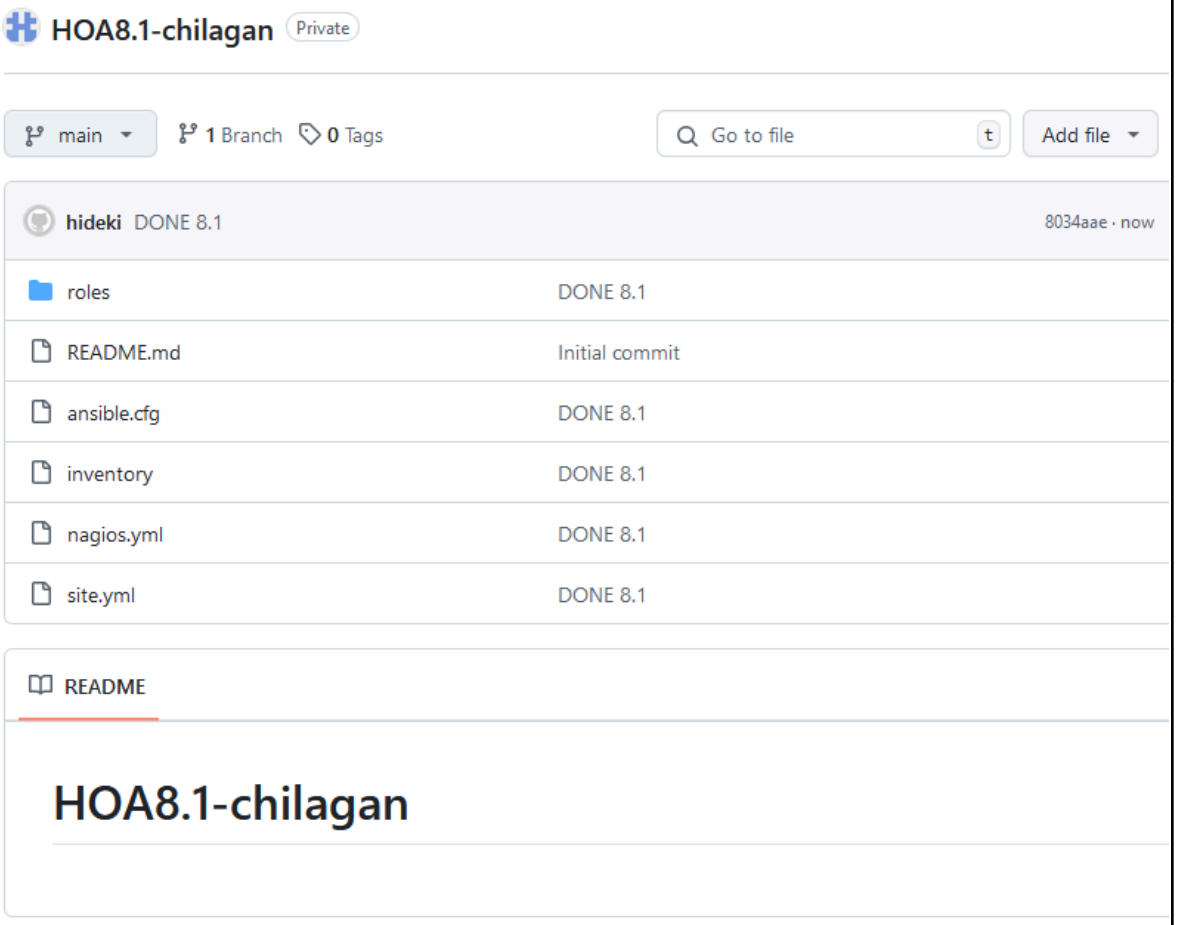




Ubuntu web interface:



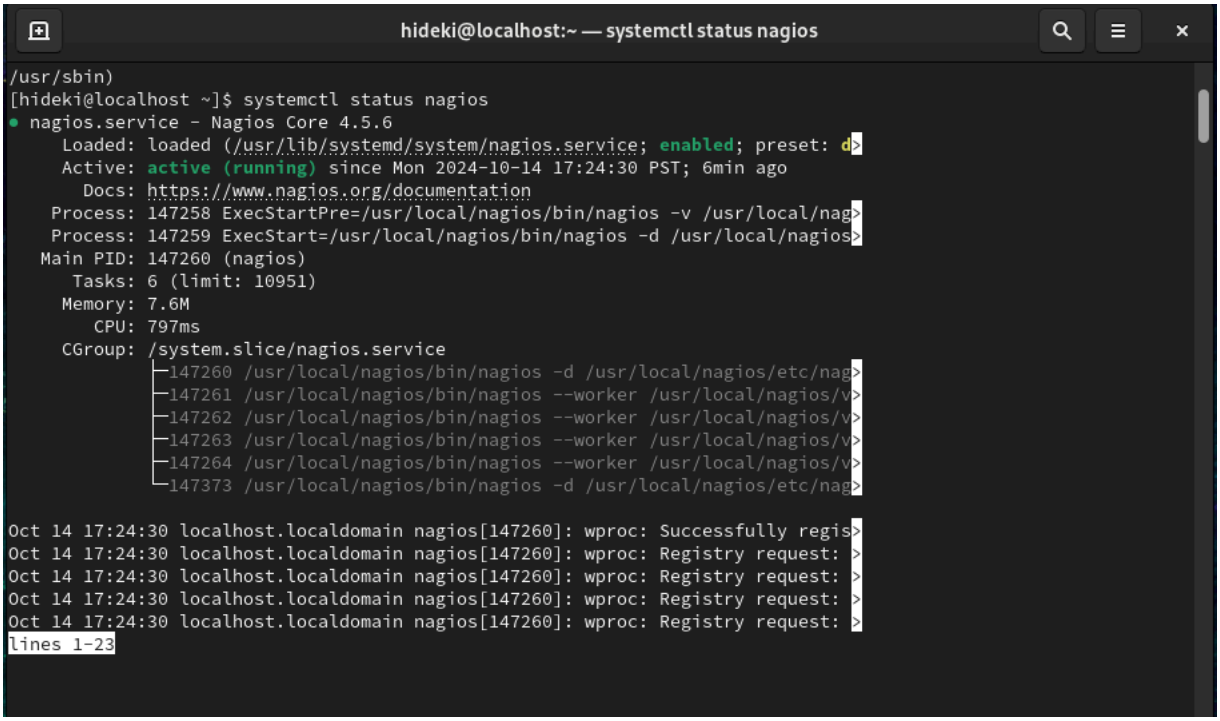
4. Make sure to create a new repository in GitHub for this activity.



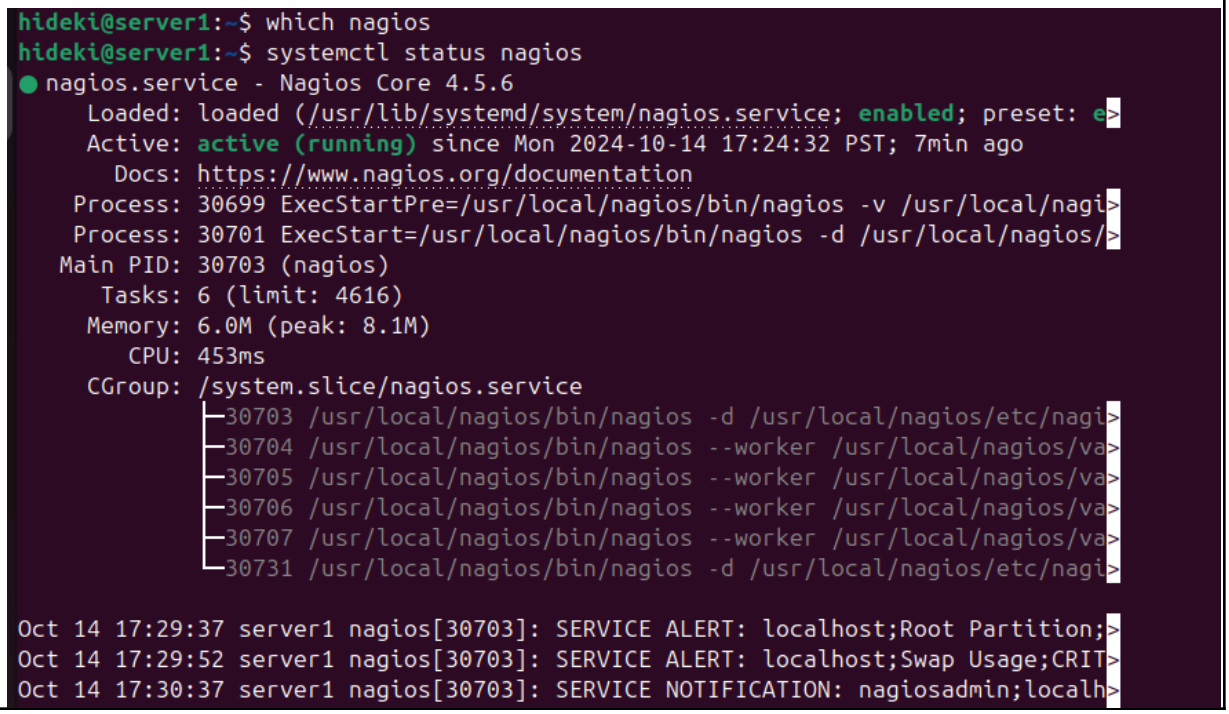
Github link: https://github.com/chilagan-github/HOA8.1-chilagan

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

CentOS Nagios Output:



Ubuntu Nagios Output:



Reflections:

Answer the following:

1. What are the benefits of having an availability monitoring tool?

   Having a monitoring tool will help a system administrator to ensure that each server is running properly and will have the ability to instantly fix issues that might occur. In this case, we used an application called Nagios. It is a powerful monitoring tool that provides insights into the performance and health of managednodes. By alerting administrators to potential issues and enabling proactive management of servers, it helps maintain the reliability and availability of critical systems and services.

Conclusions:

In conclusion, the use of monitoring tools is essential for maintaining the performance and reliability of critical workloads. By implementing tools and software such as Nagios, administrators can monitor their infrastructure, ensuring that any issues are identified and addressed before they escalate into serious problems.This practice not only provide precaution against downtime of the servers but also protect the revenue of the organization.