Name: Atienza, Stephen Lhaye C.	Date Performed:03/21/24
Course/Section:CPE231S1	Date Submitted:03/24/24
Instructor: Dr. Jonathan V. Taylar	Semester and SY: 2nd and 2023-2024

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

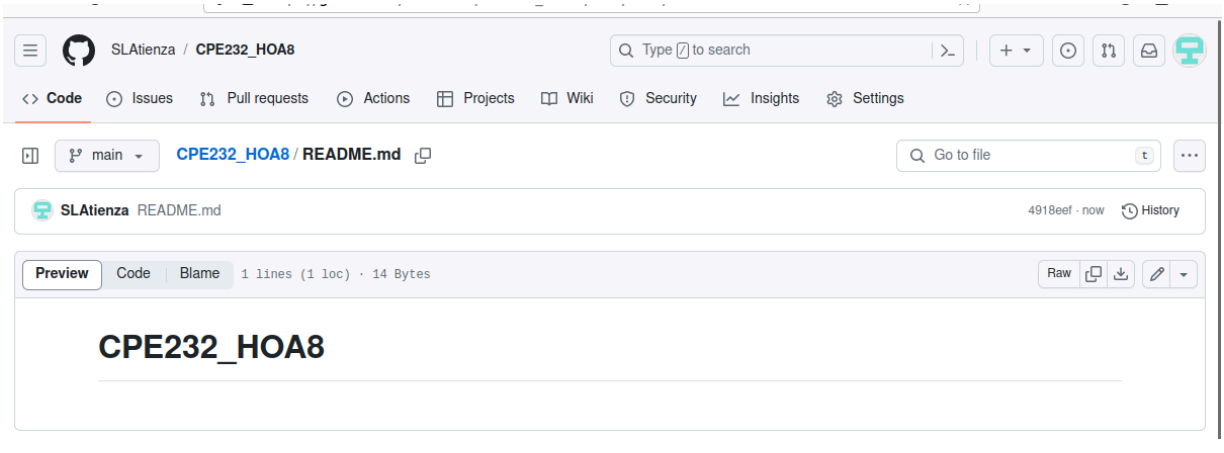
Availability monitoring is a type of monitoring tool that we use if the certain workload is up or reachable on our end. Site downtime can lead to loss of revenue, reputational damage and severe distress. Availability monitoring prevents adverse situations by checking the uptime of infrastructure components 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)

1. Create a New Repository and Clone your repository.



```
stephen@worksation:~$ git clone git@github.com:SLAtienza/CPE232_HOA8.git
Cloning into 'CPE232_HOA8'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
```

Creating Roles

```
  GNU nano 6.2                                ansible.cfg *
[defaults]

inventory = inventory
host_key_checking = False

deprecation_warning = False

remote_user = stephen
private_key_file = ~/.ssh/
```

Make site.yml

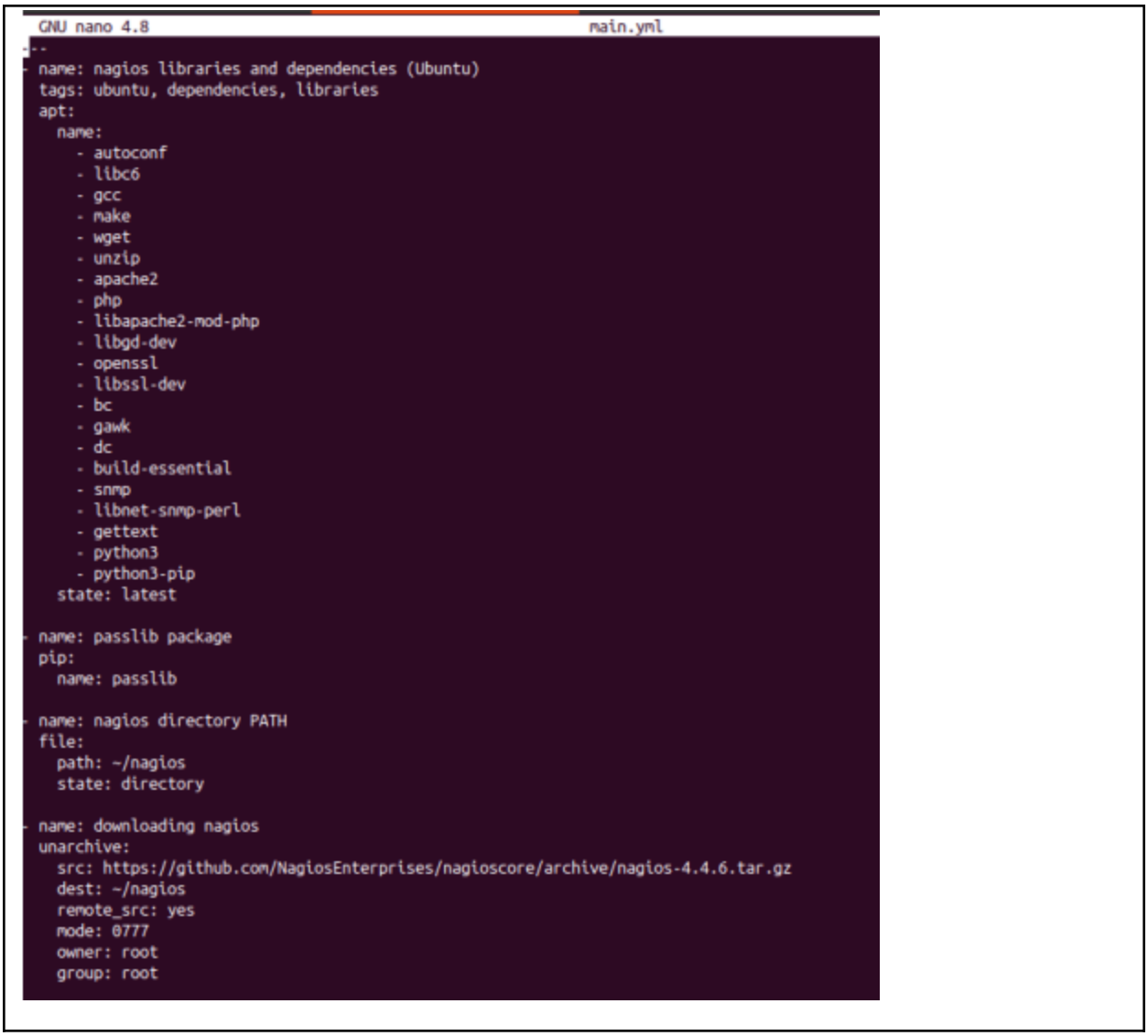
```
GNU nano 4.8                                                site.yml
---

- hosts: all
  become: true
  pre_tasks:

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

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

    - name: install updates (CentOS)
      yum:
        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_server
  become: true
  roles:
    - ubuntu_server

- hosts: centos_server
  become: true
  roles:
    - centos_server
```

Use command tree to check your directories

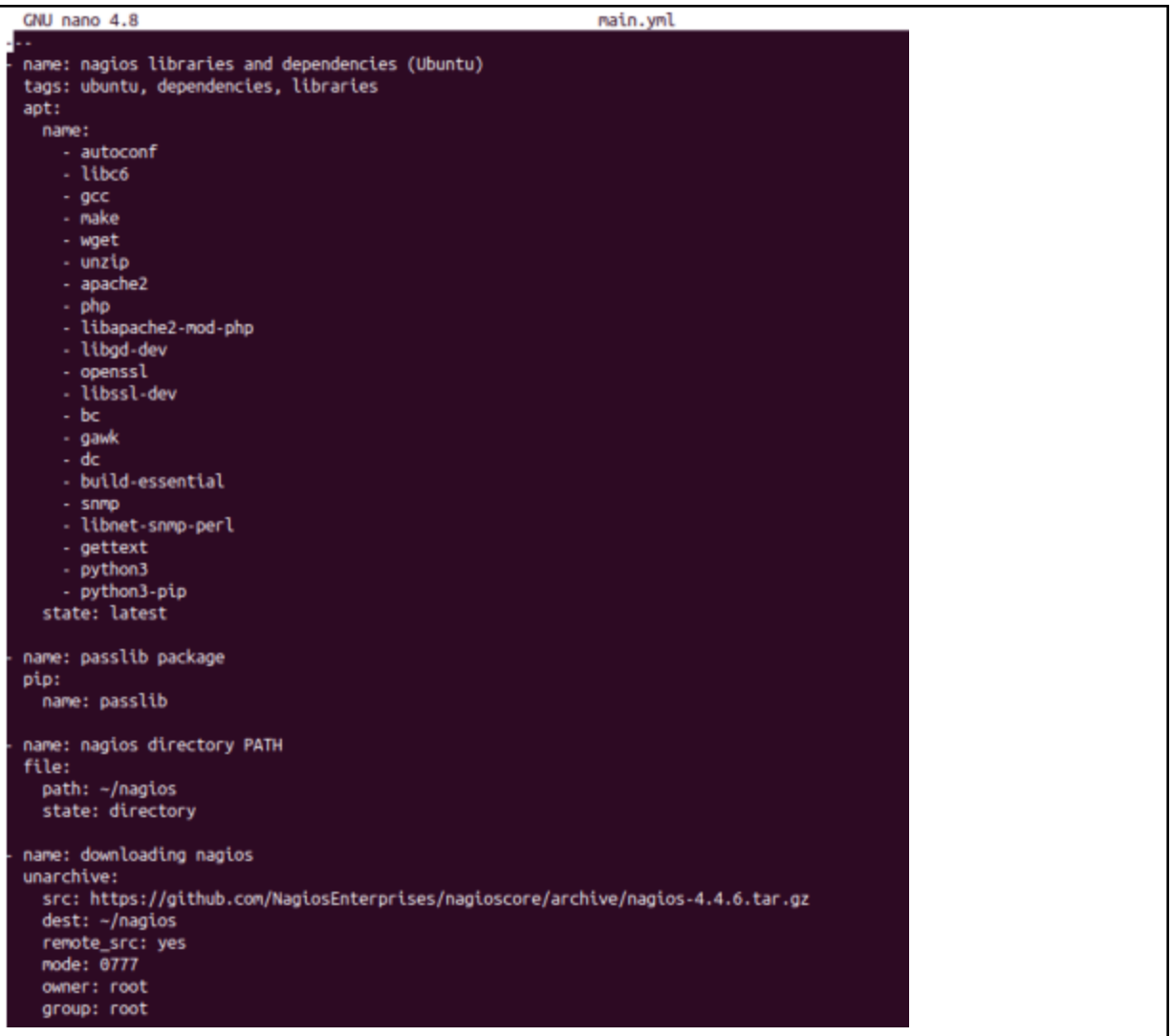
```
stephen@worksation:~/CPE232_HOA8/roles$ tree
.
├── centos_server
└── ubuntu_server
```

```
.
├── centos_server
│   └── tasks
│       └── main.yml
└── ubuntu_server
    └── tasks
        └── main.yml

4 directories, 2 files
```

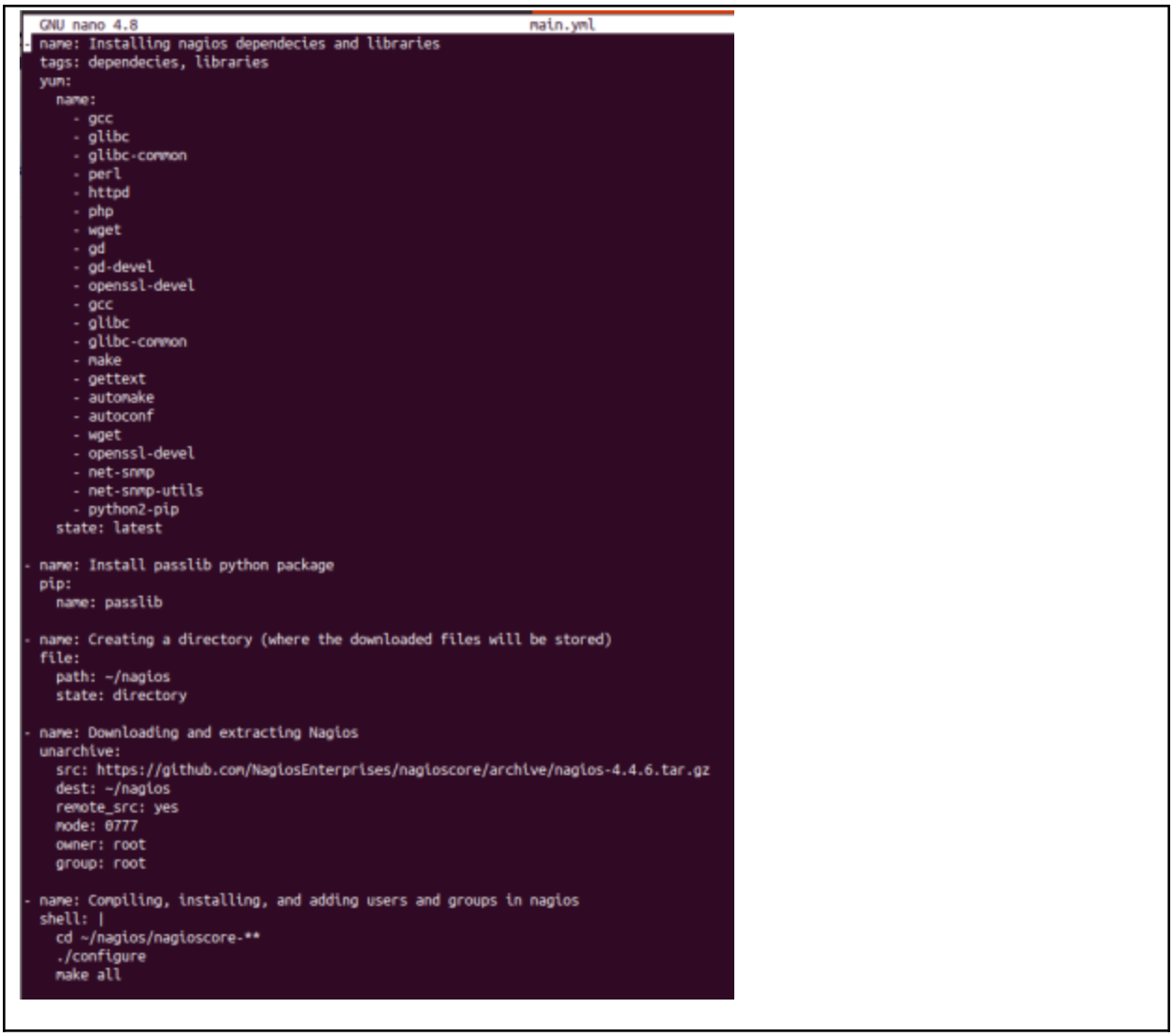
make ubuntu_server main.yml


GNU nano 4.8
main.yml

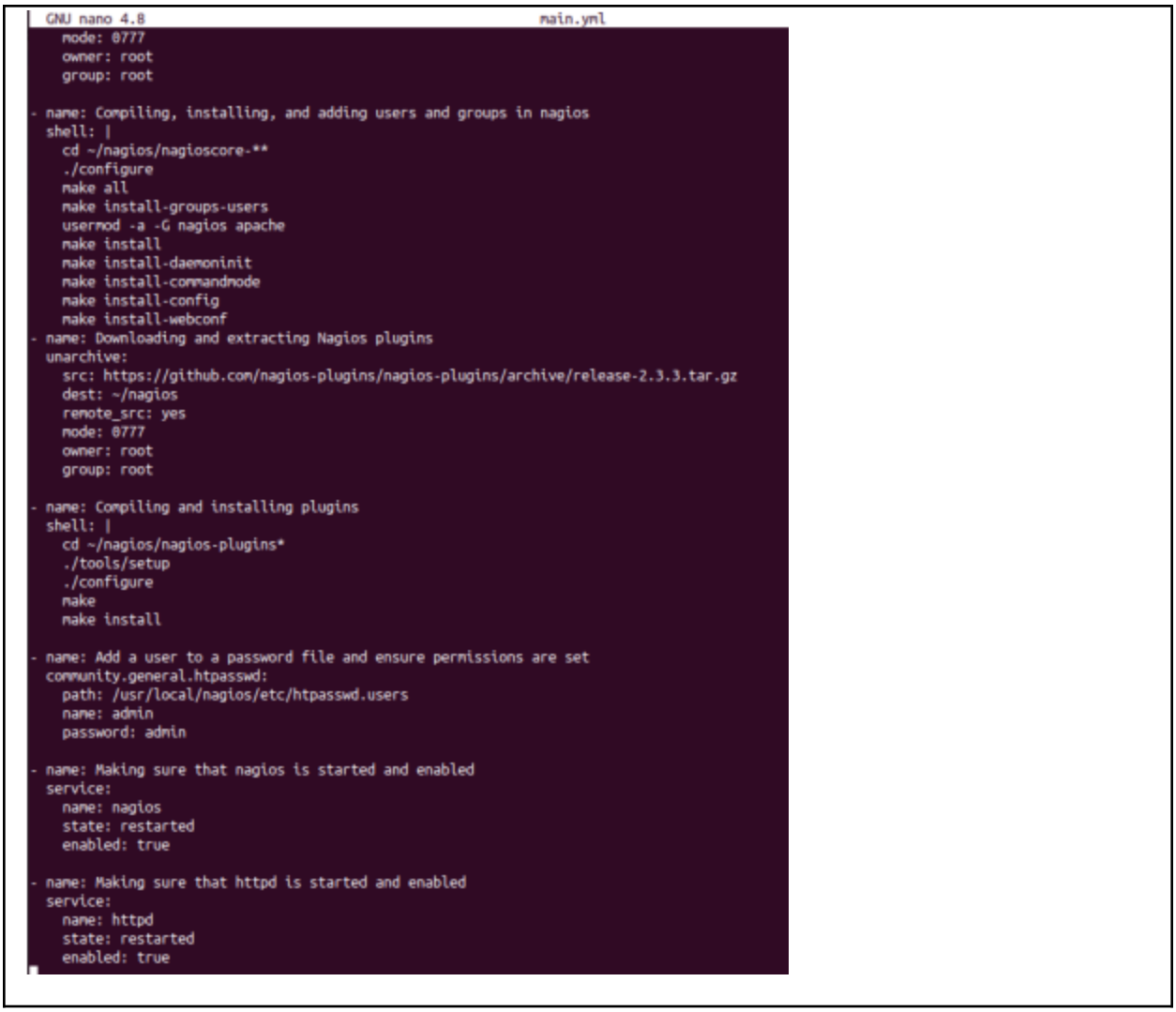
- name: nagios libraries and dependencies (Ubuntu)
tags: ubuntu, dependencies, libraries
apt:
name:
- autoconf
- libc6
- gcc
- make
- wget
- unzip
- apache2
- php
- libapache2-mod-php
- libgd-dev
- openssl
- libssl-dev
- bc
- gawk
- dc
- build-essential
- snmp
- libnet-snmp-perl
- gettext
- python3
- 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.tar.gz
dest: ~/nagios
remote_src: yes
mode: 0777
owner: root
group: root



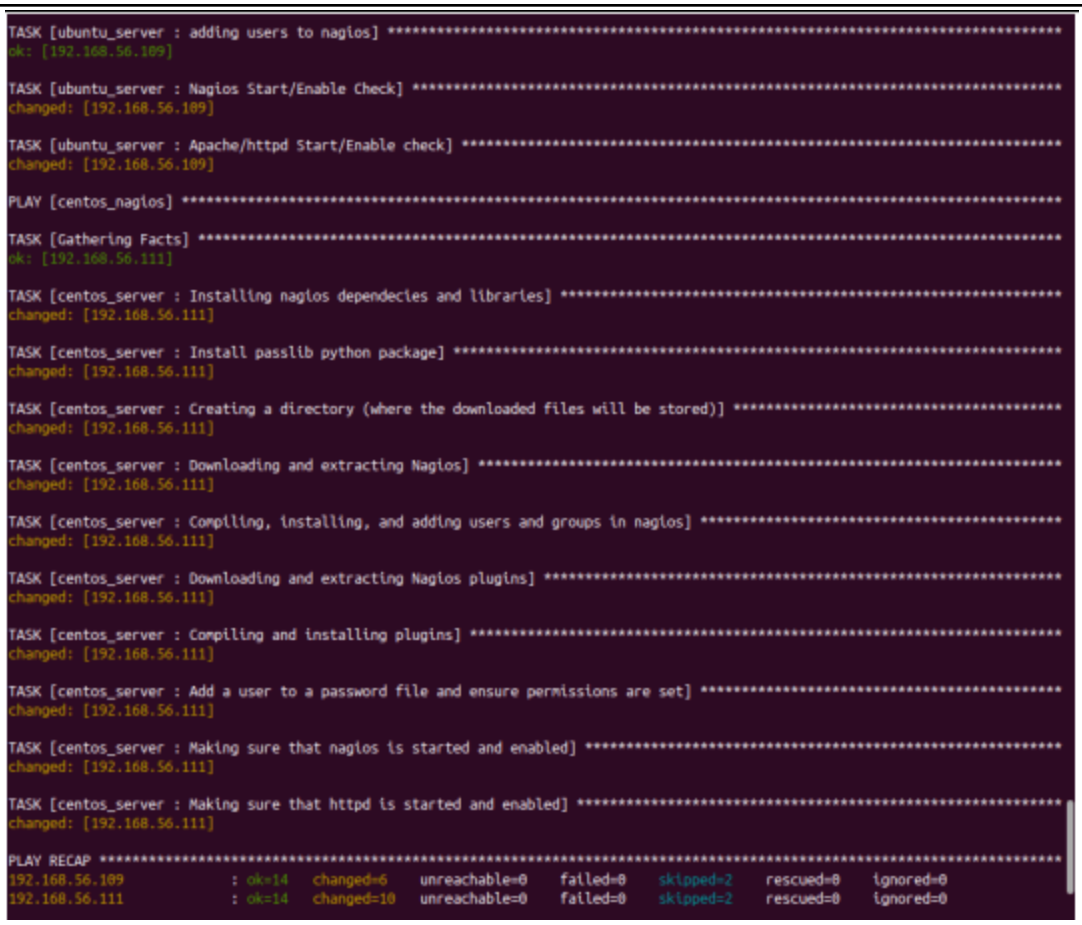
make centos_server main.yml



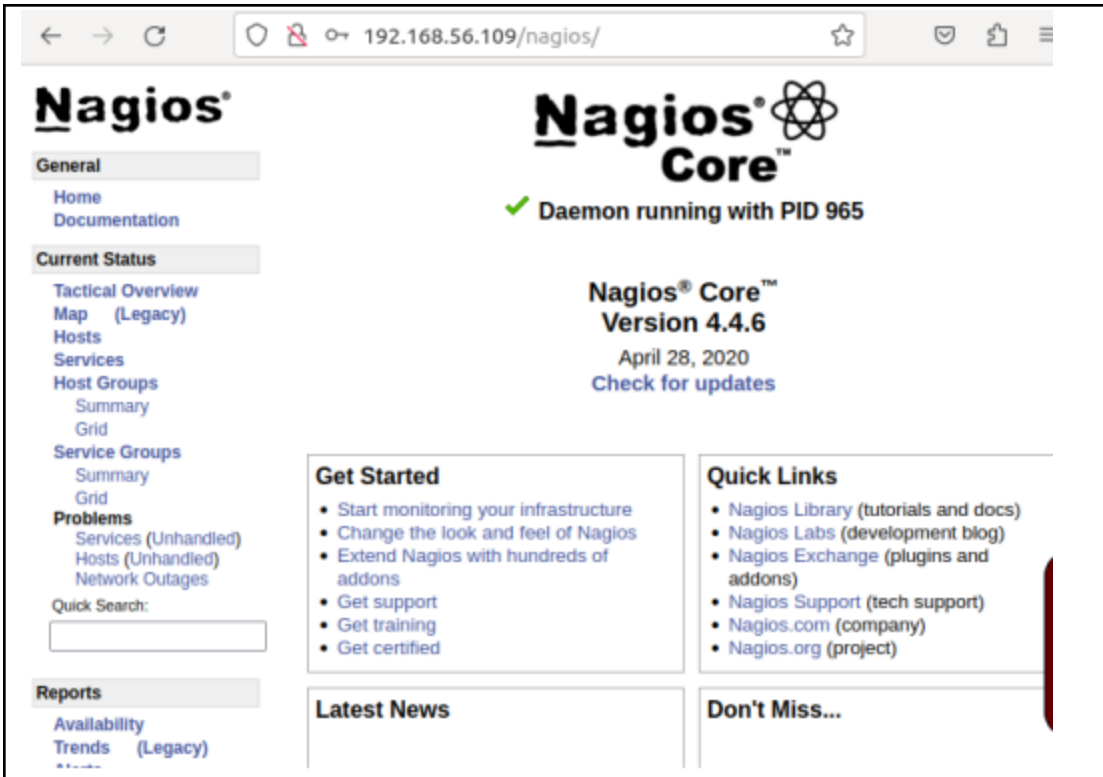
```
 GNU nano 4.8                                                main.yml
- name: Installing nagios dependecies and libraries
  tags: dependecies, libraries
  yum:
    name:
      - gcc
      - glibc
      - glibc-common
      - perl
      - httpd
      - php
      - wget
      - gd
      - gd-devel
      - openssl-devel
      - gcc
      - glibc
      - glibc-common
      - make
      - gettext
      - automake
      - autoconf
      - wget
      - openssl-devel
      - 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.tar.gz
    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
```



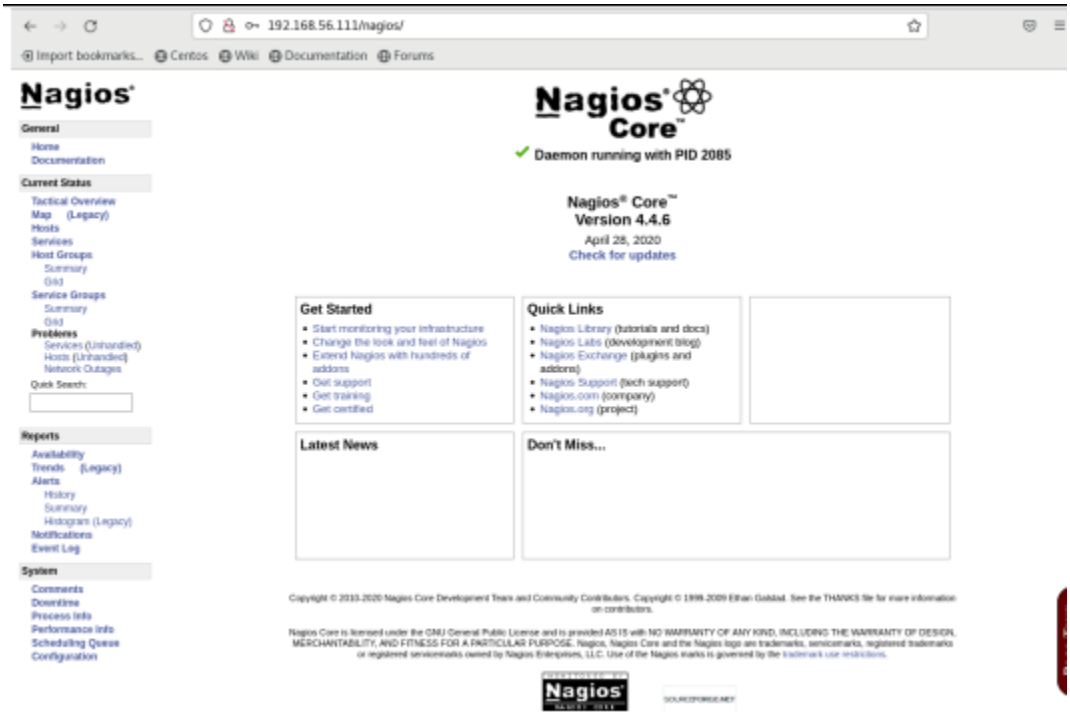
```
 GNU nano 4.8                                                main.yml
    mode: 0777
    owner: root
    group: root

- 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-2.3.3.tar.gz
    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
```



Ubuntu output



CentOS output



Reflections:

Answer the following:

1. What are the benefits of having an availability monitoring tool?
   -Organizations can benefit from availability monitoring solutions because they constantly monitor their IT infrastructure in real time, allowing for faster problem identification and less downtime. This, in turn, improves user experience, resource utilization for informed decisions, security, and compliance adherence.2 users are not using it. A robust and highly-performing IT infrastructure is achieved through historical data, automatic response, predictive analysis, and other technologies. This, in turn, leads to increased operational performance and customer satisfaction.

Conclusions:

- Using Ansible as an Infrastructure as Code (IaC) solution to deploy, configure, and manage enterprise monitoring technologies is, thus, highly recommended and essential. To automate infrastructure, this project has developed a script that uses the principles of role creation to intelligently install Nagios on CentOS and Ubuntu. Screenshots and extensive descriptions make this incredibly illustrative, giving a detailed tutorial on how to do the same. Verifying the workflow's efficacy and successfully deploying Nagios on both OSes highlights the importance of IaC in easing otherwise complex infrastructure management. In addition, creating a dedicated GitHub repository for this function encourages teamwork and information exchange, benefiting the operator and the public.