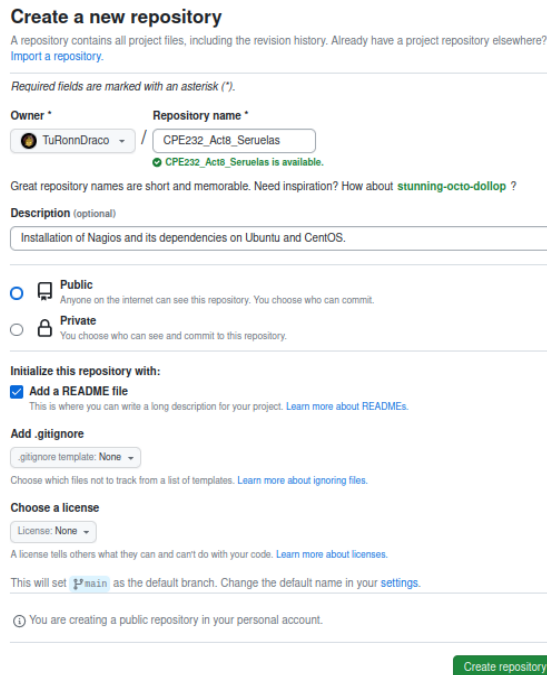


Name: Seruelas, Ronn Kristoper H.	Date Performed: 10-16-2023
Course/Section: CPE31S4	Date Submitted: 10-17-2023
Instructor: Dr. Jonathan V. Taylar	Semester and SY: 1st Sem 2022-2023
Activity 8: Install, Configure, and Manage Availability Monitoring tools	
1. Objectives	
Create and design a workflow that 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	
<ol style="list-style-type: none"> 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: Preparing a Dedicated repository

1. In your Github account, create a new repository that will be dedicated for the installation of Nagios for both your Ubuntu and CentOS server, and then clone it to your local repository in your workstation.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * TuRonnDraco / Repository name * CPE232_Act8_Seruelas

Great repository names are short and memorable. Need inspiration? How about [stunning-octo-dollop](#) ?

Description (optional)

Installation of Nagios and its dependencies on Ubuntu and CentOS.

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

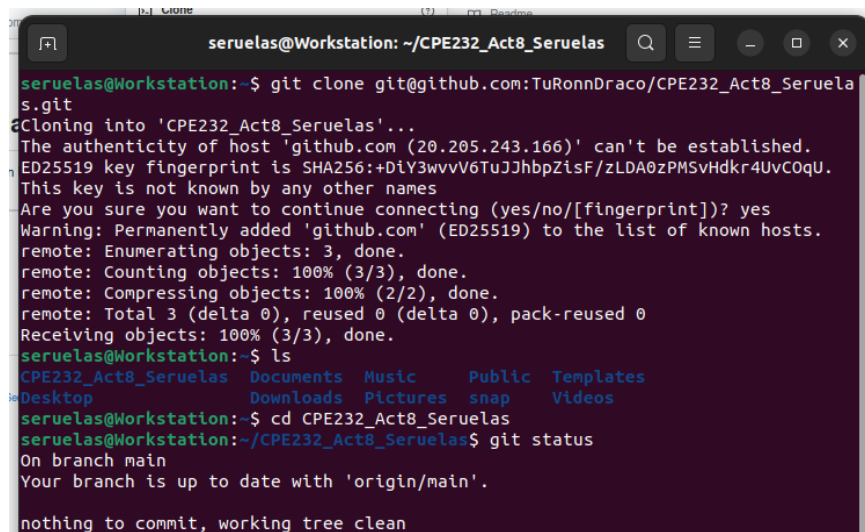
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

ⓘ You are creating a public repository in your personal account.

[Create repository](#)

Figure 1.1.1 - Creation of the CPE232_Act8_Seruelas repository.

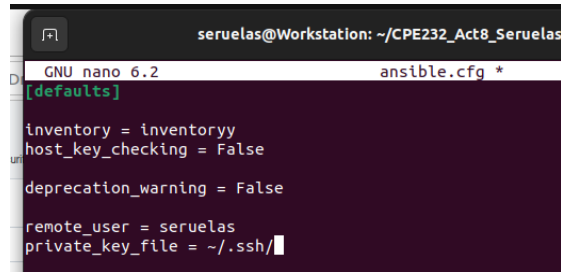


```
seruelas@Workstation: ~/$ git clone git@github.com:TuRonnDraco/CPE232_Act8_Seruelas.git
Cloning into 'CPE232_Act8_Seruelas'...
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCoQU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
seruelas@Workstation:~$ ls
CPE232_Act8_Seruelas  Documents  Music      Public  Templates
Desktop               Downloads  Pictures  snap    Videos
seruelas@Workstation:~$ cd CPE232_Act8_Seruelas
seruelas@Workstation:~/CPE232_Act8_Seruelas$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Figure 1.1.2 - Cloning of the github repository unto the local repository of the workstation.

2. In your cloned repository, create the **ansible.cfg** file that will contain the ansible configurations for your repository in order for it to be operable.



```
seruelas@Workstation: ~/CPE232_Act8_Seruelas
GNU nano 6.2 ansible.cfg *
[defaults]

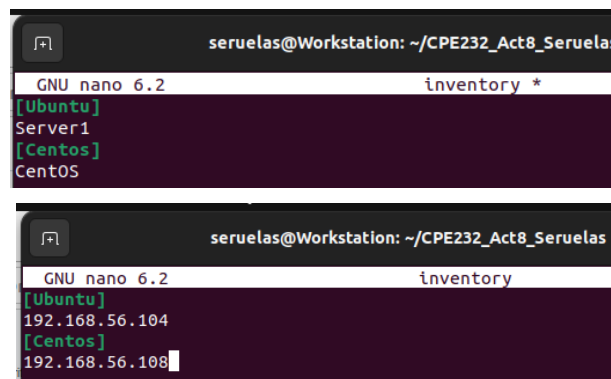
inventory = inventory
host_key_checking = False

deprecation_warning = False

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

Figure 1.2.1 - Configurations set for ansible.cfg

3. In your cloned repository, set up your **inventory** file by first doing **sudo nano inventory**, then inputting the proper hosts into their remote hosts. You can do this by inputting their respective IP address or their hostnames according to your workstation.

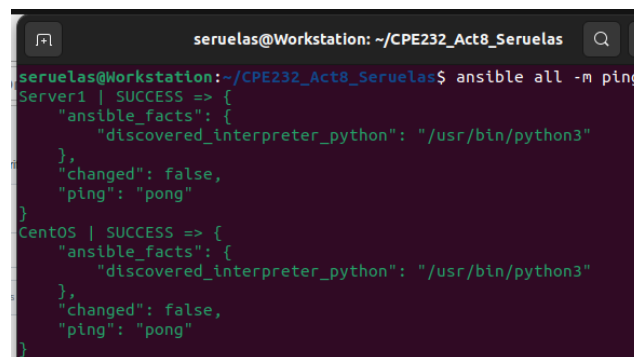


```
seruelas@Workstation: ~/CPE232_Act8_Seruelas
GNU nano 6.2 inventory *
[Ubuntu]
Server1
[Centos]
CentOS

seruelas@Workstation: ~/CPE232_Act8_Seruelas
GNU nano 6.2 inventory
[Ubuntu]
192.168.56.104
[Centos]
192.168.56.108
```

Figure 1.3.1-1.3.2 - Hosts for the inventory file and their two methodologies.

4. After configuring the ansible.cfg and inventory file, verify the connection between the workstation and the remote hosts by executing **ansible all -m ping**.



```
seruelas@Workstation: ~/CPE232_Act8_Seruelas
seruelas@Workstation:~/CPE232_Act8_Seruelas$ ansible all -m ping
Server1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
CentOS | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

Figure 1.4.1 - Verification of connecting between workstation and the hosts.

Task 2: Preparation of the playbook

1. To start off, first create a yml file using **sudo nano** that will serve as a playbook that will contain all the commands for installing nagios. In this activity, we will create the file, **install_nagios.yml**.

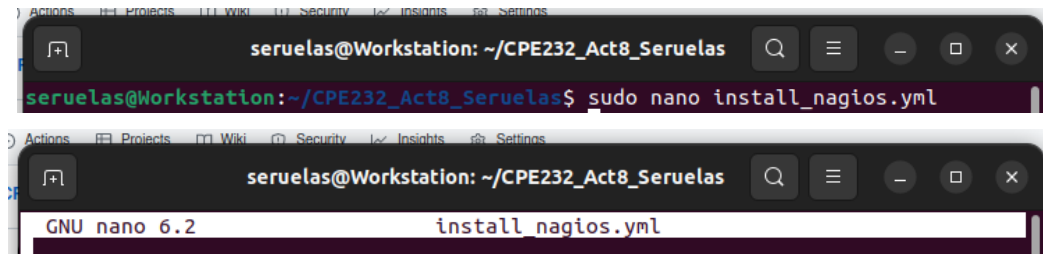


Figure 2.1.1-2.1.2 - Creation of the install_nagios.yml.

2. In the playbook, configure the playbook that it will run for all hosts, and that its designated pre-tasks is to update the repository index of Ubuntu and CentOS.

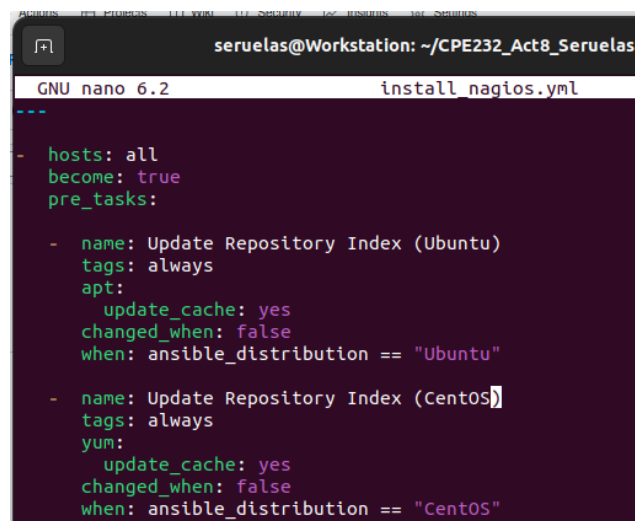


Figure 2.2.1 - Configuration of the playbook that will let it run on all hosts and update the repository indexes of both Ubuntu and CentOS.

3. To create a more efficient and cleaner playbook, we are to divide the tasks of installing nagios into multiple roles. First, we must create a new directory named **roles**, that will contain the different roles that will have their own tasks. Inside of the roles directory will contain the two different roles, Ubuntu and CentOS, within their directories will have another directory named **tasks**, and it will contain the **main.yml**.

```
seruelas@Workstation: ~/CPE232_Act8_Seruelas
seruelas@Workstation:~/CPE232_Act8_Seruelas$ mkdir roles
seruelas@Workstation:~/CPE232_Act8_Seruelas$ cd roles
seruelas@Workstation:~/CPE232_Act8_Seruelas/roles$ mkdir Ubuntu
seruelas@Workstation:~/CPE232_Act8_Seruelas/roles$ mkdir CentOS
seruelas@Workstation:~/CPE232_Act8_Seruelas/roles$ cd Ubuntu
seruelas@Workstation:~/CPE232_Act8_Seruelas/roles/Ubuntu$ mkdir tasks
seruelas@Workstation:~/CPE232_Act8_Seruelas/roles/Ubuntu$ cd tasks
seruelas@Workstation:~/CPE232_Act8_Seruelas/roles/Ubuntu/tasks$ touch main.yml
seruelas@Workstation:~/CPE232_Act8_Seruelas/roles/Ubuntu/tasks$ ..
..: command not found
seruelas@Workstation:~/CPE232_Act8_Seruelas/roles/Ubuntu/tasks$ .
bash: ..: filename argument required
..: usage: . filename [arguments]
seruelas@Workstation:~/CPE232_Act8_Seruelas/roles/Ubuntu/tasks$ cd .
seruelas@Workstation:~/CPE232_Act8_Seruelas/roles/Ubuntu/tasks$ cd .
seruelas@Workstation:~/CPE232_Act8_Seruelas/roles/Ubuntu/tasks$ cd ..
seruelas@Workstation:~/CPE232_Act8_Seruelas/roles/Ubuntu$ cd ..
seruelas@Workstation:~/CPE232_Act8_Seruelas/roles$ cd CentOS
seruelas@Workstation:~/CPE232_Act8_Seruelas/roles/CentOS$ mkdir tasks
seruelas@Workstation:~/CPE232_Act8_Seruelas/roles/CentOS$ cd tasks
seruelas@Workstation:~/CPE232_Act8_Seruelas/roles/CentOS/tasks$ touch main.yml
seruelas@Workstation:~/CPE232_Act8_Seruelas/roles/CentOS/tasks$ cd ..
seruelas@Workstation:~/CPE232_Act8_Seruelas/roles/CentOS$ cd ..
seruelas@Workstation:~/CPE232_Act8_Seruelas/roles$ cd ..
seruelas@Workstation:~/CPE232_Act8_Seruelas$ tree roles
roles
├── CentOS
│   └── tasks
│       └── main.yml
└── Ubuntu
    └── tasks
        └── main.yml

4 directories, 2 files
```

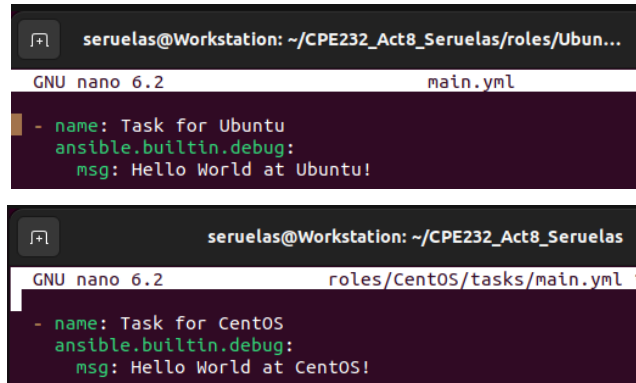
Figure 2.3.1 - Creation of the roles directory, containing different roles with different tasks.

4. After the creation of roles, modify the playbook in which in their own hosts, they will execute only the commands of their own roles.

```
seruelas@Workstation: ~/CPE232_Act8_Seruelas
File Edit View Search Terminal Help
GNU nano 6.2 install nagios.yml
---
- hosts: all
  become: true
  pre_tasks:
    - name: Update Repository Index
      tags: always
      package:
        update_cache: yes
        changed_when: false
  - hosts: Ubuntu
    become: true
    roles:
      - Ubuntu
  - hosts: CentOS
    become: true
    roles:
      - CentOS
```

Figure 2.4.1 - Adding the necessary commands to be executed in order to allow the hosts to only execute the commands of their own roles.

5. To verify if the roles are working as intended, modify the **main.yml** of each role to execute a simple command.

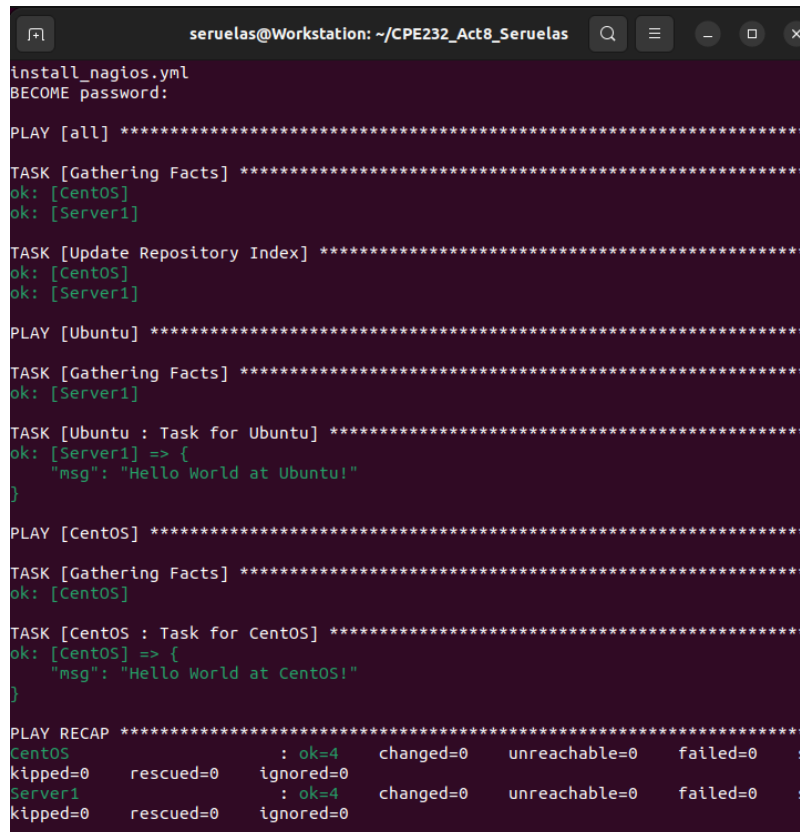


```
seruelas@Workstation: ~/CPE232_Act8_Seruelas/roles/Ubun...
GNU nano 6.2 main.yml
- name: Task for Ubuntu
  ansible.builtin.debug:
    msg: Hello World at Ubuntu!

seruelas@Workstation: ~/CPE232_Act8_Seruelas
GNU nano 6.2 roles/CentOS/tasks/main.yml *
- name: Task for CentOS
  ansible.builtin.debug:
    msg: Hello World at CentOS!
```

Figure 2.5.1-2.5.2 - Adding simple commands to main.yml of each role.

6. After adding some simple commands to each tasks of the role, execute the playbook by using **ansible-playbook -ask-become-pass <name of playbook>**.



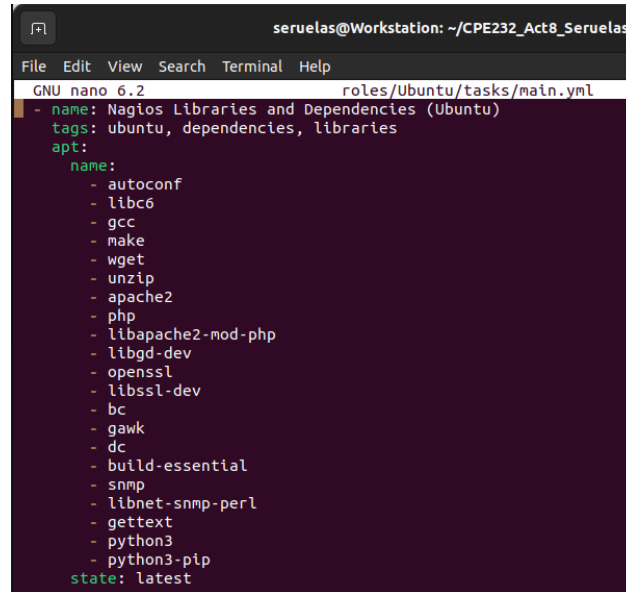
```
seruelas@Workstation: ~/CPE232_Act8_Seruelas
install_nagios.yml
BECOME password:

PLAY [all] *****
TASK [Gathering Facts] *****
ok: [CentOS]
ok: [Server1]
TASK [Update Repository Index] *****
ok: [CentOS]
ok: [Server1]
PLAY [Ubuntu] *****
TASK [Gathering Facts] *****
ok: [Server1]
TASK [Ubuntu : Task for Ubuntu] *****
ok: [Server1] => {
  "msg": "Hello World at Ubuntu!"
}
PLAY [CentOS] *****
TASK [Gathering Facts] *****
ok: [CentOS]
TASK [CentOS : Task for CentOS] *****
ok: [CentOS] => {
  "msg": "Hello World at CentOS!"
}
PLAY RECAP *****
centos      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
server1     : ok=4    changed=0    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
```

Figure 2.6.1 - Execution of the playbook for testing and verification of setup.

Task 3: Installation of Libraries and Dependencies

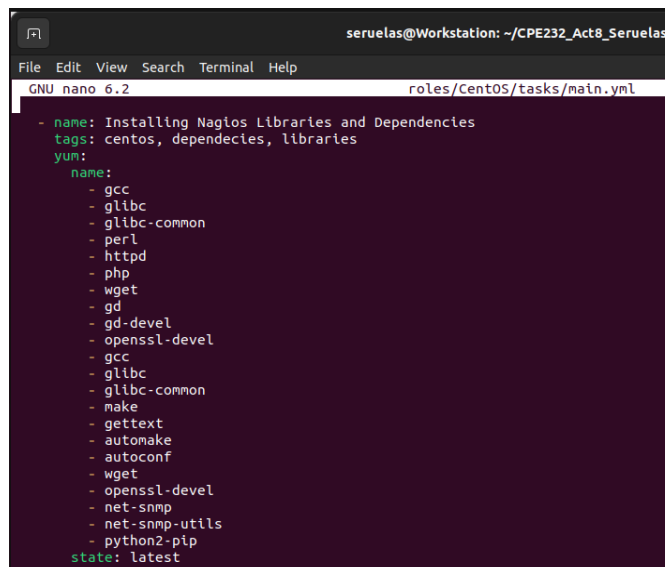
1. In Ubuntu, there are a total of 21 packages that can be installed that are the dependencies of Nagios. In the main.yml file of the Ubuntu role, input every package that is considered as a dependency for the Nagios service.



```
seruelas@Workstation: ~/CPE232_Act8_Seruelas
File Edit View Search Terminal Help
GNU nano 6.2 roles/Ubuntu/tasks/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
```

Figure 3.1.1 - Installation of Nagios' libraries and dependencies in Ubuntu.

2. In CentOS, there are a total of 22 packages that are considered as a part of the library or dependency needed by Nagios. In the main.yml of CentOS role, input the installation of every package that is considered as a dependency or library for Nagios.



```
seruelas@Workstation: ~/CPE232_Act8_Seruelas
File Edit View Search Terminal Help
GNU nano 6.2 roles/CentOS/tasks/main.yml
- name: Installing Nagios Libraries and Dependencies
  tags: centos, dependencies, 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
```

Figure 3.2.1 - Installation of Nagios' libraries and dependencies in CentOS.

Task 4: Installing Nagios4 (Ubuntu Role)

1. Install the **nagios4** and **nagios-plugins** package by including it in the Ubuntu Roles playbook.

```
- name: Install Nagios4
  become: true
  apt:
    name:
      - nagios4
      - nagios-plugins
    state: latest
```

Figure 4.1.1 - Module that will install Nagios4 into the remote host.

2. For Nagios to become operable, we need to configure the apache web server with the **a2enmod**. By using the builtin command **shell** for our playbooks, use the **sudo a2enmod rewrite cgi** to successfully reconfigure our apache server for Nagios4, after that, create a module that will restart the apache web server service.

```
- name: Configure Apache Web Server
  become: true
  shell:
    sudo a2enmod rewrite cgi

- name: Restart Apache Server after Configuration
  service:
    name: apache2
    state: restarted
    enabled: true
```

Figure 4.2.1 - Module that will configure the apache web server for Nagios4, with the module that will restart the Apache2 server after its configuration for Nagios4.

3. Finally, create a module that will enable and start the Nagios4 service.

```
- name: Enable Nagios
  service:
    name: nagios4
    state: restarted
    enabled: true
```

Figure 4.3.1 - Module that enables and starts the Nagios4 service.


```
seruelas@Workstation: ~/CPE232_Act8_Seruelas
GNU nano 6.2 roles/Ubuntu/tasks/main.yml
- name: Nagios Libraries and Dependencies (Ubuntu)
  tags: ubuntu, dependencies, libraries
  apt:
    name:
      - autoconf
      - libc6
      - gcc
      - make
      - wget
      - unzip
      - apache2
      - php
      - libapache2-mod-php7.2
      - libgd-dev
      - openssl
      - libssl-dev
      - bc
      - gawk
      - dc
      - build-essential
      - snmp
      - libnet-snmp-perl
      - gettext
      - python3
      - python3-pip
    state: latest

- name: Install Nagios4
  become: true
  apt:
    name:
      - nagios4
      - nagios-plugins
    state: latest

- name: Configure Apache Web Server
  become: true
  shell:
    sudo a2enmod rewrite cgi

- name: Restart Apache Server after Configuration
  service:
    name: apache2
    state: restarted
    enabled: true

- name: Enable Nagios
  service:
    name: nagios4
    state: restarted
    enabled: true
```

Figure 4.5 - Main.yml of the Ubuntu Root Tasks.

Task 5: Installing Nagios (CentOS Role)

1. Create a directory in which it will be the dedicated directory for Nagios.

```
- name: Creating the Nagios Directory
  file:
    path: ~/nagios
    state: directory
```

Figure 5.1.1 - Module that will create the directory for Nagios.

2. Create a module in which it will download the Nagios file from a repository, and will extract it to the directory.

```
- name: Installation of 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
```

Figure 5.2.1 - Module responsible for downloading and extracting the files of Nagios to the directory.

3. Create a module in which it will download and extract the plugins for the Nagios service directly to the directory.

```
- name: Installation of 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
```

Figure 5.3.1 - Module responsible for downloading and extracting the Nagios Plugins.

4. Create a module in which it will configure the Nagios service file in order for it to become operable, in which it will require a user for it to be operable.

```
- name: Compilation, Installation and User Modification
  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
```

Figure 5.4.1 - Module that is responsible for configuring and creating a user for Nagios.

5. Create a module that will install all the plugins downloaded for Nagios and that will configure them to Nagios.

```
- name: Compilation and Installation of Plugins
shell: |
  cd ~/nagios/nagios-plugins*
  ./tools/setup
  ./configure
  make
  make install
```

Figure 5.5.1 - Installation and configuration of the plugins for Nagios.

6. Create a module that creates a password for the user that will access the Nagio service from the browser.

```
- name: Adding a Password to the User/s
community.general.htpasswd:
  path: /usr/local/nagios/etc/htpasswd.users
  name: admin
  password: admin
```

Figure 5.6.1 - Module responsible for setting a password for the administrator.

7. Create a module that will restart and re-enable the HTTPD and the Nagio services.

```
- name: Enabling HTTPD
service:
  name: httpd
  state: restarted
  enabled: true

- name: Enabling Nagios
service:
  name: nagios
  state: restarted
  enabled: true
```

Figure 5.7.1 - Module's that will restart and re-enable the HTTPD and the Nagios services.

```
seruelas@Workstation: ~/CPE232_Act8_Seruelas
File Edit View Search Terminal Help
GNU nano 6.2 roles/CentOS/tasks/main.yml *

- name: Installing Nagios Libraries and Dependencies
  tags: centos, dependencies, 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: Creating the Nagios Directory
  file:
    path: ~/nagios
    state: directory

- name: Installation of 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: Installation of 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: Compilation, Installation and User Modification
  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: Compilation and Installation of Plugins
  shell: |
    cd ~/nagios/nagios-plugins*
    ./tools/setup
    ./configure
    make
    make install

- name: Adding a Password to the User/s
  community.general.htpasswd:
    path: /usr/local/nagios/etc/htpasswd.users
    name: admin
    password: admin

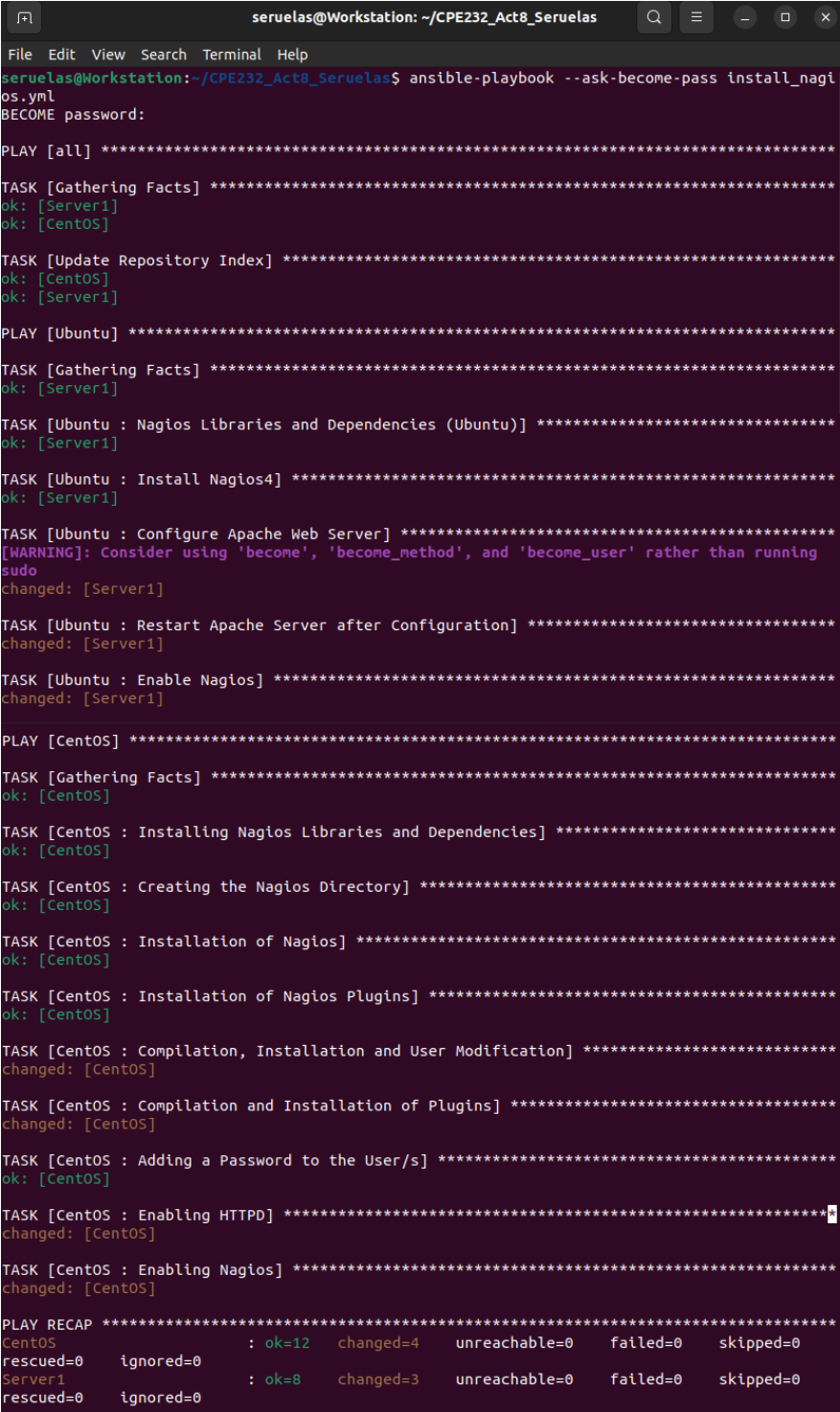
- name: Enabling HTTPD
  service:
    name: httpd
    state: restarted
    enabled: true

- name: Enabling Nagios
  service:
    name: nagios
    state: restarted
    enabled: true
```

Figure 5.8 - Whole Main.yml of CentOS role.

Task 7: Verification of Playbook

1. Run the playbook and show its output.



```
seruelas@Workstation: ~/CPE232_Act8_Seruelas
File Edit View Search Terminal Help
seruelas@Workstation:~/CPE232_Act8_Seruelas$ ansible-playbook --ask-become-pass install_nagios.yml
BECOME password:

PLAY [all] *****
TASK [Gathering Facts] *****
ok: [Server1]
ok: [CentOS]

TASK [Update Repository Index] *****
ok: [CentOS]
ok: [Server1]

PLAY [Ubuntu] *****
TASK [Gathering Facts] *****
ok: [Server1]

TASK [Ubuntu : Nagios Libraries and Dependencies (Ubuntu)] *****
ok: [Server1]

TASK [Ubuntu : Install Nagios4] *****
ok: [Server1]

TASK [Ubuntu : Configure Apache Web Server] *****
[WARNING]: Consider using 'become', 'become_method', and 'become_user' rather than running sudo
changed: [Server1]

TASK [Ubuntu : Restart Apache Server after Configuration] *****
changed: [Server1]

TASK [Ubuntu : Enable Nagios] *****
changed: [Server1]

PLAY [CentOS] *****
TASK [Gathering Facts] *****
ok: [CentOS]

TASK [CentOS : Installing Nagios Libraries and Dependencies] *****
ok: [CentOS]

TASK [CentOS : Creating the Nagios Directory] *****
ok: [CentOS]

TASK [CentOS : Installation of Nagios] *****
ok: [CentOS]

TASK [CentOS : Installation of Nagios Plugins] *****
ok: [CentOS]

TASK [CentOS : Compilation, Installation and User Modification] *****
changed: [CentOS]

TASK [CentOS : Compilation and Installation of Plugins] *****
changed: [CentOS]

TASK [CentOS : Adding a Password to the User/s] *****
ok: [CentOS]

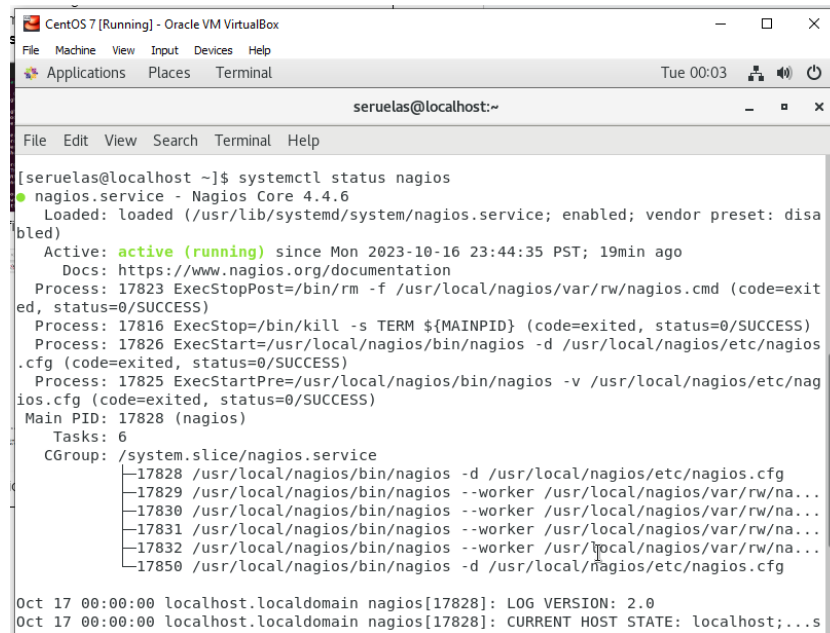
TASK [CentOS : Enabling HTTPD] *****
changed: [CentOS]

TASK [CentOS : Enabling Nagios] *****
changed: [CentOS]

PLAY RECAP *****
CentOS      : ok=12  changed=4  unreachable=0  failed=0  skipped=0
rescued=0   ignored=0
Server1     : ok=8   changed=3  unreachable=0  failed=0  skipped=0
rescued=0   ignored=0
```

Figure 7.1.1 - Successful run of the install_nagios.yml playbook.

2. Verify the installation of Nagios was successful in the CentOS by executing **systemctl status nagios**, and opening the browser at **<ip address of your host>/nagios/**.



```
CentOS 7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Tue 00:03
seruelas@localhost:~

[seruelas@localhost ~]$ systemctl status nagios
● nagios.service - Nagios Core 4.4.6
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2023-10-16 23:44:35 PST; 19min ago
     Docs: https://www.nagios.org/documentation
   Process: 17823 ExecStopPost=/bin/rm -f /usr/local/nagios/var/rw/nagios.cmd (code=exit
ed, status=0/SUCCESS)
   Process: 17816 ExecStop=/bin/kill -s TERM ${MAINPID} (code=exited, status=0/SUCCESS)
   Process: 17826 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios
.cfg (code=exited, status=0/SUCCESS)
   Process: 17825 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nag
ios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 17828 (nagios)
      Tasks: 6
   CGroup: /system.slice/nagios.service
           └─17828 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             17829 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/na...
             17830 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/na...
             17831 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/na...
             17832 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/na...
             17850 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 17 00:00:00 localhost.localdomain nagios[17828]: LOG VERSION: 2.0
Oct 17 00:00:00 localhost.localdomain nagios[17828]: CURRENT HOST STATE: localhost;...
```

Figure 7.2.1 - Verification of service Nagios running through systemctl.

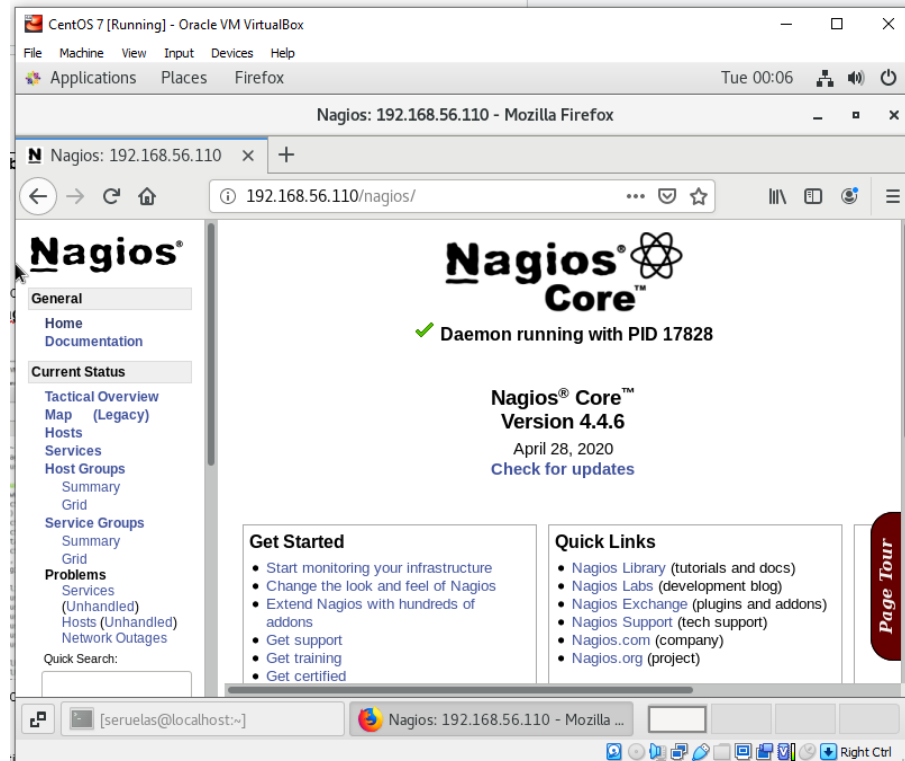


Figure 7.2.2 - Nagios Service opened at CentOS firefox browser.

3. Verify that the installation of Nagios4 was successful in the Ubuntu server by executing the command, **nagios4 -version**, and opening the browser with the url **<ip address of your host>/nagios4/**.

```
seruelas@Server1:~$ nagios4 --version
Nagios Core 4.4.6
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2020-04-28
License: GPL

Website: https://www.nagios.org

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License version 2 as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
```

Figure 4.4.1 - Verification of installation of Nagios4 in Ubuntu Server.

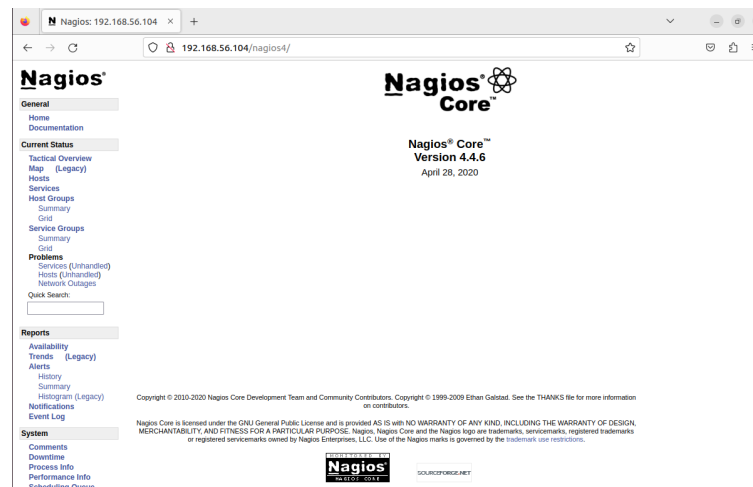
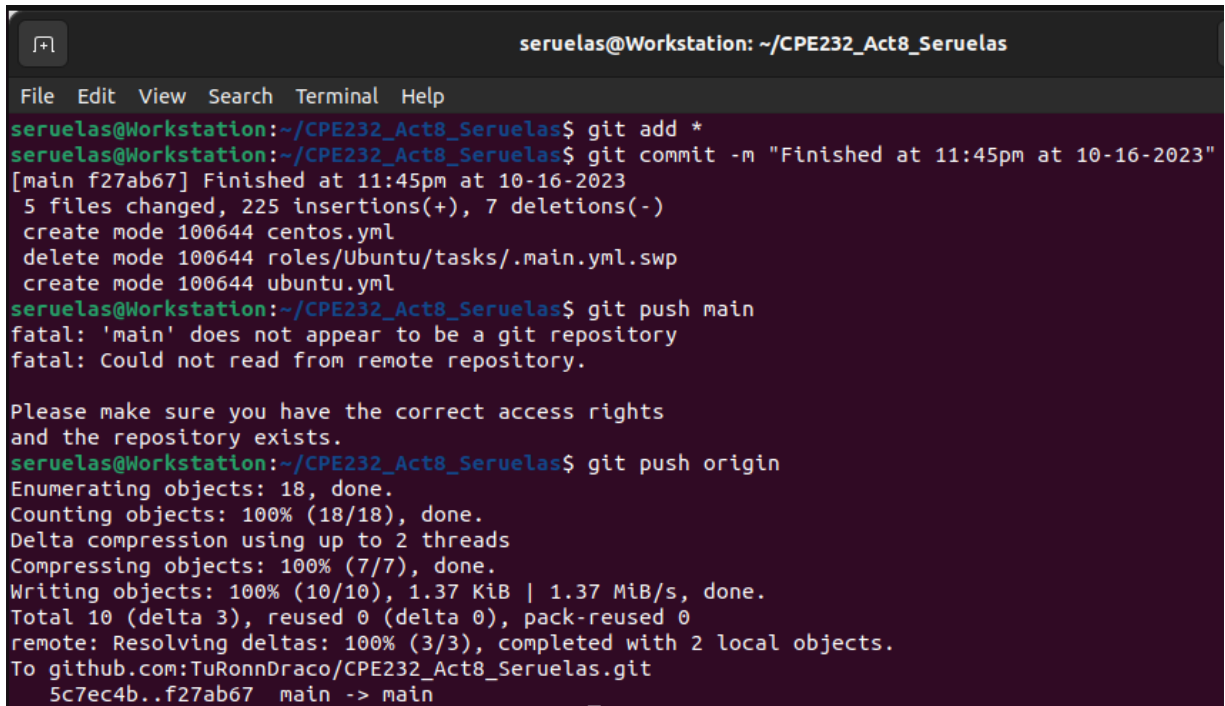


Figure 4.4.2 - Nagios4 Core via browser. (192.168.56.104/nagios4/)

The screenshot shows the 'Service Status Details For All Hosts' page in the Nagios Core 4.4.6 web interface. The page displays a table of service status details for the host 'localhost'. The table has columns for Host, Service, Status, Last Check, Duration, Attempt, and Status Information. The status of all services is 'OK'. The table lists various services including Current Load, Current Users, HTTP, PING, Root Partition, SSH, Swap Usage, and Total Processes. The status information for each service is provided in the last column, such as 'OK - load average: 3.25, 0.88, 0.30' for Current Load and 'OK - 1 users currently logged in' for Current Users. The page also includes a sidebar with navigation links and a top section with 'Current Network Status' and 'Host Status Totals'.

Figure 4.4.3 - Checking the services running tracked by Nagios4

4. Save and Push the local repository to the Github Repository



```
seruelas@Workstation: ~/CPE232_Act8_Seruelas
File Edit View Search Terminal Help
seruelas@Workstation:~/CPE232_Act8_Seruelas$ git add *
seruelas@Workstation:~/CPE232_Act8_Seruelas$ git commit -m "Finished at 11:45pm at 10-16-2023"
[main f27ab67] Finished at 11:45pm at 10-16-2023
 5 files changed, 225 insertions(+), 7 deletions(-)
 create mode 100644 centos.yml
 delete mode 100644 roles/Ubuntu/tasks/.main.yml.swp
 create mode 100644 ubuntu.yml
seruelas@Workstation:~/CPE232_Act8_Seruelas$ git push main
fatal: 'main' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
seruelas@Workstation:~/CPE232_Act8_Seruelas$ git push origin
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 2 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (10/10), 1.37 KiB | 1.37 MiB/s, done.
Total 10 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:TuRonnDraco/CPE232_Act8_Seruelas.git
 5c7ec4b..f27ab67  main -> main
```

Figure 7.4.1 - Adding, Committing, and Pushing all changes to the Github Repository.

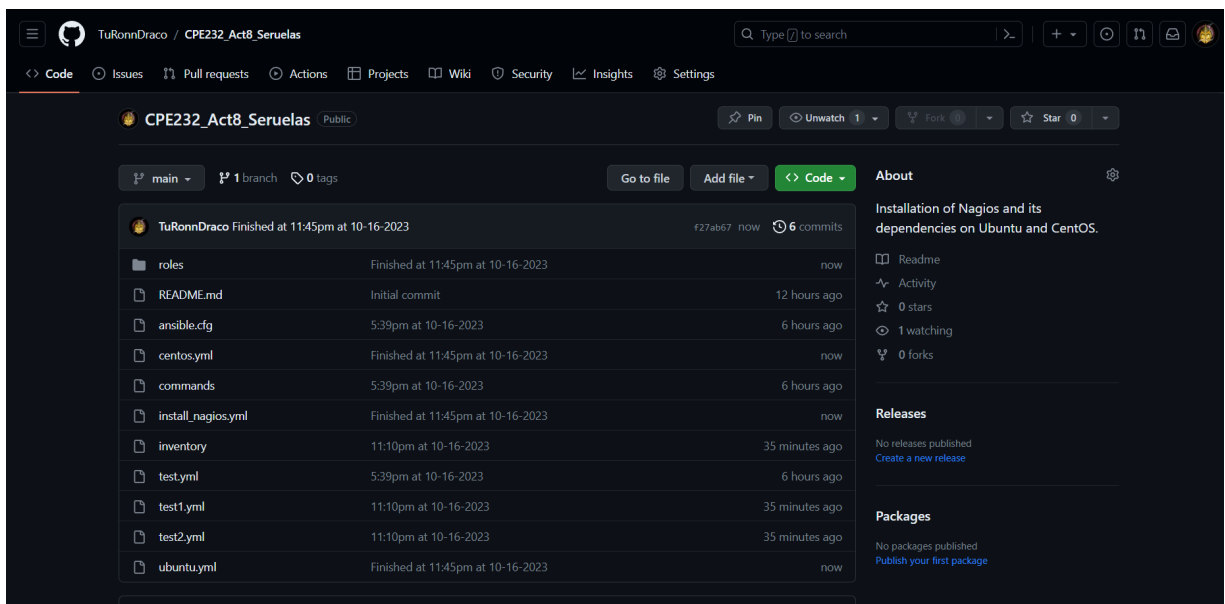


Figure 7.4.2 - Verification of all changes and files added to Github.

https://github.com/TuRonnDraco/CPE232_Act8_Seruelas

Reflections:

Answer the following:

1. What are the benefits of having an availability monitoring tool?
 - The benefits of having an availability monitoring tool is that it allows the administrators to be able to keep track of the services that are currently running in their services, allowing them to know what is running and what services are idle. With this tool, they are able to keep track and they are able to consistently monitor what services are running, unstable and possibly that can lead to stopping any time soon, allowing them to move efficiently.

Conclusions:

In this activity, we were able to learn about the importance of having an availability monitoring tool in system administration and how necessary it is to have one. We have educated ourselves that there are multiple variants of an availability monitoring tool and there are ones that suit best according to different operating systems. In this activity, we were put into a test in which we are able to test our knowledge on creating a playbook with the lessons that we have learned from previous activities. In this activity, we were able to implement the basics of creating a playbook, and also utilizing roles in our playbook in order to create a more efficient and memory saving playbook. In this activity, we were to face a challenge in which we are to install an availability monitoring tool that requires multiple steps in order to make it operable and functional. In the end of the activity, we are able to conclude that we are able to implement the knowledge that we have learned from our previous activities in creating the playbook in this activity.