
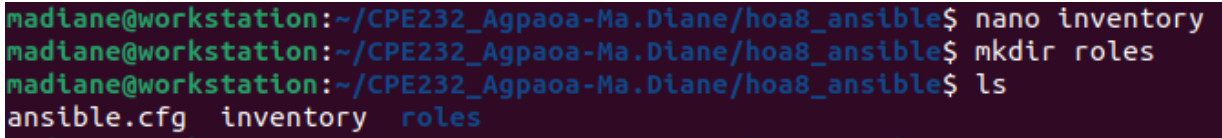



Name: Agpaoa, Ma.Diane J.	Date Performed: 10/18/2022
Course/Section: CPE232-CPE31S22	Date Submitted: 10/19/2022
Instructor: Dr. Jonathan Taylar	Semester and SY: 1st Sem 2022-2023
Activity 8: Install, Configure, and Manage Availability Monitoring tools	
1. Objectives	
Create and design a workflow that 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 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. Output (screenshots and explanations)	
Step 1: Creating the main directory and its required contents	
 <pre>madiane@workstation:~/CPE232_Agpaoa-Ma.Diane\$ mkdir hoa8_ansible</pre>	
<p>Figure 1.1 Creating a new directory for Activity 8</p> <p>By using the command mkdir I created a directory and named it as hoa8_ansible. This directory will contain the playbook that installs Nagios in both Ubuntu and CentOS, inventory and roles.</p>	
 <pre>madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/hoa8_ansible\$ nano inventory madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/hoa8_ansible\$ mkdir roles madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/hoa8_ansible\$ ls ansible.cfg inventory roles</pre>	
<p>Figure 1.2 Creating the file inventory and the roles directory</p>	

Using the command nano I created the inventory file that will contain the group of remote servers. In addition, I used the command mkdir to create the role directory inside the hoa8_ansible directory.

A screenshot of a terminal window with a dark background. The title bar shows 'madiane@workstation: ~/CPE232_Agpaoa-Ma.Diane/hoa8_a...'. The terminal shows the GNU nano 6.2 editor editing a file named 'inventory'. The content of the file is as follows:

```
[ubuntu]
192.168.56.103

[centos]
192.168.56.120
```

Figure 1.3 Contents of inventory file

Step 2: Creating a playbook for each roles

```
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/hoa8_ansible/roles$ mkdir ubuntu centos
```

```
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/hoa8_ansible/roles$ ls
centos  ubuntu
```

Figure 2.1 Creating the directories inside the roles directory

Inside the roles directory I created two new directories and named it as ubuntu and centos simultaneously by using the command mkdir.

```
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/hoa8_ansible/roles$ cd centos
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/hoa8_ansible/roles/centos$ mkdir tasks
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/hoa8_ansible/roles/centos$ cd tasks
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/hoa8_ansible/roles/centos/tasks$ sudo nano main.yml
[sudo] password for madiane:
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/hoa8_ansible/roles/centos/tasks$ ls
main.yml
```

Figure 2.2 Creating tasks directory and the playbook inside the task directory within the centos directory

First, I changed the directory from roles to centos, then I created the tasks directory by executing the command “mkdir tasks”. After that, I changed the directory from centos to the task directory and created the playbook by executing the command “sudo nano main.yml”

```

GNU nano 6.2 main.yml
- name: Installing required packages for installing Nagios
  tags: dependencies, 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
      - net-snmp
      - net-snmp-utils
      - python2-pip
    state: latest

- name: Installing passlib python package
  pip:
    name: passlib

- name: Creating directory for the downloaded files
  file:
    path: ~/nagios
    state: directory

- name: Installing 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 creating users and group for Nagios
  shell: |
    cd ~/nagios/nagioscore-4.4.6
    ./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: Installing the Nagios plugins and then extracting it
  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 the Nagios plugins
  shell: |
    cd ~/nagios/nagios-plugins
    ./tools/setup
    ./configure

make
make install

- name: Adding a user to a password file
  community.general.htpasswd:
    path: /usr/local/nagios/etc/htpasswd.users
    name: agpaoanagi
    password: mad12

- name: Starting/Restarting Nagios
  service:
    name: nagios
    state: restarted
    enabled: true

- name: Starting/Restarting httpd
  service:
    name: httpd
    state: restarted
    enabled: true

```

Figure 2.3 Content of the playbook main.yml in centos directory

Inside the main.yml playbook, you could see tasks that will install, compile and configure the packages for installing the Nagios and the installation of the Nagios itself in a remote server with an operating system of CentOS. In addition, the playbook will also create or add a user to a password file for the Nagios. After the installation of Nagios and creating or adding a user to a password file there is a task that will start or restart the Nagios and httpd.

```

madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/hoa8_ansible/roles$ cd ubuntu
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/hoa8_ansible/roles/ubuntu$ mkdir tasks
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/hoa8_ansible/roles/ubuntu$ cd tasks
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/hoa8_ansible/roles/ubuntu/tasks$ sudo nano main.yml
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/hoa8_ansible/roles/ubuntu/tasks$ ls
main.yml

```

Figure 2.4 Creating tasks directory and the playbook inside the task directory within the ubuntu directory

First, I changed the directory from roles to ubuntu, then I created the tasks directory by executing the command “mkdir tasks”. After that, I changed the directory from ubuntu to the task directory and created the playbook by executing the command “sudo nano main.yml”

```

GNU nano 6.2 main
- name: Installing required packages for installing Nagios
  tags: dependencies, libraries
  apt:
    name:
      - autoconf
      - gcc
      - libbc6
      - make
      - wget
      - unzip
      - apache2
      - php
      - libapache2-mod-php7.4
      - libgd-dev
      - openssl
      - libssl-dev
      - autoconf
      - gcc
      - libbc6
      - libncrypt-dev
      - make
      - libssl-dev
      - wget
      - bc
      - gawk
      - dc
      - build-essential
      - snmp
      - libnet-snmp-perl
      - gettext
      - python3-pip
      - python3
    state: latest

- name: Installing passlib python package
  pip:
    name: passlib

- name: Creating directory for the downloaded files
  file:
    path: ~/nagios
    state: directory

- name: Installing and extracting Nagios
  unarchive:
    src: https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.6.tar.gz
    dest: ~/nagios
    remote_src: yes
    mode: 0755
    owner: root
    group: root

- name: Compiling, installing, and Creating users and group for Nagios
  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: Installing the Nagios plugins and then extracting it
  unarchive:
    src: https://github.com/nagios-plugins/nagios-plugins/archive/release-2.3.3.tar.gz
    dest: ~/nagios
    remote_src: yes
    mode: 0755
    owner: root
    group: root

- name: Compiling and installing the Nagios
  shell: |
    cd ~/nagios/nagios-plugins*
    ./tools/setup
    ./configure
    make
    make install

- name: Adding a user to a password file
  community.general.htpasswd:
    path: /usr/local/nagios/etc/htpasswd.users
    name: agpaoanagi
    password: mad12

- name: Starting/Restarting Nagios
  service:
    name: nagios
    state: restarted
    enabled: true

- name: Starting/Restarting httpd
  service:
    name: apache2
    state: restarted
    enabled: true
  
```

Figure 2.5 Content of the playbook main.yml in ubuntu directory

Inside the main.yml playbook, you could see tasks that will install, compile and configure the packages for installing the Nagios and the installation of the Nagios itself in a remote server with an operating system of Ubuntu. In addition, the playbook will also create or add a user to a password file for the Nagios. After the installation of

Nagios and creating or adding a user to a password file there is a task that will start or restart the Nagios and httpd.

Step 3: Creating and Executing the main playbook

```
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/hoa8_ansible$ sudo nano nagio_ins.yml
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/hoa8_ansible$ ls
ansible.cfg  inventory  nagio_ins.yml  roles
```

Figure 3.1 Creating the playbook that configures the roles

I created a playbook inside the hoa8_ansible directory and named it as nagio_ins.yml



```
madiane@workstation: ~/CPE232_Agpaoa-Ma.Diane/hoa8_a...
GNU nano 6.2 nagio_ins.yml
---
- hosts: all
  become: true
  pre_tasks:
    - name: update repository index (CentOS)
      tags: always
      dnf:
        update_cache: yes
        changed_when: false
        when: ansible_distribution == "CentOS"
    - name: install updates (Ubuntu)
      tags: always
      apt:
        update_cache: yes
        changed_when: false
        when: ansible_distribution == "Ubuntu"
  - hosts: ubuntu
    become: true
    roles:
      - ubuntu
  - hosts: centos
    become: true
    roles:
      - centos
```

Figure 3.2 Contents of nagios_ins.yml

Inside the playbook, there is a pre-tasks for all the remote servers and the plays for remote servers that have an operating system of CentOS and Ubuntu.

```
nadiane@workstation:~/CPE232_Agpaa-Ma.Diane/budoy$ ansible-playbook --ask-become-pass nagio_ins.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.109]
ok: [192.168.56.115]

TASK [update repository index (CentOS)] *****
skipping: [192.168.56.115]
ok: [192.168.56.109]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.109]
ok: [192.168.56.115]

PLAY [ubuntu] *****

TASK [Gathering Facts] *****
ok: [192.168.56.115]

TASK [ubuntu : Installing nagios dependencies and libraries] *****
ok: [192.168.56.115]

TASK [ubuntu : Install passlib python package] *****
ok: [192.168.56.115]

TASK [ubuntu : Creating a directory (where the downloaded files will be stored)] *****
ok: [192.168.56.115]

TASK [ubuntu : Downloading and extracting Nagios] *****
changed: [192.168.56.115]

TASK [ubuntu : Compiling, installing, and creating users and group for Nagios] *****
changed: [192.168.56.115]

TASK [ubuntu : Downloading and extracting nagios plugins] *****
changed: [192.168.56.115]
```

```

TASK [ubuntu : Compiling and installing] *****
changed: [192.168.56.115]

TASK [ubuntu : Add a user to a password file and ensure permissions are set] *****
ok: [192.168.56.115]

TASK [ubuntu : Making sure that nagios is started and enabled] *****
changed: [192.168.56.115]

TASK [ubuntu : Making sure that httpd is started and enabled] *****
changed: [192.168.56.115]

PLAY [centos] *****

TASK [Gathering Facts] *****
ok: [192.168.56.109]

TASK [centos : Installing nagios dependencies and libraries] *****
changed: [192.168.56.109]

TASK [centos : Install passlib python package] *****
ok: [192.168.56.109]

TASK [centos : Creating a directory (where the downloaded files will be stored)] *****
ok: [192.168.56.109]

TASK [centos : Downloading and extracting Nagios] *****
changed: [192.168.56.109]

TASK [centos : Compiling, installing, and adding users and groups in nagios] *****
changed: [192.168.56.109]

TASK [centos : Downloading and extracting Nagios plugins] *****
changed: [192.168.56.109]

TASK [centos : Compiling and installing plugins] *****
changed: [192.168.56.109]

TASK [centos : Add a user to a password file and ensure permissions are set] *****
ok: [192.168.56.109]

TASK [centos : Making sure that nagios is started and enabled] *****
changed: [192.168.56.109]

TASK [centos : Making sure that httpd is started and enabled] *****
changed: [192.168.56.109]

PLAY RECAP *****
192.168.56.109      : ok=13   changed=7   unreachable=0    failed=0    skipped=1   rescued=0   ignored=0
192.168.56.115      : ok=13   changed=6   unreachable=0    failed=0    skipped=1   rescued=0   ignored=0

```

Figure 3.3 Running the playbook

After completing all of the playbook (nagio_ins.yml, main.yml for CentOS and Ubuntu remote servers), I execute the playbook by entering the command “ansible-playbook –ask-become-pass nagio_ins.yml”. First, it plays the pre-tasks assigned to all of the remote servers. After that, it runs the play for the ubuntu group and centos group consecutively that consists of tasks that would install the packages for installing the Nagios, Nagios, Nagios plugins, creation or addition of a user to a password file and then the starting/restarting of the nagios and httpd.

Step 4: Output of the Installed Nagios in Ubuntu and CentOS servers

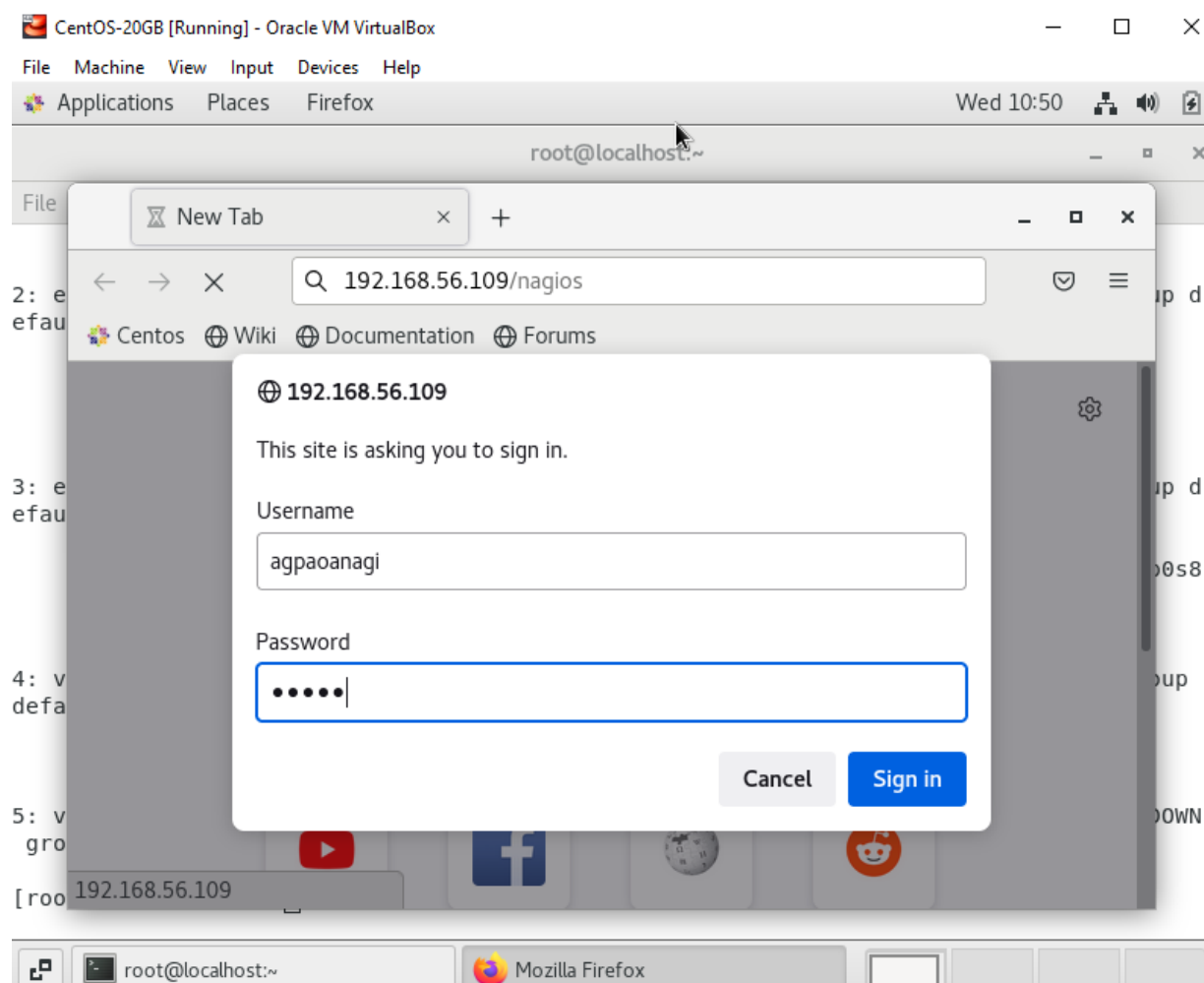


Figure 4.1 Accessing and Signing-in to the Nagios Core in the CentOS

In order to access the Nagios Core, I went to the web browser and typed "192.168.56.109/nagios" the syntax is http://IP address/nagios. I typed the username I added in the password file and entered the password.

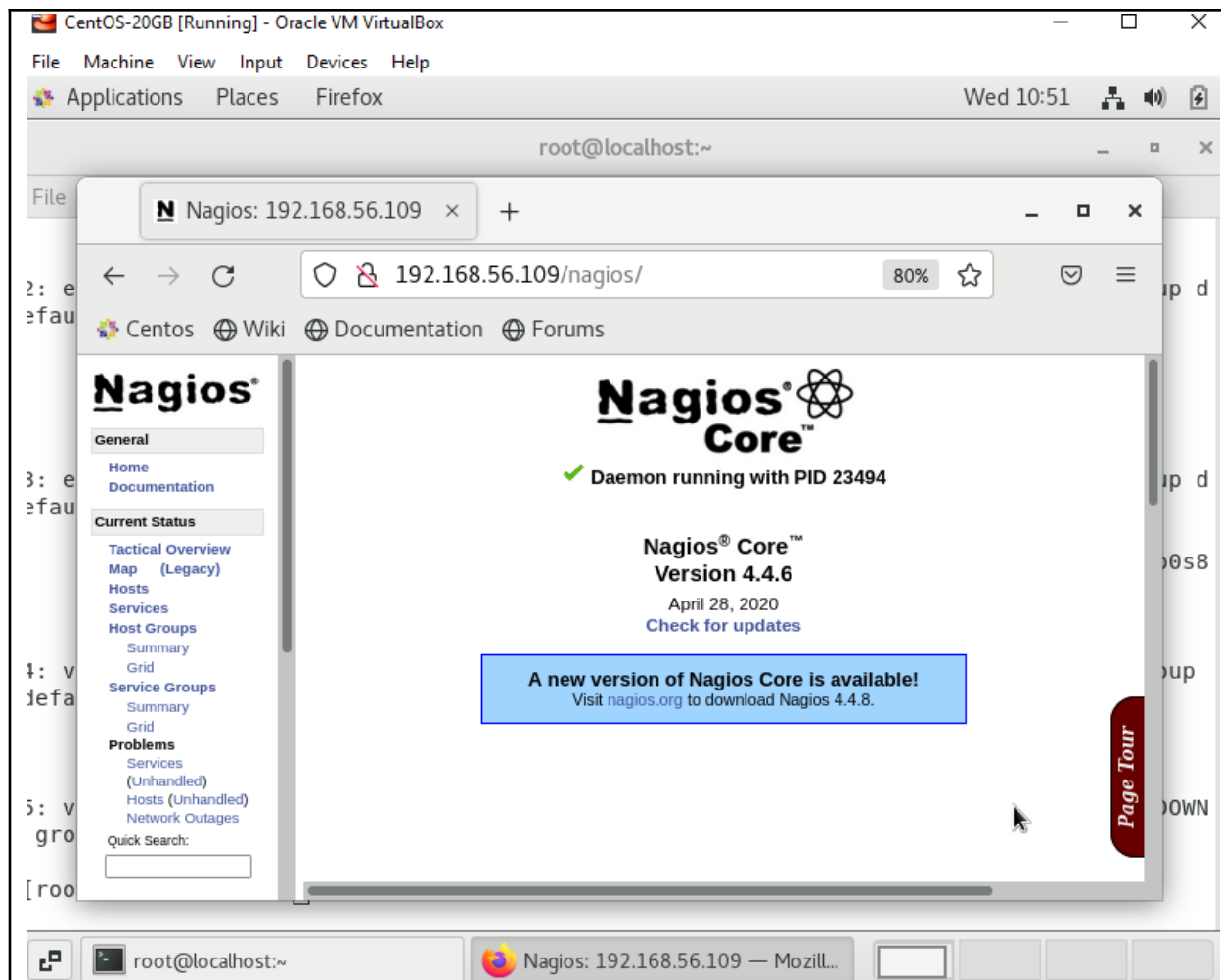


Figure 4.2 Successfully Signing-in Nagios Core

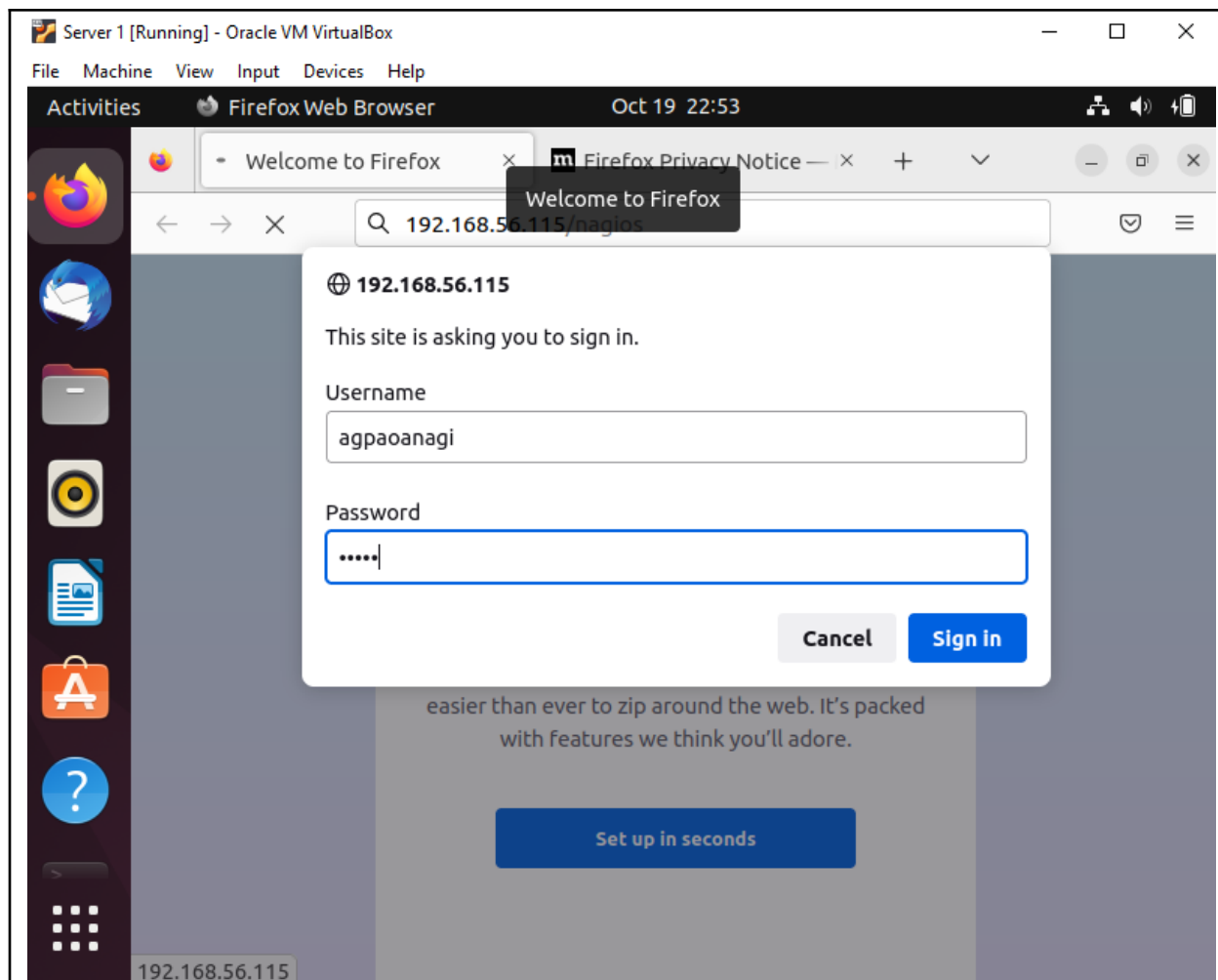


Figure 4.3 Accessing and Signing-in to the Nagios Core in Ubuntu

In order to access the Nagios Core, I went to the web browser and typed "192.168.56.115/nagios" the syntax is http://IP address/nagios. I typed the username I added in the password file and entered the password.

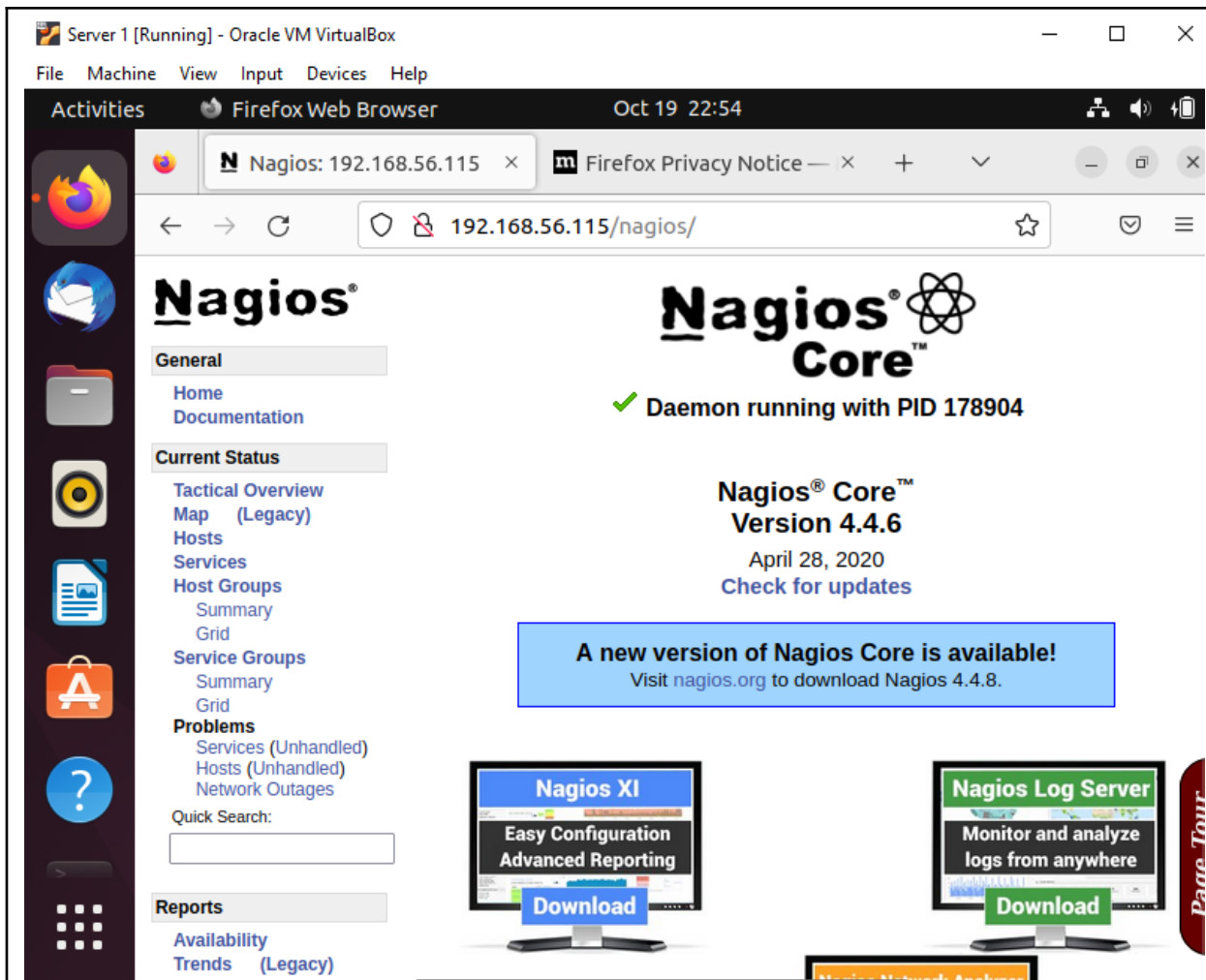


Figure 4.5 Successfully Signing-in Nagios Core

Step 5: Adding the directory to the GitHub

```
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane$ git add hoa8_ansible
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane$ git commit -m "Hands-on Activity 8"
[main 2a41657] Hands-on Activity 8
4 files changed, 201 insertions(+), 15 deletions(-)
rewrite hoa8_ansible/roles/centos/tasks/main.yml (100%)
rewrite hoa8_ansible/roles/ubuntu/tasks/main.yml (99%)
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane$ git push
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (12/12), 1.78 KiB | 454.00 KiB/s, done.
Total 12 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:qmja/CPE232_Agpaoa-Ma.Diane.git
cb07793..2a41657 main -> main
```

Figure 5.1 Saving the hoa8_ansible directory to the GitHub

In order to save the hoa8_ansible directory to the GitHub, I entered the command “git add hoa8_ansible”, then I committed the changes to GitHub and lastly entered the command git push which will execute the committed changes.

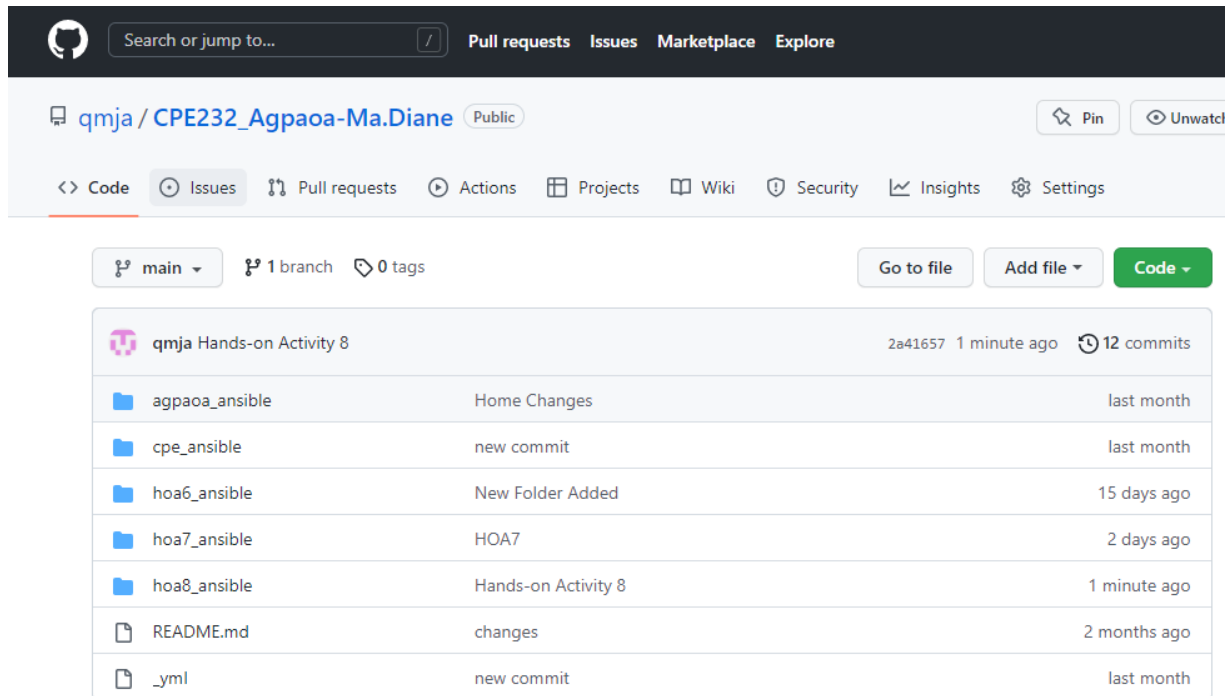


Figure 5.2 GitHub Repository

To verify the changes I checked the GitHub, which will show each of the folders and the last time it was altered, this also shows the “Hands-on Activity 8” phrase that I input in committing the changes which verifies that I successfully added the hoa8_ansible directory in the GitHub.

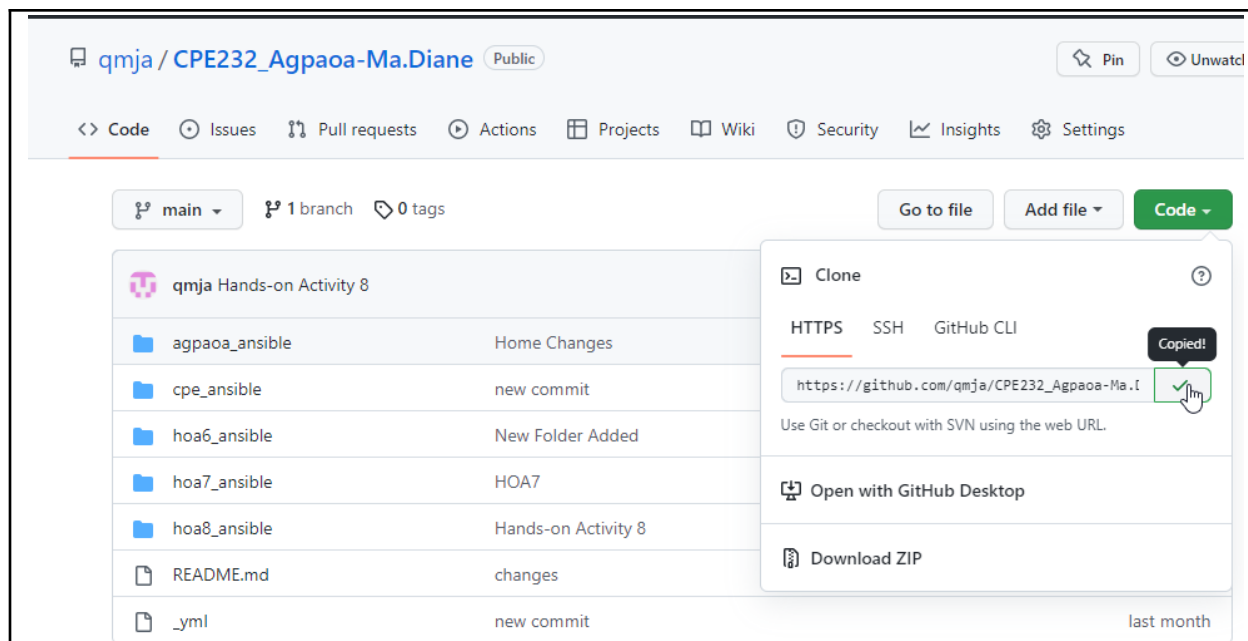


Figure 5.3 Copying the link of the GitHub Repository

GitHub Link: https://github.com/qmja/CPE232_Agpaoa-Ma.Diane.git

Reflections:

Answer the following:

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

Availability monitoring tools are beneficial in monitoring a particular workload and checking if it's reachable in our end. This is very important because downtime in a website could cause loss of revenue and as result a company or an administrator would not look dependable. In addition, the availability monitoring checks the uptime, servers and apps which prevents catastrophic problems because it notifies the administrator of errors and problems before they implicate serious problems that might affect a company.

Conclusions:

In conclusion, this activity helped me to learn about the installation of Nagios using a playbook while implementing the roles. In addition, this activity helped me to make great improvements in creating and consolidating a playbook with the implementation

of roles. The previous activities in which I gained knowledge that improves my skill in creating a playbook greatly helped in accomplishing this activity.

Lastly, I learned about the availability monitoring tools and its benefits in managing servers. Utilizing availability monitoring tools are beneficial because it checks the uptime, servers, and apps and notifies the web administrator of the problems before they cause serious damage in the company's image and revenues.