

<b>Name:</b> Peñas, Issa Victoria H.	<b>Date Performed:</b> 10/11/2022
<b>Course/Section:</b> CPE232 - CPE31S22	<b>Date Submitted:</b> 10/11/2022
<b>Instructor:</b> Dr. Jonathan V. Taylor	<b>Semester and SY:</b> 1st Semester (SY: 2021 - 2022)
<b>Activity 8: Install, Configure, and Manage Availability Monitoring tools</b>	
<b>1. Objectives</b>	
Create and design a workflow that installs, configure and manage enterprise monitoring tools using Ansible as an Infrastructure as Code (IaC) tool.	
<b>2. Discussion</b>	
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.	
<b>3. Tasks</b>	
<ol style="list-style-type: none"> <li>1. Create a playbook that installs Nagios in both Ubuntu and CentOS. Apply the concept of creating roles.</li> <li>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.)</li> <li>3. Show an output of the installed Nagios for both Ubuntu and CentOS.</li> </ol>	
<b>4. Output</b>	
<ol style="list-style-type: none"> <li>1. Create a new repository in GitHub under the name of <b>HOA-8.1- Nagios</b>, and make sure that the repository is <b>Public</b>. As a good practice add a <b>README.md</b> file and input any related information regarding your inserted repository</li> </ol>	

The screenshot shows the GitHub 'Create a new repository' interface. At the top, it says 'Create a new repository' and 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.' Below this, there are two input fields: 'Owner' with a dropdown menu showing 'IVPENAS' and 'Repository name' with the text 'HOA-8.1-Nagios' and a green checkmark. A message below the repository name says 'Great repository names are available. HOA-8.1-Nagios is available. Get inspiration? How about solid-octo-computing-machine?'. There is a 'Description (optional)' text area. Below that, there are two radio buttons: 'Public' (selected) and 'Private'. Under 'Public', it says 'Anyone on the internet can see this repository. You choose who can commit.' Under 'Private', it says 'You choose who can see and commit to this repository.' Below the visibility options, there is a section 'Initialize this repository with:' with the text 'Skip this step if you're importing an existing repository.' There are three checkboxes: 'Add a README file' (checked), 'Add .gitignore' (unchecked), and 'Choose a license' (unchecked). Below these, there are dropdown menus for '.gitignore template' (set to 'None') and 'License' (set to 'None'). At the bottom, there is a note 'You are creating a public repository in your personal account.' and a blue 'Create repository' button.

**Figure 1.1.** Shows the creation of New Repository in Github

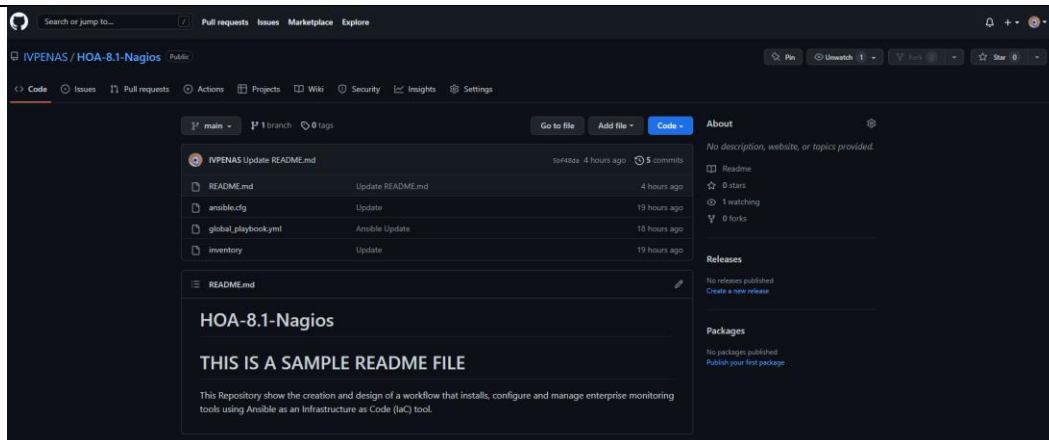


Figure 1.2. Shows the Repository (Note: I've already created it before making this Document)

2. After creating the new repository make sure that the Local Server was connected to it using the command **git clone [ssh link]** whereas the ssh link can be found inside the **code** button

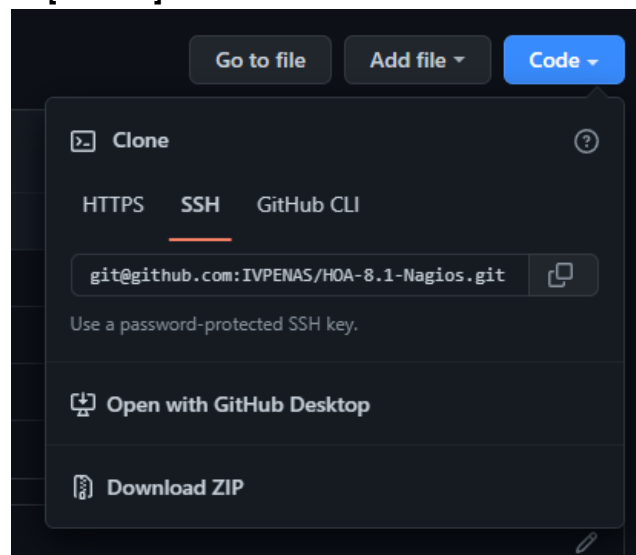


Figure 1.3. Generating a SSH Link to connect it to the servers

```

penas@penas-VirtualBox:~$ git clone git@github.com:IVPENAS/HOA-8.1-Nagios.git
Cloning into 'HOA-8.1-Nagios'...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 27 (delta 2), reused 20 (delta 1), pack-reused 0
Receiving objects: 100% (27/27), 4.68 KiB | 4.68 MiB/s, done.
Resolving deltas: 100% (2/2), done.
penas@penas-VirtualBox:~$ ls
Desktop      HOA-8.1-Nagios      Pictures      Templates
Documents    'HOA-8.1-Nagios(copy)'  Public       Videos
Downloads    Music                snap
penas@penas-VirtualBox:~$

```

Figure 1.4. Cloning the Repository to the Local Machine which serves our Folder

3. Change your directory on the cloned folder from GitHub using **cd CE232\_penas/Nagios** and create the base ansible structure **ansible.cfg** and **inventory** whereas you insert the selected Servers, in my case I chose Server 1 - 192.168.56.112 and CentOS - 192.168.56.111. and prepare a **.yaml** file for the global configuration playbook for the whole ansible.

**Note:** In this case the created directories were made during face-to-face classes which in later parts, you'll see different names of directories yet same file names except the main base .yaml file which will be named as **global\_playbook.yaml**

```
penas@penas-workstation-VirtualBox:~/CPE232_penas/Nagios$ tree
├── ansible.cfg
├── inventory
└── main_playbook.yml

0 directories, 3 files
penas@penas-workstation-VirtualBox:~/CPE232_penas/Nagios$ cat ansible.cfg
[defaults]
inventory = inventory
host_keychecking = False

deprecation_warnings = False

remote_user = penas
private_key_file = ~/.ssh/
penas@penas-workstation-VirtualBox:~/CPE232_penas/Nagios$ cat inventory
server1
CentOS
penas@penas-workstation-VirtualBox:~/CPE232_penas/Nagios$ cat main_playbook.yml
---
#Penas
- hosts: all
  become: true
  pre_tasks:

    - name: Update repository index (CentOS)
      tags: always
      dnf:
        update_cache: yes
        when: ansible_distribution == "CentOS"

    - name: Install Updates (CentOS)
      tags: always
      dnf:
        update_only: yes
        update_cache: yes
        when: ansible_distribution == "CentOS"

    - name: Install Updates (Ubuntu)
      tags: always
      apt:
        update_cache: yes
        changed_when: false
        when: ansible_distribution == "Ubuntu"
penas@penas-workstation-VirtualBox:~/CPE232_penas/Nagios$
```

**Figure 1.5.** Creating a new ansible structure where in this figure shows the basic input of the initial main playbook

```
penas@penas-workstation-VirtualBox:~/CPE232_penas/Nagios$ ansible-playbook --ask-become-pass main_playbook.yml
BECOME password:

PLAY [all] *****
TASK [Gathering Facts] *****
ok: [CentOS]
ok: [server1]

TASK [Update repository index (CentOS)] *****
skipping: [server1]
ok: [CentOS]

TASK [Install Updates (CentOS)] *****
skipping: [server1]
ok: [CentOS]

TASK [Install Updates (Ubuntu)] *****
skipping: [CentOS]
ok: [server1]

PLAY RECAP *****
CentOS      : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
server1     : ok=2    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
penas@penas-workstation-VirtualBox:~/CPE232_penas/Nagios$
```

**Figure 1.6.** Shows the output of the initial main playbook which it consist of success Update in Ubuntu and CentOS

- When initiating roles the admin should create directories using the command **mkdir roles** where we store multiple roles for the main playbook to use. Inside of the **roles** directory, make another directory regarding the types of roles where in this case it'll be named as **nagios4-CentOS** for CentOS Servers and **nagios4-Ubuntu** for Ubuntu Servers

Lastly, inside the respective directory of the role itself create another directory named as **tasks** which consist a yml file named as **main.yml**, to easily achieve this last step simply input the command **touch nagios4-CentOS/tasks/main.yml** and **touch nagios4-Ubuntu/tasks/main.yml**

```
penas@penas-VirtualBox:~/HOA-8.1-Nagios$ mkdir roles
penas@penas-VirtualBox:~/HOA-8.1-Nagios$ cd roles
penas@penas-VirtualBox:~/HOA-8.1-Nagios/roles$ mkdir nagios4-CentOS nagios4-Ubuntu
penas@penas-VirtualBox:~/HOA-8.1-Nagios/roles$ touch nagios4-CentOS/tasks/main.yml
touch: cannot touch 'nagios4-CentOS/tasks/main.yml': No such file or directory
penas@penas-VirtualBox:~/HOA-8.1-Nagios/roles$ cd nagios4-CentOS
penas@penas-VirtualBox:~/HOA-8.1-Nagios/roles/nagios4-CentOS$ mkdir tasks
penas@penas-VirtualBox:~/HOA-8.1-Nagios/roles/nagios4-CentOS$ cd ..
penas@penas-VirtualBox:~/HOA-8.1-Nagios/roles$ cd nagios4-Ubuntu
penas@penas-VirtualBox:~/HOA-8.1-Nagios/roles/nagios4-Ubuntu$ mkdir tasks
penas@penas-VirtualBox:~/HOA-8.1-Nagios/roles/nagios4-Ubuntu$ cd .
penas@penas-VirtualBox:~/HOA-8.1-Nagios/roles/nagios4-Ubuntu$ cd ..
penas@penas-VirtualBox:~/HOA-8.1-Nagios/roles$ touch nagios4-CentOS/tasks/main.yml
penas@penas-VirtualBox:~/HOA-8.1-Nagios/roles$ touch nagios4-Ubuntu/tasks/main.yml
penas@penas-VirtualBox:~/HOA-8.1-Nagios/roles$ tree
.
├── nagios4-CentOS
│   └── tasks
│       └── main.yml
└── nagios4-Ubuntu
    └── tasks
        └── main.yml

4 directories, 2 files
```

Figure 1.5. Creating Directories for Roles specifically **nagios4-CentOS** and **nagios4-Ubuntu**

- After creating the subdirectories in Roles Directories, proceed to input the commands on your main.yml of nagios4-Ubuntu

Command Name	Function
<b>Installing Nagios4 Dependecies and Libraries</b>	Installs the pre-requisites of the application in order to install the Nagios more efficient
<b>Install passlib Python Package</b>	A password hashing library that will be used when setting up password
<b>Creating a directory for the downloaded files</b>	Creating a new directory in ~nagios where all the downloaded files will be inserted
<b>Downloading and Extracting Nagios4 from Github</b>	Downloads and Extract the Nagios from Github using an external link
<b>Creating Users and group, Compiling and Installation of Nagios4</b>	Creates User and Groups within the Nagios; Compiles all the files from GitHub Link and Installs it

<b>Downloading and Extracting plugins of Nagios4 from Github</b>	Downloading and Extracting Plugins of Nagios4 from GitHub with the destination to ~nagios
<b>Compiling and Installing Nagios4 plugins</b>	Compiles and Installs the downloaded Nagios4 plugins from Github
<b>Setting User and Password</b>	Setting the desired User and Password as: <i>User: ivpenas</i> <i>Password: 2010167</i>
<b>Confirmation of Nagios4 is enabled</b>	Commands that oversees the Nagios4 application is enabled and connected to the server
<b>Confirmation of httpd is enabled</b>	Commands that oversees the httpd is enabled and connected to the server

**Table 1.1.** Shows brief information on what a set of commands do in the playbook

### Ubuntu:

```
#Penas-Ubuntu
- name: Installing Nagios4 Dependencies and Libraries
  tags: dependencies, libraries
  apt:
    name:
      - autoconf
      - gcc
      - libc6
      - make
      - wget
      - unzip
      - apache2
      - php
      - libapache2-mod-php7.4
      - libgd-dev
      - openssl
      - libssl-dev
      - autoconf
      - gcc
      - libc6
      - libmcrpt-dev
      - make
      - libssl-dev
      - wget
      - bc
      - gawk
      - dc
      - build-essential
      - snmp
      - libnet-snmp-perl
      - gettext
      - python3-pip
      - python3
    state: latest
- name: Install passlib Python Package
  pip:
    name: passlib
- name: Creating a directory for the downloaded files
  file:
    path: ~/nagios
    state: directory
- name: Setting User and Password
  community.general.htpasswd:
    path: /usr/local/nagios/etc/htpasswd.users
    name: ivpenas
    password: 2010167
- name: Confirmation of Nagios4 is enabled
  service:
    name: nagios
    state: restarted
    enabled: true
- name: Confirmation of httpd is enabled
  service:
    name: apache2
    state: restarted
    enabled: true
```

```

- name: Downloading and Extracting Nagios4 from Github
  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: Creating Users and group, Compiling and Installation of Nagios4
  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: Downloading and Extracting plugins of Nagios4 from Github
  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 Nagios4 plugins
  shell: |
    cd ~/nagios/nagios-plugins*
    ./tools/setup
    ./configure
    make
    make install

```

**Figure 1.6-8.** The Playbook of **nagios4-Ubuntu** where it installs the Nagios Application and connects to the server

6. Then create the same set of commands in the playbook of nagios4-CentOS named **main.yml**, there will be minor changes in the playbook of CentOS as some of the commands were different than Ubuntu yet it still serves the same functions.

Command Name	Function
<b>Installing Nagios4 Dependencies and Libraries</b>	Installs the pre-requisites of the application in order to install the Nagios more efficient
<b>Install passlib Python Package</b>	A password hashing library that will be used when setting up password
<b>Creating a directory for the downloaded files</b>	Creating a new directory in ~nagios where all the downloaded files will be inserted
<b>Downloading and Extracting Nagios4 from Github</b>	Downloads and Extract the Nagios from Github using an external link
<b>Creating Users and group, Compiling and Installation of Nagios4</b>	Creates User and Groups within the Nagios; Compiles all the files from GitHub Link and Installs it

<b>Downloading and Extracting plugins of Nagios4 from Github</b>	Downloading and Extracting Plugins of Nagios4 from GitHub with the destination to ~nagios
<b>Compiling and Installing Nagios4 plugins</b>	Compiles and Installs the downloaded Nagios4 plugins from Github
<b>Setting User and Password</b>	Setting the desired User and Password as: <i>User: ivpenas</i> <i>Password: 2010167</i>
<b>Confirmation of Nagios4 is enabled</b>	Commands that oversees the Nagios4 application is enabled and connected to the server
<b>Confirmation of httpd is enabled</b>	Commands that oversees the httpd is enabled and connected to the server

**Table 1.1.** Shows brief information on what a set of commands do in the playbook

### CentOS:

```
#Penas-CentOS
- name: Installing Nagios4 Dependencies and Libraries
  tags: 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: Install passlib Python Package
  pip:
    name: passlib

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

- name: Downloading and Extracting Nagios4 from Github
  unarchive:
    src: https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.6.tar.gz
    dest: ~/nagios
    remote_src: yes
    mode: 0777
```

```

- name: Adding Users and Groups, Compiling, and Installing in Nagios4 from Github
  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 Nagios4 plugins from Github
  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 Nagios4 plugins
  shell: |
    cd ~/nagios/nagios-plugins*
    ./tools/setup
    ./configure
    make
    make install

- name: Setting User and Password
  community.general.htpasswd:
    path: /usr/local/nagios/etc/htpasswd.users
    name: ivpenas
    password: 2010167

- name: Confirmation of Nagios4 is enabled
  service:
    name: nagios
    state: restarted
    enabled: true

- name: Confirmation of httpd is enabled
  service:
    name: httpd
    state: restarted
    enabled: true

```

**Figure 1.9-10.** The Playbook of **nagios4-CentOS** where it installs the Nagios Application and connects to the server

7. Dont you forget your initial `global_playbook.yml`, since we've initiated roles between **nagios4-CentOS** and **nagios4-Ubuntu**, the admin should edit and append some commands from the `global_playbook.yml`. Make sure you are in the same directory of the playbook.



```

---
- hosts: all
  become: true
  pre_tasks:

    - name: Update and upgrade remote in Ubuntu servers
      apt:
        update_cache: yes
        upgrade: 'yes'
        when: ansible_distribution == "Ubuntu"

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

    - name: Update and upgrade remote CentOS server
      dnf:
        update_cache: yes
        name: "*"
        state: latest
        when: ansible_distribution == "CentOS"

    - name: Dpkg fixing in Ubuntu Servers
      shell: |
        dpkg --configure -a
        when: ansible_distribution == "Ubuntu"

- hosts: nagios4-Ubuntu
  become: true
  roles:
    - nagios4-Ubuntu

- hosts: nagios4-CentOS
  become: true
  roles:
    - nagios4-CentOS

```

**Figure 1.11.** The Playbook of **global\_playbook.yml** where initiates the execution of both roles and other separate commands such as Updating and Upgrading CentOS and Ubuntu Servers; Installation of DNF and Epel Release; and Configuring Dpkg in Ubuntu Servers as a partial requirement of Nagios

8. When the playbooks are correctly configured, run the command **ansible-playbook --ask-become-pass global\_playbook.yml** which executes the main playbook of this folder.

**Outputs:**

```

TASK [Gathering Facts] *****
ok: [server_1]
ok: [Cent-OS]

TASK [Update and upgrade remote in Ubuntu servers] *****
skipping: [Cent-OS]
ok: [server_1]

TASK [Installing dnf and epel-release] *****
skipping: [server_1]
ok: [Cent-OS]

TASK [Update and upgrade remote CentOS server] *****
skipping: [server_1]
ok: [Cent-OS]

TASK [Dpkg fixing in Ubuntu Servers] *****
skipping: [Cent-OS]
changed: [server_1]

PLAY [nagios4-CentOS] *****

TASK [Gathering Facts] *****
ok: [Cent-OS]

TASK [nagios4-CentOS : Installing Nagios4 Dependencies and Libraries] *****
ok: [Cent-OS]

TASK [nagios4-CentOS : Install passlib Python Package] *****
ok: [Cent-OS]

TASK [nagios4-CentOS : Creating a directory for the downloaded files] *****
ok: [Cent-OS]

TASK [nagios4-CentOS : Downloading and Extracting Nagios4 from Github] *****
ok: [Cent-OS]

TASK [nagios4-CentOS : Adding Users and Groups, Compiling, and Installing in Nagios4 from Github] ***
changed: [Cent-OS]

TASK [nagios4-CentOS : Downloading and Extracting Nagios4 plugins from Github] *****
ok: [Cent-OS]

TASK [nagios4-CentOS : Compiling and Installing Nagios4 plugins] *****
changed: [Cent-OS]

TASK [nagios4-CentOS : Setting User and Password] *****
ok: [Cent-OS]

TASK [nagios4-CentOS : Confirmation of Nagios4 is enabled] *****
changed: [Cent-OS]

TASK [nagios4-CentOS : Confirmation of httpd is enabled] *****
changed: [Cent-OS]

```

```

PLAY [nagios4-Ubuntu] *****

TASK [Gathering Facts] *****
ok: [server_1]

TASK [nagios4-Ubuntu : Installing Nagios4 Dependencies and Libraries] *****
ok: [server_1]

TASK [nagios4-Ubuntu : Install passlib Python Package] *****
ok: [server_1]

TASK [nagios4-Ubuntu : Creating a directory for the downloaded files] *****
ok: [server_1]

TASK [nagios4-Ubuntu : Downloading and Extracting Nagios4 from Github] *****
ok: [server_1]

TASK [nagios4-Ubuntu : Creating Users and group, Compiling and Installation of Nagios4] *
**
changed: [server_1]

TASK [nagios4-Ubuntu : Downloading and Extracting plugins of Nagios4 from Github] *****
ok: [server_1]

TASK [nagios4-Ubuntu : Compiling and Installing Nagios4 plugins] *****
changed: [server_1]

TASK [nagios4-Ubuntu : Setting User and Password] *****
ok: [server_1]

TASK [nagios4-Ubuntu : Confirmation of Nagios4 is enabled] *****
changed: [server_1]

TASK [nagios4-Ubuntu : Confirmation of httpd is enabled] *****
changed: [server_1]

PLAY RECAP *****
Cent-OS                : ok=14   changed=4   unreachable=0   failed=0   skipped=2
                      rescued=0   ignored=0
server_1              : ok=14   changed=5   unreachable=0   failed=0   skipped=2
                      rescued=0   ignored=0

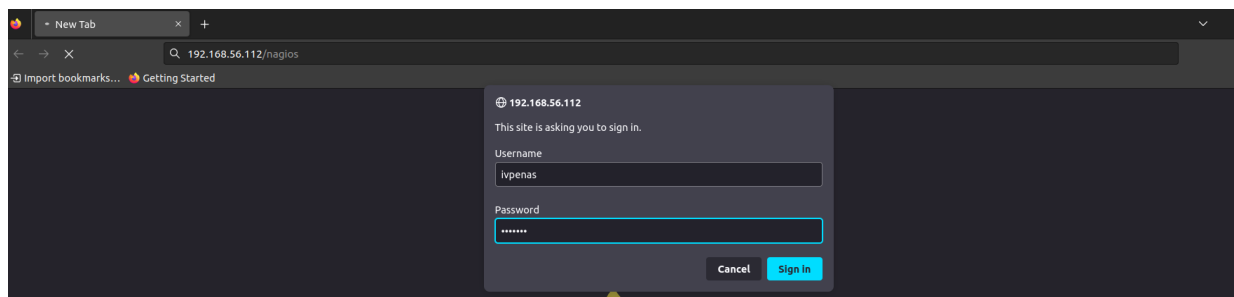
```

**Figure 1.12-14.** Shows the Output of the PLAY RUN after executing the main playbook which is **global\_playbook.yml**

9. To Verify if the Nagios was fully installed within the Servers, Open a Brower and insert the respective IP Address of the server along with **/nagios**, In Server 1 - 192.168.56.112/nagios while in CentOS - 192.168.56.111/nagios

**Verification:**

**Ubuntu:**



**Figure 1.15.** Shows a user verification where the admin will input the correct Username and Password



## Unauthorized

This server could not verify that you are authorized to access the document requested. Either you supplied the wrong credentials (e.g., bad password), or your browser doesn't understand how to supply the credentials required.

Apache/2.4.52 (Ubuntu) Server at 192.168.112 Port 80

Figure 1.16. If the Credential was not correct it'll show the Unauthorized page

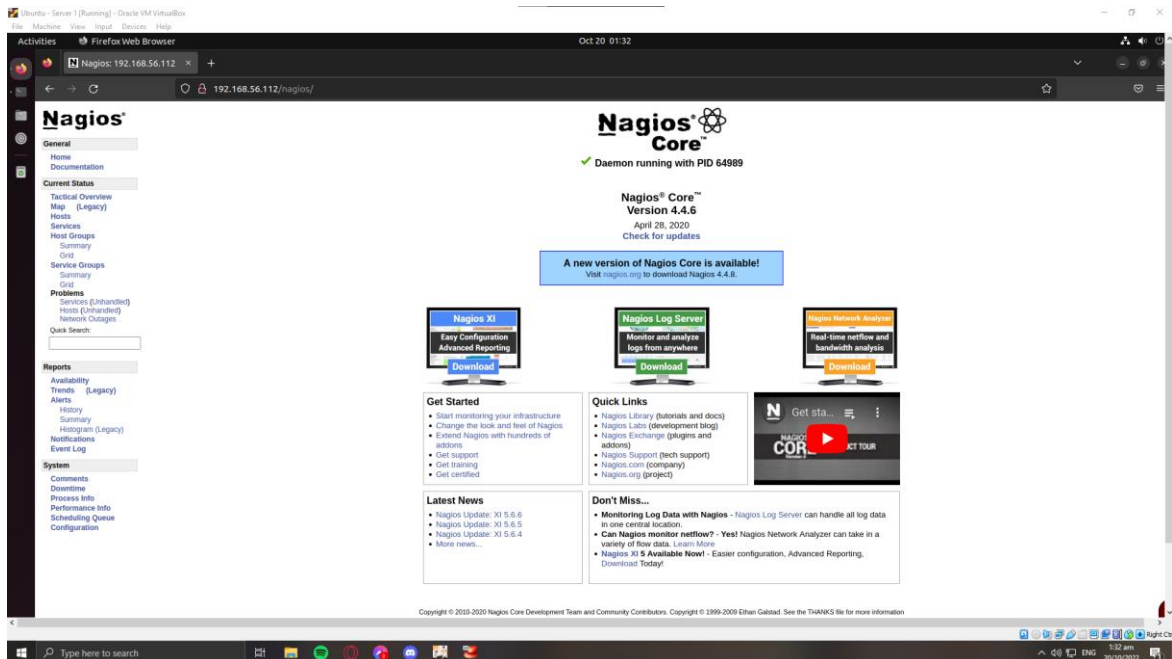


Figure 1.17. If the Credential was correct it'll show the Nagios page

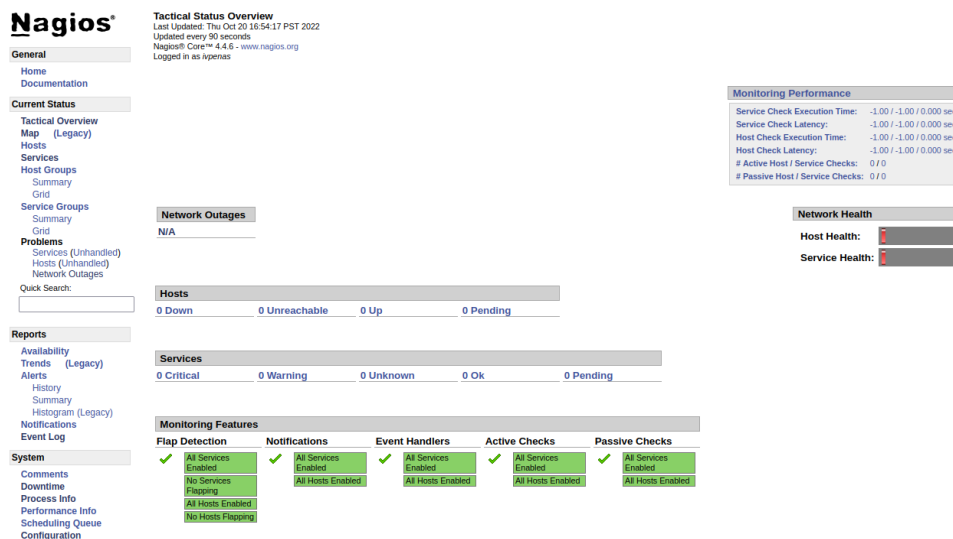


Figure 1.18. Shows the Tactical Overview of the Server

## CentOS:

192.168.56.111/nagios/

Documentation Forums

192.168.56.111

This site is asking you to sign in.

Username

admin

Password

\*\*\*\*\*

Cancel Sign in

Figure 1.19. Shows a user verification where the admin will input the correct Username and Password

CentOS [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Applications Places Firefox

Thu 01:34

N Nagios: 192.168.56.111 x Firefox Privacy Notice - | x +

192.168.56.111/nagios/

Centos Wiki Documentation Forums

# Nagios®

General

- Home
- Documentation

Current Status

- Tactical Overview
- Map (Legacy)
- Hosts
- Services
- Host Groups
- Summary
- Grid
- Service Groups
- Summary
- Grid

Problems

- Services (Unhandled)
- Hosts (Unhandled)
- Network Outages

Quick Search:

Reports

- Availability
- Trends (Legacy)
- Alerts
- History
- Summary
- Histogram (Legacy)
- Notifications
- Event Log

System

- Comments
- Downtime
- Process Info
- Performance Info
- Scheduling Queue
- Configuration

## Nagios® Core™

✓ Daemon running with PID 10292

Nagios® Core™  
Version 4.4.6  
April 28, 2020  
Check for updates

A new version of Nagios Core is available!  
Visit [nagios.org](https://nagios.org) to download Nagios 4.4.8.

**Nagios XI**  
Easy Configuration  
Advanced Reporting  
Download

**Nagios Log Server**  
Monitor and analyze  
logs from anywhere  
Download

**Nagios Network Analyzer**  
Real-time netflow and  
bandwidth analysis  
Download

**Get Started**

- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of add-ons
- Get support
- Get training
- Get certified

**Quick Links**

- Nagios Library (tutorials and docs)
- Nagios Labs (development blog)
- Nagios Exchange (plugins and addons)
- Nagios Support (tech support)
- Nagios.com (company)
- Nagios.org (project)

**Latest News**

- Nagios Update: XI 5.6.6
- Nagios Update: XI 5.6.5
- Nagios Update: XI 5.6.4
- More news...

**Don't Miss...**

- Monitoring Log Data with Nagios** - Nagios Log Server can handle all log data in one central location.
- Can Nagios monitor netflow?** - Yes! Nagios Network Analyzer can take in a variety of flow data. [Learn More](#)
- Nagios XI 5 Available Now!** - Easier configuration, Advanced Reporting. [Download Today!](#)

type here to search

1:34 am  
20/10/2022

Figure 1.20. If the Credential was correct it'll show the Nagios page

## General

[Home](#)  
[Documentation](#)

## Current Status

[Tactical Overview](#)  
[Map \(Legacy\)](#)

[Hosts](#)

[Services](#)

[Host Groups](#)

[Summary](#)

[Grid](#)

[Service Groups](#)

[Summary](#)

[Grid](#)

## Problems

[Services \(Unhandled\)](#)

[Hosts \(Unhandled\)](#)

[Network Outages](#)

Quick Search:

## Reports

[Availability](#)

[Trends \(Legacy\)](#)

[Alerts](#)

[History](#)

[Summary](#)

[Histogram \(Legacy\)](#)

[Notifications](#)

[Event Log](#)

## System

[Comments](#)

[Downtime](#)

[Process Info](#)

[Performance Info](#)

[Scheduling Queue](#)

[Configuration](#)

[ [Host Downtime](#) | [Service Downtime](#) ]

## Scheduled Host Downtime

[Schedule host downtime](#)

Host Name	Entry Time	Author	Comment	Start Time	End Time	Type	Duration	Downtime ID	Trigger ID	Actions
-----------	------------	--------	---------	------------	----------	------	----------	-------------	------------	---------

There are no hosts with scheduled downtime

## Scheduled Service Downtime

[Schedule service downtime](#)

Host Name	Service	Entry Time	Author	Comment	Start Time	End Time	Type	Duration	Downtime ID	Trigger ID	Actions
-----------	---------	------------	--------	---------	------------	----------	------	----------	-------------	------------	---------

There are no services with scheduled downtime

Figure 1.21. Shows the Downtime in CentOS Server

- When the Verification was done, fully sync the cloned folder to Github using the command '**git add \***' which adds a changed directory, '**git commit -m "ANYMESSG"**' Commits the Changes are made within the repository folder, and '**git push**' to sync and save the folder to the Git Repository.

```
penas@penas-VirtualBox:~/HOA-8.1-Nagios$ git add *
penas@penas-VirtualBox:~/HOA-8.1-Nagios$ git commit -m "Last Commit"
[main 43e909f] Last Commit
 4 files changed, 229 insertions(+), 16 deletions(-)
 create mode 100644 roles/nagios4-CentOS/tasks/main.yml
 create mode 100644 roles/nagios4-Ubuntu/tasks/main.yml
penas@penas-VirtualBox:~/HOA-8.1-Nagios$ git push
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (11/11), 2.10 KiB | 2.10 MiB/s, done.
Total 11 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
```

Figure 1.20. Insert the commands to update the Repository

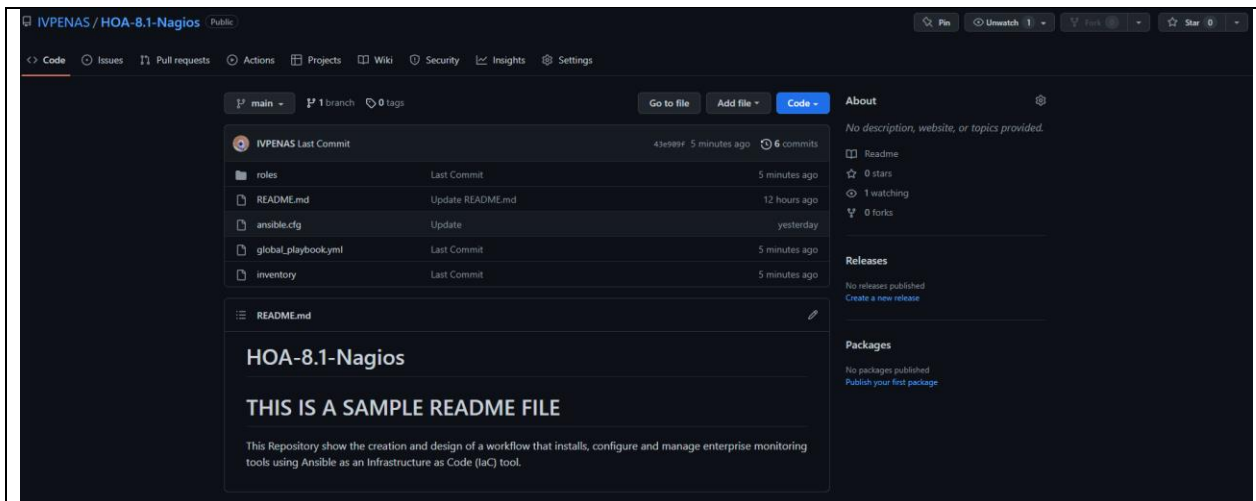


Figure 1.21. The Repository is up to date

GitHub Repository Link: <https://github.com/IVPENAS/HOA-8.1-Nagios.git>

### Reflections:

Answer the following:

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

Availability monitoring tools can be easily identified as our Task Manager in our Windows OS computers, which illustrates the *servers* and *applications opened* and their *activities* using **tables and graphs to identify their statuses**. These kinds of monitory tools are mostly used in businesses specifically companies that handle multiple servers and databases which the System Administrators handle in order to *prevent situations like DDoS or Bugs within the Servers* by notifying them beforehand before it can make catastrophic damage to the business where it can affect the structure of the Business and also the revenue.

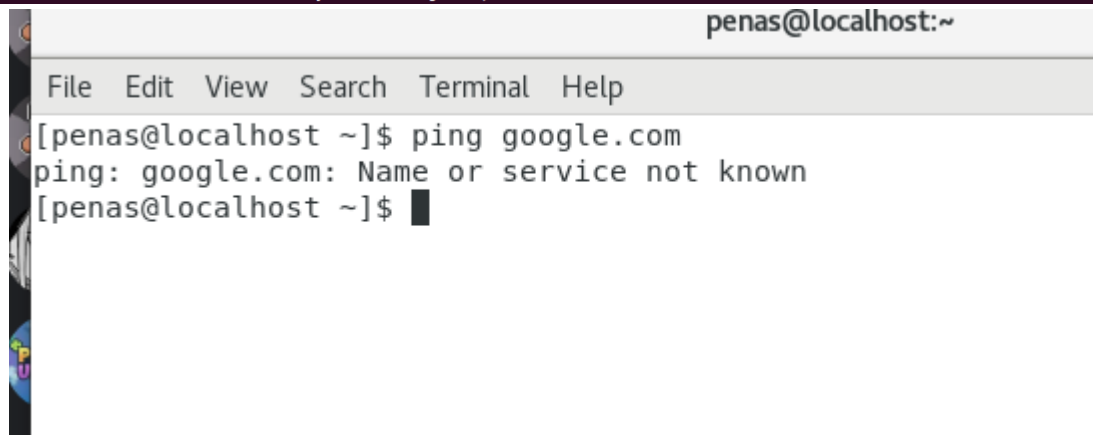
The Nagios Application is one example of an Availability Monitoring Tool, where its services include a [1] **Comprehensive Monitoring** of the server, applications, network protocols, and Operating systems. [2] **Problem Remediation** which allows the System Administrator to be notified whenever a problem or issues occur within the system and automatically restarts any failed applications or services. [3] **UPS Backup system**, and more, which can benefit the networks.

### Conclusions:

The Nagios Application is one of the monitoring tool open-source applications that a System Administrator uses, whereas it serves as the 'Task Manager' that monitors multiple types of devices such as simple Computer Systems, Servers, Network Protocols, or Operating Systems which can benefit companies that handles multiple Network and Servers as it has features that secures the data, structure and security

infrastructures. Both CentOS and Ubuntu servers were able to install and connect to the Nagios Application by verifying it on the browser, inputting the IP Address of the respective server along with the path **'/nagios'**, and inputting the correct credentials. The installation took some time as the student encountered multiple errors, of both Ubuntu and CentOS being unable to update and using their old update file which resorted to multiple reinstallations within the system and repeating the ssh connectivity. In the end, the student was able to attain multiple objectives in this Hand-On-Activity whereas creating a custom ansible playbook by installing availability monitoring tools specifically the Nagios application, and initiating roles of both Ubuntu and CentOS Servers that provides efficiency for the System Administrator to download, extract, install, update, and upgrade multiple Servers in one go.

```
penas@server-1-virtualbox:~$ sudo apt update
Ign:1 http://security.ubuntu.com/ubuntu hirsute-security InRelease
Ign:2 http://us.archive.ubuntu.com/ubuntu hirsute InRelease
Ign:3 http://us.archive.ubuntu.com/ubuntu hirsute-updates InRelease
Ign:1 http://security.ubuntu.com/ubuntu hirsute-security InRelease
Ign:2 http://us.archive.ubuntu.com/ubuntu hirsute InRelease
Ign:3 http://us.archive.ubuntu.com/ubuntu hirsute-updates InRelease
Ign:1 http://security.ubuntu.com/ubuntu hirsute-security InRelease
Ign:2 http://us.archive.ubuntu.com/ubuntu hirsute InRelease
Ign:3 http://us.archive.ubuntu.com/ubuntu hirsute-updates InRelease
Err:1 http://security.ubuntu.com/ubuntu hirsute-security InRelease
      Temporary failure resolving 'security.ubuntu.com'
Err:2 http://us.archive.ubuntu.com/ubuntu hirsute InRelease
      Temporary failure resolving 'us.archive.ubuntu.com'
Err:3 http://us.archive.ubuntu.com/ubuntu hirsute-updates InRelease
      Temporary failure resolving 'us.archive.ubuntu.com'
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
W: Failed to fetch http://us.archive.ubuntu.com/ubuntu/dists/hirsute/InRelease Temporary failure resolving 'us.archive.ubuntu.com'
W: Failed to fetch http://security.ubuntu.com/ubuntu/dists/hirsute-security/InRelease Temporary failure resolving 'security.ubuntu.com'
W: Failed to fetch http://us.archive.ubuntu.com/ubuntu/dists/hirsute-updates/InRelease Temporary failure resolving 'us.archive.ubuntu.com'
W: Some index files failed to download. They have been ignored, or old ones used instead.
```



**Figure 2.1.** Shows one of the error that the student encountered during the activity