Name: Buenvenida, Ken Benedict D.	**Date Performed: 10-16-2023**
Course/Section: CpE31S4	**Date Submitted: 10-16-2023**
Instructor: Engr. Jonathan Taylar	**Semester and SY: 1st Semester 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.

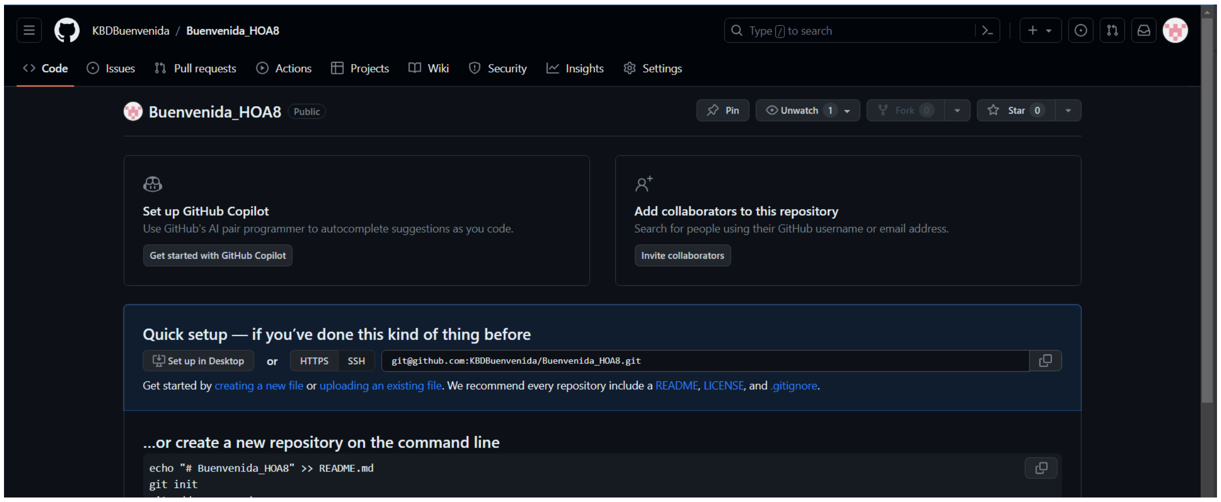
3. Tasks

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 playbook that installs Nagios in both Ubuntu and CentOS. Apply the concept of creating roles.

Step 1: Create a Repository.



KBDBuenvenida / Buenvenida_HOA8

Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights · Settings

Buenvenida_HOA8 Public

Pin · Unwatch 1 · Fork 0 · Star 0

Set up GitHub Copilot
Use GitHub's AI pair programmer to autocomplete suggestions as you code.
Get started with GitHub Copilot

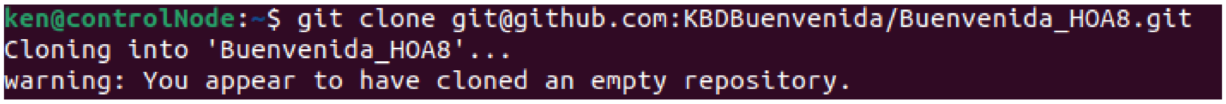
Add collaborators to this repository
Search for people using their GitHub username or email address.
Invite collaborators

Quick setup — if you've done this kind of thing before
Set up in Desktop or HTTPS SSH `git@github.com:KBDBuenvenida/Buenvenida_HOA8.git`
Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

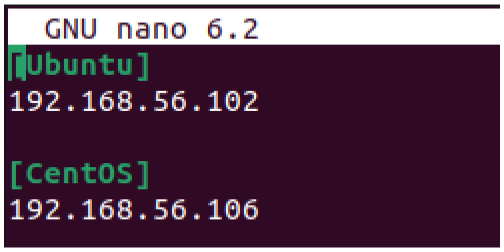
```
echo "# Buenvenida_HOA8" >> README.md
git init
```

Step 2: Git Clone the repository you just created.



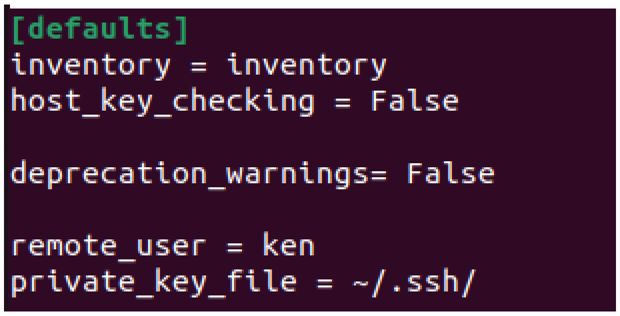
```
ken@controlNode:~$ git clone git@github.com:KBDBuenvenida/Buenvenida_HOA8.git
Cloning into 'Buenvenida_HOA8'...
warning: You appear to have cloned an empty repository.
```

Step 3: Create the inventory file.



```
  GNU nano 6.2
[Ubuntu]
192.168.56.102

[CentOS]
192.168.56.106
```

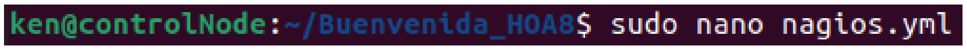
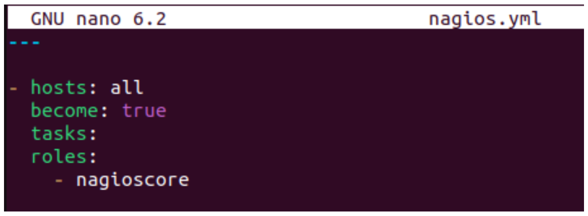
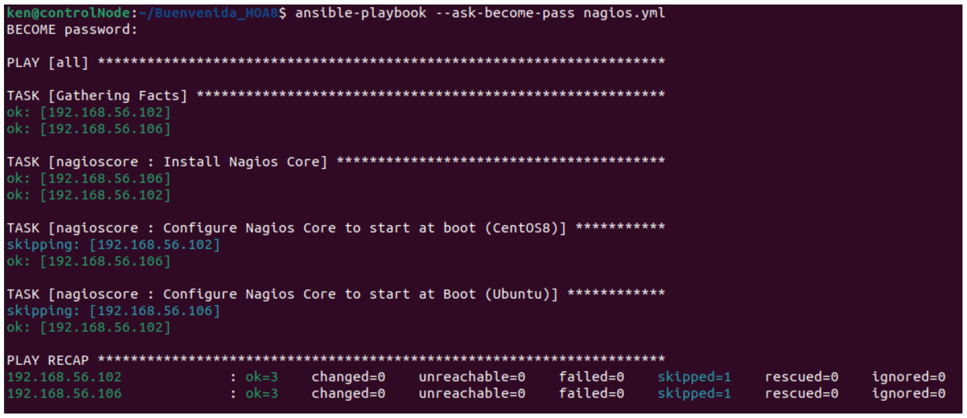
Step 4: Create an ansible.cfg



```
[defaults]
inventory = inventory
host_key_checking = False

deprecation_warnings= False

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

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.)

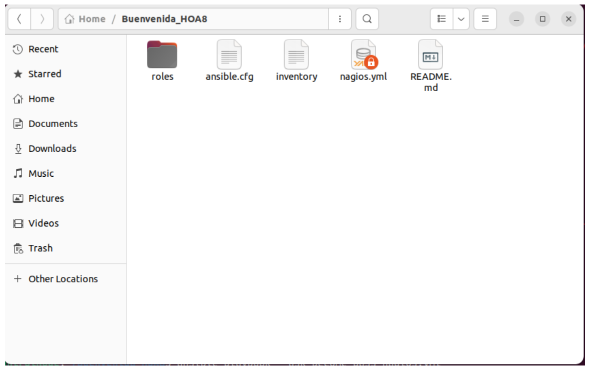
Step 5: Create a playbook named nagios.yml

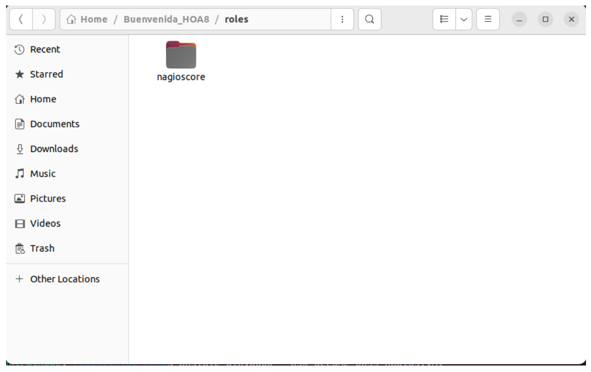
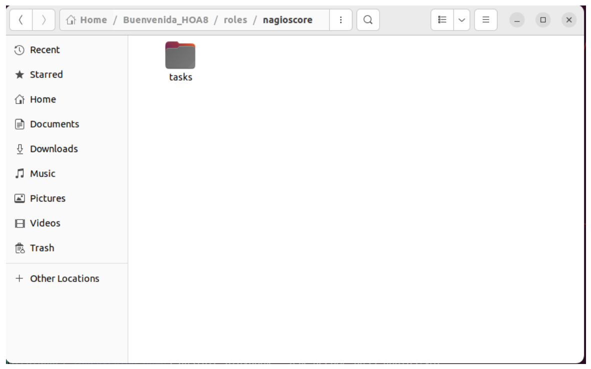
INPUT	 `ken@controlNode:~/Buenvenida_HOA8$ sudo nano nagios.yml`
PROCESS	
OUTPUT	

```
  GNU nano 6.2                           nagios.yml
---

- hosts: all
  become: true
  tasks:
  roles:
    - nagioscore
```

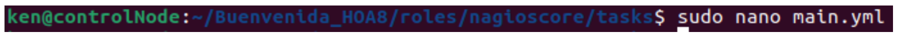
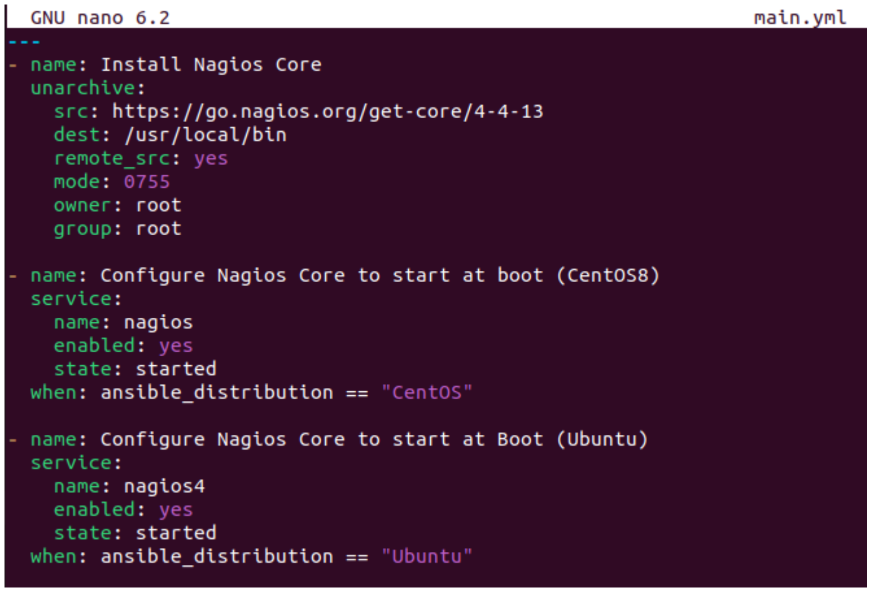
```
ken@controlNode:~/Buenvenida_HOA8$ ansible-playbook --ask-become-pass nagios.yml
BECOME password:

PLAY [all] *********************************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.102]
ok: [192.168.56.106]

TASK [nagioscore : Install Nagios Core] ****************************************
ok: [192.168.56.106]
ok: [192.168.56.102]

TASK [nagioscore : Configure Nagios Core to start at boot (CentOS8)] ***********
skipping: [192.168.56.102]
ok: [192.168.56.106]

TASK [nagioscore : Configure Nagios Core to start at Boot (Ubuntu)] ************
skipping: [192.168.56.106]
ok: [192.168.56.102]

PLAY RECAP *********************************************************************
192.168.56.102             : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.106             : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Step 6: Create a directory named “roles”, “nagioscore”, and “tasks” inside the repository.

ROLES	

NAGIOSCORE	 Home / Buenvenida_HOA8 / roles Recent Starred Home Documents Downloads Music Pictures Videos Trash Other Locations nagioscore
TASKS	 Home / Buenvenida_HOA8 / roles / nagioscore Recent Starred Home Documents Downloads Music Pictures Videos Trash Other Locations tasks

Step 7: Create the “main.yml” inside the “tasks” directory.

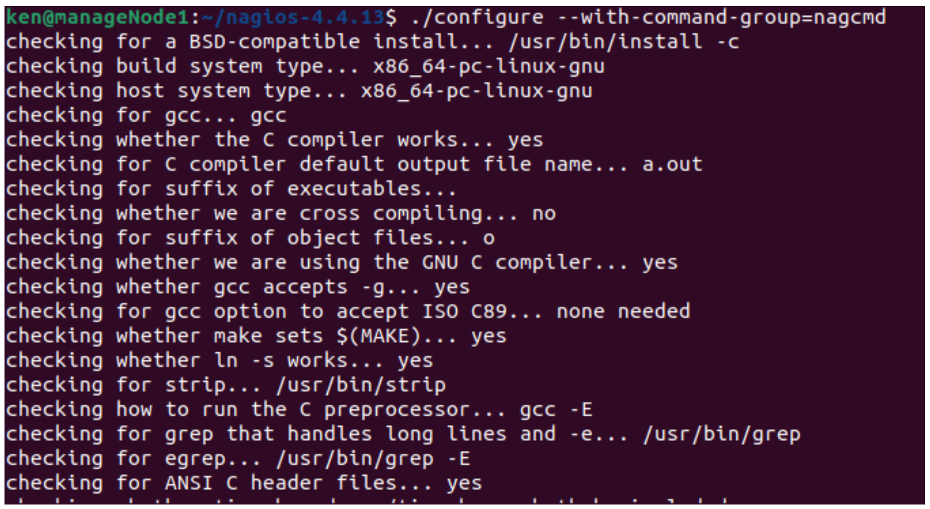
INPUT	
OUTPUT	

```
ken@controlNode:~/Buenvenida_HOA8/roles/nagioscore/tasks$ sudo nano main.yml
```

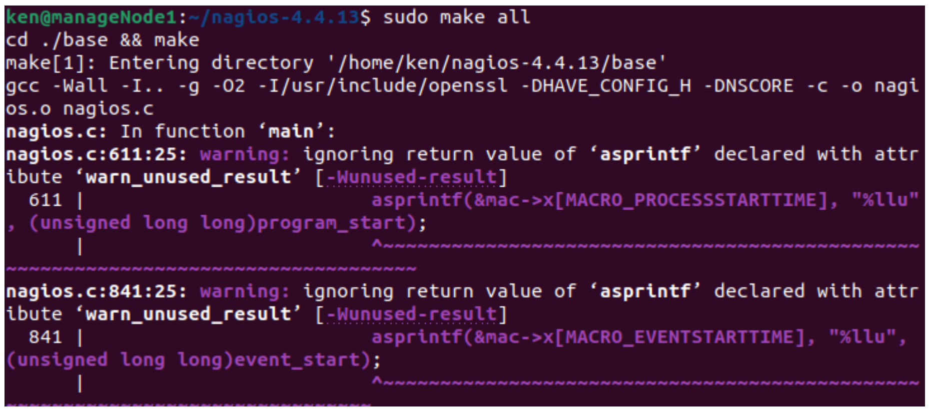
```
  GNU nano 6.2                                                       main.yml
---
- name: Install Nagios Core
  unarchive:
    src: https://go.nagios.org/get-core/4-4-13
    dest: /usr/local/bin
    remote_src: yes
    mode: 0755
    owner: root
    group: root

- name: Configure Nagios Core to start at boot (CentOS8)
  service:
    name: nagios
    enabled: yes
    state: started
  when: ansible_distribution == "CentOS"

- name: Configure Nagios Core to start at Boot (Ubuntu)
  service:
    name: nagios4
    enabled: yes
    state: started
  when: ansible_distribution == "Ubuntu"
```

Step 8: Start Nagios and Apache service for Ubuntu and CentOS. Enter these same syntax to both Ubuntu and CentOS

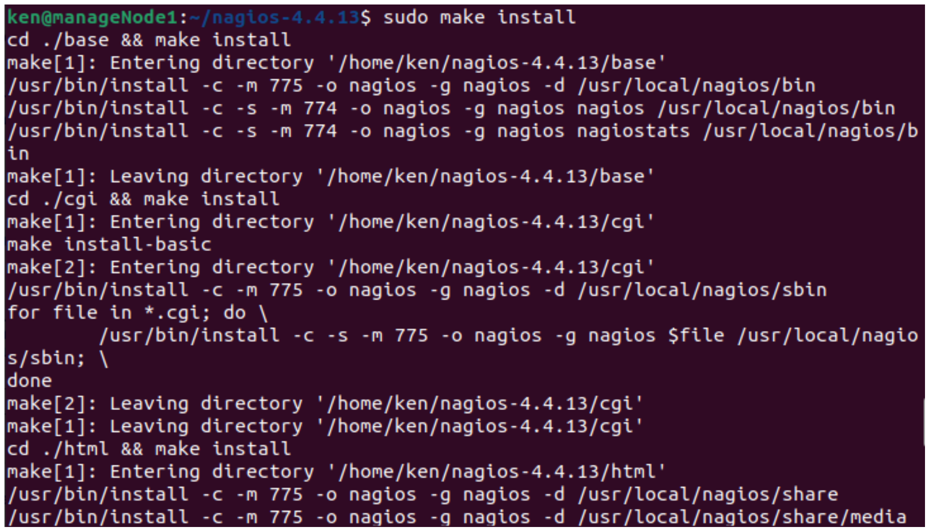
Ubuntu & CentOS



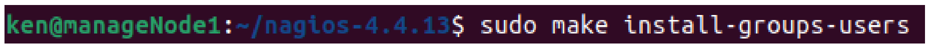
```
ken@manageNode1:~/nagios-4.4.13$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ANSI C header files... yes
```



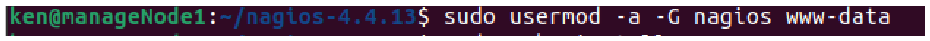
```
ken@manageNode1:~/nagios-4.4.13$ sudo make all
cd ./base && make
make[1]: Entering directory '/home/ken/nagios-4.4.13/base'
gcc -Wall -I.. -g -O2 -I/usr/include/openssl -DHAVE_CONFIG_H -DNSCORE -c -o nagi
os.o nagios.c
nagios.c: In function ‘main’:
nagios.c:611:25: warning: ignoring return value of ‘asprintf’ declared with attr
ibute ‘warn_unused_result’ [-Wunused-result]
  611 |                         asprintf(&mac->x[MACRO_PROCESSSTARTTIME], "%llu"
, (unsigned long long)program_start);
      |                         ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
nagios.c:841:25: warning: ignoring return value of ‘asprintf’ declared with attr
ibute ‘warn_unused_result’ [-Wunused-result]
  841 |                         asprintf(&mac->x[MACRO_EVENTSTARTTIME], "%llu",
(unsigned long long)event_start);
      |                         ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```



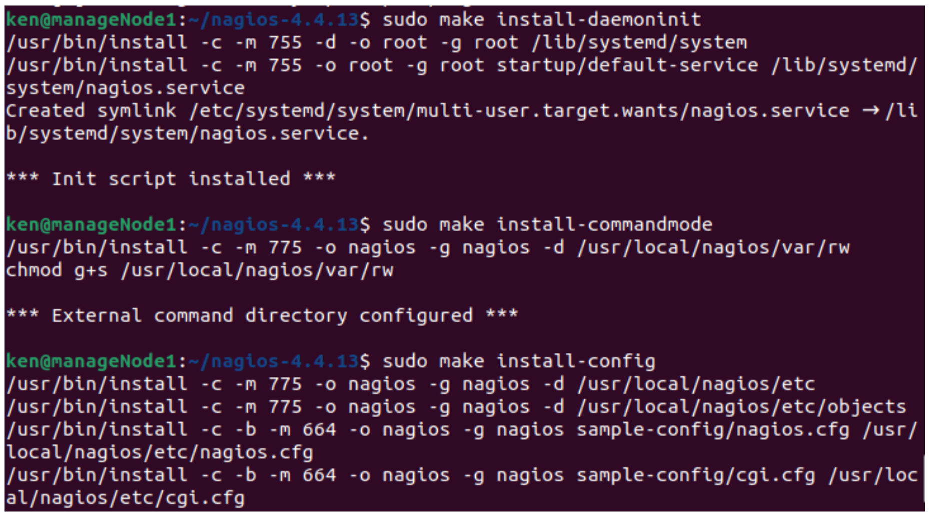
```
ken@manageNode1:~/nagios-4.4.13$ sudo make install
cd ./base && make install
make[1]: Entering directory '/home/ken/nagios-4.4.13/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiostats /usr/local/nagios/b
in
make[1]: Leaving directory '/home/ken/nagios-4.4.13/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ken/nagios-4.4.13/cgi'
make install-basic
make[2]: Entering directory '/home/ken/nagios-4.4.13/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
        /usr/bin/install -c -s -m 775 -o nagios -g nagios $file /usr/local/nagio
s/sbin; \
done
make[2]: Leaving directory '/home/ken/nagios-4.4.13/cgi'
make[1]: Leaving directory '/home/ken/nagios-4.4.13/cgi'
cd ./html && make install
make[1]: Entering directory '/home/ken/nagios-4.4.13/html'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/media
```



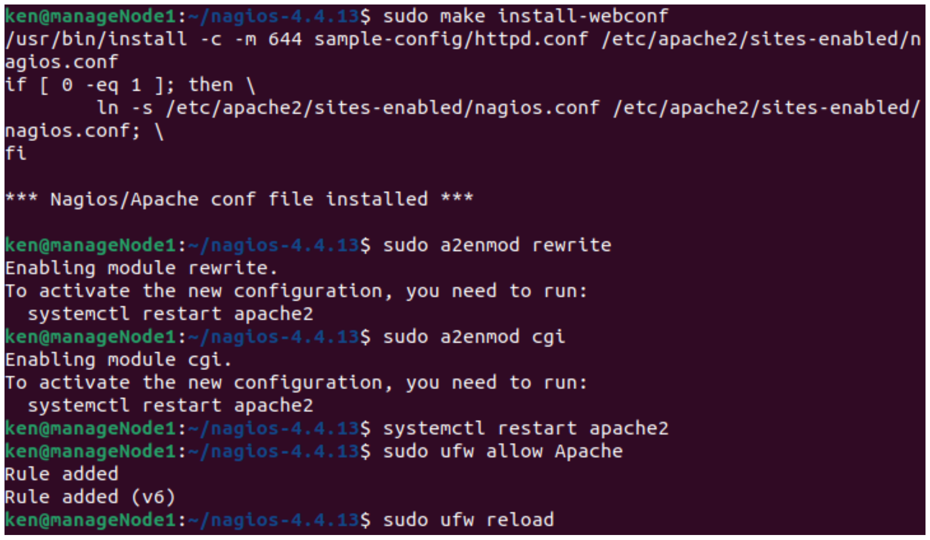
```
ken@manageNode1:~/nagios-4.4.13$ sudo make install-groups-users
```



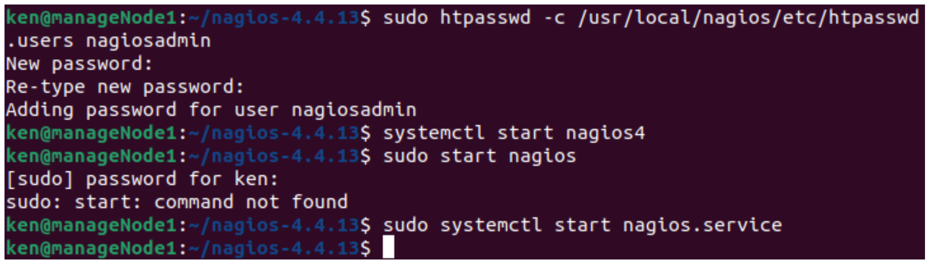
```
ken@manageNode1:~/nagios-4.4.13$ sudo usermod -a -G nagios www-data
```



```
ken@manageNode1:~/nagios-4.4.13$ sudo make install-daemoninit
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/
system/nagios.service
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /li
b/systemd/system/nagios.service.

*** Init script installed ***

ken@manageNode1:~/nagios-4.4.13$ sudo make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***

ken@manageNode1:~/nagios-4.4.13$ sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/
local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/loc
al/nagios/etc/cgi.cfg
```



```
ken@manageNode1:~/nagios-4.4.13$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-enabled/n
agios.conf
if [ 0 -eq 1 ]; then \
        ln -s /etc/apache2/sites-enabled/nagios.conf /etc/apache2/sites-enabled/
nagios.conf; \
fi

*** Nagios/Apache conf file installed ***

ken@manageNode1:~/nagios-4.4.13$ sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  systemctl restart apache2
ken@manageNode1:~/nagios-4.4.13$ sudo a2enmod cgi
Enabling module cgi.
To activate the new configuration, you need to run:
  systemctl restart apache2
ken@manageNode1:~/nagios-4.4.13$ systemctl restart apache2
ken@manageNode1:~/nagios-4.4.13$ sudo ufw allow Apache
Rule added
Rule added (v6)
ken@manageNode1:~/nagios-4.4.13$ sudo ufw reload
```

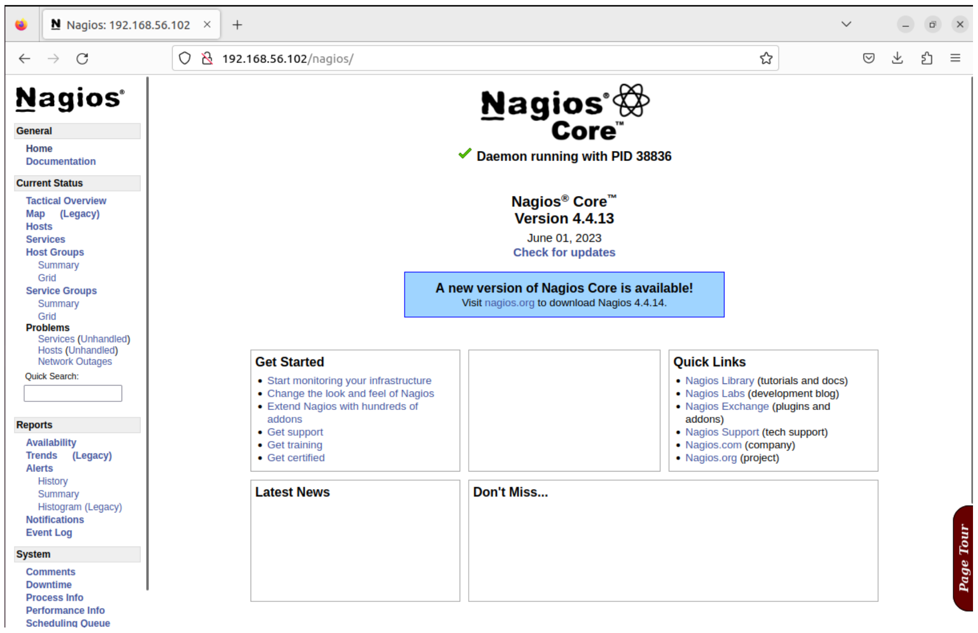


```
ken@manageNode1:~/nagios-4.4.13$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd
.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
ken@manageNode1:~/nagios-4.4.13$ systemctl start nagios4
ken@manageNode1:~/nagios-4.4.13$ sudo start nagios
[sudo] password for ken:
sudo: start: command not found
ken@manageNode1:~/nagios-4.4.13$ sudo systemctl start nagios.service
ken@manageNode1:~/nagios-4.4.13$
```

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

Step 9: Check if Nagios works for both Ubuntu and CentOS

UBUNTU



Nagios: 192.168.56.102

192.168.56.102/nagios/

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

Nagios® Core™

✓ Daemon running with PID 38836

Nagios® Core™
Version 4.4.13
June 01, 2023
Check for updates

A new version of Nagios Core is available!
Visit nagios.org to download Nagios 4.4.14.

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

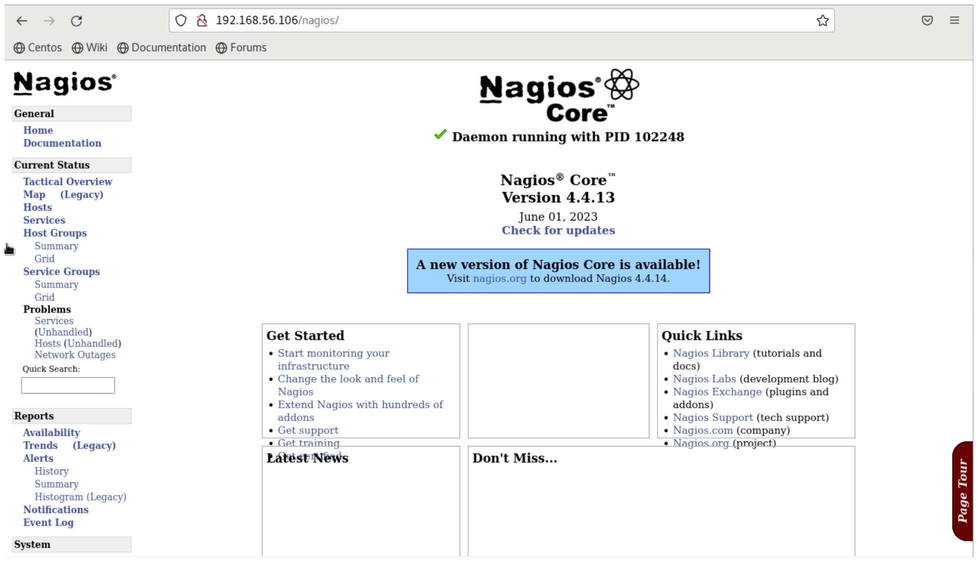
Quick Links
- Nagios Library (tutorials and docs)
- Nagios Labs (development blog)
- Nagios Exchange (plugins and addons)
- Nagios Support (tech support)
- Nagios.com (company)
- Nagios.org (project)

Latest News

Don't Miss...

Page Tour

CENTOS



192.168.56.106/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

Nagios® Core™

✓ Daemon running with PID 102248

Nagios® Core™
Version 4.4.13
June 01, 2023
Check for updates

A new version of Nagios Core is available!
Visit nagios.org to download Nagios 4.4.14.

Get Started
- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of addons
- Get support
- Get training

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)

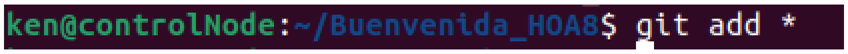
Latest News

Don't Miss...

Page Tour

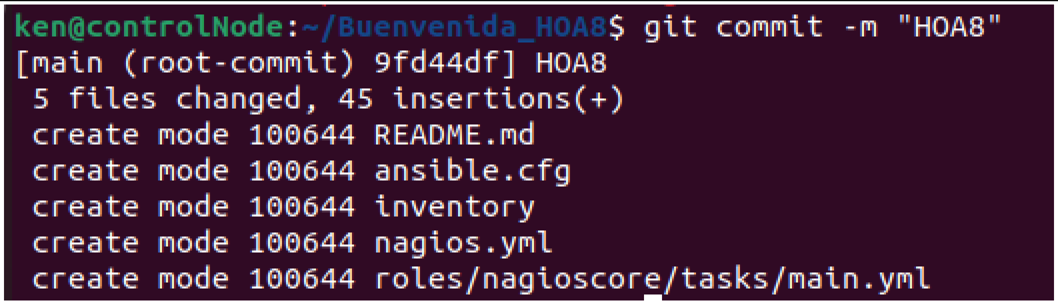
4. Make sure to create a new repository in GitHub for this activity.

Step 10: Git Add the activity you just did



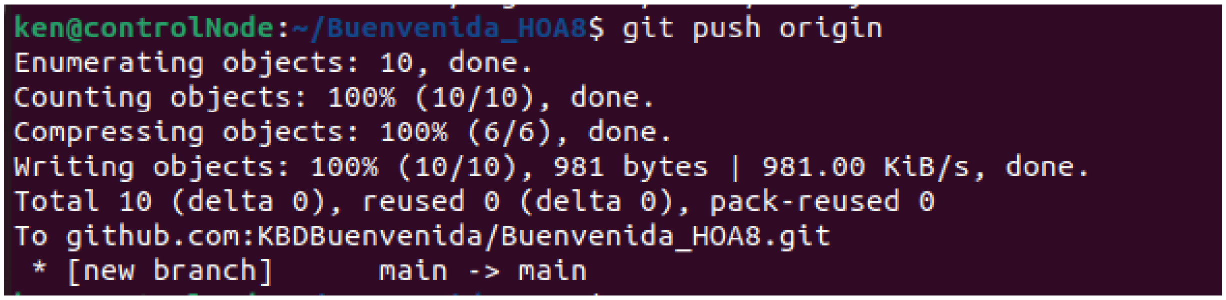
```
ken@controlNode:~/Buenvenida_HOA8$ git add *
```

Step 11: git commit -m origin after doing step 9



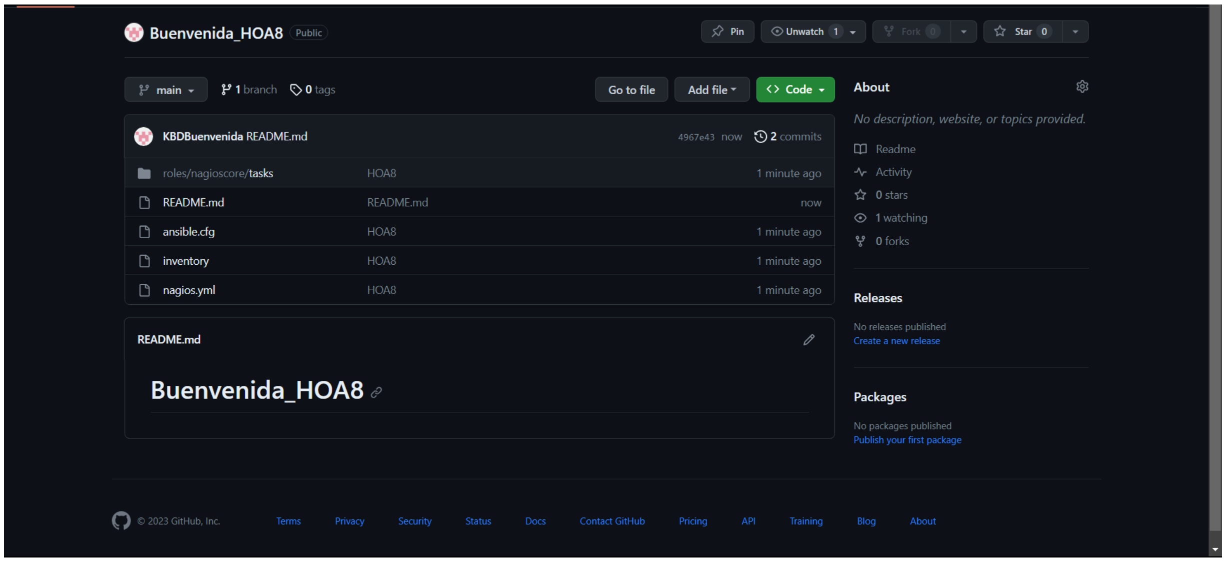
```
ken@controlNode:~/Buenvenida_HOA8$ git commit -m "HOA8"
[main (root-commit) 9fd44df] HOA8
 5 files changed, 45 insertions(+)
 create mode 100644 README.md
 create mode 100644 ansible.cfg
 create mode 100644 inventory
 create mode 100644 nagios.yml
 create mode 100644 roles/nagioscore/tasks/main.yml
```

Step 12: git push origin after doing step 10



```
ken@controlNode:~/Buenvenida_HOA8$ git push origin
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (10/10), 981 bytes | 981.00 KiB/s, done.
Total 10 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:KBDBuenvenida/Buenvenida_HOA8.git
 * [new branch]      main -> main
```

Step 13: Check the repository that you just created if it's updated.



No description, website, or topics provided.

Readme · Activity · 0 stars · 1 watching · 0 forks

Releases: No releases published. Create a new release

Packages: No packages published. Publish your first package

KBDBuenvenida README.md 4967e43 now 2 commits

roles/nagioscore/tasks	HOA8	1 minute ago
README.md	README.md	now
ansible.cfg	HOA8	1 minute ago
inventory	HOA8	1 minute ago
nagios.yml	HOA8	1 minute ago

README.md

Buenvenida_HOA8

© 2023 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

Reflections:

Answer the following:

1. What are the benefits of having an availability monitoring tool?
   - **It provides improved uptime that can help the user identify and resolve problems before they cause errors, It can also provide increased productivity that these monitoring tools help reduce outages to aid the user to quickly identify and resolve problems. It also provides a peace of mind to the user knowing that his system is being monitored by a**

monitoring tool that can easily identify what’s causing it and resolve it quickly.

Conclusions:

- In conclusion, I have been able to accomplish the activity properly by creating a manual on how to install Nagios for Ubuntu and CentOS. I encountered some problems at first but I was able to fix them by browsing the web and searching for solutions, one of the problems that I encountered was how to start Nagios in Ubuntu and CentOS and I found the solution at the Nagios Support, I just followed their instructions and It was running smoothly after that. I have realized the importance of using monitoring tools to prevent errors or outages in a system that if it happens be troublesome.