


Name: Tendencia, Jasmin Raiza S.	Date Performed: 10/16/2023
Course/Section: CPE232 - CPE31S4	Date Submitted: 10/17/2023
Instructor: Dr. Jonathan Taylar	Semester and SY: 1st/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	
<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 (screenshots and explanations)	
<ol style="list-style-type: none"> 1. Create a new repository in GitHub under the name of HOA8, and make sure that the repository is Public. Add a README file and input any related information regarding your inserted repository or leave it empty. 	

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 *

 jrstendencia ▾

Repository name *

HOA8_Tendencia

✔ HOA8_Tendencia is available.

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

Description (optional)



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

.gitignore template: None ▾

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

Choose a license

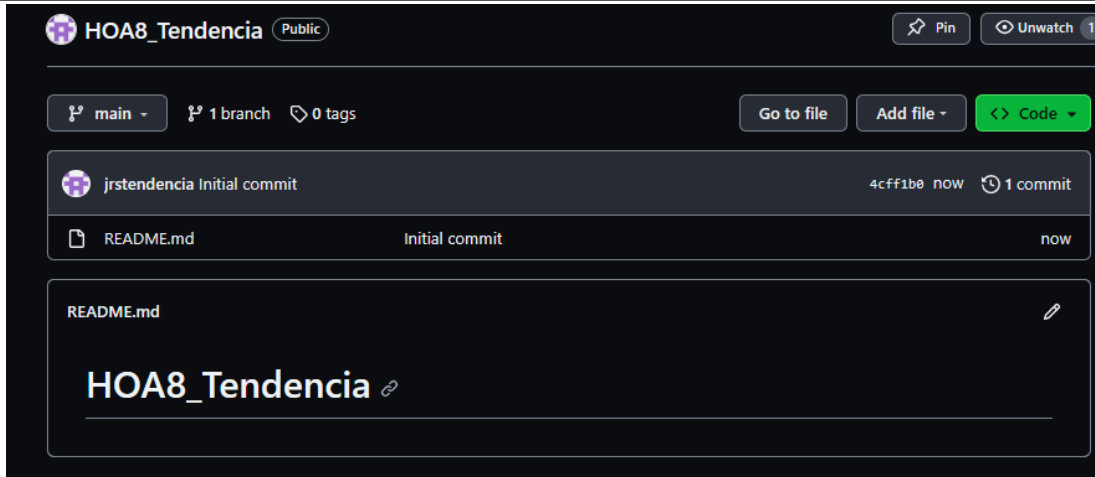
License: None ▾

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



You are creating a public repository in your personal account.

Create repository



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

```
tendencia@workstation:~$ git clone git@github.com:jrstendencia/HOA8_Tendencia.git
Cloning into 'HOA8_Tendencia'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

3. Create the base ansible structure `ansible.cfg` and inventory whereas the chosen servers, in my case I chose Server 1 - 192.168.56.102 and CentOS - 192.168.56.104.

Inventory:

```
[nagios_servers]
192.168.56.102 ansible_python_interpreter=/usr/bin/python3
192.168.56.104 ansible_python_interpreter=/usr/bin/python
```

ansible.cfg:

```
[defaults]

inventory = inventory
host_key_checking = False

deprecation warnings = False
remote user = tendencia
private_key_file = ~/.ssh/
```

4. 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. Lastly, inside the respective directory of the role itself create another directory named as tasks which consist a yml file named as `main.yml`.

```
tendencia@workstation:~/HOA8_Tendencia$ mkdir roles
```

```
tendencia@workstation:~/HOA8_Tendencia$ cd roles
tendencia@workstation:~/HOA8_Tendencia/roles$ mkdir start
tendencia@workstation:~/HOA8_Tendencia/roles$ cd start
tendencia@workstation:~/HOA8_Tendencia/roles/start$ mkdir tasks
tendencia@workstation:~/HOA8_Tendencia/roles/start$ cd tasks
tendencia@workstation:~/HOA8_Tendencia/roles/start/tasks$ sudo nano main.yml
```

```
tendencia@workstation:~/HOA8_Tendencia/roles/start$ mkdir restart
tendencia@workstation:~/HOA8_Tendencia/roles/start$ cd restart
tendencia@workstation:~/HOA8_Tendencia/roles/start/restart$ sudo nano main.yml
```

```
tendencia@workstation:~/HOA8_Tendencia/roles$ mkdir packages
tendencia@workstation:~/HOA8_Tendencia/roles$ cd packages
tendencia@workstation:~/HOA8_Tendencia/roles/packages$ sudo nano main.yml
```

tree:

```
├── roles
│   ├── packages
│   │   └── main.yml
│   ├── start
│   │   ├── restart
│   │   │   └── main.yml
│   │   └── tasks
│   │       └── main.yml
```

5. After creating the subdirectories in Roles Directories, proceed to input the commands on `main.yml` of each roles.

packages:

```

---
- name: Install Nagios packages (Ubuntu)
  apt:
    name: "{{ item }}"
    state: present
  loop:
    - nagios3
    - nagios-plugins
  when: "ansible_os_family == 'Debian'"

- name: Start and enable Nagios Service (Ubuntu)
  service:
    name: nagios3
    state: started
    enabled: yes
  when: "ansible_os_family == 'Debian'"

- name: Install Nagios packages (CentOS)
  yum:
    name: "{{ item }}"
    state: present
  loop:
    - nagios
    - nagios-plugins-all
  when: "ansible_os_family == 'RedHat'"

- name: Start and enable the Nagios service (CentOS)
  service:
    name: nagios
    state: started
    enabled: yes
  when: "ansible_os_family == 'RedHat'"

```

start/restart:

```

---
- name: restart nagios
  service:
    name: nagios
    state: restarted

```

start/tasks:

```

---
- name: Update package cache (Ubuntu)
  apt:
    update_cache: yes
  when: "ansible_os_family == 'Debian'"

- name: Install required packages
  package:
    name: "{{ item }}"
    state: present
  loop:
    - apache2
    - libapache2-mod-php
    - php-gd
    - libgd-dev
    - unzip
  when: "ansible_os_family == 'Debian'"

```

6. Prepare a .yml file for the global configuration playbook for the whole ansible and run the command *ansible-playbook --ask-become-pass nagios.yml* for the result and explain.

```

---
- name: Install Nagios (Ubuntu and CentOS)
  hosts: nagios_servers
  become: yes
  roles:
    - start
    - packages

```

Output:

```
tendencia@workstation:~/H0A8_Tendencia$ ansible-playbook --ask-become-pass nagios.yml
BECOME password:

PLAY [Install Nagios (Ubuntu and CentOS)] *****

TASK [Gathering Facts] *****
ok: [192.168.56.104]
ok: [192.168.56.102]

TASK [start : Update package cache (Ubuntu)] *****
skipping: [192.168.56.104]
changed: [192.168.56.102]

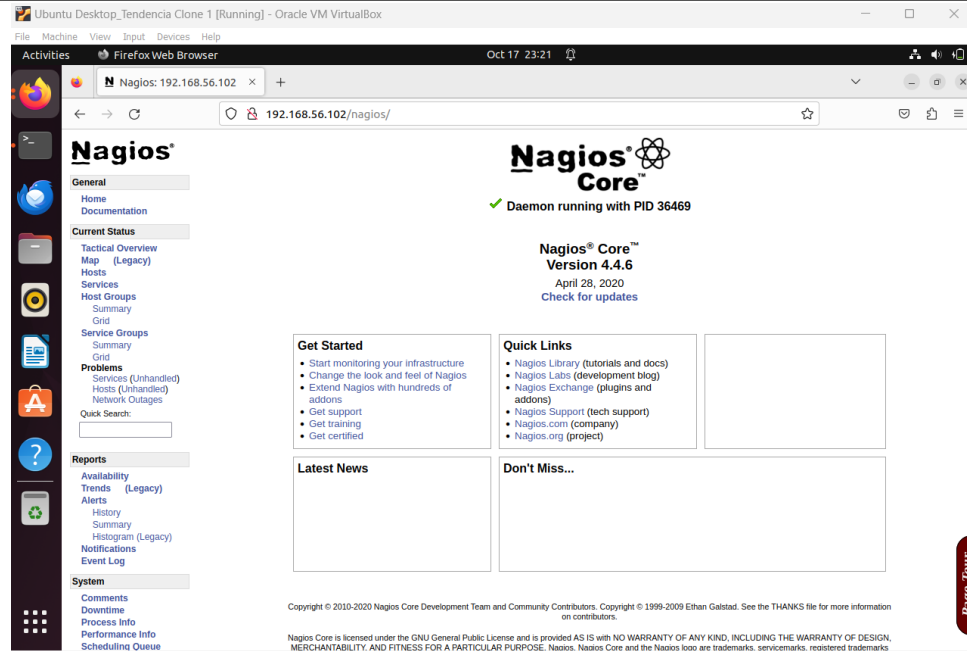
TASK [start : Install required packages] *****
skipping: [192.168.56.104] => (item=apache2)
skipping: [192.168.56.104] => (item=libapache2-mod-php)
skipping: [192.168.56.104] => (item=php-gd)
skipping: [192.168.56.104] => (item=libgd-dev)
skipping: [192.168.56.104] => (item=unzip)
skipping: [192.168.56.104]
ok: [192.168.56.102] => (item=apache2)
ok: [192.168.56.102] => (item=libapache2-mod-php)
changed: [192.168.56.102] => (item=php-gd)
ok: [192.168.56.102] => (item=libgd-dev)
ok: [192.168.56.102] => (item=unzip)

PLAY RECAP *****
192.168.56.102      : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.104      : ok=1    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

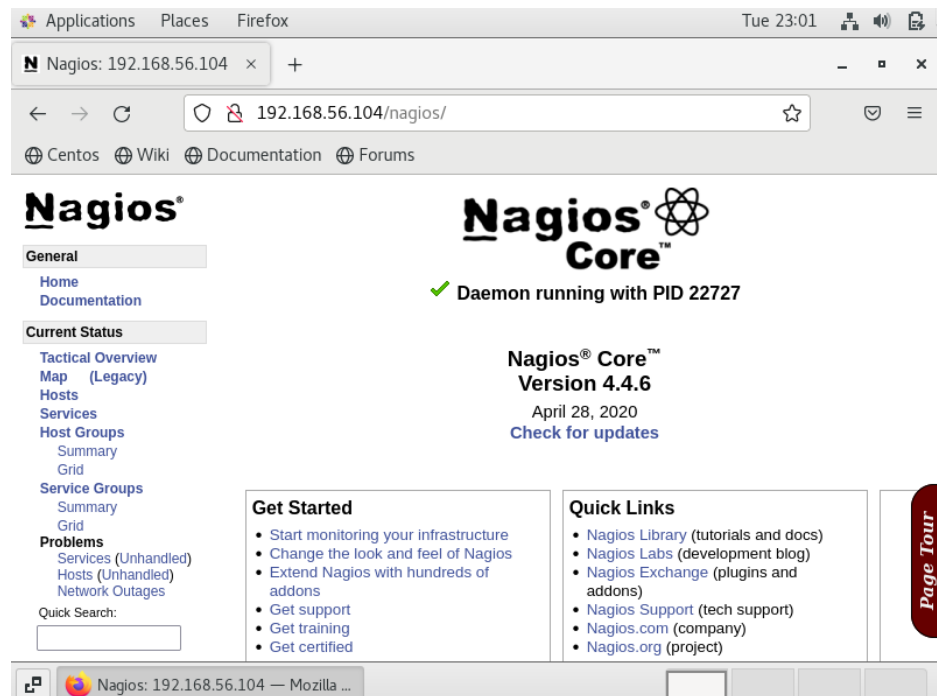
Explanation: The provided Ansible playbook executed with the following results:

- Two hosts were involved in the playbook execution, with IP addresses 192.168.56.102 (Ubuntu) and 192.168.56.104 (CentOS).
 - The "Gathering Facts" task collected information about both hosts, which completed successfully.
 - The "Update package cache (Ubuntu)" task was performed on the host with IP 192.168.56.102, but it was skipped on the host with IP 192.168.56.104.
 - The "Install required packages" task installed various packages (e.g., Apache, PHP, GD library) on the host with IP 192.168.56.102. Some of the packages were changed because they were not already installed, while others were skipped as they were already present.
 - The summary at the end of the playbook execution shows that for the host 192.168.56.102, three tasks were successful (ok), two tasks resulted in changes (changed), and there were no failures, unreachable hosts, or skipped tasks.
 - For the host 192.168.56.104, one task was successful (ok), no tasks resulted in changes (changed=0), two tasks were skipped, and there were no failures, unreachable hosts, or other issues.
7. To verify if Nagios was fully installed within the Servers, open a browser and insert the respective IP address of the server along with /nagios, In Server 1 - 192.168.56.102/nagios while in CentOS - 192.168.56.104/nagios

Ubuntu Server:



CentOS:



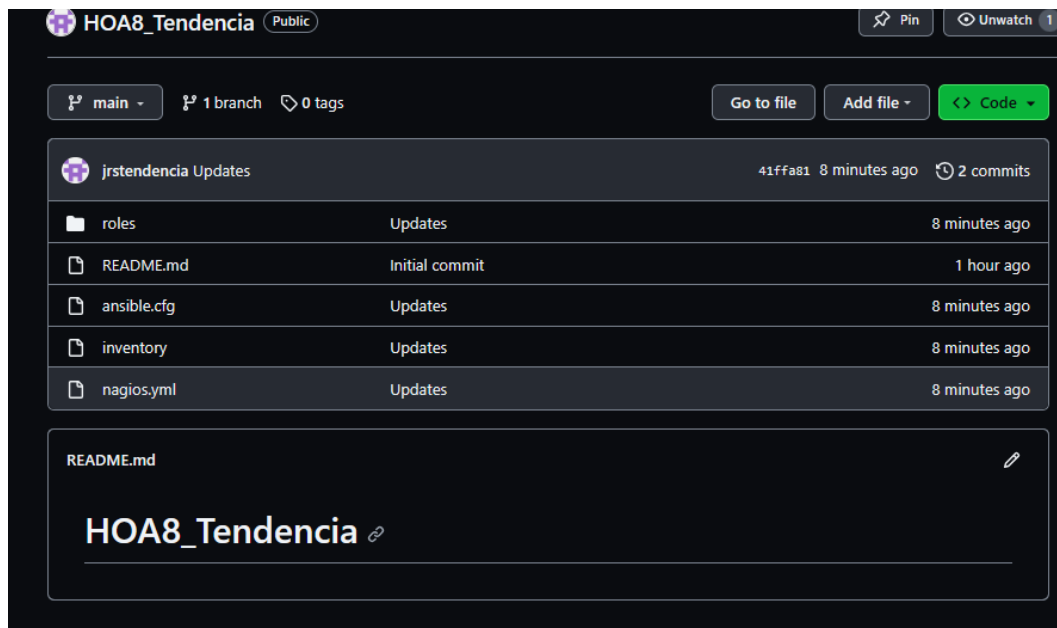
8. Sync the changes in Github.


```
tendencia@workstation:~/HOA8_Tendencia$ git add *
tendencia@workstation:~/HOA8_Tendencia$ git commit -m "Updates"

[main 41ffa81] Updates
6 files changed, 76 insertions(+)
create mode 100644 ansible.cfg
create mode 100644 inventory
create mode 100644 nagios.yml
create mode 100644 roles/packages/main.yml
create mode 100644 roles/start/restart/main.yml
create mode 100644 roles/start/tasks/main.yml
tendencia@workstation:~/HOA8_Tendencia$
tendencia@workstation:~/HOA8_Tendencia$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean
tendencia@workstation:~/HOA8_Tendencia$ git push origin main
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 2 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (13/13), 1.44 KiB | 1.44 MiB/s, done.
Total 13 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:jrstendencia/HOA8_Tendencia.git
4cff1b0..41ffa81 main -> main
```

Updated repository:



HOA8_Tendencia (Public) Pin Unwatch 1

main 1 branch 0 tags Go to file Add file <> Code

File	Commit Message	Time
roles	Updates	8 minutes ago
README.md	Initial commit	1 hour ago
ansible.cfg	Updates	8 minutes ago
inventory	Updates	8 minutes ago
nagios.yml	Updates	8 minutes ago

41ffa81 8 minutes ago 2 commits

jrstendencia Updates

README.md

HOA8_Tendencia

Reflections:

Answer the following:

- What are the benefits of having an availability monitoring tool?
 - Availability monitoring tools are crucial for managing IT infrastructure effectively. They provide benefits such as proactive issue detection, improved reliability, optimized performance, reduced downtime, resource optimization,

capacity planning, historical data analysis, security detection, SLA compliance, automation, user experience improvement, data-driven decision-making, and more. These tools ensure systems and services are available, reliable, and performing well, contributing to cost savings, user satisfaction, and efficient IT operations.

Conclusions:

This activity demonstrates how vital the availability monitoring tools for maintaining IT infrastructure health, preventing downtime, and optimizing performance. The provided Ansible playbook successfully executed on both Ubuntu and CentOS hosts, demonstrating effective system configuration. Availability monitoring, like Nagios, is essential for business continuity, and Ansible simplifies its implementation.

Another way:

tree:

```
tendencia@workstation:~/HOA8$ tree
.
├── ansible.cfg
├── global_playbook.yml
├── inventory
├── nagios.yml
├── README.md
└── roles
    ├── nagios4_Ubuntu
    │   └── tasks
    │       └── main.yml
    ├── nagios_CentOS
    │   └── tasks
    │       └── main.yml
    ├── nagios_Ubuntu
    │   └── tasks
    │       └── main.yml
    └── tasks

8 directories, 8 files
```

main playbook:

```
tendencia@workstation:~/HOA8$ cat global_playbook.yml
```

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

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

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

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

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

- hosts: nagios4_Ubuntu
  become: true
  roles:
    - nagios4_Ubuntu

- hosts: nagios_CentOS
  become: true
  roles:
    - nagios_CentOS
```