

Name:	Date Performed:
Course/Section:	Date Submitted:
Instructor:	Semester and SY:
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)	
<p>note: the repository used is HOA8 (created beforehand)</p> <p>Create the base ansible structure ansible.cfg and inventory whereas the chosen servers, in my case i chose ManagedNode and CentOS</p> <p>Inventory:</p> <pre>[nagios_servers] 192.168.56.103 ansible_python_interpreter=/usr/bin/python3 192.168.56.105 ansible_python_interpreter=/usr/bin/python</pre>	

ansible.cfg:

```
[default]
inventory = inventory
host_key_checking = False

decreation_warnings = False

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

i created directories tree consisting this:

```
joshua@ManagedNode:~/HOA8$ cd roles
joshua@ManagedNode:~/HOA8/roles$ tree
.
├── packages
│   └── main.yml
├── start
│   ├── restart
│   │   └── main.yml
│   └── tasks
│       └── main.yml
└── 5 directories, 3 files
joshua@ManagedNode:~/HOA8/roles$
```

after creating the sub directories shown above, I created commands for main.yml for each role.

Nagios.yml

```
Open ▾  [🔍]  nagios.yml ~ /HOA8
main.yml | main.yml | site.yml | r

| - -
- name: Install Nagios on Ubuntu and CentOS
  hosts: nagios_servers
  become: yes
  roles:
    - start
    - packages
```

start/restart:

```
Open ▾  [🔍]  main.yml ~ /HOA8/roles/start/restart
main.yml × | main.yml | site.yml |

| - -

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

packages:

```
Open ▾  main.yml  ~/HOA8/roles/packages
main.yml  main.yml

|---
- 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 Naguis packages (CentOs)
  yum:
    name: "[{ item }]"
    state: present
  loop:
    - nagios
    - nagos-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'"
```

output:

```
TASK [start : Update package cache (ubuntu)] *****
skipping: [192.168.56.103]
skipping: [192.168.56.105]

TASK [start : install required packages] *****
skipping: [192.168.56.105] => (item=apache2)
skipping: [192.168.56.105] => (item=libapache2-mod-php)
skipping: [192.168.56.105] => (item=php-gd)
skipping: [192.168.56.105] => (item=libgd-dev)
skipping: [192.168.56.105] => (item=unzip)
skipping: [192.168.56.105]
ok: [192.168.56.103] => (item=apache2)
ok: [192.168.56.103] => (item=libapache2-mod-php)
changed: [192.168.56.103] => (item=php-gd)
changed: [192.168.56.103] => (item=libgd-dev)
ok: [192.168.56.103] => (item=unzip)

PLAY RECAP *****
192.168.56.103      : ok=2    changed=1    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.105      : ok=1    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0

joshua@ManagedNode:~/HOA8$
```

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

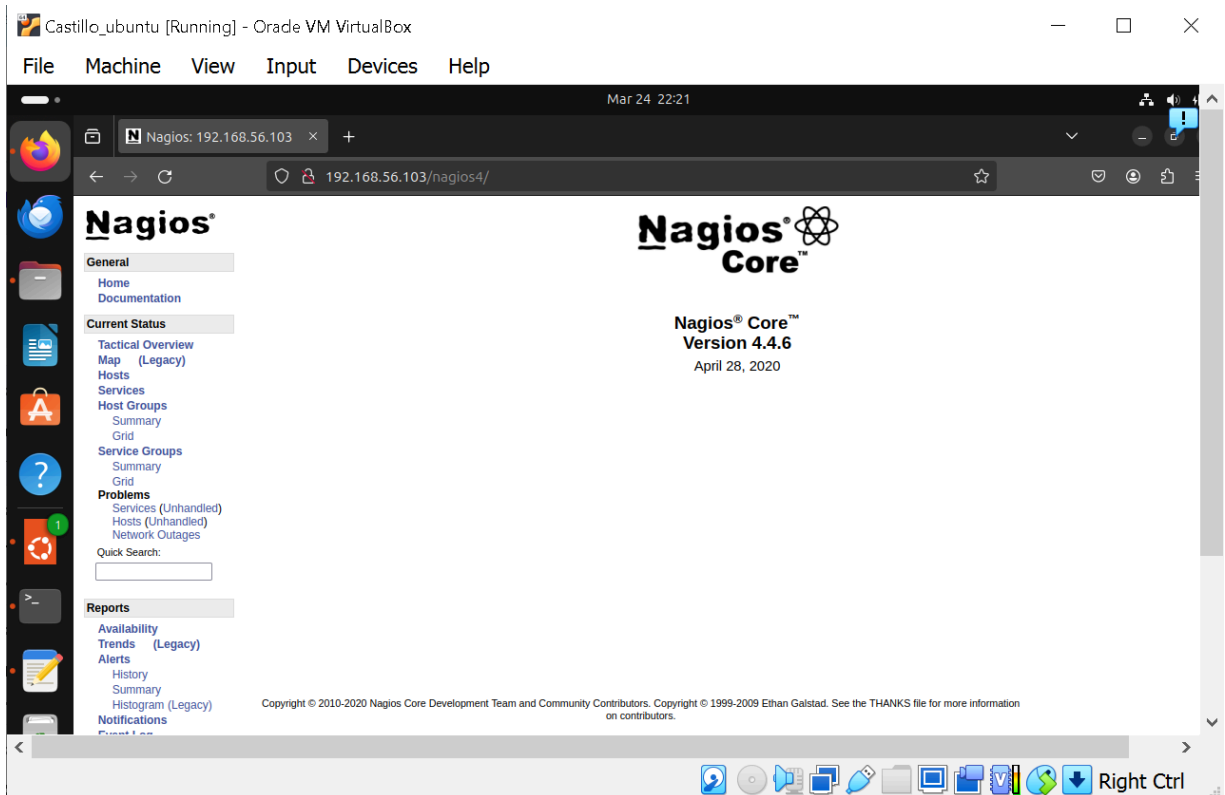
- Two hosts were involved in the playbook execution, with IP addresses 192.168.56.103 (Ubuntu) and 192.168.56.105 (CentOS).
- The "Gathering Facts" task collected information about both hosts, which completed successfully.
- The "Update package cache (Ubuntu)" task was performed on the host with IP 192.168.56.103, but it was skipped on the host with IP 192.168.56.105.
- The "Install required packages" task installed various packages (e.g., Apache, PHP, GD library) on the host with IP 192.168.56.103. 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 192.168.56.103, three tasks were successful (ok), two tasks resulted in

changes (changed), and there were no failures, unreachable hosts, or skipped tasks.

- For the host 192.168.56.105, 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.

Verify Nagios configuration

192.168.56.103/nagios4



192.168.56.105/nagios

Castillo_CentOS [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Applications Places Firefox Sun 22:43

Nagios: 192.168.56.105 x +

192.168.56.105/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:

Nagios® Core™

✓ Daemon running with PID 17510

Nagios® Core™
Version 4.4.14
August 01, 2023
[Check for updates](#)

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 Li](#)
- [Nagios L](#)
- [Nagios E](#)
- [Nagios E](#)
- [Nagios S](#)
- [Nagios.c](#)
- [Nagios.or](#)

Getting Started root@localhost:~ Nagios: 192.168.5...

Right Ctrl

add the configurations to github repository:

```
no changes added to commit (use "git add" and/or "git commit -a")
joshua@ManagedNode:~/HOA8$ git add .
joshua@ManagedNode:~/HOA8$ git status
On branch main
Your branch is based on 'origin/main', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    playbook.yml

joshua@ManagedNode:~/HOA8$ git push origin main
Enumerating objects: 24, done.
Counting objects: 100% (24/24), done.
Delta compression using up to 2 threads
Compressing objects: 100% (17/17), done.
Writing objects: 100% (24/24), 3.45 KiB | 1.72 MiB/s, done.
Total 24 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To github.com:qjlcastillo/HOA8.git
 * [new branch]      main -> main
joshua@ManagedNode:~/HOA8$ git pull
Already up to date.
joshua@ManagedNode:~/HOA8$
```

The screenshot shows the GitHub web interface for the repository 'HOA8' by user 'qjlcastillo'. The repository is public and has 1 branch (main) and 0 tags. The file list shows the following files, all committed 1 minute ago:

File Name	Commit Hash	Commit Time
roles	hoa8	1 minute ago
global_playbook.yml	hoa8	1 minute ago
inventory	hoa8	1 minute ago
nagios.yml	hoa8	1 minute ago
playbook.yml	hoa8	1 minute ago

Reflections:

Answer the following:

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

Having an availability monitoring tool like Nagios provides several significant benefits for organizations. It enables proactive monitoring, allowing businesses to identify and address issues before they impact operations or users. By continuously checking the availability and performance of systems, services, and network devices, Nagios helps ensure high uptime and reliability.

Nagios also offers performance monitoring capabilities, allowing organizations to track resource utilization, identify bottlenecks, and optimize system performance. Its customizable and flexible nature allows tailoring to specific monitoring needs, making it suitable for diverse environments.

Conclusions:

In this Activity , we explored the process of installing, configuring, and managing availability monitoring tools. Specifically, we focused on Nagios as an example of such a tool.

By effectively installing, configuring, and managing availability monitoring tools like Nagios, organizations can enhance system reliability, reduce downtime, improve incident response, and ensure optimal performance. These tools contribute to better customer satisfaction, productivity, and overall business success.