

<b>Name: Jozshua Amiel Alonzo</b>	<b>Date Performed: October 20, 2022</b>
<b>Course/Section: CPE31S23</b>	<b>Date Submitted:</b>
<b>Instructor: Dr. Jonathan Taylar</b>	<b>Semester and SY: 2022-2023</b>
<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>	
<p><b>Github Link:</b> <a href="https://github.com/jozshua/HOA8_Alonzo.git">https://github.com/jozshua/HOA8_Alonzo.git</a></p> <p>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.</p>	
<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> (screenshots and explanations)	

```

GNU nano 6.2                                site.yml
-
hosts: all
become: true
pre_tasks:
- name: install updates (Ubuntu)
  tags: always
  apt:
    update_cache: yes
    changed_when: false
    when: ansible_distribution == "Ubuntu"
- name: update repository index (CentOS)
  tags: always
  dnf:
    update_cache: yes
    changed_when: false
    when: ansible_distribution == "CentOS"

```

Create the site.yml file and type these codes inside, these codes are for the two server distributions. Run the playbook.

```

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.112]
ok: [192.168.56.116]

TASK [install updates (Ubuntu)] *****
*
skipping: [192.168.56.116]
ok: [192.168.56.112]

TASK [update repository index (CentOS)] *****
*
skipping: [192.168.56.112]
ok: [192.168.56.116]

PLAY RECAP *****
*
192.168.56.112      : ok=2    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.116      : ok=2    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0

```

From running this playbook the tasks were successfully applied on each server distribution. There are 2 skipped states in each server because of having the different distribution from the ansible command.

```
jozshua@jozshua-VirtualBox:~/HOA8_Alonzo/cpe_HOA8$ tree
.
├── ansible.cfg
├── inventory
├── roles
│   ├── nagios
│   │   └── tasks
│   │       └── main.yml
└── site.yml
```

Create the roles, nagios and tasks directories for creating the main.yml file. Issue the command tree to display the route.

```
GNU nano 6.2 site.yml
--
- hosts: all
  become: true
  pre_tasks:

    - name: install updates (Ubuntu)
      tags: always
      apt:
        update_cache: yes
        changed_when: false
        when: ansible_distribution == "Ubuntu"

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

- hosts: all
  become: true
  roles:
    - nagios
```

Edit the site.yml file and insert these codes below.

```

GNU nano 6.2                                     main.yml
- name: nagios installation (Ubuntu)
  apt:
    name:
      - nagios4
    state: latest
    update_cache: yes
  when: ansible_distribution == "Ubuntu"

- name: nagios installation (CentOS)
  dnf:
    name:
      - nagios
    state: latest
    update_cache: yes
  when: ansible_distribution == "CentOS"

```

Apply the concept of creating roles and from the tasks directory create the main.yml file. After that go back to the original directory by issuing the command “cd ..” Run the playbook once again.

```

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.112]
ok: [192.168.56.116]

TASK [install updates (Ubuntu)] *****
*
skipping: [192.168.56.116]
ok: [192.168.56.112]

TASK [update repository index (CentOS)] *****
*
skipping: [192.168.56.112]
ok: [192.168.56.116]

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.116]
ok: [192.168.56.112]

```

```

TASK [nagios : nagios installation (Ubuntu)] *****
*
skipping: [192.168.56.116]
changed: [192.168.56.112]

TASK [nagios : nagios installation (CentOS)] *****
*
skipping: [192.168.56.112]
changed: [192.168.56.116]

PLAY RECAP *****
*
192.168.56.112      : ok=4    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.116      : ok=4    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0

```

The tasks were successfully executed which is displayed below that the 4 tasks were displayed in ok state. There are also a total of 4 skipped states because of the different server distributions for each task. Also, there is one changed state in each server from the task of installing the nagios.

### Showing the output for installation of nagios:

For Ubuntu:

```

jozshua@Server1-VirtualBox:~$ nagios4 -- version

Nagios Core 4.4.6
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2020-04-28
License: GPL

```

For CentOS:

```

[jozshua@localhost ~]$ nagios --version

Nagios Core 4.4.6
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2020-04-28
License: GPL

```

Both server distributions have the Nagios core 4.4.6 version installed in it.

### **Reflections:**

Answer the following:

1. What are the benefits of having an availability monitoring tool? The benefits of having this monitoring tool could be helped to reduce the manual effort. It can also help to detect the issues in advance so that you can solve it immediately without encountering the error. It could also fix the issues automatically instead of issuing that there are issues. It can also prevent site downtime where you can avoid losing time and money from fixing the issue.

### **Conclusions:**

In this activity, Using ansible as an Infrastructure as Code tool I created and designed the workflow of installing the nagios for both server distributions. Installing the nagios for both server distributions needs a site.yml file and main.yml file. Inside these files there are codes that are necessary for executing the tasks for the ansible playbooks for the output. After running the playbook successfully I am able to display the installation of nagios for both servers by issuing the nagios --version command. Therefore there is no sign of failure from the outputs so I conclude that I had finished this activity well.