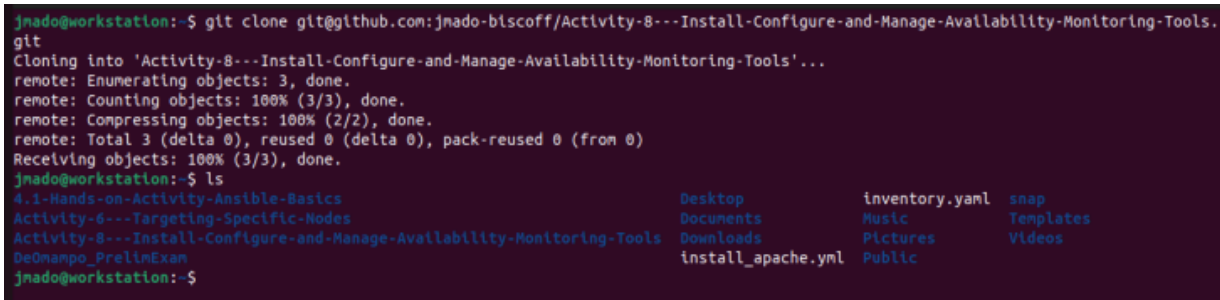


Name: De Omampo, Julius Mark A.	Date Performed: 10/14/2024
Course/Section: CPE212 - CPE31S2	Date Submitted:
Instructor: Engr. Robin Valenzuela	Semester and SY:
Activity 8: Install, Configure, and Manage Availability Monitoring tools	
1. Objectives	
Create and design a workflow that installs, configure 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	
 <pre> jnado@workstation:~\$ git clone git@github.com:jnado-biscoff/Activity-8---Install-Configure-and-Manage-Availability-Monitoring-Tools. git Cloning into 'Activity-8---Install-Configure-and-Manage-Availability-Monitoring-Tools'... remote: Enumerating objects: 3, done. remote: Counting objects: 100% (3/3), done. remote: Compressing objects: 100% (2/2), done. remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0) Receiving objects: 100% (3/3), done. jnado@workstation:~\$ ls 4.1-Hands-on-Activity-Ansible-Basics Desktop inventory.yaml snap Activity-6---Targeting-Specific-Nodes Documents Music Templates Activity-8---Install-Configure-and-Manage-Availability-Monitoring-Tools Downloads Pictures Videos DeOmampo_PrelimExam install_apache.yml Public jnado@workstation:~\$ </pre>	
<i>Clone Github repository to local workstation</i>	

```

GNU nano 6.2                                inventory.yaml *
[server1_ubuntu]
192.168.56.109

[server2_ubuntu]
192.168.56.108

[server3_centos]
192.168.56.110

```

inventory.yaml configuration

```

jnado@workstation:~/Activity-8---Install-Configure-and-Manage-Availability-Monitoring-Tools/roles$ mkdir base server1_ubuntu server2_ubuntu server3_centos
jnado@workstation:~/Activity-8---Install-Configure-and-Manage-Availability-Monitoring-Tools/roles$ ls
base server1_ubuntu server2_ubuntu server3_centos
jnado@workstation:~/Activity-8---Install-Configure-and-Manage-Availability-Monitoring-Tools/roles$

```

roles directory creation

```

GNU nano 6.2                                execute.yml *
---
- hosts: all
  become: true
  pre_tasks:

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

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

- hosts: all
  become: true
  roles:
    - base

- hosts: server1_ubuntu
  become: true
  roles:
    - server1_ubuntu

- hosts: server2_ubuntu
  become: true
  roles:
    - server2_ubuntu

- hosts: server3_centos
  become: true

roles:
  - server3_centos

```

execute.yml playbook

```
GNU nano 6.2 tasks.yml *
---
- name: install updates (CentOS)
  tags: always,update
  dnf:
    update_only: yes
    update_cache: yes
  when: ansible_distribution == "CentOS"

- name: install updates (Ubuntu)
  tags: always,update
  apt:
    upgrade: dist
    update_cache: yes
  when: ansible_distribution == "Ubuntu"
```

base tasks.yml

```
GNU nano 6.2 server1_ubuntu/tasks.yml
---
- name: Install Nagios (server1_ubuntu)
  tags: nagios, server1
  apt:
    name:
      - nagios4
      - nagios-plugins
      - nagios4-core
      - apache2
    update_cache: yes
    state: present
  when: ansible_distribution == "Ubuntu"
```

server1_ubuntu tasks.yml

```
GNU nano 6.2 server2_ubuntu/tasks.yml *
---
- name: Install Nagios (server2_ubuntu)
  tags: nagios, server1
  apt:
    name:
      - nagios4
      - nagios-plugins
      - nagios4-core
      - apache2
    update_cache: yes
    state: present
  when: ansible_distribution == "Ubuntu"
```

server2_ubuntu tasks.yml

```
GNU nano 6.2 tasks.yml *
---
- name: install nagios (server3_centos)
  dnf:
    name:
      - nagios
      - nagios-plugins-all
      - httpd
    state: present
  when: ansible_distribution == "CentOS"
```

server3_centos tasks.yml

```
jnado@workstation:~/Activity-8---Install-Configure-and-Manage-Availability-Monitoring-Tools$ ansible-playbook --ask-become-pass execute.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.109]
ok: [192.168.56.108]
[DEPRECATION WARNING]: Distribution centos 9 on host 192.168.56.110 should use /usr/libexec/platform-python, but is using /usr/bin/python for backward compatibility with prior Ansible releases. A future Ansible release will default to using the discovered platform python for this host. See https://docs.ansible.com/ansible/2.10/reference_appendices/interpreter_discovery.html for more information. This feature will be removed in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
ok: [192.168.56.110]

TASK [update repository index (CentOS)] *****
skipping: [192.168.56.109]
skipping: [192.168.56.108]
ok: [192.168.56.110]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.110]
ok: [192.168.56.109]
ok: [192.168.56.108]

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.109]
ok: [192.168.56.108]
ok: [192.168.56.110]

PLAY [server1_ubuntu] *****

TASK [Gathering Facts] *****
ok: [192.168.56.109]

PLAY [server2_ubuntu] *****

TASK [Gathering Facts] *****
ok: [192.168.56.108]

PLAY [server3_centos] *****

TASK [Gathering Facts] *****
ok: [192.168.56.110]

PLAY RECAP *****
192.168.56.108      : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.109      : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.110      : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

playbook execution

```
julius-de-omampo@server1:~$ 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

Website: https://www.nagios.org
This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License version 2 as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
```

Reflections:

Answer the following:

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

Using Nagios4 for availability monitoring offers real-time issue detection and minimizes downtime by sending automated alerts when problems arise. It monitors system performance, tracks uptime and resource usage, and provides historical data for reporting and capacity planning. Nagios4 helps maintain SLA compliance, optimize infrastructure, and enhance security by identifying anomalies. With its centralized management, Nagios4 simplifies monitoring across large networks, ensuring reliable system performance and efficient operations.

Conclusions:

Using a playbook to deploy Nagios4 automates the installation and configuration process, ensuring a consistent and efficient setup across multiple systems. By leveraging Ansible, administrators can streamline the deployment of Nagios4, reducing manual errors and saving time. This approach simplifies the management of large-scale monitoring environments, allowing for quick scalability, centralized monitoring, and improved system reliability, all while maintaining the flexibility and power of Nagios4's monitoring capabilities.