Name: Villasenor, Hans Rainier A.	Date Performed: 10-17-25
Course/Section: CPE31S2	Date Submitted: 10-17-25
Instructor: Robin Valenzuela	Semester and SY: 1st sem 2025-26

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.

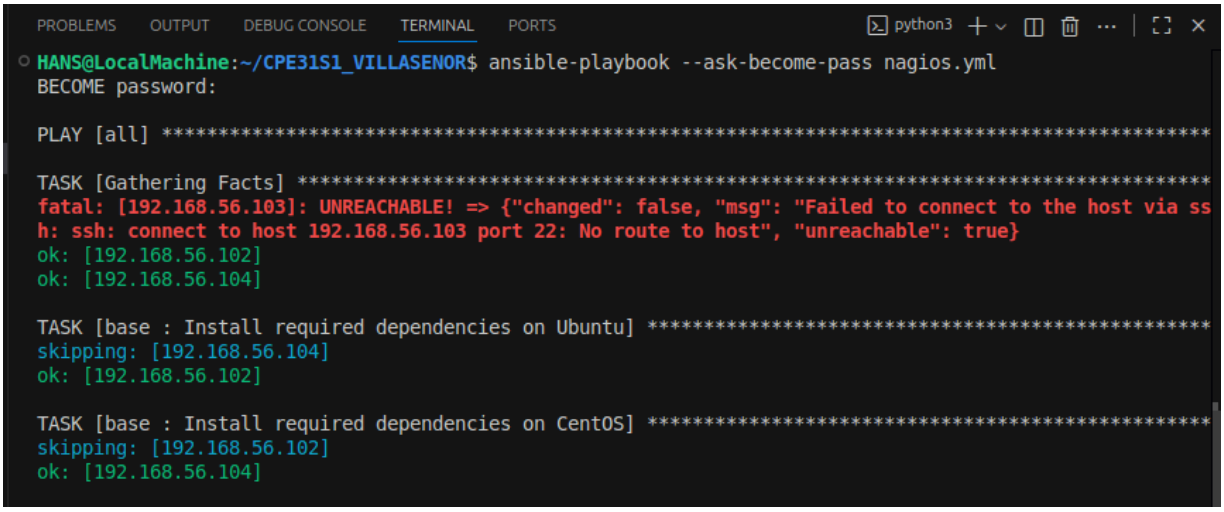
2. Discussion

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

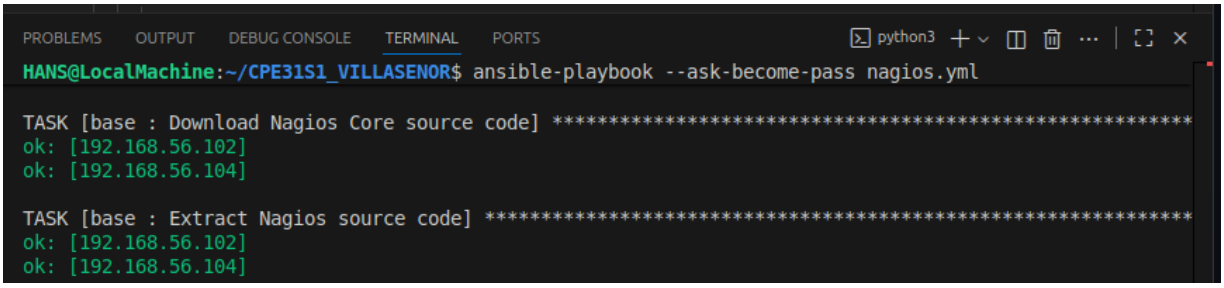
1. Create a playbook that installs Nagios in both Ubuntu and CentOS. Apply the concept of creating roles.
2. Describe how you did step 1. (Provide screenshots and explanations in your report. Make your report detailed such that it will look like a manual.)
3. Show an output of the installed Nagios for both Ubuntu and CentOS.
4. Make sure to create a new repository in GitHub for this activity.

4. **Output** (screenshots and explanations)

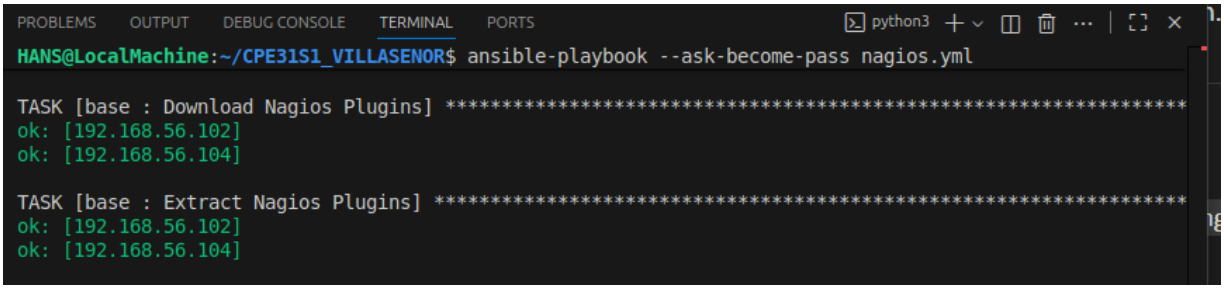


In this image it show the successfully installed dependencies on both servers.

the dependencies these are the tools needed to install nagios in both servers

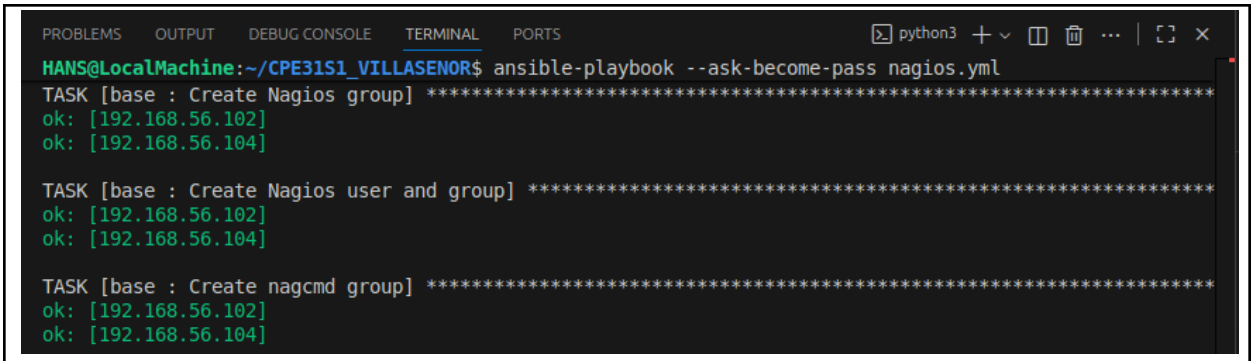


This part shown is to download the source file of nagios and extract it or unzip it.

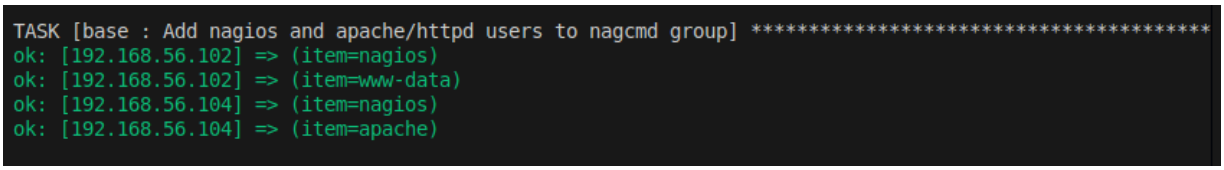


this part shown is to download the nagios plugins and extract it.

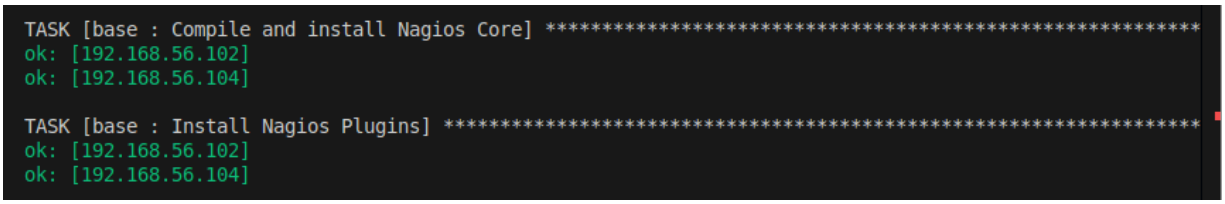
Nagios Plugins is a tool to monitor the service of nagios.



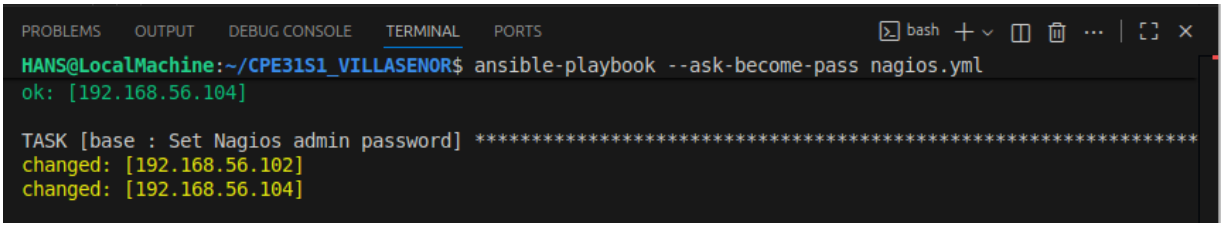
In this part it creates the needed users and groups for Nagios. The nagios user and group are made for running Nagios, and the nagcmd group lets Nagios and the web server work together safely.



In this part shown the install of the apache and httpd for web command that will be sent in Nagios



In this part shown the compilation and installation of the nagios core and plugins



In this part shows to add an admin password in nagios

```
TASK [base : Enable and start Apache/Httpd service on Ubuntu] ****************************************
skipping: [192.168.56.104]
ok: [192.168.56.102]

TASK [base : Enable and start Apache/Httpd service on CentOS] ****************************************
skipping: [192.168.56.102]
ok: [192.168.56.104]
```

In this part shows to enable the apache and httpd in both servers

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                                   bash

HANS@LocalMachine:~/CPE31S1_VILLASENOR$ ansible-playbook --ask-become-pass nagios.yml
TASK [base : Enable and start Nagios service] ********************************************************
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [base : Enable external command execution in Nagios] ********************************************
ok: [192.168.56.102]
ok: [192.168.56.104]
```

This part shows the start and enable the Nagios service so it runs automatically, and also allows external commands in the configuration so Nagios can take actions from the web interface.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                                   bash

HANS@LocalMachine:~/CPE31S1_VILLASENOR$ ansible-playbook --ask-become-pass nagios.yml
TASK [base : Restart Nagios service to apply changes] ************************************************
changed: [192.168.56.102]
changed: [192.168.56.104]

TASK [base : Restart Apache/Httpd to apply changes on Ubuntu] ****************************************
skipping: [192.168.56.104]
changed: [192.168.56.102]

TASK [base : Restart Apache/Httpd to apply changes on CentOS] ****************************************
skipping: [192.168.56.102]
changed: [192.168.56.104]
```

This part shows the restart Nagios and the web server (Apache or Httpd) to apply all the new settings and make sure everything works properly.

PLAYBOOK

```
---

  - hosts: all
    become: true
    roles:
      -  base

  - hosts: Ubuntu, CentOS
    become: true
    roles:
      - workstations

---

  - name: Install required dependencies on Ubuntu
    apt:
      name:
        - gcc
        - libc6
        - make
        - wget
        - unzip
        - apache2
        - php
        - libgd-dev
        - openssl
        - libssl-dev
        - autoconf
        - bc
        - gawk
        - dc
        - build-essential
        - snmp
        - libnet-snmp-perl
        - gettext
      state: present
    when: ansible_distribution == "Ubuntu"
```

```
  - name: Install required dependencies on CentOS
    yum:
      name:
        - gcc
        - glibc
        - glibc-common
        - wget
        - unzip
        - httpd
        - php
        - gd
        - gd-devel
        - perl
        - postfix
        - openssl
        - openssl-devel
        - make
        - autoconf
      state: present
    when: ansible_distribution == "CentOS"

  - name: Download Nagios Core source code
    get_url:
      url:
"https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.6
.tar.gz"
      dest: /tmp/nagios-4.5.6.tar.gz

  - name: Extract Nagios source code
    unarchive:
      src: /tmp/nagios-4.5.6.tar.gz
      dest: /tmp
      remote_src: yes

  - name: Download Nagios Plugins
    get_url:
      url:
"https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz"
      dest: /tmp/nagios-plugins-2.4.11.tar.gz
```

```
  - name: Extract Nagios Plugins
    unarchive:
      src: /tmp/nagios-plugins-2.4.11.tar.gz
      dest: /tmp
      remote_src: yes

  - name: Create Nagios group
    group:
      name: nagios

  - name: Create Nagios user and group
    user:
      name: nagios
      group: nagios

  - name: Create nagcmd group
    group:
      name: nagcmd

  - name: Add nagios and apache/httpd users to nagcmd group
    user:
      name: "{{ item }}"
      groups: nagcmd
      append: yes
    loop:
      - nagios
      - "{{ 'www-data' if ansible_os_family == 'Debian' else 'apache'
}}"

  - name: Compile and install Nagios Core
    shell: |
      cd /tmp/nagios-4.5.6
      ./configure --with-command-group=nagcmd
      make all
      make install
      make install-init
      make install-commandmode
      make install-config
      make install-webconf
```

```
    args:
      creates: /usr/local/nagios/bin/nagios

  - name: Install Nagios Plugins
    shell: |
      cd /tmp/nagios-plugins-2.4.11
      ./configure --with-nagios-user=nagios
--with-nagios-group=nagios
      make
      make install
    args:
      creates: /usr/local/nagios/libexec/check_http

  - name: Set Nagios admin password
    command: htpasswd -b -c /usr/local/nagios/etc/htpasswd.users
zamora_admin "sample"

  - name: Enable and start Apache/Httpd service on Ubuntu
    service:
      name: apache2
      enabled: yes
      state: started
    when: ansible_distribution == "Ubuntu"

  - name: Enable and start Apache/Httpd service on CentOS
    service:
      name: httpd
      enabled: yes
      state: started
    when: ansible_distribution == "CentOS"

  - name: Enable and start Nagios service
    service:
      name: nagios
      enabled: yes
      state: started

  - name: Enable external command execution in Nagios
    lineinfile:
      path: /usr/local/nagios/etc/nagios.cfg
```

```
    regexp: '^#?check_external_commands='
    line: 'check_external_commands=1'

- name: Restart Nagios service to apply changes
  service:
    name: nagios
    state: restarted

- name: Restart Apache/Httpd to apply changes on Ubuntu
  service:
    name: apache2
    state: restarted
  when: ansible_distribution == "Ubuntu"

- name: Restart Apache/Httpd to apply changes on CentOS
  service:
    name: httpd
    state: restarted
  when: ansible_distribution == "CentOS"
```

Reflections:

Answer the following:

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

An availability monitoring tool helps ensure systems are always running, alerts you when problems occur, and lets you fix issues quickly to avoid downtime and keep everything reliable.

Conclusions:

In conclusion, this activity helped me learn how to install Nagios automatically on both Ubuntu and CentOS using an Ansible playbook. I understood how important monitoring tool in system.