Name: Baltazar, Paul Eimar R.	**Date Performed:** Oct 14, 2024
Course/Section: CPE212 - CPE31S2	**Date Submitted:** Oct 16, 2024
Instructor: Engr. Robin Valenzuela	**Semester and SY:** 1st Sem 2024 - 2025

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.

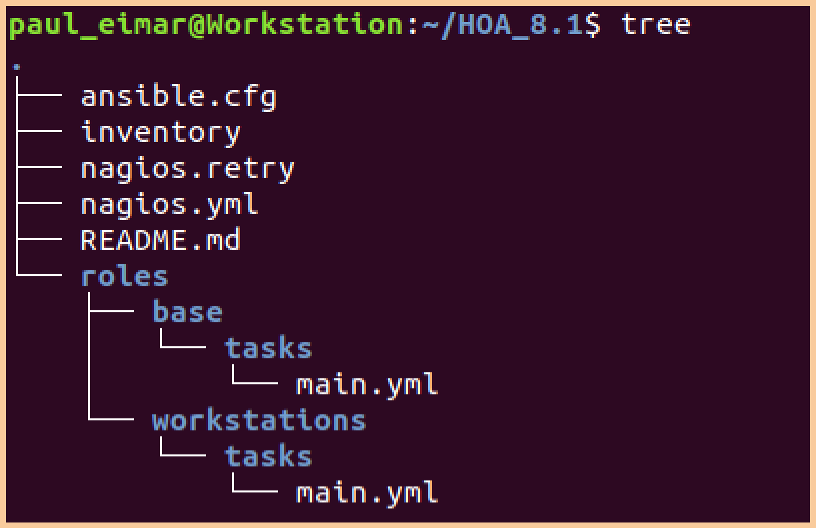
3. Tasks

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)

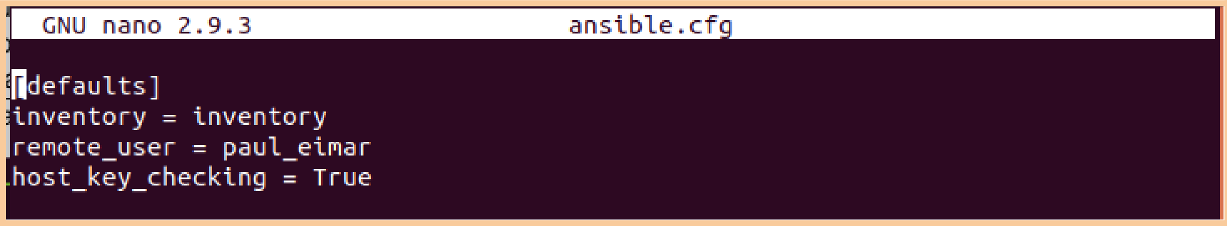
Setting up the playbook

Create a directory for your playbook. They should contain an ansible configuration file, inventory file, a roles folder; which is where we are going to place the main.yml files that contains the command for a certain group, and the playbook itself, which is going to run the command inside the roles folder.



```
paul_eimar@Workstation:~/HOA_8.1$ tree
.
├── ansible.cfg
├── inventory
├── nagios.retry
├── nagios.yml
├── README.md
└── roles
    ├── base
    │   └── tasks
    │       └── main.yml
    └── workstations
        └── tasks
            └── main.yml
```

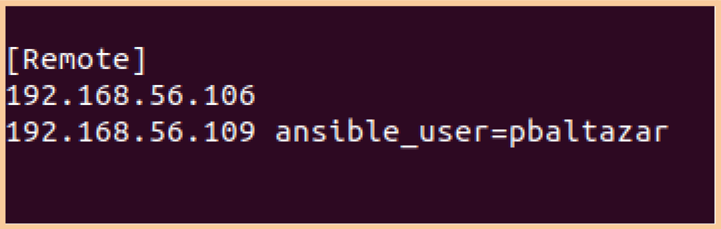
Directories



```
  GNU nano 2.9.3                    ansible.cfg

[defaults]
inventory = inventory
remote_user = paul_eimar
host_key_checking = True
```

Ansible.cfg

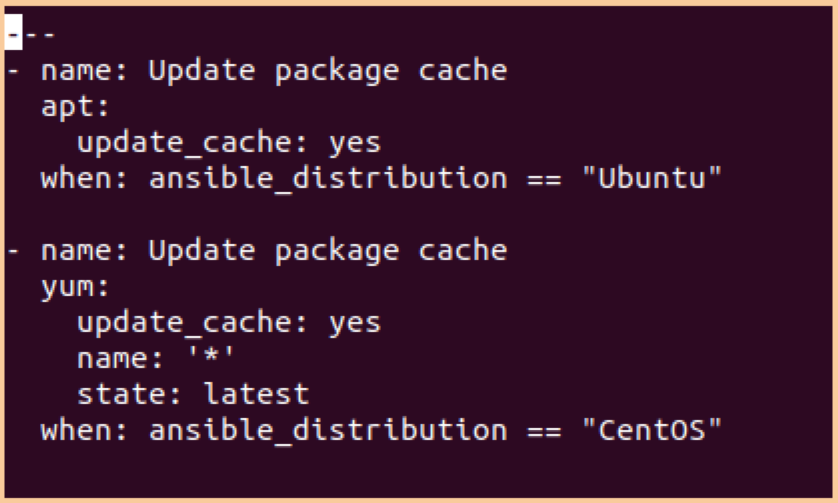
Your ansible configuration file should contain the following. This allows the managed server to locate your inventory file and also the name of the remote_user.



```
[Remote]
192.168.56.106
192.168.56.109 ansible_user=pbaltazar
```

Inventory file

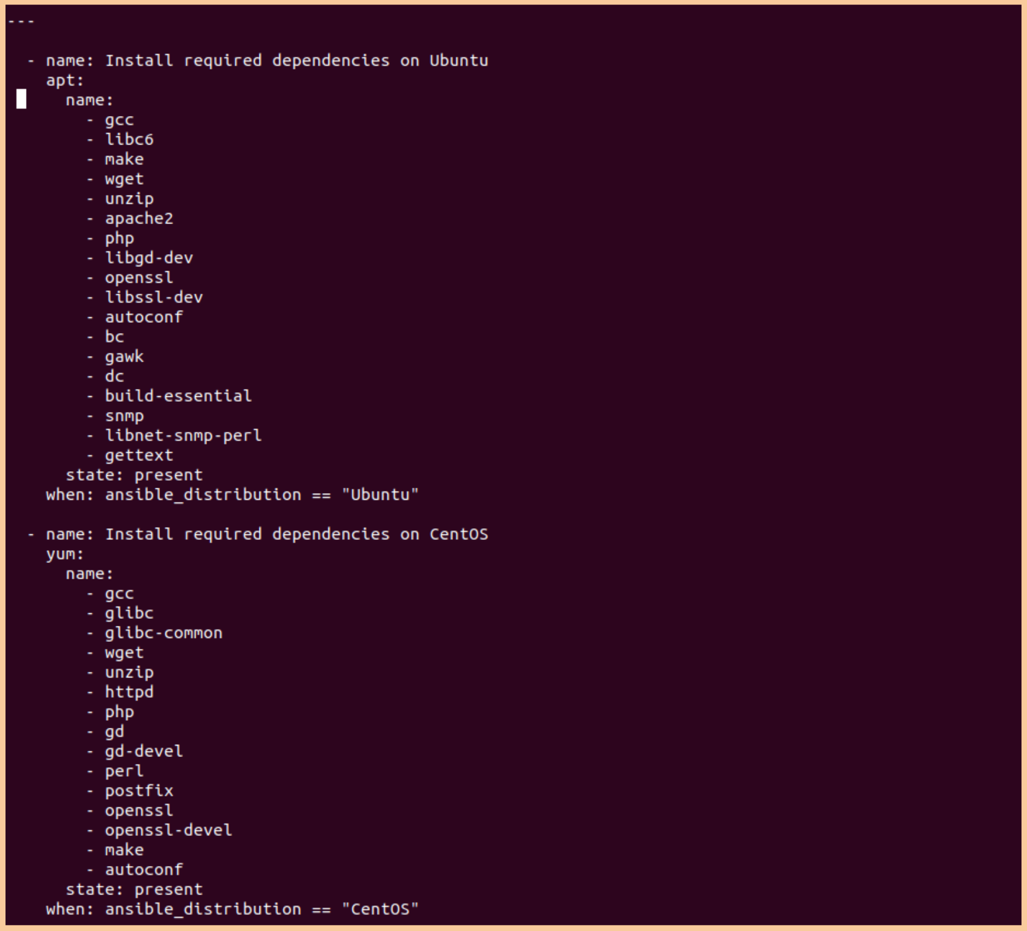
The Inventory file should contain the IP address of the system/s that you would be installing the nagios into. The IP address may vary depending on your network adapter settings.



```
---
- name: Update package cache
  apt:
    update_cache: yes
  when: ansible_distribution == "Ubuntu"

- name: Update package cache
  yum:
    update_cache: yes
    name: '*'
    state: latest
  when: ansible_distribution == "CentOS"
```

main.yml (base/task)

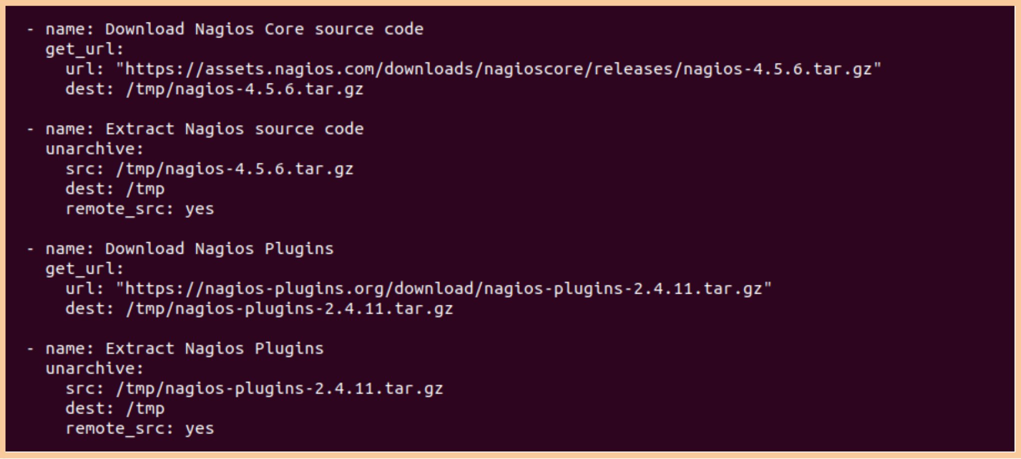
This is the yml file for the base roles. The base task covers all the remote servers, regardless of their group. This acts like the (hosts: all) line in a playbook. In the base task, we are going to update the package cache of our Ubuntu and CentOS servers.



```
---

  - 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"
```

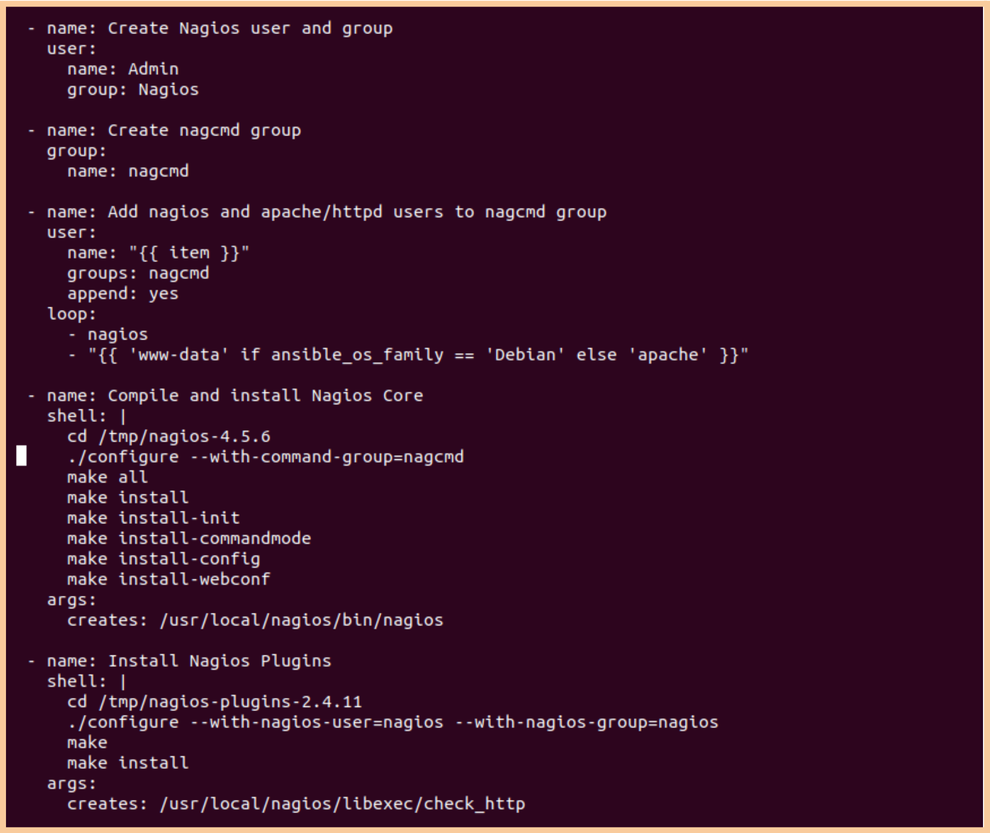
main.yml (nagios/task)

The command above installs the prerequisites of nagios. These dependencies are required so that nagios will run properly on your system.



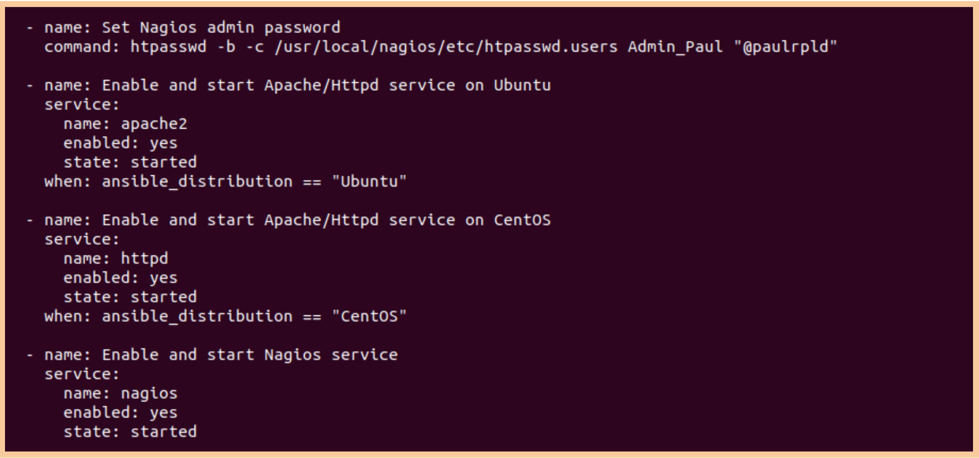
```
- name: Download Nagios Core source code
  get_url:
    url: "https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.6.tar.gz"
    dest: /tmp/nagios-4.5.6.tar.gz

- 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
```

These are the commands to download Nagios' source code and also extract it. This allows the system to have the required files for Nagios' installation.



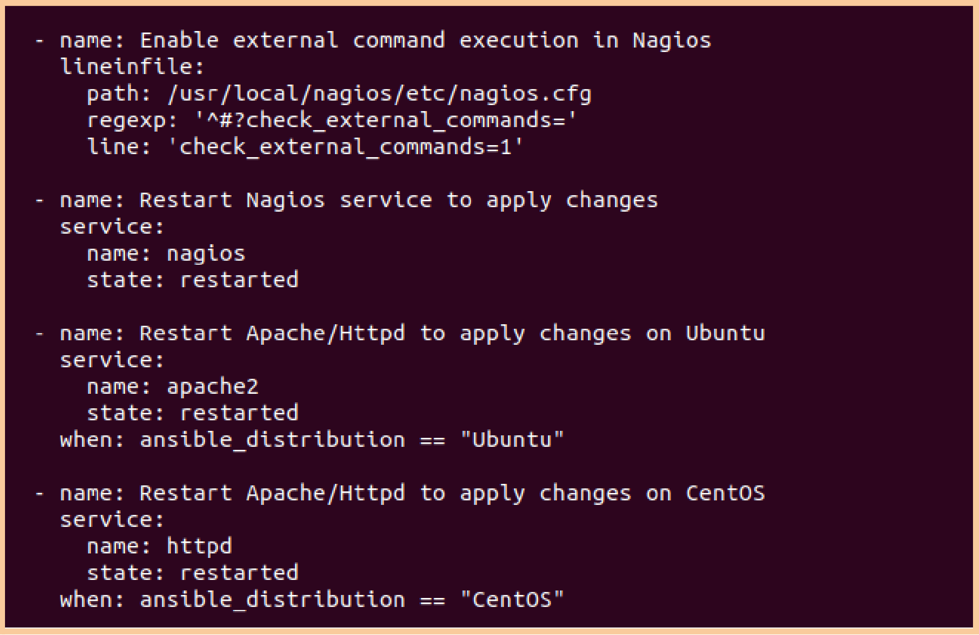
```
- name: Create Nagios user and group
  user:
    name: Admin
    group: Nagios

- 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
```

These commands create Nagios user and group and also adds apache users to a group. It also processes the compilation and installation of the Nagios itself. This part will also install the plugins for Nagios



```
- name: Set Nagios admin password
  command: htpasswd -b -c /usr/local/nagios/etc/htpasswd.users Admin_Paul "@paulrpld"

- 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
```

This part of the playbook sets the Nagios user and password, it also enables the apache service and Nagios on both systems.

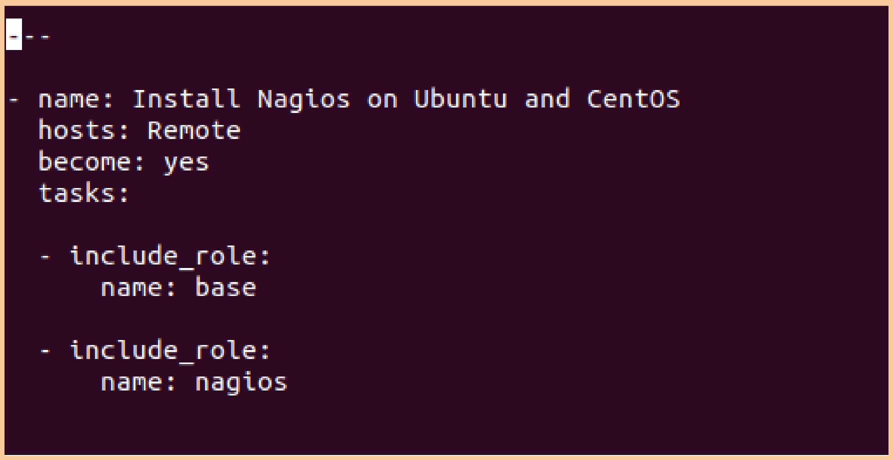


```
- name: Enable external command execution in Nagios
  lineinfile:
    path: /usr/local/nagios/etc/nagios.cfg
    regexp: '^#?check_external_commands='
    line: 'check_external_commands=1'

- 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"
```

This is the final part of the playbook. It allows the external command execution in Nagios, and also restarting the service in order to save the changes.

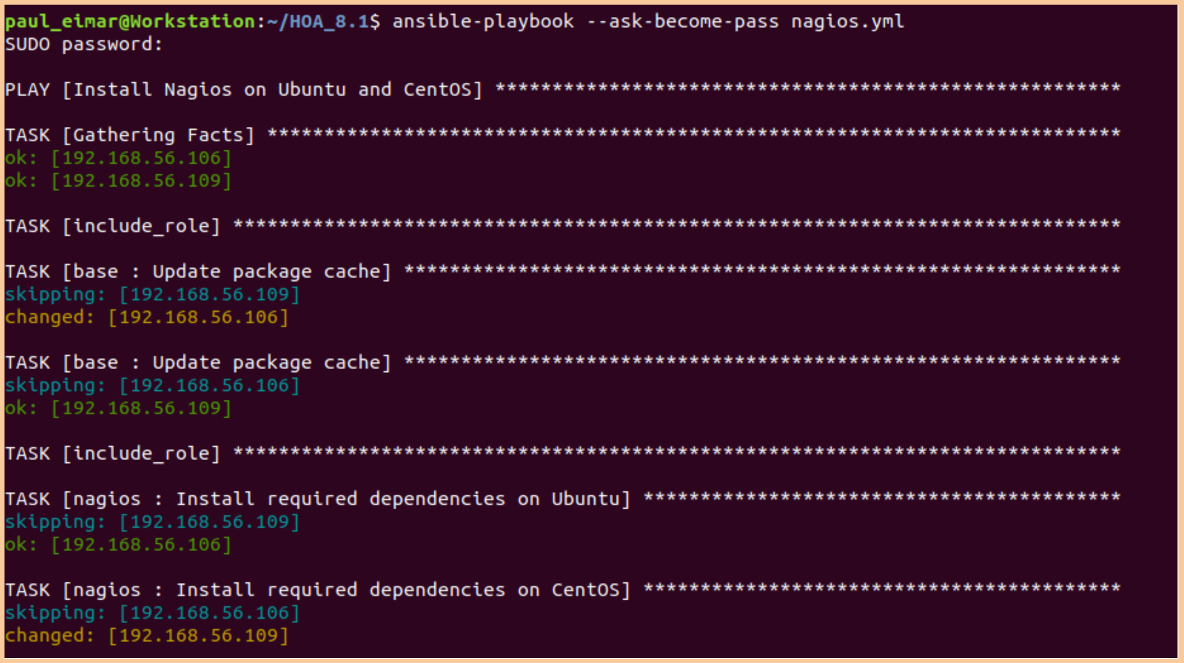
Running the playbook

In order to run the commands written in the main.yml files on both role folders, a yml file should be created on the main directory

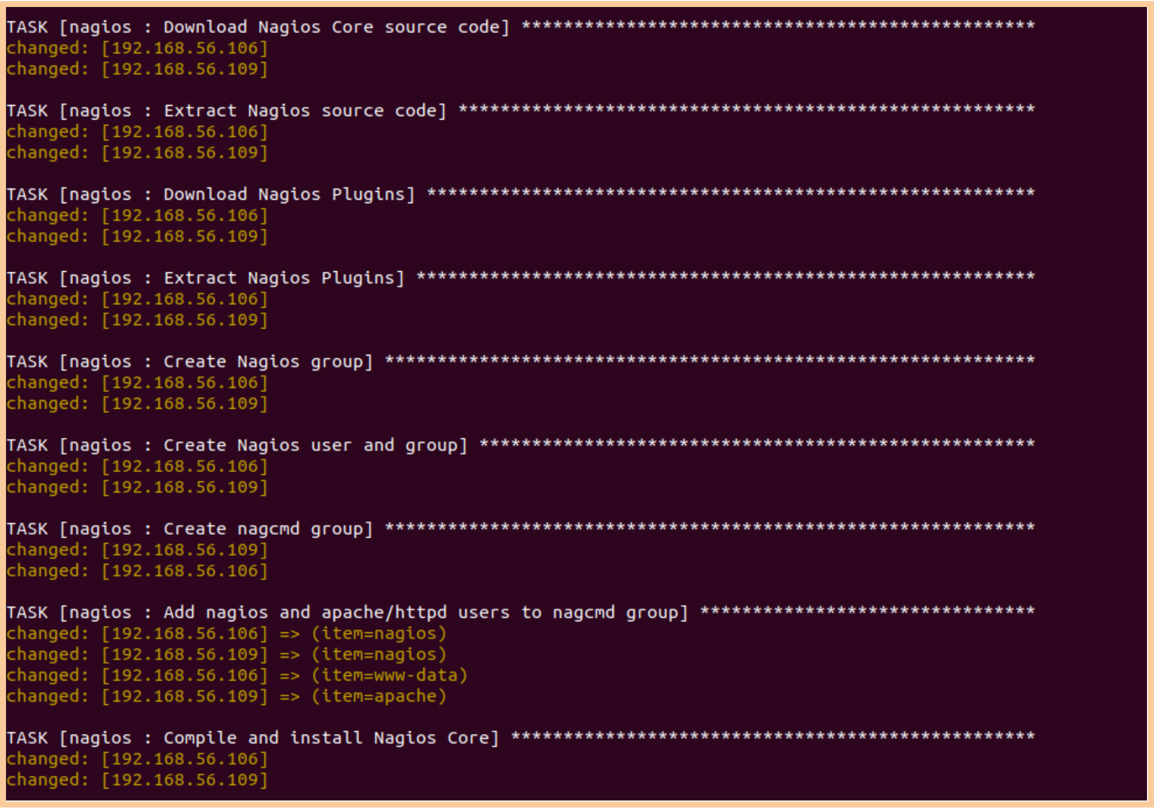


```
---

- name: Install Nagios on Ubuntu and CentOS
  hosts: Remote
  become: yes
  tasks:

  - include_role:
      name: base

  - include_role:
      name: nagios
```

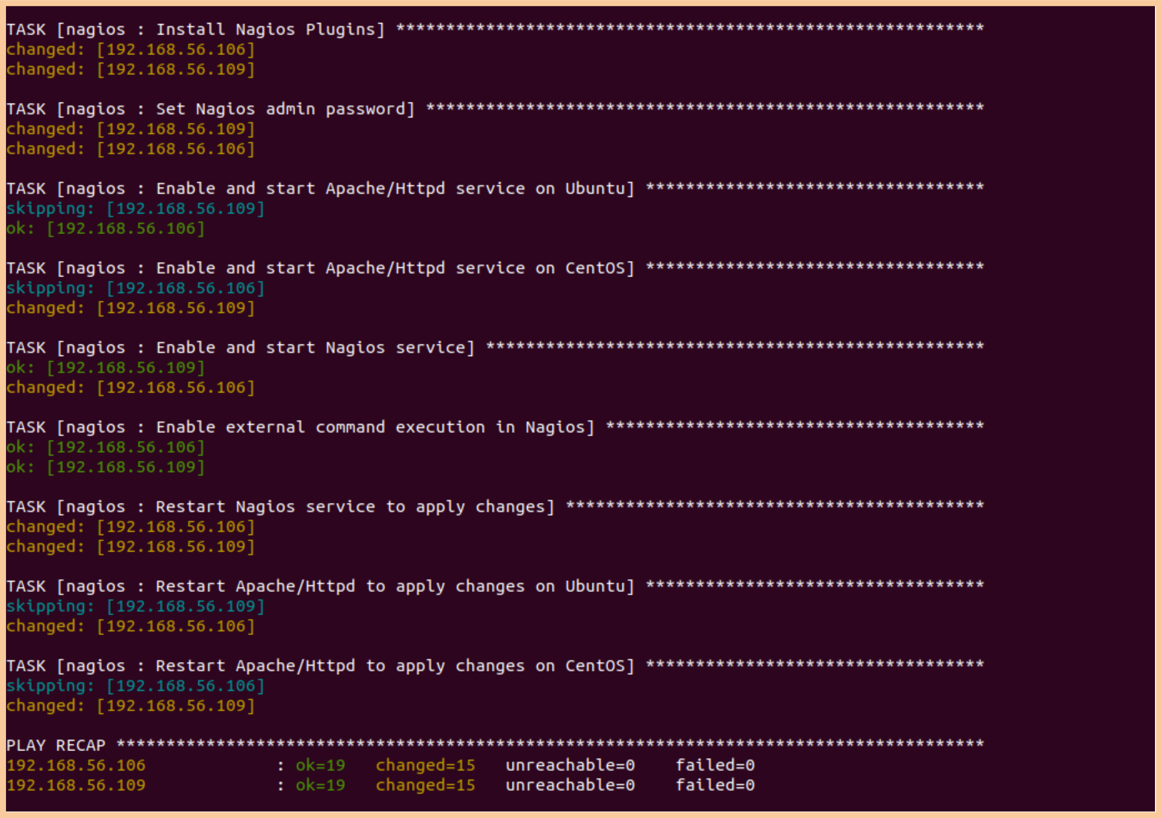
This yml file is going to execute the yml files inside the roles folder: base and nagios. To run the playbook, execute the command ***ansible-playbook –ask-become-pass [filename].yml***



```
paul_eimar@Workstation:~/HOA_8.1$ ansible-playbook --ask-become-pass nagios.yml
SUDO password:

PLAY [Install Nagios on Ubuntu and CentOS] *******************************************************

TASK [Gathering Facts] ***************************************************************************
ok: [192.168.56.106]
ok: [192.168.56.109]

TASK [include_role] ******************************************************************************

TASK [base : Update package cache] ***************************************************************
skipping: [192.168.56.109]
changed: [192.168.56.106]

TASK [base : Update package cache] ***************************************************************
skipping: [192.168.56.106]
ok: [192.168.56.109]

TASK [include_role] ******************************************************************************

TASK [nagios : Install required dependencies on Ubuntu] ******************************************
skipping: [192.168.56.109]
ok: [192.168.56.106]

TASK [nagios : Install required dependencies on CentOS] ******************************************
skipping: [192.168.56.106]
changed: [192.168.56.109]
```

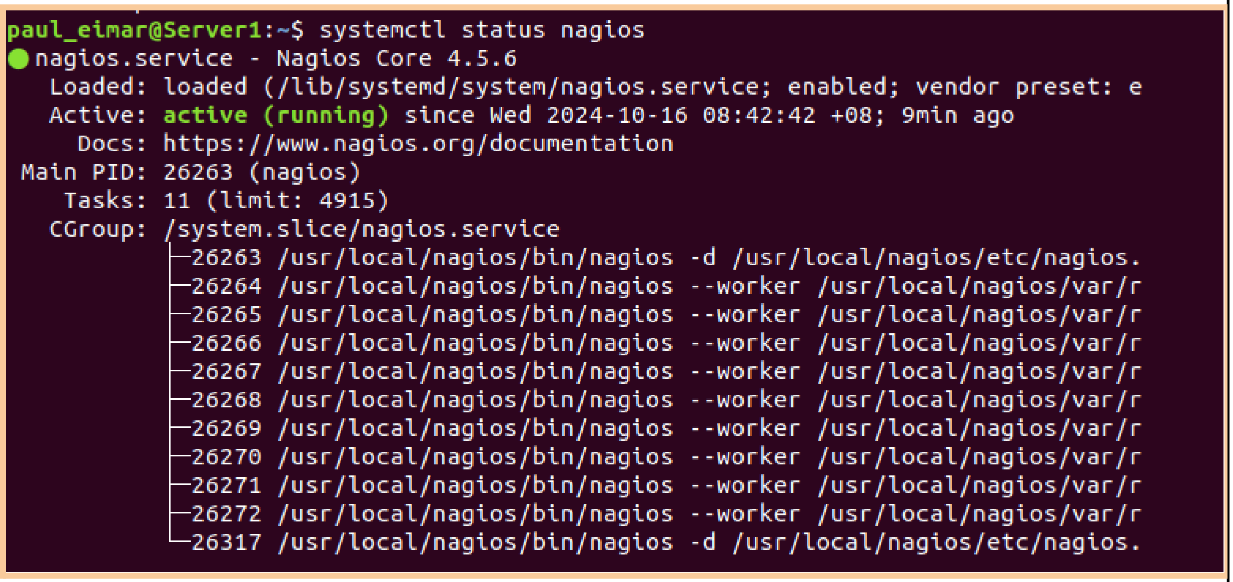


```
TASK [nagios : Download Nagios Core source code] *************************************************
changed: [192.168.56.106]
changed: [192.168.56.109]

TASK [nagios : Extract Nagios source code] *******************************************************
changed: [192.168.56.106]
changed: [192.168.56.109]

TASK [nagios : Download Nagios Plugins] **********************************************************
changed: [192.168.56.106]
changed: [192.168.56.109]

TASK [nagios : Extract Nagios Plugins] ***********************************************************
changed: [192.168.56.106]
changed: [192.168.56.109]

TASK [nagios : Create Nagios group] **************************************************************
changed: [192.168.56.106]
changed: [192.168.56.109]

TASK [nagios : Create Nagios user and group] *****************************************************
changed: [192.168.56.106]
changed: [192.168.56.109]

TASK [nagios : Create nagcmd group] **************************************************************
changed: [192.168.56.109]
changed: [192.168.56.106]

TASK [nagios : Add nagios and apache/httpd users to nagcmd group] ********************************
changed: [192.168.56.106] => (item=nagios)
changed: [192.168.56.109] => (item=nagios)
changed: [192.168.56.106] => (item=www-data)
changed: [192.168.56.109] => (item=apache)

TASK [nagios : Compile and install Nagios Core] **************************************************
changed: [192.168.56.106]
changed: [192.168.56.109]
```



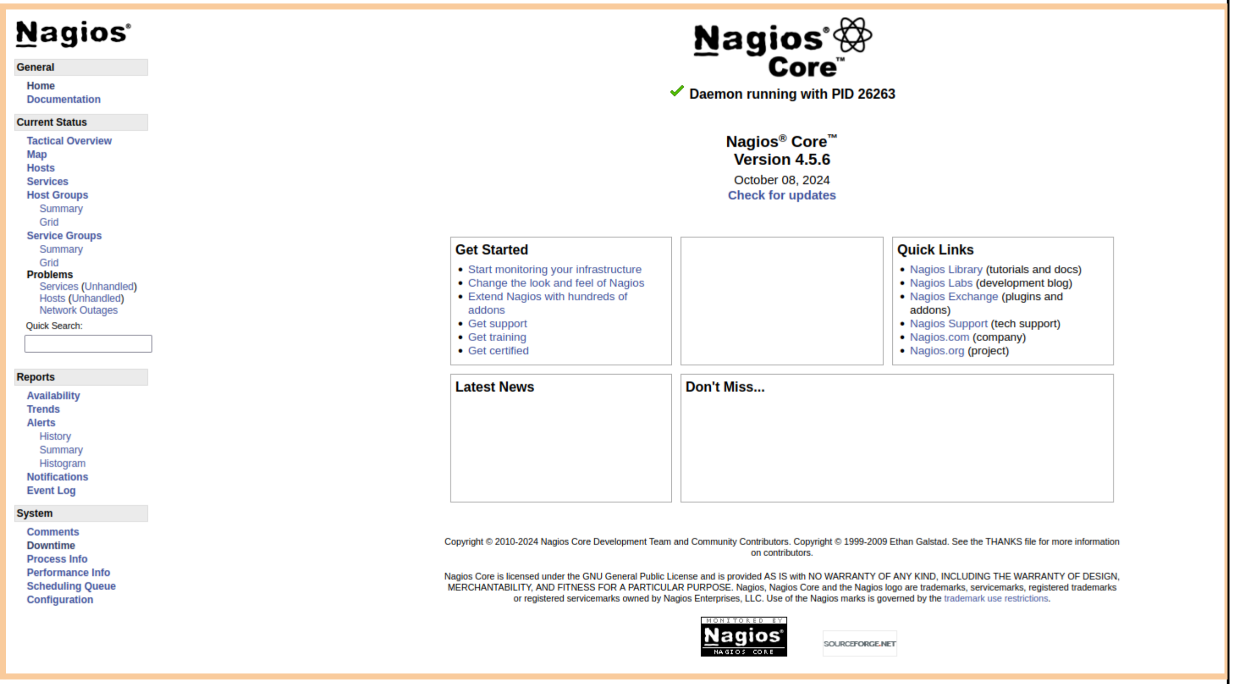
```
TASK [nagios : Install Nagios Plugins] ***********************************************************
changed: [192.168.56.106]
changed: [192.168.56.109]

TASK [nagios : Set Nagios admin password] ********************************************************
changed: [192.168.56.109]
changed: [192.168.56.106]

TASK [nagios : Enable and start Apache/Httpd service on Ubuntu] **********************************
skipping: [192.168.56.109]
ok: [192.168.56.106]

TASK [nagios : Enable and start Apache/Httpd service on CentOS] **********************************
skipping: [192.168.56.106]
changed: [192.168.56.109]

TASK [nagios : Enable and start Nagios service] **************************************************
ok: [192.168.56.109]
changed: [192.168.56.106]

TASK [nagios : Enable external command execution in Nagios] **************************************
ok: [192.168.56.106]
ok: [192.168.56.109]

TASK [nagios : Restart Nagios service to apply changes] ******************************************
changed: [192.168.56.106]
changed: [192.168.56.109]

TASK [nagios : Restart Apache/Httpd to apply changes on Ubuntu] **********************************
skipping: [192.168.56.109]
changed: [192.168.56.106]

TASK [nagios : Restart Apache/Httpd to apply changes on CentOS] **********************************
skipping: [192.168.56.106]
changed: [192.168.56.109]

PLAY RECAP ***************************************************************************************
192.168.56.106             : ok=19    changed=15   unreachable=0    failed=0
192.168.56.109             : ok=19    changed=15   unreachable=0    failed=0
```

This is how your terminal should look after executing the command. In order to check if Nagios is installed and running, go to your remote servers and execute the command ***systemctl service nagios***



```
paul_eimar@Server1:~$ systemctl status nagios
● nagios.service - Nagios Core 4.5.6
   Loaded: loaded (/lib/systemd/system/nagios.service; enabled; vendor preset: e
   Active: active (running) since Wed 2024-10-16 08:42:42 +08; 9min ago
     Docs: https://www.nagios.org/documentation
 Main PID: 26263 (nagios)
    Tasks: 11 (limit: 4915)
   CGroup: /system.slice/nagios.service
           ├─26263 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.
           ├─26264 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/r
           ├─26265 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/r
           ├─26266 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/r
           ├─26267 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/r
           ├─26268 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/r
           ├─26269 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/r
           ├─26270 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/r
           ├─26271 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/r
           ├─26272 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/r
           └─26317 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.
```

You can also check using the web browser if it is running by going to ***localhost/nagios/***



✓ Daemon running with PID 26263

Nagios® Core™
Version 4.5.6
October 08, 2024
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

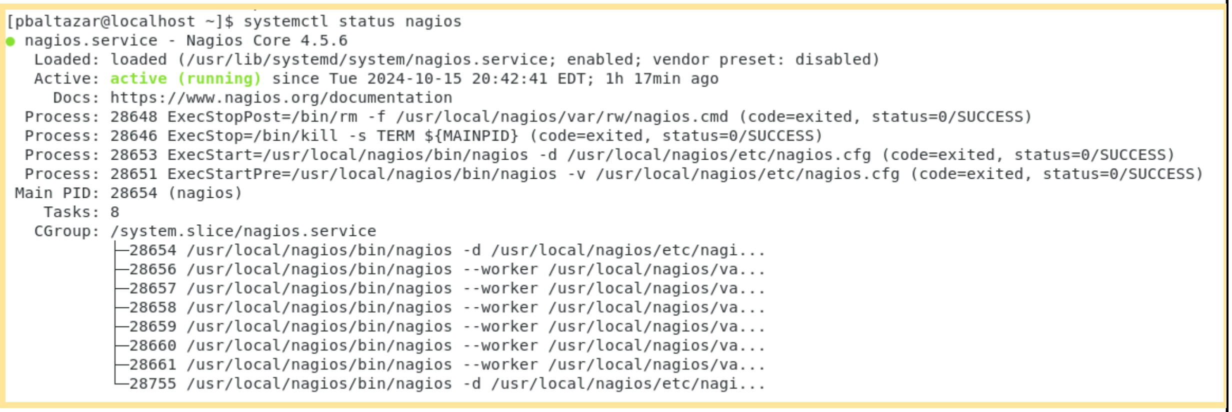
Quick Links

- Nagios Library (tutorials and docs)
- Nagios Labs (development blog)
- Nagios Exchange (plugins and addons)
- Nagios Support (tech support)
- Nagios.com (company)
- Nagios.org (project)

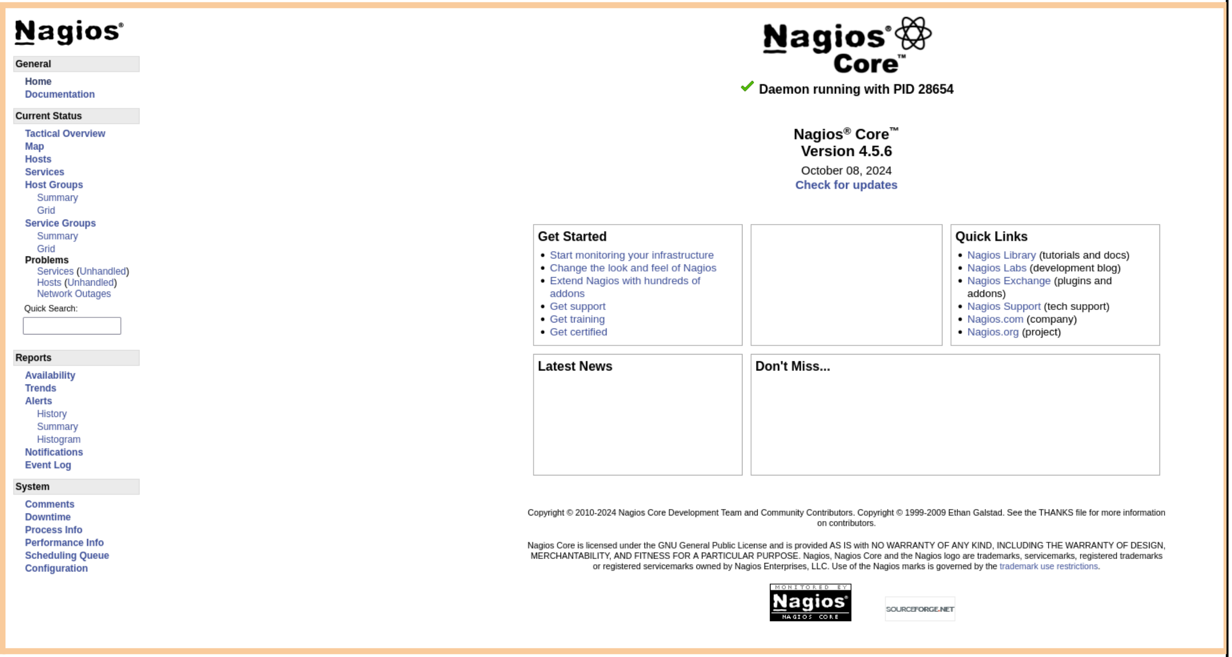
Latest News

Don't Miss...

The following can be done on CentOS servers



```
[pbaltazar@localhost ~]$ systemctl status nagios
● nagios.service - Nagios Core 4.5.6
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2024-10-15 20:42:41 EDT; 1h 17min ago
     Docs: https://www.nagios.org/documentation
  Process: 28648 ExecStopPost=/bin/rm -f /usr/local/nagios/var/rw/nagios.cmd (code=exited, status=0/SUCCESS)
  Process: 28646 ExecStop=/bin/kill -s TERM ${MAINPID} (code=exited, status=0/SUCCESS)
  Process: 28653 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
  Process: 28651 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 28654 (nagios)
    Tasks: 8
   CGroup: /system.slice/nagios.service
           ├─28654 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagi...
           ├─28656 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/va...
           ├─28657 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/va...
           ├─28658 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/va...
           ├─28659 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/va...
           ├─28660 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/va...
           ├─28661 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/va...
           └─28755 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagi...
```



✓ Daemon running with PID 28654

Nagios® Core™
Version 4.5.6
October 08, 2024
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 Library (tutorials and docs)
- Nagios Labs (development blog)
- Nagios Exchange (plugins and addons)
- Nagios Support (tech support)
- Nagios.com (company)
- Nagios.org (project)
Latest News
Don't Miss...

Copyright © 2010-2024 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Nagios Core is licensed under the GNU General Public License and is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Nagios, Nagios Core and the Nagios logo are trademarks, servicemarks, registered trademarks or registered servicemarks owned by Nagios Enterprises, LLC. Use of the Nagios marks is governed by the trademark use restrictions.

Reflections:

Answer the following:

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

   – Availability monitoring tools like Nagios, offer a lot of benefits for organizations that need their IT systems to stay up and running smoothly. One of the biggest advantages is the ability to catch problems early. These tools provide real-time alerts when something goes wrong, like when a server goes down or a network issue. It can help you fix it quickly and avoid costly downtime. They also help improve system performance by keeping track of how resources like CPU, memory, and disk space are being used, which makes it easier to balance the load and keep things running efficiently. Another benefit is making your system more reliable. By monitoring critical services and applications, these tools help ensure that everything is running as it should. They also store historical data, so you can look back and find patterns that might help prevent recurring problems. With tools like Nagios, you can monitor your entire network, from servers to apps, all from one place with a customizable dashboard.

Conclusions:

– In conclusion, by learning how to create and design a workflow that installs, configures, and manages enterprise monitoring tools using Ansible as an Infrastructure as Code (IaC) tool, I will be able to simplify the process of setting up and managing monitoring systems. This skill will help me automate tasks, keep everything consistent, and scale the system easily as the organization grows. Using Ansible will also reduce mistakes that come with manual work and allow me to build a reliable monitoring setup that gives important insights into system performance, helping keep IT operations running smoothly.