

Name: Abigail Laxamana	Date Performed: Oct 9, 2023
Course/Section: CPE232 - CPE31S6	Date Submitted: Oct 9, 2023
Instructor: Dr. Jonathan Taylar	Semester and SY: 1st sem, SY: 2022-2023ho
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	
<b>1. Create a playbook that installs Nagios in both Ubuntu and CentOS. Apply the concept of creating roles.</b>	
1.1 Create an ansible file and an inventory which includes the ip addresses of the servers you plan to make changes with.	

**ansible.cfg** file

```
[defaults]

inventory = inventory
host_key_checking = False

deprecation_warnings = False

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

**inventory** file

```
[ubuntu_nagios]
192.168.56.103

[centos_nagios]
Laxamana@192.168.56.110
```

1.2 Inside the new repository, I created a tree where the first branch is called **roles** followed by creating another directories for ubuntu and centos. Then, inside the two directories, I created another directory called **tasks** where it contains a file called **main.yml**.

**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.)**

**HOA8\_Laxamana/roles/centos\_nagios/tasks/main.yml**

```
- name: Installing nagios dependencies and libraries
tags: dependencies, libraries
```

```
yum:
```

```
  name:
```

```
    - gcc
    - glibc
    - glibc-common
    - perl
    - httpd
    - php
    - wget
    - gd
    - gd-devel
    - openssl-devel
    - gcc
    - glibc
    - glibc-common
    - make
    - gettext
    - automake
    - autoconf
    - wget
    - openssl-devel
    - net-snmp
    - net-snmp-utils
```

```
    - net-snmp-utils
    - python2-pip
state: latest
```

Using the yum module, this process installs a variety of packages and dependencies necessary for nagios to function, including web servers (httpd), libraries, compilers (gcc), and more. The packages are up-to-date because of the **state: latest** option.

```
- name: Install passlib python package
```

```
  pip:
```

```
    name: passlib
```

```
- name: Creating a directory (where the downloaded files will be stored)
```

```
  file:
```

```
    path: ~/nagios
```

```
    state: directory
```

```
- name: Downloading and extracting Nagios
```

```
  unarchive:
```

```
    src: https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.6.t
```

```
ar.gz
```

```
    dest: ~/nagios
```

```
    remote_src: yes
```

```
    mode: 0777
```

```
    owner: root
```

```
    group: root
```

In this task, a library often used for password hashing and validation called **passlib** was installed using the pip module. Then, I

created a path for nagios which is stated as a directory using the file module. Also, this task downloaded and extracted nagios from a source from github, to a destination which is the directory path created earlier. The extracted files' permissions, ownership, and group are even established.

```
- name: Compiling, installing, and adding users and groups in nagios
  shell: |
    cd ~/nagios/nagioscore-**
    ./configure
    make all
```

```
    make install-groups-users
    usermod -a -G nagios apache
    make install
    make install-daemoninit
    make install-commandmode
    make install-config
    make install-webconf
- name: Downloading and extracting Nagios plugins
  unarchive:
    src: https://github.com/nagios-plugins/nagios-plugins/archive/release-2.3.3
    .tar.gz
    dest: ~/nagios
    remote_src: yes
    mode: 0777
    owner: root
    group: root

- name: Compiling and installing plugins
  shell: |
    cd ~/nagios/nagios-plugins*
    ./tools/setup
    ./configure
    make
    make install

- name: Add a user to a password file and ensure permissions are set
  community.general.htpasswd:
    path: /usr/local/nagios/etc/htpasswd.users
```

```

make install-groups-users
usermod -a -G nagios apache
make install
make install-daemoninit
make install-commandmode
make install-config
make install-webconf
- name: Downloading and extracting Nagios plugins
  unarchive:
    src: https://github.com/nagios-plugins/nagios-plugins/archive/release-2.3.3
    tar.gz
    dest: ~/nagios
    remote_src: yes
    mode: 0777
    owner: root
    group: root

- name: Compiling and installing plugins
  shell: |
    cd ~/nagios/nagios-plugins*
    ./tools/setup
    ./configure
    make
    make install
- name: Add a user to a password file and ensure permissions are set
  community.general.htpasswd:
    path: /usr/local/nagios/etc/htpasswd.users

```

```

name: admin
password: admin

```

This task configures nagios as it does install user groups, compile nagios, establish apache, and navigate to Nagios source directory. While, the other one downloads and extracts nagios plugins to the “nagios” directory from a different url. In compiling and installing plugins, nagios plugins source directory is navigated, which then builds and installs the plugins. In this part, community.general.htpasswd has been utilized to add a user named “admin” and the password “admin” to an htpasswd file which is required in nagios authentication. It also guarantees appropriate file permissions.

```

- name: Making sure that nagios is started and enabled
  service:
    name: nagios
    state: restarted
    enabled: true

- name: Making sure that httpd is started and enabled
  service:
    name: httpd
    state: restarted
    enabled: true
laxamana_ubuntu@workstation:~/H0A8_Laxamana/roles/centos_nagios/tasks$ █

```

Using the services module, nagios service will be restarted and enabled by this task.

## HOA8\_Laxamana/roles/ubuntu\_nagios/tasks/main.yml

```
laxamana_ubuntu@workstation:~/HOA8_Laxamana/roles/ubuntu_nagios/tasks$ cat main
.yml
---
- name: nagios libraries and dependencies (Ubuntu)
  tags: ubuntu, dependencies, libraries
  apt:
    name:
      - autoconf
      - libc6
      - gcc
      - make
      - wget
      - unzip
      - apache2
      - php
      - libapache2-mod-php7.2
      - libgd-dev
      - openssl
      - libssl-dev
      - bc
      - gawk
      - dc
      - build-essential
      - snmp
      - libnet-snmp-perl
      - gettext

      - python3
      - python3-pip
  state: latest
```

This task is to install the nagios libraries and dependencies for ubuntu. This will install the latest libraries and dependencies of nagios since the state is indicated as latest. This will also use the apt (advanced packaging tool) to install those packages.

```
- name: passlib package
  pip:
    name: passlib

- name: nagios directory PATH
  file:
    path: ~/nagios
    state: directory

- name: downloading nagios
  unarchive:
    src: https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.6.tar.gz
    dest: ~/nagios
    remote_src: yes
    mode: 0777
    owner: root
    group: root
```

The task “passlib package” will install the passlib package using the pip tool. The task “nagios directory PATH” is to add a new directory named “nagios” to the home directory. The third task will unarchive the src path, since the source path is a url of github we

declare that the `remote_src` is yes, meaning it should download the file first before unarchiving. After unarchiving it in the `dst` path, it will make the permission of the file to 777 and set the owner and group to root.

```
- name: downloading nagios plugins
  unarchive:
    src: https://github.com/nagios-plugins/nagios-plugins/archive/release-2.3.3
    tar.gz
```

```
    dest: ~/nagios
    remote_src: yes
    mode: 0777
    owner: root
    group: root

- name: install, compile, adding users and groups
  shell: |
    cd ~/nagios/nagioscore-*
    sudo ./configure --with-httpd-conf=/etc/apache2/sites-enabled
    sudo make all
    sudo make install-groups-users
    sudo usermod -a -G nagios www-data
    sudo make install
    sudo make install-daemoninit
    sudo make install-commandmode
    sudo make install-config
    sudo make install-webconf
    sudo a2enmod rewrite
    sudo a2enmod cgi
```

The task “downloading nagios plugins” is the same process that happened on the task “downloading nagios” but with a different src url. The task “install, compile, adding users and group” is where it will install all the nagios from `nagioscore-*` path, this task will follow the given linux/shell command.

```
- name: compile and install plugins
  shell: |
    cd ~/nagios/nagios-plugins*
    ./tools/setup
    ./configure
    make
```

```
    make install

- name: adding users to nagios
  community.general.htpasswd:
    path: /usr/local/nagios/etc/htpasswd.users
    name: admin
    password: admin
```

The task “compile and install plugins” this is also same with the last task “install, compile, adding users and group”, but this time to install the plugins of nagios and will also follow the given linux/shell command. The task “adding users to nagios” is to add admin to users and nagios

```
- name: Nagios Start/Enable Check
  service:
    name: nagios
    state: restarted
    enabled: true

- name: Apache/httpd Start/Enable check
  service:
    name: apache2
    state: restarted
    enabled: true
laxamana_ubuntu@workstation:~/HOA8_Laxamana/roles/ubuntu_nagios/tasks$ top: //mic
```

These both task is to enable the nagios and the apache2.

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)

##### abbyhoa8.yml

```
laxamana_ubuntu@workstation:~/HOA8_Laxamana$ cat abbyhoa8.yml
---
- hosts: all
  become: true
  pre_tasks:

    - name: dnf and epel installation
      yum:
        name:
          - epel-release
          - dnf
      when: ansible_distribution == "CentOS"

    - name: dpkg in ubuntu
      shell: |
        dpkg --configure -a
      when: ansible_distribution == "Ubuntu"

    - name: install updates (CentOS)
      dnf:
        update_cache: yes
        update_only: yes
      when: ansible_distribution == "CentOS"
```



```

- name: install updates (Ubuntu)
  apt:
    upgrade: dist
    update_cache: yes
    when: ansible_distribution == "Ubuntu"

- hosts: ubuntu_nagios
  become: true
  roles:
    - ubuntu_nagios

- hosts: centOS_nagios
  become: true
  roles:
    - centos_nagios
laxamana_ubuntu@workstation:~/HOA8_Laxamana$

```

This is the playbook used to run the roles and the pre-tasks. The pre-tasks contain commands to install packages, and updates. After the pre-tasks, hosts such as **ubuntu\_nagios** and **centOS nagios** are set to run. Meaning, after running the playbook's pre-tasks, the playbook for ubuntu inside a role will play, followed by the playbook for centos inside a role as well.

## Playbook process

```

laxamana_ubuntu@workstation:~/HOA8_Laxamana$ ansible-playbook --ask-become-pass
abbyhoa8.yml
BECOME password:

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.103]
ok: [Laxamana@192.168.56.110]

TASK [dnf and epel installation] *****
*
skipping: [192.168.56.103]
ok: [Laxamana@192.168.56.110]

TASK [dpkg in ubuntu] *****
*
skipping: [Laxamana@192.168.56.110]
changed: [192.168.56.103]

TASK [install updates (CentOS)] *****
*
skipping: [192.168.56.103]
ok: [Laxamana@192.168.56.110]

```

```
TASK [install updates (Ubuntu)] *****
*
skipping: [Laxamana@192.168.56.110]
ok: [192.168.56.103]

PLAY [ubuntu_nagios] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.103]

TASK [ubuntu_nagios : nagios libraries and dependencies (Ubuntu)] *****
*
ok: [192.168.56.103]

TASK [ubuntu_nagios : passlib package] *****
*
ok: [192.168.56.103]

TASK [ubuntu_nagios : nagios directory PATH] *****
*
ok: [192.168.56.103]

TASK [ubuntu_nagios : downloading nagios] *****
*
ok: [192.168.56.103]
```

```
TASK [ubuntu_nagios : downloading nagios plugins] *****
*
ok: [192.168.56.103]

TASK [ubuntu_nagios : install, compile, adding users and groups] *****
*
changed: [192.168.56.103]

TASK [ubuntu_nagios : compile and install plugins] *****
*
changed: [192.168.56.103]

TASK [ubuntu_nagios : adding users to nagios] *****
*
ok: [192.168.56.103]

TASK [ubuntu_nagios : Nagios Start/Enable Check] *****
*
changed: [192.168.56.103]

TASK [ubuntu_nagios : Apache/httpd Start/Enable check] *****
*
changed: [192.168.56.103]

PLAY [centOS_nagios] *****
*
```

```
TASK [Gathering Facts] *****
*
ok: [Laxamana@192.168.56.110]

TASK [centos_nagios : Installing nagios dependencies and libraries] *****
*
changed: [Laxamana@192.168.56.110]

TASK [centos_nagios : Install passlib python package] *****
*
changed: [Laxamana@192.168.56.110]

TASK [centos_nagios : Creating a directory (where the downloaded files will be
stored)] ***
changed: [Laxamana@192.168.56.110]

TASK [centos_nagios : Downloading and extracting Nagios] *****
*
changed: [Laxamana@192.168.56.110]

TASK [centos_nagios : Compiling, installing, and adding users and groups in nag
ios] ***
changed: [Laxamana@192.168.56.110]
```

```

TASK [centos_nagios : Downloading and extracting Nagios plugins] *****
*
changed: [Laxamana@192.168.56.110]

TASK [centos_nagios : Compiling and installing plugins] *****
*
changed: [Laxamana@192.168.56.110]

TASK [centos_nagios : Add a user to a password file and ensure permissions are
set] ***
changed: [Laxamana@192.168.56.110]

TASK [centos_nagios : Making sure that nagios is started and enabled] *****
*
changed: [Laxamana@192.168.56.110]

TASK [centos_nagios : Making sure that httpd is started and enabled] *****
*
changed: [Laxamana@192.168.56.110]

PLAY RECAP *****
*
192.168.56.103      : ok=14   changed=5    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
Laxamana@192.168.56.110 : ok=14   changed=10   unreachable=0    failed=0
skipped=2    rescued=0    ignored=0

```

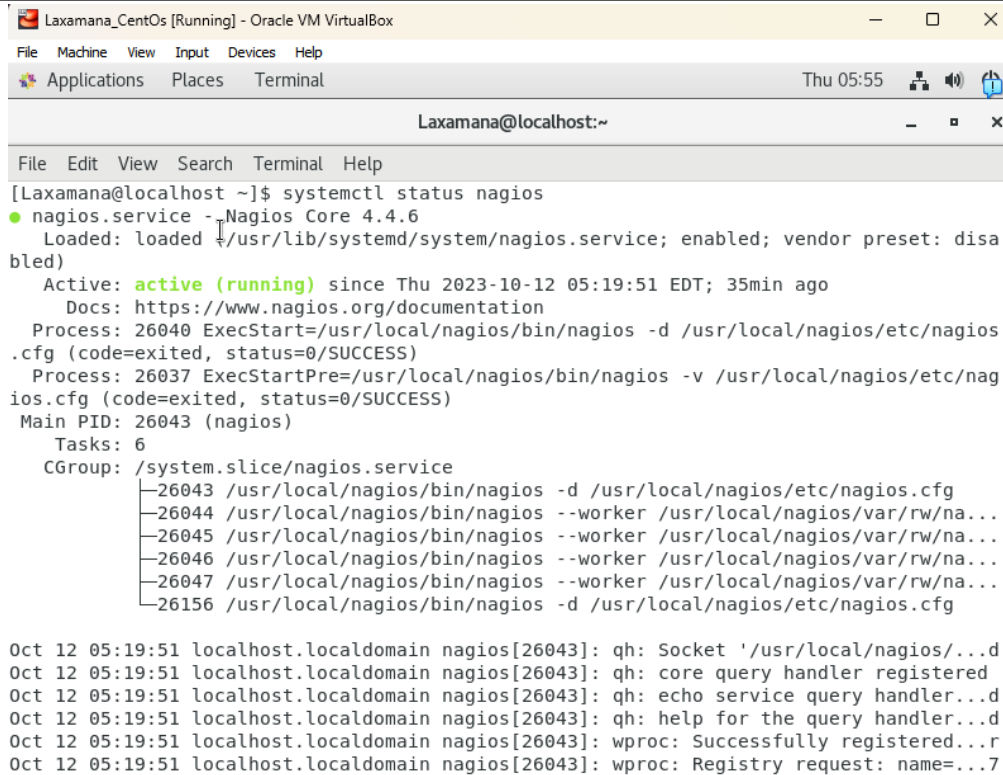
## verification

```

laxamana_ubuntu@server1:~$ systemctl status nagios
● nagios.service - Nagios Core 4.4.6
   Loaded: loaded (/lib/systemd/system/nagios.service; enabled; vendor preset:
   Active: active (running) since Thu 2023-10-12 17:17:34 PST; 36min ago
     Docs: https://www.nagios.org/documentation
   Process: 25412 ExecStopPost=/bin/rm -f /usr/local/nagios/var/rw/nagios.cmd (c
   Process: 25411 ExecStop=/bin/kill -s TERM ${MAINPID} (code=exited, status=0/S
   Process: 25414 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/et
   Process: 25413 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios
   Main PID: 25415 (nagios)
      Tasks: 6 (limit: 4656)
     CGroup: /system.slice/nagios.service
             └─25415 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios
               └─25416 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/
                 └─25417 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/
                   └─25418 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/
                     └─25419 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/
                       └─25420 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios

Oct 12 17:17:34 server1 nagios[25415]: qh: Socket '/usr/local/nagios/var/rw/nag
Oct 12 17:17:34 server1 nagios[25415]: qh: core query handler registered
Oct 12 17:17:34 server1 nagios[25415]: qh: echo service query handler registere
Oct 12 17:17:34 server1 nagios[25415]: qh: help for the query handler registere
Oct 12 17:17:34 server1 nagios[25415]: wproc: Successfully registered manager a
Oct 12 17:17:34 server1 nagios[25415]: wproc: Registry request: name=Core Worke
Oct 12 17:17:34 server1 nagios[25415]: wproc: Registry request: name=Core Worke
Oct 12 17:17:34 server1 nagios[25415]: wproc: Registry request: name=Core Worke

```



```
Laxamana_CentOs [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal Thu 05:55
Laxamana@localhost:~
File Edit View Search Terminal Help
[Laxamana@localhost ~]$ systemctl status nagios
● nagios.service - Nagios Core 4.4.6
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2023-10-12 05:19:51 EDT; 35min ago
     Docs: https://www.nagios.org/documentation
    Process: 26040 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 26037 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 26043 (nagios)
      Tasks: 6
     CGroup: /system.slice/nagios.service
             └─26043 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
               └─26044 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/na...
                 └─26045 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/na...
                   └─26046 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/na...
                     └─26047 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/na...
                       └─26156 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 12 05:19:51 localhost.localdomain nagios[26043]: qh: Socket '/usr/local/nagios/...d
Oct 12 05:19:51 localhost.localdomain nagios[26043]: qh: core query handler registered
Oct 12 05:19:51 localhost.localdomain nagios[26043]: qh: echo service query handler...d
Oct 12 05:19:51 localhost.localdomain nagios[26043]: qh: help for the query handler...d
Oct 12 05:19:51 localhost.localdomain nagios[26043]: wproc: Successfully registered...r
Oct 12 05:19:51 localhost.localdomain nagios[26043]: wproc: Registry request: name=...7
```

## Reflections:

### Answer the following:

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

Tools for availability monitoring provide several benefits for organizations and IT operations. These solutions offer early issue detection, enabling businesses to spot issues and fix them before they result in service interruptions or outages. A major advantage is proactive problem solving, which enables IT teams to react quickly and efficiently to reduce downtime and its effects on users. These tools help to improve user experience, increasing customer happiness and productivity by maintaining service uptime. They also result in less downtime, which is important for sectors where service disruptions cost money and harm reputations. By effectively leveraging infrastructure resources, availability monitoring solutions promote efficient resource allocation, capacity planning, and cost reductions. They are essential for monitoring security, reporting, and compliance. Organizations may decide on system improvements based on historical data and trend analysis, and alarms, notifications, and redundancy testing enable prompt resolution of problems and ongoing commercial

operations. These technologies make it easier for enterprises to make data-driven decisions, enabling them to operate with confidence and meet their performance and reliability objectives.

### **Conclusions:**

I therefore conclude that installing, managing, and configuring something from a server to another serves, even when servers support different packages, it is better to use playbooks as tasks can be simplified and specified in the ansible playbook file. Through the use of playbooks, we can also monitor better if there are changes in the servers when a playbook was run. As we can organize and assign roles within a playbook, debugging was made easier since we can point from a broad set of ideas and tasks to a specific one who's having an error.

### **github repository link:**

[https://github.com/Abigaiiii/HOA8\\_Laxamana.git](https://github.com/Abigaiiii/HOA8_Laxamana.git)