Name: Dalena, Justin Miguel S.	**Date Performed: Oct. 18, 2024**
Course/Section: CPE31S21	**Date Submitted: Oct. 18, 2024**
Instructor: Engr. Robin Valenzuela	**Semester and SY: 1st sem 2024-25**

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

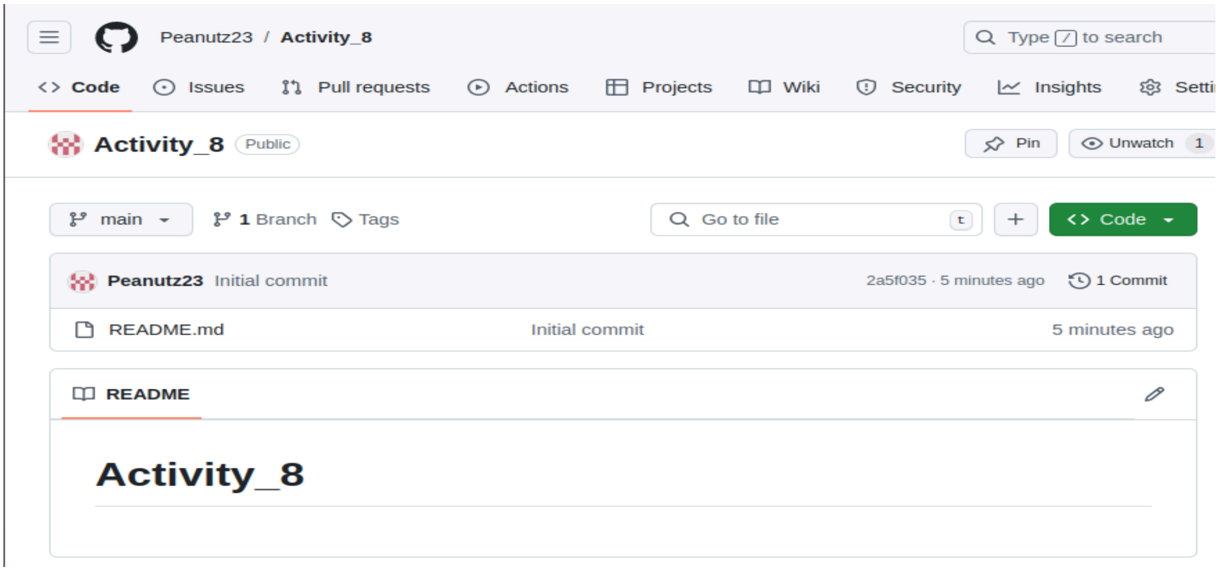
Availability monitoring is a type of monitoring tool that we use if the certain workload is up or reachable on our end. Site downtime can lead to loss of revenue, reputational damage and severe distress. Availability monitoring prevents adverse situations by checking the uptime of infrastructure components 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)

Create a New github Repository

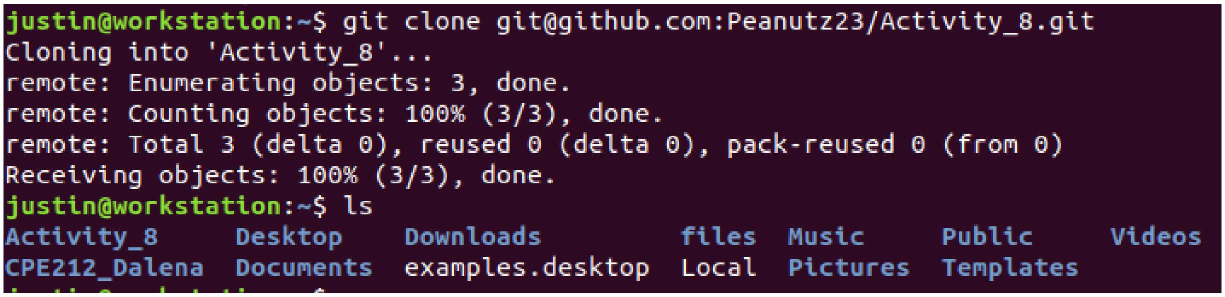


Peanutz23 Initial commit 2a5f035 · 5 minutes ago

README.md Initial commit 5 minutes ago

Activity_8

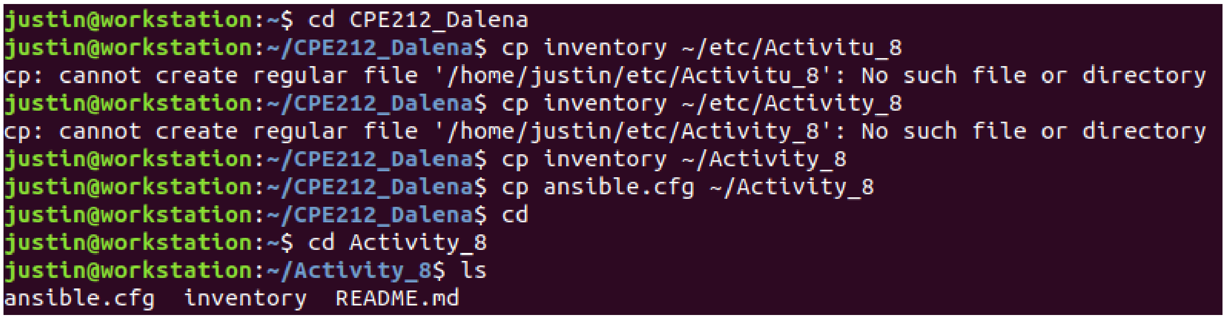
Clone the repository to your workstation



```
justin@workstation:~$ git clone git@github.com:Peanutz23/Activity_8.git
Cloning into 'Activity_8'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
justin@workstation:~$ ls
Activity_8     Desktop    Downloads         files  Music     Public     Videos
CPE212_Dalena  Documents  examples.desktop  Local  Pictures  Templates
```

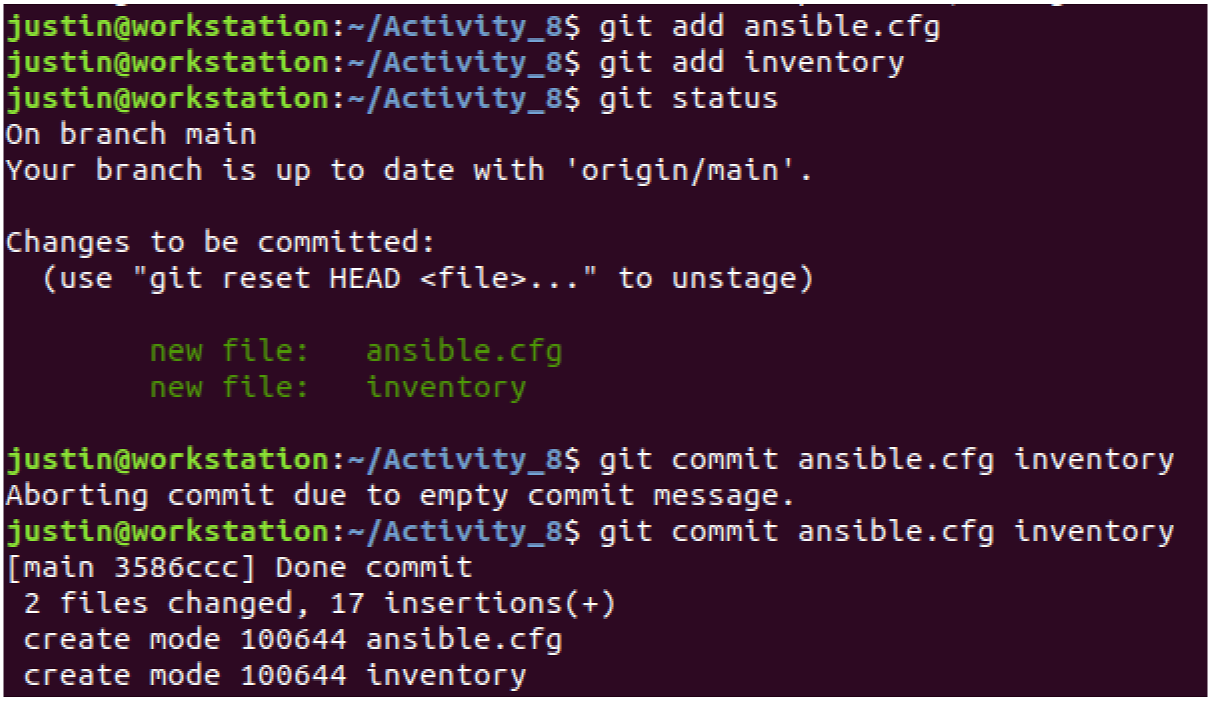
In the copied repository create a Playbook and copy the ansible and inventory from the previous repository.

Copying of Inventory and Ansible.cfg from the other repository



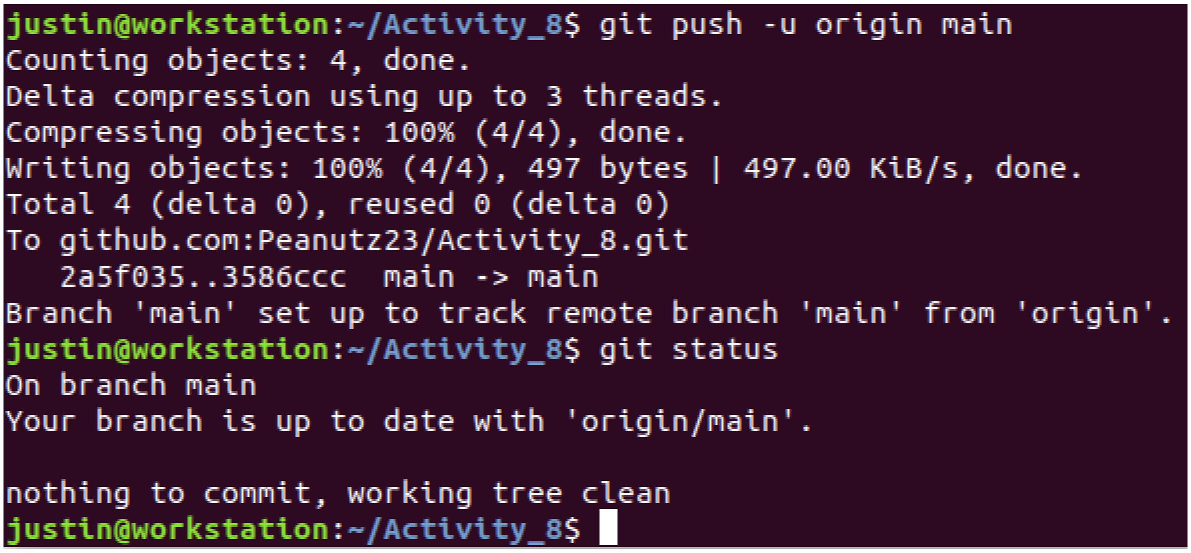
```
justin@workstation:~$ cd CPE212_Dalena
justin@workstation:~/CPE212_Dalena$ cp inventory ~/etc/Activitu_8
cp: cannot create regular file '/home/justin/etc/Activitu_8': No such file or directory
justin@workstation:~/CPE212_Dalena$ cp inventory ~/etc/Activity_8
cp: cannot create regular file '/home/justin/etc/Activity_8': No such file or directory
justin@workstation:~/CPE212_Dalena$ cp inventory ~/Activity_8
justin@workstation:~/CPE212_Dalena$ cp ansible.cfg ~/Activity_8
justin@workstation:~/CPE212_Dalena$ cd
justin@workstation:~$ cd Activity_8
justin@workstation:~/Activity_8$ ls
ansible.cfg  inventory  README.md
```

Use the command “git add ‘filename’” to make a queue for the commit command. (make sure to put a commit message)



```
justin@workstation:~/Activity_8$ git add ansible.cfg
justin@workstation:~/Activity_8$ git add inventory
justin@workstation:~/Activity_8$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   ansible.cfg
        new file:   inventory

justin@workstation:~/Activity_8$ git commit ansible.cfg inventory
Aborting commit due to empty commit message.
justin@workstation:~/Activity_8$ git commit ansible.cfg inventory
[main 3586ccc] Done commit
 2 files changed, 17 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 inventory
```

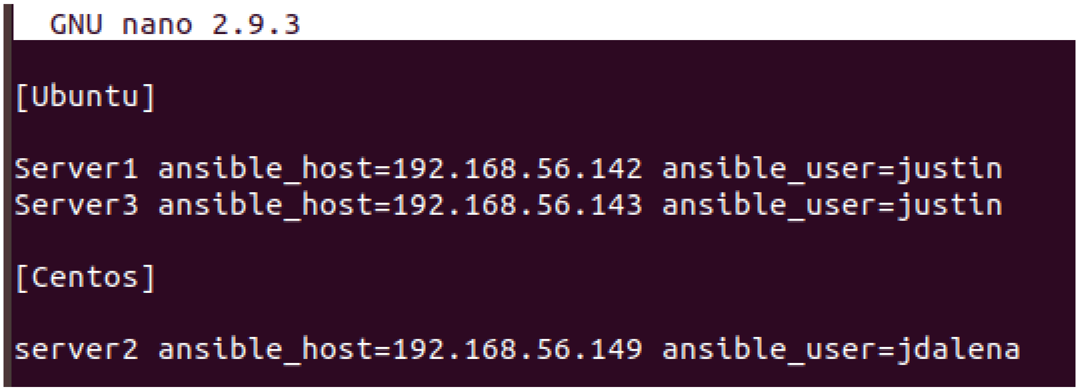
Use git push to properly apply the changes to your repository from the workstation.



```
justin@workstation:~/Activity_8$ git push -u origin main
Counting objects: 4, done.
Delta compression using up to 3 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 497 bytes | 497.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To github.com:Peanutz23/Activity_8.git
   2a5f035..3586ccc  main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
justin@workstation:~/Activity_8$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
justin@workstation:~/Activity_8$ 
```

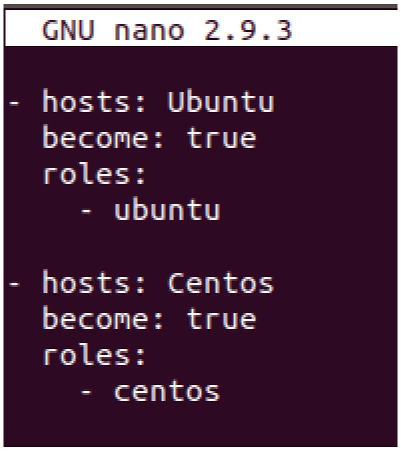
Now that the ansible and inventory are added create a playbook file. In the playbook file, create a set of codes that will install Nagios in both your Ubuntu and Centos servers.

INVENTORY



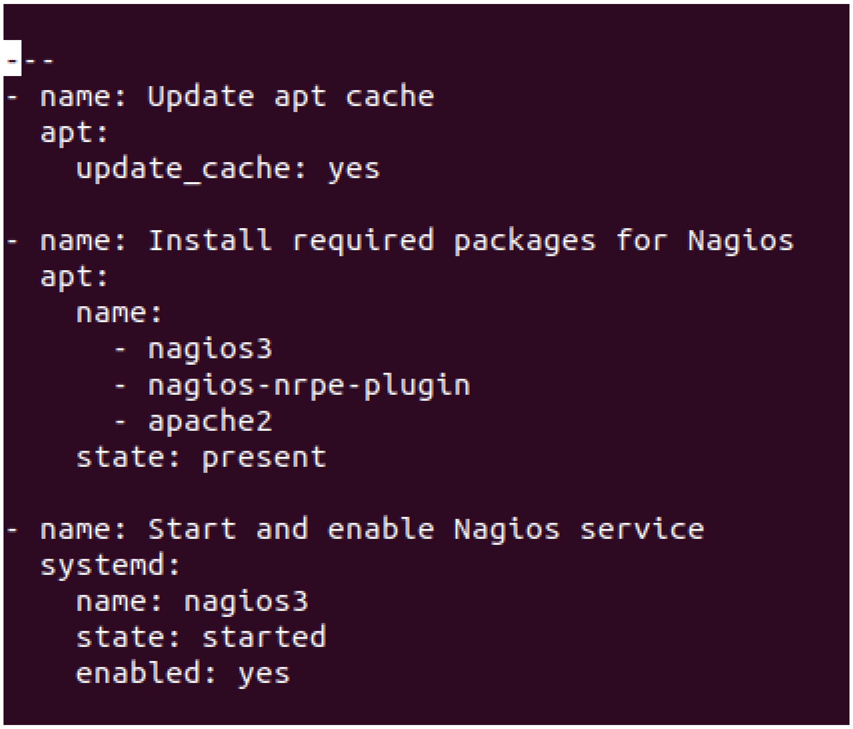
```
  GNU nano 2.9.3

[Ubuntu]

Server1 ansible_host=192.168.56.142 ansible_user=justin
Server3 ansible_host=192.168.56.143 ansible_user=justin

[Centos]

server2 ansible_host=192.168.56.149 ansible_user=jdalena
```

RUN.YML



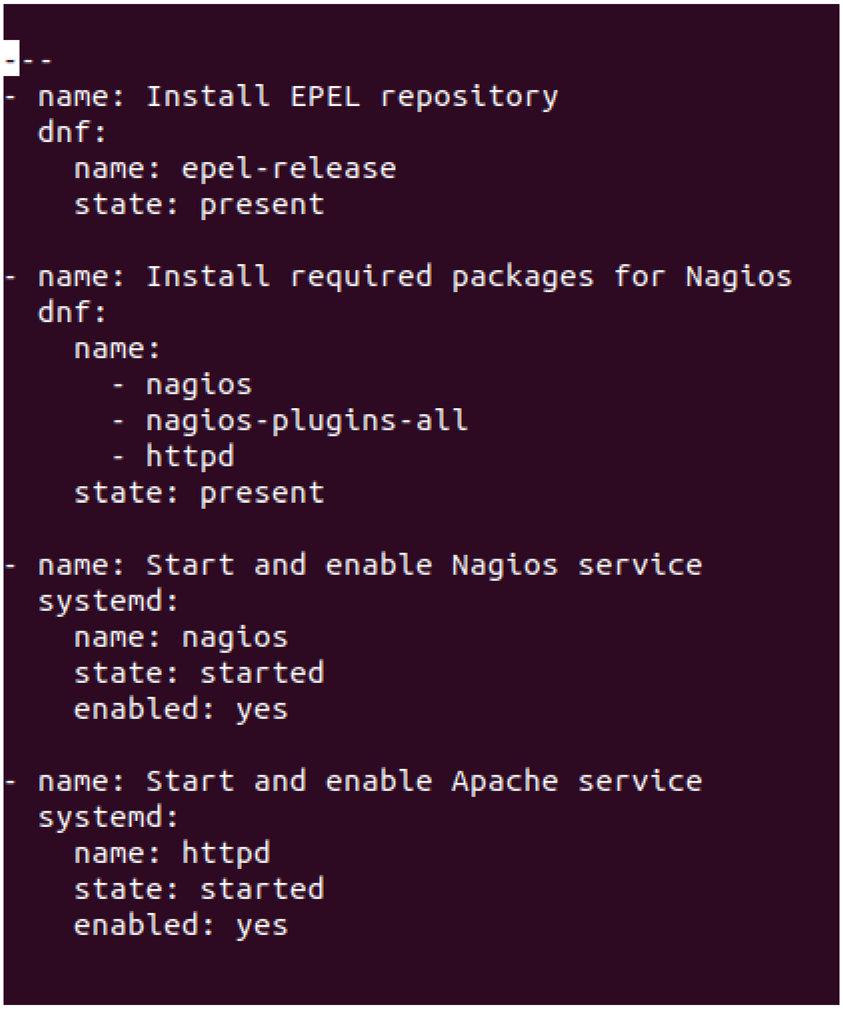
```
  GNU nano 2.9.3

- hosts: Ubuntu
  become: true
  roles:
    - ubuntu

- hosts: Centos
  become: true
  roles:
    - centos
```

MAIN.YML (Ubuntu)

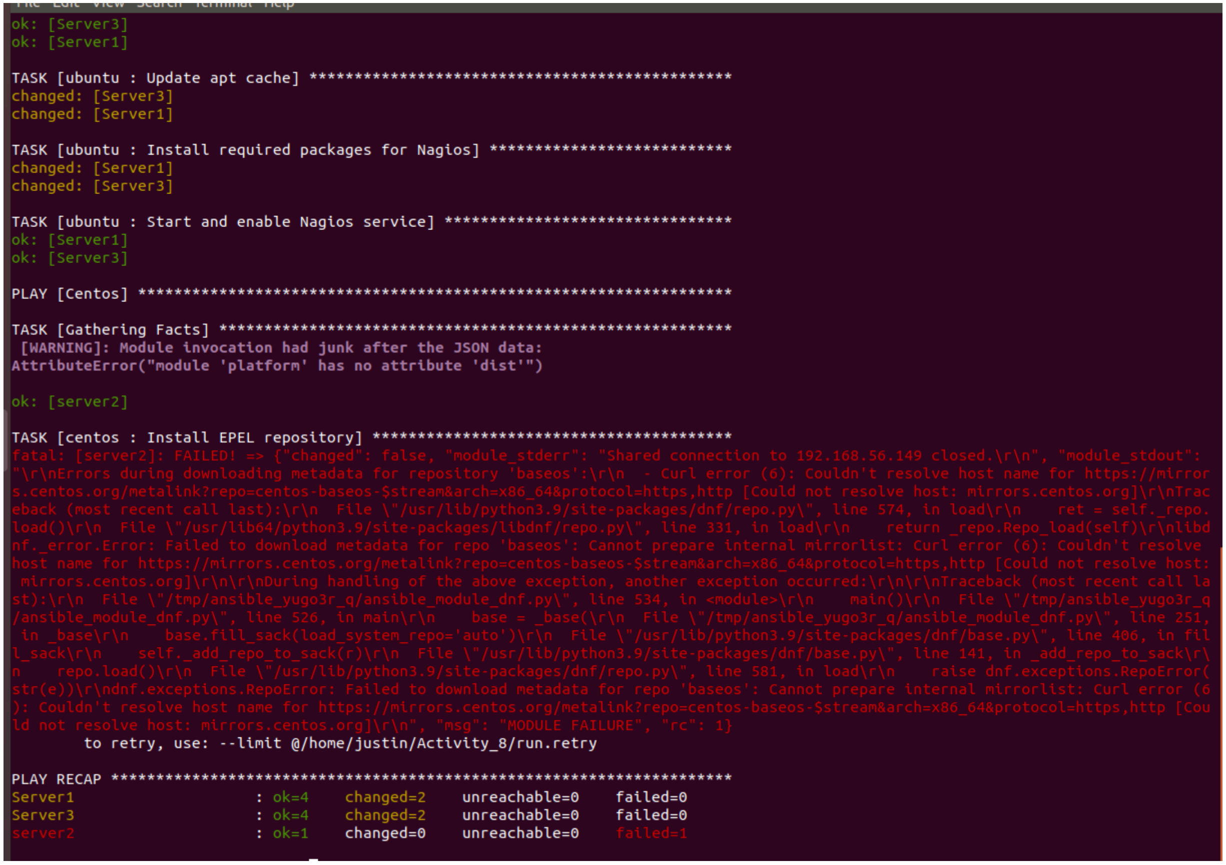


```
---
- name: Update apt cache
  apt:
    update_cache: yes

- name: Install required packages for Nagios
  apt:
    name:
      - nagios3
      - nagios-nrpe-plugin
      - apache2
    state: present

- name: Start and enable Nagios service
  systemd:
    name: nagios3
    state: started
    enabled: yes
```

MAIN.YML (CentOS)



```
---
- name: Install EPEL repository
  dnf:
    name: epel-release
    state: present

- name: Install required packages for Nagios
  dnf:
    name:
      - nagios
      - nagios-plugins-all
      - httpd
    state: present

- name: Start and enable Nagios service
  systemd:
    name: nagios
    state: started
    enabled: yes

- name: Start and enable Apache service
  systemd:
    name: httpd
    state: started
    enabled: yes
```



```
ok: [Server3]
ok: [Server1]

TASK [ubuntu : Update apt cache] ***********************************************
changed: [Server3]
changed: [Server1]

TASK [ubuntu : Install required packages for Nagios] ***************************
changed: [Server1]
changed: [Server3]

TASK [ubuntu : Start and enable Nagios service] ********************************
ok: [Server1]
ok: [Server3]

PLAY [Centos] ******************************************************************

TASK [Gathering Facts] *********************************************************
 [WARNING]: Module invocation had junk after the JSON data:
AttributeError("module 'platform' has no attribute 'dist'")

ok: [server2]

TASK [centos : Install EPEL repository] ****************************************
fatal: [server2]: FAILED! => {"changed": false, "module_stderr": "Shared connection to 192.168.56.149 closed.\r\n", "module_stdout":
"\r\nErrors during downloading metadata for repository 'baseos':\r\n  - Curl error (6): Couldn't resolve host name for https://mirror
s.centos.org/metalink?repo=centos-baseos-$stream&arch=x86_64&protocol=https,http [Could not resolve host: mirrors.centos.org]\r\nTrac
eback (most recent call last):\r\n  File \"/usr/lib/python3.9/site-packages/dnf/repo.py\", line 574, in load\r\n    ret = self._repo.
load()\r\n  File \"/usr/lib64/python3.9/site-packages/libdnf/repo.py\", line 331, in load\r\n    return _repo.Repo_load(self)\r\nlibd
nf._error.Error: Failed to download metadata for repo 'baseos': Cannot prepare internal mirrorlist: Curl error (6): Couldn't resolve
host name for https://mirrors.centos.org/metalink?repo=centos-baseos-$stream&arch=x86_64&protocol=https,http [Could not resolve host:
 mirrors.centos.org]\r\n\r\nDuring handling of the above exception, another exception occurred:\r\n\r\nTraceback (most recent call la
st):\r\n  File \"/tmp/ansible_yugo3r_q/ansible_module_dnf.py\", line 534, in <module>\r\n    main()\r\n  File \"/tmp/ansible_yugo3r_q
/ansible_module_dnf.py\", line 526, in main\r\n    base = _base(\r\n  File \"/tmp/ansible_yugo3r_q/ansible_module_dnf.py\", line 251,
 in _base\r\n    base.fill_sack(load_system_repo='auto')\r\n  File \"/usr/lib/python3.9/site-packages/dnf/base.py\", line 406, in fil
l_sack\r\n    self._add_repo_to_sack(r)\r\n  File \"/usr/lib/python3.9/site-packages/dnf/base.py\", line 141, in _add_repo_to_sack\r\
n    repo.load()\r\n  File \"/usr/lib/python3.9/site-packages/dnf/repo.py\", line 581, in load\r\n    raise dnf.exceptions.RepoError(
str(e))\r\ndnf.exceptions.RepoError: Failed to download metadata for repo 'baseos': Cannot prepare internal mirrorlist: Curl error (6
): Couldn't resolve host name for https://mirrors.centos.org/metalink?repo=centos-baseos-$stream&arch=x86_64&protocol=https,http [Cou
ld not resolve host: mirrors.centos.org]\r\n", "msg": "MODULE FAILURE", "rc": 1}
        to retry, use: --limit @/home/justin/Activity_8/run.retry

PLAY RECAP *********************************************************************
Server1                    : ok=4    changed=2    unreachable=0    failed=0
Server3                    : ok=4    changed=2    unreachable=0    failed=0
server2                    : ok=1    changed=0    unreachable=0    failed=1
```

Reflections:

Answer the following:

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

   Availability monitoring tools, such as Nagios, enable proactive issue detection and significantly reduce downtime, leading to improved system reliability. They enhance visibility into system performance, allowing for better resource allocation and faster resolution of problems. Ultimately, these tools contribute to increased customer satisfaction and reduced operational costs.

Conclusions:

In conclusion, availability monitoring tools like Nagios are vital for maintaining high system reliability and minimizing downtime. By facilitating proactive issue detection and providing real-time insights into performance, these tools empower IT teams to respond swiftly to potential problems. Improved resource allocation and faster resolution times further enhance operational efficiency. Ultimately, the use of such monitoring tools leads to greater customer satisfaction and lower operational costs for organizations.