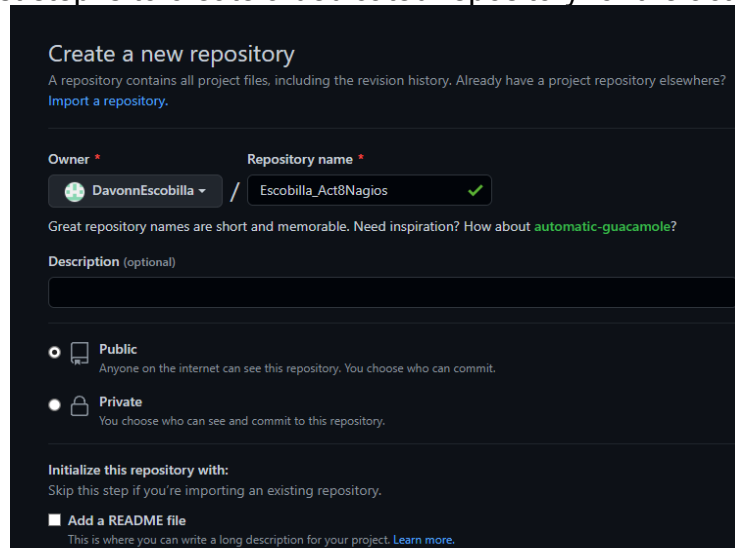


Name: Davonn Escobilla	Date Performed: 25/10/2022
Course/Section: CPE31S24	Date Submitted: 25/10/2022
Instructor: Dr. Jonathan Taylar	Semester and SY: 1st Sem, 2022-2023
Activity 8: Install, Configure, and Manage Availability Monitoring tools	
1. Objectives	
Create and design a workflow that installs, 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 (screenshots and explanations)

First step is to create a dedicated repository for the activity.



The screenshot shows the GitHub 'Create a new repository' page. The 'Owner' is set to 'DavonnEscobilla' and the 'Repository name' is 'Escobilla_Act8Nagios'. The 'Description' field is empty. The 'Public' option is selected under 'Initialize this repository with:'. Below the form, there is a link to 'Learn more'.

Second step is to clone the repository created into the managed node.

```
davonn@workstation: ~$ git clone git@github.com:DavonnEscobilla/Escobilla_Act8Nagios.git
```

```
davonn@workstation:~$ ls
CPE232_Davonn  Documents  Music  Public
CPE232_Escobilla  Downloads  nano.save  Templates
Desktop        Escobilla_Act8Nagios  Pictures  Videos
```

Third step is to create configuration for ansible and inventory. This is to connect the managed node to control nodes.

```
davonn@workstation: ~/Escobilla_Act8Nagios
GNU nano 4.8  ansible.cfg
[defaults]
inventory = inventory
private_key_file = ~/.ssh/ansible
```

```
davonn@workstation: ~/Escobilla_Act8Nagios
GNU nano 4.8                                inventory
192.168.56.105
192.168.56.103
```

Before proceeding to the next step, the connectivity should be verified by pinging the control nodes.

```
davonn@workstation: ~/Escobilla_Act8Nagios
davonn@workstation:~/Escobilla_Act8Nagios$ ansible -m ping all
192.168.56.103 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
192.168.56.105 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
```

Fourth step, I will now configure the site.yml playbook to update the repository and the cache also assign roles for the nagios installation.

```
davonn@workstation: ~/Escobilla_Act8Nagios
GNU nano 4.8 site.yml
---
- hosts: all
  become: true
  pre_tasks:
    - name: update repository index (Ubuntu)
      tags: always
      apt:
        update_cache: yes
        changed_when: false
        when: ansible_distribution == "Ubuntu"
    - name: update repository index (CentOS)
      tags: always
      yum:
        update_cache: yes
        changed_when: false
        when: ansible_distribution == "CentOS"
- hosts: all
  become: true
  roles:
    - nagios
```

Fifth step, I will now create directories for the roles inside the repository and then create main.yml that is configured with the tasks for nagios installation.

```
davonn@workstation:~/Escobilla_Act8Nagios$ mkdir roles
davonn@workstation:~/Escobilla_Act8Nagios$ cd roles
davonn@workstation:~/Escobilla_Act8Nagios/roles$ mkdir nagios
davonn@workstation:~/Escobilla_Act8Nagios/roles$ cd nagios
davonn@workstation:~/Escobilla_Act8Nagios/roles/nagios$ mkdir tasks
davonn@workstation:~/Escobilla_Act8Nagios/roles/nagios$ cd tasks
davonn@workstation:~/Escobilla_Act8Nagios/roles/nagios/tasks$ nano main.yml
```

```
davonn@workstation: ~/Escobilla_Act8Nagios/roles/nagios/...
GNU nano 4.8 main.yml
- name: Install nagios in Ubuntu
  apt:
    name:
      - nagios4
    state: latest
    update_cache: yes
    when: ansible_distribution == "Ubuntu"

- name: Install nagios in CentOS
  yum:
    name:
      - nagios
    state: latest
    update_cache: yes
    when: ansible_distribution == "CentOS"

- name: Enabling/Starting Nagios on CentOS
  tags: centos, apache, httpd, nagios
  service:
    name: httpd
    state: started
    when: ansible_distribution == "CentOS"
```

Before proceeding to the next step, verification for the directory must be checked using the tree command.

```
davonn@workstation:~/Escobilla_Act8Nagios$ tree
.
├── ansible.cfg
├── inventory
├── roles
│   ├── nagios
│   │   └── tasks
│   │       └── main.yml
└── site.yml

3 directories, 4 files
davonn@workstation:~/Escobilla_Act8Nagios$
```

Now run the playbook.

```
davonn@workstation: ~/Escobilla_Act8Nagios
davonn@workstation:~/Escobilla_Act8Nagios$ ansible-playbook --ask-become-pass site.yml
BECOME password:

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.105]
ok: [192.168.56.103]

TASK [update repository index (Ubuntu)] *****
*
skipping: [192.168.56.105]
ok: [192.168.56.103]

TASK [update repository index (CentOS)] *****
*
skipping: [192.168.56.103]
ok: [192.168.56.105]

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.103]
ok: [192.168.56.105]
```

```
davonn@workstation: ~/Escobilla_Act8Nagios
TASK [Gathering Facts] *****
*
ok: [192.168.56.103]
ok: [192.168.56.105]

TASK [nagios : Install nagios in Ubuntu] *****
*
skipping: [192.168.56.103]
ok: [192.168.56.103]

TASK [nagios : Install nagios in CentOS] *****
*
skipping: [192.168.56.103]
ok: [192.168.56.105]

TASK [nagios : Enabling/Starting Nagios on CentOS] *****
*
skipping: [192.168.56.103]
changed: [192.168.56.105]

PLAY RECAP *****
*
192.168.56.103      : ok=4    changed=0    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
192.168.56.105      : ok=5    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0

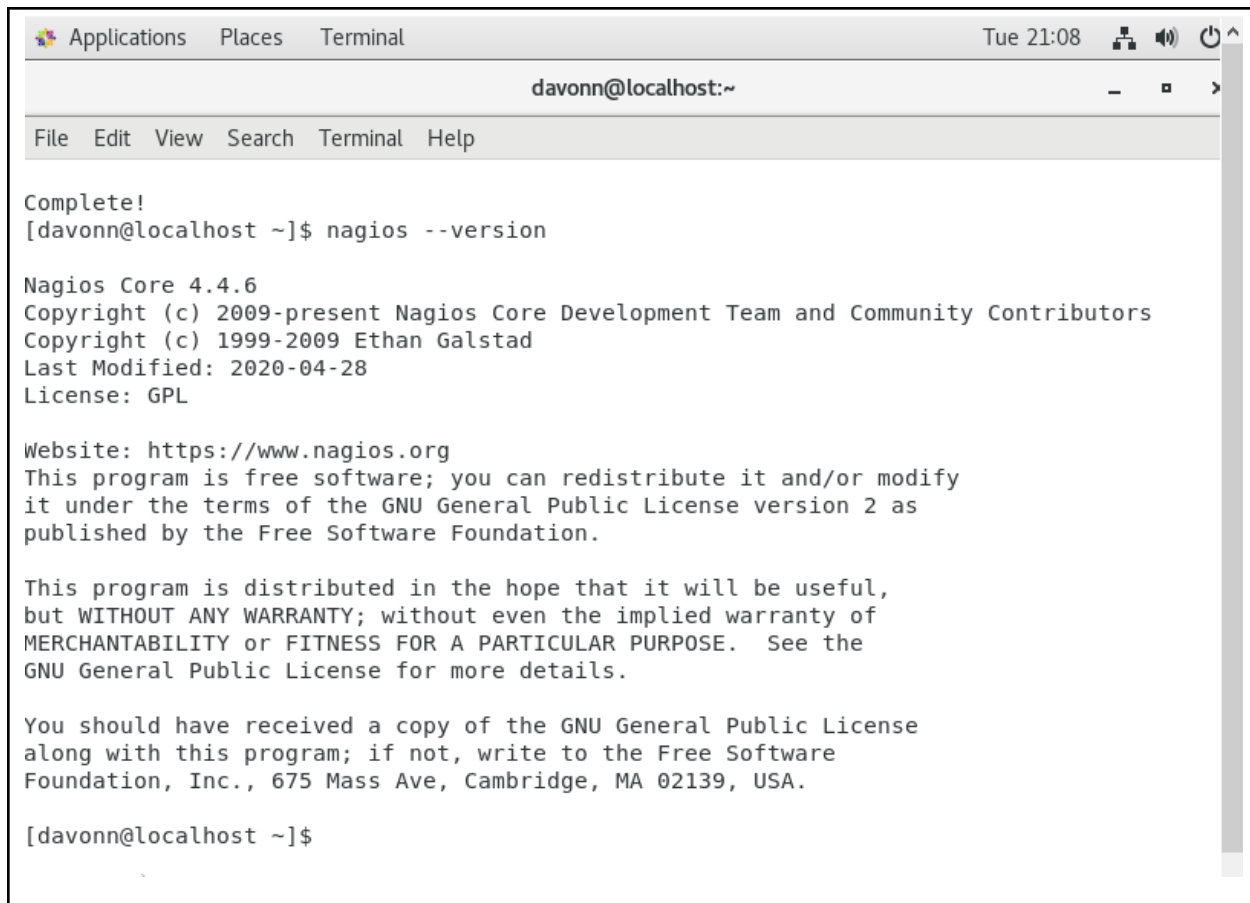
davonn@workstation:~/Escobilla_Act8Nagios$
```

The playbook ran successfully in this output as the control nodes responded with no error.

OUTPUT FOR UBUNTU SERVER 1:

```
davonn@server1: ~  
davonn@server1:~$ nagios4 --version  
  
Nagios Core 4.3.4  
Copyright (c) 2009-present Nagios Core Development Team and Community  
d Thunderbird Mail  
Copyright (c) 1999-2009 Ethan Galstad  
Last Modified: 2017-08-24  
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.
```

OUTPUT FOR CENTOS:

A screenshot of a Linux terminal window. The title bar shows 'Applications', 'Places', and 'Terminal' on the left, and 'Tue 21:08' with system icons on the right. The terminal title is 'davonn@localhost:~'. The menu bar includes 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal content shows the command 'nagios --version' being executed, resulting in a multi-line output detailing the Nagios Core version (4.4.6), copyright information (2009-present Nagios Core Development Team and Community Contributors, 1999-2009 Ethan Galstad), last modified date (2020-04-28), license (GPL), website (https://www.nagios.org), and the GNU General Public License terms.

```
Complete!
[davonn@localhost ~]$ nagios --version

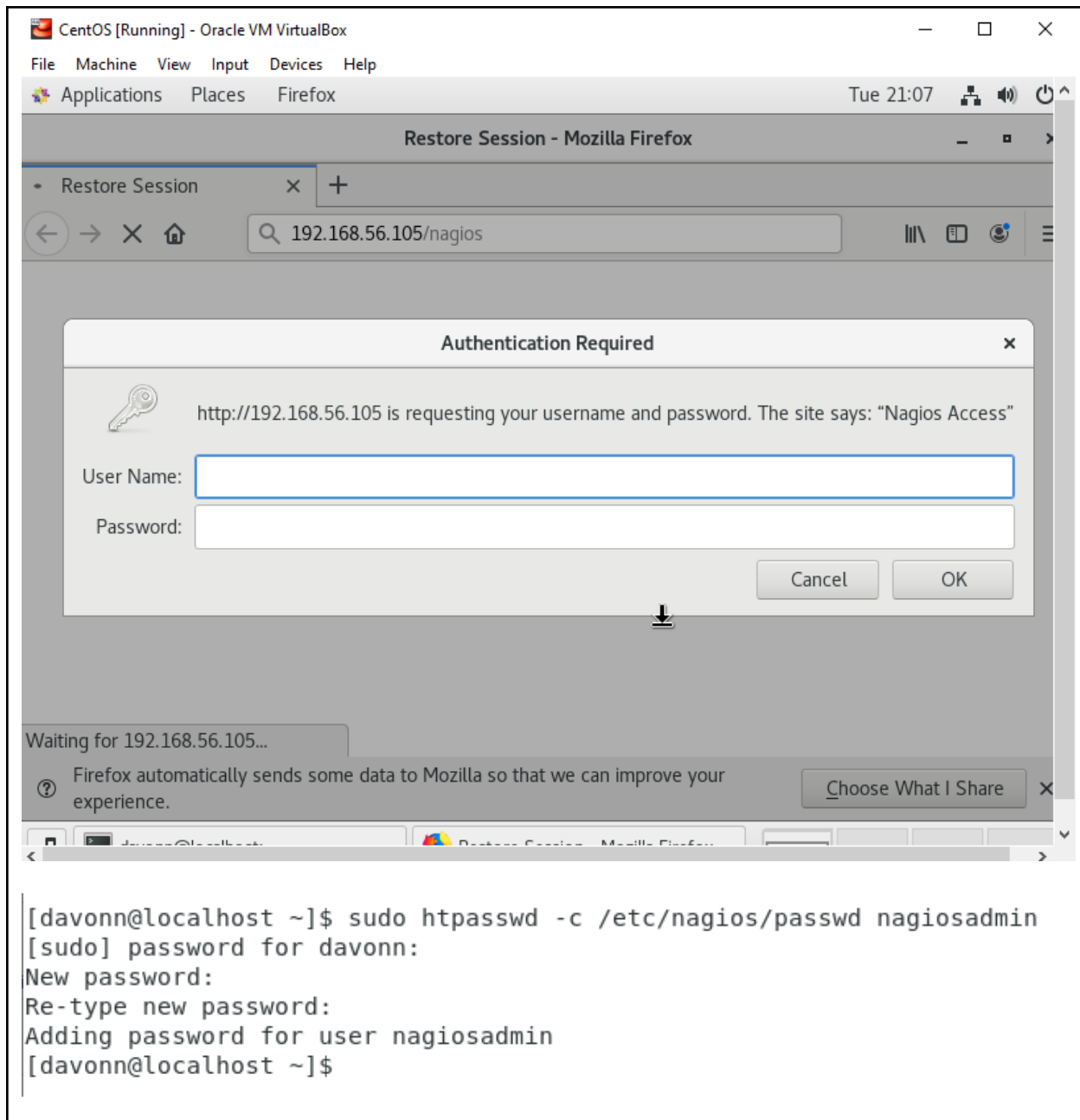
Nagios Core 4.4.6
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2020-04-28
License: GPL

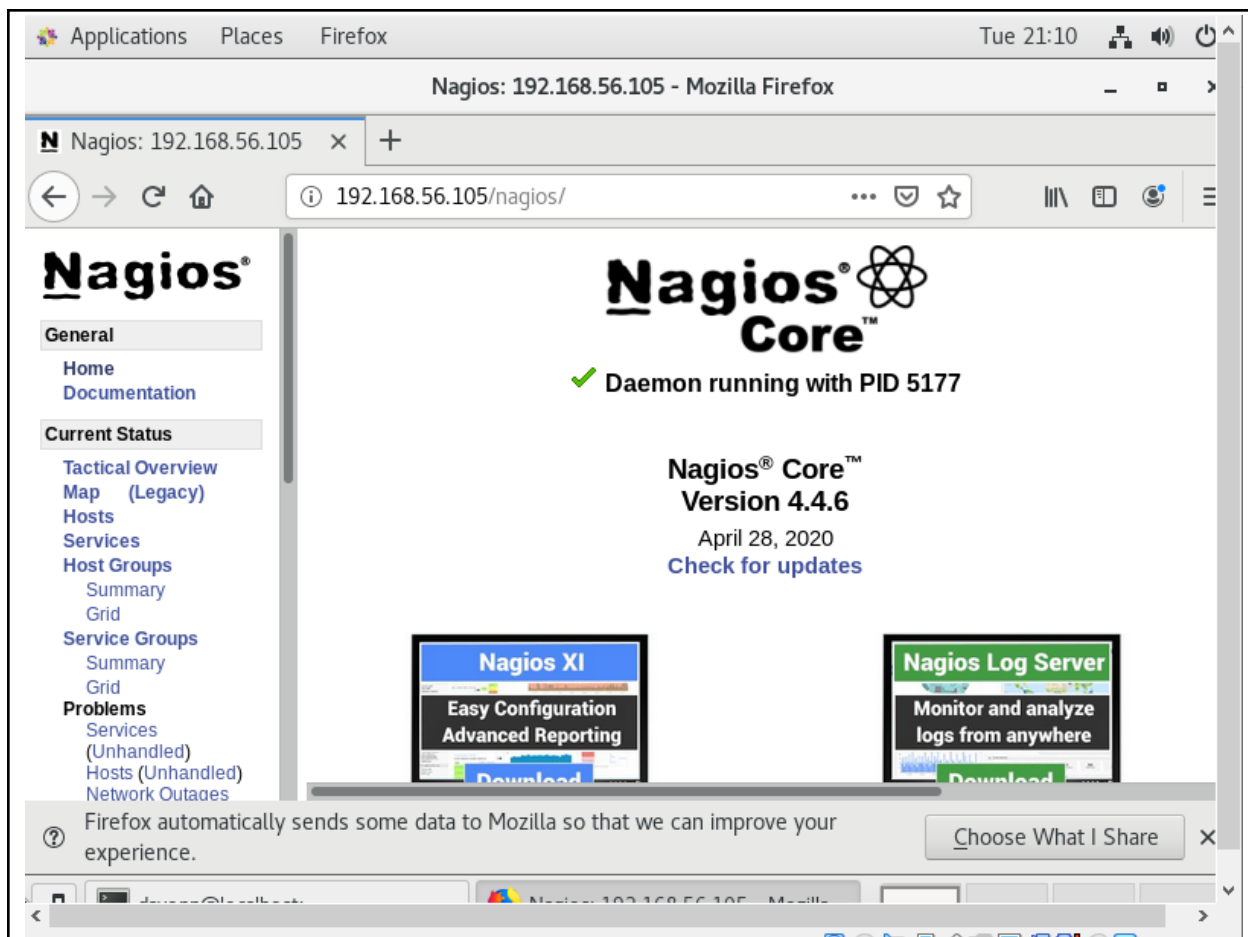
Website: https://www.nagios.org
This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License version 2 as
published by the Free Software Foundation.

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.

[davonn@localhost ~]$
```





As you can see, all the installation of nagios is a success. While on CentOS, I have encountered an issue about admin and password, so by configuring and creating an admin, I can now access the nagios on CentOS.

Finally, all we have to do is save our work on our repository in github.

```
davonn@workstation: ~/Escobilla_Act8Nagios
davonn@workstation:~/Escobilla_Act8Nagios$ git add -A
davonn@workstation:~/Escobilla_Act8Nagios$ git commit -a
Aborting commit due to empty commit message.
davonn@workstation:~/Escobilla_Act8Nagios$ git commit -m "Nagios Installation"
[master (root-commit) b7bf56c] Nagios Installation
 5 files changed, 84 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 inventory
 create mode 100644 roles/nagios/tasks/main.yml
 create mode 100644 site.yml
 create mode 100644 site.yml.save
davonn@workstation:~/Escobilla_Act8Nagios$ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (10/10), 1.03 KiB | 525.00 KiB/s, done.
Total 10 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:DavonnEscobilla/Escobilla_Act8Nagios.git
 * [new branch]      master -> master
davonn@workstation:~/Escobilla_Act8Nagios$
```

Reflections:

Answer the following:

1. What are the benefits of having an availability monitoring tool? The benefits of a monitoring tool is that you can do the maintenance on all of the connected nodes efficiently without resorting to manual configuration. All you have to do is manage your main workstation in which site downtimes can be prevented using this tool. We can check the uptime of the servers and apps while notifying the webmasters for any impending problems which can be mitigated and prevent loss.

Conclusions:

While doing the activity, my understanding on configuration about managing availability monitoring tools have been satisfied. I understand how roles from the last activity helped me to manage the control nodes and use monitoring tools to further understand and prevent future problems by thoroughly reviewing the uptime of the server.