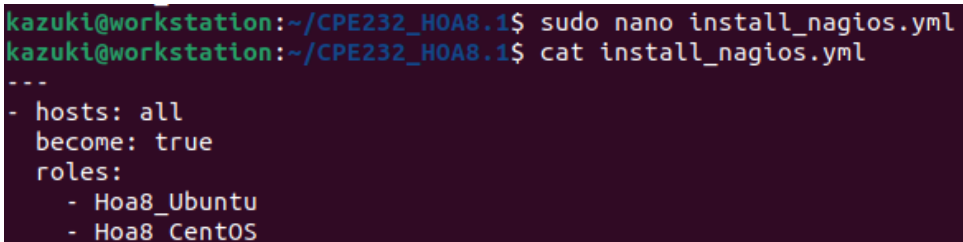


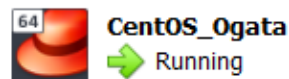
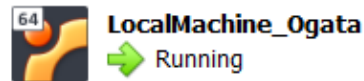


Name: Kazuki A. Ogata	Date Performed: October 20, 2023
Course/Section: CPE 232 - CPE31S5	Date Submitted: October 20, 2023
Instructor: Engr. Roman Richard	Semester and SY: 1st semester S.Y 2023-2024
Activity 8: Install, Configure, and Manage Availability Monitoring tools	
1. Objectives	
Create and design 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	
<div>1. Create a playbook that installs Nagios in both Ubuntu and CentOS. Apply the concept of creating roles.</div> <div>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.)</div> <div>3. Show an output of the installed Nagios for both Ubuntu and CentOS.</div> <div>4. Make sure to create a new repository in GitHub for this activity.</div>	
4. Output (screenshots and explanations)	
<div>1. Create a playbook that installs Nagios in both Ubuntu and CentOS. Apply the concept of creating roles.</div> <div></div> <div>This is my final Playbook file, it contains 2 roles (Ubuntu and CentOS). I applied all of my learnings to the previous activity in this activity.</div>	
<div></div> <div></div> <div>This is the contents of my two roles. You can see in number 2 and 3 the output of this playbook.</div>	

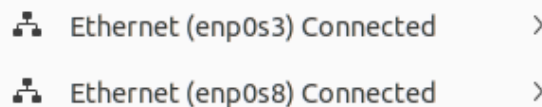
2. Describe how you did step 1. (Provide screenshots and explanations in your report. Make your report detailed such that it will look like a manual.)

Step 1: Open Virtual Machines



Open Oracle VM Virtual box and start 3 virtual machines, 1 for control node (LocalMachine_Ogata) and 2 for managed nodes for Ubuntu and CentOS (Server2_Ogata and CentOS_Ogata).

Step 2: Check the Internet Connection of Virtual Machines



On the local machine and remote servers, type “ping google.com” or click the top right side of your vm to check if the Ethernets are connected (like image above). We are doing this, to avoid errors.

Step 3: Verify Connection

```
kazuki@server2:~$ hostname -I [kazuki@centos ~]$ hostname -I
10.0.2.15 192.168.56.123 10.0.2.15 192.168.56.127 192.168.122.1

kazuki@workstation:~$ ping 192.168.56.123
PING 192.168.56.123 (192.168.56.123) 56(84) bytes of data:
64 bytes from 192.168.56.123: icmp_seq=1 ttl=64 time=2.55 ms
64 bytes from 192.168.56.123: icmp_seq=2 ttl=64 time=0.634 ms
^C
--- 192.168.56.123 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 0.634/1.591/2.549/0.957 ms
kazuki@workstation:~$ ping 192.168.56.127
PING 192.168.56.127 (192.168.56.127) 56(84) bytes of data:
64 bytes from 192.168.56.127: icmp_seq=1 ttl=64 time=1.71 ms
64 bytes from 192.168.56.127: icmp_seq=2 ttl=64 time=7.89 ms
^C
--- 192.168.56.127 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 1.707/4.798/7.890/3.091 ms

kazuki@workstation:~$ ssh kazuki@192.168.56.123
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-33-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Expanded Security Maintenance for Applications is not enabled.

47 updates can be applied immediately.
41 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Wed Oct 18 20:10:45 2023 from 192.168.56.121
kazuki@server2:~$
logout
Connection to 192.168.56.123 closed.
kazuki@workstation:~$ ssh kazuki@192.168.56.127
Last login: Fri Oct 20 15:42:01 2023
[kazuki@centos ~]$
```

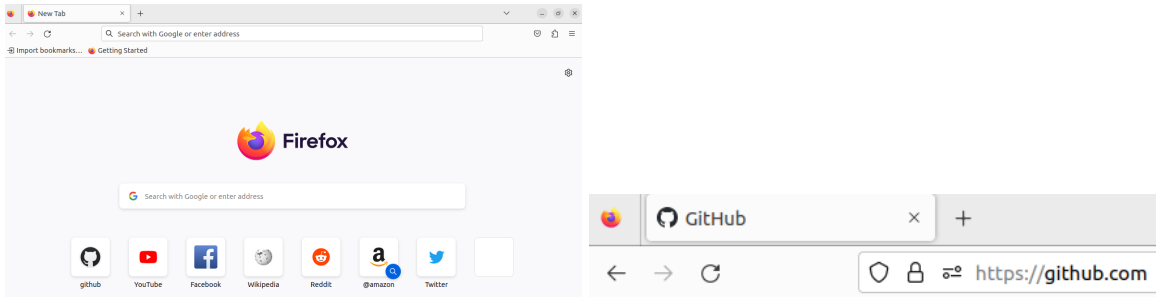
The most important part before starting this activity is checking if there is a connection between control nodes and managed nodes. In this part, we can use *ifconfig* or *hostname -I* to check the IP address and then use *ping* or *ssh user@host* to verify the connections. To check if it is successful you should get the same output shown above.

Step 4: Setup Git

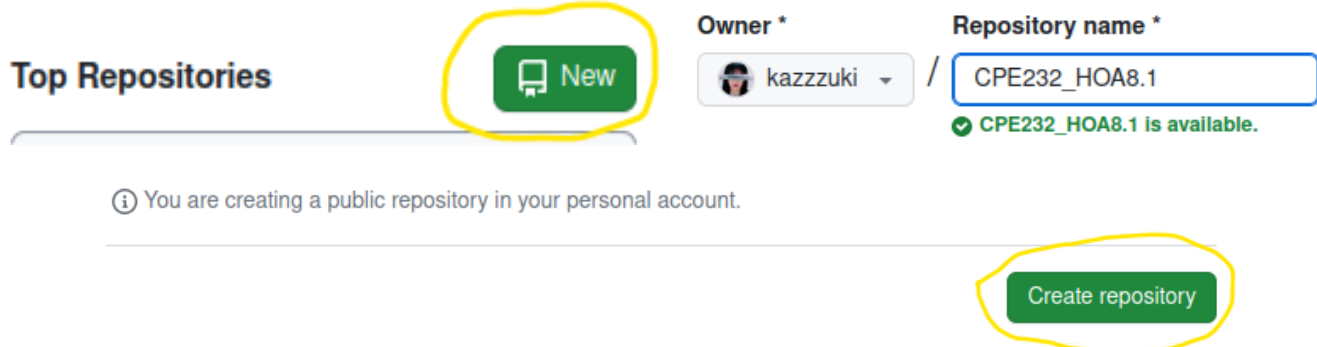
```
kazuki@workstation:~$ which git  
/usr/bin/git
```

```
kazuki@workstation:~$ git --version  
git version 2.34.1
```

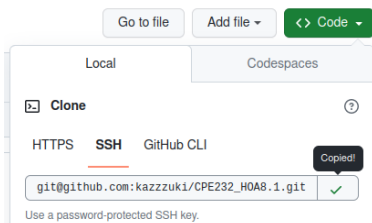
On the local machine, use the *which git* command to check if you already have git. If the directory *usr/bin/git* is displayed then you already have git and you don't need to install git. Just in case you still don't have a git you can use the command *sudo apt install git*. You can use *git --version* to check the version.



Open any web server available in your workstation and go to www.github.com, if you don't have an account create one by signing up. If you already have an account proceed to the next step.



Create a new repository and name it "CPE232_HOA8.1". Check the images above in creating a repository and follow them. I already have an SSH key on my GitHub, if you don't have one yet, you can create a new SSH key on GitHub by going to your profile's settings and then click the "SSH and GPG keys" and click the "New SSH Key". On the local machine, use the command "cat .ssh/id_rsa/pub" and copy its output and then paste it on the GitHub "Key".



Clone GitHub repository in local machine, browse your new repository and click the "<> Code" to get the SSH link, copy it.

```
kazuki@workstation:~$ git clone git@github.com:kazzzuki/CPE232_HOA8.1.git
Cloning into 'CPE232_HOA8.1'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

Clone the new repository in your local machine by using the command

`git clone git@github.com:jvtaylor-cpe/CPE232_yourname.git`.

```
kazuki@workstation:~$ ls
CPE232_HOA6.1  CPE232_KAZUKI  Documents  Music  Pictures  snap  Videos
CPE232_HOA8.1  Desktop        Downloads  Ogata_PrelimExam  Public  Templates
```

Use the `ls` command to check the cloned repository. You can also personalize your git by using the commands `git config --global user.name "Your Name"` and `git config --global user.email "Your Email"`. Use the `cat ~/.gitconfig` command to verify it.

Step 5: Create a Playbook

```
kazuki@workstation:~$ cd CPE232_HOA8.1
kazuki@workstation:~/CPE232_HOA8.1$
```

On the local machine, use the command `cd` command to go to the new repository directory. So we can proceed in creating files needed for this activity.

```
kazuki@workstation:~/CPE232_HOA8.1$ sudo nano inventory
GNU nano 6.2                                inventory *

[Hoas8_Ubuntu]
192.168.56.123  ansible_connection=ssh

[Hoas8_CentOS]
192.168.56.127  ansible_connection=ssh
```

Create an inventory file. Inside the inventory put the managed nodes IP addresses. I created two groups (Hoas8_Ubuntu and Hoas8_CentOS) and put the IP address of remote servers and added the "ansible_connection=ssh" to specify the connection to be used in connecting to the remote servers. I created a group so it will be easily called in roles later.

```
kazuki@workstation:~/CPE232_HOA8.1$ sudo nano ansible.cfg
GNU nano 6.2                                ansible.cfg *

[defaults]

inventory = inventory
```

Create an ansible.cfg file. I added the default configuration and set the default inventory file that Ansible should use.

```
kazuki@workstation:~/CPE232_HOA8.1$ ansible all --list-hosts
hosts (2):
 192.168.56.123
 192.168.56.127
kazuki@workstation:~/CPE232_HOA8.1$ ansible all -n ping
192.168.56.123 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
192.168.56.127 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
```

Verify the hosts in the inventory. As we can see in the output, it is all successful, we can now proceed in creating a playbook.

```
kazuki@workstation:~/CPE232_HOA8.1$ mkdir -p roles/Hoa8_Ubuntu/tasks
kazuki@workstation:~/CPE232_HOA8.1$ mkdir -p roles/Hoa8_CentOS/tasks
kazuki@workstation:~/CPE232_HOA8.1$ ls
ansible.cfg  install_nagios.yml  inventory  README.md  roles
kazuki@workstation:~/CPE232_HOA8.1$ cd roles
kazuki@workstation:~/CPE232_HOA8.1/roles$ ls
Hoa8_CentOS  Hoa8_Ubuntu
kazuki@workstation:~/CPE232_HOA8.1/roles$ cd Hoa8_Ubuntu
kazuki@workstation:~/CPE232_HOA8.1/roles/Hoa8_Ubuntu$ ls
tasks
kazuki@workstation:~/CPE232_HOA8.1/roles/Hoa8_Ubuntu$ cd ../Hoa8_CentOS
kazuki@workstation:~/CPE232_HOA8.1/roles/Hoa8_CentOS$ ls
tasks
```

Create directories using the **mkdir** command. I am creating directories where I will put a role. I created Hoa8_Ubuntu and Hoa8_CentOS with tasks directories inside them.

```
kazuki@workstation:~/CPE232_HOA8.1$ sudo nano /home/kazuki/CPE232_HOA8.1/roles/Hoa8_Ubuntu/tasks/main.yml
kazuki@workstation:~/CPE232_HOA8.1$ cat /home/kazuki/CPE232_HOA8.1/roles/Hoa8_Ubuntu/tasks/main.yml
- name: Update Ubuntu Package Cache
  apt:
    update_cache: yes
    state: present
  when: ansible_distribution == "Ubuntu"

- name: Resolving dpkg Issue
  command: sudo dpkg --configure -a
  when: ansible_distribution == "Ubuntu"
  async: 3600
  poll: 0

- name: Waiting(pause) for dpkg to complete
  pause:
    seconds: 10
  when: ansible_distribution == "Ubuntu"

- name: Install Nagios in Ubuntu
  apt:
    name: nagios4-core
    state: present
  when: ansible_distribution == "Ubuntu"
  async: 3600
  poll: 0
```

Inside the tasks of Hoa8_Ubuntu, I create another file “main.yml” where I put the tasks in installing Nagios in Ubuntu. I added the update just to make sure my Ubuntu server is updated and there will be no error. I added resolving the dpkg issue, since when running my playbook, I encountered an error related to that so in order to solve it, I added it inside the roles so I will not do it manually. I put “async” and “poll” under the resolving dpkg issue, which will make that play run faster. I added the waiting for dpkg to complete, since it takes time for it to complete, so I added 10 second, but in running the playbook, you have a choice either run the waiting 10 second or abort it. And lastly, the play for installing Nagios on Ubuntu, my Ubuntu supports the “nagios4-core” package so I use that to install the Nagios.

```
kazuki@workstation:~/CPE232_HOA8.1$ sudo nano /home/kazuki/CPE232_HOA8.1/roles/Hoa8_CentOS/tasks/main.yml
kazuki@workstation:~/CPE232_HOA8.1$ cat /home/kazuki/CPE232_HOA8.1/roles/Hoa8_CentOS/tasks/main.yml
- name: Update CentOS Package Cache
  dnf:
    update_cache: yes
    state: present
    use_backend: dnf
    when: ansible_distribution == "CentOS"

- name: Resolving dpkg Issue
  command: sudo dpkg --configure -a
  when: ansible_distribution == "CentOS"
  async: 3600
  poll: 0

- name: Waiting(pause) for dpkg to complete
  pause:
    seconds: 10
    when: ansible_distribution == "CentOS"

- name: Install Nagios in CentOS
  dnf:
    name: nagios
    state: present
    use_backend: dnf
    when: ansible_distribution == "CentOS"
    async: 3600
    poll: 0
```

Inside the tasks of Hoa8_CentOS, I create another file “main.yml” where I put the tasks in installing Nagios in CentOS. I added the update just to make sure my Ubuntu server is updated and there will be no error. I added resolving the dpkg issue, since when running my playbook, I encountered an error related to that so in order to solve it, I added it inside the roles so I will not do it manually. I put “async” and “poll” under the resolving dpkg issue, which will make that play run faster. I added the waiting for dpkg to complete, since it takes time for it to complete, so I added 10 second, but in running the playbook, you have a choice either run the waiting 10 second or abort it. And lastly, the play for installing Nagios on CentOS, my CentOS supports the “nagios” package so I use that to install the Nagios.

```
kazuki@workstation:~/CPE232_HOA8.1$ sudo nano install_nagios.yml
kazuki@workstation:~/CPE232_HOA8.1$ cat install_nagios.yml
---
- name: Install Nagios on Ubuntu
  hosts: Hoa8_Ubuntu
  become: true
  roles:
    - Hoa8_Ubuntu

- name: Install Nagios on CentOS
  hosts: Hoa8_CentOS
  become: true
  roles:
    - Hoa8_CentOS
```

I created a playbook using the **sudo nano** command and put the hosts and roles inside it. We can still make this playbook shorter and get the same output. The image below is the shorter version of the playbook above.

```
kazuki@workstation:~/CPE232_HOA8.1$ sudo nano install_nagios.yml
kazuki@workstation:~/CPE232_HOA8.1$ cat install_nagios.yml
---
- hosts: all
  become: true
  roles:
    - Hoa8_Ubuntu
    - Hoa8_CentOS
```


Step 6: Running the Playbook

```
kazuki@workstation:~/CPE232_HOA8.1$ ansible-playbook --ask-become-pass install_nagios.yml
BECOME password:

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

TASK [Gathering Facts] *****
ok: [192.168.56.123]

TASK [Hoa8_Ubuntu : Update Ubuntu Package Cache] *****
changed: [192.168.56.123]

TASK [Hoa8_Ubuntu : Resolving dpkg Issue] *****
changed: [192.168.56.123]

TASK [Hoa8_Ubuntu : Waiting(pause) for dpkg to complete] *****
Pausing for 10 seconds
(ctrl+C then 'c' = continue early, ctrl+C then 'A' = abort)
ok: [192.168.56.123]

TASK [Hoa8_Ubuntu : Install Nagios in Ubuntu] *****
changed: [192.168.56.123]
PLAY [Install Nagios on CentOS] *****

TASK [Gathering Facts] *****
ok: [192.168.56.127]

TASK [Hoa8_CentOS : Update CentOS Package Cache] *****
ok: [192.168.56.127]

TASK [Hoa8_CentOS : Resolving dpkg Issue] *****
changed: [192.168.56.127]

TASK [Hoa8_CentOS : Waiting(pause) for dpkg to complete] *****
Pausing for 10 seconds
(ctrl+C then 'c' = continue early, ctrl+C then 'A' = abort)
ok: [192.168.56.127]

TASK [Hoa8_CentOS : Install Nagios in CentOS] *****
changed: [192.168.56.127]

PLAY RECAP *****
192.168.56.123      : ok=5    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.127      : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

This is the output of the first playbook (the longer one) .

```
kazuki@workstation:~/CPE232_HOA8.1$ ansible-playbook --ask-become-pass install_nagios.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.127]
ok: [192.168.56.123]

TASK [Hoa8_Ubuntu : Update Ubuntu Package Cache] *****
skipping: [192.168.56.127]
changed: [192.168.56.123]

TASK [Hoa8_Ubuntu : Resolving dpkg Issue] *****
skipping: [192.168.56.127]
changed: [192.168.56.123]

TASK [Hoa8_Ubuntu : Waiting(pause) for dpkg to complete] *****
Pausing for 10 seconds
(ctrl+C then 'c' = continue early, ctrl+C then 'A' = abort)
ok: [192.168.56.123]

TASK [Hoa8_Ubuntu : Install Nagios in Ubuntu] *****
skipping: [192.168.56.127]
changed: [192.168.56.123]

TASK [Hoa8_CentOS : Update CentOS Package Cache] *****
skipping: [192.168.56.123]
ok: [192.168.56.127]

TASK [Hoa8_CentOS : Resolving dpkg Issue] *****
skipping: [192.168.56.123]
changed: [192.168.56.127]

TASK [Hoa8_CentOS : Waiting(pause) for dpkg to complete] *****
skipping: [192.168.56.123]

TASK [Hoa8_CentOS : Install Nagios in CentOS] *****
skipping: [192.168.56.123]
changed: [192.168.56.127]

PLAY RECAP *****
192.168.56.123      : ok=5    changed=3    unreachable=0    failed=0    skipped=4    rescued=0    ignored=0
192.168.56.127      : ok=4    changed=2    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
```

This is the output of the second playbook (the shorter one)

As we can see, both playbooks have the same output, the only difference is the length of the playbook. The shorter playbook is better for me, it's easy to modify when encountering an error, like earlier I encountered errors in roles and it's hard to go up and down to fix the error.

3. Show an output of the installed Nagios for both Ubuntu and CentOS.

```
kazuki@server2:~$ nagios4 --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.
```

In Ubuntu, I used the command “nagios4 –version” to check if my Ansible Playbook really installed Nagios. As we can see, it indeed installed the Nagios since we can see its version.

```
[kazuki@centos ~]$ nagios --version

Nagios Core 4.4.9
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2022-11-16
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.

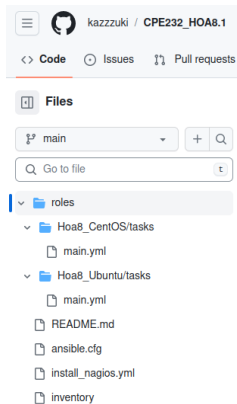
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
```

In CentOS, I used the command “nagios –version” to check if my Ansible Playbook really installed Nagios. As we can see, it indeed installed the Nagios since we can see its version.

4. Make sure to create a new repository in GitHub for this activity.

```
kazuki@workstation:~/CPE232_HOA8.1$ git add .
kazuki@workstation:~/CPE232_HOA8.1$ git commit -m "hoa8"
[main 8fee6e4] hoa8
5 files changed, 71 insertions(+)
create mode 100644 ansible.cfg
create mode 100644 install_nagios.yml
create mode 100644 inventory
create mode 100644 roles/Hoa8_CentOS/tasks/main.yml
create mode 100644 roles/Hoa8_Ubuntu/tasks/main.yml
kazuki@workstation:~/CPE232_HOA8.1$ git push
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 2 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (12/12), 1.17 KiB | 598.00 KiB/s, done.
Total 12 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To github.com:kazzzuki/CPE232_HOA8.1.git
9127a07..8fee6e4 main -> main
```

I use the **git add** command to add all the files inside my repository directory on my GitHub followed by the **git commit -m** command to add a message and then lastly the **git push** command to push it in my GitHub repository.



-> All files created in this activity are all successfully saved in my GitHub repository. I check everything, all of them are the same as my home repository directory. Here is the link of my GitHub Repository:

https://github.com/kazzzuki/CPE232_HOA8.1

Reflections:

Answer the following:

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

- The benefits of having an availability monitoring tool are issue/error detection and resolution, improved performance + user experience, minimized downtime, and security. For example, the ARIS website of TIP uses an availability monitoring tool. If that tool detects an error in their website like can't login, can't see enrolled subjects, can't see grade, etc. There will be something like a notification or alarm that alerts the IT team, once they are notified they will now be able to fix that error as soon as possible to avoid inconvenience to the users.

Conclusions:

This activity, help me realize the error I made in the last activity. In the last activity, I thought why my playbook is running without any error, it is successful. But in this activity, since it's about installation of Nagios. I made a playbook for it, and it was successful but when I check my Ubuntu and CentOS servers there are no Nagios installed. It means my playbook is running without error but it is not doing the tasks inside my roles directory. This activity taught me to always check if the task is really doing its tasks. Overall, I learned how to install, configure, troubleshoot, and manage enterprise monitoring tools using Ansible.