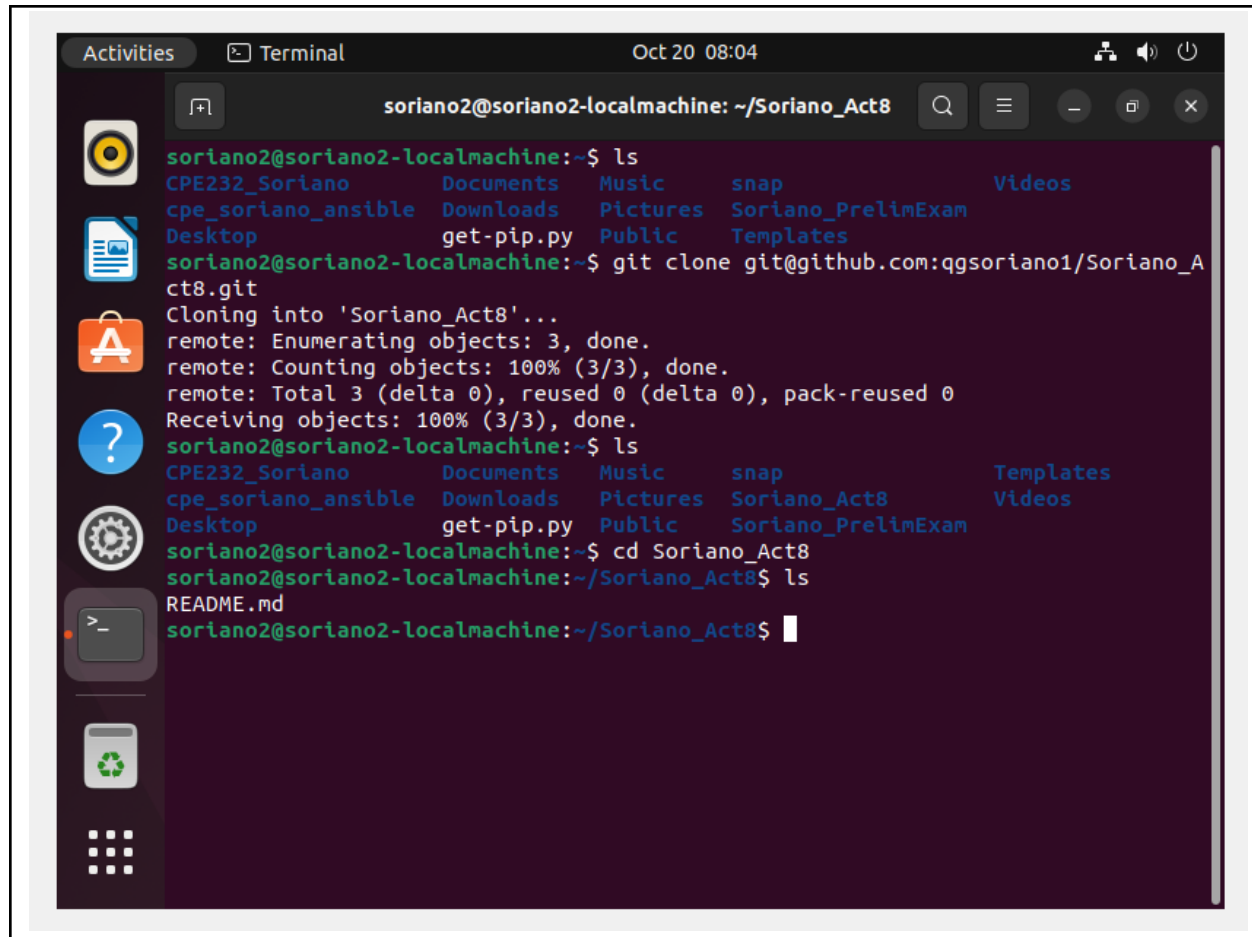
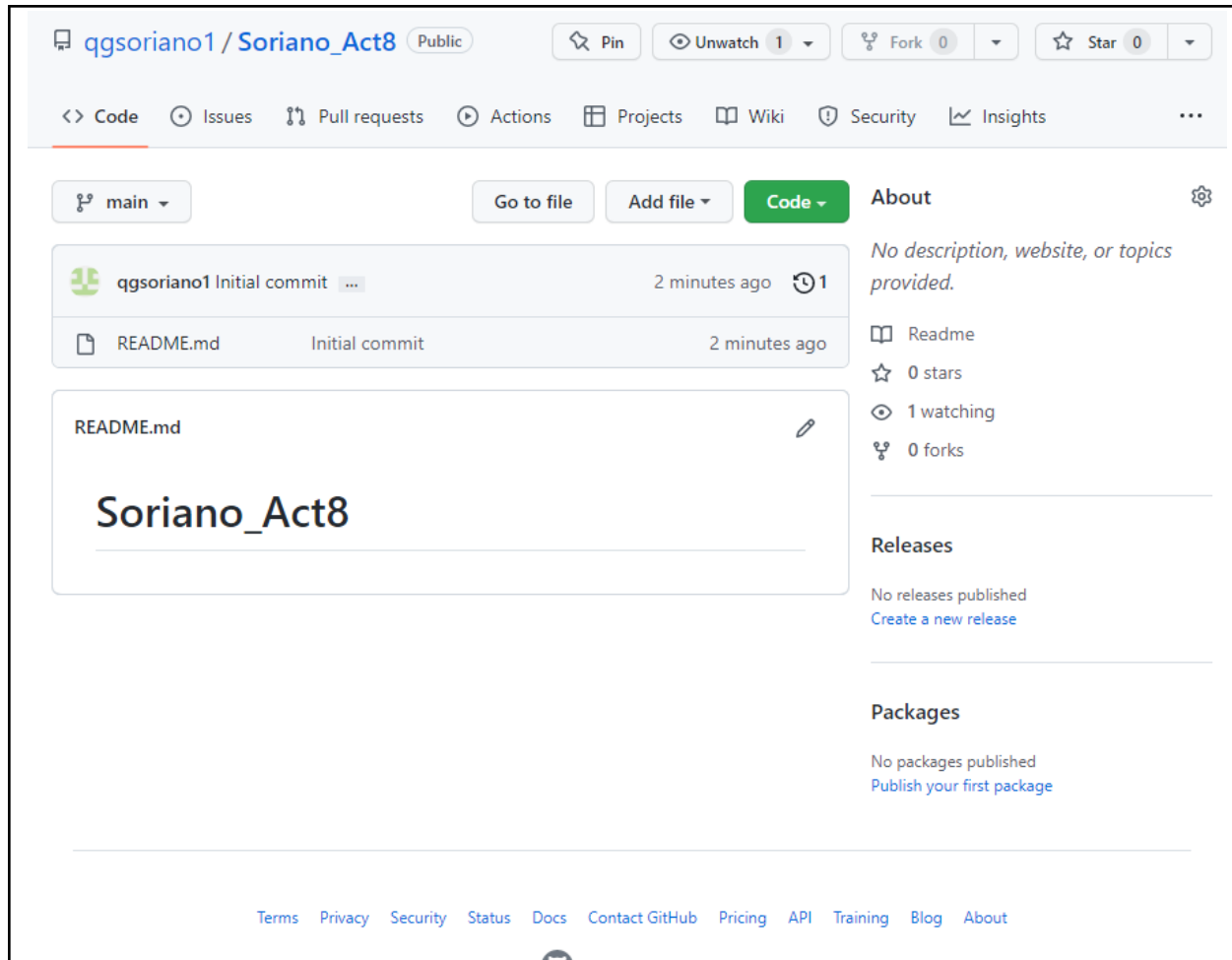


Name: Gabriel Soriano	Date Performed: October 20, 2022
Course/Section:	Date Submitted: October 20, 2022
Instructor: Engr. Taylar	Semester and SY: 1st sem - SY 2022-2023
Activity 8: Install, Configure, and Manage Availability Monitoring tools	
1. Objectives	
Create and design a workflow that installs, 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	
REQUIREMENT: Create a separate repository for this activity. SCREENSHOT:	





The screenshot shows a GitHub repository page for 'qgsoriano1 / Soriano_Act8'. The repository is public and has 0 forks, 0 stars, and 1 watch. The main branch is 'main'. The repository contains a single file, 'README.md', which was committed 2 minutes ago. The README content is 'Soriano_Act8'. The right sidebar shows the repository's metadata, including the README, 0 stars, 1 watching, and 0 forks. Below this, there are sections for 'Releases' and 'Packages', both of which are empty. The footer of the page includes links to Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About, along with the GitHub logo and copyright notice.

qgsoriano1 / Soriano_Act8 Public

Pin Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights

main Go to file Add file Code About

qgsoriano1 Initial commit 2 minutes ago 1

README.md Initial commit 2 minutes ago

README.md

Soriano_Act8

No description, website, or topics provided.

Readme 0 stars 1 watching 0 forks

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

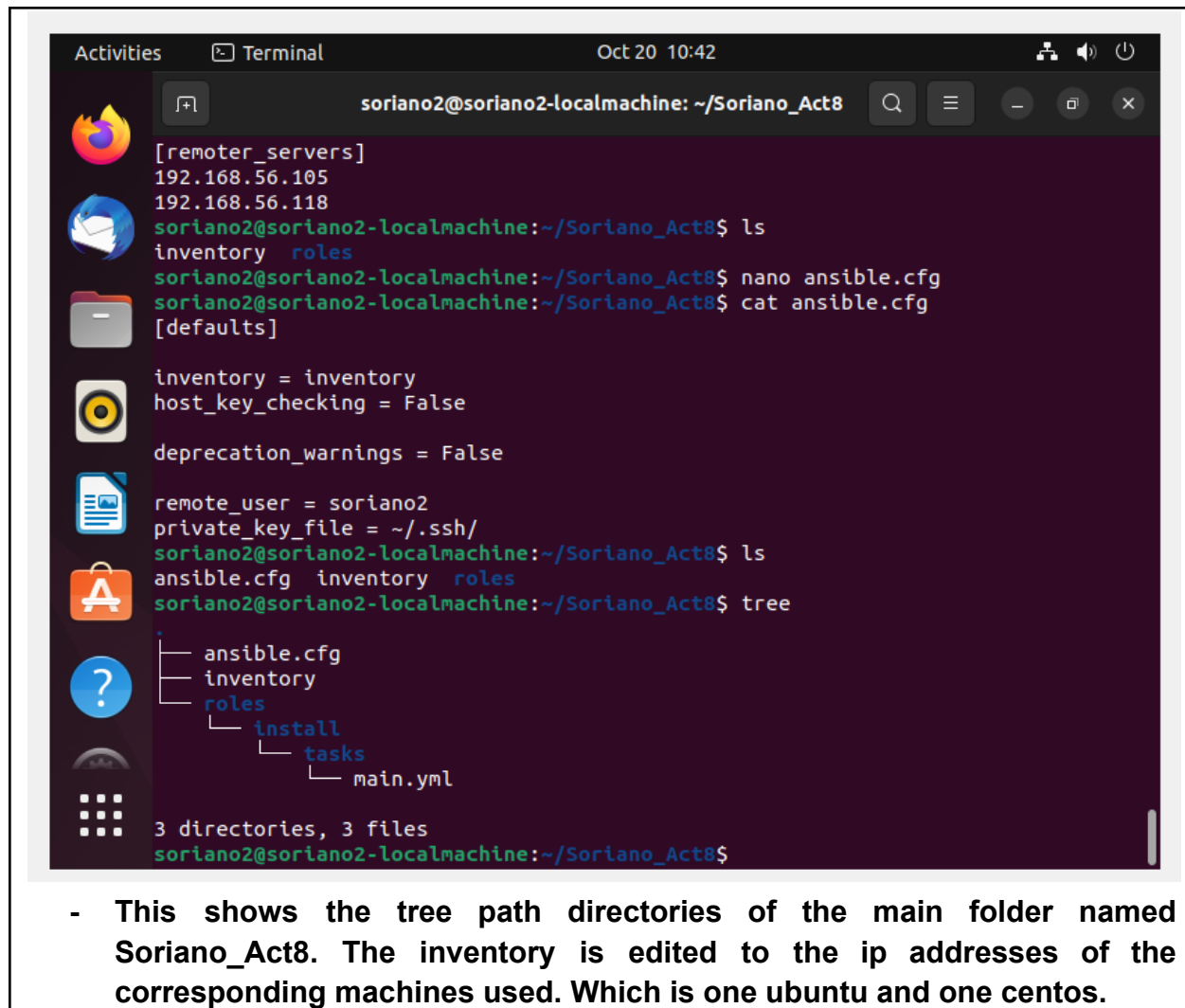
[Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

© 2022 GitHub, Inc.

- This shows the successful creation and git cloning of a new repository for this activity. The repository is named Soriano_Act8.

1. Create a playbook that installs Nagios in both Ubuntu and CentOS. Apply the concept of creating roles.

SCREENSHOTS:



The screenshot shows a terminal window titled "Terminal" with the date and time "Oct 20 10:42". The user is logged in as "soriano2" on a machine named "soriano2-localmachine", and the current directory is "~/Soriano_Act8". The terminal output shows the following commands and results:

```
[remoter_servers]
192.168.56.105
192.168.56.118
soriano2@soriano2-localmachine:~/Soriano_Act8$ ls
inventory  roles
soriano2@soriano2-localmachine:~/Soriano_Act8$ nano ansible.cfg
soriano2@soriano2-localmachine:~/Soriano_Act8$ cat ansible.cfg
[defaults]

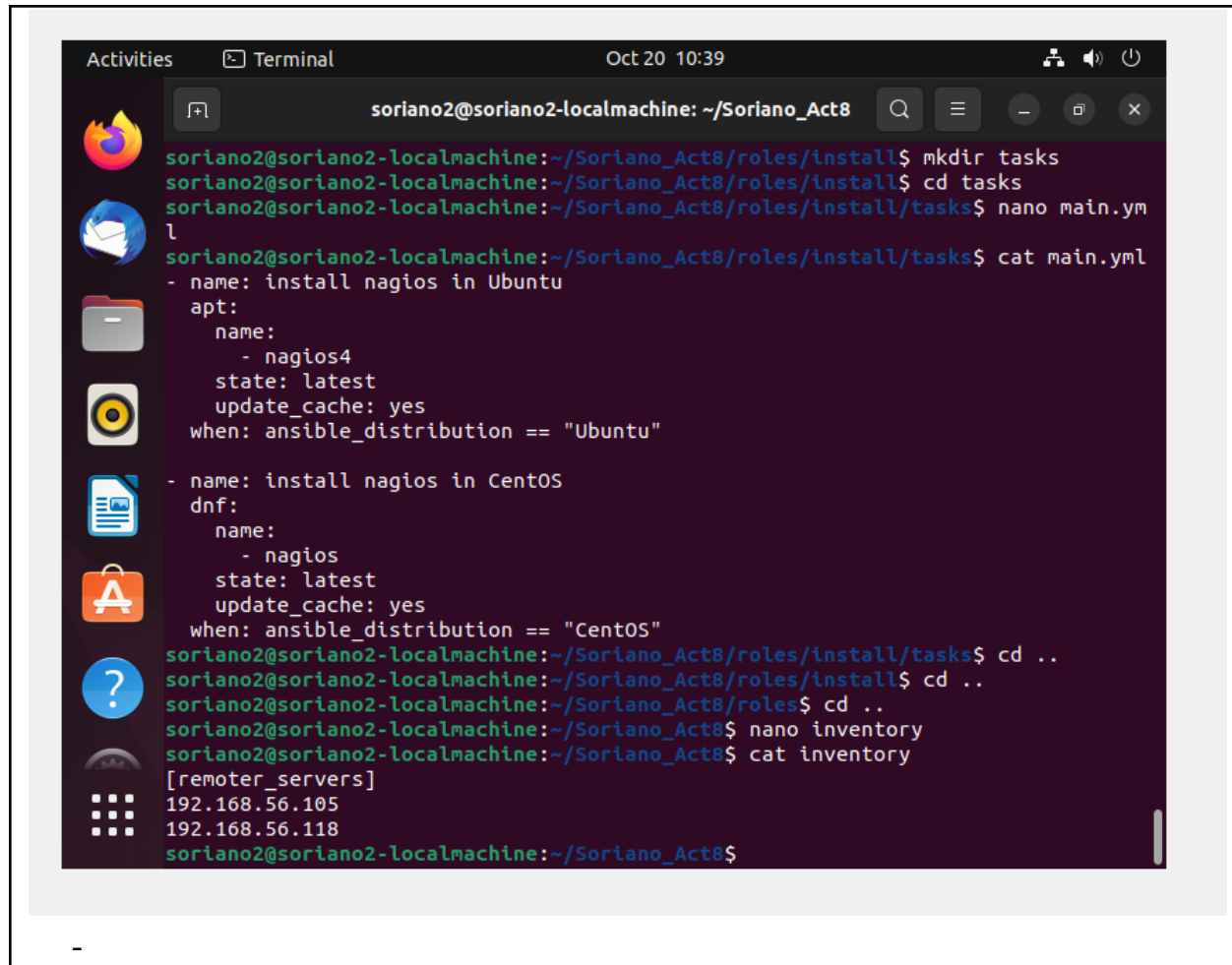
inventory = inventory
host_key_checking = False

deprecation_warnings = False

remote_user = soriano2
private_key_file = ~/.ssh/
soriano2@soriano2-localmachine:~/Soriano_Act8$ ls
ansible.cfg  inventory  roles
soriano2@soriano2-localmachine:~/Soriano_Act8$ tree
.
├── ansible.cfg
├── inventory
└── roles
    ├── install
    └── tasks
        └── main.yml

3 directories, 3 files
soriano2@soriano2-localmachine:~/Soriano_Act8$
```

- This shows the tree path directories of the main folder named Soriano_Act8. The inventory is edited to the ip addresses of the corresponding machines used. Which is one ubuntu and one centos.



The screenshot shows a terminal window titled "Terminal" with the date and time "Oct 20 10:39". The user is logged in as "soriano2" on a machine named "soriano2-localmachine". The current directory is "~/Soriano_Act8". The terminal shows the following sequence of commands and outputs:

```
soriano2@soriano2-localmachine: ~/Soriano_Act8
soriano2@soriano2-localmachine:~/Soriano_Act8/roles/install$ mkdir tasks
soriano2@soriano2-localmachine:~/Soriano_Act8/roles/install$ cd tasks
soriano2@soriano2-localmachine:~/Soriano_Act8/roles/install/tasks$ nano main.yml
soriano2@soriano2-localmachine:~/Soriano_Act8/roles/install/tasks$ cat main.yml
- name: install nagios in Ubuntu
  apt:
    name:
      - nagios4
    state: latest
    update_cache: yes
    when: ansible_distribution == "Ubuntu"

- name: install nagios in CentOS
  dnf:
    name:
      - nagios
    state: latest
    update_cache: yes
    when: ansible_distribution == "CentOS"
soriano2@soriano2-localmachine:~/Soriano_Act8/roles/install/tasks$ cd ..
soriano2@soriano2-localmachine:~/Soriano_Act8/roles/install$ cd ..
soriano2@soriano2-localmachine:~/Soriano_Act8/roles$ cd ..
soriano2@soriano2-localmachine:~/Soriano_Act8$ nano inventory
soriano2@soriano2-localmachine:~/Soriano_Act8$ cat inventory
[remoter_servers]
192.168.56.105
192.168.56.118
soriano2@soriano2-localmachine:~/Soriano_Act8$
```

The left sidebar of the terminal window displays several application icons: Firefox, a mail icon, a folder icon, a CD icon, a document icon, an application icon, a question mark icon, and a grid icon.

Activities Terminal Oct 20 10:41

soriano2@soriano2-localmachine: ~/Soriano_Act8

```
- name: install nagios in CentOS
  dnf:
    name:
      - nagios
    state: latest
    update_cache: yes
  when: ansible_distribution == "CentOS"
```

soriano2@soriano2-localmachine:~/Soriano_Act8/roles/install/tasks\$ cd ..

soriano2@soriano2-localmachine:~/Soriano_Act8/roles/install\$ cd ..

soriano2@soriano2-localmachine:~/Soriano_Act8/roles\$ cd ..

soriano2@soriano2-localmachine:~/Soriano_Act8\$ nano inventory

soriano2@soriano2-localmachine:~/Soriano_Act8\$ cat inventory

```
[remoter_servers]
192.168.56.105
192.168.56.118
```

soriano2@soriano2-localmachine:~/Soriano_Act8\$ ls

```
inventory  roles
```

soriano2@soriano2-localmachine:~/Soriano_Act8\$ nano ansible.cfg

soriano2@soriano2-localmachine:~/Soriano_Act8\$ cat ansible.cfg

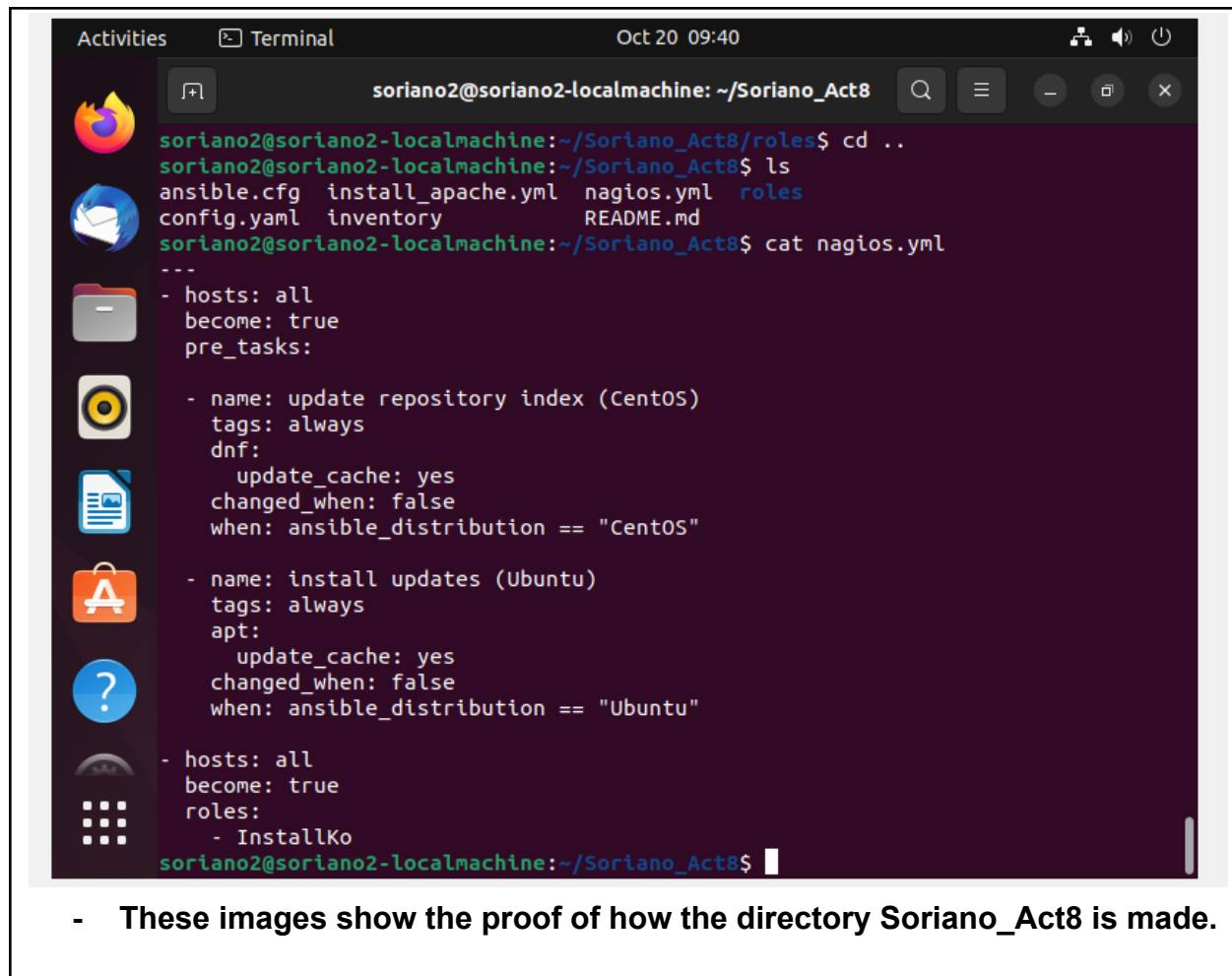
```
[defaults]

inventory = inventory
host_key_checking = False

deprecation_warnings = False

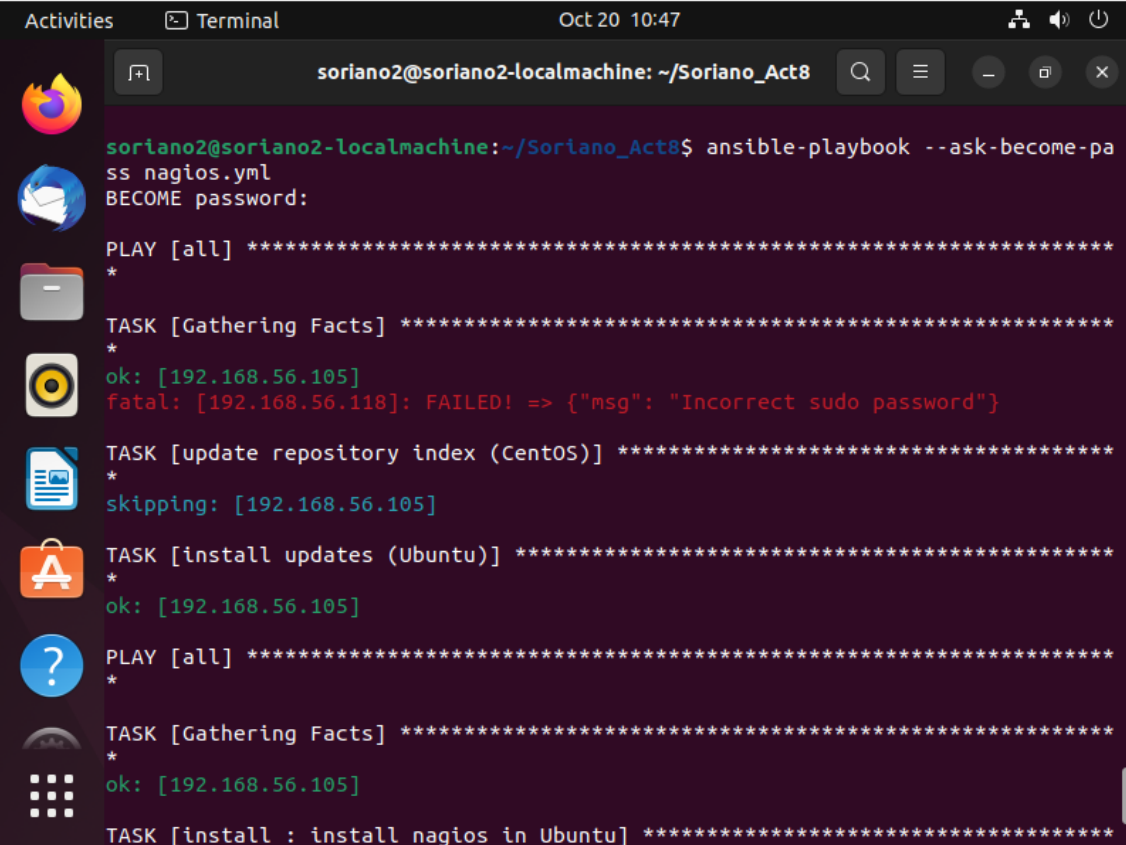
remote_user = soriano2
private_key_file = ~/.ssh/
```

soriano2@soriano2-localmachine:~/Soriano_Act8\$



ActivitiesTerminalOct 20 10:47

soriano2@soriano2-localmachine: ~/Soriano_Act8



```
soriano2@soriano2-localmachine:~/Soriano_Act8$ ansible-playbook --ask-become-pass nagios.yml
BECOME password:

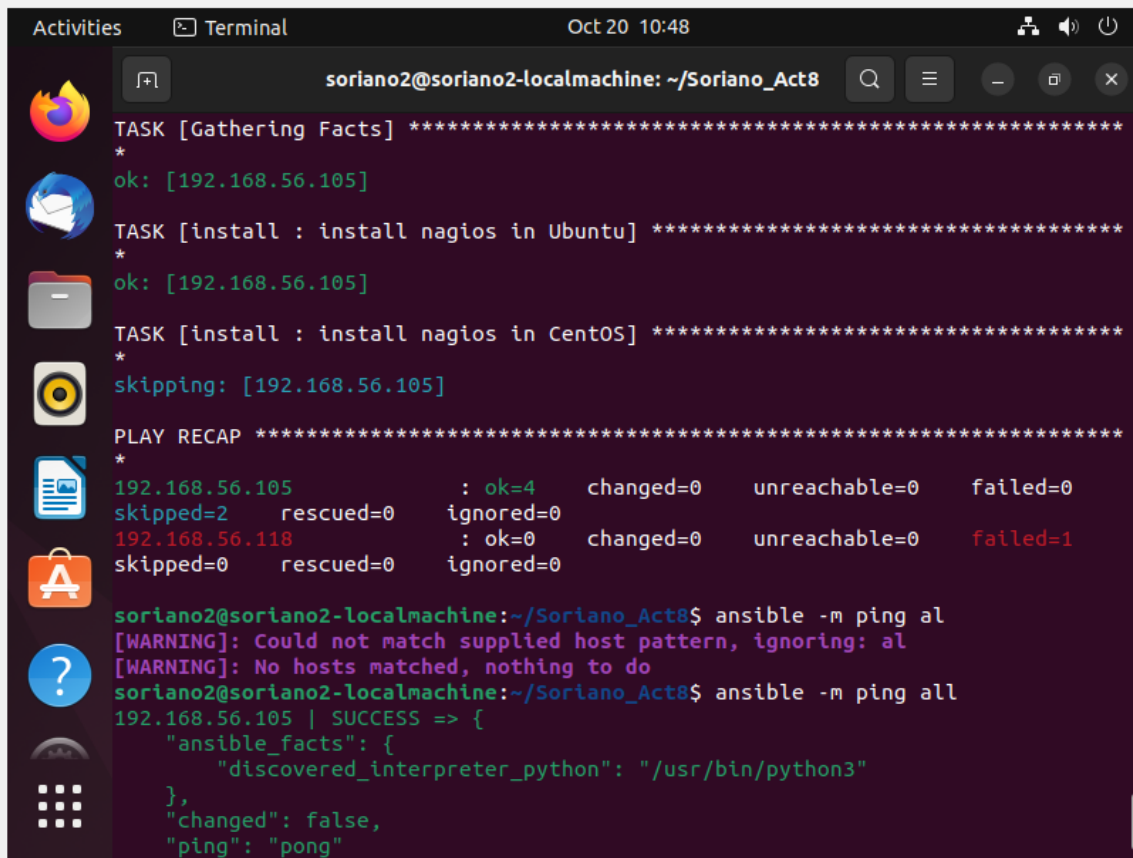
PLAY [all] *****
*
TASK [Gathering Facts] *****
*
ok: [192.168.56.105]
fatal: [192.168.56.118]: FAILED! => {"msg": "Incorrect sudo password"}

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

TASK [install updates (Ubuntu)] *****
*
ok: [192.168.56.105]

PLAY [all] *****
*
TASK [Gathering Facts] *****
*
ok: [192.168.56.105]

TASK [install : install nagios in Ubuntu] *****
```

The terminal window displays the output of an Ansible playbook. It shows tasks for gathering facts, installing Nagios on Ubuntu, and installing Nagios on CentOS. A 'PLAY RECAP' section summarizes the results for two hosts: 192.168.56.105 and 192.168.56.118. The results for 192.168.56.105 are: ok=4, changed=0, unreachable=0, failed=0, skipped=2, rescued=0, ignored=0. The results for 192.168.56.118 are: ok=0, changed=0, unreachable=0, failed=1, skipped=0, rescued=0, ignored=0. Below the recap, the user runs 'ansible -m ping al' and 'ansible -m ping all'. The first command results in a warning that no hosts matched the pattern 'al'. The second command results in a success for 192.168.56.105, showing the 'ansible_facts' and 'ping' results.

```
Activities Terminal Oct 20 10:48
soriano2@soriano2-localmachine: ~/Soriano_Act8

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

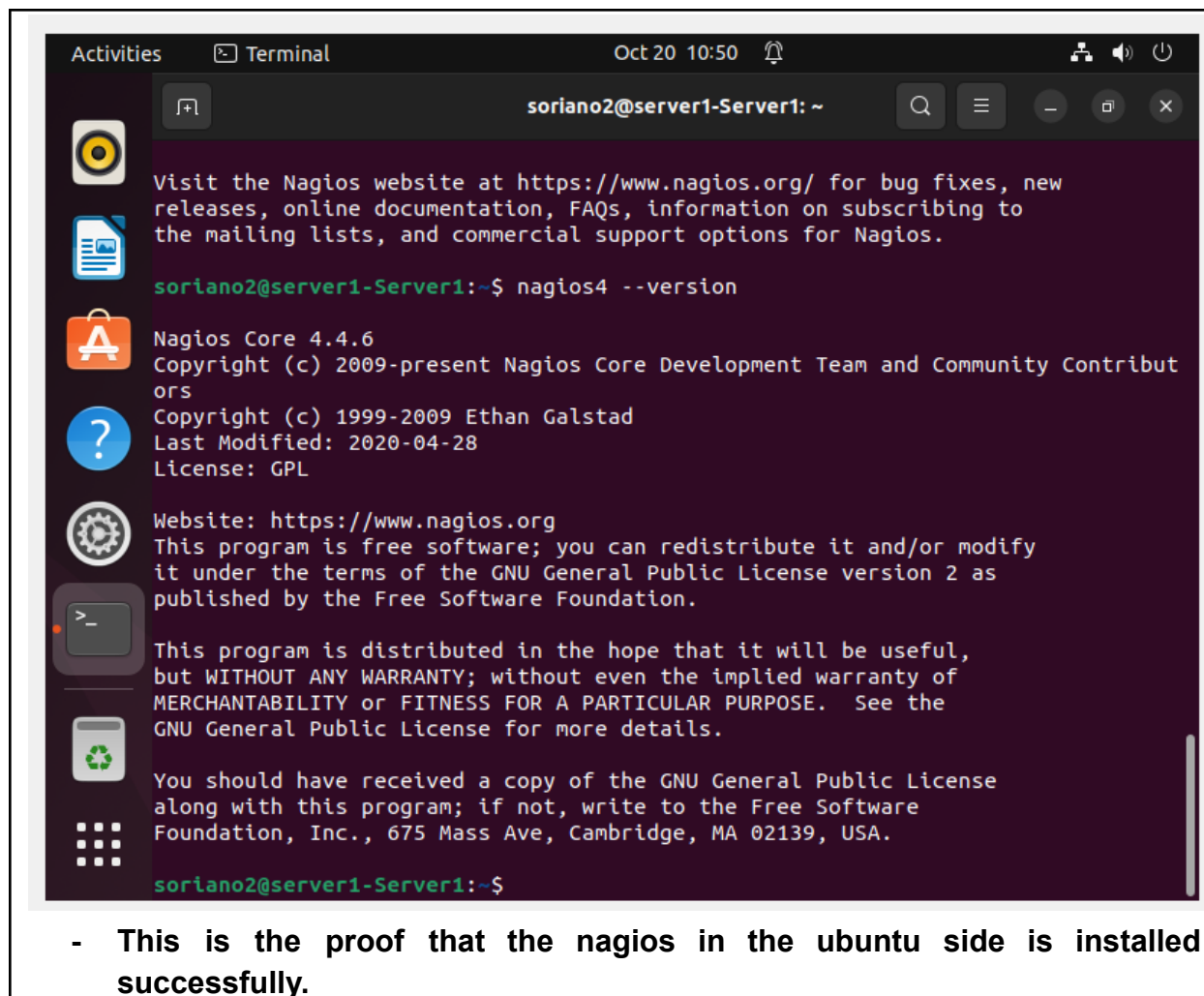
TASK [install : install nagios in Ubuntu] *****
*
ok: [192.168.56.105]

TASK [install : install nagios in CentOS] *****
*
skipping: [192.168.56.105]

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

soriano2@soriano2-localmachine:~/Soriano_Act8$ ansible -m ping al
[WARNING]: Could not match supplied host pattern, ignoring: al
[WARNING]: No hosts matched, nothing to do
soriano2@soriano2-localmachine:~/Soriano_Act8$ ansible -m ping all
192.168.56.105 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

- This shows that when running the playbook for the side of ubuntu, there is one error enco





soriano2@soriano2-localmachine: ~/Soriano_Act8



```
skipping: [192.168.56.118]
ok: [192.168.56.105]
```



```
PLAY [all] *****
*
```



```
TASK [Gathering Facts] *****
*
ok: [192.168.56.118]
ok: [192.168.56.105]
```



```
TASK [install : install nagios in Ubuntu] *****
*
skipping: [192.168.56.118]
ok: [192.168.56.105]
```



```
TASK [install : install nagios in CentOS] *****
*
skipping: [192.168.56.105]
changed: [192.168.56.118]
```



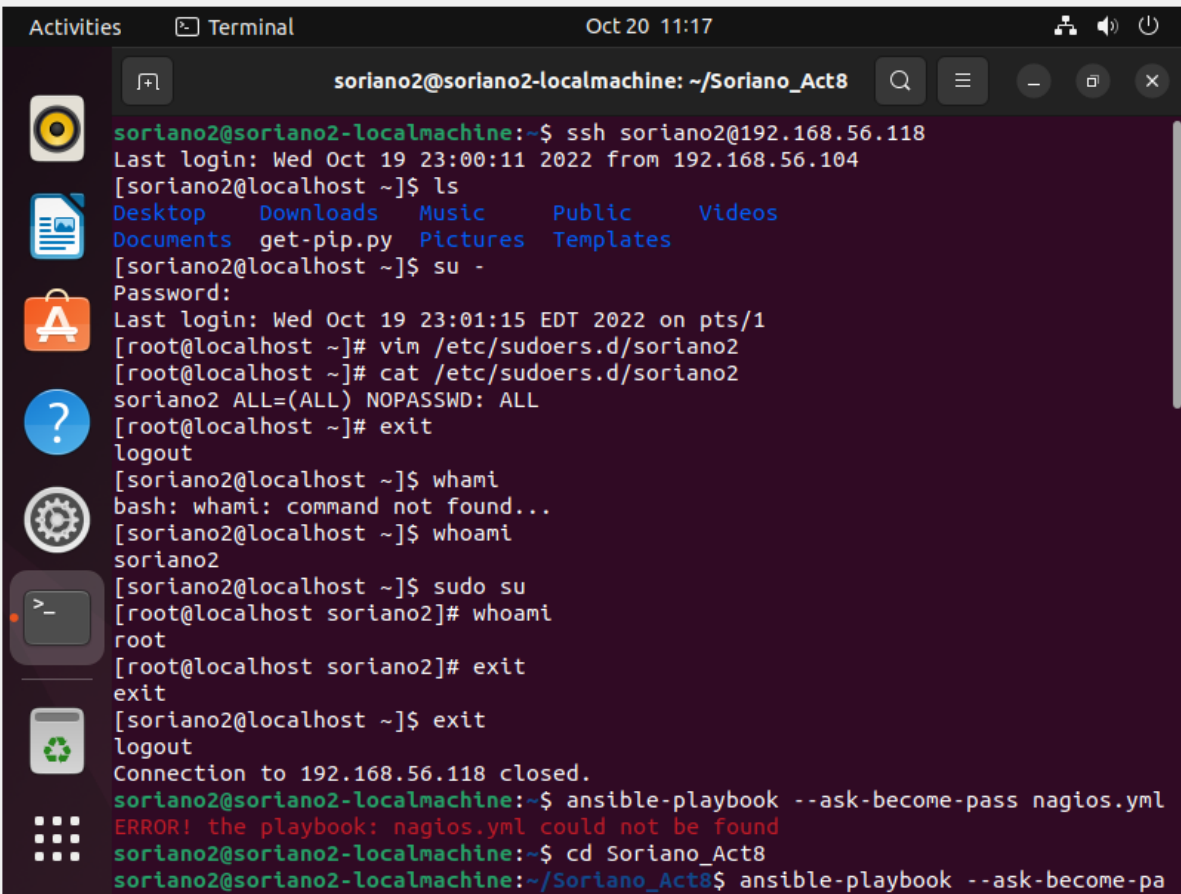
```
PLAY RECAP *****
*
```



```
192.168.56.105 : ok=4    changed=0    unreachable=0    failed=0
skipped=2     rescued=0    ignored=0
192.168.56.118 : ok=4    changed=1    unreachable=0    failed=0
skipped=2     rescued=0    ignored=0
```



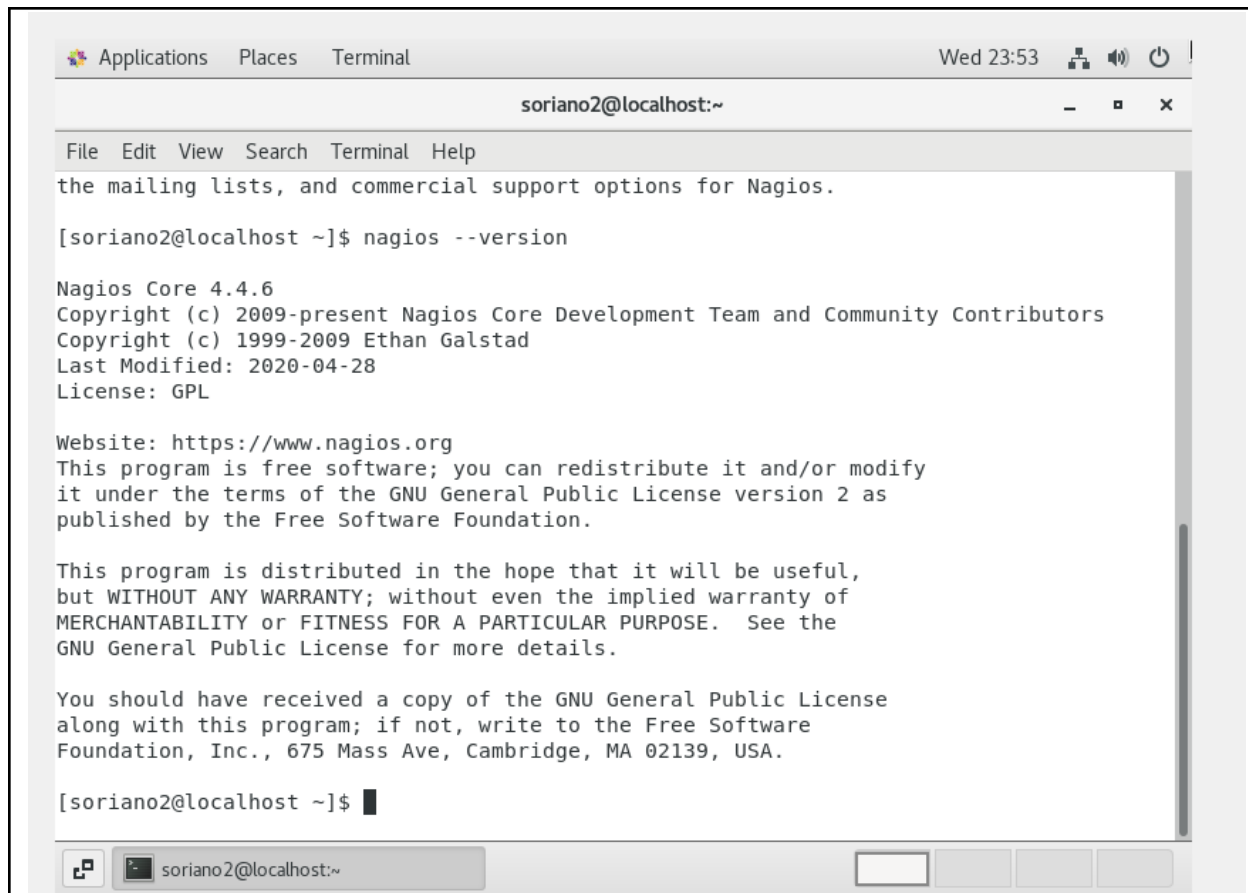
```
soriano2@soriano2-localmachine:~/Soriano_Act8$
```



The image shows a terminal window titled "soriano2@soriano2-localmachine: ~/Soriano_Act8". The user starts by SSHing into a CentOS machine at 192.168.56.118. They then list files, switch to root using 'su', and edit the sudoers file to grant root access to the 'soriano2' user. After exiting root, they use 'sudo su' to become root again. Finally, they run an Ansible playbook, which initially fails with an error, but then succeeds after they change to the 'Soriano_Act8' directory.

```
soriano2@soriano2-localmachine:~$ ssh soriano2@192.168.56.118
Last login: Wed Oct 19 23:00:11 2022 from 192.168.56.104
[soriano2@localhost ~]$ ls
Desktop  Downloads  Music      Public     Videos
Documents get-pip.py Pictures  Templates
[soriano2@localhost ~]$ su -
Password:
Last login: Wed Oct 19 23:01:15 EDT 2022 on pts/1
[root@localhost ~]# vim /etc/sudoers.d/soriano2
[root@localhost ~]# cat /etc/sudoers.d/soriano2
soriano2 ALL=(ALL) NOPASSWD: ALL
[root@localhost ~]# exit
logout
[soriano2@localhost ~]$ whami
bash: whami: command not found...
[soriano2@localhost ~]$ whoami
soriano2
[soriano2@localhost ~]$ sudo su
[root@localhost soriano2]# whoami
root
[root@localhost soriano2]# exit
exit
[soriano2@localhost ~]$ exit
logout
Connection to 192.168.56.118 closed.
soriano2@soriano2-localmachine:~$ ansible-playbook --ask-become-pass nagios.yml
ERROR! the playbook: nagios.yml could not be found
soriano2@soriano2-localmachine:~$ cd Soriano_Act8
soriano2@soriano2-localmachine:~/Soriano_Act8$ ansible-playbook --ask-become-pa
```

- In this image, the error has been resolved. I resolved this problem/error by accessing the root of the CentOS via ssh from the main ubuntu server. The method is shown above.

A screenshot of a Linux terminal window. The window title is 'soriano2@localhost:~'. The terminal shows the output of the command 'nagios --version'. The output includes: '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', and a statement about the GNU General Public License. The terminal also shows the prompt '[soriano2@localhost ~]\$' and a cursor. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The top status bar shows 'Wed 23:53' and system icons. The bottom status bar shows the terminal title and some window controls.

```
soriano2@localhost:~  
File Edit View Search Terminal Help  
the mailing lists, and commercial support options for Nagios.  
  
[soriano2@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.  
  
[soriano2@localhost ~]$
```

- **This shows the successful test opening and installation of nagios in the centos.**
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.)
 - **Steps, methods, error solving and troubleshooting are stated above with screenshots.**
 3. Show an output of the installed Nagios for both Ubuntu and CentOS.
 - **Outputs are shown above by step.**
 4. Make sure to create a new repository in GitHub for this activity.

SCREENSHOT:

qgsoriano1 / Soriano_Act8F Public

Code Issues Pull requests Actions Projects Wiki Security Insights

main Go to file Add file Code About

File	Commit	Time
final	FINAL COMMIT OCT 20, 2022 AT 11:44PM	24 seconds ago
roles/install/tasks	FINAL COMMIT OCT 20, 2022 AT 11:44PM	24 seconds ago
README.md	Initial commit	4 hours ago

README.md

- This shows the successful git commit and push of the files that has been done and successfully performed in this activity.

Git repo link/url: https://github.com/qgsoriano1/Soriano_Act8F.git
git@github.com:qgsoriano1/Soriano_Act8F.git
gh repo clone qgsoriano1/Soriano_Act8F

Reflections:

Answer the following:

- What are the benefits of having an availability monitoring tool?
 - Having a monitoring tool available helps our servers and devices avoid outages. You have the visibility you need to stay ahead of potential problems thanks to network monitoring. Network monitoring software assists you in locating outages that might result in bottlenecks by displaying real-time network performance data in an intuitive interface. Additionally, problems are resolved more quickly. When things are bad, time is money. For time-pressed network professionals, network monitoring makes problem-solving simpler and quicker. A network monitoring tool can offer that first level of security and identify security threats. The biggest advantage is a picture of what typical performance for your organization looks like, which makes it simple to identify anything out of the ordinary.

Conclusions:

In this activity, it is a bit stressful because we have no instructional resources at all from this activity, we have to research on our own and do the activity

appropriately. I have encountered a lot of errors during this activity, including the unreachable status of the centos from the main ubuntu server. This has been resolved by accessing the centos through ssh and creating a file.