Name: John Renzo L. Mendoza	**Date Performed:** October 17, 2023
Course/Section: CPE31S5	**Date Submitted:** October 21, 2023
Instructor: Engr. Roman Richard	**Semester and SY:**

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.

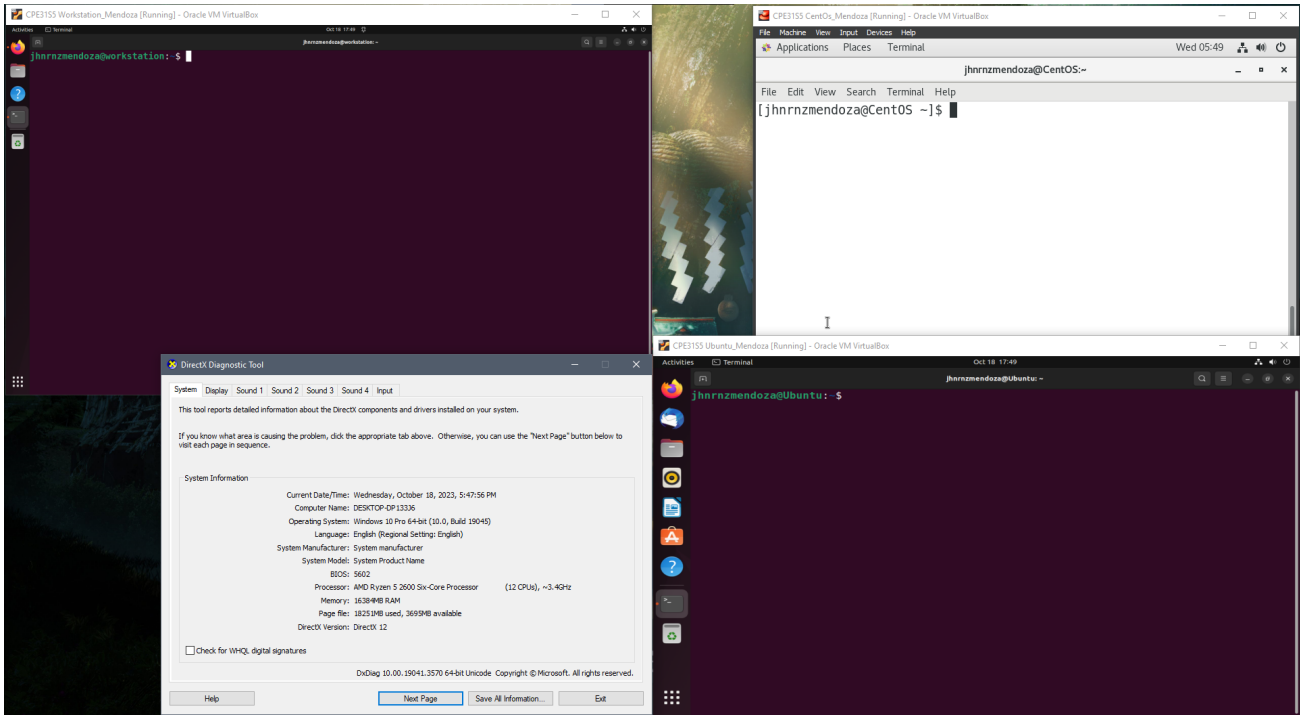
3. Tasks

- Create a playbook that installs Nagios in both Ubuntu and CentOS. Apply the concept of creating roles.
- 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.)
- Show an output of the installed Nagios for both Ubuntu and CentOS.
- Make sure to create a new repository in GitHub for this activity.

4. **Output** (screenshots and explanations)

Initial Procedures:

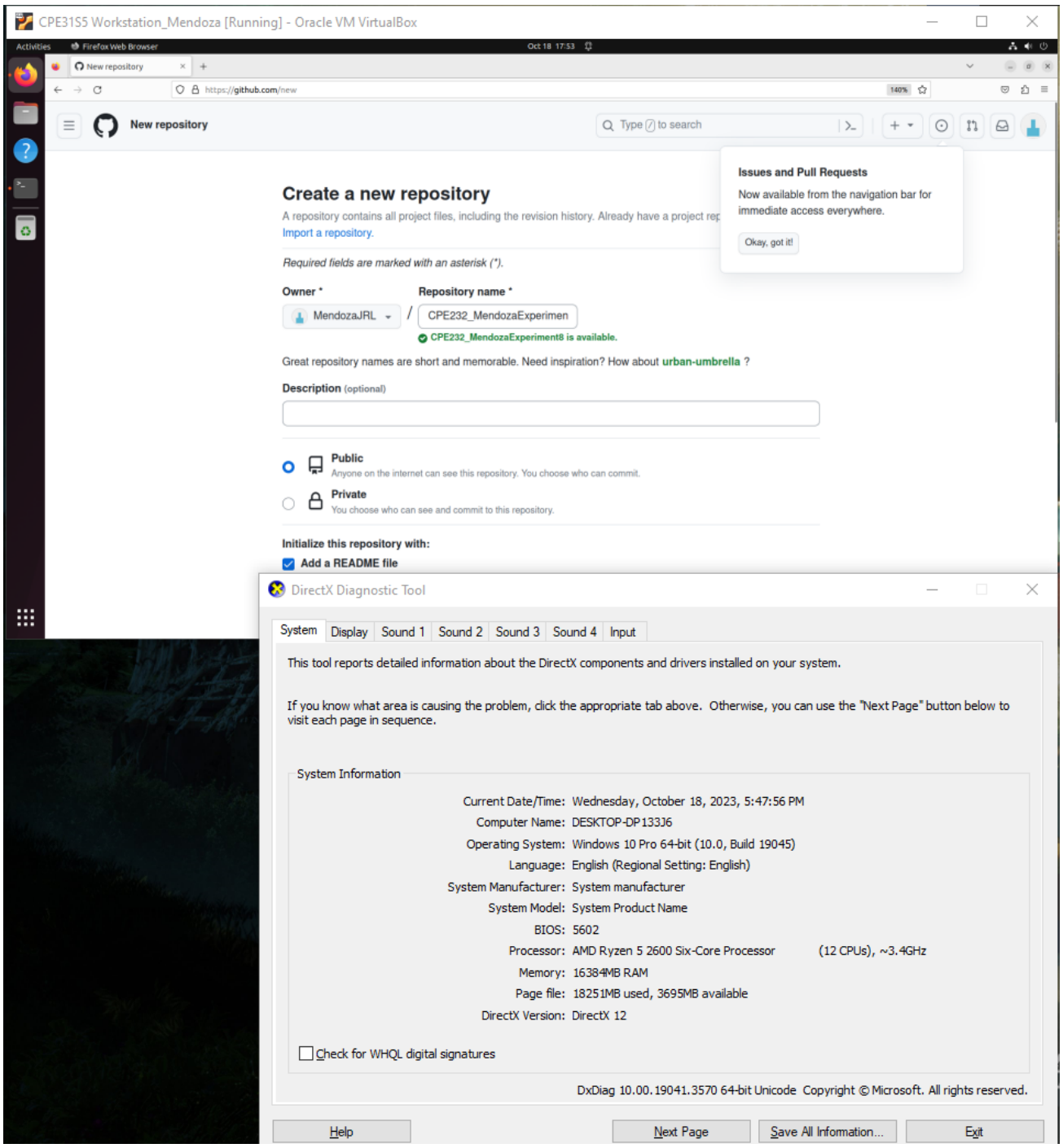
Setting up the Control Node (Ubuntu), and 2 Managed Nodes (Ubuntu and CentOS).



Observation:

- In this experiment, I have chosen to use the previously created remote servers which are Ubuntu and CentOS.

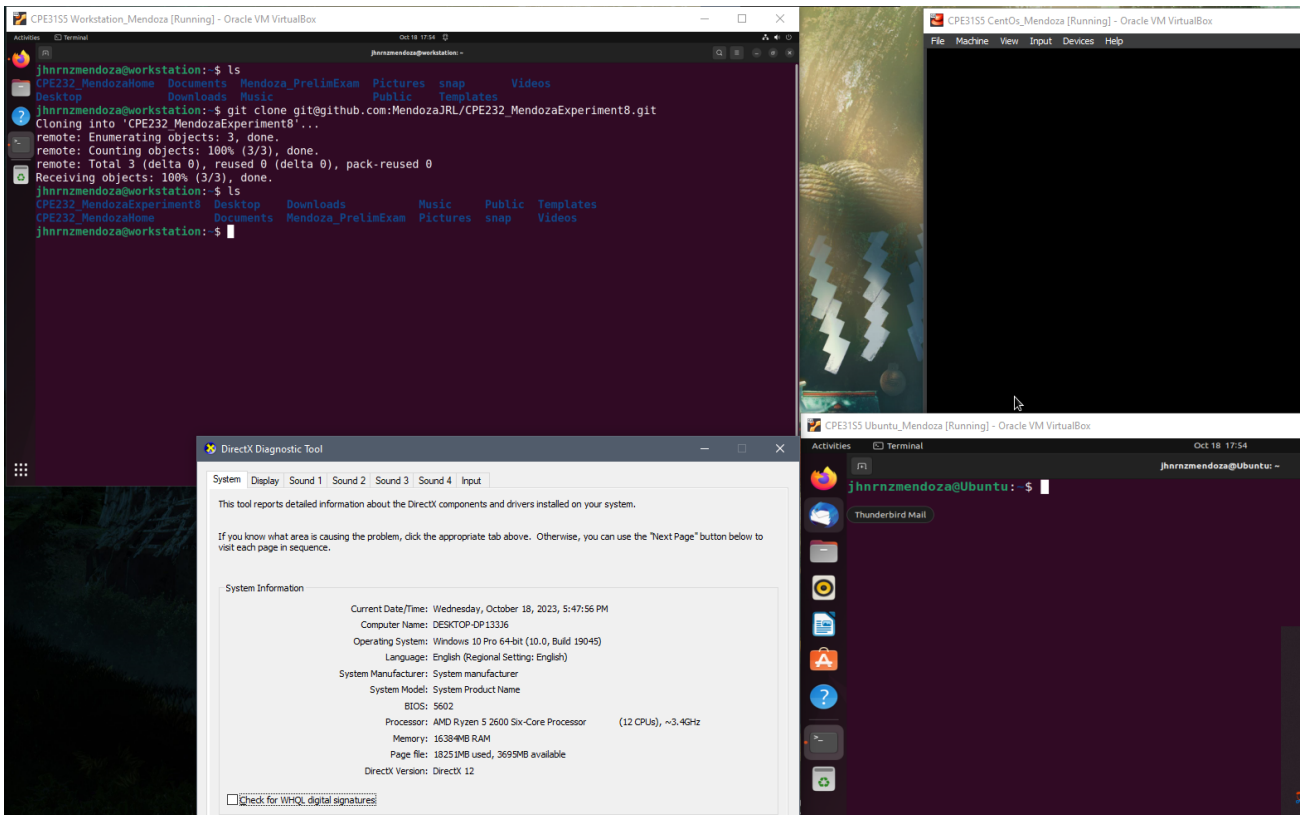
Creation of a new repository to be used on this activity, as required.



Observation:

- I have created a new repository CPE232_MendozaExperiment8 by making one in the GitHub website.

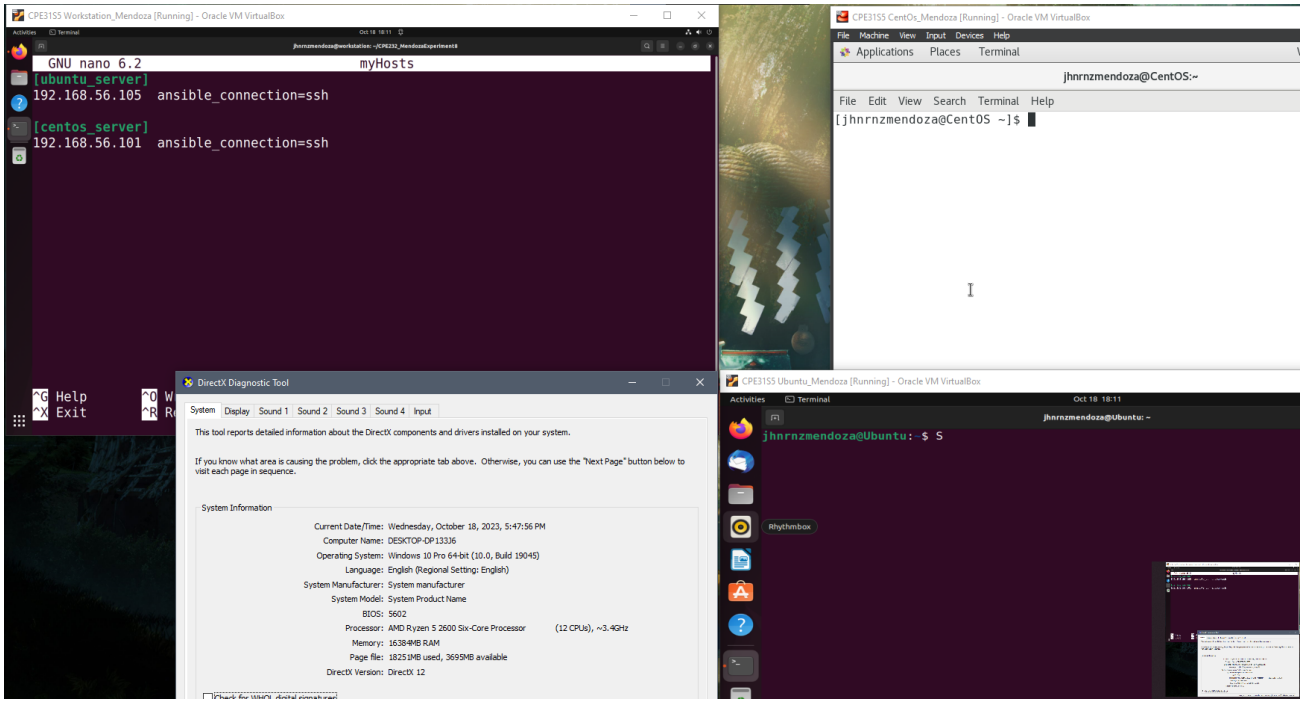
Cloning the created GitHub repository to the Control Node.



Observation:

- Using the `git clone` command, I have imported the created repository to the local control node.

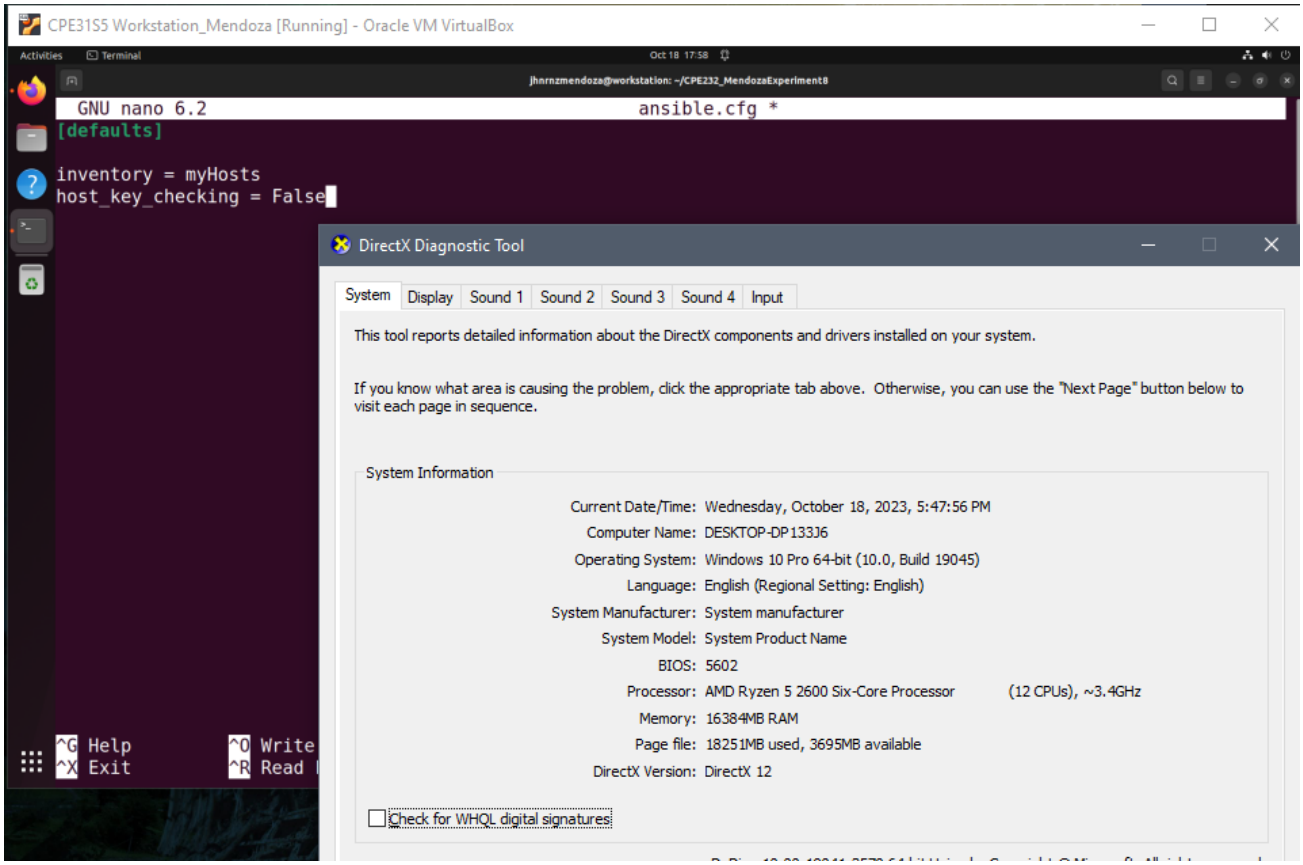
Creation of the inventory file “`myHosts`” under the cloned repository.



Observation:

- I have created `myHosts` as my inventory file which stores the ip addresses of the managed nodes.

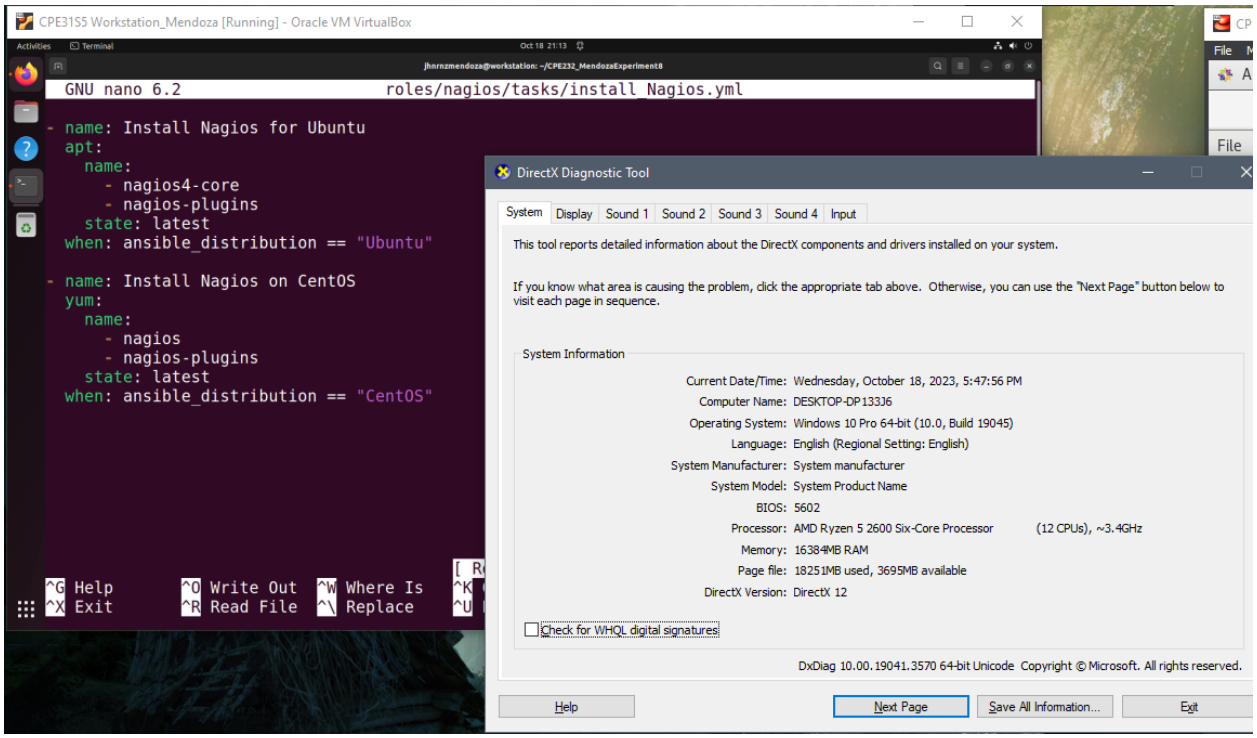
Creation of the ansible configuration file `ansible.cfg`.



Observation:

- I have also created ansible configuration file that stores the configuration needed for running ansible.

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



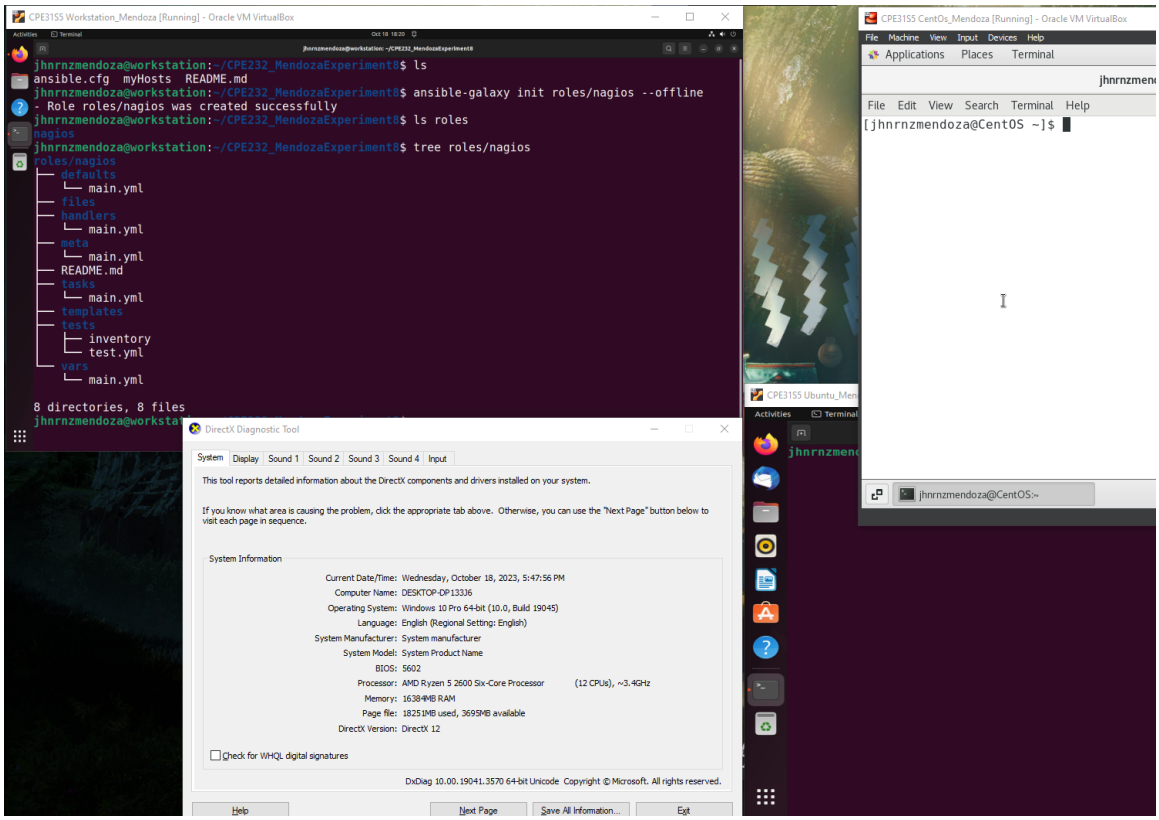
Observation:

- This playbook is created under the role “nagios’ which I have created to be the ansible roles for this activity. Basically, the playbook will install Nagios on Ubuntu and CentOS respectively. I have used conditional statements “when” in order to filter the managed nodes since the package manager for Ubuntu and CentOS is different.

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.)

- To achieve the created playbook on Step 1, here are the steps that I have done in order to have a working playbook with ansible roles implementation.

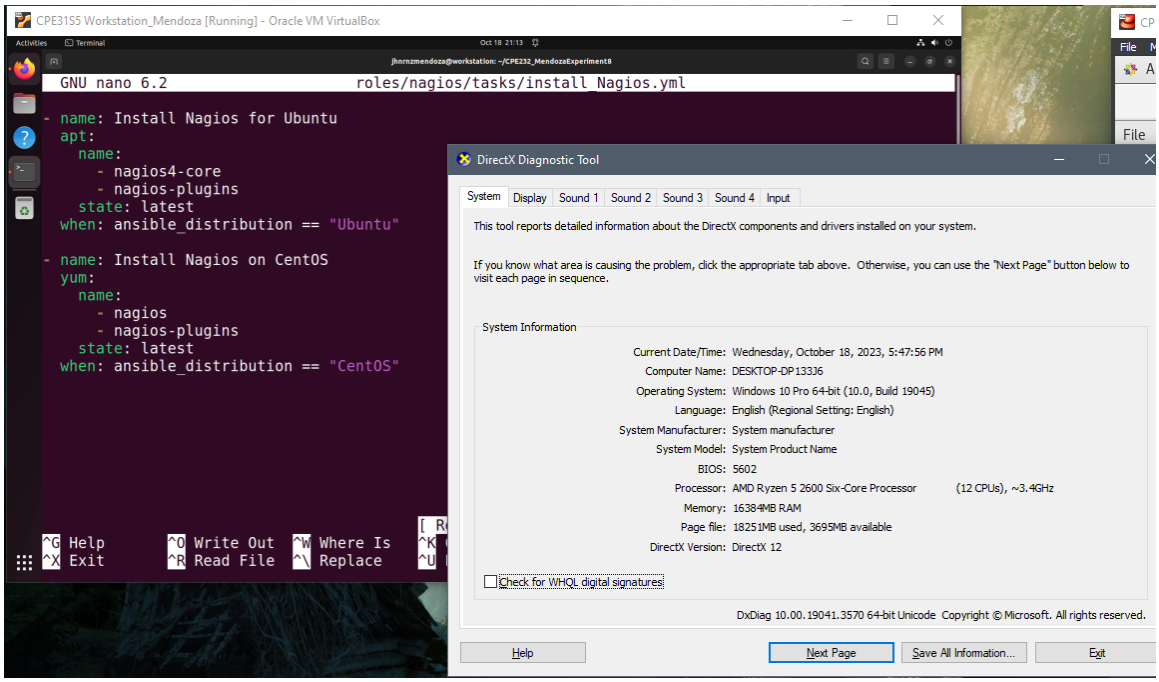
a. Creation of ansible roles to be implemented in this experiment using the `ansible_galaxy` command.



Observation:

- To create the ansible roles, I have used the command `ansible-galaxy` in order to create the directory of the role which is "`nagios`". As observed, the directory has been successfully created including the `main.yml` files.

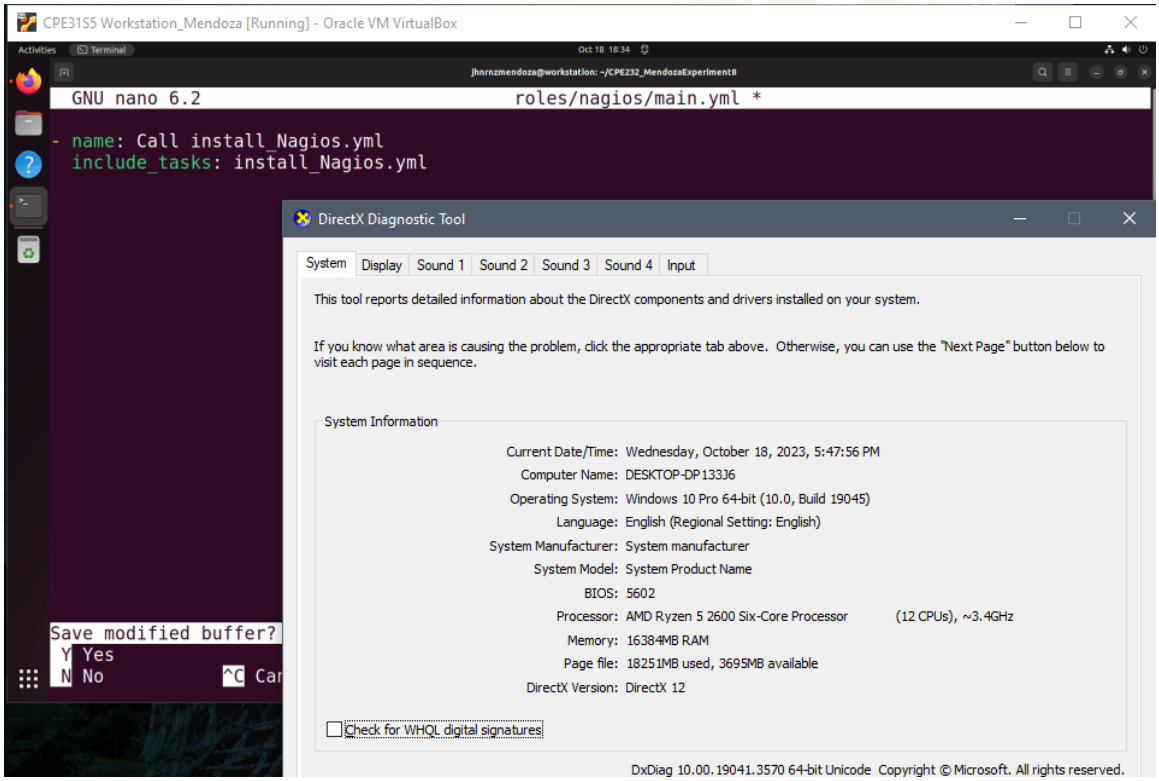
b. Creation of the playbook file that installs the Nagios on the roles.



Observation:

- Under the directory `roles/nagios/tasks/` I have created a playbook file `install_Nagios.yml` which has the commands needed to install the `Nagios` service on both Ubuntu and CentOS.

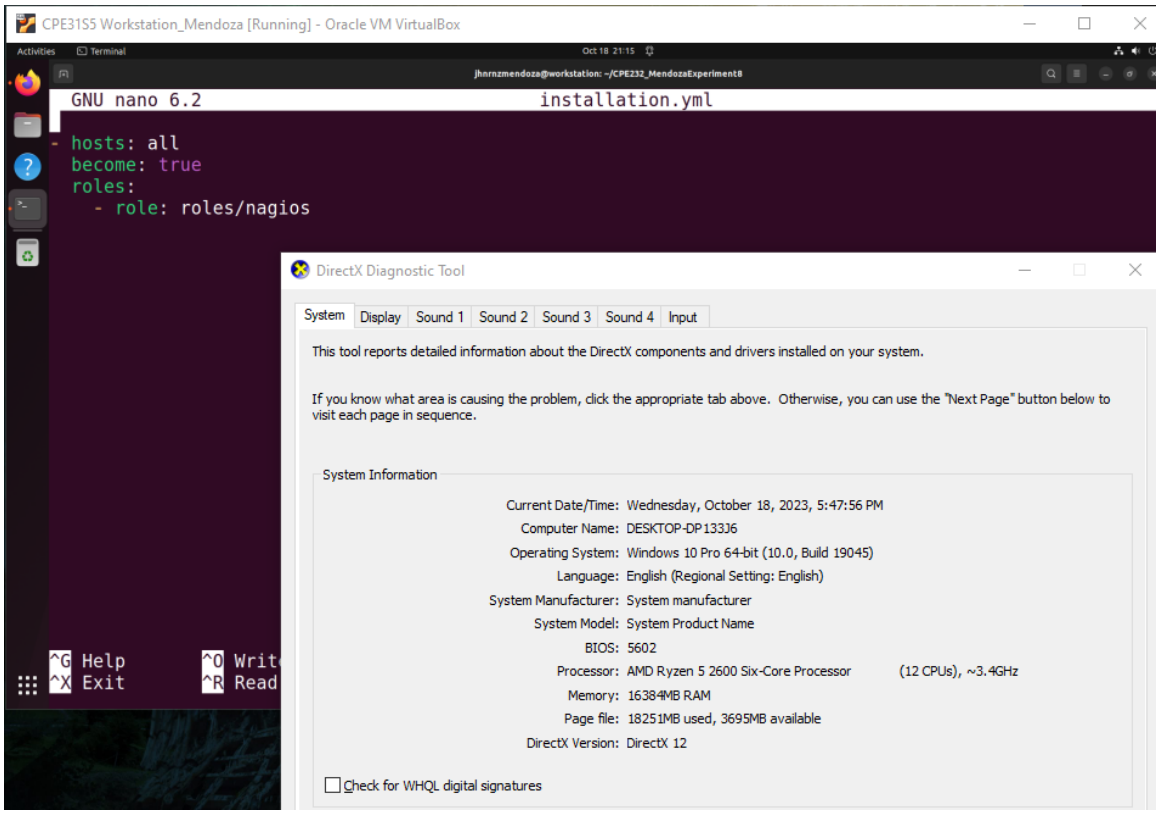
c. Configuring the `main.yml` to execute the created playbook.



Observation:

- Similarly, I have configured the `main.yml` under the `roles/nagios/tasks/` in which I have added a command to call the playbook "`install_Nagios.yml`". Basically, the `main.yml` will execute the `install_Nagios.yml`.

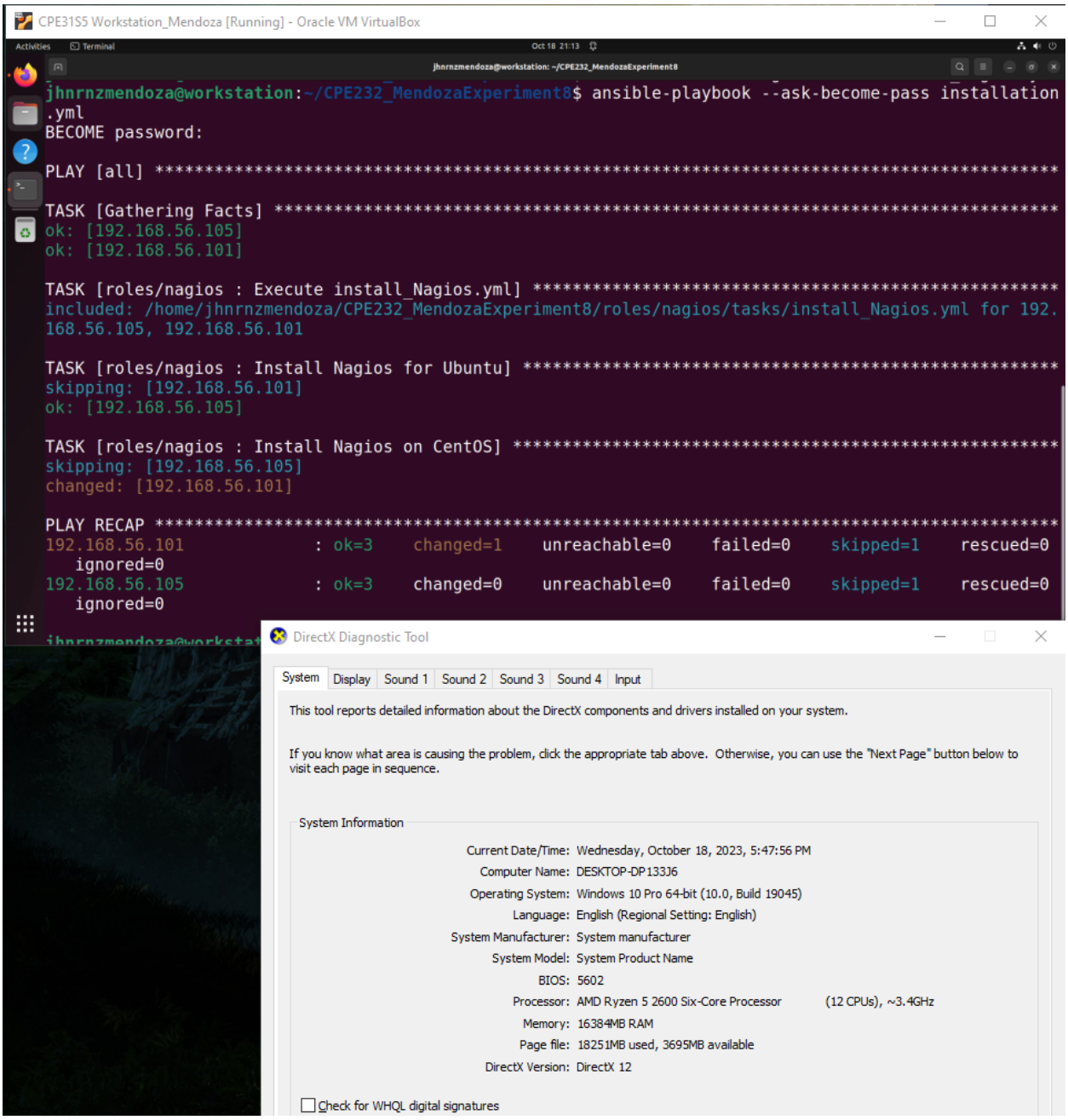
d. Creation of the playbook file that executes the roles.



Observation:

- Going back to our repository's directory `~/CPE232_MendozaExperiment8` I have created a playbook entitled "`installation.yml`" that will execute the role "nagios". This will essentially execute the tasks defined on the "nagios" role. In which case, it will run the `main.yml` and `install_Nagios.yml`

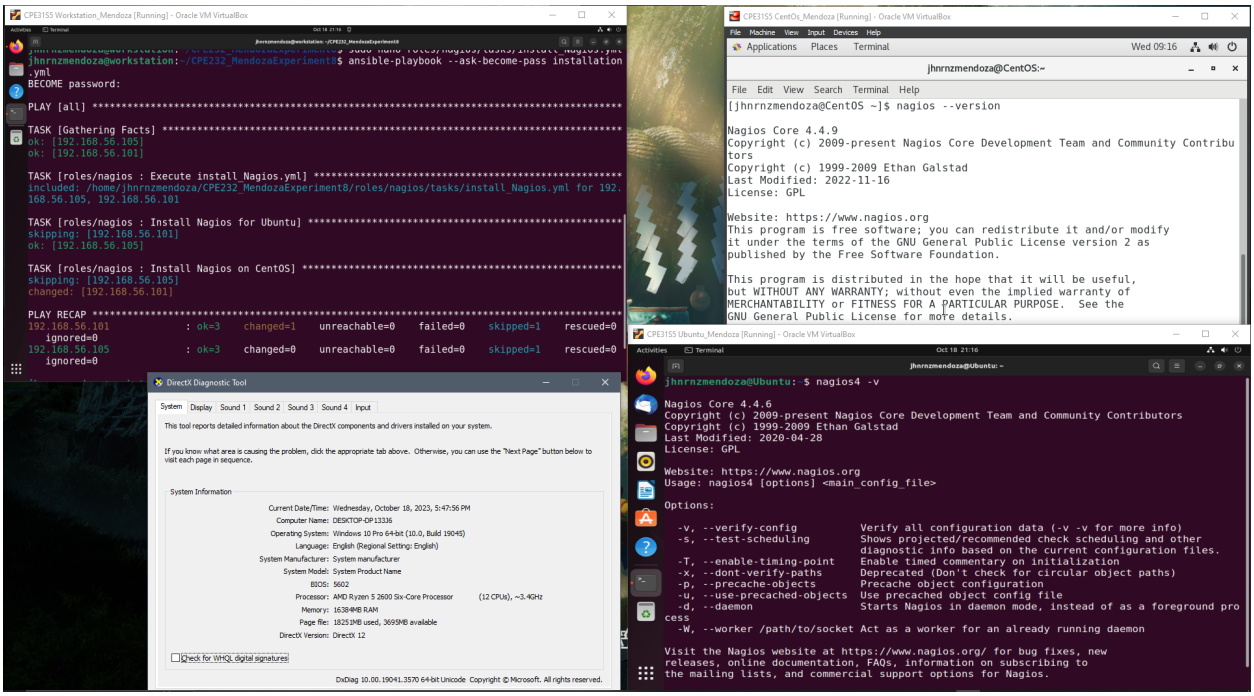
3. Show an output of the installed Nagios for both Ubuntu and CentOS.



Observation:

- After running the main playbook, we can observe that the playbook run is successful and has a login which has changed or configured both the Ubuntu and CentOS remote servers.

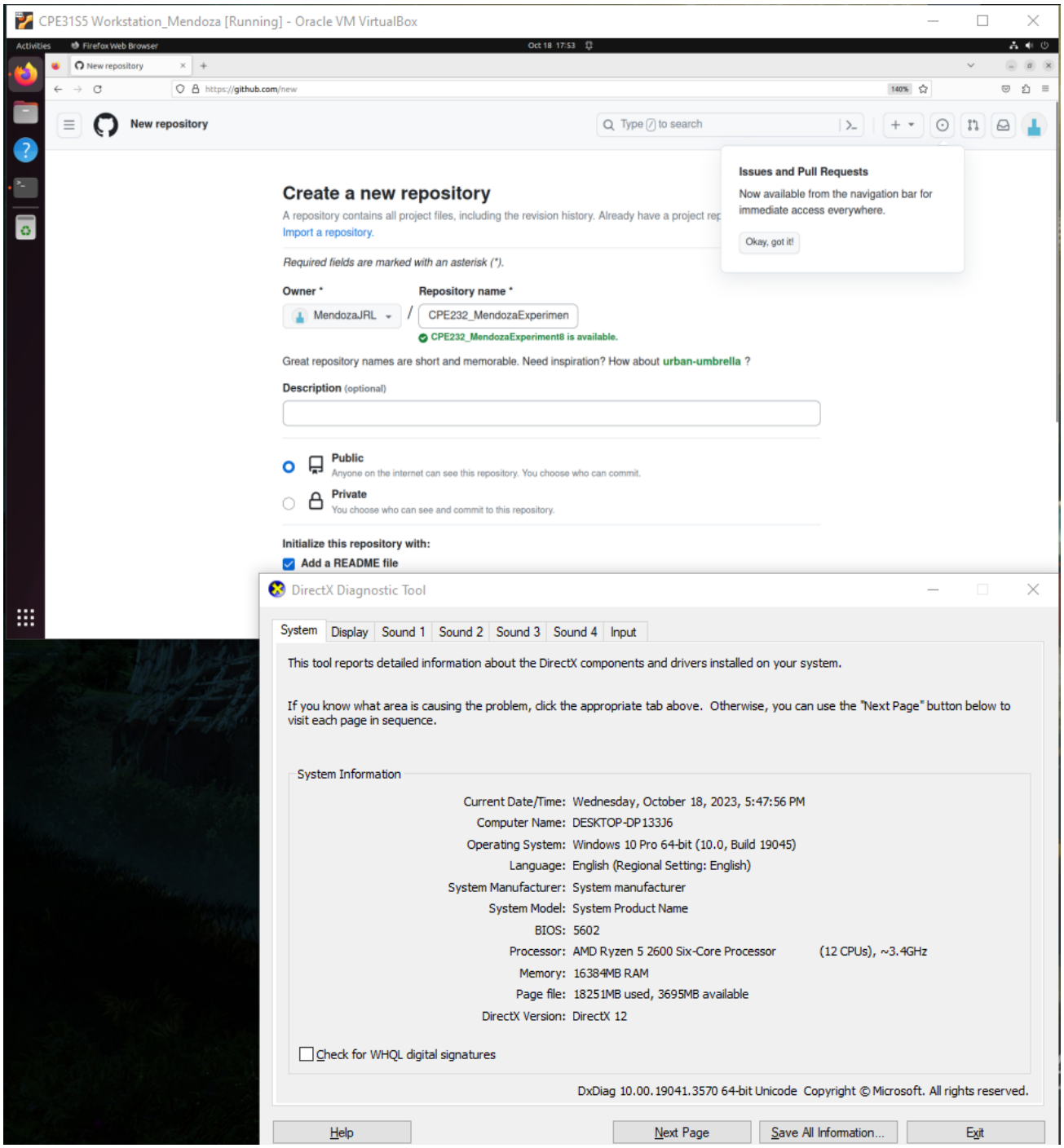
Verification: Ubuntu



Observation:

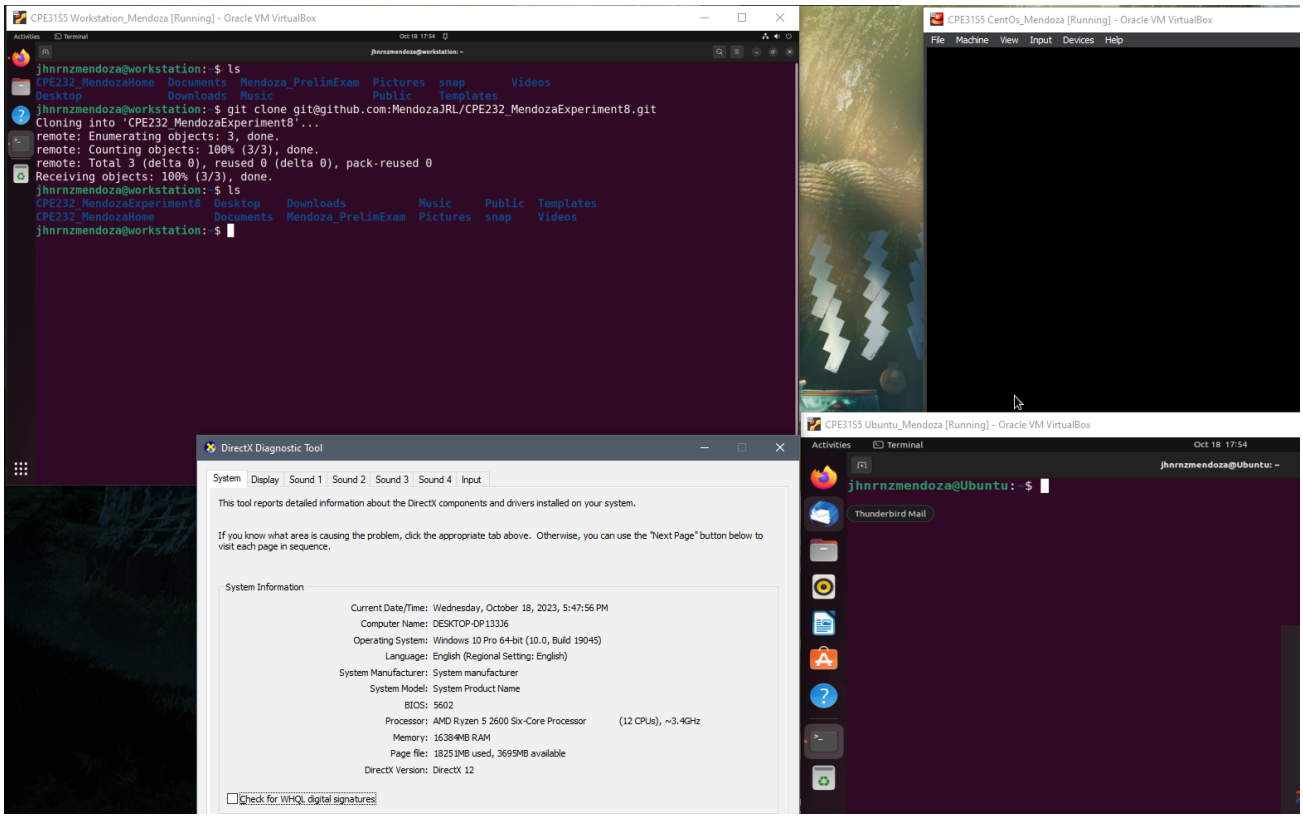
- To verify if it is successfully installed, I have used the Ubuntu remote server and CentOS remote server to check if the Nagios service is installed. As observed, we can see that the Nagios service is installed in both Ubuntu and CentOs remote servers using the commands `nagios -version`.

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



Observation:

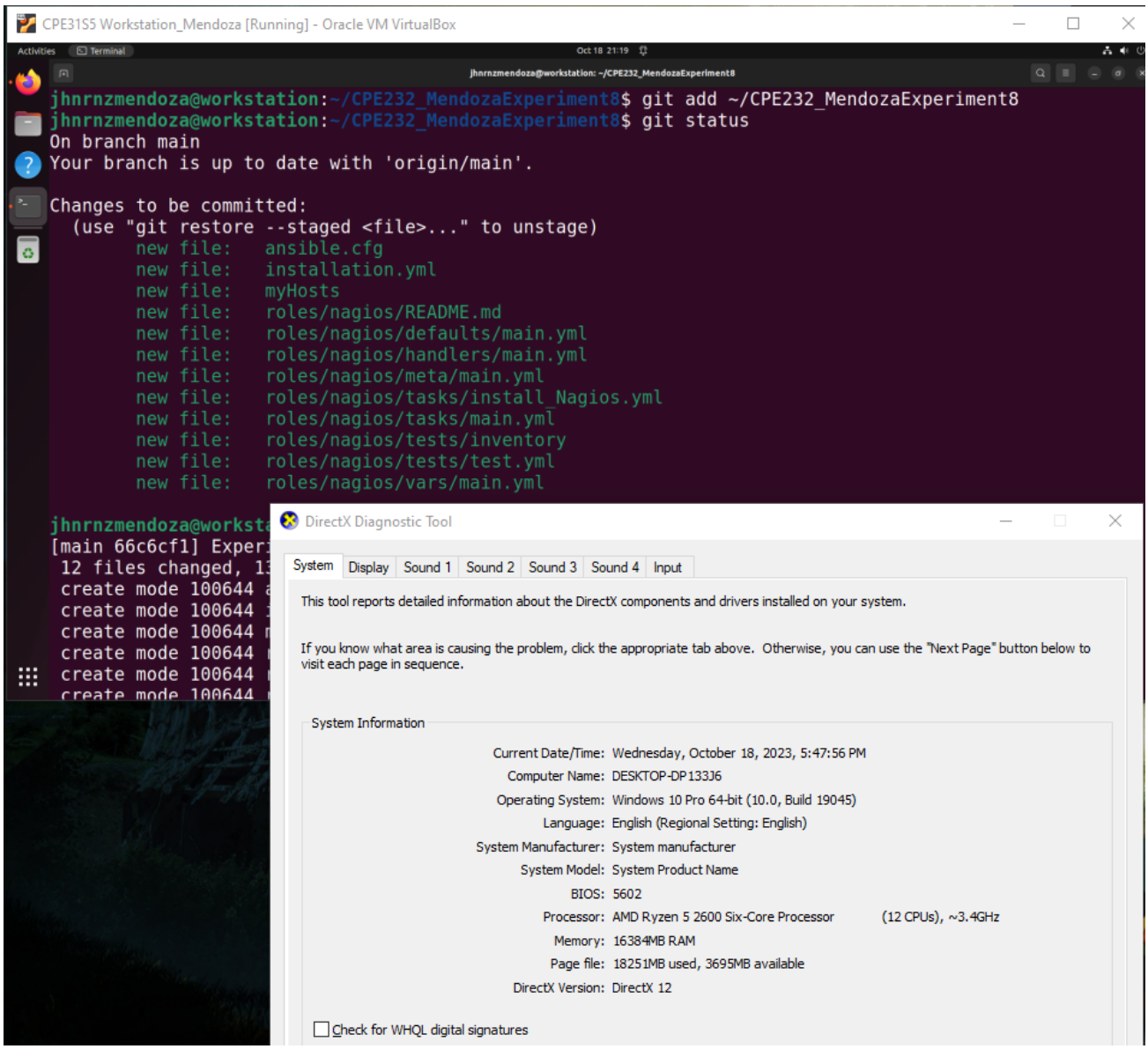
- This step should be done before doing the entirety of the procedure as we will use the repository and initialize it to do this activity.



Observation:

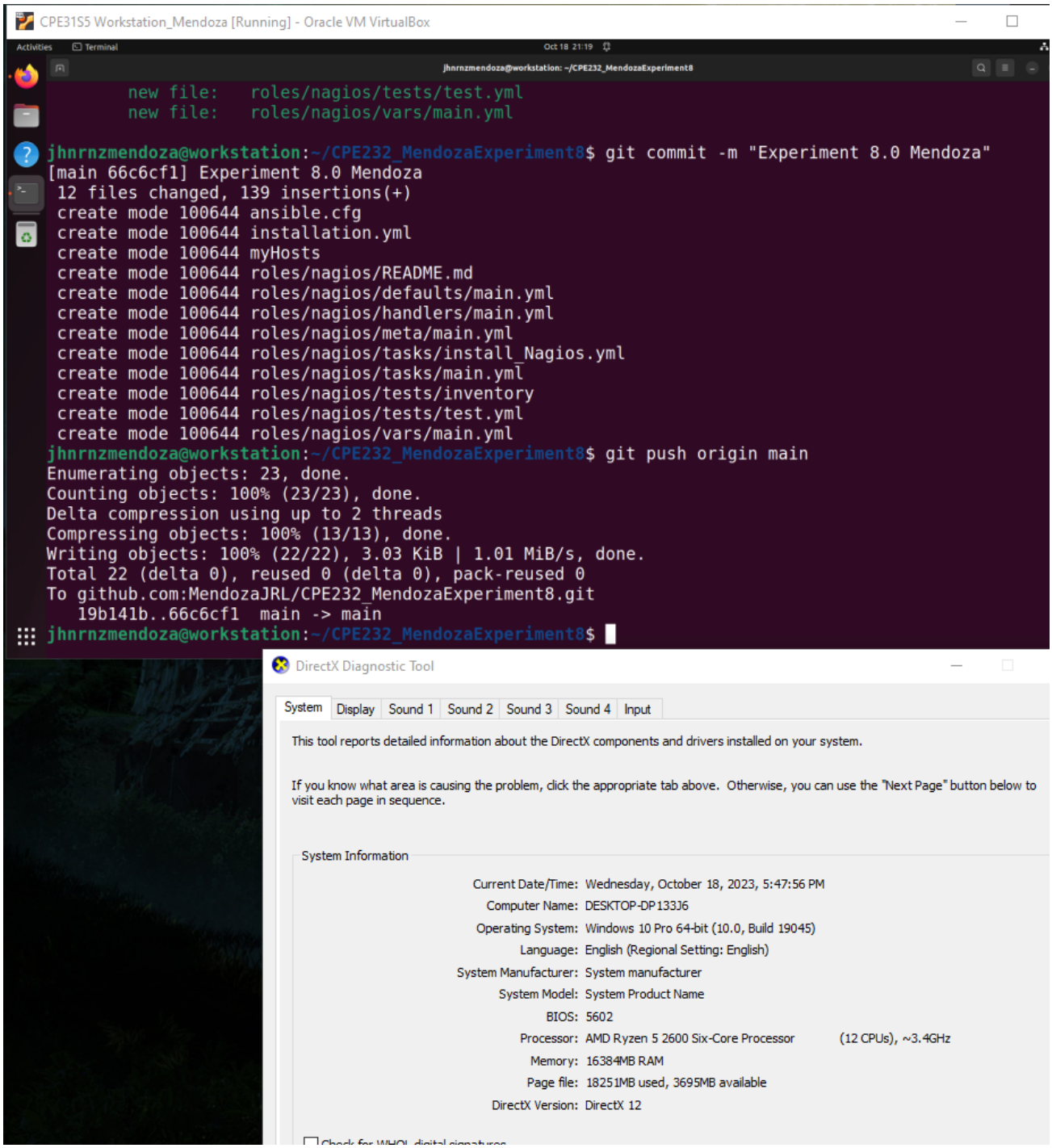
- I have used the `git clone` command to import the created GitHub repository to the local terminal of the control node.

Pushing the changes to my repository.



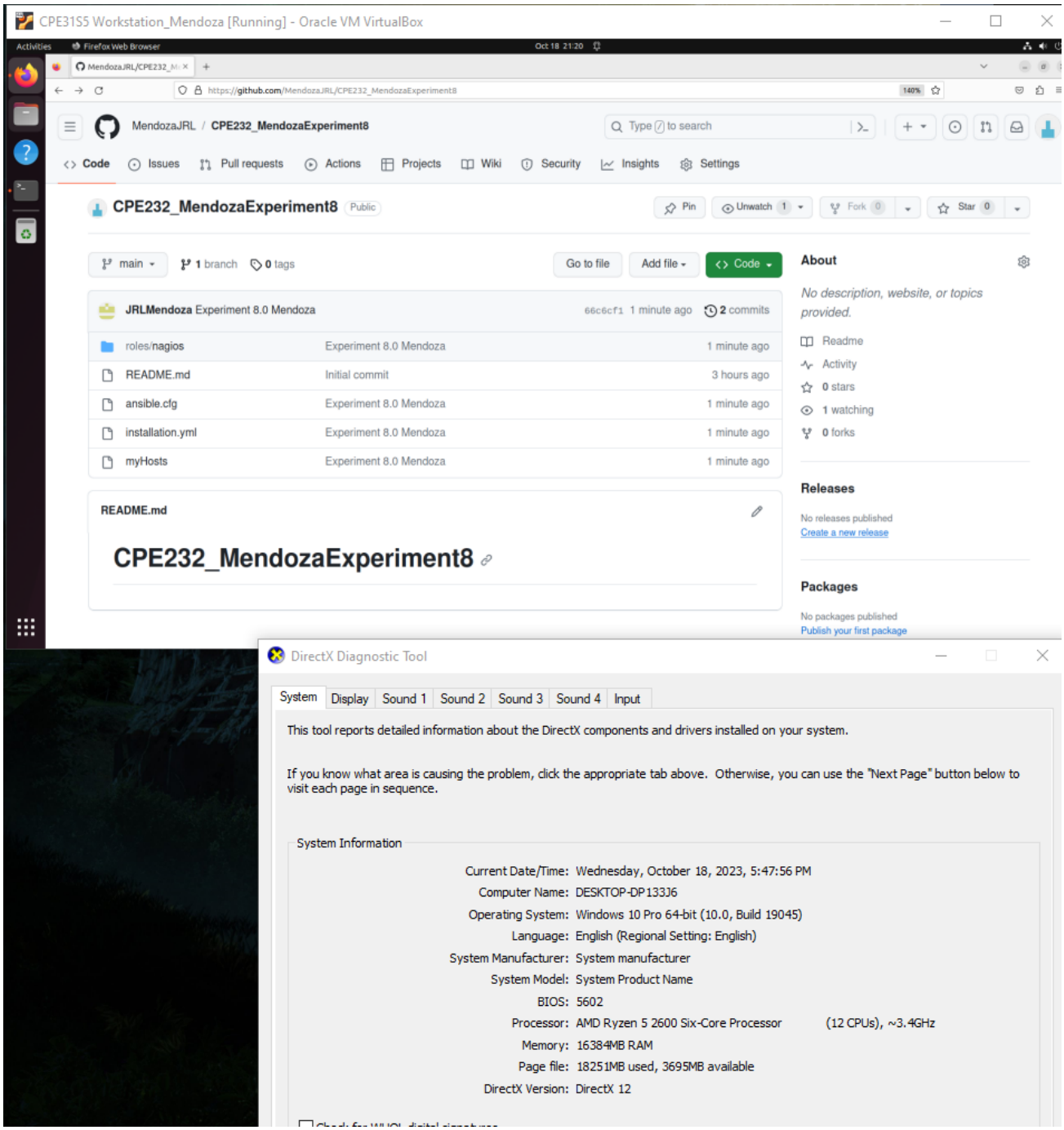
Observation:

- I have used `git add` and `git status` commands to show that the changes in the repository were added to the GitHub repository.



Observation:

- Using the git commit command, I have committed the changes with an ambiguous text tag, that describes the contents to be added.



Observation:

- By checking on the GitHub website, we can observe that the changes was added to our online repository.

Reflections:

Answer the following:

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

- Having monitoring services such as Nagios would help the system administrator consult the system if there are known errors that have been identified by the service. Moreover, monitoring services in a network architecture, also helps in monitoring the remote managed nodes by giving an error log to the control node that will inform the system administrator about the issue. In addition, the system administrator would be able to debug these errors ahead of time and also avoid having downtimes in the network.

Conclusions:

- In this activity, the students were able to implement the previously understood commands related to ansible. The students were also able to implement ansible roles to the experiment by utilizing the ansible-galaxy command. Moreover, the students have used the roles as a way to simplify the execution of the playbook. This is because the main executed playbook file will only call the tasks defined on the called ansible role. As observed, the created playbook file was able to perform the desired process of the experiment still taking the ansible roles into consideration.
- Furthermore, the students also used the previously learned git commands in order to do the experiment in a local repository which was then exported to the GitHub online repository in the web.