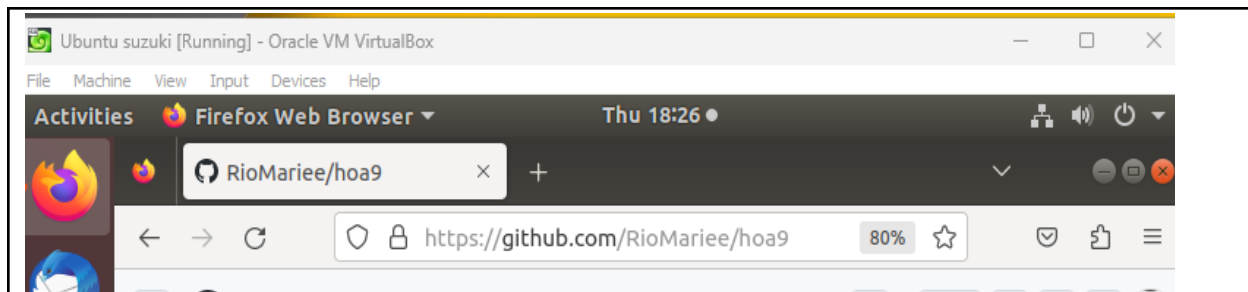
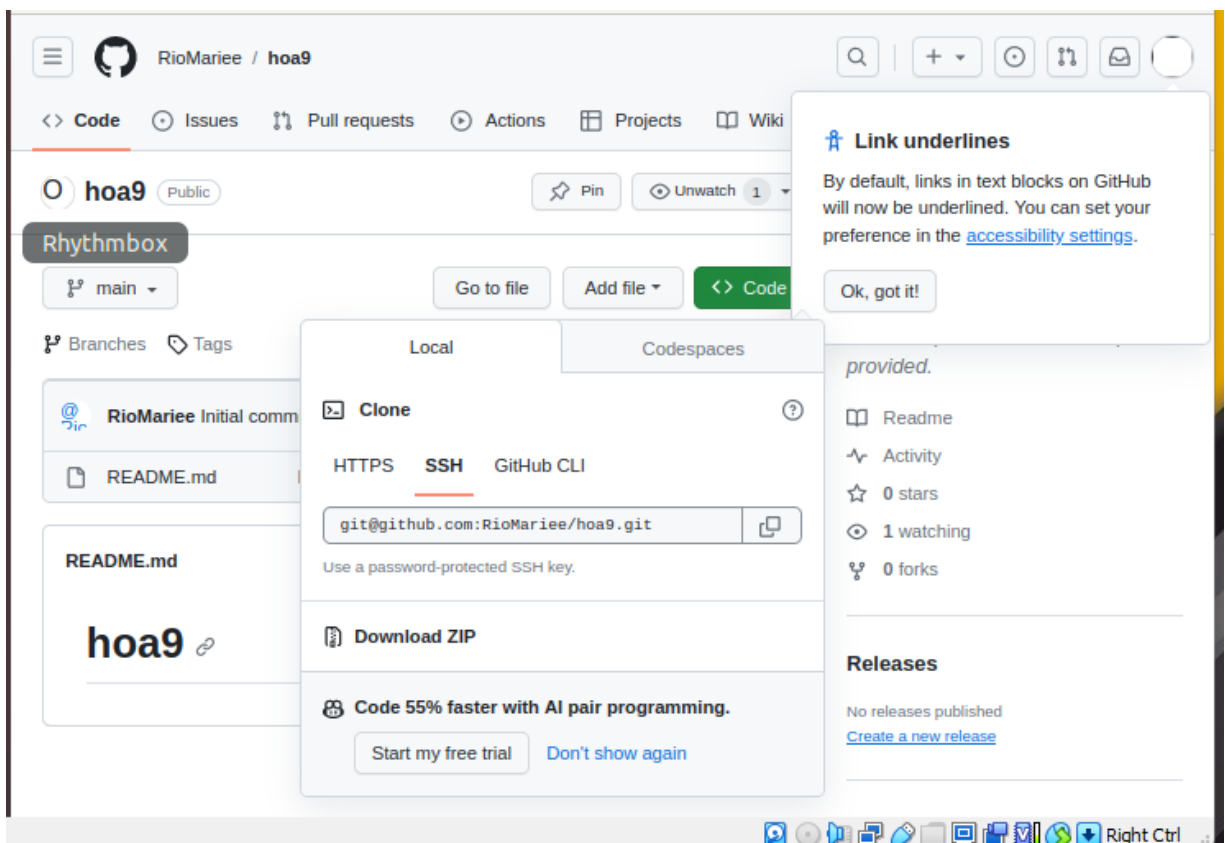


<b>Name:</b> Rio Marie G. Suzuki	<b>Date Performed:</b> 10/19/2023
<b>Course/Section:</b> CPE31S6	<b>Date Submitted:</b> 10/19/2023
<b>Instructor:</b> Dr. Jonathan Taylar	<b>Semester and SY:</b> 1st sem 2023-2024
<b>Activity 9: Install, Configure, and Manage Performance Monitoring tools</b>	
<b>1. Objectives</b>	
Create and design a workflow that installs, configure and manage enterprise performance tools using Ansible as an Infrastructure as Code (IaC) tool.	
<b>2. Discussion</b>	
<p>Performance monitoring is a type of monitoring tool that identifies current resource consumption of the workload, in this page we will discuss multiple performance monitoring tool.</p> <p><b>Prometheus</b></p> <p>Prometheus fundamentally stores all data as timeseries: streams of timestamped values belonging to the same metric and the same set of labeled dimensions. Besides stored time series, Prometheus may generate temporary derived time series as the result of queries. Source: <a href="#">Prometheus - Monitoring system &amp; time series database</a></p> <p><b>Cacti</b></p> <p>Cacti is a complete network graphing solution designed to harness the power of RRDTool's data storage and graphing functionality. Cacti provides a fast poller, advanced graph templating, multiple data acquisition methods, and user management features out of the box. All of this is wrapped in an intuitive, easy to use interface that makes sense for LAN-sized installations up to complex networks with thousands of devices. Source: <a href="#">Cacti® - The Complete RRDTool-based Graphing Solution</a></p>	
<b>3. Tasks</b>	
<ol style="list-style-type: none"> <li>1. Create a playbook that installs Prometheus in both Ubuntu and CentOS. Apply the concept of creating roles.</li> <li>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.)</li> <li>3. Show an output of the installed Prometheus for both Ubuntu and CentOS.</li> <li>4. Make sure to create a new repository in GitHub for this activity.</li> </ol>	
<b>4. Output (screenshots and explanations)</b>	
<p><b>Step 1.</b> First we open our virtual machines. and go to the main virtual machine's search engine then search github.com</p>	



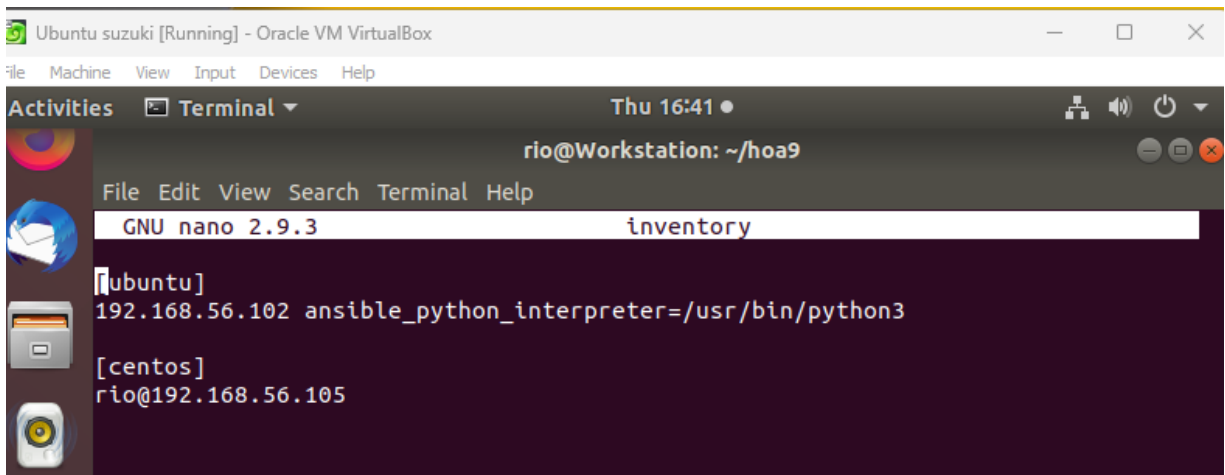
**Step 2.** Create a separate github repository for this activity and copy the link of the repository in the code option then select ssh and copy the url.



**Step 3.** After copying the url of the repository you can now clone it to the workstation's terminal using the command "git clone (paste the url here)"

```
rio@Workstation:~$ git clone git@github.com:RioMarilee/hoa9.git
Cloning into 'hoa9'...
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.
rio@Workstation:~$
```

**Step 4.** After cloning the git hub repository we need to first create the inventory for the hands-on activity. This contains the ip addresses of the ubuntu and centos that we used in the previous activities.

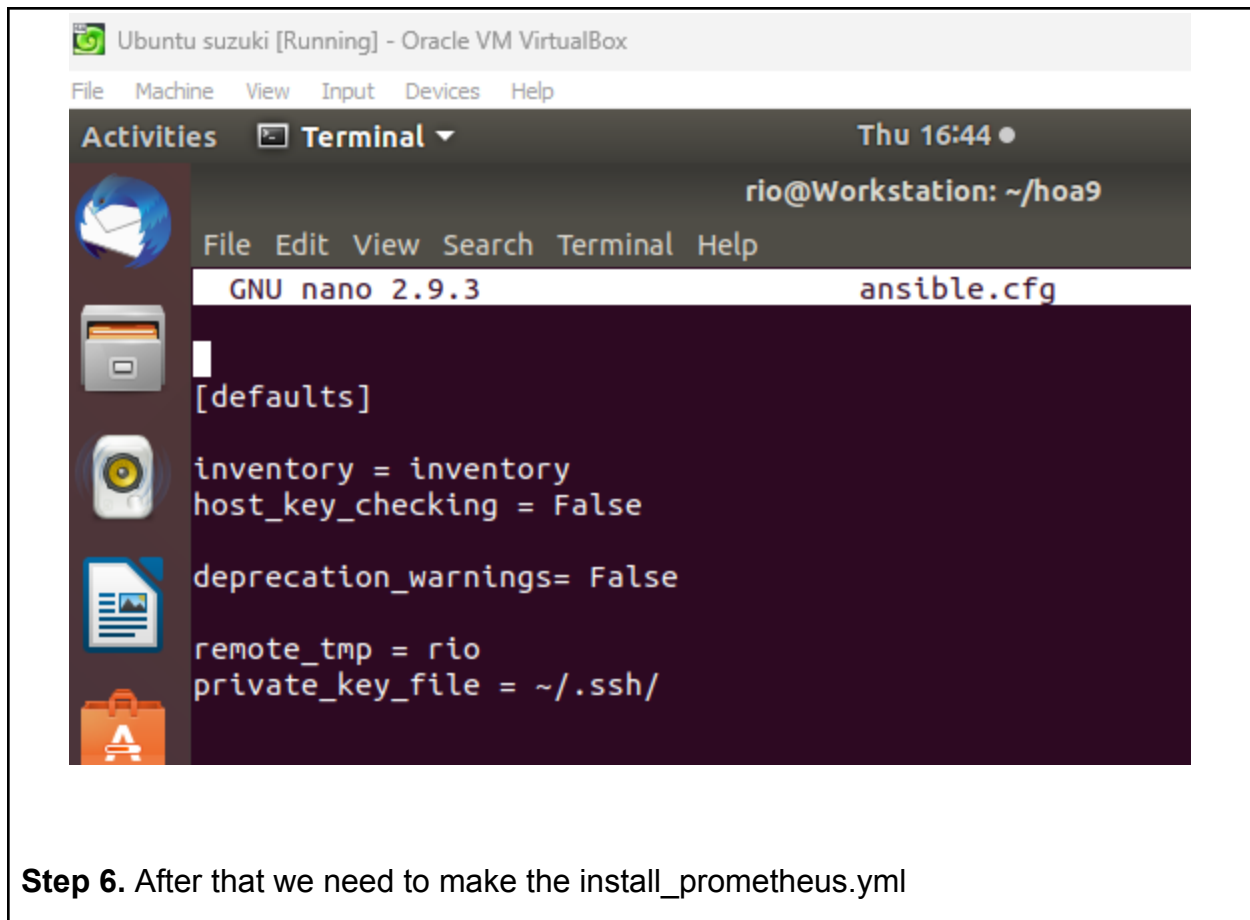


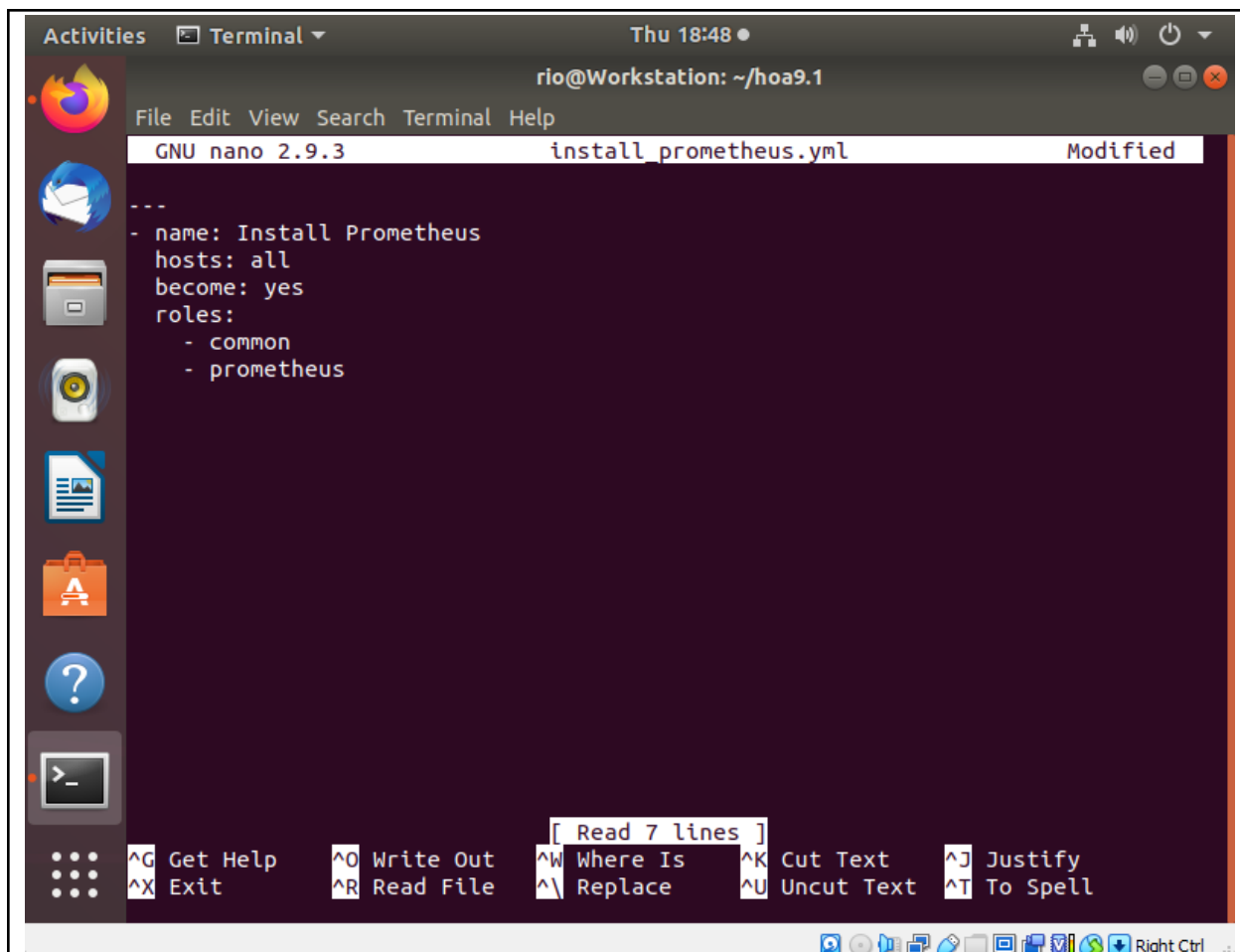
The screenshot shows a terminal window titled "Ubuntu suzuki [Running] - Oracle VM VirtualBox". The terminal output shows the user "rio@Workstation" in the directory "~/.hoa9" using the "nano" text editor to create a file named "inventory". The content of the file is as follows:

```
GNU nano 2.9.3 inventory
[ubuntu]
192.168.56.102 ansible_python_interpreter=/usr/bin/python3

[centos]
rio@192.168.56.105
```

**Step 5.** After creating the inventory we need to create the ansible.cfg which contains the defaults group, in which I just copied in the past hands on activity given to us.



A screenshot of a Linux terminal window. The window title is 'rio@Workstation: ~/hoa9.1'. The terminal shows the nano text editor editing a file named 'install\_prometheus.yml'. The content of the file is a YAML configuration for an Ansible play. The configuration specifies the play name as 'Install Prometheus', targets 'all' hosts, and assigns two roles: 'common' and 'prometheus'. The nano editor's status bar at the bottom shows 'GNU nano 2.9.3' and 'Modified'. The terminal window has a sidebar with application icons on the left and a menu bar at the top. The bottom of the terminal shows a prompt 'rio@Workstation: ~/hoa9\$' and a list of nano editor shortcuts.

```
rio@Workstation: ~/hoa9.1
File Edit View Search Terminal Help
GNU nano 2.9.3 install_prometheus.yml Modified
---
- name: Install Prometheus
  hosts: all
  become: yes
  roles:
    - common
    - prometheus
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify  
^X Exit ^R Read File ^\_ Replace ^U Uncut Text ^T To Spell

**Step 7.** Next step is to create a directory structure for this activity. First we need to create roles by using the command “mkdir roles” and change the directory to roles.

```
rio@Workstation:~/hoa9$ mkdir roles
rio@Workstation:~/hoa9$ cd roles
rio@Workstation:~/hoa9/roles$
```

**Step 8.** After creating the roles we need to create 2 separate roles under the roles which are the common and prometheus.

```

rio@Workstation:~/hoa9/roles$ mkdir common
rio@Workstation:~/hoa9/roles$ mkdir prometheus
rio@Workstation:~/hoa9/roles$ ls
common  prometheus
rio@Workstation:~/hoa9/roles$ cd ..
rio@Workstation:~/hoa9$ tree
.
├── ansible.cfg
├── inventory
├── README.md
└── roles
    ├── common
    └── prometheus

3 directories, 3 files

```

**Step 9.** After creating the common we will make another directory under the common name it as task we will put here the tasks that are both applicable on ubuntu and centOS, like updating packages and installing essential packages. After creating we will now go back to roles using the command “cd ..”.

```

rio@Workstation:~/hoa9$ cd roles
rio@Workstation:~/hoa9/roles$ cd common
rio@Workstation:~/hoa9/roles/common$ mkdir tasks
rio@Workstation:~/hoa9/roles/common$ cd ..

```

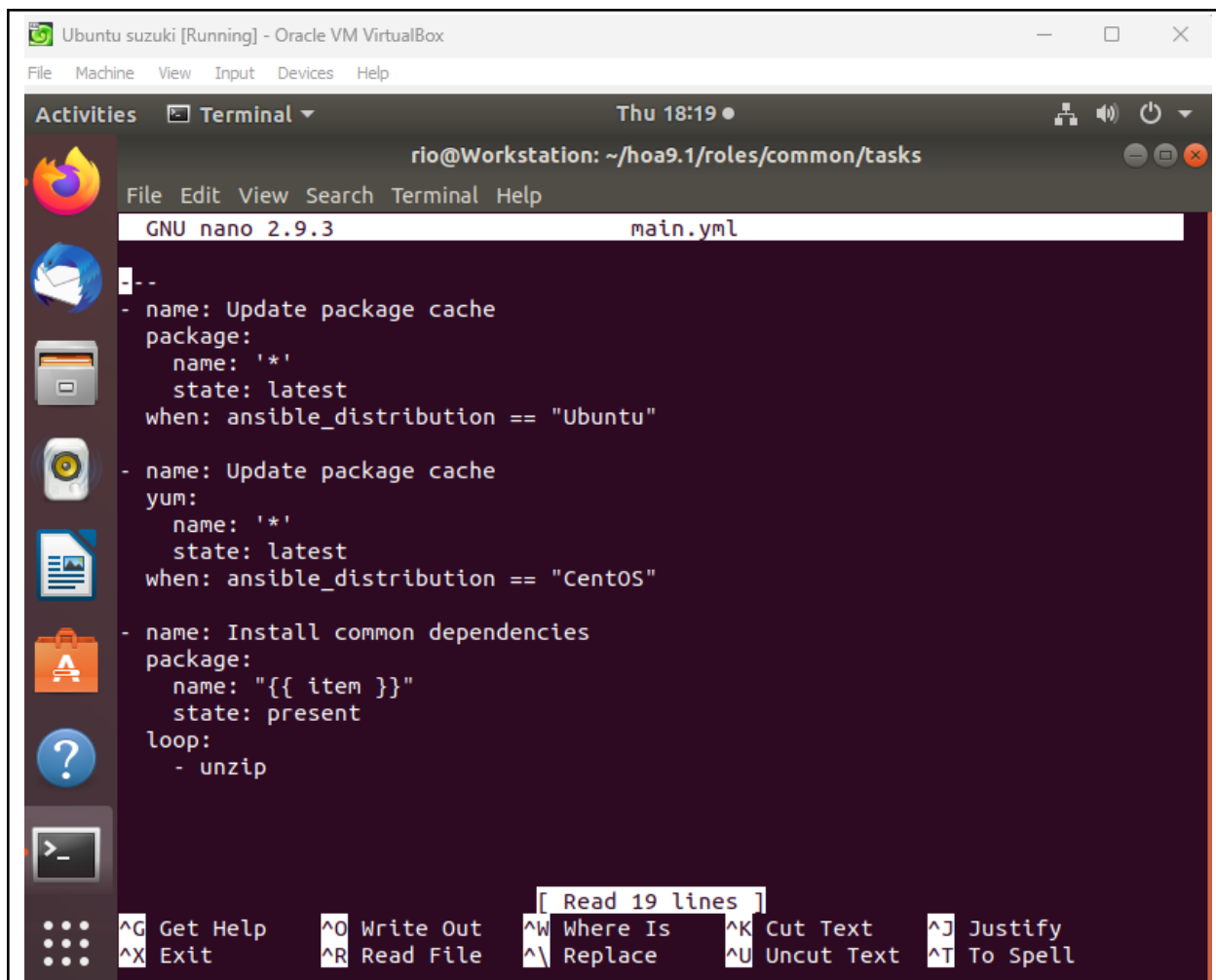
**Step 10.** In this step we will create the directory of prometheus under the roles which will contain the file of tasks that would be related to installing the prometheus.

```

rio@Workstation:~/hoa9.1/roles/common$ cd ..
rio@Workstation:~/hoa9.1/roles$ mkdir prometheus
rio@Workstation:~/hoa9.1/roles$ cd prometheus
rio@Workstation:~/hoa9.1/roles/prometheus$ mkdir tasks
rio@Workstation:~/hoa9.1/roles/prometheus$ cd taks
bash: cd: taks: No such file or directory
rio@Workstation:~/hoa9.1/roles/prometheus$ cd tasks
rio@Workstation:~/hoa9.1/roles/prometheus/tasks$ sudo nano main.yml

```

**Step 11.** For the next step we will see the codes inside the main.yml of common under the roles.

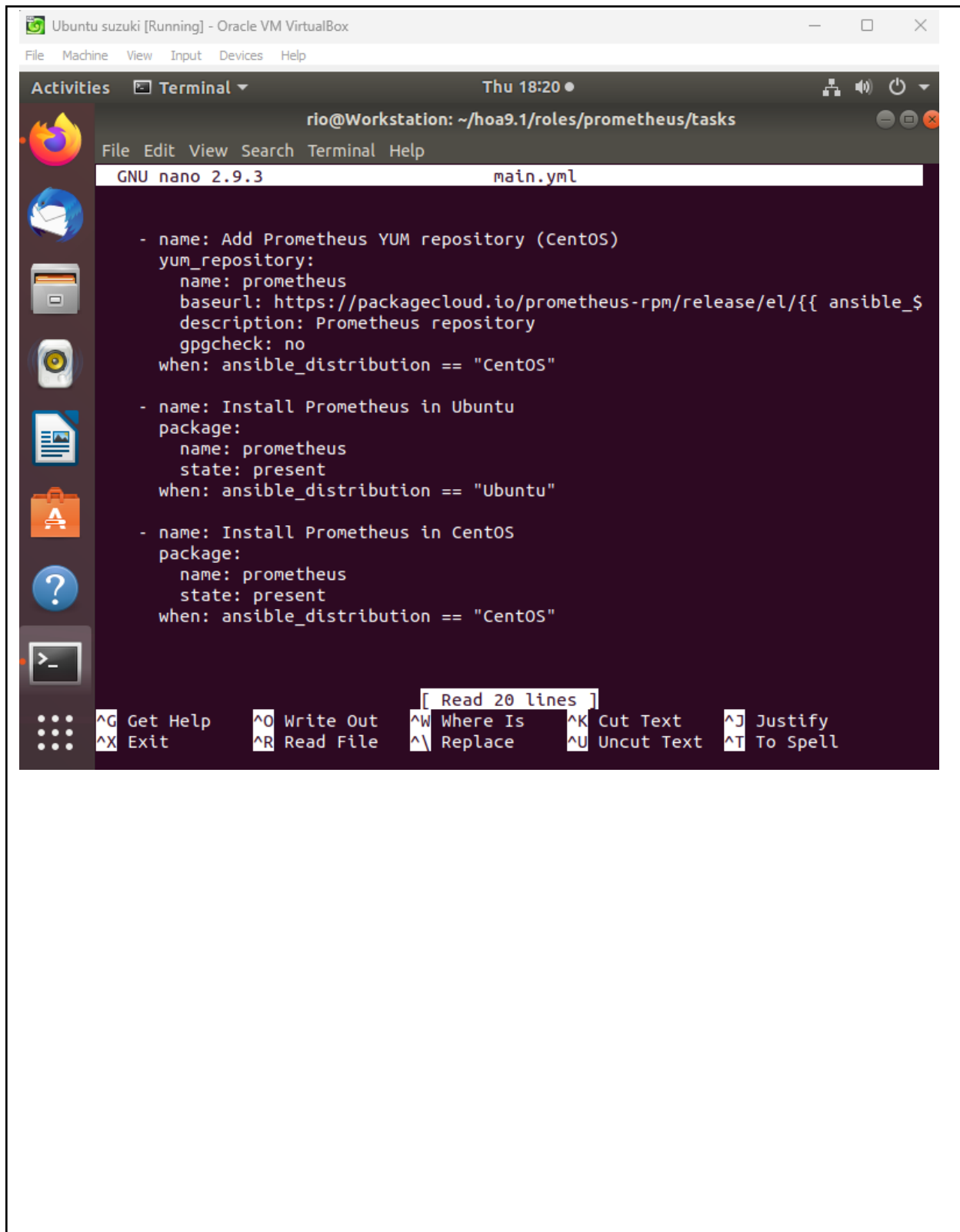


The screenshot shows a terminal window titled "Ubuntu suzuki [Running] - Oracle VM VirtualBox". The terminal is running the nano text editor, editing a file named "main.yml" located at "~/.hoa9.1/roles/common/tasks". The editor's status bar indicates "GNU nano 2.9.3" and "main.yml". The file content is as follows:

```
--  
- name: Update package cache  
  package:  
    name: '*'  
    state: latest  
  when: ansible_distribution == "Ubuntu"  
  
- name: Update package cache  
  yum:  
    name: '*'  
    state: latest  
  when: ansible_distribution == "CentOS"  
  
- name: Install common dependencies  
  package:  
    name: "{{ item }}"  
    state: present  
  loop:  
    - unzip
```

The terminal window also displays a sidebar with application icons and a bottom status bar with keyboard shortcuts.

**Step 12.** Next we will see the codes inside the main.yml of prometheus under the roles.





**Step 13.** To run the playbook to see if it's running we will use the command "ansible-playbook --ask-become-pass install\_prometheus.yml" if there's no error in the output then the playbook has the right command. Here is the output of this playbook:

```
rio@Workstation:~/hoa9.1$ ansible-playbook --ask-become-pass install_prometheus.yml
BECOME password:

PLAY [Install Prometheus] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.102]
ok: [rio@192.168.56.105]

TASK [common : Update package cache] *****
*
skipping: [rio@192.168.56.105]
ok: [192.168.56.102]

TASK [common : Update package cache] *****
*
skipping: [192.168.56.102]
ok: [rio@192.168.56.105]

TASK [common : Install common dependencies] *****
*
ok: [rio@192.168.56.105] => (item=unzip)
ok: [192.168.56.102] => (item=unzip)

TASK [prometheus : Add Prometheus YUM repository (CentOS)] *****
```

```

*
ok: [rio@192.168.56.105] => (item=unzip)
ok: [192.168.56.102] => (item=unzip)

TASK [prometheus : Add Prometheus YUM repository (CentOS)] *****
*
skipping: [192.168.56.102]
changed: [rio@192.168.56.105]

TASK [prometheus : Install Prometheus in Ubuntu] *****
*
skipping: [rio@192.168.56.105]
changed: [192.168.56.102]

TASK [prometheus : Install Prometheus in CentOS] *****
*
skipping: [192.168.56.102]
changed: [rio@192.168.56.105]

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

rio@Workstation:~/hoa9.1$ cd roles
rio@Workstation:~/hoa9.1/roles$ tree
.
├── common
│   └── tasks
│       └── main.yml
└── prometheus
    ├── tasks
    └── main.yml

4 directories, 2 files
rio@Workstation:~/hoa9.1/roles$

```

**Step 14.** Since we have no errors in the output we will then again check the tree if the tree is correct.

```

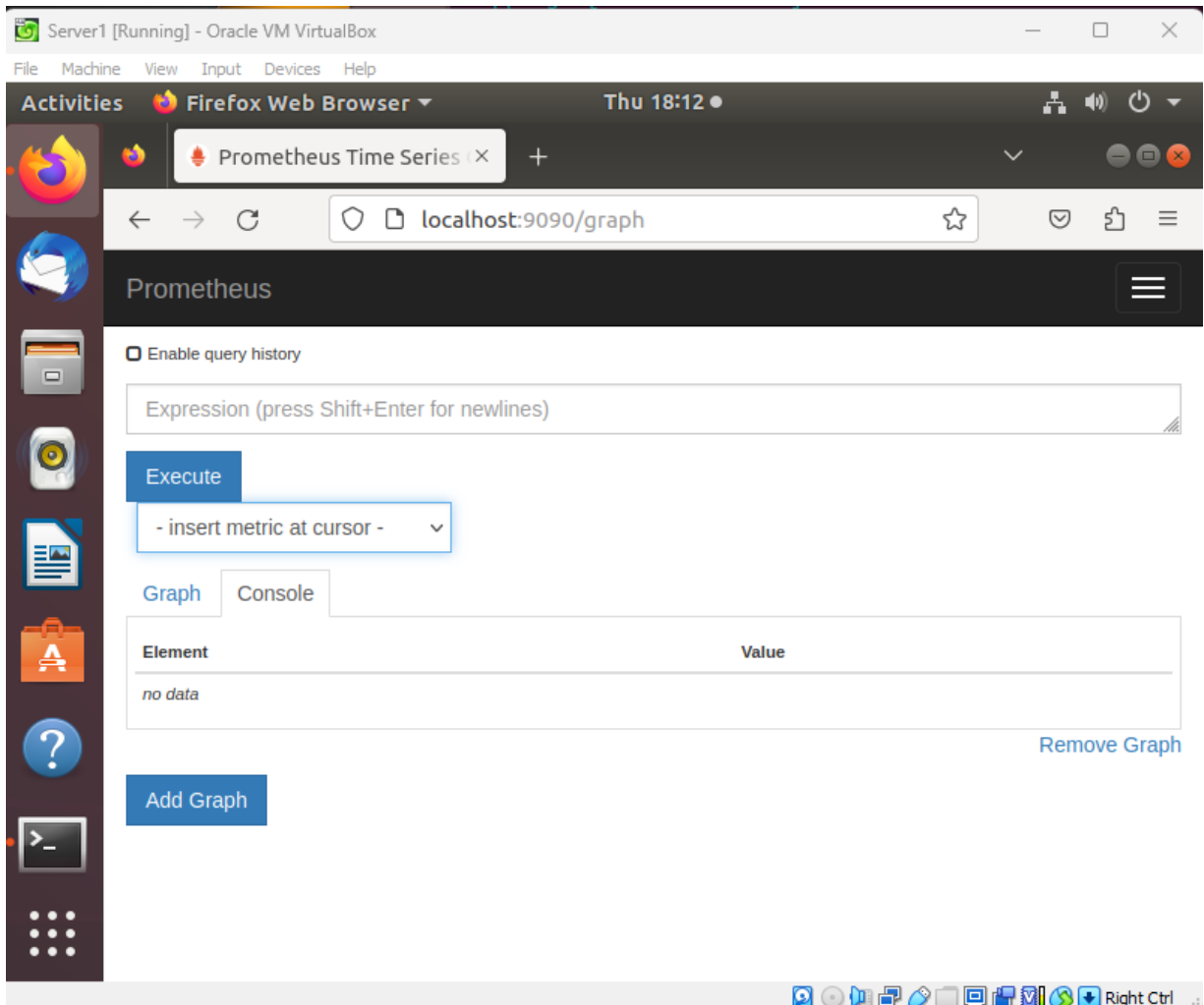
rio@Workstation:~/hoa9.1$ cd roles
rio@Workstation:~/hoa9.1/roles$ tree

.
├── common
│   └── tasks
│       └── main.yml
└── prometheus
    ├── tasks
    └── main.yml

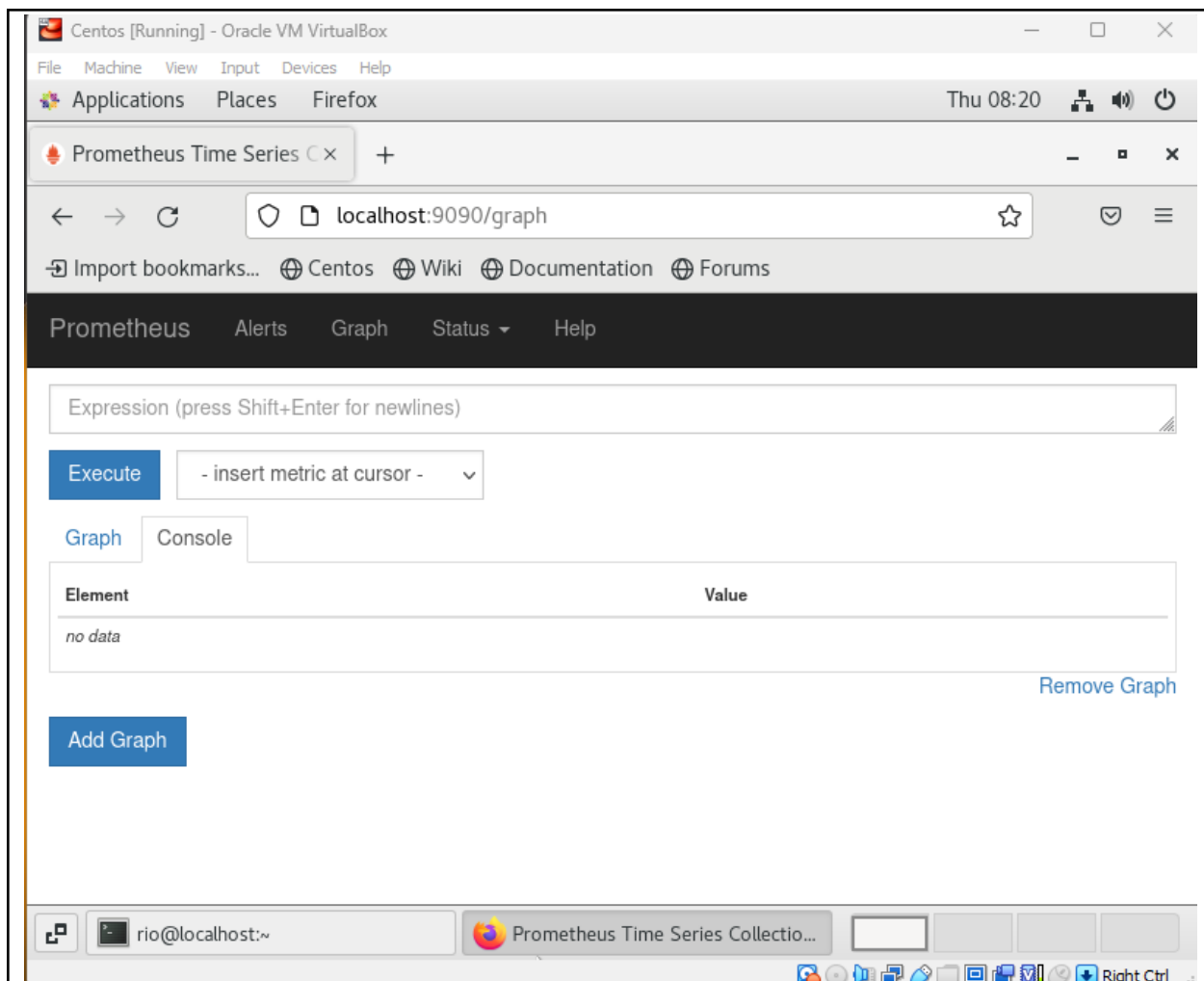
4 directories, 2 files
rio@Workstation:~/hoa9.1/roles$

```

**Step 15.** We will also check in the firefox of the Server 1 and CentOS. The command we will use is the “http:localhost:9090”



**Step 16.** We will also check in the Firefox of Server 1 and Firefox of CentOS. The command we will use is the same in the server 1 which is the “http:localhost:9090”.



**Step 17.** Lastly, we will now upload the things we did in the hoa9.1 repository to the github using the command “git add .”, “git commit”, and “git push”.

```
rio@Workstation:~/hoa9.1$ git add .
rio@Workstation:~/hoa9.1$ git commit -m "hoa9.1"
[main e484565] hoa9.1
 5 files changed, 57 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 install_prometheus.yml
 create mode 100644 inventory
 create mode 100644 roles/common/tasks/main.yml
 create mode 100644 roles/prometheus/tasks/main.yml
rio@Workstation:~/hoa9.1$ git push
Counting objects: 12, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (12/12), 1.26 KiB | 1.26 MiB/s, done.
Total 12 (delta 0), reused 0 (delta 0)
To github.com:RioMarieee/hoa9.1.git
 b9db6e3..e484565  main -> main
rio@Workstation:~/hoa9.1$
```

**Repository Link:** <https://github.com/RioMarieee/hoa9.1.git>

### Reflections:

Answer the following:

1. What are the benefits of having a performance monitoring tool?

- Performance monitoring tools are important among industries, ensuring the seamless operation of IT systems. It is used in proactive issue detection, empowering efficient resource allocation, troubleshooting processes, and elevating the end-user experience. These tools serve as guardians of security, regulatory compliance, and optimal resource distribution. Their arsenal includes data analytics, predictive insights, and customizable alerts, empowering organizations with informed decision-making capabilities. Adaptable and equipped for remote monitoring, these tools meet the diverse operational demands of modern businesses. Furthermore, they excel in resource optimization, trimming expenses linked to downtime, and supporting adherence to service level agreements. In essence, integrating performance monitoring tools is a strategic necessity for dependable, efficient, and cost-effective IT systems, reinforcing the competitive edge and triumph of businesses across the spectrum.

### Conclusions:

In conclusion, the pursuit of crafting a tailored workflow for the deployment, configuration, and maintenance of enterprise performance tools using Ansible as an Infrastructure as Code (IaC) instrument signifies a significant leap toward a more agile, resource-efficient, and robust infrastructure management approach. Through the leverage of Ansible's automation prowess, we've not only streamlined what were once intricate and time-consuming tasks associated with performance tools, but we've also laid the groundwork for standardized and scalable practices. The positive influence of this workflow manifests in various dimensions, most notably the substantial reduction in human errors, the nimble adaptability to evolving requirements, meticulous adherence to documentation and compliance norms, and the facilitation of collaborative efforts with enhanced version control. Moreover, the fiscal advantages, highlighted by cost savings and optimized resource allocation, underscore the immense importance of this approach. In essence, the integration of Ansible as an IaC tool for the administration of enterprise performance tools has firmly established itself as a cornerstone in modern infrastructure management. It fully embodies the core tenets of operational efficiency, unwavering reliability, and adaptive versatility, all of which are indispensable in upholding peak performance within the ever-evolving landscape of contemporary IT environments.

## Assessment Rubrics

Rubric for SO 7													
Criteria	Ratings								Pts				
SO 7 PI 1 Acquire and apply new knowledge from outside sources threshold: 4.8 pts	6 pts Excellent   Educational interests and pursuits exist and flourish outside classroom requirements, knowledge and/or experiences are pursued independently and applies knowledge learned into practice		5 pts Good   Educational interests and pursuits exist and flourish outside classroom requirements, knowledge and/or experiences are pursued independently		4 pts Satisfactory   Look beyond classroom requirements, showing interest in pursuing knowledge independently		3 pts Unsatisfactory   Begins to look beyond classroom requirements, showing interest in pursuing knowledge independently		2 pts Poor   Relies on classroom instruction only		1 pts Very Poor   No initiative or interest in acquiring new knowledge		6 pts
SO 7 PI 2 Learn independently. threshold: 4.8 pts	6 pts Excellent   Completes an assigned task independently and practices continuous improvement		5 pts Good   Completes an assigned task without supervision or guidance		4 pts Satisfactory   Requires minimal guidance to complete an assigned task		3 pts Unsatisfactory   Requires detailed or step-by-step instructions to complete a task		2 pts Poor   Shows little interest to complete a task independently		1 pts Very Poor   No interest to complete a task independently		6 pts
SO 7 PI 3 Critical thinking in the broadest context of technological change threshold: 4.8 pts	6 pts Excellent   Synthesizes and integrates information from a variety of sources; formulates a clear and precise perspective; draws appropriate conclusions		5 pts Good   Evaluate information from a variety of sources; formulates a clear and precise perspective.		4 pts Satisfactory   Analyze information from a variety of sources; formulates a clear and precise perspective.		3 pts Unsatisfactory   Apply the gathered information to formulate the problem		2 pts Poor   Gather and summarized the information from a variety of sources but failed to formulate the problem		1 pts Very Poor   Gather information from a variety of sources		6 pts
SO 7 PI 4 Creativity and adaptability to new and emerging technologies threshold: 4.8 pts	6 pts Excellent   Ideas are combined in original and creative ways in line with the new and emerging technology trends to solve a problem or address an issue.		5 pts Good   Ideas are creative and adapt the new knowledge to solve a problem or address an issue		4 pts Satisfactory   Ideas are creative in solving a problem, or address an issue		3 pts Unsatisfactory   Shows some creative ways to solve the problem		2 pts Poor   Shows initiative and attempt to develop creative ideas to solve the problem		1 pts Very Poor   Ideas are copied or restated from the sources consulted		6 pts
Total Points: 24													