-

Name: FRIAS, ABEGAIL L.	**Date Performed: Oct. 20, 2024**
Course/Section: CPE212 - CPE31S21	**Date Submitted: Oct. 20, 2024**
Instructor: Engr. Robin Valenzuela	**Semester and SY: 1st Sem/2024-2025**

Activity 9: Install, Configure, and Manage Performance Monitoring tools

1. Objectives

Create and design a workflow that installs, configure and manage enterprise performance tools using Ansible as an Infrastructure as Code (IaC) tool.

2. Discussion

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.

Prometheus

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: Prometheus - Monitoring system & time series database

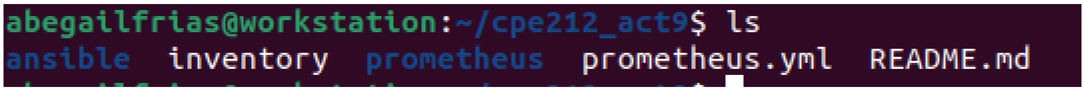
Cacti

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: Cacti® - The Complete RRDTool-based Graphing Solution

3. Tasks

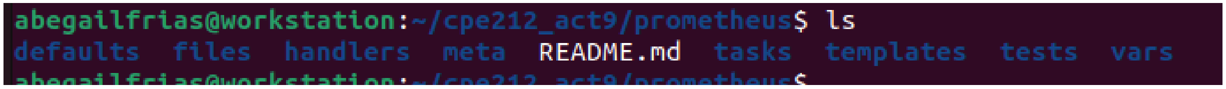
1. Create a playbook that installs Prometheus in both Ubuntu and CentOS. Apply the concept of creating roles.
2. Describe how you did step 1. (Provide screenshots and explanations in your report. Make your report detailed such that it will look like a manual.)
3. Show an output of the installed Prometheus for both Ubuntu and CentOS.
4. Make sure to create a new repository in GitHub for this activity.

4. Output (screenshots and explanations)



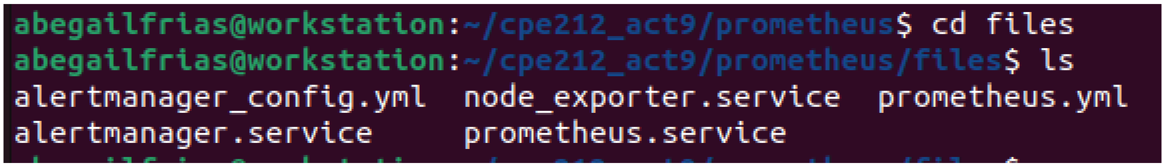
```
abegailfrias@workstation:~/cpe212_act9$ ls
ansible  inventory  prometheus  prometheus.yml  README.md
```

Step 1: After creating a new repository type ansible-galaxy init prometheus



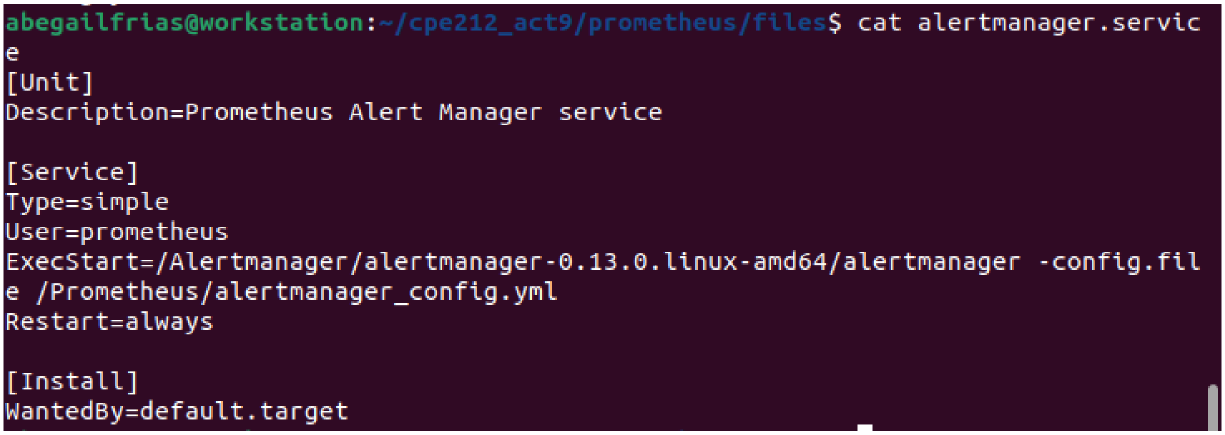
```
abegailfrias@workstation:~/cpe212_act9/prometheus$ ls
defaults  files  handlers  meta  README.md  tasks  templates  tests  vars
abegailfrias@workstation:~/cpe212_act9/prometheus$
```

Step 2: ls and check the directories of prometheus.



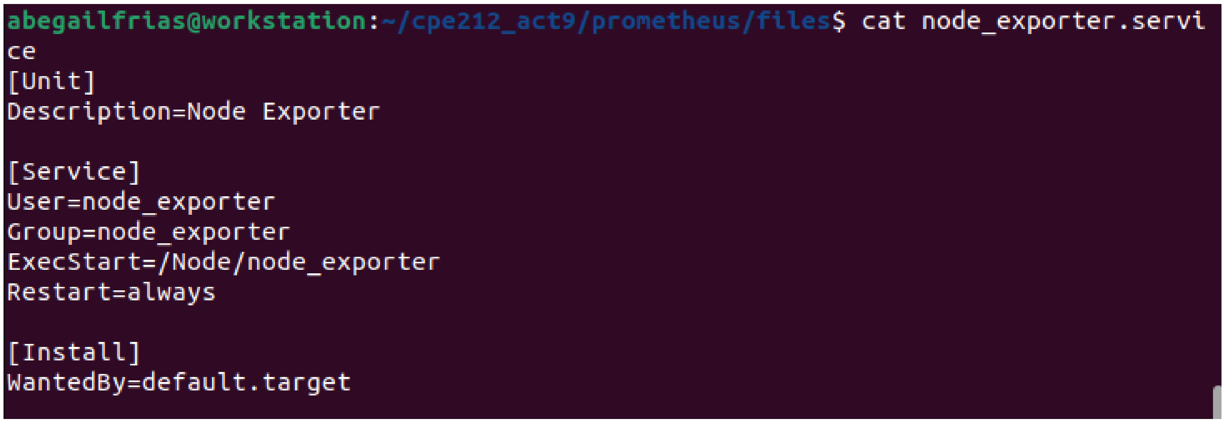
```
abegailfrias@workstation:~/cpe212_act9/prometheus$ cd files
abegailfrias@workstation:~/cpe212_act9/prometheus/files$ ls
alertmanager_config.yml  node_exporter.service  prometheus.yml
alertmanager.service     prometheus.service
```

Step 3: Now go to the file directory and touch all the file seen on the picture above.



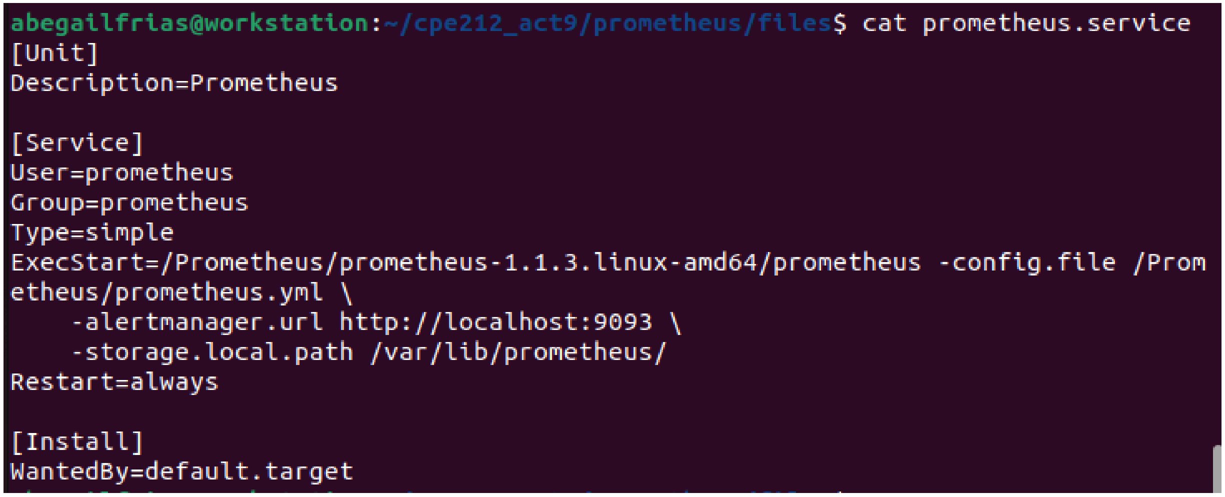
```
abegailfrias@workstation:~/cpe212_act9/prometheus/files$ cat alertmanager.servic
e
[Unit]
Description=Prometheus Alert Manager service

[Service]
Type=simple
User=prometheus
ExecStart=/Alertmanager/alertmanager-0.13.0.linux-amd64/alertmanager -config.fil
e /Prometheus/alertmanager_config.yml
Restart=always

[Install]
WantedBy=default.target
```

Step 4: edit the alertmanager.service shown above.



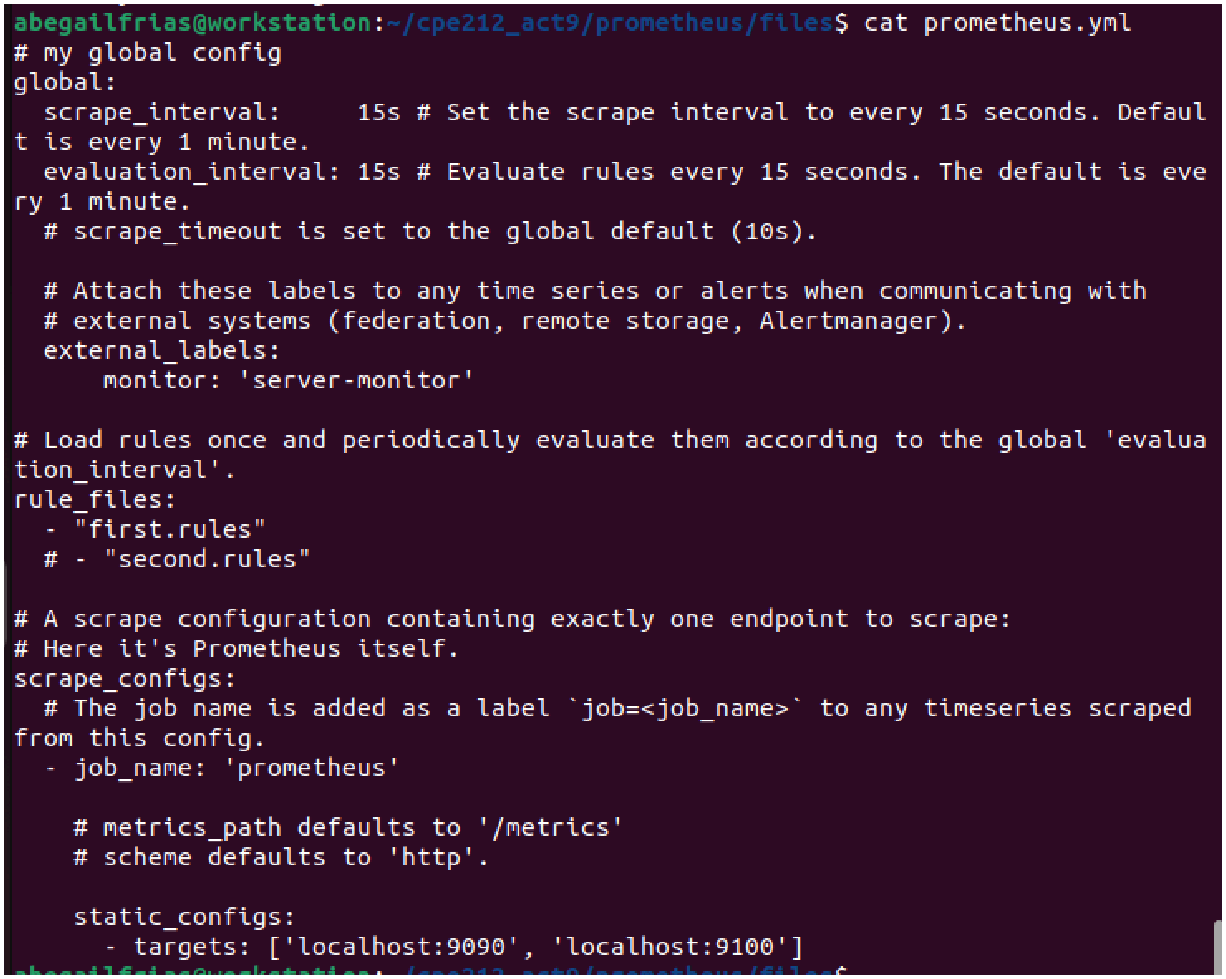
```
abegailfrias@workstation:~/cpe212_act9/prometheus/files$ cat node_exporter.servi
ce
[Unit]
Description=Node Exporter

[Service]
User=node_exporter
Group=node_exporter
ExecStart=/Node/node_exporter
Restart=always

[Install]
WantedBy=default.target
```

Step 5: Edit the node_exporter.service



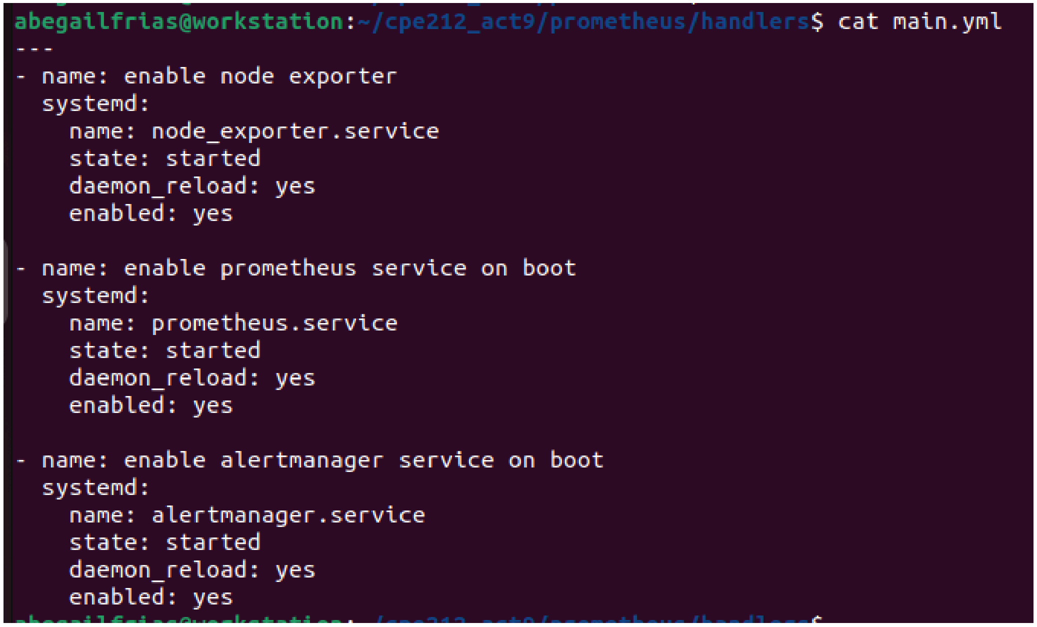
```
abegailfrias@workstation:~/cpe212_act9/prometheus/files$ cat prometheus.service
[Unit]
Description=Prometheus

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/Prometheus/prometheus-1.1.3.linux-amd64/prometheus -config.file /Prom
etheus/prometheus.yml \
    -alertmanager.url http://localhost:9093 \
    -storage.local.path /var/lib/prometheus/
Restart=always

[Install]
WantedBy=default.target
```

Step 6: Edit the prometheus.service



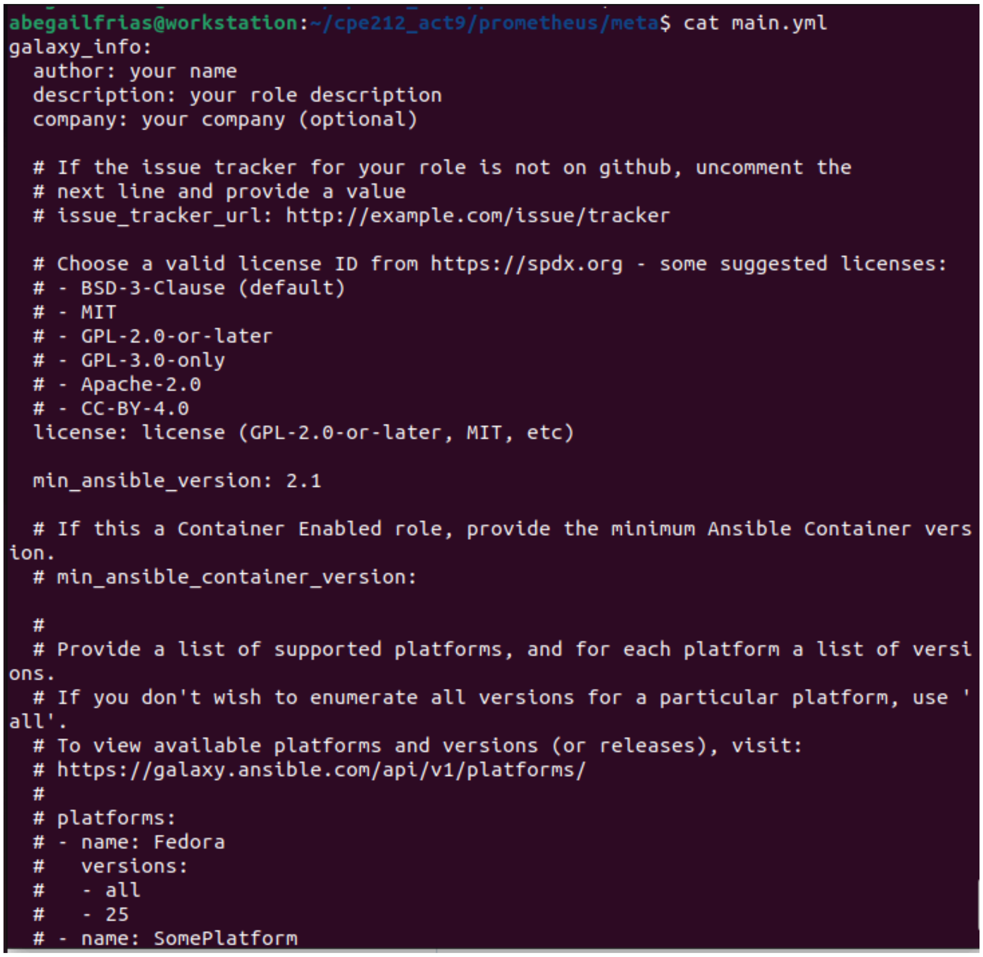
```
abegailfrias@workstation:~/cpe212_act9/prometheus/files$ cat prometheus.yml
# my global config
global:
  scrape_interval:     15s # Set the scrape interval to every 15 seconds. Defaul
t is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is eve
ry 1 minute.
  # scrape_timeout is set to the global default (10s).

  # Attach these labels to any time series or alerts when communicating with
  # external systems (federation, remote storage, Alertmanager).
  external_labels:
      monitor: 'server-monitor'

# Load rules once and periodically evaluate them according to the global 'evalua
tion_interval'.
rule_files:
  - "first.rules"
  # - "second.rules"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped
from this config.
  - job_name: 'prometheus'

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ['localhost:9090', 'localhost:9100']
```

Step 7: Edit the prometheus.yml

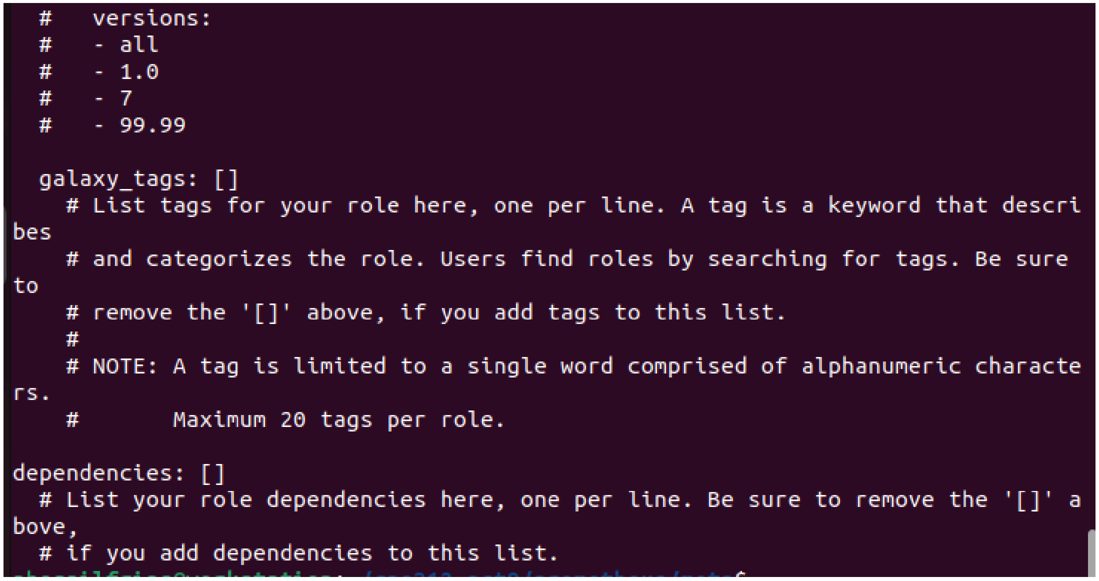


```
abegailfrias@workstation:~/cpe212_act9/prometheus/handlers$ cat main.yml
---
- name: enable node exporter
  systemd:
    name: node_exporter.service
    state: started
    daemon_reload: yes
    enabled: yes

- name: enable prometheus service on boot
  systemd:
    name: prometheus.service
    state: started
    daemon_reload: yes
    enabled: yes

- name: enable alertmanager service on boot
  systemd:
    name: alertmanager.service
    state: started
    daemon_reload: yes
    enabled: yes
abegailfrias@workstation:~/cpe212_act9/prometheus/handlers$
```

Step 8: Cd the handlers, ls and edit the main.yml

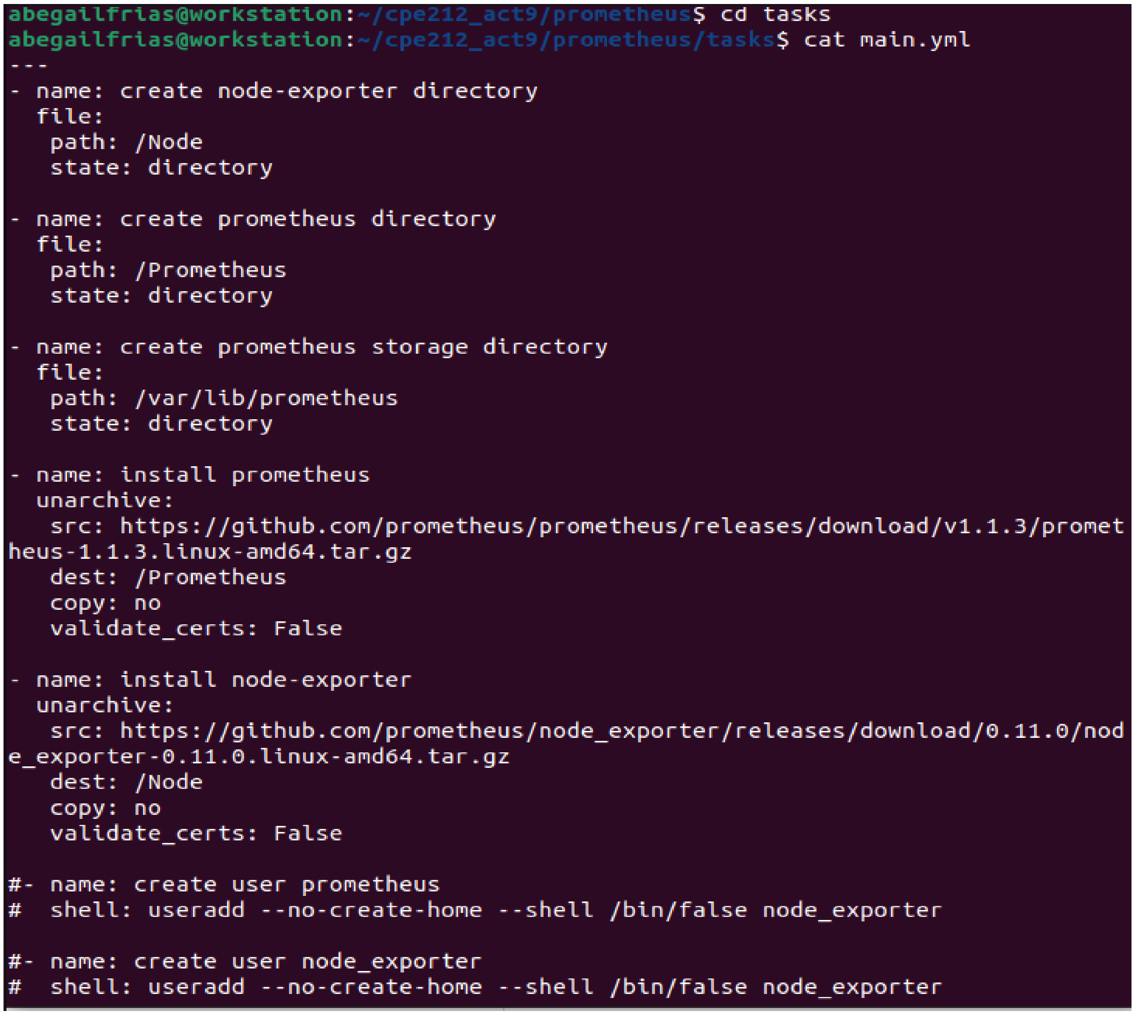


```
abegailfrias@workstation:~/cpe212_act9/prometheus/meta$ cat main.yml
galaxy_info:
  author: your name
  description: your role description
  company: your company (optional)

  # If the issue tracker for your role is not on github, uncomment the
  # next line and provide a value
  # issue_tracker_url: http://example.com/issue/tracker

  # Choose a valid license ID from https://spdx.org - some suggested licenses:
  # - BSD-3-Clause (default)
  # - MIT
  # - GPL-2.0-or-later
  # - GPL-3.0-only
  # - Apache-2.0
  # - CC-BY-4.0
  license: license (GPL-2.0-or-later, MIT, etc)

  min_ansible_version: 2.1

  # If this a Container Enabled role, provide the minimum Ansible Container vers
ion.
  # min_ansible_container_version:

  #
  # Provide a list of supported platforms, and for each platform a list of versi
ons.
  # If you don't wish to enumerate all versions for a particular platform, use '
all'.
  # To view available platforms and versions (or releases), visit:
  # https://galaxy.ansible.com/api/v1/platforms/
  #
  # platforms:
  # - name: Fedora
  #   versions:
  #   - all
  #   - 25
  # - name: SomePlatform
```

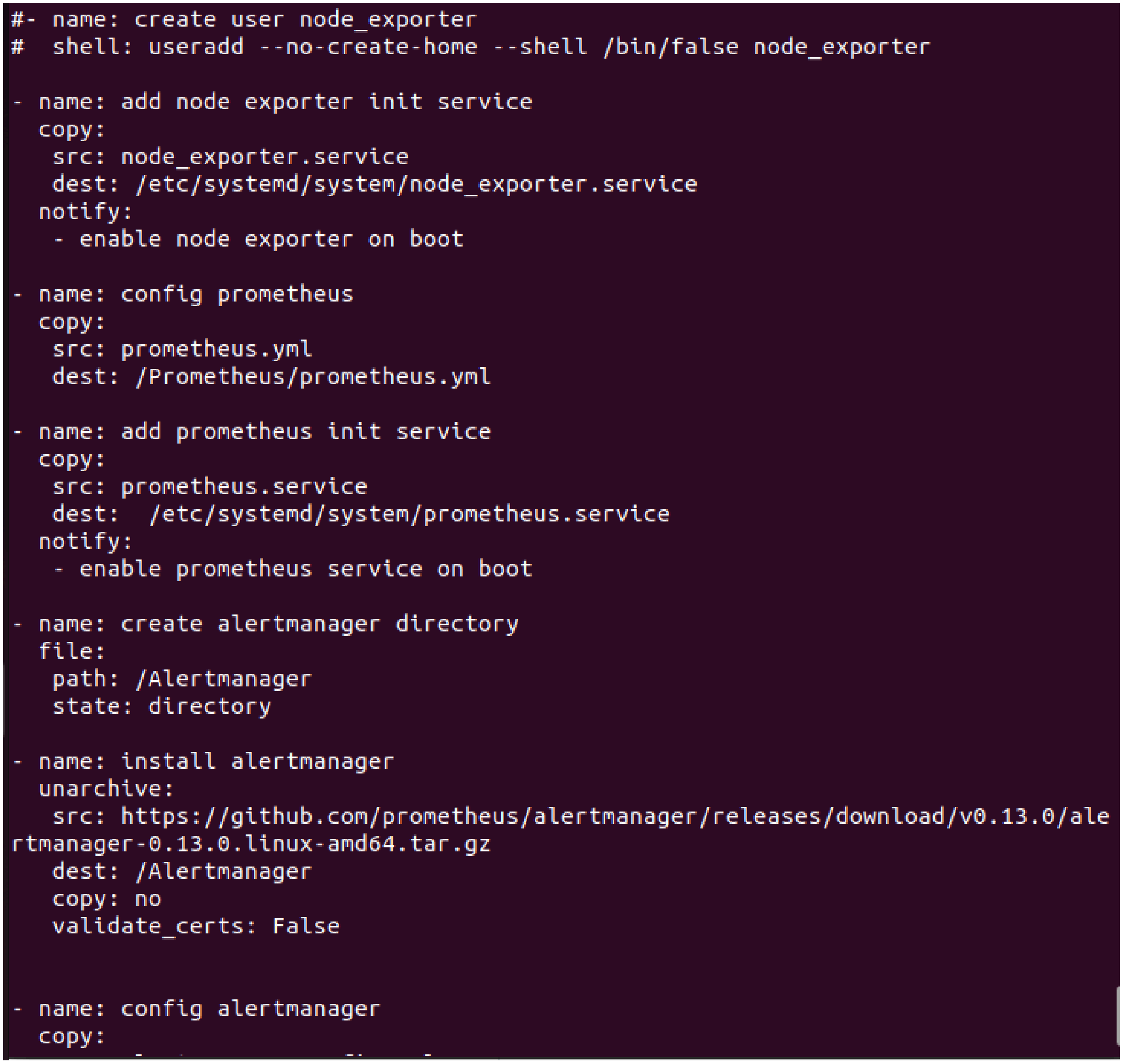


```
  #   versions:
  #   - all
  #   - 1.0
  #   - 7
  #   - 99.99

  galaxy_tags: []
    # List tags for your role here, one per line. A tag is a keyword that descri
bes
    # and categorizes the role. Users find roles by searching for tags. Be sure
to
    # remove the '[]' above, if you add tags to this list.
    #
    # NOTE: A tag is limited to a single word comprised of alphanumeric characte
rs.
    #       Maximum 20 tags per role.

dependencies: []
  # List your role dependencies here, one per line. Be sure to remove the '[]' a
bove,
  # if you add dependencies to this list.
```

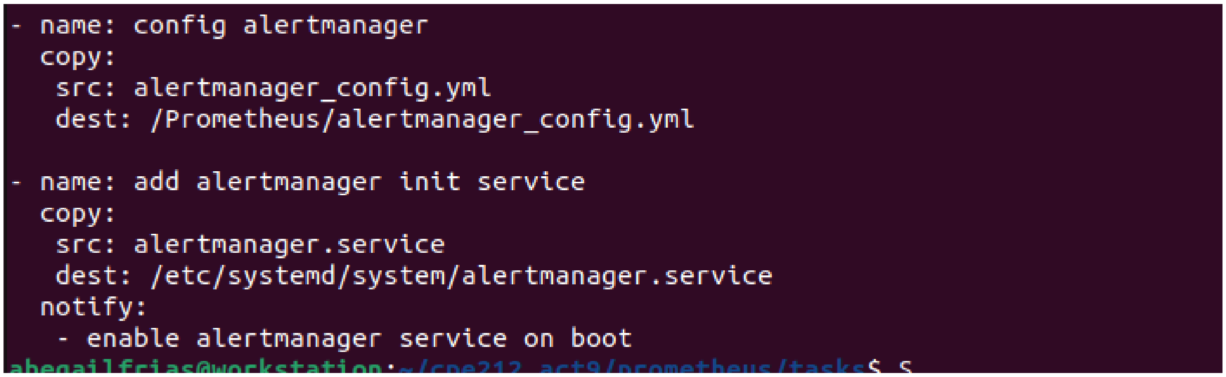
Step 9: Cd to meta, ls and edit the main.yml



```
abegailfrias@workstation:~/cpe212_act9/prometheus$ cd tasks
abegailfrias@workstation:~/cpe212_act9/prometheus/tasks$ cat main.yml
---
- name: create node-exporter directory
  file:
   path: /Node
   state: directory

- name: create prometheus directory
  file:
   path: /Prometheus
   state: directory

- name: create prometheus storage directory
  file:
   path: /var/lib/prometheus
   state: directory

- name: install prometheus
  unarchive:
   src: https://github.com/prometheus/prometheus/releases/download/v1.1.3/promet
heus-1.1.3.linux-amd64.tar.gz
   dest: /Prometheus
   copy: no
   validate_certs: False

- name: install node-exporter
  unarchive:
   src: https://github.com/prometheus/node_exporter/releases/download/0.11.0/nod
e_exporter-0.11.0.linux-amd64.tar.gz
   dest: /Node
   copy: no
   validate_certs: False

#- name: create user prometheus
#  shell: useradd --no-create-home --shell /bin/false node_exporter

#- name: create user node_exporter
#  shell: useradd --no-create-home --shell /bin/false node_exporter
```

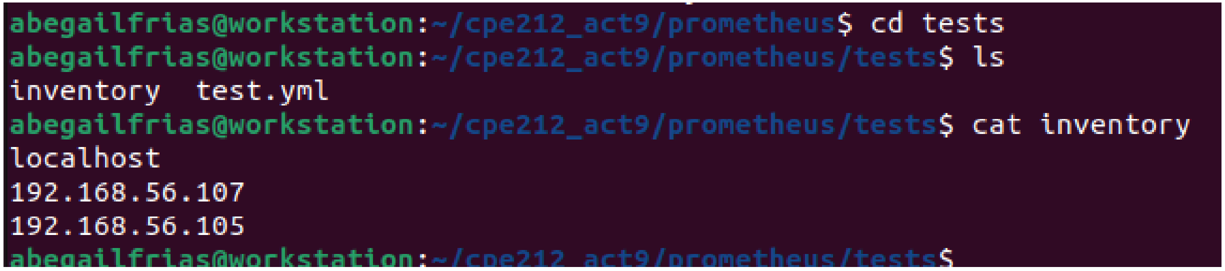


```
#- name: create user node_exporter
#  shell: useradd --no-create-home --shell /bin/false node_exporter

- name: add node exporter init service
  copy:
   src: node_exporter.service
   dest: /etc/systemd/system/node_exporter.service
  notify:
   - enable node exporter on boot

- name: config prometheus
  copy:
   src: prometheus.yml
   dest: /Prometheus/prometheus.yml

- name: add prometheus init service
  copy:
   src: prometheus.service
   dest:  /etc/systemd/system/prometheus.service
  notify:
   - enable prometheus service on boot

- name: create alertmanager directory
  file:
   path: /Alertmanager
   state: directory

- name: install alertmanager
  unarchive:
   src: https://github.com/prometheus/alertmanager/releases/download/v0.13.0/ale
rtmanager-0.13.0.linux-amd64.tar.gz
   dest: /Alertmanager
   copy: no
   validate_certs: False


- name: config alertmanager
  copy:
```

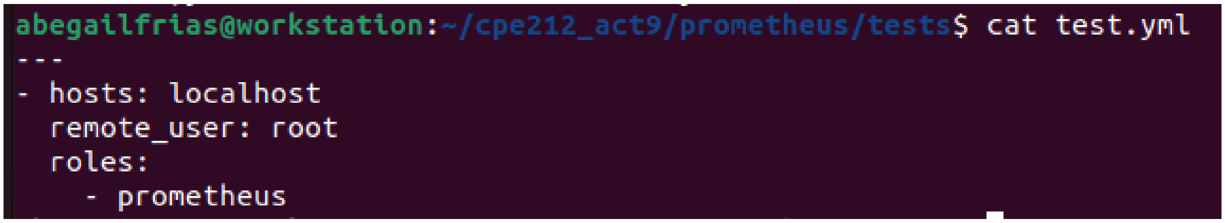


```
- name: config alertmanager
  copy:
   src: alertmanager_config.yml
   dest: /Prometheus/alertmanager_config.yml

- name: add alertmanager init service
  copy:
   src: alertmanager.service
   dest: /etc/systemd/system/alertmanager.service
  notify:
   - enable alertmanager service on boot
abegailfrias@workstation:~/cpe212_act9/prometheus/tasks$ $
```

Step 10: Cd tasks and edit the main.yml

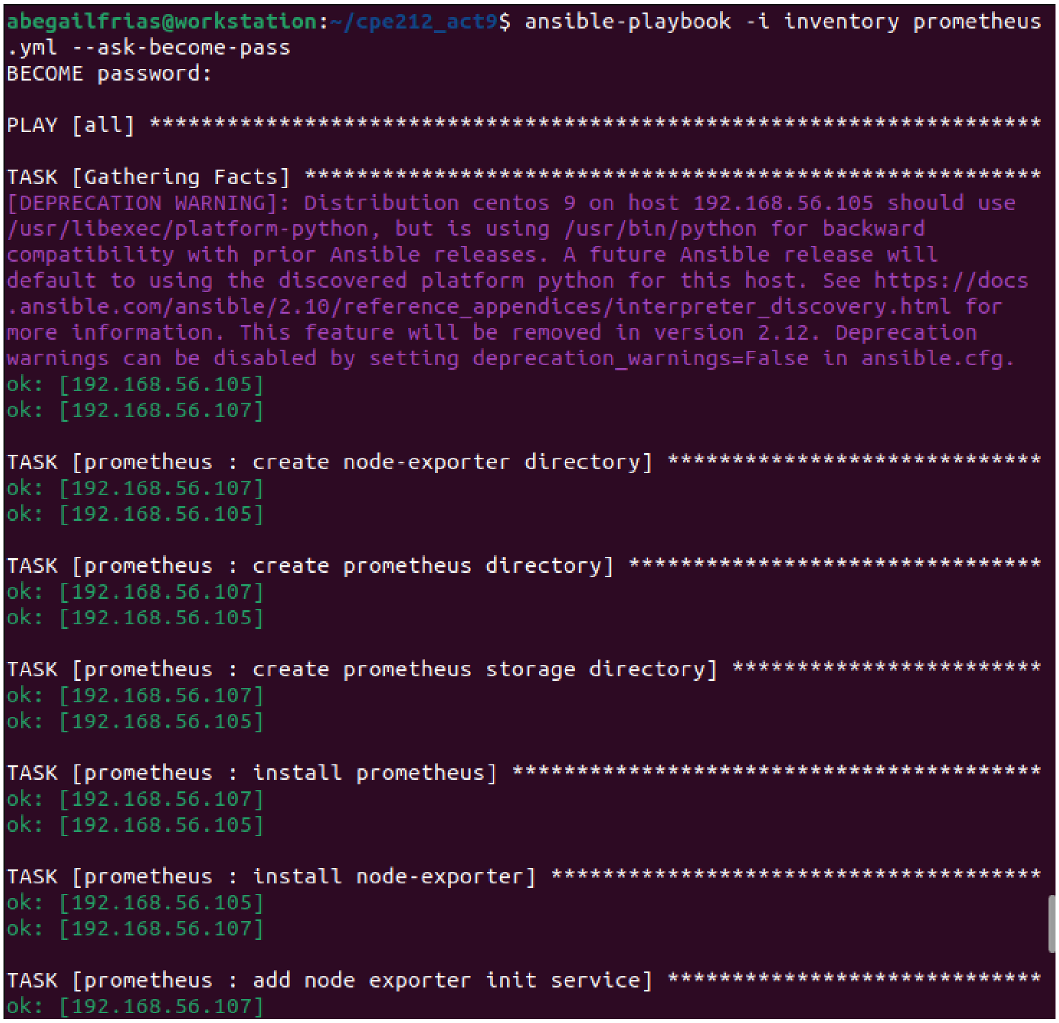


```
abegailfrias@workstation:~/cpe212_act9/prometheus$ cd tests
abegailfrias@workstation:~/cpe212_act9/prometheus/tests$ ls
inventory  test.yml
abegailfrias@workstation:~/cpe212_act9/prometheus/tests$ cat inventory
localhost
192.168.56.107
192.168.56.105
abegailfrias@workstation:~/cpe212_act9/prometheus/tests$
```

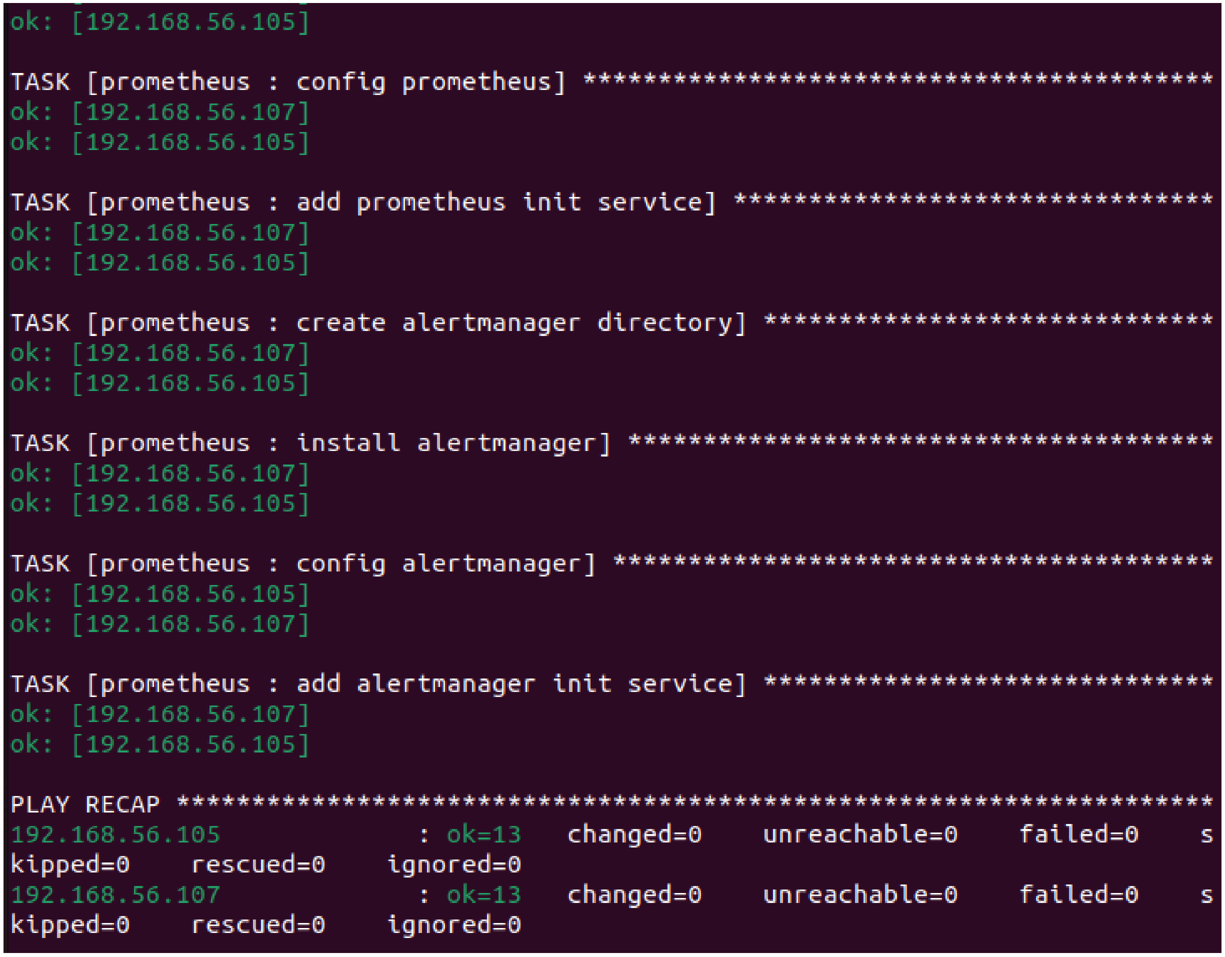


```
abegailfrias@workstation:~/cpe212_act9/prometheus/tests$ cat test.yml
---
- hosts: localhost
  remote_user: root
  roles:
    - prometheus
```

Step 11: Cd to test and edit the inventory and test.yml



```
abegailfrias@workstation:~/cpe212_act9$ ansible-playbook -i inventory prometheus
.yml --ask-become-pass
BECOME password:

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

TASK [Gathering Facts] *********************************************************
[DEPRECATION WARNING]: Distribution centos 9 on host 192.168.56.105 should use
/usr/libexec/platform-python, but is using /usr/bin/python for backward
compatibility with prior Ansible releases. A future Ansible release will
default to using the discovered platform python for this host. See https://docs
.ansible.com/ansible/2.10/reference_appendices/interpreter_discovery.html for
more information. This feature will be removed in version 2.12. Deprecation
warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
ok: [192.168.56.105]
ok: [192.168.56.107]

TASK [prometheus : create node-exporter directory] *****************************
ok: [192.168.56.107]
ok: [192.168.56.105]

TASK [prometheus : create prometheus directory] ********************************
ok: [192.168.56.107]
ok: [192.168.56.105]

TASK [prometheus : create prometheus storage directory] ************************
ok: [192.168.56.107]
ok: [192.168.56.105]

TASK [prometheus : install prometheus] *****************************************
ok: [192.168.56.107]
ok: [192.168.56.105]

TASK [prometheus : install node-exporter] **************************************
ok: [192.168.56.105]
ok: [192.168.56.107]

TASK [prometheus : add node exporter init service] *****************************
ok: [192.168.56.107]
```



```
ok: [192.168.56.105]

TASK [prometheus : config prometheus] *******************************************
ok: [192.168.56.107]
ok: [192.168.56.105]

TASK [prometheus : add prometheus init service] *********************************
ok: [192.168.56.107]
ok: [192.168.56.105]

TASK [prometheus : create alertmanager directory] *******************************
ok: [192.168.56.107]
ok: [192.168.56.105]

TASK [prometheus : install alertmanager] ****************************************
ok: [192.168.56.107]
ok: [192.168.56.105]

TASK [prometheus : config alertmanager] *****************************************
ok: [192.168.56.105]
ok: [192.168.56.107]

TASK [prometheus : add alertmanager init service] *******************************
ok: [192.168.56.107]
ok: [192.168.56.105]

PLAY RECAP **********************************************************************
192.168.56.105             : ok=13   changed=0    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
192.168.56.107             : ok=13   changed=0    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
```

This is the result.

github link: https://github.com/wonbe/cpe212_act9

Reflections:

Answer the following:

1. What are the benefits of having a performance monitoring tool?
    - By locating and resolving performance bottlenecks early on, performance monitoring solutions offer proactive issue identification, maintain system reliability, and save downtime. They allow quicker troubleshooting and data-driven decision-making through real-time analytics and historical data, while also optimizing resource utilization, which results in cost savings and increased application performance. Moreover, they facilitate security monitoring, scalability, and compliance, which improves user happiness and business continuity—particularly in intricate, expanding IT settings.

Conclusions:

- **In conclusion, despite encountering multiple challenges while creating the playbook for installing Prometheus on both Ubuntu and CentOS**

using roles, I successfully overcame them and completed the task. The process enhanced my understanding of Ansible playbooks, roles, and how to structure automation tasks efficiently. This experience also taught me the importance of troubleshooting and persistence when dealing with technical hurdles.