| | |
|---|---|
| **Name:** De Omampo, Julius Mark A. | **Date Performed:** 11/06/24 |
| **Course/Section:** CPE31S2 | **Date Submitted:** 11/06/24 |
| **Instructor:** Engr. Robin Valenzuela | **Semester and SY:** 1st (2024-2025) |

**Midterm Skills Exam: Install, Configure, and Manage Log Monitoring tools**

## 1. Objectives

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

## 2. Instructions

1. Create a repository in your GitHub account and label it CPE_MIDEXAM_SURNAME.
2. Clone the repository and do the following:
   2.1. Create an Ansible playbook that does the following with an input of a config.yaml file and arranged Inventory file:
   2.2. Install and configure Elastic Stack in separate hosts (Elastic Search, Kibana, Logstash) • Install Nagios in one host
   2.3. Install Grafana,Prometheus and Influxdb in seperate hosts (Influxdb,Grafana,Prometheus)
   2.4. Install Lamp Stack in separate hosts (Httpd + Php,Mariadb)
3. Document all your tasks using this document. Provide proofs of all the ansible playbooks codes and successful installations.
4. Document the push and commit from the local repository to GitHub.
5. Finally, paste also the link of your GitHub repository in the documentation.

## 3. Output (screenshots and explanations)

```
 GNU nano 6.2                                  ansible.cfg
[defaults]
inventory = ~/CPE_MIDEXAM_DEOMAMPO/inventory.yaml
remote_user = jmado
host_key_checking = True
```

*ansible.cfg File Configuration*

```
 GNU nano 6.2                                  inventory.yaml
[server1]
192.168.56.109

[server2]
192.168.56.110
```

*inventory.yaml File Configuration*

```
jnado@workstation:~/CPE_MIDEXAM_DEOMAMPO$ ls -Ra roles
roles:
.  ..  CentOS  Ubuntu

roles/CentOS:
.  ..  handlers  tasks

roles/CentOS/handlers:
.  ..  main.yml

roles/CentOS/tasks:
.  ..  main.yml

roles/Ubuntu:
.  ..  handlers  tasks

roles/Ubuntu/handlers:
.  ..  main.yml

roles/Ubuntu/tasks:
.  ..  main.yml
jnado@workstation:~/CPE_MIDEXAM_DEOMAMPO$
```

*roles Directory Contents*

```
 GNU nano 6.2                                    install.yaml
---

- hosts: all
  become: true
  pre_tasks:

  - name: Install Updates (Ubuntu)
    tags: always
    apt:
      upgrade: dist
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: Install Updates (CentOS)
    tags: always
    dnf:
      update_cache: yes
    when: ansible_distribution == "CentOS"

- hosts: server1
  become: true
  roles:
    - Ubuntu

- hosts: server2
  become: true
  roles:
    - CentOS
```

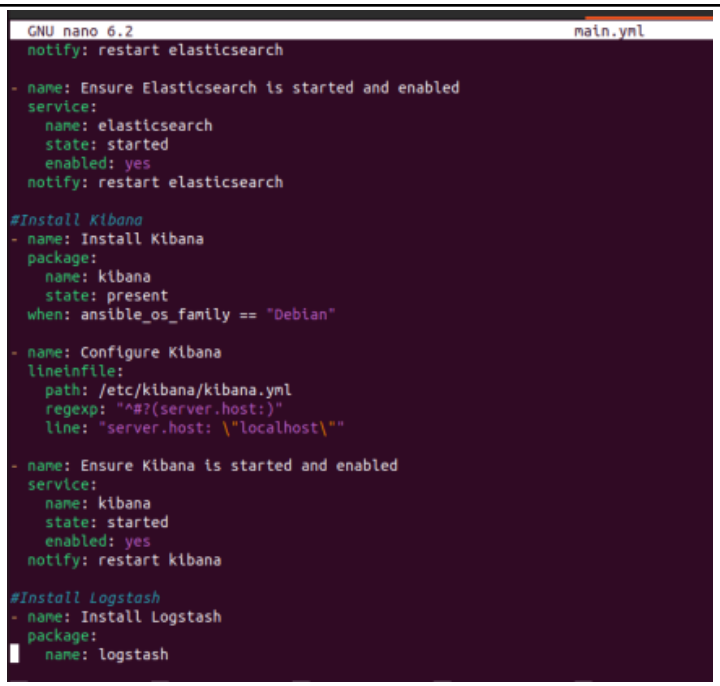
*install.yaml Playbook*

## I. Installing Elastic Stack in Separate Hosts (Elastic Search, Kibana, Logstash)

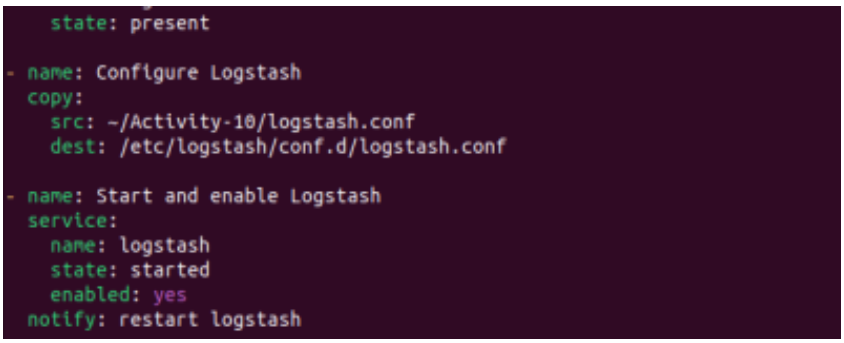
***Ubuntu - server1***

```
GNU nano 6.2                                                    main.yml
---

#Install Elastic Stack
- name: Ensure required GPG keys are added (Ubuntu)
  apt_key:
    url: https://artifacts.elastic.co/GPG-KEY-elasticsearch
    state: present
  when: ansible_os_family == "Debian"

- name: Ensure required repositories and keys are added (Ubuntu)
  apt_repository:
    repo: "deb https://artifacts.elastic.co/packages/8.x/apt stable main"
    state: present
    filename: elasticsearch
  when: ansible_os_family == "Debian"

- name: Update apt cache (Ubuntu)
  apt:
    update_cache: yes
  when: ansible_os_family == "Debian"

#Install ElasticSearch
- name: Install Elasticsearch
  package:
    name: elasticsearch
    state: present
  when: ansible_os_family == "Debian"

- name: Configure Elasticsearch
  lineinfile:
    path: /etc/elasticsearch/elasticsearch.yml
    regexp: "^#?(network.host:)"
    line: "network.host: localhost"
```

*Elastic Stack Ubuntu main.yml (1)*




*Elastic Stack Ubuntu main.yml (2)*




*Elastic Stack Ubuntu main.yml (3)*

```
GNU nano 6.2                                    roles/Ubuntu/handlers/main.yml
---

- name: restart elasticsearch
  service:
    name: elasticsearch
    state: restarted

- name: restart kibana
  service:
    name: kibana
    state: restarted

- name: restart logstash
  service:
    name: logstash
    state: restarted
```

*Elastic Stack Handlers Ubuntu main.yml*

***CentOS - server2***

```
GNU nano 6.2                                    roles/CentOS/tasks/main.yml
---

#Install Elastic Stack
- name: Ensure required repositories and keys are added (CentOS)
  yum_repository:
    name: logstash
    description: "Logstash repository"
    baseurl: "https://artifacts.elastic.co/packages/8.x/yum"
    gpgcheck: yes
    gpgkey: "https://artifacts.elastic.co/GPG-KEY-elasticsearch"
    enabled: yes
  when: ansible_os_family == "RedHat"

- name: Update package cache (CentOS)
  yum:
    name: "*"
    state: latest
  when: ansible_os_family == "RedHat"

#Install ElasticSearch
- name: Install Elasticsearch
  package:
    name: elasticsearch
    state: present
  when: ansible_os_family == "Debian"

- name: Configure Elasticsearch
  lineinfile:
    path: /etc/elasticsearch/elasticsearch.yml
    regexp: "^#?(network.host:)"
    line: "network.host: localhost"
  notify: restart elasticsearch
```

*Elastic Stack CentOS main.yml (1)*

```
GNU nano 6.2                                    roles/CentOS/tasks/main.yml
#Install Kibana
- name: Install Kibana
  package:
    name: kibana
    state: present
  when: ansible_os_family == "Debian"

- name: Configure Kibana
  lineinfile:
    path: /etc/kibana/kibana.yml
    regexp: "^#?(server.host:)"
    line: "server.host: \"localhost\""

- name: Ensure Kibana is started and enabled
  service:
    name: kibana
    state: started
    enabled: yes
  notify: restart kibana

#install Logstash
- name: Install Logstash
  package:
    name: logstash
    state: present

- name: Configure Logstash
  copy:
    src: ~/Activity-10/logstash.conf
    dest: /etc/logstash/conf.d/logstash.conf
```

*Elastic Stack CentOS main.yml (2)*

```
- name: Start and enable Logstash
  service:
    name: logstash
    state: started
    enabled: yes
  notify: restart logstash
```

*Elastic Stack CentOS main.yml (3)*

## II. Installing Nagios in One Host

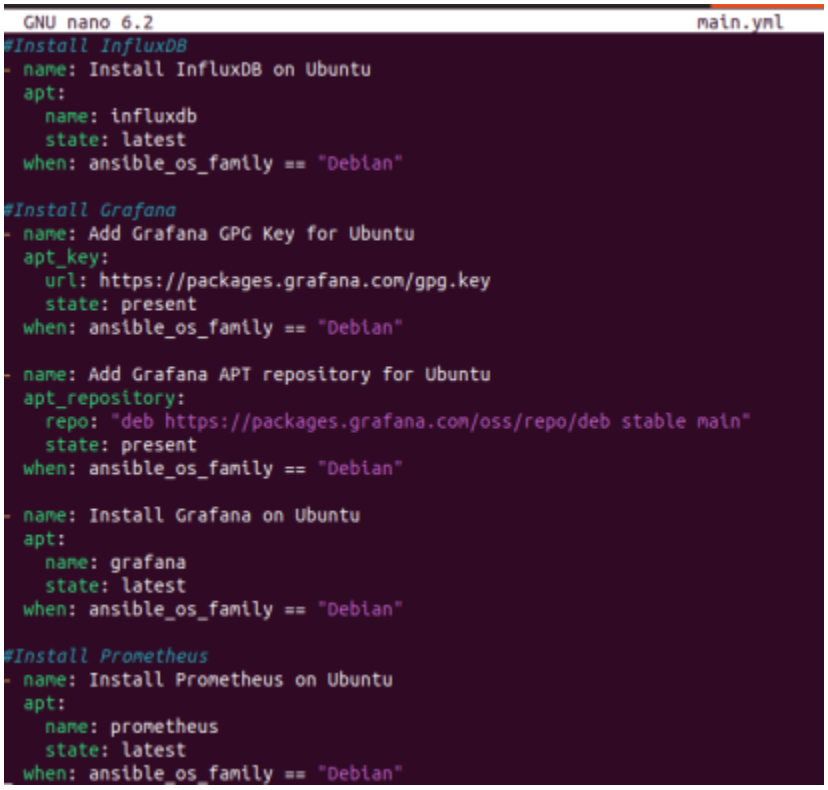
***Ubuntu - server1***

```
#Install Nagios on this server only(Ubuntu)
- name: Install Nagios (server1_ubuntu)
  tags: nagios, server1
  apt:
    name:
      - nagios4
      - nagios-plugins
      - nagios4-core
      - apache2
    update_chache: yes
    state: present
  when: ansible_os_family == "Debian"
```

*Elastic Stack Ubuntu main.yml*

## III. Installing Grafana, Prometheus, and InfluxDB in Separate Hosts

***Ubuntu - server1***




*Installing Grafana Prometheus InfluxDB Ubuntu main.yml*

***CentOS - server2***

```
GNU nano 6.2                                    roles/CentOS/tasks/main.yml
#Install EPEL release
- name: Install EPEL release (CentOS)
  yum:
    name: epel-release
    state: present
  when: ansible_os_family == "RedHat"

#Install InfluxDB
- name: Install InfluxDB on CentOS
  yum:
    name: influxdb
    state: latest
  when: ansible_os_family == "RedHat"

#Install Grafana
- name: Install Grafana on CentOS
  yum:
    name: grafana
    state: latest
  when: ansible_os_family == "RedHat"

#Install Prometheus
- name: Install Prometheus on CentOS
  yum:
    name: prometheus
    state: latest
  when: ansible_os_family == "RedHat"
```

*Install EPEL  InfluxDB Grafana Prometheus CentOS main.yml*

```
#Starting and Enabling InfluxDB, Grafana, Prometheus
- name: Start and enable InfluxDB
  service:
    name: influxdb
    state: started
    enabled: yes

- name: Start and enable Grafana
  service:
    name: grafana-server
    state: started
    enabled: yes

- name: Start and enable Prometheus
  service:
    name: prometheus
    state: started
    enabled: yes
```

*Starting and Enabling InfluxDB, Grafana, Prometheus CentOS main.yml*

## IV. Installing Lamp Stack in Separate Hosts (Apache/httpd, PHP, MariaDB)

### *Ubuntu - server1*

```
#Install Lamp Stack (apache/httpd, php, and MariaDB)
- name: Install Apache, PHP, and MariaDB on Ubuntu
  apt:
    name: "{{ item }}"
    state: present
  loop:
    - apache2
    - php
    - php-mysql
    - mariadb-server
  when: ansible_os_family == "Debian"

#Start and Enabling Lamp Stack
- name: Start and enable Apache on Ubuntu
  service:
    name: apache2
    state: started
    enabled: yes
  when: ansible_os_family == "Debian"

- name: Start and enable MariaDB on Ubuntu
  service:
    name: mysql
    state: started
    enabled: yes
  when: ansible_os_family == "Debian"
```

*Installing Starting and Enabling Lamp Stack Ubuntu main.yml*

### *CentOS - server2*

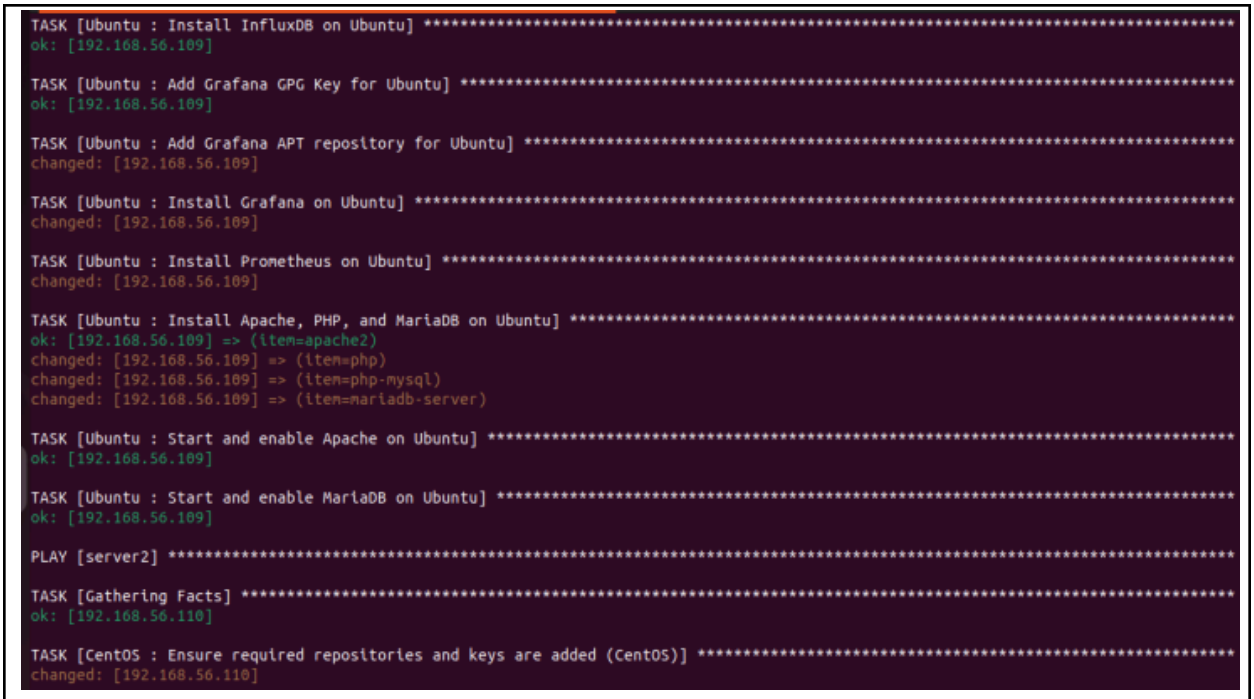
```
 GNU nano 6.2                                    roles/CentOS/tasks/main.yml
#Install Lamp Stack (apache/httpd, php, and MariaDB)
- name: Install Apache, PHP, and MariaDB on CentOS
  yum:
    name: "{{ item }}"
    state: present
  loop:
    - httpd
    - php
    - php-mysqlnd
    - mariadb-server
  when: ansible_os_family == "RedHat"

#Starting and Enabling Lamp Stack
- name: Start and enable Apache on CentOS
  service:
    name: httpd
    state: started
    enabled: yes
  when: ansible_os_family == "RedHat"

- name: Start and enable MariaDB on CentOS
  service:
    name: mariadb
    state: started
    enabled: yes
  when: ansible_os_family == "RedHat"

- name: Ensure php is installed (CentOS)
  yum:
    name: php
    state: present
  when: ansible_os_family == "RedHat"
```

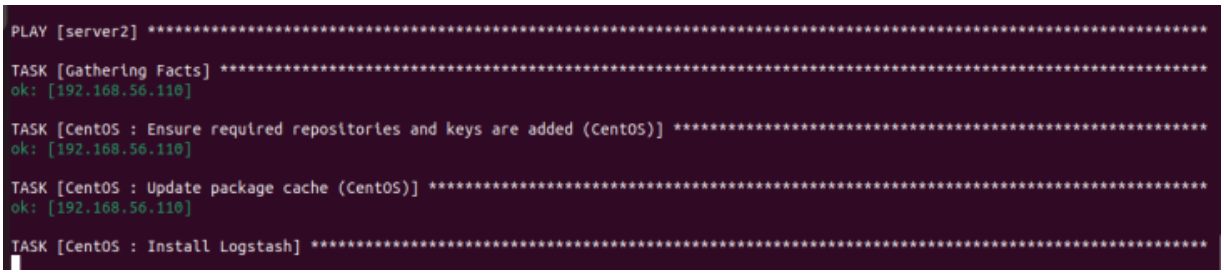
*Installing Starting and Enabling Lamp Stack CentOS main.yml*

## V. Playbook Log

```
jnado@workstation:~/CPE_MIDEXAM_DEOMAMPO$ ansible-playbook --ask-become-pass install.yaml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************
[DEPRECATION WARNING]: Distribution centos 9 on host 192.168.56.110 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: [192.168.56.110]
ok: [192.168.56.109]

TASK [Install Updates (Ubuntu)] ************************************************************************************
skipping: [192.168.56.110]
ok: [192.168.56.109]

TASK [Install Updates (CentOS)] ************************************************************************************
skipping: [192.168.56.109]
ok: [192.168.56.110]

PLAY [server1] *****************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.109]

TASK [Ubuntu : Ensure required GPG keys are added (Ubuntu)] ********************************************************
ok: [192.168.56.109]

TASK [Ubuntu : Ensure required repositories and keys are added (Ubuntu)] *******************************************
ok: [192.168.56.109]

TASK [Ubuntu : Update apt cache (Ubuntu)] **************************************************************************
changed: [192.168.56.109]
```

```
TASK [Ubuntu : Install Elasticsearch] ******************************************************************************
ok: [192.168.56.109]

TASK [Ubuntu : Configure Elasticsearch] ****************************************************************************
ok: [192.168.56.109]

TASK [Ubuntu : Ensure Elasticsearch is started and enabled] ********************************************************
ok: [192.168.56.109]

TASK [Ubuntu : Install Kibana] *************************************************************************************
ok: [192.168.56.109]

TASK [Ubuntu : Configure Kibana] ***********************************************************************************
ok: [192.168.56.109]

TASK [Ubuntu : Ensure Kibana is started and enabled] ***************************************************************
ok: [192.168.56.109]

TASK [Ubuntu : Install Logstash] ***********************************************************************************
ok: [192.168.56.109]

TASK [Ubuntu : Configure Logstash] *********************************************************************************
ok: [192.168.56.109]

TASK [Ubuntu : Start and enable Logstash] **************************************************************************
ok: [192.168.56.109]

TASK [Ubuntu : Install Nagios (server1_ubuntu)] ********************************************************************
ok: [192.168.56.109] => (item=nagios4)
ok: [192.168.56.109] => (item=nagios-plugins)
ok: [192.168.56.109] => (item=nagios4-core)
ok: [192.168.56.109] => (item=apache2)
```




**GitHub link:**
https://github.com/jmado-biscoff/CPE_MIDEXAM_DEOMAMPO.git

**Conclusions:** (link your conclusion from the objective)

In conclusion, the Ansible playbook successfully completed the deployment of essential monitoring and logging components across both Ubuntu and CentOS server nodes. This included the installation of the Elastic Stack—comprising Elasticsearch, Kibana, and Logstash—alongside NGINX for web serving, Grafana and Prometheus for metrics visualization, InfluxDB for time-series data storage, and the LAMP stack, which features Apache, PHP, and MariaDB for web application support. Each component was configured to ensure seamless operation and integration, providing a robust infrastructure for data collection, analysis, and application hosting. The standardized approach using Ansible not only streamlined the installation process but also ensured consistency and repeatability across the server environments.