Name: Jozshua Amiel Alonzo	Date Performed: October 27, 2022
Course/Section: CPE31S23	Date Submitted:
Instructor: Dr. Jonathan Taylar	Semester and SY: 2022 - 2023

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

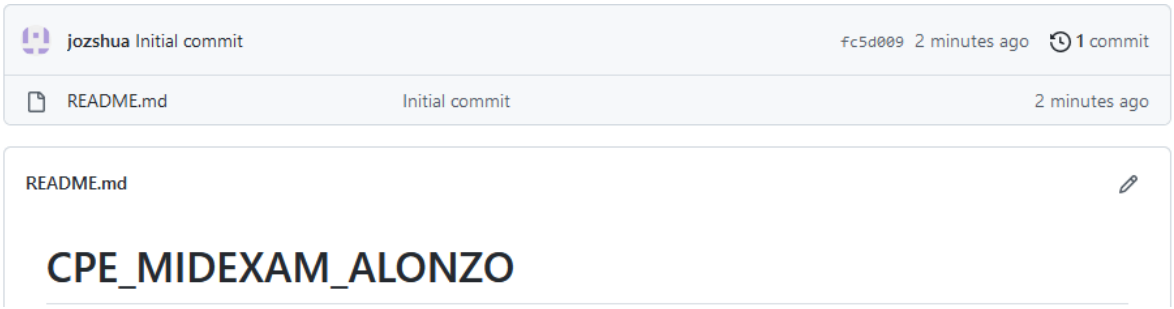
1. Objectives

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)



Logged in on the personal github account and created the new repository.

```
jozshua@jozshua-VirtualBox:~$ git clone git@github.com:jozshua/CPE_MIDEXAM_ALONZ
O.git
Cloning into 'CPE_MIDEXAM_ALONZO'...
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.
```

Used git clone command to clone the repository to the local machine.

```
jozshua@jozshua-VirtualBox:~/CPE_MIDEXAM_ALONZO/CPE_MEXAM$ ls
ansible.cfg  config.yaml  inventory  roles
```

```
  GNU nano 6.2                                config.yaml
---
- hosts: all
  become: true
  pre_tasks:

  - name: install updates (Ubuntu)
    tags: always
    apt:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "Ubuntu"

  - name: update repository index (CentOS)
    tags: always
    dnf:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "CentOS"
```

Created the config.yaml and inventory file inside the cloned repository.

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.116]
ok: [192.168.56.112]

TASK [install updates (Ubuntu)] ************************************************
*
skipping: [192.168.56.116]
ok: [192.168.56.112]

TASK [update repository index (CentOS)] ****************************************
*
skipping: [192.168.56.112]
ok: [192.168.56.116]

PLAY RECAP *********************************************************************
*
192.168.56.112             : ok=2    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.116             : ok=2    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
```

Run the config.yaml file: Both remote servers successfully executed the tasks. There are 2 skipped states because of having a different ansible distribution on the command from the config.yaml file.

Installation of Elastic stack:

```
  GNU nano 6.2                                  main.yml
    - name: install Elastik Stack (Ubuntu)
      apt:
        name:
          - elasticsearch
          - kibana
          - logstash
        state: latest
        update_cache: yes
      when: ansible_distribution == "Ubuntu"

    - name: install Elastik Stack (CentOS)
      dnf:
        name:
          - elasticsearch
          - kibana
          - logstash
        state: latest
        update_cache: yes
      when: ansible_distribution == "CentOS"
```

```
TASK [prome : install prometheus in ubuntu] ***********************************
*
skipping: [192.168.56.110]
ok: [192.168.56.106]

TASK [prome : install prometheus requisites for centos] ***********************
*
skipping: [192.168.56.106]
ok: [192.168.56.110]

TASK [prome : enabling sockets for centos] *************************************
*
skipping: [192.168.56.106]
changed: [192.168.56.110]

TASK [prome : finishing installation of prometheus in centos] ******************
*
skipping: [192.168.56.106]
changed: [192.168.56.110]
```

Create the main.yml file for the nagios inside of the tasks directory. There are 2 successfully executed tasks, 2 changed states and 4 skipped states from the different ansible distribution.

Proof of successfully installation:

```
jozshua@server1-VirtualBox:~$ sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendo
     Active: active (running) since Wed 2022-10-26 11:19:37 PST; 10min ago
       Docs: https://www.elastic.co
   Main PID: 11692 (java)
      Tasks: 57 (limit: 1080)
     Memory: 351.1M
        CPU: 45.169s
     CGroup: /system.slice/elasticsearch.service
             ├─11692 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.n
             └─11850 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux

Oct 26 11:18:52 Server1 systemd[1]: Starting Elasticsearch...
Oct 26 11:19:37 Server1 systemd[1]: Started Elasticsearch.
```

```
jozshua@server1-VirtualBox:~$ sudo systemctl status kibana
● kibana.service - Kibana
     Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor prese
     Active: active (running) since Wed 2022-10-26 11:35:01 PST; 25s ago
       Docs: https://www.elastic.co
   Main PID: 12373 (node)
      Tasks: 11 (limit: 1080)
     Memory: 263.7M
        CPU: 13.254s
     CGroup: /system.slice/kibana.service
             └─12373 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/

Oct 26 11:35:01 Server1 systemd[1]: Started Kibana.
```

```
jozshua@server1-VirtualBox:~$ sudo systemctl status logstash
● logstash.service - logstash
     Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor pre>
     Active: active (running) since Wed 2022-10-26 11:36:29 PST; 21s ago
   Main PID: 12502 (java)
      Tasks: 14 (limit: 1080)
     Memory: 209.8M
        CPU: 8.036s
     CGroup: /system.slice/logstash.service
             └─12502 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseCo>

Oct 26 11:36:29 Server1 systemd[1]: Started logstash.
Oct 26 11:36:29 Server1 logstash[12502]: Using bundled JDK: /usr/share/logstas>
Oct 26 11:36:30 Server1 logstash[12502]: OpenJDK 64-Bit Server VM warning: Opt>
```

From CentOS:

```
[jozshua@localhost ~]$ sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; vendor prese
t: disabled)
   Active: active (running) since Wed 2022-10-26 13:26:34 PST; 6min ago
     Docs: https://www.elastic.co
 Main PID: 3274 (java)
   CGroup: /system.slice/elasticsearch.service
           ├─3274 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.networkadd...
           └─3445 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/b...

Oct 26 13:25:38 localhost.localdomain systemd[1]: Starting Elasticsearch...
Oct 26 13:26:34 localhost.localdomain systemd[1]: Started Elasticsearch.
```

```
[jozshua@localhost ~]$ sudo systemctl status kibana
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: disabled
)
   Active: active (running) since Wed 2022-10-26 13:33:45 PST; 21s ago
     Docs: https://www.elastic.co
 Main PID: 4301 (node)
    Tasks: 11
   CGroup: /system.slice/kibana.service
           └─4301 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/../sr...

Oct 26 13:33:45 localhost.localdomain systemd[1]: Started Kibana.
```

```
[jozshua@localhost ~]$ sudo systemctl status logstash
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: disabl
ed)
   Active: active (running) since Wed 2022-10-26 13:35:33 PST; 46s ago
 Main PID: 4438 (java)
   CGroup: /system.slice/logstash.service
           └─4438 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSwe...

Oct 26 13:35:33 localhost.localdomain systemd[1]: Started logstash.
Oct 26 13:35:33 localhost.localdomain logstash[4438]: Using bundled JDK: /usr/share...k
Oct 26 13:35:34 localhost.localdomain logstash[4438]: OpenJDK 64-Bit Server VM warn....
Hint: Some lines were ellipsized, use -l to show in full.
```

Issue to command sudo systemctl status with the package.

Installation of Nagios:

```
  GNU nano 6.2                                  main.yml
- name: nagios installation (Ubuntu)
  apt:
    name:
      - nagios4
    state: latest
    update_cache: yes
  when: ansible_distribution == "Ubuntu"

- name: nagios installation (CentOS)
  dnf:
    name:
      - nagios
    state: latest
    update_cache: yes
  when: ansible_distribution == "CentOS"
```

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.112]
ok: [192.168.56.116]

TASK [nagios : nagios installation (Ubuntu)] ***********************************
*
skipping: [192.168.56.116]
ok: [192.168.56.112]

TASK [nagios : nagios installation (CentOS)] ***********************************
*
skipping: [192.168.56.112]
ok: [192.168.56.116]
```

Create the main.yml file for the nagios inside of the tasks directory. There are 4 successfully executed tasks and 2 skipped states from the different ansible distribution.

Proof of successfully installation:

```
jozshua@Server1-VirtualBox:~$ nagios4 -- version

Nagios Core 4.4.6
Copyright (c) 2009-present Nagios Core Development Team and Community Contribut
ors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2020-04-28
License: GPL
```

```
[jozshua@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
```

From issuing the command nagios –version you can display the version of the package that it is already installed.

Installation of Prometheus:

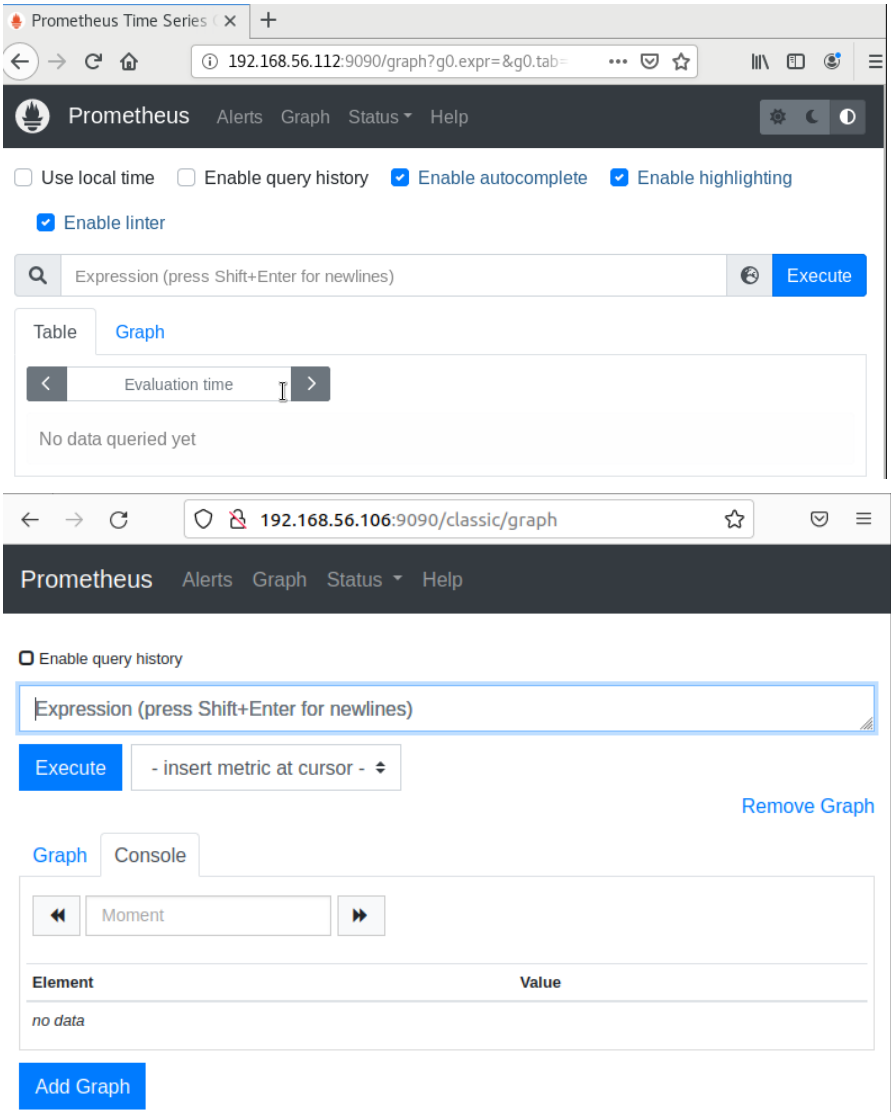
```
  GNU nano 6.2                                  main.yml *
- name: prometheus installation (Ubuntu)
  apt:
    name:
      - prometheus
    state: latest
    update_cache: yes
  when: ansible_distribution == "Ubuntu"

- name: Required installlation of prometheus (CentOS)
  tags: centos,snapd,epel-release
  yum:
    name:
      - epel-release
      - snapd
    state: latest
  when: ansible_distribution == "CentOS"

- name: Allowing snapd (CentOS)
  tags: snapd,centos
  command: systemctl enable --now snapd.socket
  when: ansible_distribution == "CentOS"
```

```
- name: prometheus installation (CentOS)
  tags: centos,prometheus
  command: snap install prometheus --classic
  when: ansible_distribution == "CentOS"
```

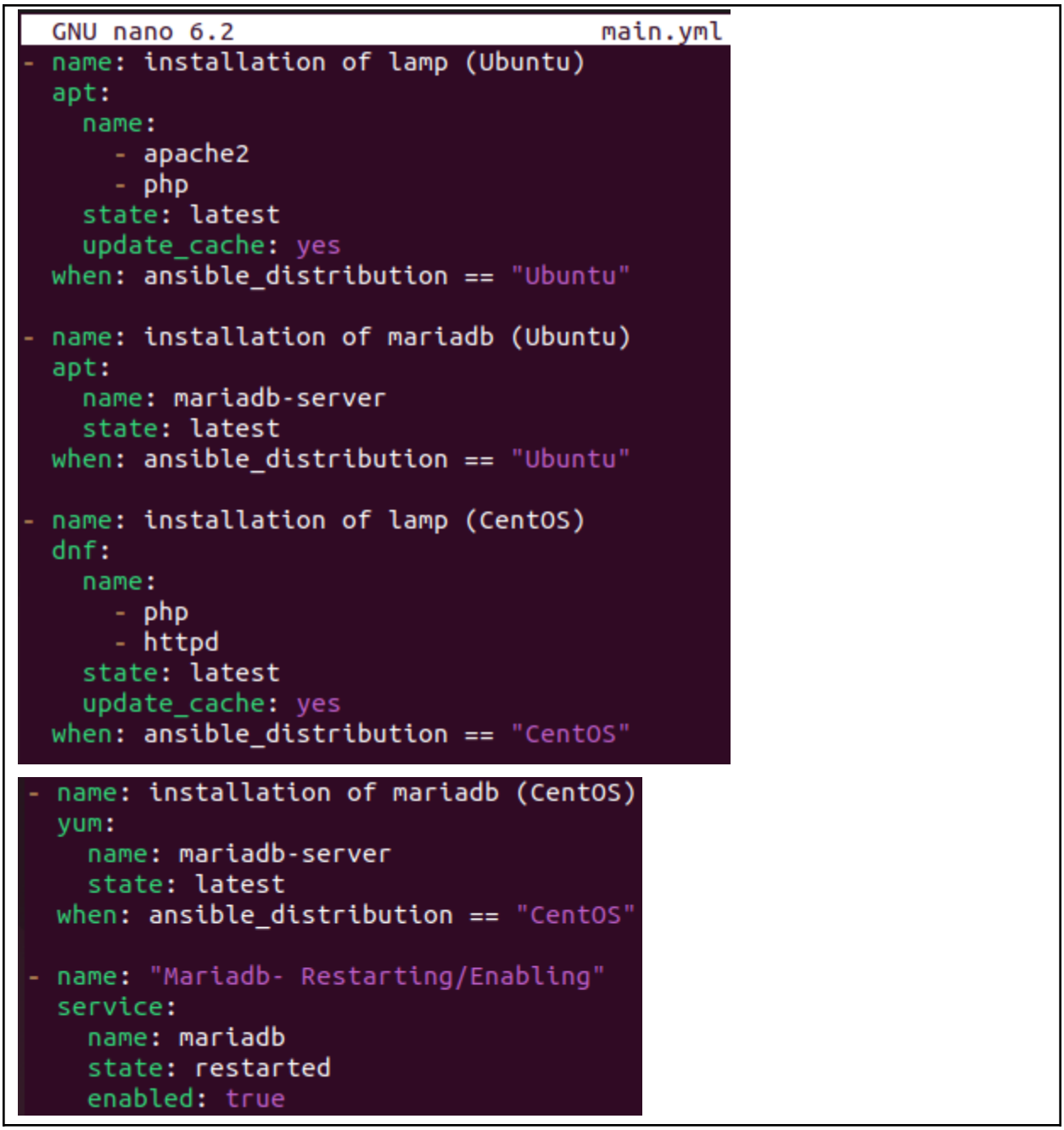
```
TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.106]
ok: [192.168.56.112]

TASK [prometheus : Prometheus Installation (Ubuntu)] ***************************************
skipping: [192.168.56.112]
ok: [192.168.56.106]

TASK [prometheus : Pre-requisite of Installation for CentOS] *******************************
skipping: [192.168.56.106]
ok: [192.168.56.112]

TASK [prometheus : Enabling Snapd Sockets for CentOS] **************************************
skipping: [192.168.56.106]
changed: [192.168.56.112]

TASK [prometheus : Prometheus Installation (CentOS)] ***************************************
skipping: [192.168.56.106]
changed: [192.168.56.112]
```

Created the main.yml file for the prometheus inside of the tasks directory. There are 4 successfully executed tasks and 4 skipped states because of having the different ansible distribution from running the main yml file from the prometheus.

Proof of successfully installation:



Installation of Lampstack:



```
  GNU nano 6.2                                  main.yml
- name: installation of lamp (Ubuntu)
  apt:
    name:
      - apache2
      - php
    state: latest
    update_cache: yes
  when: ansible_distribution == "Ubuntu"

- name: installation of mariadb (Ubuntu)
  apt:
    name: mariadb-server
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: installation of lamp (CentOS)
  dnf:
    name:
      - php
      - httpd
    state: latest
    update_cache: yes
  when: ansible_distribution == "CentOS"

- name: installation of mariadb (CentOS)
  yum:
    name: mariadb-server
    state: latest
  when: ansible_distribution == "CentOS"

- name: "Mariadb- Restarting/Enabling"
  service:
    name: mariadb
    state: restarted
    enabled: true
```

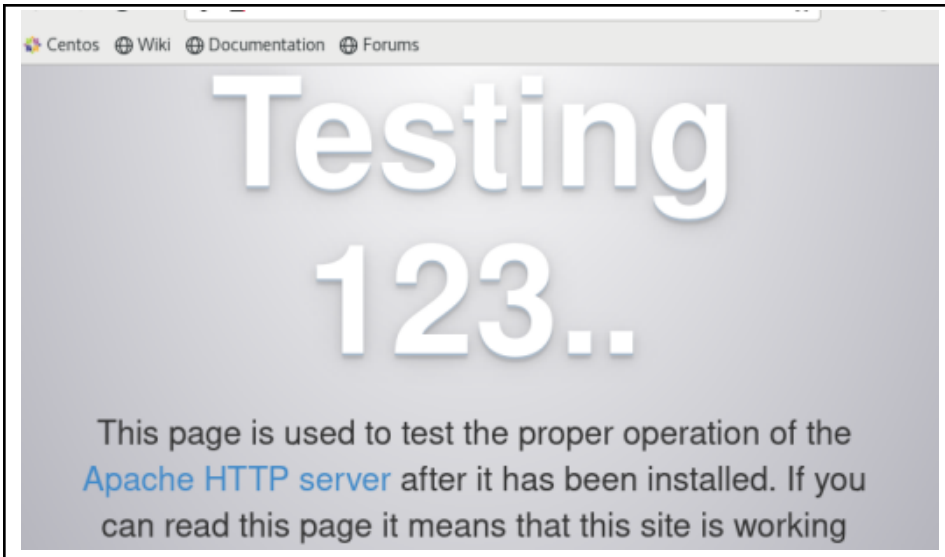
```
PLAY [all] *********************************************************************
*

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.112]
ok: [192.168.56.116]

TASK [lamp : installation of lamp (Ubuntu)] ************************************
*
skipping: [192.168.56.116]
changed: [192.168.56.112]

TASK [lamp : installation of mariadb (Ubuntu)] *********************************
*
skipping: [192.168.56.116]
ok: [192.168.56.112]

TASK [lamp : installation of lamp (CentOS)] ************************************
*
skipping: [192.168.56.112]
ok: [192.168.56.116]

TASK [lamp : installation of mariadb (CentOS)] *********************************
*
skipping: [192.168.56.112]
changed: [192.168.56.116]
```

```
TASK [lamp : Mariadb- Restarting/Enabling] **************************************
*
changed: [192.168.56.116]
changed: [192.168.56.112]

PLAY RECAP *********************************************************************
*
192.168.56.112             : ok=6    changed=2    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
192.168.56.116             : ok=6    changed=2    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
```

Created the main.yml file for the lamp stack inside of the tasks directory. There are 6 successfully executed tasks, 2 changed states and 3 skipped states because of having the different ansible distribution from running the main yml file from the lamp stack.

Proof of successfully installed:

```
jozshua@Server1-VirtualBox:~$ systemctl status mariadb
● mariadb.service - MariaDB 10.6.7 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor pres>
     Active: active (running) since Thu 2022-10-06 10:59:01 PST; 4min 13s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 51852 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /v>
    Process: 51853 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_>
    Process: 51855 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] &>
    Process: 51895 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP>
    Process: 51897 ExecStartPost=/etc/mysql/debian-start (code=exited, status=>
   Main PID: 51884 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 8 (limit: 1080)
     Memory: 59.3M
        CPU: 490ms
     CGroup: /system.slice/mariadb.service
             └─51884 /usr/sbin/mariadbd
```



From the screenshot the checking of the installation of the package from mariadb and httpd.

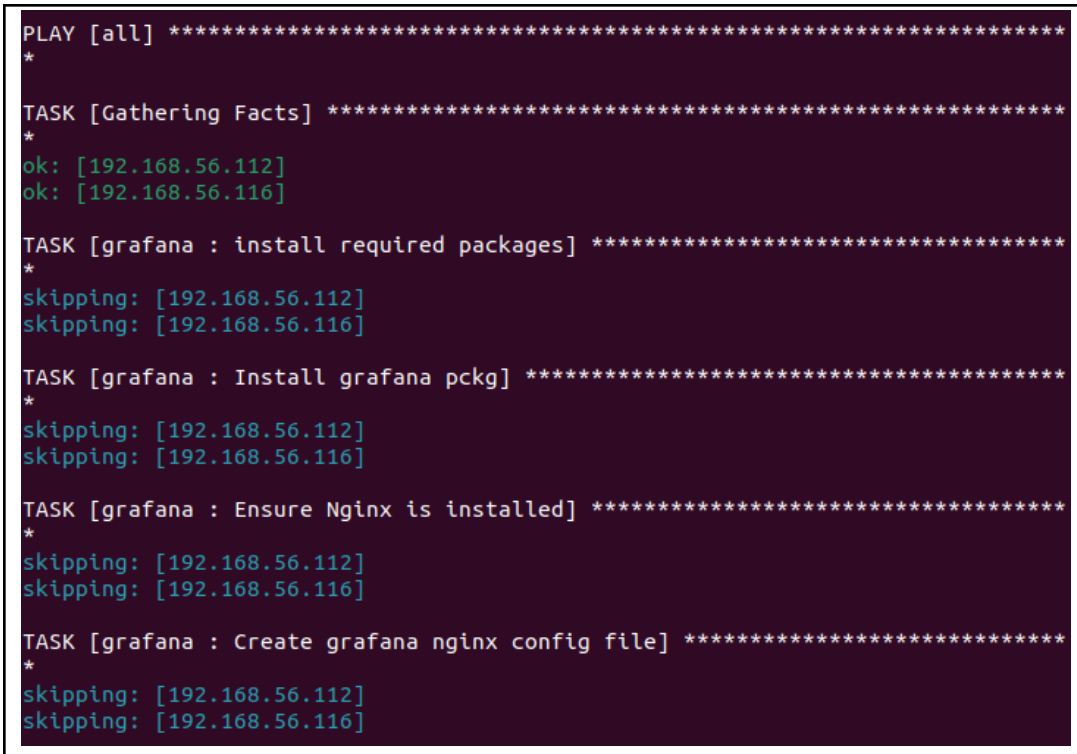
Installation of grifana:

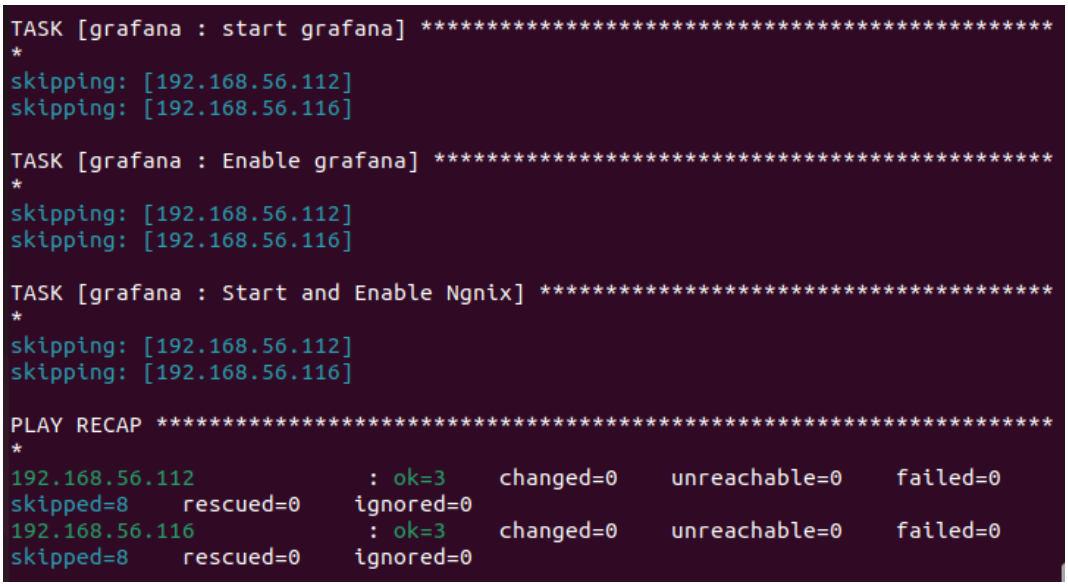
```
---

- name: install required packages
  apt:
    name:
      - gnupg2
      - curl
      - software-properties-common
    state: present
  when: ansible_distribution == "ubuntu"

- name: Install grafana pckg
  apt:
    name: grafana-server
    state: present
    update_cache: yes
  notify:
    - Start grafana
    - Enable grafana
  when: ansible_distribution == "ubuntu"

- name: Ensure Nginx is installed
  apt:
    name: nginx
    state: present
```

```
  notify: Start and Enable Nginx
  when: ansible_distribution == "ubuntu"

- name: Create grafana nginx config file
  copy:
      dest: /etc/nginx/conf.d/grafana.conf
      mode: 0755
      content: |
        server {
          listen 80;
          server_tokens off;
          client_max_body_size 10M;
          server_name grafana.citizix.com;

          ignore_invalid_headers off;

          if ($host !~* ^(grafana.citizix.com)$ ) {
              return 444;
          }

          location / {
              send_timeout            600;
              proxy_read_timeout      600;
              proxy_send_timeout      600;
              proxy_connect_timeout   600;
```

```
              proxy_redirect          off;
              proxy_set_header        Host $host;
              proxy_set_header        X-Real-IP $remote_addr;
              proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_fo>
              proxy_set_header        X-Forwarded-Host $server_name;
              proxy_set_header        X-Forwarded-Proto $Scheme;
              proxy_pass http:/127.0.0.1:3000;
          }
      }
  when: ansible_distribution == "ubuntu"

- name: start grafana
  systemd:
    name: grafana-server
    state: started
  when: ansible_distribution == "ubuntu"

- name: Enable grafana
  systemd:
    name: grafana-server
    enabled: yes
  when: ansible_distribution == "ubuntu"
```

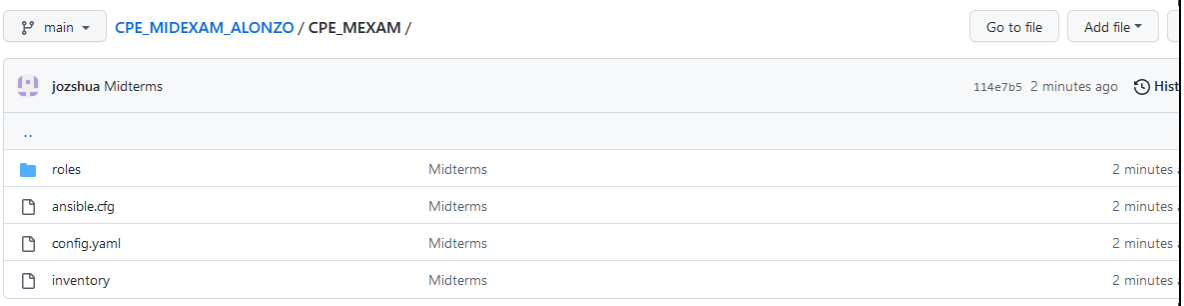
```
- name: Start and Enable Ngnix
  systemd:
    name: nginx
    state: started
    enabled: yes
  when: ansible_distribution == "ubuntu"
```





Created the main.yml file for the lamp stack inside of the tasks directory. There are 3 successfully executed tasks and 8 skipped states because of having the different ansible distribution from running the main yml file from the grafana.

GitHub link: https://github.com/jozshua/CPE_MIDEXAM_ALONZO.git



Conclusions: (link your conclusion from the objective)

In this exam activity, Using ansible as an Infrastructure as Code tool I created and designed the workflow of installing the nagios, prometheus, elastic stack for both server distributions. Installing the nagios, prometheus, elastic stack, grafana, and lampstack for both server distributions that needs a site.yml file and main.yml file. Inside these files there are codes that are necessary for executing the tasks for the ansible playbooks for the output. After running the playbook successfully I am able to display the installation of nagios for both servers. From Nagios I used the nagios –version command, for prometheus I used the browser and typed the ip address, for elastic stack I used the sudo systemctl with the different package to check if it is active and for lamp stack I used systemctl mariadb and used the IP address from the browser of CentOS. Lastly, I make sure that all files that I created for this exam activity were committed to my Github repository.