| | |
|---|---|
| **Name:** Ian Carlo T. Bello | **Date Performed:** October 30, 2022 |
| **Course/Section:** CPE31S24 | **Date Submitted:** October 30, 2022 |
| **Instructor:** Dr. Jonathan V. Taylar | **Semester and SY:** 1st sem – 3rd year |

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

## 1. Objectives

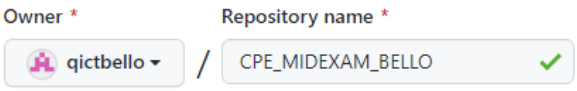
Create and design a workflow that installs, configure and manage enterprise availability, performance and log monitoring tools using Ansible as an Infrastructure as Code (IaC) tool.

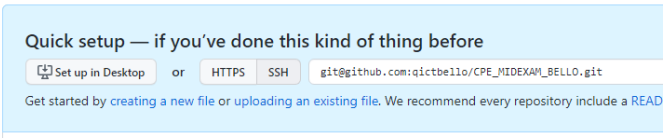
## 2. Instructions

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.

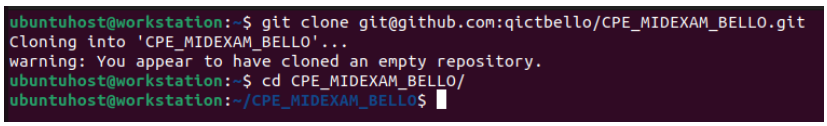
## 3. Output (screenshots and explanations)

First, we will create our repository we will name it CPE_MIDEXAM_BELLO




After creating we will clone it into our Control Node




Next, we will create the ansible config and inventory file

```
 GNU nano 6.2                                                    inventory
[ubuntu]
server1

[centos]
servercent
```

```
 GNU nano 6.2                                                    ansible.cfg
[defaults]
timeout=500
inventory = inventory
private_key_file = ~/.ssh/ansible
```

We need to create roles for each task, the first task is elastic stack since we are already done with elastic stack, we just need to get the roles from the previous activity

```
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO$ cp ~/CPE232ELK/roles .
cp: -r not specified; omitting directory '/home/ubuntuhost/CPE232ELK/roles'
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO$ cp -r ~/CPE232ELK/roles .
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO$ ls
ansible.cfg  inventory  roles
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO$ ls roles
centoselk  ubuntuelk
```

Now we both have elastic for both host we will also provide the contents of it using GitHub link of the roles. Next, we will be installing Nagios for one host we only need to remove one host from the previous activity

```
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO$ cd roles
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO/roles$ cp -r ~/CPE232Nagios/roles/nagios .
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO/roles$ ls
centoselk  nagios  ubuntuelk
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO/roles$
```

```
 GNU nano 6.2
- name: Install nagios in Ubuntu
  apt:
    name:
      - nagios4
    state: latest
    update_cache: yes
  when: ansible_distribution == "Ubuntu"
```

We will remove the centos so only ubuntu host will install. Next, we need to create roles for Grafana, Prometheus and Influxdb to install in both hosts.

```
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO/roles$ mkdir -p igp/tasks
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO/roles$ nano igp/tasks/main.yml
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO/roles$
```

We created tasks for both servers to install IGP we will include the screenshots of yml with link at the bottom. Next, we will create roles for lamp stack for separate host.

```
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO/roles$ mkdir -p lampstack/tasks
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO/roles$ nano lampstack/tasks/main.yml
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO/roles$ ls
centoselk  igp  lampstack  nagios  ubuntuelk
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO/roles$
```

Contents are in the GitHub repository. Next, we will create test in each task first before running it in one file.

```
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO$ cp ~/CPE232ELK/elk.yml .
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO$ ls
ansible.cfg  elk.yml  inventory  roles
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO$
```

We will run the ELK first

```
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO$ ansible-playbook --ask-become-pass elk.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [server1]
ok: [servercent]

TASK [update repository index CentOS] ******************************************
skipping: [server1]
ok: [servercent]

TASK [install updates Ubuntu] **************************************************
skipping: [servercent]
ok: [server1]

TASK [install unzip] ***********************************************************
ok: [server1]
ok: [servercent]

PLAY [ubuntu] ******************************************************************

TASK [Gathering Facts] *********************************************************
ok: [server1]

TASK [ubuntuelk : Install ELK Prereq Ubuntu] ***********************************
ok: [server1]

TASK [ubuntuelk : Get PGP Key Ubuntu] ******************************************
ok: [server1]
```

```
TASK [ubuntuelk : Install Elasticsearch repository into sources list Ubuntu] ************
ok: [server1]

TASK [ubuntuelk : Install Elasticsearch Ubuntu] ****************************************
ok: [server1]

TASK [ubuntuelk : Configure Elasticsearch change cluster name Ubuntu] ******************
ok: [server1]

TASK [ubuntuelk : Configure Elasticsearch give cluster descriptive name Ubuntu] *********
ok: [server1]

TASK [ubuntuelk : Configure Elasticsearch Add network.host Ubuntu] *********************
ok: [server1]

TASK [ubuntuelk : Configure Elasticsearch Add http.port Ubuntu] ************************
ok: [server1]

TASK [ubuntuelk : Configure Elasticsearch Add discovery.type Ubuntu] *******************
ok: [server1]

TASK [ubuntuelk : Creating an empty file for startup-timeout.conf 1 of 2 Ubuntu] ********
ok: [server1]

TASK [ubuntuelk : Creating an empty file for startup-timeout.conf 2 of 2 Ubuntu] ********
changed: [server1]

TASK [ubuntuelk : Prevent systemd service start operation from timing out Ubuntu] ********
ok: [server1]

TASK [ubuntuelk : Run daemon-reload for elasticsearch Ubuntu] **************************
ok: [server1]
```

```
TASK [ubuntuelk : Enable service Elasticsearch and ensure it is not masked Ubuntu] *
ok: [server1]

TASK [ubuntuelk : ensure elasticsearch is running Ubuntu] ************************
changed: [server1]

TASK [ubuntuelk : Install Logstash (Ubuntu)] *************************************
ok: [server1]

TASK [ubuntuelk : Run daemon-reload for logstash Ubuntu] *************************
ok: [server1]

TASK [ubuntuelk : Enable service logstash Ubuntu] ********************************
ok: [server1]

TASK [ubuntuelk : ensure logstash is running Ubuntu] *****************************
ok: [server1]

TASK [ubuntuelk : Install Kibana Ubuntu] *****************************************
ok: [server1]

TASK [ubuntuelk : Configure Kibana Add server.port Ubuntu] ***********************
ok: [server1]

TASK [ubuntuelk : Configure Kibana Add server.host Ubuntu] ***********************
ok: [server1]

TASK [ubuntuelk : Configure Kibana Add server.name Ubuntu] ***********************
ok: [server1]
```

```
TASK [ubuntuelk : Configure Kibana Add elasticsearch.hosts Ubuntu] ****
ok: [server1]

TASK [ubuntuelk : Run daemon-reload for kibana Ubuntu] ***************
ok: [server1]

TASK [ubuntuelk : Enable service Kibana Ubuntu] **********************
ok: [server1]

TASK [ubuntuelk : Start Elasticsearch service] ***********************
changed: [server1]

TASK [ubuntuelk : Start Kibana] **************************************
changed: [server1]

PLAY [centos] ********************************************************

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

TASK [centoselk : Install ELK Prereq CentOs] *************************
ok: [servercent]

TASK [centoselk : install elasticsearch rpm key CentOs] **************
ok: [servercent]

TASK [centoselk : install elasticsearch 7.x rpm repository] **********
ok: [servercent]

TASK [centoselk : Install Elasticsearch CentOs] **********************
ok: [servercent]
```

```
TASK [centoselk : Configure Elasticsearch change cluster name CentOs] ************************************************************
ok: [servercent]

TASK [centoselk : Configure Elasticsearch give cluster descriptive name CentOs] **************************************************
ok: [servercent]

TASK [centoselk : Configure Elasticsearch Add network.host CentOs] ***************************************************************
ok: [servercent]

TASK [centoselk : Configure Elasticsearch Add http.port CentOs] ******************************************************************
ok: [servercent]

TASK [centoselk : Configure Elasticsearch Add discovery.type CentOs] *************************************************************
ok: [servercent]

TASK [centoselk : Creating an empty file for startup-timeout.conf 1 of 2 CentOs] *************************************************
ok: [servercent]

TASK [centoselk : Creating an empty file for startup-timeout.conf 2 of 2 CentOs] *************************************************
changed: [servercent]

TASK [centoselk : Prevent systemd service start operation from timing out CentOs] ************************************************
ok: [servercent]

TASK [centoselk : Run daemon-reload for elasticsearch CentOs] ********************************************************************
fatal: [servercent]: FAILED! => {"changed": false, "msg": "failure 1 during daemon-reload: Failed to execute operation: Connection timed out\n
"}

PLAY RECAP ***********************************************************************************************************************
server1                    : ok=32   changed=4    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
servercent                 : ok=16   changed=1    unreachable=0    failed=1    skipped=1    rescued=0    ignored=0
```

Due to loss of connection, I got this error. Ubuntu is successful so we only need to run the centoselk.

```
#- hosts: ubuntu
#  become: true
#  roles:
#    - ubuntuelk

- hosts: centos
  become: true
  roles:
    - centoselk
```

```
PLAY [centos] ******************************************************************

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

TASK [centoselk : Install ELK Prereq CentOs] ***********************************
ok: [servercent]

TASK [centoselk : install elasticsearch rpm key CentOs] ************************
ok: [servercent]

TASK [centoselk : install elasticsearch 7.x rpm repository] ********************
ok: [servercent]

TASK [centoselk : Install Elasticsearch CentOs] ********************************
ok: [servercent]

TASK [centoselk : Configure Elasticsearch change cluster name CentOs] ************
ok: [servercent]

TASK [centoselk : Configure Elasticsearch give cluster descriptive name CentOs] **
ok: [servercent]

TASK [centoselk : Configure Elasticsearch Add network.host CentOs] ***************
ok: [servercent]

TASK [centoselk : Configure Elasticsearch Add http.port CentOs] ******************
ok: [servercent]

TASK [centoselk : Configure Elasticsearch Add discovery.type CentOs] *************
ok: [servercent]
```

```
TASK [centoselk : Creating an empty file for startup-timeout.conf 1 of 2 CentOs] *
ok: [servercent]

TASK [centoselk : Creating an empty file for startup-timeout.conf 2 of 2 CentOs] *
changed: [servercent]

TASK [centoselk : Prevent systemd service start operation from timing out CentOs]
ok: [servercent]

TASK [centoselk : Run daemon-reload for elasticsearch CentOs] ********************
ok: [servercent]

TASK [centoselk : Enable service Elasticsearch and ensure it is not masked CentOs]
ok: [servercent]

TASK [centoselk : ensure elasticsearch is running for CentOs] ********************
ok: [servercent]

TASK [centoselk : Install Logstash CentOs] ***************************************
ok: [servercent]

TASK [centoselk : Run daemon-reload for logstash for CentOs] *********************
ok: [servercent]

TASK [centoselk : Enable service logstash for CentOs] ****************************
ok: [servercent]

TASK [centoselk : ensure logstash is running for CentOs] *************************
ok: [servercent]

TASK [centoselk : Install Kibana for CentOs] *************************************
ok: [servercent]
```

```
TASK [centoselk : Configure Kibana Add server.port for CentOs] *******
ok: [servercent]

TASK [centoselk : Configure Kibana Add server.host for CentOs] *******
ok: [servercent]

TASK [centoselk : Configure Kibana Add server.name for CentOs] *******
ok: [servercent]

TASK [centoselk : Configure Kibana Add elasticsearch.hosts for CentOs]
ok: [servercent]

TASK [centoselk : Run daemon-reload for kibana for CentOs] ***********
ok: [servercent]

TASK [centoselk : Enable service Kibana for CentOs] ******************
ok: [servercent]

TASK [centoselk : Start Elasticsearch for CentOS] ********************
changed: [servercent]

TASK [centoselk : Start Kibana for CentOS] ***************************
changed: [servercent]

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

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

TASK [install apache and php for Ubuntu servers] *********************
skipping: [servercent]
```

```
TASK [install apache and php for CentOS servers] *********************************************************************
ok: [servercent]

TASK [start httpd CentOS] ********************************************************************************************
ok: [servercent]

PLAY RECAP ***********************************************************************************************************
server1                    : ok=1    changed=0    unreachable=1    failed=0    skipped=1    rescued=0    ignored=0
servercent                 : ok=35   changed=3    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0

ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO$
```

We closed the ubuntu to maximize ram for other vms now both elk are installed we can make this into our config

```
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO$ mv elk.yml config.yaml
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO$ ls
ansible.cfg  config.yaml  inventory  roles
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO$
```

```yaml
---
- hosts: all
  become: true
  pre_tasks:

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

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

- hosts: ubuntu
  become: true
  roles:
    - ubuntuelk

- hosts: centos
  become: true
  roles:
    - centoselk
```

Now we can add another task to our config yaml.

```yaml
- hosts: ubuntu
  become: true
  roles:
    - ubuntuelk
    - nagios

- hosts: centos
  become: true
  roles:
    - centoselk
```

Here we added Nagios lets try to run Nagios by itself.

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

TASK [Gathering Facts] *********************************************************************************************
ok: [server1]
ok: [servercent]

TASK [update repository index CentOS] ******************************************************************************
skipping: [server1]
ok: [servercent]

TASK [install updates Ubuntu] **************************************************************************************
skipping: [servercent]
ok: [server1]

PLAY [ubuntu] ******************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [server1]

TASK [nagios : Install nagios in Ubuntu] ***************************************************************************
ok: [server1]

PLAY [centos] ******************************************************************************************************

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

PLAY RECAP *********************************************************************************************************
server1                    : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
servercent                 : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Nagios will only be installed in Ubuntu and skip centos. Next is IGP and we will run it by itself.

```
- hosts: ubuntu
  become: true
  roles:
    - ubuntuelk
    - nagios
    - igp

- hosts: centos
  become: true
  roles:
    - centoselk
    - igp
```

```
PLAY [ubuntu] ***********************************************

TASK [Gathering Facts] **************************************
ok: [server1]

TASK [igp : Download Influxdb Package] **********************
changed: [server1]

TASK [igp : Update Ubuntu to read Influxdb Package] ********
ok: [server1]

TASK [igp : Install Influxdb Ubuntu] ************************
ok: [server1]

TASK [igp : Download the Influxdb CentOS] *******************
skipping: [server1]

TASK [igp : Install Influxdb CentOS] ************************
skipping: [server1]

TASK [igp : Start Influxdb] *********************************
changed: [server1]

TASK [igp : Grafaana Package Ubuntu] ************************
changed: [server1]

TASK [igp : Update Ubuntu to read Grafana Package] *********
ok: [server1]

TASK [igp : Install Grafana Ubuntu] *************************
ok: [server1]
```

```
TASK [igp : Download the Grafana CentOS] ************
skipping: [server1]

TASK [igp : Install Grafana CentOS] *****************
skipping: [server1]

TASK [igp : Start Grafana] **************************
changed: [server1]

TASK [igp : Install Prometheus Ubuntu] **************
ok: [server1]

TASK [igp : Install Prometheus CentOS] **************
skipping: [server1]

TASK [igp : Enabling snapd] *************************
skipping: [server1]

TASK [igp : Prometheus CentOS] **********************
skipping: [server1]

PLAY [centos] ***************************************

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

TASK [igp : Download Influxdb Package] **************
skipping: [servercent]

TASK [igp : Update Ubuntu to read Influxdb Package] *
skipping: [servercent]
```

```
TASK [igp : Install Influxdb Ubuntu] ************************
skipping: [servercent]

TASK [igp : Download the Influxdb CentOS] *******************
changed: [servercent]

TASK [igp : Install Influxdb CentOS] ************************
changed: [servercent]

TASK [igp : Start Influxdb] *********************************
changed: [servercent]

TASK [igp : Grafaana Package Ubuntu] ************************
skipping: [servercent]

TASK [igp : Update Ubuntu to read Grafana Package] **********
skipping: [servercent]

TASK [igp : Install Grafana Ubuntu] *************************
skipping: [servercent]

TASK [igp : Download the Grafana CentOS] ********************
changed: [servercent]

TASK [igp : Install Grafana CentOS] *************************
changed: [servercent]

TASK [igp : Start Grafana] **********************************
fatal: [servercent]: FAILED! => {"changed": false, "msg": "Unal
se a timeout was exceeded. See \"systemctl status grafana-serv
```

```
TASK [igp : Start Grafana] ********************************************************************************************************
fatal: [servercent]: FAILED! => {"changed": false, "msg": "Unable to start service grafana-server: Job for grafana-server.service failed
se a timeout was exceeded. See \"systemctl status grafana-server.service\" and \"journalctl -xe\" for details.\n"}

PLAY RECAP ************************************************************************************************************************
server1                    : ok=10   changed=4    unreachable=0    failed=0    skipped=7    rescued=0    ignored=0
servercent                 : ok=6    changed=5    unreachable=0    failed=1    skipped=6    rescued=0    ignored=0

ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO$ ssh servercent
Last login: Sun Oct 30 18:05:48 2022 from 192.168.56.102
Ansible Managed node by Bello
[ubuntuhost@localhost ~]$ systemctl status grafana-server.service
● grafana-server.service - Grafana instance
   Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; disabled; vendor preset: disabled)
   Active: active (running) since Sun 2022-10-30 18:07:58 PST; 1min 42s ago
     Docs: http://docs.grafana.org
 Main PID: 21654 (grafana-server)
    Tasks: 16
   CGroup: /system.slice/grafana-server.service
           └─21654 /usr/sbin/grafana-server --config=/etc/grafana/grafana.ini --pidfile=/var/run/grafana/grafana-server.pid --packaging=r

Oct 30 18:07:57 localhost.localdomain grafana-server[21654]: logger=server t=2022-10-30T18:07:57.996853849+08:00 level=info msg="Writ...=
Oct 30 18:07:58 localhost.localdomain grafana-server[21654]: logger=provisioning.alerting t=2022-10-30T18:07:58.103657623+08:00 level...r
Oct 30 18:07:58 localhost.localdomain grafana-server[21654]: logger=provisioning.alerting t=2022-10-30T18:07:58.103715216+08:00 level...r
Oct 30 18:07:58 localhost.localdomain systemd[1]: Started Grafana instance.
Oct 30 18:07:58 localhost.localdomain grafana-server[21654]: logger=ngalert t=2022-10-30T18:07:58.107670134+08:00 level=info msg="war...a
Oct 30 18:07:58 localhost.localdomain grafana-server[21654]: logger=ticker t=2022-10-30T18:07:58.108040294+08:00 level=info msg=start...+
Oct 30 18:07:58 localhost.localdomain grafana-server[21654]: logger=grafanaStorageLogger t=2022-10-30T18:07:58.111119369+08:00 level=...r
Oct 30 18:07:58 localhost.localdomain grafana-server[21654]: logger=report t=2022-10-30T18:07:58.113606099+08:00 level=warn msg="Sche...a
Oct 30 18:07:58 localhost.localdomain grafana-server[21654]: logger=ngalert.multiorg.alertmanager t=2022-10-30T18:07:58.114998646+08:...n
Oct 30 18:07:58 localhost.localdomain grafana-server[21654]: logger=http.server t=2022-10-30T18:07:58.233214566+08:00 level=info msg=...o
Hint: Some lines were ellipsized, use -l to show in full.
[ubuntuhost@localhost ~]$ S
```

We got an error timeout, but it is running fine. We will just rerun the centos since ubuntu is working successfully.

```
PLAY [centos] ***********************************************

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

TASK [igp : Download Influxdb Package] **********************
skipping: [servercent]

TASK [igp : Update Ubuntu to read Influxdb Package] ****
skipping: [servercent]

TASK [igp : Install Influxdb Ubuntu] ************************
skipping: [servercent]

TASK [igp : Download the Influxdb CentOS] *******************
ok: [servercent]

TASK [igp : Install Influxdb CentOS] ************************
ok: [servercent]

TASK [igp : Start Influxdb] *********************************
changed: [servercent]

TASK [igp : Grafaana Package Ubuntu] ************************
skipping: [servercent]

TASK [igp : Update Ubuntu to read Grafana Package] *****
skipping: [servercent]
```

```
TASK [igp : Install Grafana Ubuntu] ********************************************************************************
skipping: [servercent]

TASK [igp : Download the Grafana CentOS] ***************************************************************************
ok: [servercent]

TASK [igp : Install Grafana CentOS] ********************************************************************************
ok: [servercent]

TASK [igp : Start Grafana] *****************************************************************************************
changed: [servercent]

TASK [igp : Install Prometheus Ubuntu] *****************************************************************************
skipping: [servercent]

TASK [igp : Install Prometheus CentOS] *****************************************************************************
changed: [servercent]

TASK [igp : Enabling snapd] ****************************************************************************************
changed: [servercent]

TASK [igp : Prometheus CentOS] *************************************************************************************
changed: [servercent]

PLAY RECAP *********************************************************************************************************
server1                    : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
servercent                 : ok=10   changed=5    unreachable=0    failed=0    skipped=7    rescued=0    ignored=0
```

Now IGP is installed in both servers we move on adding command for lamp stack and running it by itself.

```yaml
- hosts: ubuntu
  become: true
  roles:
    - ubuntuelk
    - nagios
    - igp
    - lampstack

- hosts: centos
  become: true
  roles:
    - centoselk
    - igp
    - lampstack
```

```
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO$ ansible-playbook --ask-become-pass config.yaml
BECOME password:

PLAY [ubuntu] ******************************************************************************

TASK [Gathering Facts] *********************************************************************
ok: [server1]

TASK [lampstack : Install Httpd in Ubuntu] *************************************************
ok: [server1]

TASK [lampstack : Start Apache in Ubuntu] **************************************************
changed: [server1]

TASK [lampstack : Install Httpd in CentOS] *************************************************
skipping: [server1]

TASK [lampstack : Start Httpd in CentOS] ***************************************************
skipping: [server1]

TASK [lampstack : Start Httpd in Ubuntu] ***************************************************
changed: [server1]

TASK [lampstack : Install Mariadb in Ubuntu and CentOS] ************************************
ok: [server1]

TASK [lampstack : Start Mariadb] ***********************************************************
changed: [server1]

TASK [lampstack : Install Php] *************************************************************
changed: [server1]
```

```
PLAY [centos] ******************************************************************************

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

TASK [lampstack : Install Httpd in Ubuntu] *************************************************
skipping: [servercent]

TASK [lampstack : Start Apache in Ubuntu] **************************************************
skipping: [servercent]

TASK [lampstack : Install Httpd in CentOS] *************************************************
ok: [servercent]

TASK [lampstack : Start Httpd in CentOS] ***************************************************
changed: [servercent]

TASK [lampstack : Start Httpd in Ubuntu] ***************************************************
skipping: [servercent]

TASK [lampstack : Install Mariadb in Ubuntu and CentOS] ************************************
ok: [servercent]

TASK [lampstack : Start Mariadb] ***********************************************************
changed: [servercent]

TASK [lampstack : Install Php] *************************************************************
ok: [servercent]

PLAY RECAP *********************************************************************************
server1                    : ok=7    changed=4    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
servercent                 : ok=6    changed=2    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
```

Now all tools are installed we can show the final config.yaml file and we're done.

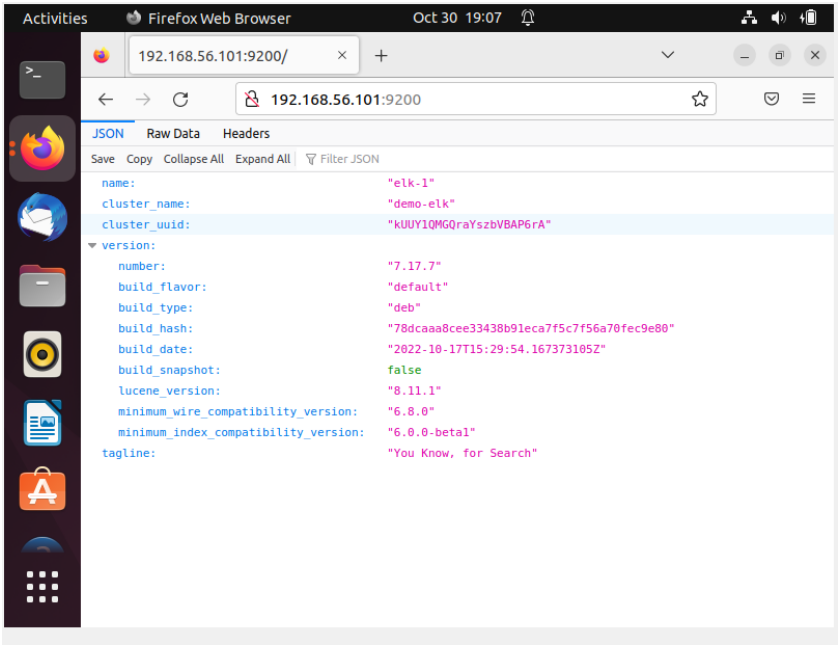
```yaml
---
- hosts: all
  become: true
  pre_tasks:

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

  -  name: install updates Ubuntu
     tags: always
     apt:
       upgrade: dist
       update_cache: yes
     changed_when: false
     when: ansible_distribution == "Ubuntu"
```

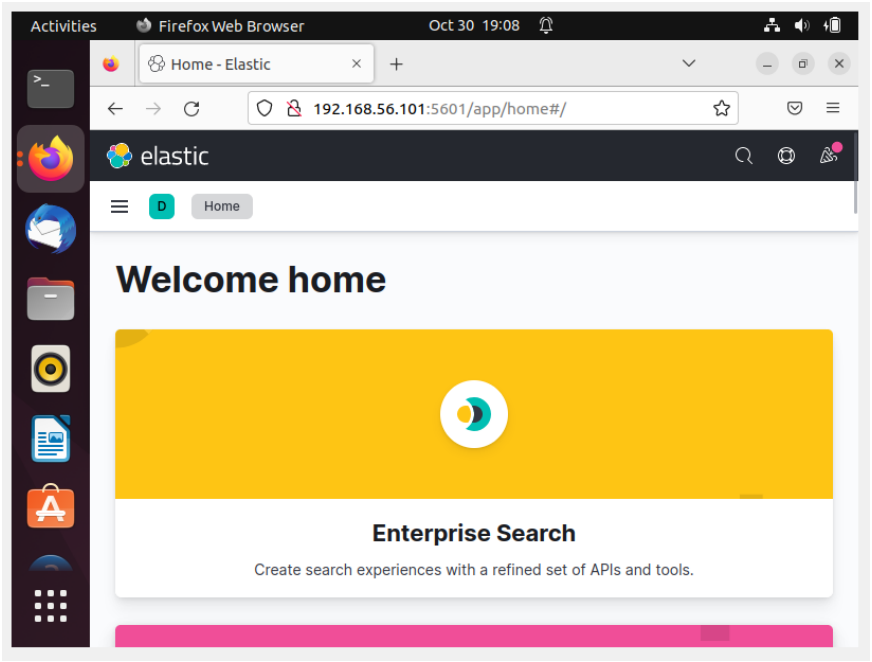
```yaml
- hosts: ubuntu
  become: true
  roles:
     - ubuntuelk
     - nagios
     - igp
     - lampstack

- hosts: centos
  become: true
  roles:
     - centoselk
     - igp
     - lampstack
```

# OUTPUTS

## Ubuntu

ELK



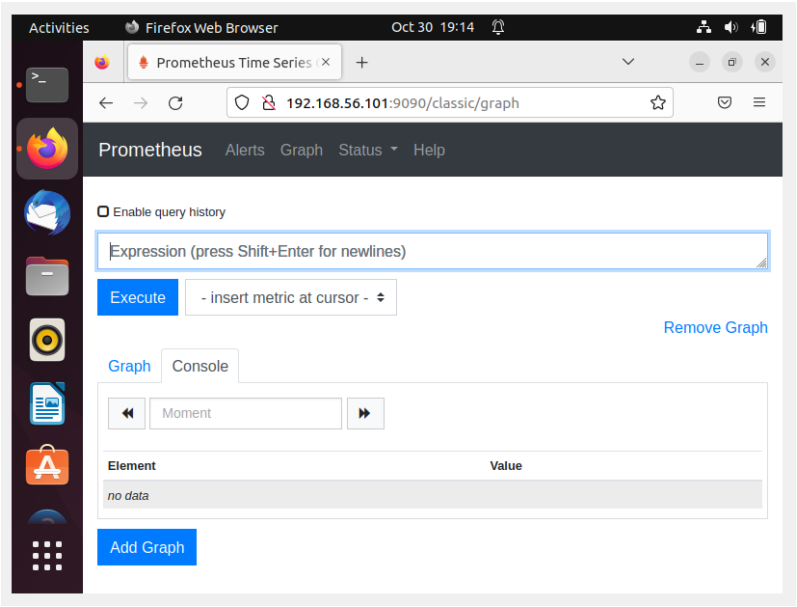
```
ubuntuhost@server1:~$ systemctl status elasticsearch.service
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendo>
    Drop-In: /etc/systemd/system/elasticsearch.service.d
             └─startup-timeout.conf
     Active: active (running) since Sun 2022-10-30 19:06:20 PST; 2min 35s ago
       Docs: https://www.elastic.co
   Main PID: 1151 (java)
      Tasks: 64 (limit: 5567)
     Memory: 2.0G
        CPU: 57.675s
     CGroup: /system.slice/elasticsearch.service
             ├─1151 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.ne>
             └─2806 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux->

Oct 30 19:02:46 server1 systemd[1]: Starting Elasticsearch...
Oct 30 19:06:20 server1 systemd[1]: Started Elasticsearch.
```

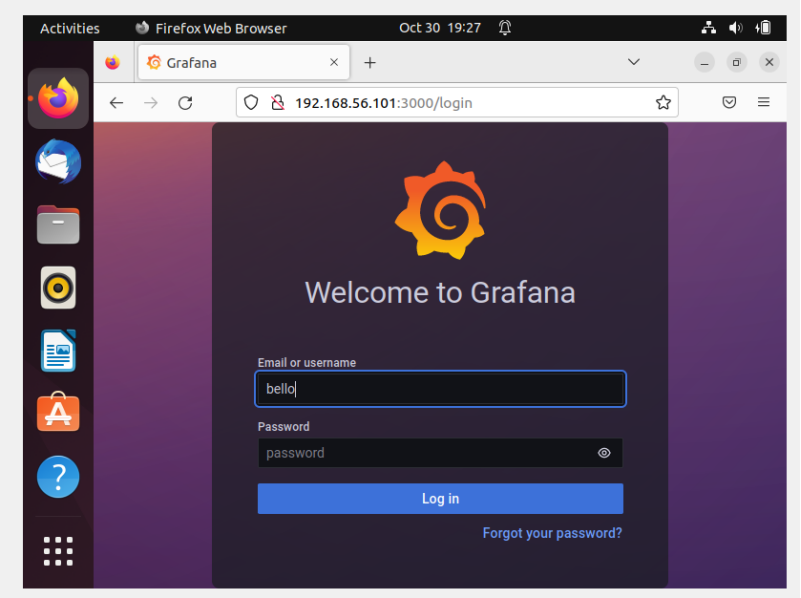
```
ubuntuhost@server1:~$ systemctl status kibana.service
● kibana.service - Kibana
     Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor prese>
     Active: active (running) since Sun 2022-10-30 19:02:46 PST; 6min ago
       Docs: https://www.elastic.co
   Main PID: 1154 (node)
      Tasks: 11 (limit: 5567)
     Memory: 356.5M
        CPU: 25.107s
     CGroup: /system.slice/kibana.service
             └─1154 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/b>

Oct 30 19:02:46 server1 systemd[1]: Started Kibana.
```

```
ubuntuhost@server1:~$ systemctl status logstash.service
● logstash.service - logstash
     Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor pre>
     Active: active (running) since Sun 2022-10-30 19:09:18 PST; 1s ago
   Main PID: 4510 (java)
      Tasks: 14 (limit: 5567)
     Memory: 64.7M
        CPU: 1.271s
     CGroup: /system.slice/logstash.service
             └─4510 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseCon>

Oct 30 19:09:18 server1 systemd[1]: Started logstash.
Oct 30 19:09:18 server1 logstash[4510]: Using bundled JDK: /usr/share/logstash>
Oct 30 19:09:18 server1 logstash[4510]: OpenJDK 64-Bit Server VM warning: Opti>
lines 1-13/13 (END)
```

## Nagios

IGP


Activities
Firefox Web Browser
Oct 30 19:14
Prometheus Time Series
192.168.56.101:9090/classic/graph
Prometheus
Alerts
Graph
Status
Help
Enable query history
Expression (press Shift+Enter for newlines)
Execute
- insert metric at cursor -
Remove Graph
Graph
Console
Moment
Element
Value
no data
Add Graph


Activities
Firefox Web Browser
Oct 30 19:27
Grafana
192.168.56.101:3000/login
Welcome to Grafana
Email or username
bello
Password
password
Log in
Forgot your password?

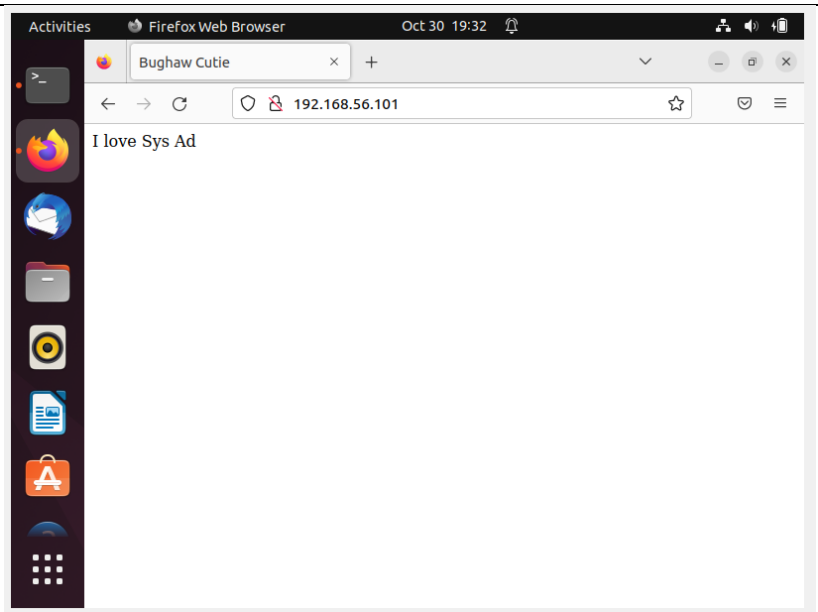
```
ubuntuhost@server1:~$ systemctl status grafana-server.service
● grafana-server.service - Grafana instance
     Loaded: loaded (/lib/systemd/system/grafana-server.service; disabled; ven>
     Active: active (running) since Sun 2022-10-30 19:26:14 PST; 1min 41s ago
       Docs: http://docs.grafana.org
   Main PID: 6261 (grafana-server)
      Tasks: 12 (limit: 5567)
     Memory: 120.5M
        CPU: 670ms
     CGroup: /system.slice/grafana-server.service
             └─6261 /usr/sbin/grafana-server --config=/etc/grafana/grafana.ini>

Oct 30 19:26:18 server1 grafana-server[6261]: logger=infra.usagestats.collecto>
Oct 30 19:26:18 server1 grafana-server[6261]: logger=server t=2022-10-30T19:26>
Oct 30 19:26:18 server1 grafana-server[6261]: logger=provisioning.alerting t=2>
Oct 30 19:26:18 server1 grafana-server[6261]: logger=provisioning.alerting t=2>
Oct 30 19:26:18 server1 grafana-server[6261]: logger=ngalert t=2022-10-30T19:2>
Oct 30 19:26:18 server1 grafana-server[6261]: logger=grafanaStorageLogger t=20>
Oct 30 19:26:18 server1 grafana-server[6261]: logger=http.server t=2022-10-30T>
Oct 30 19:26:18 server1 grafana-server[6261]: logger=ticker t=2022-10-30T19:26>
Oct 30 19:26:18 server1 grafana-server[6261]: logger=ngalert.multiorg.alertman>
Oct 30 19:26:22 server1 grafana-server[6261]: logger=context userId=0 orgId=0 >
lines 1-21/21 (END)
```

```
ubuntuhost@server1:~$ systemctl status prometheus
● prometheus.service - Monitoring system and time series database
     Loaded: loaded (/lib/systemd/system/prometheus.service; enabled; vendor p>
     Active: active (running) since Sun 2022-10-30 19:02:32 PST; 13min ago
       Docs: https://prometheus.io/docs/introduction/overview/
             man:prometheus(1)
   Main PID: 723 (prometheus)
      Tasks: 10 (limit: 5567)
     Memory: 51.5M
        CPU: 2.160s
     CGroup: /system.slice/prometheus.service
             └─723 /usr/bin/prometheus

Oct 30 19:03:33 server1 prometheus[723]: ts=2022-10-30T11:03:33.693Z caller=he>
Oct 30 19:03:33 server1 prometheus[723]: ts=2022-10-30T11:03:33.696Z caller=ch>
Oct 30 19:03:33 server1 prometheus[723]: ts=2022-10-30T11:03:33.822Z caller=he>
Oct 30 19:03:35 server1 prometheus[723]: ts=2022-10-30T11:03:35.338Z caller=co>
Oct 30 19:03:35 server1 prometheus[723]: ts=2022-10-30T11:03:35.369Z caller=db>
Oct 30 19:03:35 server1 prometheus[723]: ts=2022-10-30T11:03:35.490Z caller=db>
Oct 30 19:03:36 server1 prometheus[723]: ts=2022-10-30T11:03:36.739Z caller=co>
Oct 30 19:03:36 server1 prometheus[723]: ts=2022-10-30T11:03:36.781Z caller=db>
Oct 30 19:03:36 server1 prometheus[723]: ts=2022-10-30T11:03:36.789Z caller=db>
Oct 30 19:03:36 server1 prometheus[723]: ts=2022-10-30T11:03:36.847Z caller=db>
lines 1-22/22 (END)
```

```
ubuntuhost@server1:~$ systemctl status influxdb.service
● influxdb.service - InfluxDB is an open-source, distributed, time series data>
     Loaded: loaded (/lib/systemd/system/influxdb.service; enabled; vendor pre>
     Active: active (running) since Sun 2022-10-30 19:02:38 PST; 26min ago
       Docs: man:influxd(1)
   Main PID: 1022 (influxd)
      Tasks: 7 (limit: 5567)
     Memory: 15.1M
        CPU: 1.161s
     CGroup: /system.slice/influxdb.service
             └─1022 /usr/bin/influxd -config /etc/influxdb/influxdb.conf
```

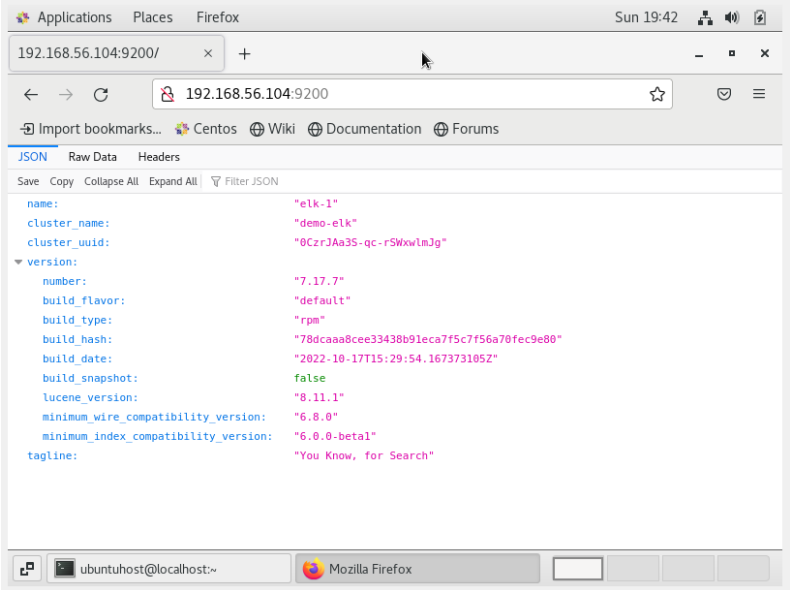
## Lamp Stack

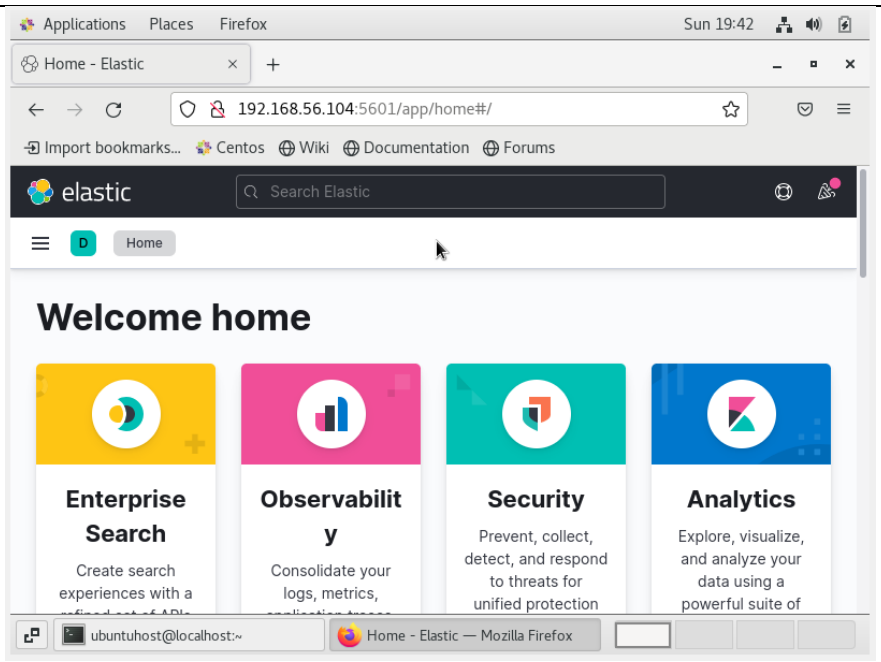
```
ubuntuhost@server1:~$ mariadb --version
mariadb  Ver 15.1 Distrib 10.6.7-MariaDB, for debian-linux-gnu (x86_64) using
EditLine wrapper
```


Activities
Firefox Web Browser
Oct 30 19:32
Bughaw Cutie
192.168.56.101
I love Sys Ad

## CentOS

ELK


Applications
Places
Firefox
Sun 19:42
192.168.56.104:9200/
192.168.56.104:9200
Import bookmarks...
Centos
Wiki
Documentation
Forums
JSON
Raw Data
Headers
Save
Copy
Collapse All
Expand All
Filter JSON
name: "elk-1"
cluster_name: "demo-elk"
cluster_uuid: "0CzrJAa3S-qc-rSWxwlmJg"
version:
number: "7.17.7"
build_flavor: "default"
build_type: "rpm"
build_hash: "78dcaaa8cee33438b91eca7f5c7f56a70fec9e80"
build_date: "2022-10-17T15:29:54.167373105Z"
build_snapshot: false
lucene_version: "8.11.1"
minimum_wire_compatibility_version: "6.8.0"
minimum_index_compatibility_version: "6.0.0-beta1"
tagline: "You Know, for Search"
ubuntuhost@localhost:~
Mozilla Firefox

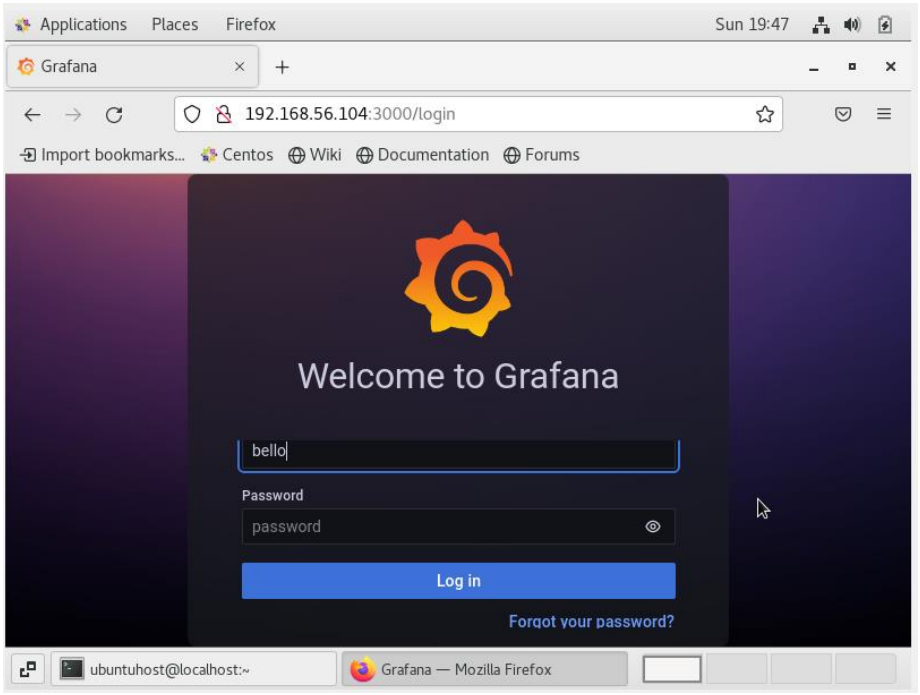


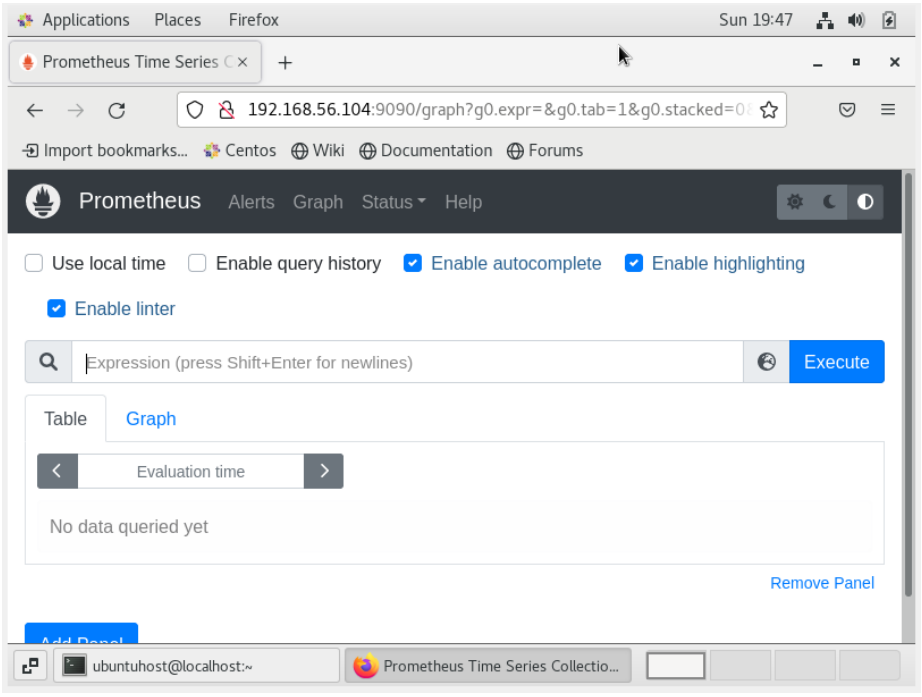

```
[ubuntuhost@localhost ~]$ systemctl status kibana.service
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: disabled
)
   Active: active (running) since Sun 2022-10-30 19:34:49 PST; 5min ago
     Docs: https://www.elastic.co
 Main PID: 1254 (node)
    Tasks: 11
   CGroup: /system.slice/kibana.service
           └─1254 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/../sr...

Oct 30 19:34:49 localhost.localdomain systemd[1]: Started Kibana.
[ubuntuhost@localhost ~]$
```

```
[ubuntuhost@localhost ~]$ systemctl status elasticsearch.service
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; vendor prese
t: disabled)
  Drop-In: /etc/systemd/system/elasticsearch.service.d
           └─startup-timeout.conf
   Active: active (running) since Sun 2022-10-30 19:36:02 PST; 3min 45s ago
     Docs: https://www.elastic.co
 Main PID: 1232 (java)
    Tasks: 62
   CGroup: /system.slice/elasticsearch.service
           ├─1232 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.networkadd...
           └─2477 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/b...

Oct 30 19:34:49 localhost.localdomain systemd[1]: Starting Elasticsearch...
Oct 30 19:36:02 localhost.localdomain systemd[1]: Started Elasticsearch.
[ubuntuhost@localhost ~]$
```

```
[ubuntuhost@localhost ~]$ systemctl status logstash.service
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: disabl
ed)
   Active: active (running) since Sun 2022-10-30 19:39:51 PST; 1min 29s ago
 Main PID: 4119 (java)
    Tasks: 14
   CGroup: /system.slice/logstash.service
           └─4119 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSwe...

Oct 30 19:39:51 localhost.localdomain systemd[1]: Started logstash.
Oct 30 19:39:53 localhost.localdomain logstash[4119]: Using bundled JDK: /usr/share...k
Oct 30 19:39:57 localhost.localdomain logstash[4119]: OpenJDK 64-Bit Server VM warn....
Hint: Some lines were ellipsized, use -l to show in full.
[ubuntuhost@localhost ~]$
```

IGP

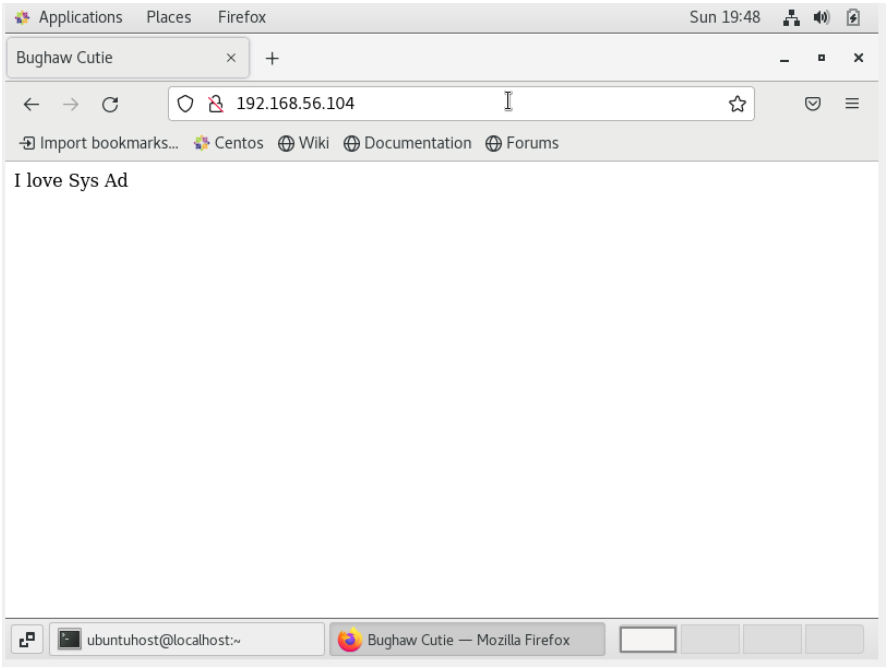

Applications
Places
Firefox
Sun 19:47
Grafana
192.168.56.104:3000/login
Import bookmarks...
Centos
Wiki
Documentation
Forums
Welcome to Grafana
bello
Password
password
Log in
Forgot your password?
ubuntuhost@localhost:~
Grafana — Mozilla Firefox


Applications
Places
Firefox
Sun 19:47
Prometheus Time Series
192.168.56.104:9090/graph?g0.expr=&g0.tab=1&g0.stacked=0
Import bookmarks...
Centos
Wiki
Documentation
Forums
Prometheus
Alerts
Graph
Status
Help
Use local time
Enable query history
Enable autocomplete
Enable highlighting
Enable linter
Expression (press Shift+Enter for newlines)
Execute
Table
Graph
Evaluation time
No data queried yet
Remove Panel
ubuntuhost@localhost:~
Prometheus Time Series Collectio...

```
[ubuntuhost@localhost ~]$ systemctl status influxdb.service
● influxdb.service - InfluxDB is an open-source, distributed, time series database
   Loaded: loaded (/usr/lib/systemd/system/influxdb.service; enabled; vendor preset: di
sabled)
   Active: active (running) since Sun 2022-10-30 19:34:49 PST; 10min ago
     Docs: https://docs.influxdata.com/influxdb/
 Main PID: 1233 (influxd)
    Tasks: 8
   CGroup: /system.slice/influxdb.service
           └─1233 /usr/bin/influxd -config /etc/influxdb/influxdb.conf
```

```
[ubuntuhost@localhost ~]$ systemctl status grafana-server.service
● grafana-server.service - Grafana instance
   Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; disabled; vendor pre
set: disabled)
   Active: active (running) since Sun 2022-10-30 19:43:53 PST; 7s ago
     Docs: http://docs.grafana.org
 Main PID: 4709 (grafana-server)
    Tasks: 11
   CGroup: /system.slice/grafana-server.service
           └─4709 /usr/sbin/grafana-server --config=/etc/grafana/grafana.ini --pidfi...
```

## Lamp Stack



## Git add commit and push

```
ubuntuhost@workstation:~$ cd CPE_MIDEXAM_BELLO/
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO$ git add -A
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO$ git commit -m "midterm exam"
[main (root-commit) d92204b] midterm exam
 8 files changed, 482 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 config.yaml
 create mode 100644 inventory
 create mode 100644 roles/centoselk/tasks/main.yml
 create mode 100644 roles/igp/tasks/main.yml
 create mode 100644 roles/lampstack/tasks/main.yml
 create mode 100644 roles/nagios/tasks/main.yml
 create mode 100644 roles/ubuntuelk/tasks/main.yml
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO$ git push
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Compressing objects: 100% (10/10), done.
Writing objects: 100% (21/21), 3.45 KiB | 1.72 MiB/s, done.
Total 21 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To github.com:qictbello/CPE_MIDEXAM_BELLO.git
 * [new branch]      main -> main
ubuntuhost@workstation:~/CPE_MIDEXAM_BELLO$
```

## Contents of the repository

### Config.yaml

```
---
- hosts: all
  become: true
  pre_tasks:

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

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

- hosts: ubuntu
  become: true
  roles:
    - ubuntuelk
    - nagios
    - igp
    - lampstack

- hosts: centos
  become: true
  roles:
    - centoselk
    - igp
    - lampstack
```

### Ansible.cfg

4 lines (4 sloc) | 79 Bytes

```
1  [defaults]
2  timeout=500
3  inventory = inventory
4  private_key_file = ~/.ssh/ansible
```

## Inventory

5 lines (4 sloc) | 38 Bytes

```
1  [ubuntu]
2  server1
3
4  [centos]
5  servercent
```

# ROLES

## Centoselk

```
- name: Install ELK Prereq CentOs
  yum:
    name:
      - java-11-openjdk
      - curl
      - gnupg
    state: latest

- name: install elasticsearch rpm key CentOs
  rpm_key:
    key: https://artifacts.elastic.co/GPG-KEY-elasticsearch
    state: present
  become: true

- name: install elasticsearch 7.x rpm repository
  yum_repository:
    name: Elastic_7.X_repo
    baseurl: https://artifacts.elastic.co/packages/7.x/yum
    gpgcheck: true
    gpgkey: https://artifacts.elastic.co/GPG-KEY-elasticsearch
    description: Elastic 7.X Repo
  become: true

- name: Install Elasticsearch CentOs
  yum:
    name: elasticsearch
    state: latest
    update_cache: yes

- name: Configure Elasticsearch change cluster name CentOs
  lineinfile:
    dest: /etc/elasticsearch/elasticsearch.yml
    line: "cluster.name: demo-elk"
    state: present

- name: Configure Elasticsearch give cluster descriptive name CentOs
  lineinfile:
    dest: /etc/elasticsearch/elasticsearch.yml
    line: "node.name: elk-1"
    state: present

- name: Configure Elasticsearch Add network.host CentOs
  lineinfile:
    dest: /etc/elasticsearch/elasticsearch.yml
    line: "network.host: 0.0.0.0"
    state: present

- name: Configure Elasticsearch Add http.port CentOs
  lineinfile:
    dest: /etc/elasticsearch/elasticsearch.yml
    line: "http.port: 9200"
    state: present

- name: Configure Elasticsearch Add discovery.type CentOs
  lineinfile:
    dest: /etc/elasticsearch/elasticsearch.yml
    line: "discovery.type: single-node"
    state: present

- name: Creating an empty file for startup-timeout.conf 1 of 2 CentOs
  file:
    path: "/etc/systemd/system/elasticsearch.service.d"
    state: directory

- name: Creating an empty file for startup-timeout.conf 2 of 2 CentOs
  file:
    path: "/etc/systemd/system/elasticsearch.service.d/startup-timeout.conf"
    state: touch

- name: Prevent systemd service start operation from timing out CentOs
  copy:
    dest: "/etc/systemd/system/elasticsearch.service.d/startup-timeout.conf"
    content: |
      [Service]
      TimeoutStartSec=900

- name: Run daemon-reload for elasticsearch CentOs
  systemd: daemon_reload=yes

- name: Enable service Elasticsearch and ensure it is not masked CentOs
  systemd:
    name: elasticsearch
    enabled: yes
    masked: no
```

```
    - name: ensure elasticsearch is running for CentOs
      systemd: state=started name=elasticsearch

    - name: Install Logstash CentOs
      yum:
        name: logstash
        state: latest
        update_cache: yes

    - name: Run daemon-reload for logstash for CentOs
      systemd: daemon_reload=yes

    - name: Enable service logstash for CentOs
      systemd:
        name: logstash
        enabled: yes

    - name: ensure logstash is running for CentOs
      systemd: state=started name=logstash


    - name: Install Kibana for CentOs
      yum:
        name: kibana
        state: latest
        update_cache: yes

    - name: Configure Kibana Add server.port for CentOs
      lineinfile:
        dest: /etc/kibana/kibana.yml
        line: "server.port: 5601"
        state: present

    - name: Configure Kibana Add server.host for CentOs
      lineinfile:
        dest: /etc/kibana/kibana.yml
        line: 'server.host: "0.0.0.0"'
        state: present
```

```
- name: Configure Kibana Add server.name for CentOs
  lineinfile:
    dest: /etc/kibana/kibana.yml
    line: 'server.name: "demo-kibana"'
    state: present

- name: Configure Kibana Add elasticsearch.hosts for CentOs
  lineinfile:
    dest: /etc/kibana/kibana.yml
    line: 'elasticsearch.hosts: ["http://0.0.0.0:9200"]'
    state: present

- name: Run daemon-reload for kibana for CentOs
  systemd: daemon_reload=yes

- name: Enable service Kibana for CentOs
  systemd:
    name: kibana
    enabled: yes

- name: Start Elasticsearch for CentOS
  shell: systemctl start elasticsearch

- name: Start Kibana for CentOS
  shell: systemctl start kibana
```

## Ubuntuelk

```
- name: Install ELK Prereq Ubuntu
  apt:
    name:
      - openjdk-11-jdk
      - apt-transport-https
      - curl
      - gpgv
      - gpgsm
      - gnupg-l10n
      - gnupg
      - dirmngr
    state: latest

- name: Get PGP Key Ubuntu
  apt_key:
    url: https://artifacts.elastic.co/GPG-KEY-elasticsearch
    state: present

- name: Install Elasticsearch repository into sources list Ubuntu
  apt_repository:
    repo: deb https://artifacts.elastic.co/packages/7.x/apt stable main
    state: present

- name: Install Elasticsearch Ubuntu
  apt:
    name: elasticsearch
    state: latest
    update_cache: yes

- name: Configure Elasticsearch change cluster name Ubuntu
  lineinfile:
    dest: /etc/elasticsearch/elasticsearch.yml
    line: "cluster.name: demo-elk"
    state: present

- name: Configure Elasticsearch give cluster descriptive name Ubuntu
  lineinfile:
    dest: /etc/elasticsearch/elasticsearch.yml
    line: "node.name: elk-1"
    state: present
```

```
- name: Configure Elasticsearch Add network.host Ubuntu
  lineinfile:
    dest: /etc/elasticsearch/elasticsearch.yml
    line: "network.host: 0.0.0.0"
    state: present

- name: Configure Elasticsearch Add http.port Ubuntu
  lineinfile:
    dest: /etc/elasticsearch/elasticsearch.yml
    line: "http.port: 9200"
    state: present

- name: Configure Elasticsearch Add discovery.type Ubuntu
  lineinfile:
    dest: /etc/elasticsearch/elasticsearch.yml
    line: "discovery.type: single-node"
    state: present

- name: Creating an empty file for startup-timeout.conf 1 of 2 Ubuntu
  file:
    path: "/etc/systemd/system/elasticsearch.service.d"
    state: directory

- name: Creating an empty file for startup-timeout.conf 2 of 2 Ubuntu
  file:
    path: "/etc/systemd/system/elasticsearch.service.d/startup-timeout.conf"
    state: touch

- name: Prevent systemd service start operation from timing out Ubuntu
  copy:
    dest: "/etc/systemd/system/elasticsearch.service.d/startup-timeout.conf"
    content: |
      [Service]
      TimeoutStartSec=300

- name: Run daemon-reload for elasticsearch Ubuntu
  systemd: daemon_reload=yes
```

```
- name: Enable service Elasticsearch and ensure it is not masked Ubuntu
  systemd:
    name: elasticsearch
    enabled: yes
    masked: no

- name: ensure elasticsearch is running Ubuntu
  systemd: state=started name=elasticsearch

- name: Install Logstash (Ubuntu)
  apt:
    name: logstash
    state: latest
    update_cache: yes

- name: Run daemon-reload for logstash Ubuntu
  systemd: daemon_reload=yes

- name: Enable service logstash Ubuntu
  systemd:
    name: logstash
    enabled: yes

- name: ensure logstash is running Ubuntu
  systemd: state=started name=logstash

- name: Install Kibana Ubuntu
  apt:
    name: kibana
    state: latest
    update_cache: yes

- name: Configure Kibana Add server.port Ubuntu
  lineinfile:
    dest: /etc/kibana/kibana.yml
    line: "server.port: 5601"
    state: present
```

```
- name: Configure Kibana Add server.host Ubuntu
  lineinfile:
    dest: /etc/kibana/kibana.yml
    line: 'server.host: "0.0.0.0"'
    state: present

- name: Configure Kibana Add server.name Ubuntu
  lineinfile:
    dest: /etc/kibana/kibana.yml
    line: 'server.name: "demo-kibana"'
    state: present

- name: Configure Kibana Add elasticsearch.hosts Ubuntu
  lineinfile:
    dest: /etc/kibana/kibana.yml
    line: 'elasticsearch.hosts: ["http://0.0.0.0:9200"]'
    state: present

- name: Run daemon-reload for kibana Ubuntu
  systemd: daemon_reload=yes

- name: Enable service Kibana Ubuntu
  systemd:
    name: kibana
    enabled: yes

- name: Start Elasticsearch service
  shell: systemctl start elasticsearch

- name: Start Kibana
  shell: systemctl start kibana
```

Nagios

7 lines (7 sloc) | 147 Bytes

```
1  - name: Install nagios in Ubuntu
2    apt:
3      name:
4        - nagios4
5      state: latest
6      update_cache: yes
7    when: ansible_distribution == "Ubuntu"
```

# IGP

```
- name: Download Influxdb Package
  shell: curl -sL https://repos.influxdata.com/influxdb.key | sudo apt-key add -
  when: ansible_distribution == "Ubuntu"

- name: Update Ubuntu to read Influxdb Package
  apt:
    upgrade: dist
    update_cache: yes
  changed_when: false
  when: ansible_distribution == "Ubuntu"

- name: Install Influxdb Ubuntu
  apt:
    name: influxdb
    state: present
  when: ansible_distribution == "Ubuntu"

- name: Download the Influxdb CentOS
  get_url:
    url: https://repos.influxdata.com/rhel/8/x86_64/stable/influxdb-1.8.5.x86_64.rpm
    dest: /tmp/influxdb-1.8.5.x86_64.rpm
  when: ansible_distribution == "CentOS"

- name: Install Influxdb CentOS
  yum:
    name: /tmp/influxdb-1.8.5.x86_64.rpm
    state: present
  when: ansible_distribution == "CentOS"

- name: Start Influxdb
  service:
    name: influxdb
    state: restarted

- name: Grafaana Package Ubuntu
  shell: wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add - | sudo add-apt-repository "deb https://packages.grafana.com/oss/deb stable main"
  when: ansible_distribution == "Ubuntu"
```

```
- name: Update Ubuntu to read Grafana Package
  apt:
    upgrade: dist
    update_cache: yes
  changed_when: false
  when: ansible_distribution == "Ubuntu"

- name: Install Grafana Ubuntu
  apt:
    name: grafana
    state: present
  when: ansible_distribution == "Ubuntu"

- name: Download the Grafana CentOS
  get_url:
    url: https://dl.grafana.com/enterprise/release/grafana-enterprise-9.2.2-1.x86_64.rpm
    dest: /tmp/grafana-enterprise-9.2.2-1.x86_64.rpm
  when: ansible_distribution == "CentOS"

- name: Install Grafana CentOS
  yum:
    name: /tmp/grafana-enterprise-9.2.2-1.x86_64.rpm
    state: present
  when: ansible_distribution == "CentOS"

- name: Start Grafana
  service:
    name: grafana-server
    state: restarted

- name: Install Prometheus Ubuntu
  apt:
   name:
    - prometheus
   state: latest
  when: ansible_distribution == "Ubuntu"
```

```
- name: Install Prometheus CentOS
  dnf:
   name:
    - epel-release
    - snapd
   state: latest
  when: ansible_distribution == "CentOS"

- name: Enabling snapd
  command: systemctl enable --now snapd.socket
  when: ansible_distribution == "CentOS"

- name: Prometheus CentOS
  command: snap install prometheus --classic
  when: ansible_distribution == "CentOS"
```

Lamp Stack

```
- name: Install Httpd in Ubuntu
  apt:
    name: apache2
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: Start Apache in Ubuntu
  service:
    name: apache2
    state: restarted
  when: ansible_distribution == "Ubuntu"

- name: Install Httpd in CentOS
  dnf:
    name: httpd
    state: latest
  when: ansible_distribution == "CentOS"

- name: Start Httpd in CentOS
  service:
    name: httpd
    state: restarted
  when: ansible_distribution == "CentOS"

- name: Start Httpd in Ubuntu
  service:
    name: apache2
    state: restarted
  when: ansible_distribution == "Ubuntu"
```

```
- name: Install Mariadb in Ubuntu and CentOS
  package:
     name: mariadb-server
     state: latest

- name: Start Mariadb
  service:
    name: mariadb
    state: restarted
    enabled: true

- name: Install Php
  package:
    name: php
    state: latest
```

**GitHub link:**

https://github.com/qictbello/CPE_MIDEXAM_BELLO

**Conclusions:** (link your conclusion from the objective)
In conclusion, we created and designed a workflow in the Ansible playbook using roles that configure and manage servers. We installed monitoring tools and applications that can be run or modified on the servers. Most of the errors we encounter are timeout errors. This is because of having a limit on our workstation, which is just 8 GB of RAM. However, we found a solution by running the simulation on each task while every server is running. In the end, we installed all the required tools and followed the tasks on both hosts. I'm looking forward to using this on enterprise and more faster devices so I can see how maximized ansible works.

**Proof of evaluation:**