Name: Sonny Jay B. Bencito	**Date Performed: 10/27/2022**
Course/Section: CPE31S24	**Date Submitted: 10/28/2022**
Instructor: Dr. Jonathan Taylar	**Semester and SY: 1st sem 2022-2023**

Activity 10: Install, Configure, and Manage Log Monitoring tools

1. Objectives

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

2. Discussion

Log monitoring software scans and monitors log files generated by servers, applications, and networks. By detecting and alerting users to patterns in these log files, log monitoring software helps solve performance and security issues. System administrators use log monitoring software to detect common important events indicated by log files.

Log monitoring software helps maintain IT infrastructure performance and pinpoints issues to prevent downtime and mitigate risks. These tools will often integrate with IT alerting software, log analysis software, and other IT issue resolution products to more aptly flesh out the IT infrastructure maintenance ecosystem.

To qualify for inclusion in the Log Monitoring category, a product must:

- Monitor the log files generated by servers, applications, or networks
- Alert users when important events are detected
- Provide reporting capabilities for log files

Elastic Stack

ELK suite stands for Elasticsearch, Kibana, Beats, and Logstash (also known as the ELK Stack). Source: https://www.elastic.co/elastic-stack

The Elastic Stack is a group of open source products from Elastic designed to help users take data from any type of source and in any format, and search, analyze and visualize that data in real time. The product group was formerly known as the ELK Stack for the core products in the group -- Elasticsearch, Logstash and Kibana -- but has been rebranded as the Elastic Stack. A fourth product, Beats, was subsequently added to the stack. The Elastic Stack can be deployed on premises or made available as software as a service (SaaS). Elasticsearch supports Amazon Web Services (AWS), Google Cloud Platform and Microsoft Azure.

GrayLog

Graylog is a powerful platform that allows for easy log management of both structured and unstructured data along with debugging applications.

It is based on Elasticsearch, MongoDB, and Scala. Graylog has a main server, which receives data from its clients installed on different servers, and a web interface, which visualizes the data and allows to work with logs aggregated by the main server.

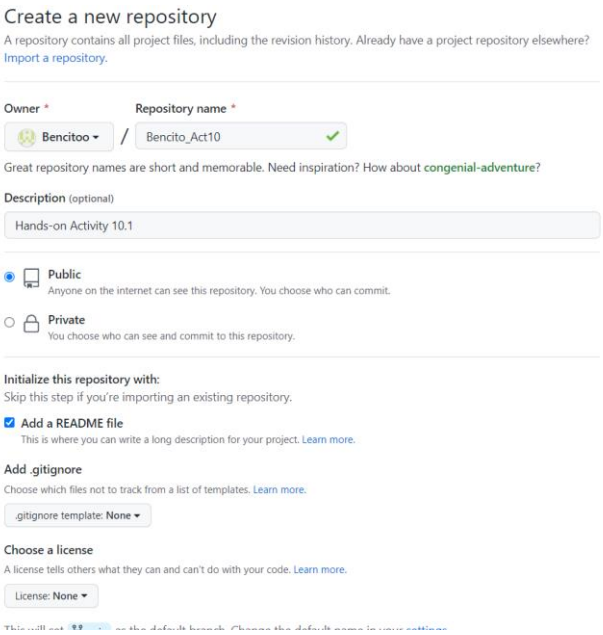
We use Graylog primarily as the stash for the logs of the web applications we build. However, it is also effective when working with raw strings (i.e. syslog): the tool parses it into the structured data we need. It also allows advanced custom search in the logs using structured queries. In other words, when integrated properly with a web app, Graylog helps engineers to analyze the system behavior on almost per code line basis.

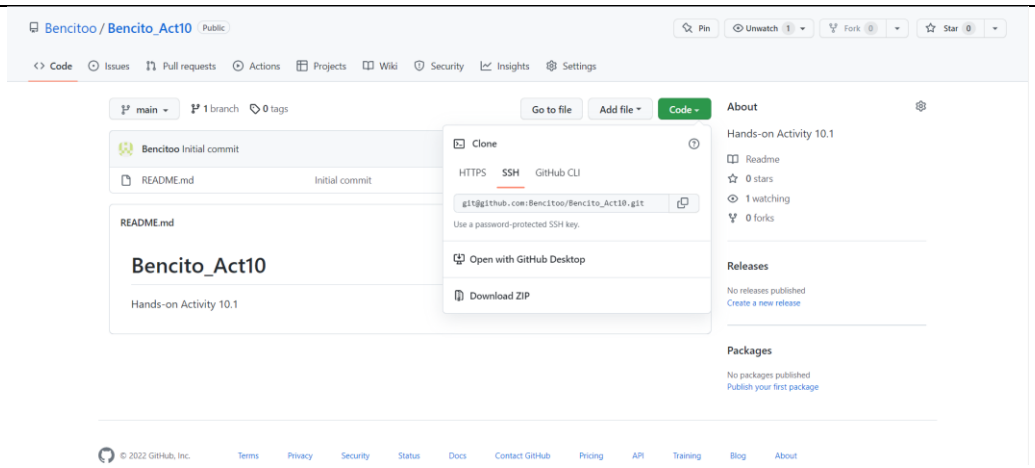
Source: https://www.graylog.org/products/open-source

3. Tasks

1. Create a playbook that:
    a. Install and configure Elastic Stack in separate hosts (Elastic Search, Kibana, Logstash)
2. Apply the concept of creating roles.
3. Describe how you did step 1. (Provide screenshots and explanations in your report. Make your report detailed such that it will look like a manual.)
4. Show an output of the installed Elastic Stack for both Ubuntu and CentOS.
5. Make sure to create a new repository in GitHub for this activity.

4. Output (screenshots and explanations)






```
bencito@workstation:~$ git clone git@github.com:Bencitoo/Bencito_Act10.git
Cloning into 'Bencito_Act10'...
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.
bencito@workstation:~$ cd Bencito_Act10
bencito@workstation:~/Bencito_Act10$
```

I create a repository name "Bencito Act10" and I git clone it on my manage node.

```
bencito@workstation:~/Bencito_Act10$ nano inventory
bencito@workstation:~/Bencito_Act10$ nano ansible.cfg
bencito@workstation:~/Bencito_Act10$
```

```
  GNU nano 6.2                          inventory
[CentOS]

192.168.56.105

[Ubuntu]

192.168.56.102
```

```
  GNU nano 6.2                          ansible.cfg
[defaults]

inventory = inventory
host_key_checking = False

deprecation_warnings = False

remote_user = bencito
private_key_file = ~/.ssh/
```

I create a new inventory and ansible.cfg in the new repository.

```
bencito@workstation:~/Bencito_Act10$ ansible -m ping all
192.168.56.105 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
192.168.56.102 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
bencito@workstation:~/Bencito_Act10$
```

After creating it I show ping the two control nodes and it is successful.

```
bencito@workstation:~/Bencito_Act10$ nano elsk.yml
bencito@workstation:~/Bencito_Act10$
```

```
  GNU nano 6.2                          elsk.yml *
---

- hosts: all
  become: true
  pre_tasks:

  -  name: install updates (Ubuntu)
     tags: always
     apt:
       upgrade: dist
       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"

  -  name: install unzip
     package:
       name: unzip
```

```
 GNU nano 6.2                                elsk.yml *
- hosts: ubuntu
  become: true
  roles:
    - ubuntu

- hosts: centos
  become: true
  roles:
    - centos


- hosts: all
  become: true
  tasks:

  -  name: install apache and php for Ubuntu servers
     tags: apache, apache2, ubuntu
     apt:
       name:
         - apache2
         - libapache2-mod-php
       state: latest
     when: ansible_distribution == "Ubuntu"
```

```
  -  name: install apache and php for CentOS servers
     tags: apache,apache2,centos
     dnf:
       name:
         - httpd
         - php
       state: latest
     when: ansible_distribution == "CentOS"

  -  name: start httpd (CentOS)
     tags: apache,centos,httpd
     service:
       name: httpd
       state: started
     when: ansible_distribution == "CentOS"
```

I create a main playbook name "elsk.yml" I will use that later after creating the installation playbook. Also, I add the install apache because I use a new computer.

```
bencito@workstation:~/Bencito_Act10$ mkdir roles
bencito@workstation:~/Bencito_Act10$ cd roles
bencito@workstation:~/Bencito_Act10/roles$ mkdir ubuntu
bencito@workstation:~/Bencito_Act10/roles$ cd ubuntu
bencito@workstation:~/Bencito_Act10/roles/ubuntu$ mkdir tasks
bencito@workstation:~/Bencito_Act10/roles/ubuntu$ cd tasks
bencito@workstation:~/Bencito_Act10/roles/ubuntu/tasks$ nano main.yml
bencito@workstation:~/Bencito_Act10/roles/ubuntu/tasks$
```

```
GNU nano 6.2                          main.yml
- name: Install Elastic Dependencies (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 sources list (Ubuntu)
  apt_repository:
    repo: deb https://artifacts.elastic.co/packages/7.x/apt stable main
    state: present
```

```
GNU nano 6.2                          main.yml
- name: Install Elasticsearch (Ubuntu)
  apt:
    name: elasticsearch
    state: latest
    update_cache: yes

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

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

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

```
GNU nano 6.2                          main.yml
- name: Configure Elasticsearch Adding http.port (Ubuntu)
  lineinfile:
    dest: /etc/elasticsearch/elasticsearch.yml
    line: "http.port: 9200"
    state: present

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

- name: Creating 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"
```

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

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

I created a roles directory name "ubuntu" after that on the inside of it was the tasks main.yml playbook to install the elasticsearch.

```
bencito@workstation:~/Bencito_Act10/roles$ cd ..
bencito@workstation:~/Bencito_Act10$ cd roles
bencito@workstation:~/Bencito_Act10/roles$ mkdir CentOS
bencito@workstation:~/Bencito_Act10/roles$ cd CentOS
bencito@workstation:~/Bencito_Act10/roles/CentOS$ mkdir tasks
bencito@workstation:~/Bencito_Act10/roles/CentOS$ cd tasks
bencito@workstation:~/Bencito_Act10/roles/CentOS/tasks$ nano main.yml
bencito@workstation:~/Bencito_Act10/roles/CentOS/tasks$
```

```
  GNU nano 6.2                          main.yml
- name: Install ELK Dependencies 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 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
```

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

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

```
GNU nano 6.2                          main.yml
    content: |
      [Service]
      TimeoutStartSec=3min
- 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
```

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

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

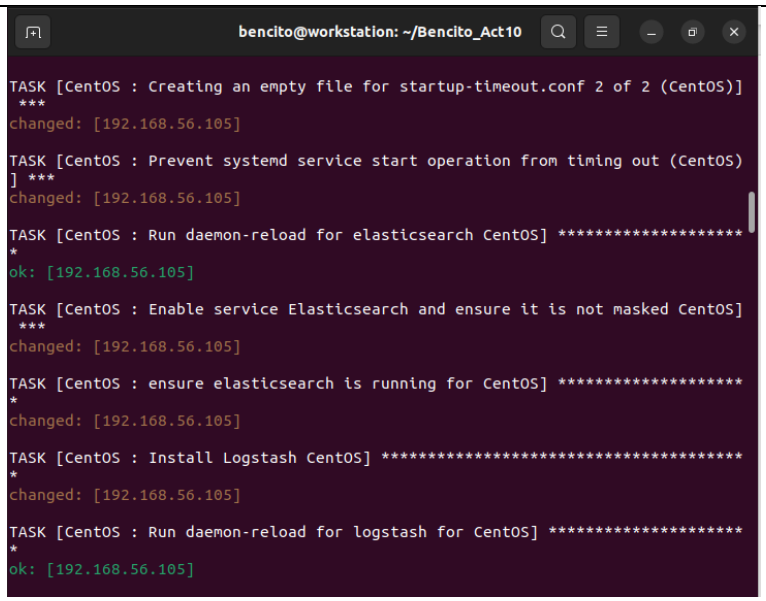
After on the role’s directory on ubuntu. I created another directory for CentOS that inside of it was the tasks main.yml and the code installation of elasticsearch. I separate the Ubuntu and CentOS to make it clean installation.

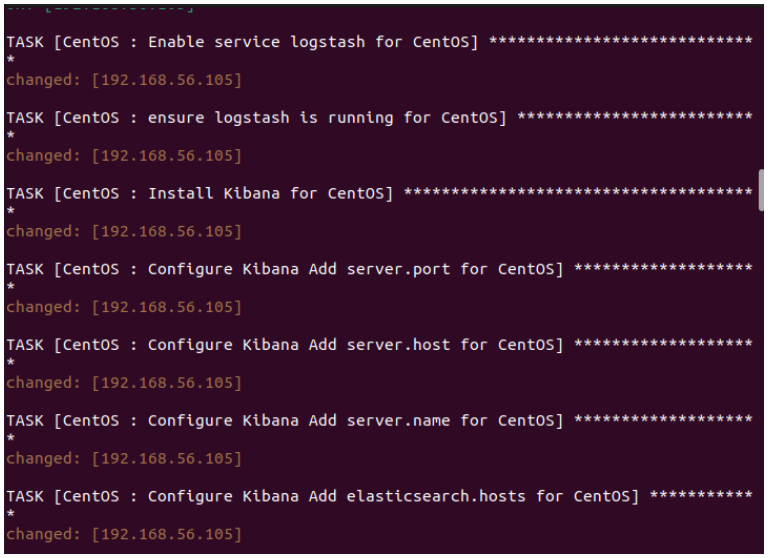
Running Playbook

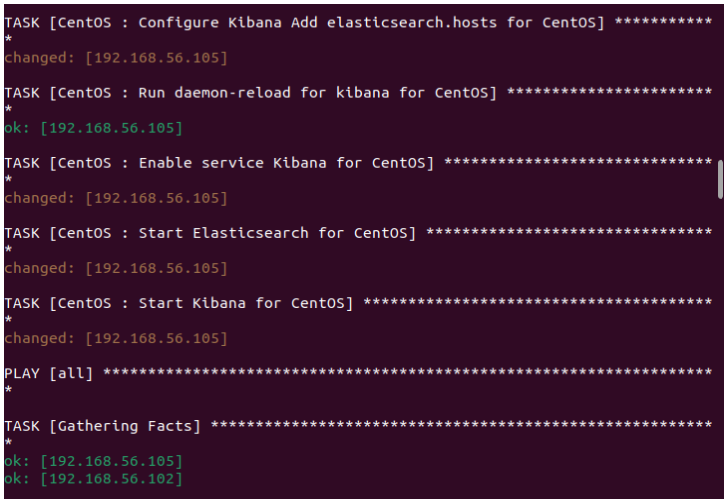
```
bencito@workstation:~/Bencito_Act10$ ansible-playbook --ask-become-pass elsk.ym
l
BECOME password:

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

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

TASK [install updates (Ubuntu)] ************************************************
*
skipping: [192.168.56.105]
ok: [192.168.56.102]

TASK [update repository index (CentOS)] ****************************************
*
skipping: [192.168.56.102]
ok: [192.168.56.105]

TASK [install unzip] ***********************************************************
*
ok: [192.168.56.105]
ok: [192.168.56.102]
```

```
PLAY [ubuntu] ******************************************************************
*
skipping: no hosts matched

PLAY [CentOS] ******************************************************************
*

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.105]

TASK [CentOS : Install ELK Dependencies CentOS] ********************************
*
changed: [192.168.56.105]

TASK [CentOS : install elasticsearch rpm key CentOS] ***************************
*
changed: [192.168.56.105]

TASK [CentOS : install elasticsearch 7.x repository] ***************************
*
changed: [192.168.56.105]

TASK [CentOS : Install Elasticsearch (CentOS)] *********************************
*
changed: [192.168.56.105]
```

```
bencito@workstation: ~/Bencito_Act10

TASK [CentOS : Configure Elasticsearch change cluster name (CentOS)] **********
*
changed: [192.168.56.105]

TASK [CentOS : Configure Elasticsearch give cluster descriptive name (CentOS)]
***
changed: [192.168.56.105]

TASK [CentOS : Configure Elasticsearch Add network.host (CentOS)] *************
*
changed: [192.168.56.105]

TASK [CentOS : Configure Elasticsearch Add http.port (CentOS)] ****************
*
changed: [192.168.56.105]

TASK [CentOS : Configure Elasticsearch Add discovery.type (CentOS)] ***********
*
changed: [192.168.56.105]

TASK [CentOS : Creating an empty file for startup-timeout.conf 1 of 2 (CentOS)]
 ***
changed: [192.168.56.105]

TASK [CentOS : Creating an empty file for startup-timeout.conf 2 of 2 (CentOS)]
 ***
changed: [192.168.56.105]
```


bencito@workstation: ~/Bencito_Act10
TASK [CentOS : Creating an empty file for startup-timeout.conf 2 of 2 (CentOS)] ***
changed: [192.168.56.105]
TASK [CentOS : Prevent systemd service start operation from timing out (CentOS)] ***
changed: [192.168.56.105]
TASK [CentOS : Run daemon-reload for elasticsearch CentOS] **********
ok: [192.168.56.105]
TASK [CentOS : Enable service Elasticsearch and ensure it is not masked CentOS] ***
changed: [192.168.56.105]
TASK [CentOS : ensure elasticsearch is running for CentOS] **********
changed: [192.168.56.105]
TASK [CentOS : Install Logstash CentOS] **********
changed: [192.168.56.105]
TASK [CentOS : Run daemon-reload for logstash for CentOS] **********
ok: [192.168.56.105]


TASK [CentOS : Enable service logstash for CentOS] **********
changed: [192.168.56.105]
TASK [CentOS : ensure logstash is running for CentOS] **********
changed: [192.168.56.105]
TASK [CentOS : Install Kibana for CentOS] **********
changed: [192.168.56.105]
TASK [CentOS : Configure Kibana Add server.port for CentOS] **********
changed: [192.168.56.105]
TASK [CentOS : Configure Kibana Add server.host for CentOS] **********
changed: [192.168.56.105]
TASK [CentOS : Configure Kibana Add server.name for CentOS] **********
changed: [192.168.56.105]
TASK [CentOS : Configure Kibana Add elasticsearch.hosts for CentOS] **********
changed: [192.168.56.105]


TASK [CentOS : Configure Kibana Add elasticsearch.hosts for CentOS] **********
changed: [192.168.56.105]
TASK [CentOS : Run daemon-reload for kibana for CentOS] **********
ok: [192.168.56.105]
TASK [CentOS : Enable service Kibana for CentOS] **********
changed: [192.168.56.105]
TASK [CentOS : Start Elasticsearch for CentOS] **********
changed: [192.168.56.105]
TASK [CentOS : Start Kibana for CentOS] **********
changed: [192.168.56.105]
PLAY [all] **********
TASK [Gathering Facts] **********
ok: [192.168.56.105]
ok: [192.168.56.102]

```
TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.105]
ok: [192.168.56.102]

TASK [install apache and php for Ubuntu servers] *******************************
*
skipping: [192.168.56.105]
changed: [192.168.56.102]

TASK [install apache and php for CentOS servers] *******************************
*
skipping: [192.168.56.102]
changed: [192.168.56.105]

TASK [start httpd (CentOS)] ****************************************************
*
skipping: [192.168.56.102]
changed: [192.168.56.105]

PLAY RECAP *********************************************************************
*
192.168.56.102             : ok=5    changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
192.168.56.105             : ok=35   changed=27   unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
```

As you can see here only the CentOS successfully installed. Because my inventory of ubuntu was difference on my created directory in roles. It needs to be same as on the creating roles.

Re-run again the playbook.

```
TASK [ubuntu : Install Elastic Dependencies (Ubuntu)] **************************
*
changed: [192.168.56.102]

TASK [ubuntu : Get PGP Key (Ubuntu)] *******************************************
*
changed: [192.168.56.102]

TASK [ubuntu : Install Elasticsearch sources list (Ubuntu)] ********************
*

changed: [192.168.56.102]

TASK [ubuntu : Install Elasticsearch (Ubuntu)] *********************************
*
changed: [192.168.56.102]

TASK [ubuntu : Configure Elasticsearch cluster name (Ubuntu)] ******************
*
changed: [192.168.56.102]

TASK [ubuntu : Configure Elasticsearch descriptive name (Ubuntu)] **************
*
changed: [192.168.56.102]

TASK [ubuntu : Configure Elasticsearch Adding network.host (Ubuntu)] ***********
*
changed: [192.168.56.102]
```

```
changed: [192.168.56.102]

TASK [ubuntu : Configure Elasticsearch Adding http.port (Ubuntu)] *************
*
changed: [192.168.56.102]

TASK [ubuntu : Configure Elasticsearch Adding discovery.type (Ubuntu)] ********
*
changed: [192.168.56.102]

TASK [ubuntu : Creating empty file for startup-timeout.conf 1 of 2 (Ubuntu)] **
*
changed: [192.168.56.102]

TASK [ubuntu : Creating an empty file for startup-timeout.conf 2 of 2 (Ubuntu)]
 ***
changed: [192.168.56.102]

TASK [ubuntu : Prevent systemd service start operation from timing out (Ubuntu)
] ***
changed: [192.168.56.102]

TASK [ubuntu : Run daemon-reload for elasticsearch (Ubuntu)] ******************
*
ok: [192.168.56.102]

TASK [ubuntu : Enable service Elasticsearch and ensure it is not masked (Ubuntu
)] ***
changed: [192.168.56.102]
```

```
TASK [ubuntu : ensure elasticsearch is running (Ubuntu)] **********************
*
changed: [192.168.56.102]

TASK [ubuntu : Install Logstash (Ubuntu)] *************************************
*
changed: [192.168.56.102]

TASK [ubuntu : Run daemon-reload for logstash (Ubuntu)] ***********************
*
ok: [192.168.56.102]

TASK [ubuntu : Enable service logstash (Ubuntu)] ******************************
*
changed: [192.168.56.102]

TASK [ubuntu : ensure logstash is running (Ubuntu)] ***************************
*
changed: [192.168.56.102]

TASK [ubuntu : Install Kibana (Ubuntu)] ***************************************
*
changed: [192.168.56.102]

TASK [ubuntu : Configure Kibana Add server.port (Ubuntu)] *********************
*
changed: [192.168.56.102]

TASK [ubuntu : Configure Kibana Add server.host (Ubuntu)] *********************
```

```
TASK [ubuntu : Configure Kibana Add server.port (Ubuntu)] *********************
*
changed: [192.168.56.102]

TASK [ubuntu : Configure Kibana Add server.host (Ubuntu)] *********************
*
changed: [192.168.56.102]

TASK [ubuntu : Configure Kibana Add server.name (Ubuntu)] *********************
*
changed: [192.168.56.102]

TASK [ubuntu : Configure Kibana Add elasticsearch.hosts (Ubuntu)] *************
*
changed: [192.168.56.102]

TASK [ubuntu : Run daemon-reload for kibana (Ubuntu)] *************************
*
ok: [192.168.56.102]

TASK [ubuntu : Enable service Kibana (Ubuntu)] ********************************
*
changed: [192.168.56.102]

TASK [ubuntu : Start Elasticsearch service] ***********************************
*
changed: [192.168.56.102]
```

```
TASK [install apache and php for Ubuntu servers] ****************************
*
skipping: [192.168.56.105]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] ****************************
*
skipping: [192.168.56.102]
ok: [192.168.56.105]

TASK [start httpd (CentOS)] *************************************************
*
skipping: [192.168.56.102]
ok: [192.168.56.105]

PLAY RECAP ******************************************************************
*
192.168.56.102             : ok=34   changed=25   unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
192.168.56.105             : ok=35   changed=3    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
```

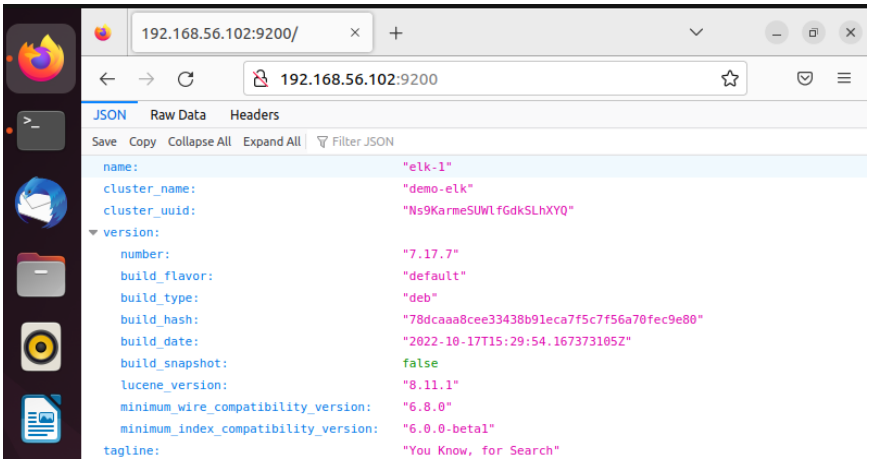
<u>After I edit the inventory name. I re-run again and it was successfully installed the elastic search on my Ubuntu. There is no error found.</u>

Output Ubuntu

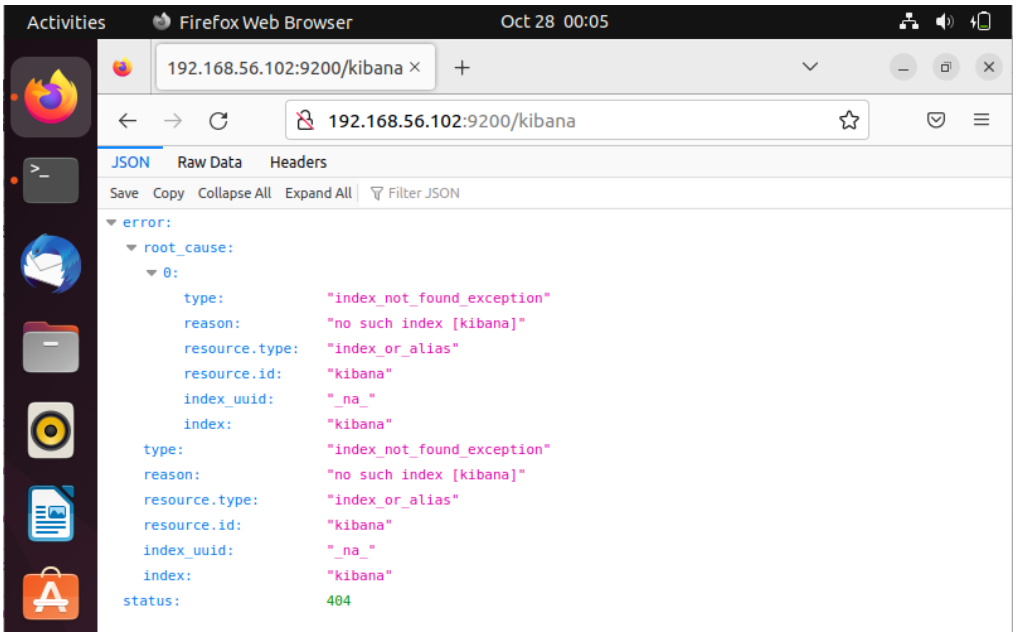
```
bencito@Server1:~$ systemctl status elasticsearch.service
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendo>
    Drop-In: /etc/systemd/system/elasticsearch.service.d
             └─startup-timeout.conf
     Active: active (running) since Fri 2022-10-28 00:01:18 PST; 5min ago
       Docs: https://www.elastic.co
   Main PID: 4697 (java)
      Tasks: 55 (limit: 1640)
     Memory: 184.7M
        CPU: 1min 1.483s
     CGroup: /system.slice/elasticsearch.service
             ├─4697 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.ne>
             └─4860 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux->

Oct 27 23:58:25 Server1 systemd[1]: Starting Elasticsearch...
Oct 28 00:01:18 Server1 systemd[1]: Started Elasticsearch.
lines 1-16/16 (END)
```

```
bencito@Server1:~$ cd /usr/share/elasticsearch
bencito@Server1:/usr/share/elasticsearch$
```

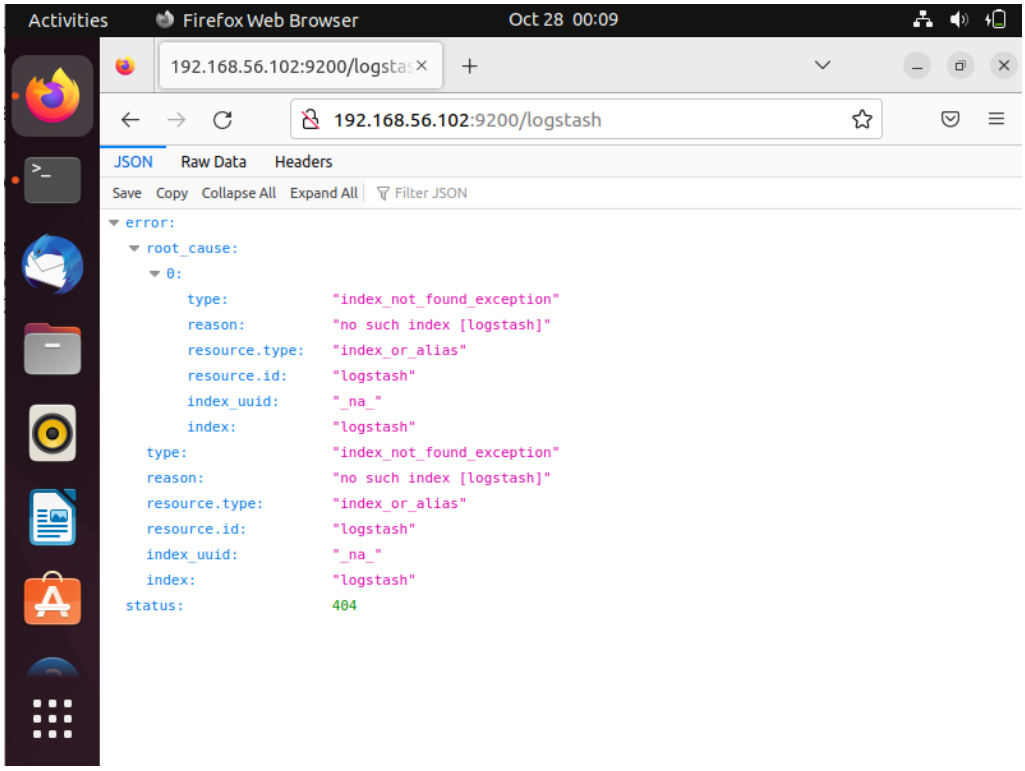
Elastic Search



Kibana



Logstash

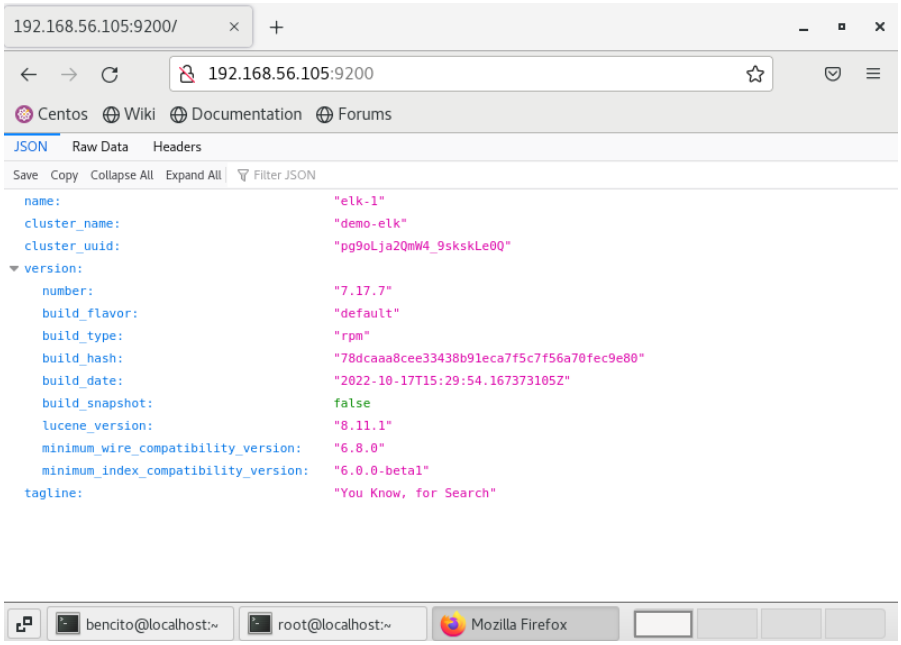


As you can see it was successfully installed on my Ubuntu and It was running smoothly.

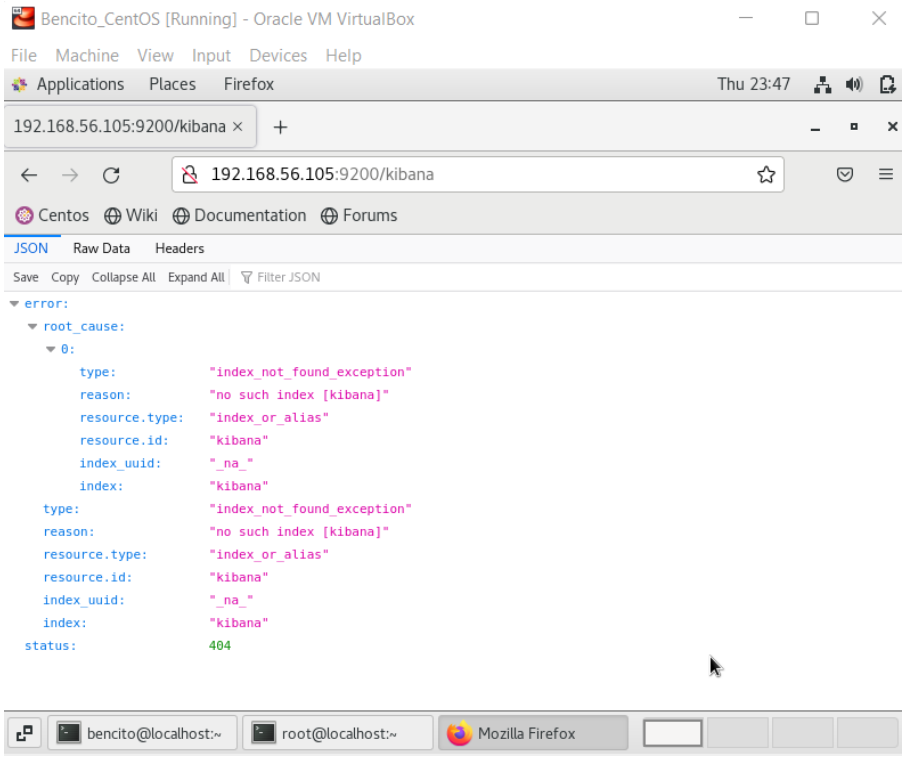
Output CentOS

```
[root@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 Thu 2022-10-27 22:31:51 PST; 1h 39min ago
     Docs: https://www.elastic.co
 Main PID: 5970 (java)
    Tasks: 54
   CGroup: /system.slice/elasticsearch.service
           ├─5970 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.networkadd...
           └─6140 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/b...
```

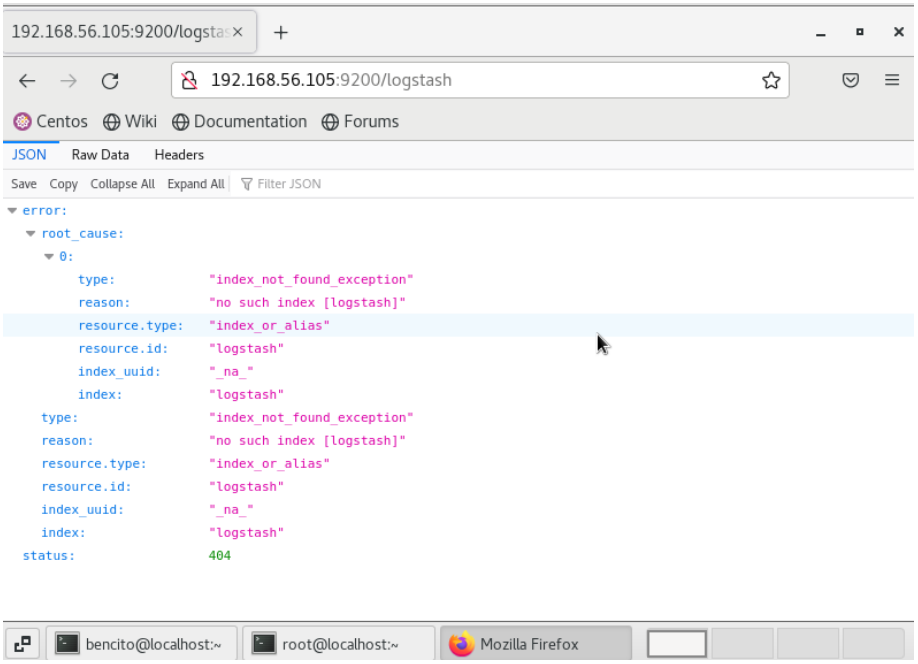
Elastic Search



Kibana



Logstash

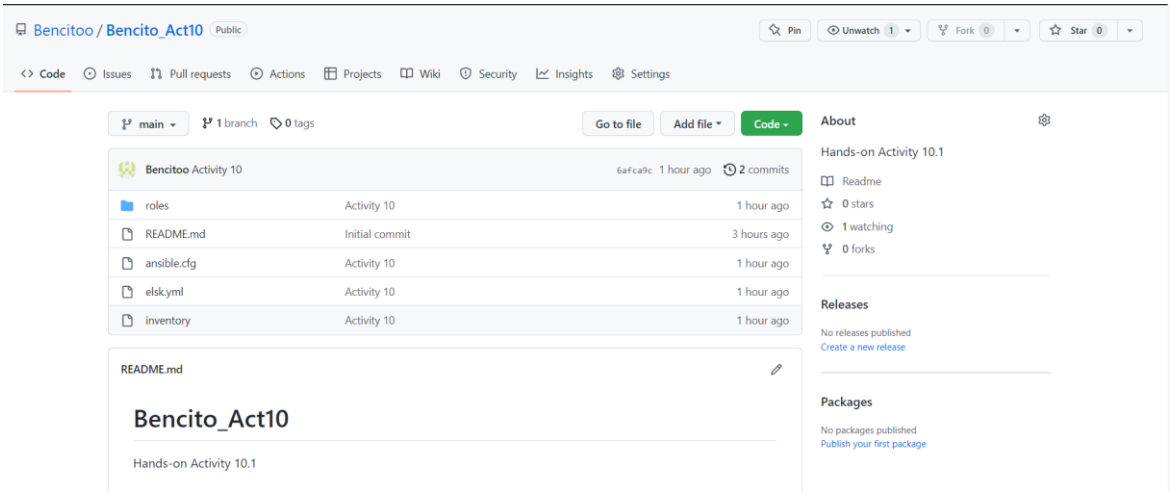


<u>As you can see. Also, on my CentOS it was successfully installed the elastic search and it was running smoothly.</u>

Pushing to Repository

```
bencito@workstation:~/Bencito_Act10$
bencito@workstation:~/Bencito_Act10$ git add inventory
bencito@workstation:~/Bencito_Act10$ git add ansible.cfg
bencito@workstation:~/Bencito_Act10$ git add elsk.yml
bencito@workstation:~/Bencito_Act10$ git add roles/
```

```
bencito@workstation:~/Bencito_Act10$ git commit -m "Activity 10"
[main 6afca9c] Activity 10
 5 files changed, 375 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 elsk.yml
 create mode 100644 inventory
 create mode 100644 roles/CentOS/tasks/main.yml
 create mode 100644 roles/ubuntu/tasks/main.yml
bencito@workstation:~/Bencito_Act10$ git push
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (12/12), 2.53 KiB | 864.00 KiB/s, done.
Total 12 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To github.com:Bencitoo/Bencito_Act10.git
   57e1260..6afca9c  main -> main
bencito@workstation:~/Bencito_Act10$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
bencito@workstation:~/Bencito_Act10$
```



<u>After successfully installed on my control nodes. I git push on my repository.</u>

Reflections:

Answer the following:

1. What are the benefits of having log monitoring tool?

   The benefits of have a log monitoring tool is you can check all the log in details on time and it use to check the security detailed of your system. Also, it used to maintain the security of your system.

Conclusions:

In conclusion, after creating this activity I learn to find a way to get the code of the Elasticsearch because it was my first time creating and after find some references. I learn that you need to have a separate directory for a clean installation. I encounter some minor error's because the capitalization and the name of the main code. It needs to be the same of the main and the creating directory roles. I happy because the result was successful.