Name: Ian Carlo T. Bello	**Date Performed:** October 27, 2022
Course/Section: CPE31S24	**Date Submitted:** October 28, 2022
Instructor: Dr. Jonathan V. Taylar	**Semester and SY:** 1st sem – 3rd year

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.

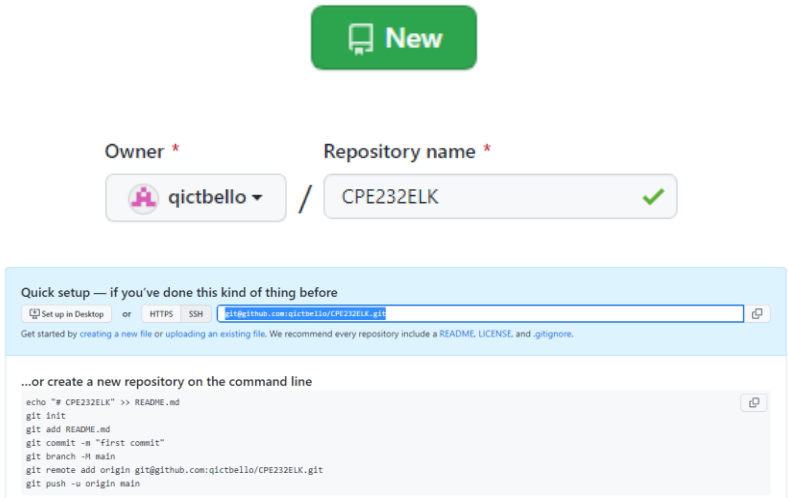
Source: https://www.graylog.org/products/open-source

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)

First we need to create a repository we will name it CPE232ELK



After creating the repository, we will git clone it in our control node or workstation

```
ubuntuhost@workstation:~$ git clone git@github.com:qictbello/CPE232ELK.git
Cloning into 'CPE232ELK'...
warning: You appear to have cloned an empty repository.
ubuntuhost@workstation:~$ cd CPE232ELK/
ubuntuhost@workstation:~/CPE232ELK$
```

We will create the inventory for the Ip of both servers we also put separate host for them, and we will create ansible config

```
ubuntuhost@workstation:~/CPE232ELK$ nano inventory
ubuntuhost@workstation:~/CPE232ELK$ nano ansible.cfg
ubuntuhost@workstation:~/CPE232ELK$
```

```
  GNU nano 6.2                                    inventory *
[ubuntu]
server1

[centos]
servercent
```

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

After creating both needed file we will create the installation tasks for each host we will name it centoselk and ubuntuelk both will have the installation of the 3 tools

```
ubuntuhost@workstation:~/CPE232ELK$ mkdir roles
ubuntuhost@workstation:~/CPE232ELK$ cd roles/
ubuntuhost@workstation:~/CPE232ELK/roles$ mkdir centoselk
ubuntuhost@workstation:~/CPE232ELK/roles$ mkdir ubuntuelk
ubuntuhost@workstation:~/CPE232ELK/roles$ cd centoselk/
ubuntuhost@workstation:~/CPE232ELK/roles/centoselk$ mkdir tasks
ubuntuhost@workstation:~/CPE232ELK/roles/centoselk$ cd ..
ubuntuhost@workstation:~/CPE232ELK/roles$ cd ubuntuelk/
ubuntuhost@workstation:~/CPE232ELK/roles/ubuntuelk$ mkdir tasks
ubuntuhost@workstation:~/CPE232ELK/roles/ubuntuelk$ cd tasks/
ubuntuhost@workstation:~/CPE232ELK/roles/ubuntuelk/tasks$
```

We will create tasks for ubuntu first then cd to centoselk to create tasks for centos this tasks will install all three tools and configure/modify them to run.

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

```yaml
- 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=3min

- 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
```

```yaml
- 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
```

```yaml
- 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
```

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

Then we will create tasks for centos installation same as ubuntu but for centos

```yaml
- 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
```

```yaml
- 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
```

```yaml
- 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=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
```

```yaml
- 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
```

```yaml
- 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
```

Both tasks will install, configure, modify, and start the tools. After creating role task, we will create elk.yml to run the ansible playbook

```
---
- 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"

  -  name: install unzip
     package:
       name: unzip

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

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

- 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"
```

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

We updated both servers and install unzip for our packages and run both tasks for specific host after running both task we required to update/install apache and php and run it, in order for our tools to run.

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

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

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

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

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

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

TASK [centoselk : Install Logstash CentOs] *************************************
fatal: [servercent]: FAILED! => {"msg": "Timeout (12s) waiting for privilege escalation prompt: "}

PLAY RECAP *********************************************************************
server1                    : ok=32   changed=25   unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
servercent                 : ok=19   changed=14   unreachable=0    failed=1    skipped=1    rescued=0    ignored=0
```

After running first time I got timeout but everything is successful, we will run the playbook again after adding timeout=900 in the config and also timeout in opening elastic search

```
ubuntuhost@workstation:~/CPE232ELK$ ansible-playbook --ask-become-pass elk.yml
BECOME password:

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

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

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] *********************
ok: [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] ******************
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] *************************************
changed: [servercent]

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

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

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

TASK [centoselk : Install Kibana for CentOs] ***********************************
changed: [servercent]

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

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

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

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

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

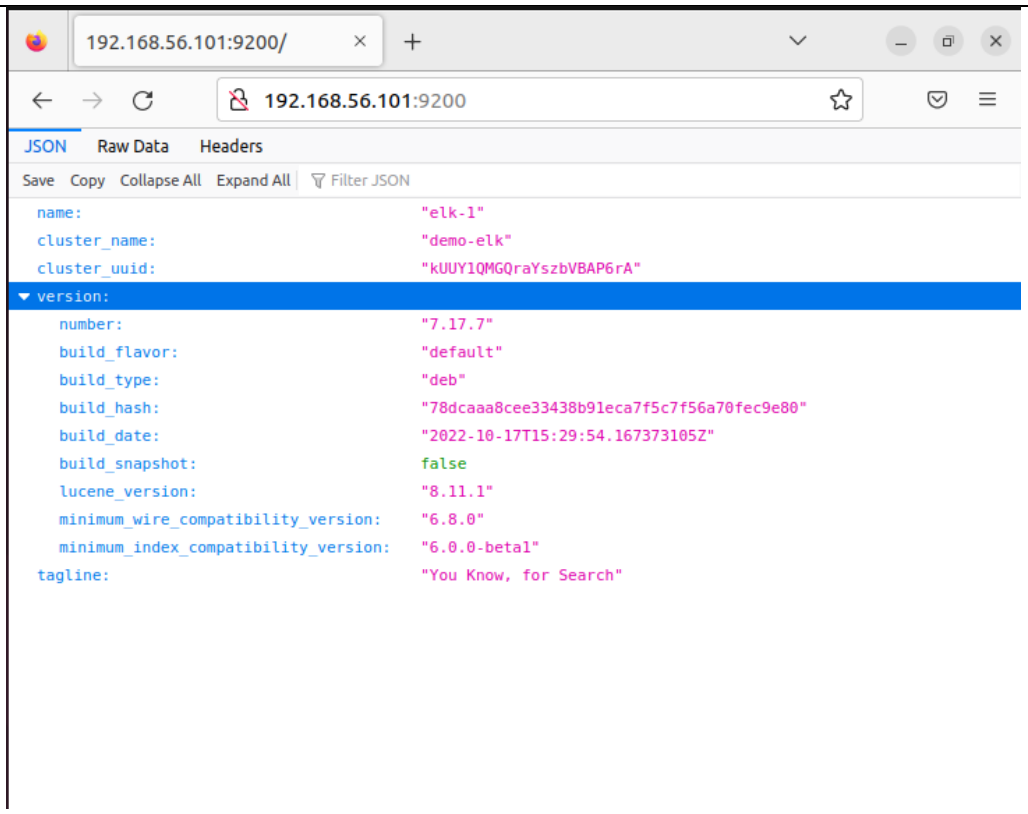
```
TASK [centoselk : Enable service Kibana for CentOs] ***************************
changed: [servercent]

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

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

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

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

TASK [install apache and php for Ubuntu servers] ******************************
skipping: [servercent]
ok: [server1]

TASK [install apache and php for CentOS servers] ******************************
skipping: [server1]
ok: [servercent]

TASK [start httpd CentOS] *****************************************************
skipping: [server1]
ok: [servercent]

PLAY RECAP ********************************************************************
server1                    : ok=34   changed=3    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
servercent                 : ok=35   changed=12   unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

After tweaking timeouts we finally install all of the tools now we will check their output if they're running

UBUNTU

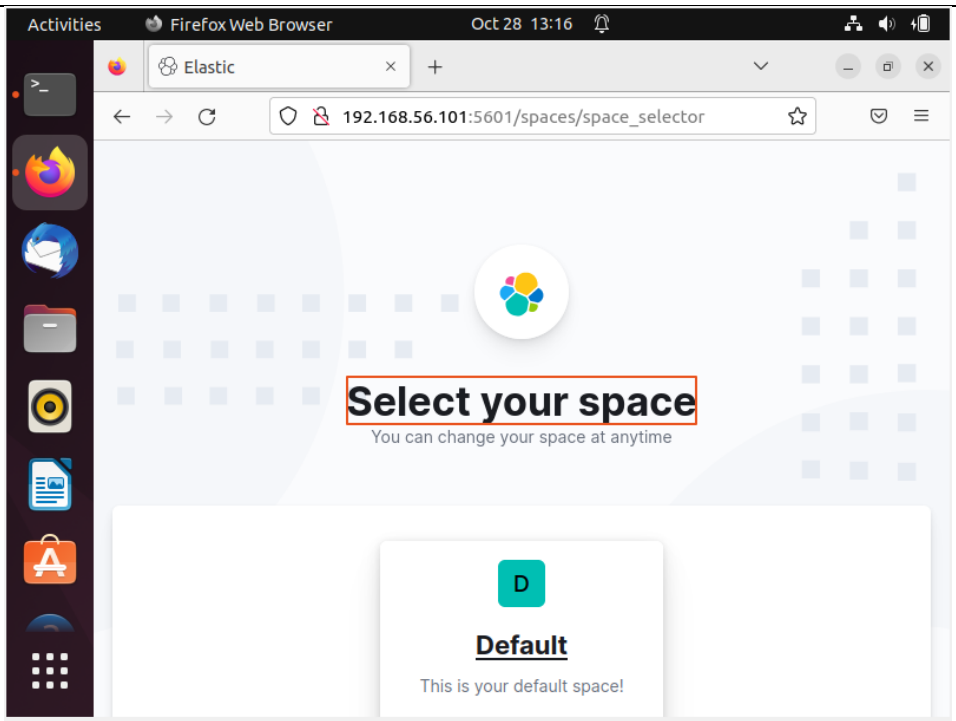
elasticsearch

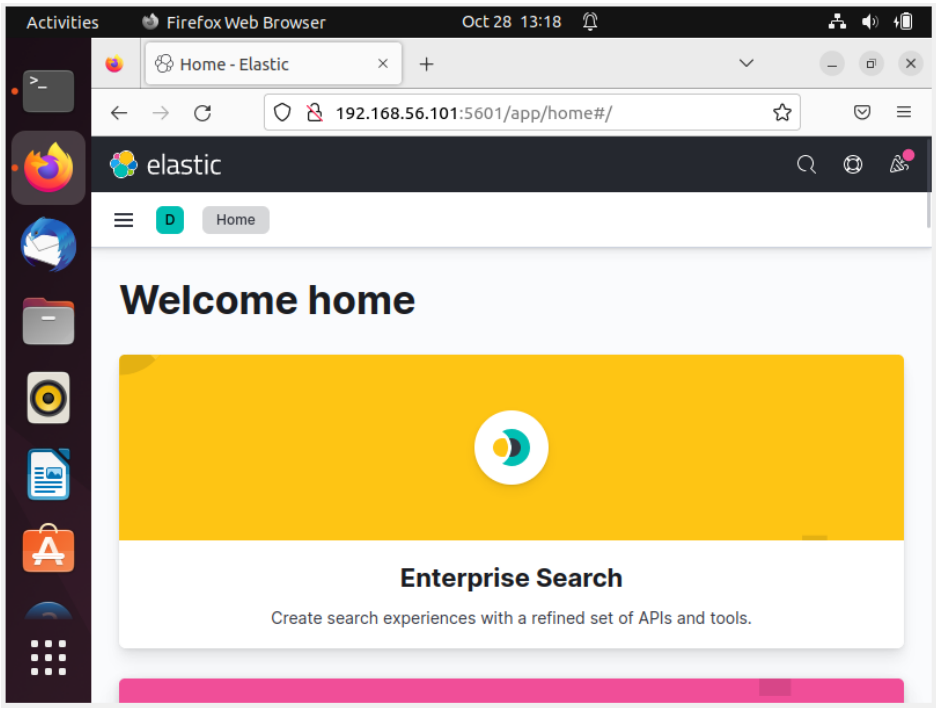
```
ubuntuhost@server1:~$ systemctl status elasticsearch.service
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendo>
    Drop-In: /etc/systemd/system/elasticsearch.service.d
             └─startup-timeout.conf
     Active: active (running) since Fri 2022-10-28 05:28:31 PST; 3min 40s ago
       Docs: https://www.elastic.co
   Main PID: 1127 (java)
      Tasks: 53 (limit: 5865)
     Memory: 2.3G
        CPU: 46.432s
     CGroup: /system.slice/elasticsearch.service
             ├─1127 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.ne>
             └─2888 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux->

Oct 28 05:24:52 server1 systemd[1]: Starting Elasticsearch...
Oct 28 05:28:31 server1 systemd[1]: Started Elasticsearch.
lines 1-16/16 (END)
```



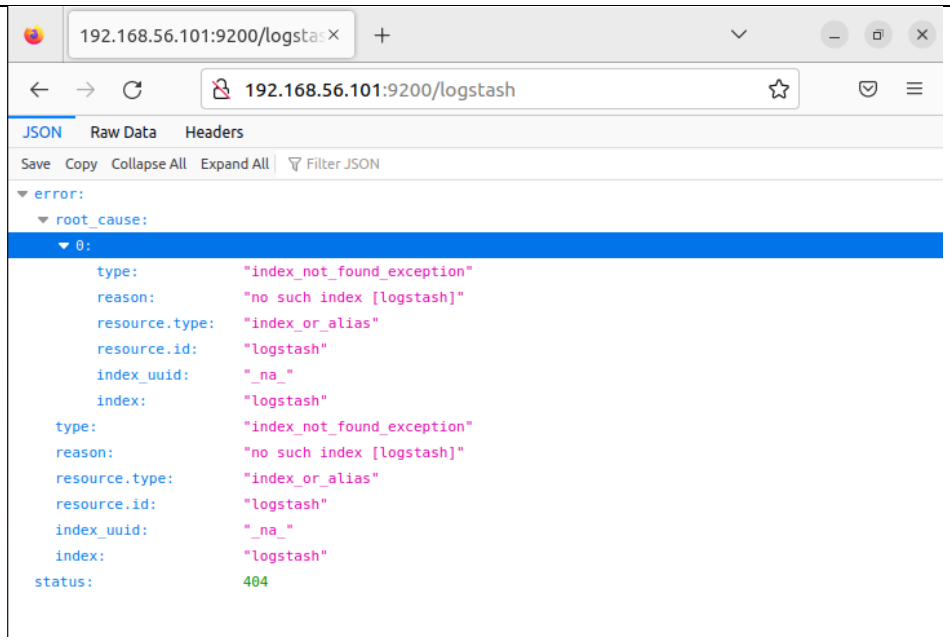
Kibana

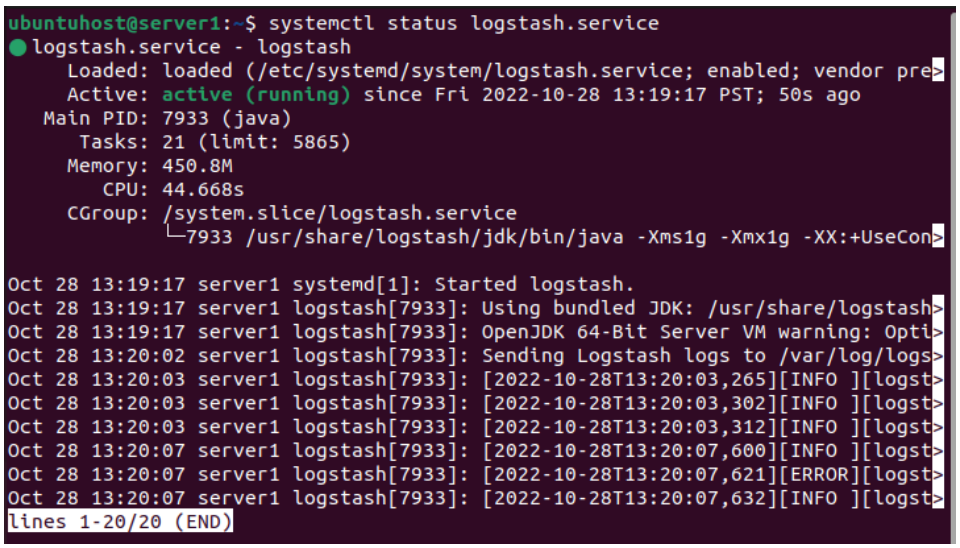
```
ubuntuhost@server1:~$ systemctl status kibana.service
● kibana.service - Kibana
     Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor prese>
     Active: active (running) since Fri 2022-10-28 13:14:08 PST; 5min ago
       Docs: https://www.elastic.co
   Main PID: 7102 (node)
      Tasks: 11 (limit: 5865)
     Memory: 394.1M
        CPU: 27.121s
     CGroup: /system.slice/kibana.service
             └─7102 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/b>

Oct 28 13:14:08 server1 systemd[1]: Started Kibana.
lines 1-12/12 (END)
```


Activities
Firefox Web Browser
Oct 28 13:16
Elastic
192.168.56.101:5601/spaces/space_selector
Select your space
You can change your space at anytime
D
Default
This is your default space!


Activities
Firefox Web Browser
Oct 28 13:18
Home - Elastic
192.168.56.101:5601/app/home#/
elastic
D
Home
Welcome home
Enterprise Search
Create search experiences with a refined set of APIs and tools.

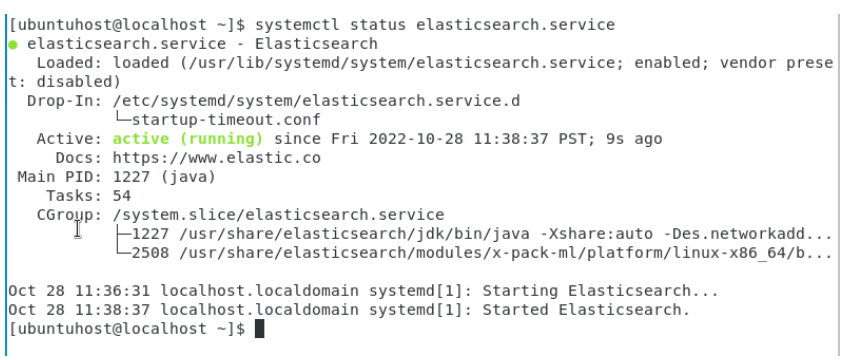
logstash

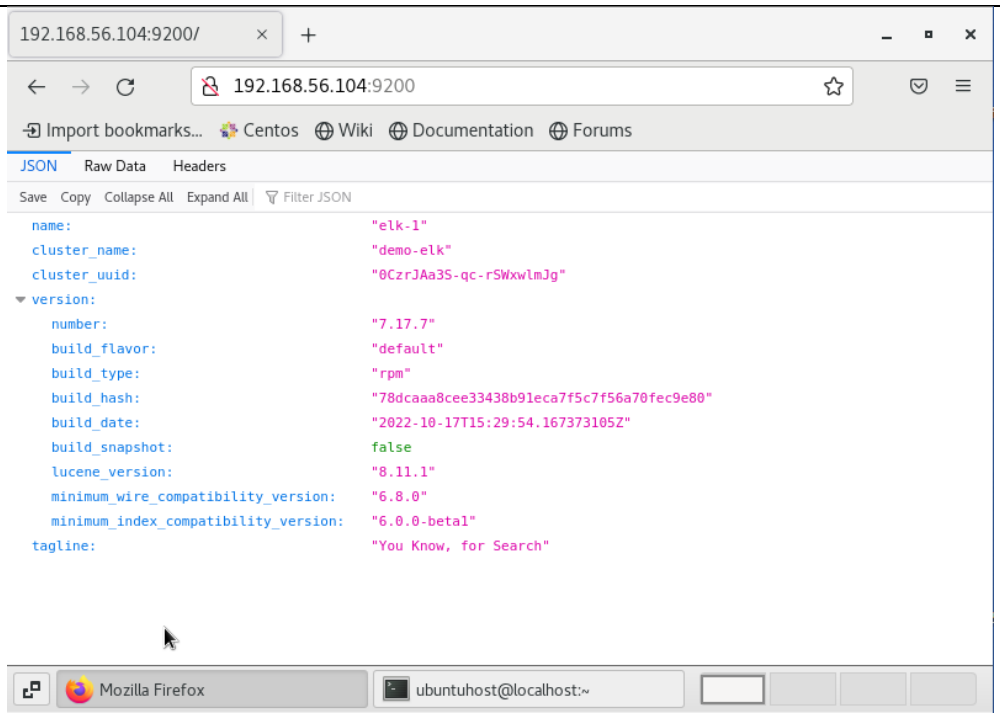




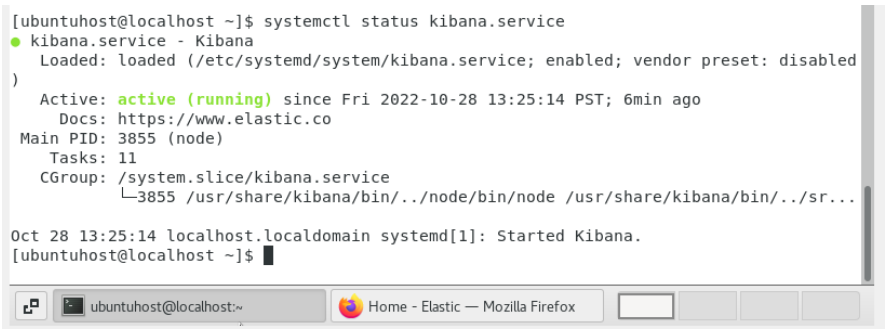
CENTOS

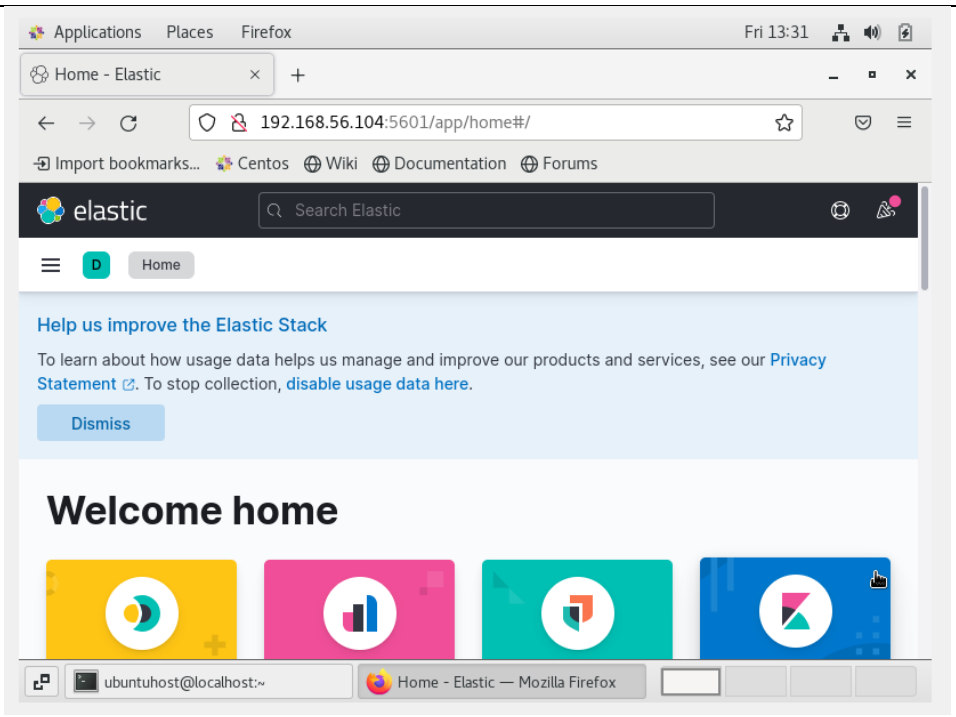
elasticsearch



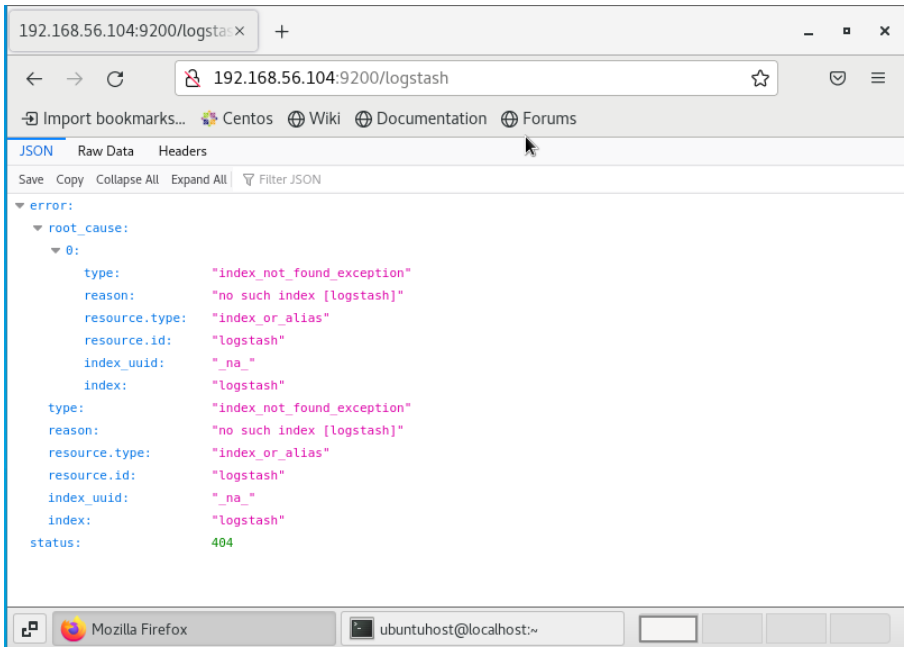

192.168.56.104:9200/
192.168.56.104:9200
Import bookmarks...
Centos
Wiki
Documentation
Forums
JSON
Raw Data
Headers
Save
Copy
Collapse All
Expand All
Filter JSON
name: "elk-1"
cluster_name: "demo-elk"
cluster_uuid: "0CzrJAa3S-qc-rSWxwlmJg"
version:
number: "7.17.7"
build_flavor: "default"
build_type: "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"
Mozilla Firefox
ubuntuhost@localhost:~

Kibana


[ubuntuhost@localhost ~]$ systemctl status kibana.service
● kibana.service - Kibana
Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: disabled
)
Active: active (running) since Fri 2022-10-28 13:25:14 PST; 6min ago
Docs: https://www.elastic.co
Main PID: 3855 (node)
Tasks: 11
CGroup: /system.slice/kibana.service
└─3855 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/../sr...
Oct 28 13:25:14 localhost.localdomain systemd[1]: Started Kibana.
[ubuntuhost@localhost ~]$
ubuntuhost@localhost:~
Home - Elastic — Mozilla Firefox


Applications Places Firefox
Fri 13:31
Home - Elastic
192.168.56.104:5601/app/home#/
Import bookmarks... Centos Wiki Documentation Forums
elastic
Search Elastic
D
Home
Help us improve the Elastic Stack
To learn about how usage data helps us manage and improve our products and services, see our Privacy Statement. To stop collection, disable usage data here.
Dismiss
Welcome home
ubuntuhost@localhost:~
Home - Elastic — Mozilla Firefox

logstash


192.168.56.104:9200/logstash
Import bookmarks... Centos Wiki Documentation Forums
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON
error:
root_cause:
0:
type: "index_not_found_exception"
reason: "no such index [logstash]"
resource.type: "index_or_alias"
resource.id: "logstash"
index_uuid: "_na_"
index: "logstash"
type: "index_not_found_exception"
reason: "no such index [logstash]"
resource.type: "index_or_alias"
resource.id: "logstash"
index_uuid: "_na_"
index: "logstash"
status: 404
Mozilla Firefox
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 Fri 2022-10-28 12:41:52 PST; 10min ago
 Main PID: 8728 (java)
    Tasks: 21
   CGroup: /system.slice/logstash.service
           └─8728 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSwe...

Oct 28 12:41:52 localhost.localdomain systemd[1]: Started logstash.
Oct 28 12:41:52 localhost.localdomain logstash[8728]: Using bundled JDK: /usr/share...k
Oct 28 12:41:52 localhost.localdomain logstash[8728]: OpenJDK 64-Bit Server VM warn....
Oct 28 12:42:34 localhost.localdomain logstash[8728]: Sending Logstash logs to /var...s
Oct 28 12:42:34 localhost.localdomain logstash[8728]: [2022-10-28T12:42:34,921][INF...s
Oct 28 12:42:34 localhost.localdomain logstash[8728]: [2022-10-28T12:42:34,929][INF...}
Oct 28 12:42:34 localhost.localdomain logstash[8728]: [2022-10-28T12:42:34,933][INFO...
Oct 28 12:42:38 localhost.localdomain logstash[8728]: [2022-10-28T12:42:38,724][INF...}
Oct 28 12:42:38 localhost.localdomain logstash[8728]: [2022-10-28T12:42:38,755][ERR....
Oct 28 12:42:38 localhost.localdomain logstash[8728]: [2022-10-28T12:42:38,783][INF...}
Hint: Some lines were ellipsized, use -l to show in full.
[ubuntuhost@localhost ~]$
```

Git add commit and push to repository

```
ubuntuhost@workstation:~/CPE232ELK$ git add -A
ubuntuhost@workstation:~/CPE232ELK$ git commit -m "ELK"
git[main (root-commit) a1ab4f9] ELK
 5 files changed, 368 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 elk.yml
 create mode 100644 inventory
 create mode 100644 roles/centoselk/tasks/main.yml
 create mode 100644 roles/ubuntuelk/tasks/main.yml
ubuntuhost@workstation:~/CPE232ELK$ git push
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (12/12), 2.33 KiB | 793.00 KiB/s, done.
Total 12 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To github.com:qictbello/CPE232ELK.git
 * [new branch]      main -> main
ubuntuhost@workstation:~/CPE232ELK$
```

Reflections:

Answer the following:

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

Having log monitoring tools will help us fix and debug servers easily, and it will be easily seen and debugged from this tool. They collect a log of errors, commands, and executions that control node users can read and write. The tool that we installed scans and monitors the logs that are being generated by the servers. They do detect and alert on these logs and will help the admin to solve the problem.

Conclusions:

In conclusion, we installed log monitoring tools like elasticsearch, kibana, and logstash on both servers. We created roles for separate servers and spliced commands to work. We encountered many problems, especially in elasticsearch, kibana, and the timeout of ansible. This tool eats too much RAM. It makes it difficult for lower-end devices to conduct this activity, but luckily we did finish it smoothly. We can use these tools to monitor logs that detect and alert on problems that will help the admin of the servers debug or fix problems. I do enjoy this activity since I struggle with creating the commands and debugging the tools.