Name: Clarence B. Chavez	**Date Performed:** October 28, 2024
Course/Section: CPE31S2	**Date Submitted:** November 4, 2024
Instructor: Engr. Robin Valenzuela	**Semester and SY:** 1st Sem, 2024-2025

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

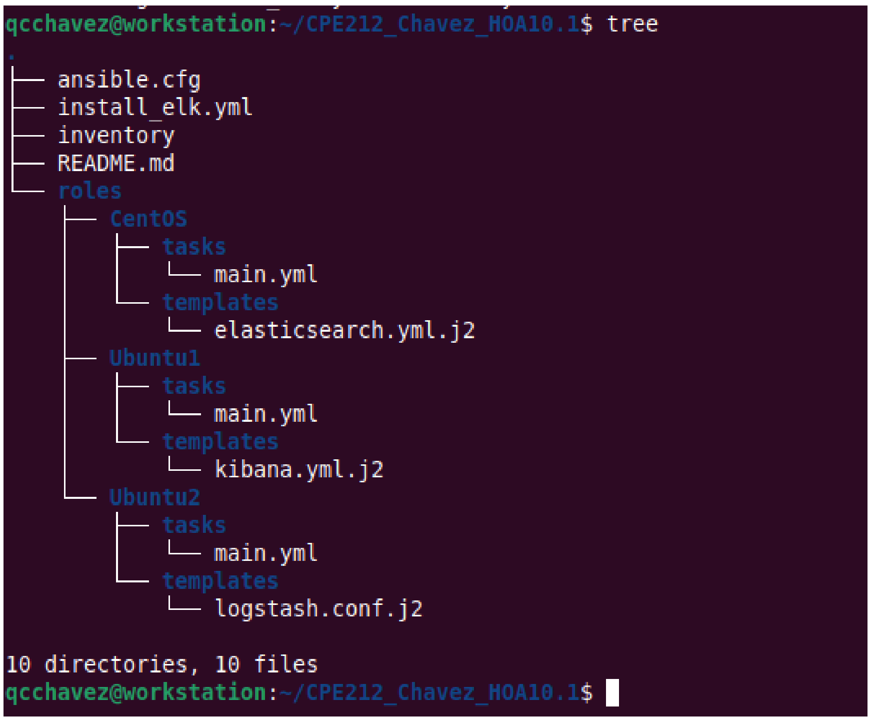
3. Tasks

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)

Task 3.1 - Task 3.3

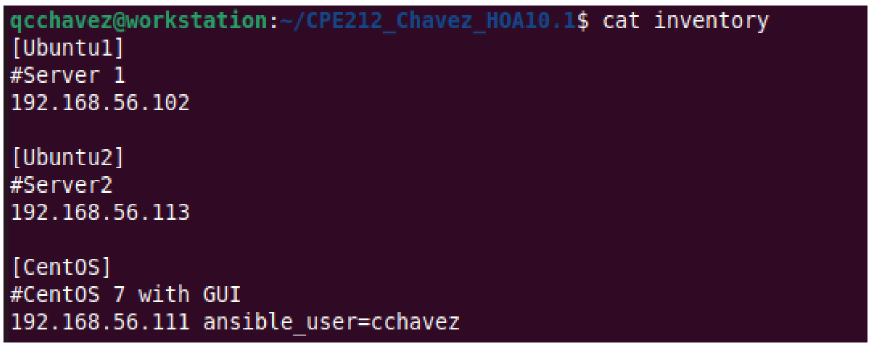
Tree of Files of the Repository for the HOA 10.1



```
qcchavez@workstation:~/CPE212_Chavez_HOA10.1$ tree
.
├── ansible.cfg
├── install_elk.yml
├── inventory
├── README.md
└── roles
    ├── CentOS
    │   ├── tasks
    │   │   └── main.yml
    │   └── templates
    │       └── elasticsearch.yml.j2
    ├── Ubuntu1
    │   ├── tasks
    │   │   └── main.yml
    │   └── templates
    │       └── kibana.yml.j2
    └── Ubuntu2
        ├── tasks
        │   └── main.yml
        └── templates
            └── logstash.conf.j2

10 directories, 10 files
qcchavez@workstation:~/CPE212_Chavez_HOA10.1$
```

1. The content of the repository for this activity should look like this. The ELK components should be installed separately, and also use 1 CentOS 7, and 2 Ubuntus as your remote servers.

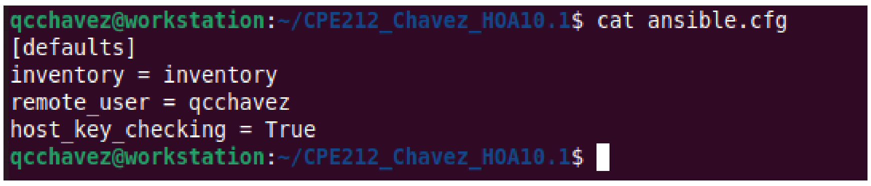
Inventory file



```
qcchavez@workstation:~/CPE212_Chavez_HOA10.1$ cat inventory
[Ubuntu1]
#Server 1
192.168.56.102

[Ubuntu2]
#Server2
192.168.56.113

[CentOS]
#CentOS 7 with GUI
192.168.56.111 ansible_user=cchavez
```

2. Make sure that the content of your inventory file for this activity is correct, double check whether the assigned IP addresses are correct by prompting **ansible all -m ping**.

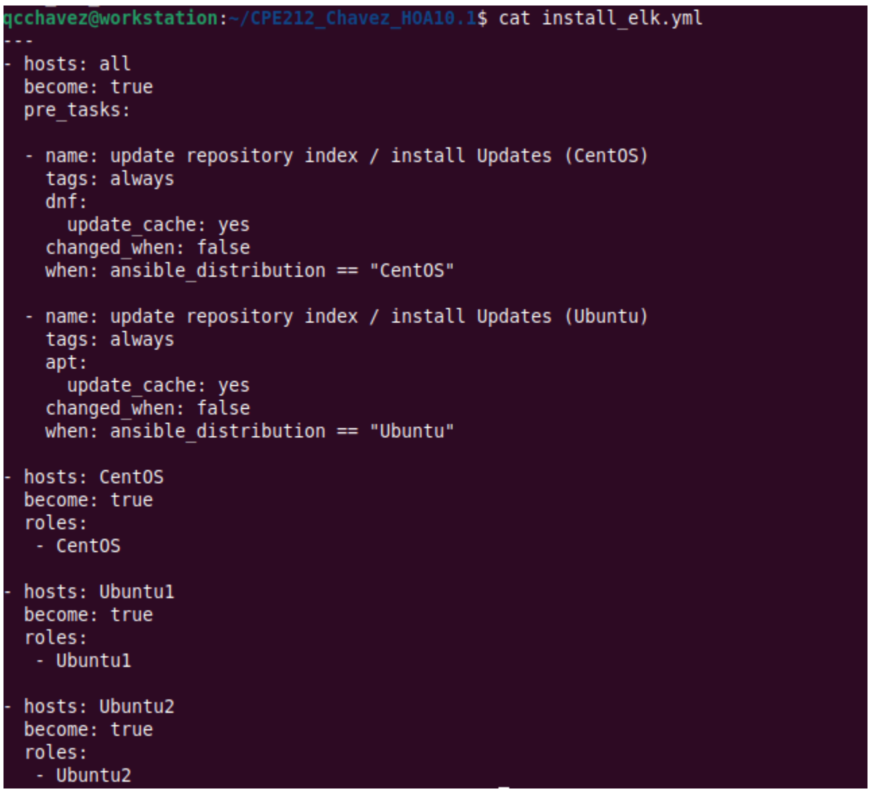
Ansible Configuration File



```
qcchavez@workstation:~/CPE212_Chavez_HOA10.1$ cat ansible.cfg
[defaults]
inventory = inventory
remote_user = qcchavez
host_key_checking = True
qcchavez@workstation:~/CPE212_Chavez_HOA10.1$ 
```

3. Ensure that the ansible configuration file is also correct because this is one of the important files when running the ansible playbook.

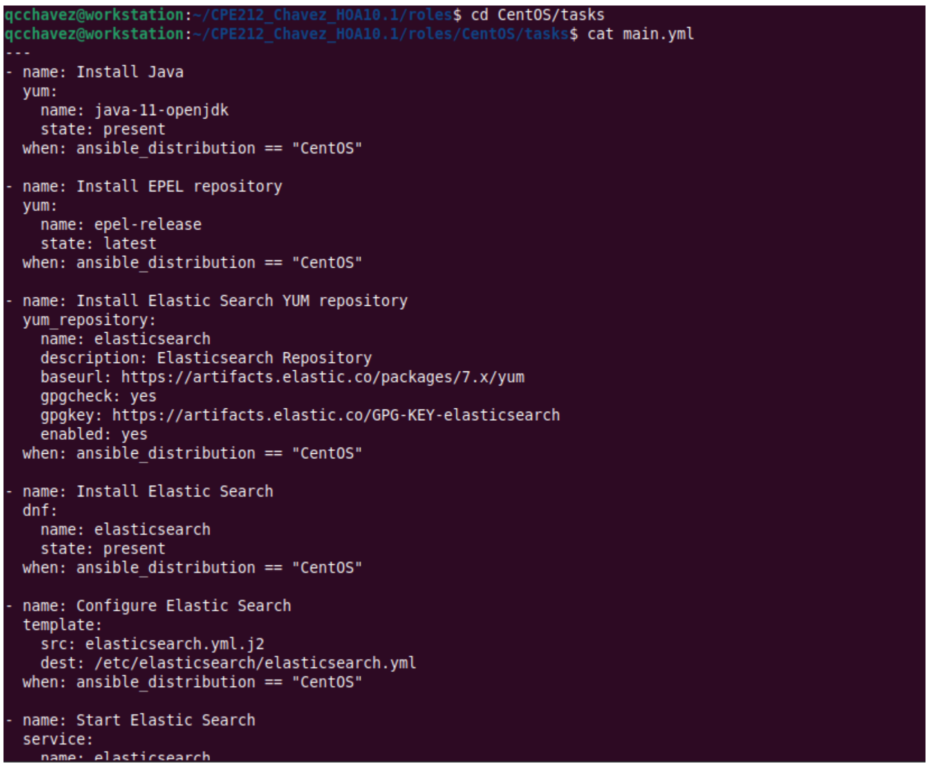
Main Playbook of the Whole Activity



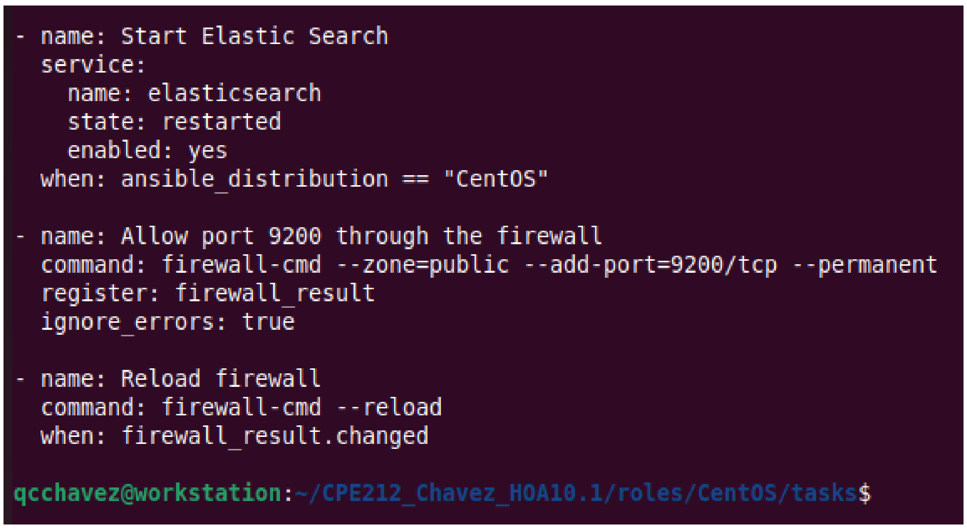
```
qcchavez@workstation:~/CPE212_Chavez_HOA10.1$ cat install_elk.yml
---
- hosts: all
  become: true
  pre_tasks:

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

  - name: update repository index / install Updates (Ubuntu)
    tags: always
    apt:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "Ubuntu"

- hosts: CentOS
  become: true
  roles:
   - CentOS

- hosts: Ubuntu1
  become: true
  roles:
   - Ubuntu1

- hosts: Ubuntu2
  become: true
  roles:
   - Ubuntu2
```

4. The **install_elk.yml** file is the main playbook file for running all of the tasks, this is where the necessary tasks for each remote server happens. Firstly, it updates the repository index for CentOS 7 and Ubuntu. And after that, the tasks for the remote servers (installing, configuring, etc.) will be done.

Playbook Tasks of ElasticSearch (CentOS 7)



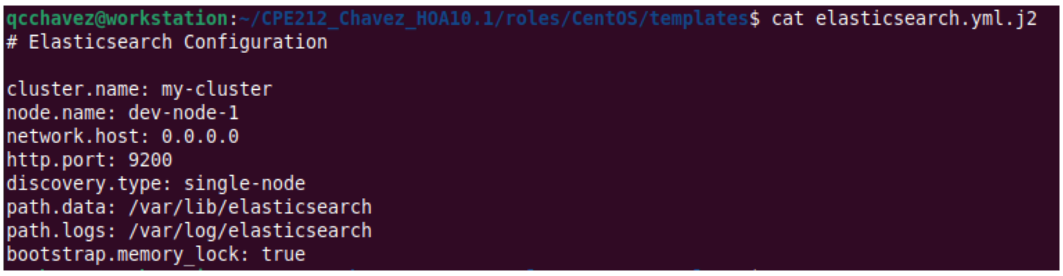
```
qcchavez@workstation:~/CPE212_Chavez_HOA10.1/roles$ cd CentOS/tasks
qcchavez@workstation:~/CPE212_Chavez_HOA10.1/roles/CentOS/tasks$ cat main.yml
---
- name: Install Java
  yum:
    name: java-11-openjdk
    state: present
  when: ansible_distribution == "CentOS"

- name: Install EPEL repository
  yum:
    name: epel-release
    state: latest
  when: ansible_distribution == "CentOS"

- name: Install Elastic Search YUM repository
  yum_repository:
    name: elasticsearch
    description: Elasticsearch Repository
    baseurl: https://artifacts.elastic.co/packages/7.x/yum
    gpgcheck: yes
    gpgkey: https://artifacts.elastic.co/GPG-KEY-elasticsearch
    enabled: yes
  when: ansible_distribution == "CentOS"

- name: Install Elastic Search
  dnf:
    name: elasticsearch
    state: present
  when: ansible_distribution == "CentOS"

- name: Configure Elastic Search
  template:
    src: elasticsearch.yml.j2
    dest: /etc/elasticsearch/elasticsearch.yml
  when: ansible_distribution == "CentOS"

- name: Start Elastic Search
  service:
    name: elasticsearch
```



```
- name: Start Elastic Search
  service:
    name: elasticsearch
    state: restarted
    enabled: yes
  when: ansible_distribution == "CentOS"

- name: Allow port 9200 through the firewall
  command: firewall-cmd --zone=public --add-port=9200/tcp --permanent
  register: firewall_result
  ignore_errors: true

- name: Reload firewall
  command: firewall-cmd --reload
  when: firewall_result.changed

qcchavez@workstation:~/CPE212_Chavez_HOA10.1/roles/CentOS/tasks$
```

5. This is the **main.yml** tasks for **CentOS 7,** first, it will install **Java** since **ElasticSearch** is dependent on it, after that, it will install the **EPEL repository** or the **Extra Packages for Enterprise Linux**. After that, the playbook will search for the yum repository of **ElasticSearch** and will install right after. In order to make **ElasticSearch** run properly, it creates an configuration file right after installing it and after doing all of these, it will start the **ElasticSearch** itself and also will make sure that the **port 9200** in the firewall will work since it is one of the ports that will let the **ELK stacks** connect to each other.

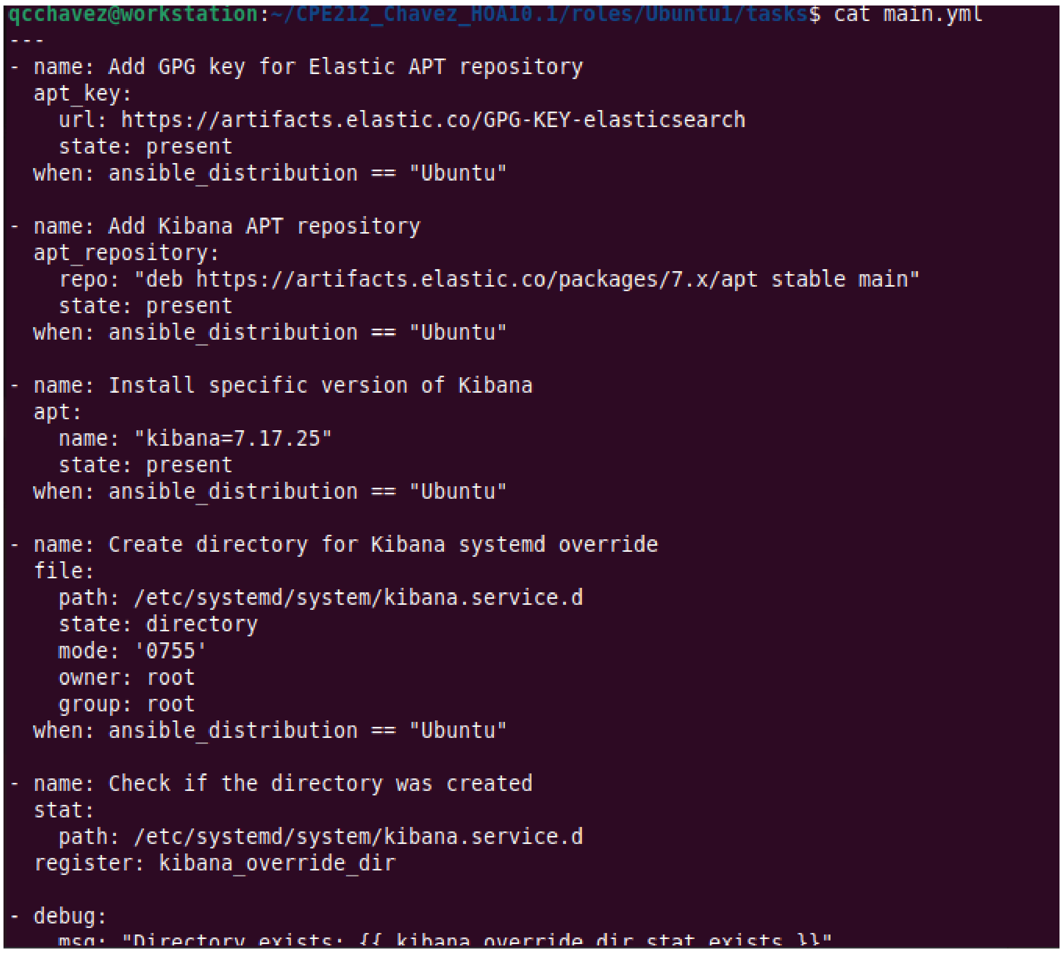
Configuration File of ElasticSearch (CentOS 7)



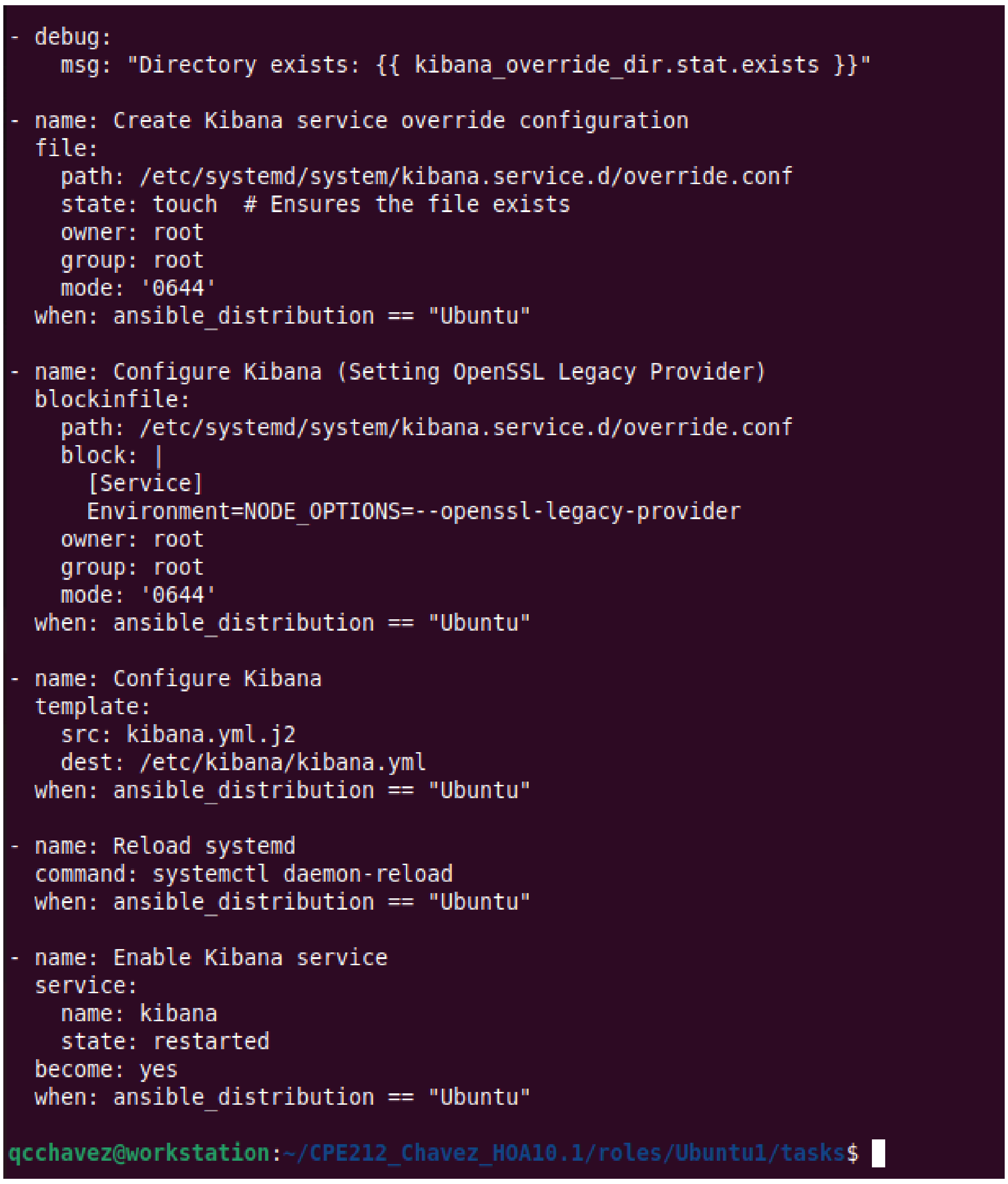
```
qcchavez@workstation:~/CPE212_Chavez_HOA10.1/roles/CentOS/templates$ cat elasticsearch.yml.j2
# Elasticsearch Configuration

cluster.name: my-cluster
node.name: dev-node-1
network.host: 0.0.0.0
http.port: 9200
discovery.type: single-node
path.data: /var/lib/elasticsearch
path.logs: /var/log/elasticsearch
bootstrap.memory_lock: true
```

6. Since a configuration file is required for every ELK stack component, this is the configuration file needed for ElasticSearch. It also includes the one that I've mentioned earlier which is the **http.port**.

Playbook Tasks of Kibana (Ubuntu Desktop 1)



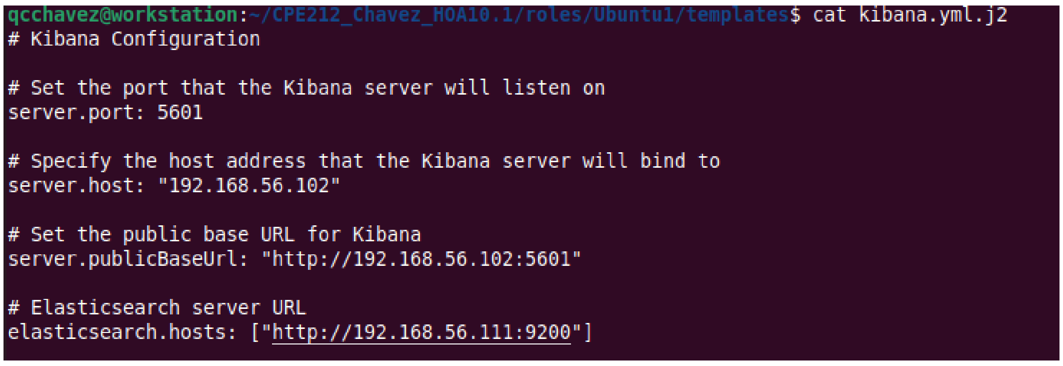
```
qcchavez@workstation:~/CPE212_Chavez_HOA10.1/roles/Ubuntu1/tasks$ cat main.yml
---
- name: Add GPG key for Elastic APT repository
  apt_key:
    url: https://artifacts.elastic.co/GPG-KEY-elasticsearch
    state: present
  when: ansible_distribution == "Ubuntu"

- name: Add Kibana APT repository
  apt_repository:
    repo: "deb https://artifacts.elastic.co/packages/7.x/apt stable main"
    state: present
  when: ansible_distribution == "Ubuntu"

- name: Install specific version of Kibana
  apt:
    name: "kibana=7.17.25"
    state: present
  when: ansible_distribution == "Ubuntu"

- name: Create directory for Kibana systemd override
  file:
    path: /etc/systemd/system/kibana.service.d
    state: directory
    mode: '0755'
    owner: root
    group: root
  when: ansible_distribution == "Ubuntu"

- name: Check if the directory was created
  stat:
    path: /etc/systemd/system/kibana.service.d
  register: kibana_override_dir

- debug:
    msg: "Directory exists: {{ kibana_override_dir.stat.exists }}"
```



```
- debug:
    msg: "Directory exists: {{ kibana_override_dir.stat.exists }}"

- name: Create Kibana service override configuration
  file:
    path: /etc/systemd/system/kibana.service.d/override.conf
    state: touch  # Ensures the file exists
    owner: root
    group: root
    mode: '0644'
  when: ansible_distribution == "Ubuntu"

- name: Configure Kibana (Setting OpenSSL Legacy Provider)
  blockinfile:
    path: /etc/systemd/system/kibana.service.d/override.conf
    block: |
      [Service]
      Environment=NODE_OPTIONS=--openssl-legacy-provider
    owner: root
    group: root
    mode: '0644'
  when: ansible_distribution == "Ubuntu"

- name: Configure Kibana
  template:
    src: kibana.yml.j2
    dest: /etc/kibana/kibana.yml
  when: ansible_distribution == "Ubuntu"

- name: Reload systemd
  command: systemctl daemon-reload
  when: ansible_distribution == "Ubuntu"

- name: Enable Kibana service
  service:
    name: kibana
    state: restarted
  become: yes
  when: ansible_distribution == "Ubuntu"

qcchavez@workstation:~/CPE212_Chavez_HOA10.1/roles/Ubuntu1/tasks$ 
```

7. This is the **main.yml** tasks for **Ubuntu1,** first, it will add a GPG key for **Elastic APT repository**, and then it will also add the **Kibana APT repository** in order for the system to find the required **Kibana** file. After installing the required **APT repositories**, it will install the **Kibana** component, make sure also to indicate the exact version because running with different versions of **ElasticSearch** and **Kibana** will lead you to version incompatibilities. Now, it will create a directory for the .service file of Kibana, and create an override configuration. While doing this activity, I mostly encountered SSL issues and made sure that it will not produce errors again by altering the **environment attribute in the service group** of the override configuration file. After that, it will configure the **Kibana**, reload the system daemon and enable the **Kibana** component.

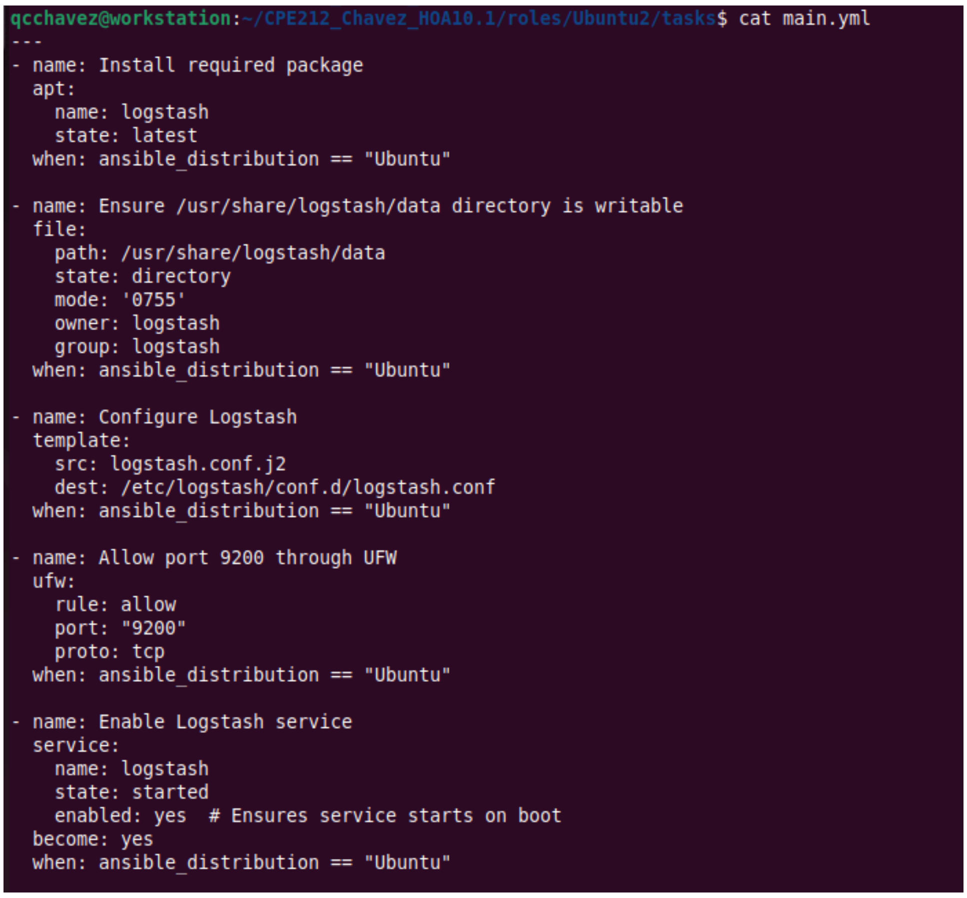
Configuration File of Kibana (Ubuntu Desktop 1)



```
qcchavez@workstation:~/CPE212_Chavez_HOA10.1/roles/Ubuntu1/templates$ cat kibana.yml.j2
# Kibana Configuration

# Set the port that the Kibana server will listen on
server.port: 5601

# Specify the host address that the Kibana server will bind to
server.host: "192.168.56.102"

# Set the public base URL for Kibana
server.publicBaseUrl: "http://192.168.56.102:5601"

# Elasticsearch server URL
elasticsearch.hosts: ["http://192.168.56.111:9200"]
```

8. Just like the ElasticSearch, Kibana also requires a configuration file, this is the content of the kibana configuration file, make sure to double check the assigned IP addresses here as well as the other configuration files that will be or was mentioned. Confirm that the IP address in **server.host and server.publicBaseUrl** is the same as the **IP address:server.port** of your target **Ubuntu** server (the one that will be installed with **Kibana**), and the IP address in **elasticsearch.hosts** is same as the **IP address:http.port** of your target **CentOS** server (the one that is installed with **ElasticSearch**)

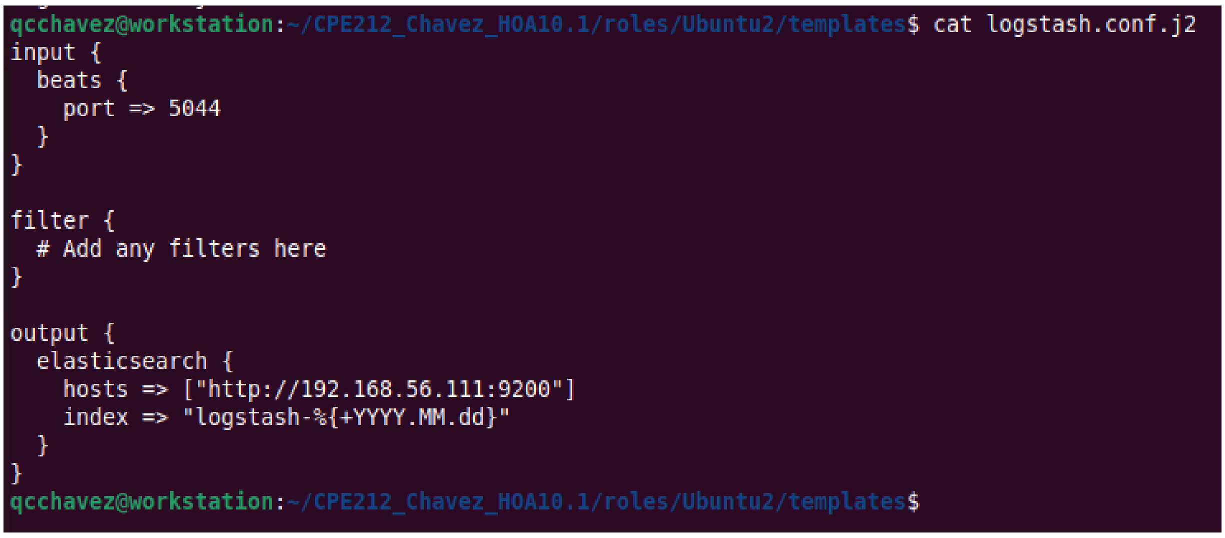
Playbook Tasks of LogStash(Ubuntu Desktop 2)



```
qcchavez@workstation:~/CPE212_Chavez_HOA10.1/roles/Ubuntu2/tasks$ cat main.yml
---
- name: Install required package
  apt:
    name: logstash
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: Ensure /usr/share/logstash/data directory is writable
  file:
    path: /usr/share/logstash/data
    state: directory
    mode: '0755'
    owner: logstash
    group: logstash
  when: ansible_distribution == "Ubuntu"

- name: Configure Logstash
  template:
    src: logstash.conf.j2
    dest: /etc/logstash/conf.d/logstash.conf
  when: ansible_distribution == "Ubuntu"

- name: Allow port 9200 through UFW
  ufw:
    rule: allow
    port: "9200"
    proto: tcp
  when: ansible_distribution == "Ubuntu"

- name: Enable Logstash service
  service:
    name: logstash
    state: started
    enabled: yes  # Ensures service starts on boot
  become: yes
  when: ansible_distribution == "Ubuntu"
```

9. This is the **main.yml** tasks for **Ubuntu2,** first, it will installed the required package which is the **logstash**, it will make sure that the directory data of **logstash** is writable by changing its permissions. After that, a configuration file will be made, and also will configure the firewall by allowing the **port 9200 (http.port)** to be accessible. After doing all these tasks, it will now enable the **logstash** service.

Config File of LogStash (Ubuntu Desktop 2)

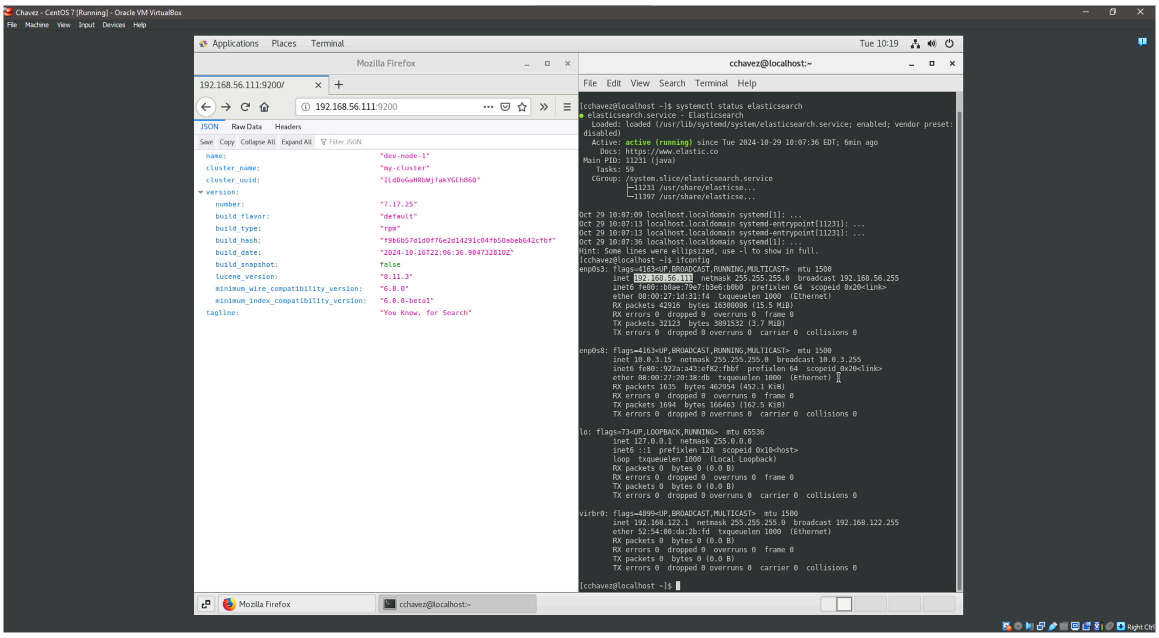


```
qcchavez@workstation:~/CPE212_Chavez_HOA10.1/roles/Ubuntu2/templates$ cat logstash.conf.j2
input {
  beats {
    port => 5044
  }
}

filter {
  # Add any filters here
}

output {
  elasticsearch {
    hosts => ["http://192.168.56.111:9200"]
    index => "logstash-%{+YYYY.MM.dd}"
  }
}
qcchavez@workstation:~/CPE212_Chavez_HOA10.1/roles/Ubuntu2/templates$
```

10. Just like the **ElasticSearch** and **Kibana**, **LogStash** also requires a configuration file. The file's content should include the server.port which is **5044** and the **ElasticSearch** details, most importantly the IP address of the host (**ElasticSearch remote server IP: server.port**)

Task 3.4

ElasticSearch in CentOS 7

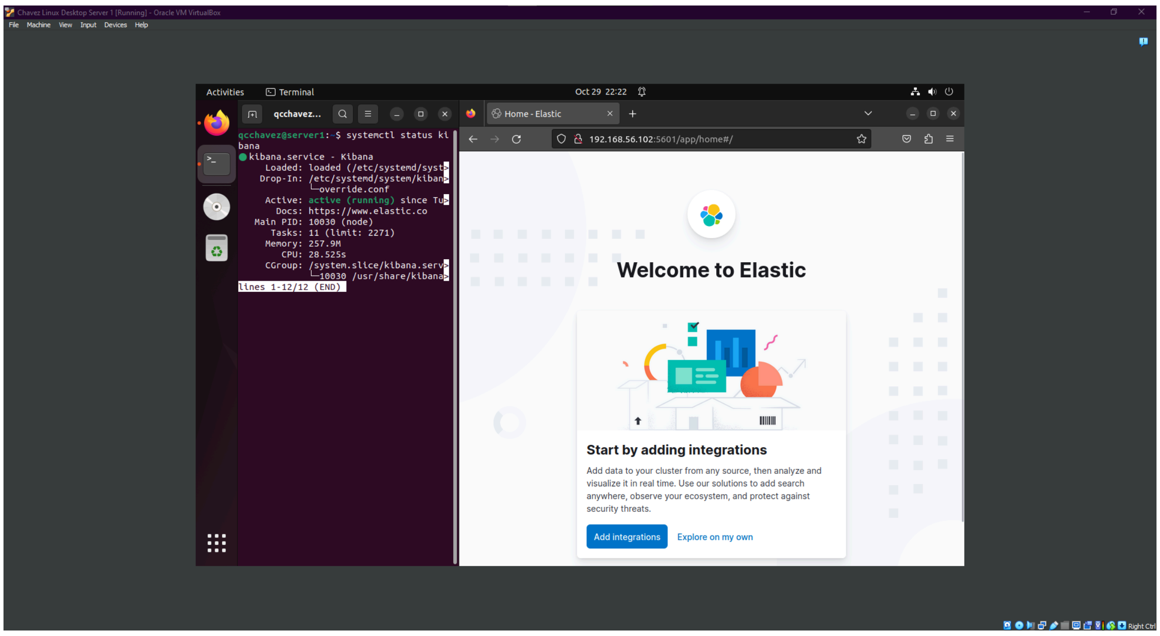

Chavez - CentOS 7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Tue 10:19
Mozilla Firefox
192.168.56.111:9200/
192.168.56.111:9200
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON
cchavez@localhost:~
File Edit View Search Terminal Help
Mozilla Firefox
cchavez@localhost:~
Right Ctrl

```
name:                                 "dev-node-1"
cluster_name:                         "my-cluster"
cluster_uuid:                         "ILdDoGaHRbWjfakYGCh86Q"
▾ version:
  number:                             "7.17.25"
  build_flavor:                       "default"
  build_type:                         "rpm"
  build_hash:                         "f9b6b57d1d0f76e2d14291c04fb50abeb642cfbf"
  build_date:                         "2024-10-16T22:06:36.904732810Z"
  build_snapshot:                     false
  lucene_version:                     "8.11.3"
  minimum_wire_compatibility_version: "6.8.0"
  minimum_index_compatibility_version: "6.0.0-beta1"
tagline:                              "You Know, for Search"
```

```
[cchavez@localhost ~]$ systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2024-10-29 10:07:36 EDT; 6min ago
     Docs: https://www.elastic.co
 Main PID: 11231 (java)
    Tasks: 59
   CGroup: /system.slice/elasticsearch.service
           ├─11231 /usr/share/elasticse...
           └─11397 /usr/share/elasticse...

Oct 29 10:07:09 localhost.localdomain systemd[1]: ...
Oct 29 10:07:13 localhost.localdomain systemd-entrypoint[11231]: ...
Oct 29 10:07:13 localhost.localdomain systemd-entrypoint[11231]: ...
Oct 29 10:07:36 localhost.localdomain systemd[1]: ...
Hint: Some lines were ellipsized, use -l to show in full.
[cchavez@localhost ~]$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.111  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::b8ae:79e7:b3e6:b0b0  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:1d:31:f4  txqueuelen 1000  (Ethernet)
        RX packets 42916  bytes 16308086 (15.5 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 32123  bytes 3891532 (3.7 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.3.15  netmask 255.255.255.0  broadcast 10.0.3.255
        inet6 fe80::922a:a43:ef82:fbbf  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:20:38:db  txqueuelen 1000  (Ethernet)
        RX packets 1635  bytes 462954 (452.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1694  bytes 166463 (162.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.122.1  netmask 255.255.255.0  broadcast 192.168.122.255
        ether 52:54:00:da:2b:fd  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[cchavez@localhost ~]$
```

- To make sure that all of the ELK stack components (ElasticSearch, LogStash, Kibana) are working properly, you can do it by prompting **systemctl status ElasticSearch**, and also, you can enter the **IP-Address-of-ElasticSearch-Remote-Server:server.port (192.168.56.111:9200)** to check if **ElasticSearch** is properly working.

Kibana in Linux Ubuntu Desktop (Server 1)



```
qcchavez@server1:~$ systemctl status kibana
● kibana.service - Kibana
     Loaded: loaded (/etc/systemd/syst>
    Drop-In: /etc/systemd/system/kiban>
             └─override.conf
     Active: active (running) since Tu>
       Docs: https://www.elastic.co
   Main PID: 10030 (node)
      Tasks: 11 (limit: 2271)
     Memory: 257.9M
        CPU: 28.525s
     CGroup: /system.slice/kibana.serv>
             └─10030 /usr/share/kibana>
lines 1-12/12 (END)
```

192.168.56.102:5601/app/home#/

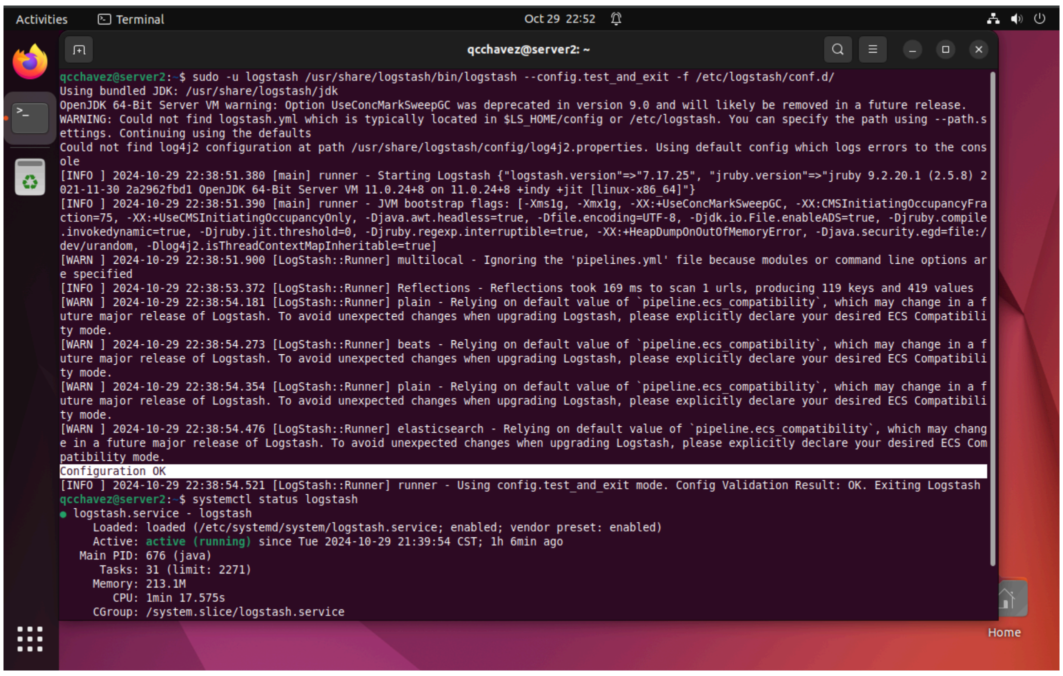
Welcome to Elastic

Start by adding integrations

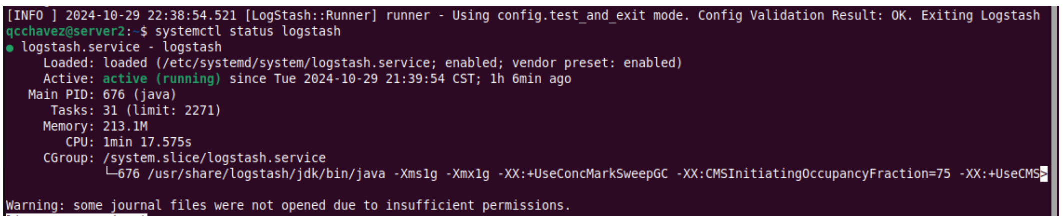
Add data to your cluster from any source, then analyze and visualize it in real time. Use our solutions to add search anywhere, observe your ecosystem, and protect against security threats.

- In Kibana, you can do it by prompting **systemctl status Kibana** , and also, you can enter the **IP-Address-of-Kibana-Remote-Server:server.port (192.168.56.102:5601 in my case)** to check if Kibana is properly working.

LogStash in Linux Ubuntu Desktop (Server 2)



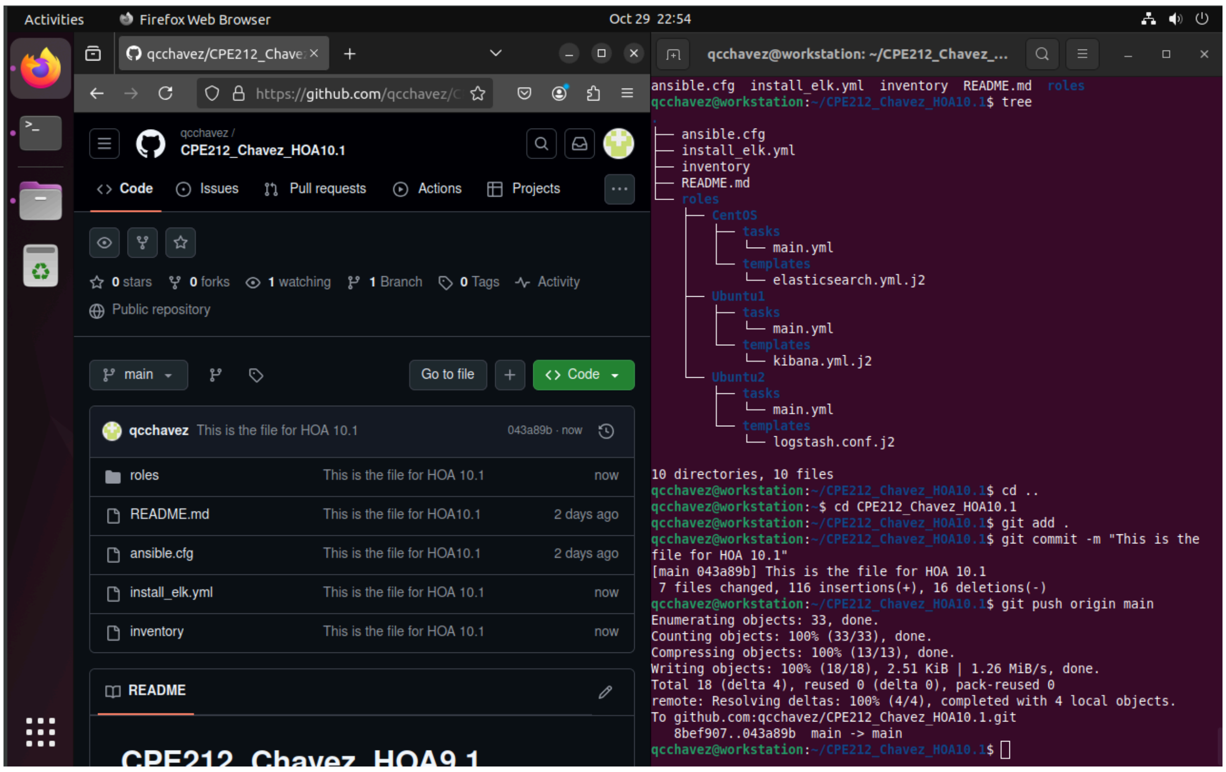
```
qcchavez@server2:~$ sudo -u logstash /usr/share/logstash/bin/logstash --config.test_and_exit -f /etc/logstash/conf.d/
Using bundled JDK: /usr/share/logstash/jdk
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.
WARNING: Could not find logstash.yml which is typically located in $LS_HOME/config or /etc/logstash. You can specify the path using --path.s
ettings. Continuing using the defaults
Could not find log4j2 configuration at path /usr/share/logstash/config/log4j2.properties. Using default config which logs errors to the cons
ole
[INFO ] 2024-10-29 22:38:51.380 [main] runner - Starting Logstash {"logstash.version"=>"7.17.25", "jruby.version"=>"jruby 9.2.20.1 (2.5.8) 2
021-11-30 2a2962fbd1 OpenJDK 64-Bit Server VM 11.0.24+8 on 11.0.24+8 +indy +jit [linux-x86_64]"}
[INFO ] 2024-10-29 22:38:51.390 [main] runner - JVM bootstrap flags: [-Xms1g, -Xmx1g, -XX:+UseConcMarkSweepGC, -XX:CMSInitiatingOccupancyFra
ction=75, -XX:+UseCMSInitiatingOccupancyOnly, -Djava.awt.headless=true, -Dfile.encoding=UTF-8, -Djdk.io.File.enableADS=true, -Djruby.compile
.invokedynamic=true, -Djruby.jit.threshold=0, -Djruby.regexp.interruptible=true, -XX:+HeapDumpOnOutOfMemoryError, -Djava.security.egd=file:/
dev/urandom, -Dlog4j2.isThreadContextMapInheritable=true]
[WARN ] 2024-10-29 22:38:51.900 [LogStash::Runner] multilocal - Ignoring the 'pipelines.yml' file because modules or command line options ar
e specified
[INFO ] 2024-10-29 22:38:53.372 [LogStash::Runner] Reflections - Reflections took 169 ms to scan 1 urls, producing 119 keys and 419 values
[WARN ] 2024-10-29 22:38:54.181 [LogStash::Runner] plain - Relying on default value of `pipeline.ecs_compatibility`, which may change in a f
uture major release of Logstash. To avoid unexpected changes when upgrading Logstash, please explicitly declare your desired ECS Compatibili
ty mode.
[WARN ] 2024-10-29 22:38:54.273 [LogStash::Runner] beats - Relying on default value of `pipeline.ecs_compatibility`, which may change in a f
uture major release of Logstash. To avoid unexpected changes when upgrading Logstash, please explicitly declare your desired ECS Compatibili
ty mode.
[WARN ] 2024-10-29 22:38:54.354 [LogStash::Runner] plain - Relying on default value of `pipeline.ecs_compatibility`, which may change in a f
uture major release of Logstash. To avoid unexpected changes when upgrading Logstash, please explicitly declare your desired ECS Compatibili
ty mode.
[WARN ] 2024-10-29 22:38:54.476 [LogStash::Runner] elasticsearch - Relying on default value of `pipeline.ecs_compatibility`, which may chang
e in a future major release of Logstash. To avoid unexpected changes when upgrading Logstash, please explicitly declare your desired ECS Com
patibility mode.
Configuration OK
[INFO ] 2024-10-29 22:38:54.521 [LogStash::Runner] runner - Using config.test_and_exit mode. Config Validation Result: OK. Exiting Logstash
qcchavez@server2:~$ systemctl status logstash
● logstash.service - logstash
     Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2024-10-29 21:39:54 CST; 1h 6min ago
   Main PID: 676 (java)
      Tasks: 31 (limit: 2271)
     Memory: 213.1M
        CPU: 1min 17.575s
     CGroup: /system.slice/logstash.service
```



```
[INFO ] 2024-10-29 22:38:54.521 [LogStash::Runner] runner - Using config.test_and_exit mode. Config Validation Result: OK. Exiting Logstash
qcchavez@server2:~$ systemctl status logstash
● logstash.service - logstash
     Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2024-10-29 21:39:54 CST; 1h 6min ago
   Main PID: 676 (java)
      Tasks: 31 (limit: 2271)
     Memory: 213.1M
        CPU: 1min 17.575s
     CGroup: /system.slice/logstash.service
             └─676 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMS>

Warning: some journal files were not opened due to insufficient permissions.
```

- In **LogStash**, you can do it by prompting **sudo -u logstash (location of logstash) --config.test_and_exit -f (location of logstash configuration)**, to check the configuration files for errors and at the same time, even without starting it. and also, you can just prompt **systemctl status logstash** to confirm if it is running.

Task 3.5



qcchavez This is the file for HOA 10.1 043a89b · now

roles This is the file for HOA 10.1 now

README.md This is the file for HOA10.1 2 days ago

ansible.cfg This is the file for HOA10.1 2 days ago

install_elk.yml This is the file for HOA 10.1 now

inventory This is the file for HOA 10.1 now

```
ansible.cfg  install_elk.yml  inventory  README.md  roles
qcchavez@workstation:~/CPE212_Chavez_HOA10.1$ tree
.
├── ansible.cfg
├── install_elk.yml
├── inventory
├── README.md
└── roles
    ├── CentOS
    │   ├── tasks
    │   │   └── main.yml
    │   └── templates
    │       └── elasticsearch.yml.j2
    ├── Ubuntu1
    │   ├── tasks
    │   │   └── main.yml
    │   └── templates
    │       └── kibana.yml.j2
    └── Ubuntu2
        ├── tasks
        │   └── main.yml
        └── templates
            └── logstash.conf.j2

10 directories, 10 files
qcchavez@workstation:~/CPE212_Chavez_HOA10.1$ cd ..
qcchavez@workstation:~$ cd CPE212_Chavez_HOA10.1
qcchavez@workstation:~/CPE212_Chavez_HOA10.1$ git add .
qcchavez@workstation:~/CPE212_Chavez_HOA10.1$ git commit -m "This is the file for HOA 10.1"
[main 043a89b] This is the file for HOA 10.1
 7 files changed, 116 insertions(+), 16 deletions(-)
qcchavez@workstation:~/CPE212_Chavez_HOA10.1$ git push origin main
Enumerating objects: 33, done.
Counting objects: 100% (33/33), done.
Compressing objects: 100% (13/13), done.
Writing objects: 100% (18/18), 2.51 KiB | 1.26 MiB/s, done.
Total 18 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To github.com:qcchavez/CPE212_Chavez_HOA10.1.git
   8bef907..043a89b  main -> main
qcchavez@workstation:~/CPE212_Chavez_HOA10.1$ 
```

11. Make sure that all of the **ELK** stack components (**ElasticSearch, LogStash, Kibana**) are working properly. And if they do, commit changes to the **GitHub** repository.

Reflections:

Answer the following:

1. What are the benefits of having a log monitoring tool?
   - The benefits of a log monitoring tool is that you'll be able to see the real-time insights to the performance of the system, being able to troubleshoot more accurately by looking at the logs, increase of security since log monitoring can detect and prohibit unauthorized access.

Conclusions:

- In this activity, I have faced a lot of trials and errors to make the three essential tools to work together. I've struggled a lot on **Kibana** since it is the one that is used for visualizing and interacting with the data stored in **Elasticsearch**. Throughout the process, I encountered several challenges, such as **compatibility issues between versions**, **configuration errors,** and **permission problems**. Each obstacle required careful troubleshooting and adjustments, particularly in **Kibana**, where misconfigurations often led to errors that made my access to the main interface of **Kibana** not accessible since it shows a message "Kibana server is not ready yet".