Name: Renier L. Lope	Date Performed: 10/28/2024
Course/Section: CPE212 - CPE31S2	Date Submitted: 11/04/2024
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)

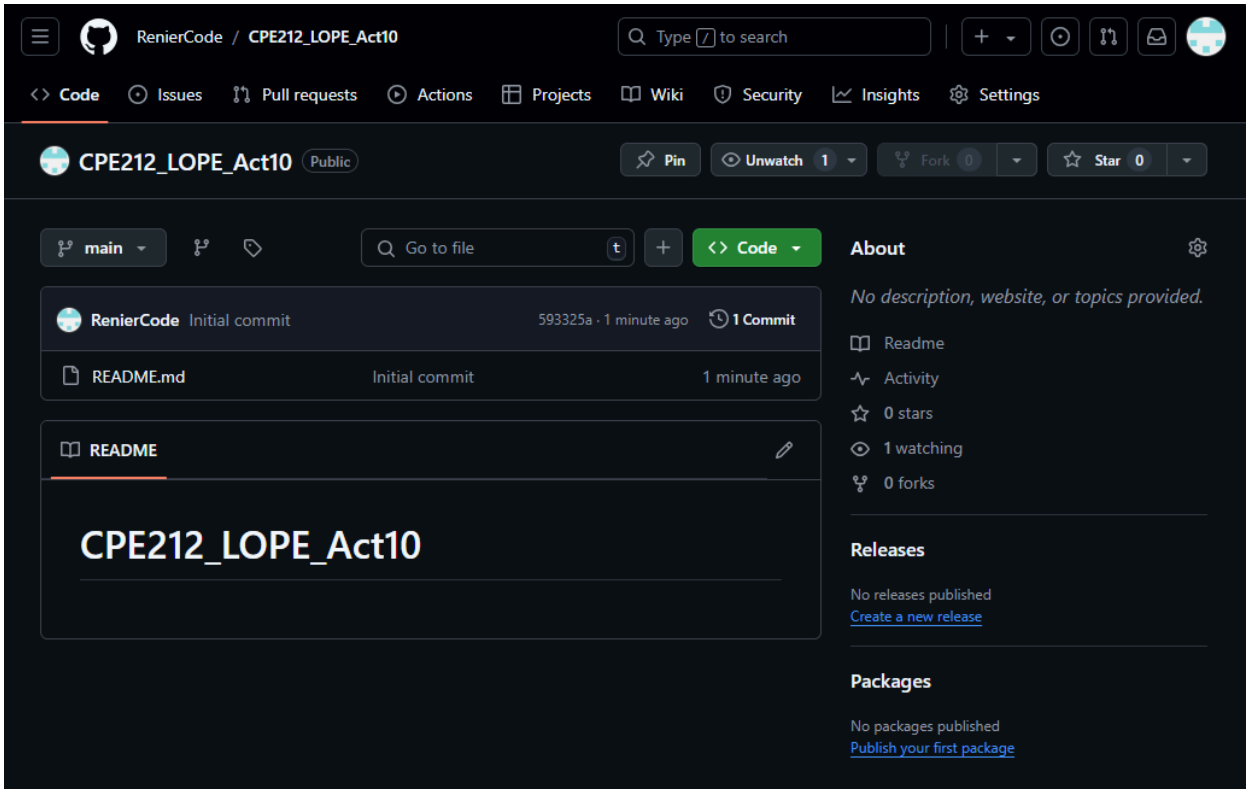


Figure 10.1: Create a new repository in github for this activity.

```
rnrlope@workstation:~$ git clone git@github.com:RenierCode/CPE212_LOPE_Act10
Cloning into 'CPE212_LOPE_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 (from 0)
Receiving objects: 100% (3/3), done.
```

Figure 10.2: Clone the new repository to the local machine.

```
rnrlope@workstation:~$ cp CPE*9/inventory CPE*10
rnrlope@workstation:~$ cp CPE*9/ansible.cfg CPE*10
rnrlope@workstation:~$ cd CPE*10
rnrlope@workstation:~/CPE212_LOPE_Act10$ ls
ansible.cfg  inventory  README.md
rnrlope@workstation:~/CPE212_LOPE_Act10$
```

Figure 10.3: Copy the ansible.cfg and inventory of the previous activity to the new repository.

```
rnrlope@workstation:~/CPE212_LOPE_Act10$ cat ansible.cfg
[defaults]
inventory = inventory
remote_user = rnrlope
host_key_checking = True
deprecation_warnings = False
rnrlope@workstation:~/CPE212_LOPE_Act10$ cat inventory
[web_servers]
server1

[db_servers]
centOS

rnrlope@workstation:~/CPE212_LOPE_Act10$
```

Figure 10.4: Contents of the ansible.cfg and inventory files.

```
rnrlope@workstation:~/CPE212_LOPE_Act10$ nano elasticStack.yml
rnrlope@workstation:~/CPE212_LOPE_Act10$ cat elasticStack.yml
---
- 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:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "Ubuntu"

- hosts: all
  become: true
  roles:
    -  base

- hosts: db_servers:web_servers
  become: true
  roles:
    - requirements
    - elasticSearch
    - kibana
    - logstash

rnrlope@workstation:~/CPE212_LOPE_Act10$
```

Figure 10.5: Create a playbook named “elasticStack.yml”. This playbook will play the task inside the desired roles.

```
rnrlope@workstation:~/CPE212_LOPE_Act10$ mkdir roles
rnrlope@workstation:~/CPE212_LOPE_Act10$ cd roles
rnrlope@workstation:~/CPE212_LOPE_Act10/roles$ mkdir base requirements elasticSearch
kibana logstash
rnrlope@workstation:~/CPE212_LOPE_Act10/roles$ ls
base  elasticSearch  kibana  logstash  requirements
```

Figure 10.6: Create a new directory named "roles", then inside "roles" create new directories named "base", "requirements", "elasticSearch", "kibana", and "logstash".

This roles will contain tasks according to its assigned name.

```
rnrlope@workstation:~/CPE212_LOPE_Act10/roles$ cd base
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/base$ mkdir tasks
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/base$ cd ..
rnrlope@workstation:~/CPE212_LOPE_Act10/roles$ cd requirements
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/requirements$ mkdir tasks
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/requirements$ cd ..
rnrlope@workstation:~/CPE212_LOPE_Act10/roles$ cd elasticSearch
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/elasticSearch$ mkdir tasks
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/elasticSearch$ cd ..
rnrlope@workstation:~/CPE212_LOPE_Act10/roles$
rnrlope@workstation:~/CPE212_LOPE_Act10/roles$ cd kibana
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/kibana$ mkdir tasks
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/kibana$ cd ..
rnrlope@workstation:~/CPE212_LOPE_Act10/roles$ cd logstash
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/logstash$
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/logstash$ mkdir tasks
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/logstash$ cd ..
rnrlope@workstation:~/CPE212_LOPE_Act10/roles$
```

Figure 10.7: Create a directory named "tasks" inside the directories under roles. This directory "tasks" will contain the playbooks assigned for each roles.

```
rnrlope@workstation:~/CPE212_LOPE_Act10$ cd roles/base/tasks
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/base/tasks$ nano main.yml
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/base/tasks$ cat main.yml
---
- name: install updates (CentOS)
  tags: always
  dnf:
    update_only: yes
    update_cache: yes
  when: ansible_distribution == "CentOS"

- name: install updates (Ubuntu)
  tags: always
  apt:
    upgrade: dist
    update_cache: yes
  when: ansible_distribution == "Ubuntu"
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/base/tasks$
```

Figure 10.8: Create a playbook file named "main.yml" inside "roles/base/tasks". This playbook will update both CentOS and Ubuntu.

```
rnrlope@workstation:~/CPE212_LOPE_Act10/roles$ cd requirements/tasks
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/requirements/tasks$ nano main.yml
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/requirements/tasks$ cat main.yml
---
- name: install java (Ubuntu)
  apt:
    name: openjdk-11-jdk
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: install java (CentOS)
  dnf:
    name: java-11-openjdk
    state: latest
  when: ansible_distribution == "CentOS"

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

- name: Add GPG key for ElasticSearch (Ubuntu)
  tags: ubuntu
  apt_key:
    url: https://artifacts.elastic.co/GPG-KEY-elasticsearch
    state: present
  when: ansible_distribution == "Ubuntu"

- name: Allow Port 9200 through Firewall (CentOS)
  firewalld:
    zone: public
    port: 9200/tcp
    permanent: yes
    state: enabled
    immediate: yes
  when: ansible_distribution == "CentOS"

- name: Allow Port 9200 through Firewall (Ubuntu)
  ufw:
    rule: allow
    port: 9200
    proto: tcp
  when: ansible_distribution == "Ubuntu"

- name: Add ElasticSearch to APT repository (Ubuntu)
  tags: ubuntu
  apt_repository:
    repo: "deb https://artifacts.elastic.co/packages/7.x/apt stable main"
    #filename: 'elastic-7.x'
  when: ansible_distribution == "Ubuntu"

- name: Install ElasticSearch to Yum repository (CentOS)
  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"
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/requirements/tasks$
```

Figure 10.9: Create a playbook file named “main.yml” inside “roles/requirements/tasks”. This playbook will install java, allow port 9200, add gpd key and add elastic to packages of hosts.

```
rnrlope@workstation:~/CPE212_LOPE_Act10/roles$ cd elasticSearch/tasks
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/elasticSearch/tasks$ nano main.yml
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/elasticSearch/tasks$ cat main.yml
---
- name: Configure ElasticSearch
  blockinfile:
    path: /etc/elasticsearch/elasticsearch.yml
    block: |
      # 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
    state: present
    create: yes

- name: Install ElasticSearch
  package:
    name:
      - elasticsearch
    state: latest

- name: Force systemd to reread configs
  systemd:
    daemon_reload: yes

- name: Enable ElasticSearch Service
  service:
     name: elasticsearch
     enabled: yes
     state: started
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/elasticSearch/tasks$
```

Figure 10.10: Create a playbook file named “main.yml” inside “roles/elasticSearch/tasks”. This playbook will configure, install, and start elasticsearch in hosts.

```
rnrlope@workstation:~/CPE212_LOPE_Act10/roles$ cd kibana/tasks
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/kibana/tasks$ nano main.yml
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/kibana/tasks$ cat main.yml
---
- name: Configure Kibana
  blockinfile:
    path: /etc/kibana/kibana.yml
    block: |
      # Kibana Configuration

      server.port: 5601
      server.host: "localhost"
      server.name: "Lope-Act10"
      elasticsearch.hosts: ["http://localhost:9200"]
      kibana.index: ".kibana"
      elasticsearch.requestTimeout: 180000
    state: present
    create: yes

- name: Install Kibana
  package:
    name:
      - kibana
    state: latest

- name: Force systemd to reread configs
  systemd:
    daemon_reload: yes

- name: Enable Kibana Service
  service:
     name: kibana
     enabled: yes
     state: started
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/kibana/tasks$
```

Figure 10.11: Create a playbook file named “main.yml” inside “roles/kibana/tasks”. This playbook will configure, install, and start kibana in hosts.

```
rnrlope@workstation:~/CPE212_LOPE_Act10/roles$ cd logstash/tasks
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/logstash/tasks$ nano main.yml
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/logstash/tasks$ cat main.yml
---
- name: Configure Logstash
  blockinfile:
    path: /etc/logstash/conf.d/logstash.conf
    block: |
      # Logstash Configuration

      input {
        beats {
          port => 5044
          host => "127.0.0.1"
         }
      }

      filter {
      # nginx access log
        if [source] =~ /\/(access)\d{0,10}\.(log)/ {
          grok {
            match => {"message" => "%{COMBINEDAPACHELOG}"}
            add_tag => ["nginx_access_log"]
          }
          mutate {
            rename => {"timestamp" => "log_timestamp"}
          }
        }
      }

      output {
        elasticsearch { hosts => ["localhost:9200"] }
        stdout { codec => rubydebug }
      }
    state: present
    create: yes
- name: Force systemd to reread configs
  systemd:
    daemon_reload: yes

- name: Enable Logstash Service
  service:
     name: logstash
     enabled: yes
     state: started
rnrlope@workstation:~/CPE212_LOPE_Act10/roles/logstash/tasks$
```

Figure 10.12: Create a playbook file named “main.yml” inside “roles/logstash/tasks”. This playbook will configure, install, and start logstash in hosts.

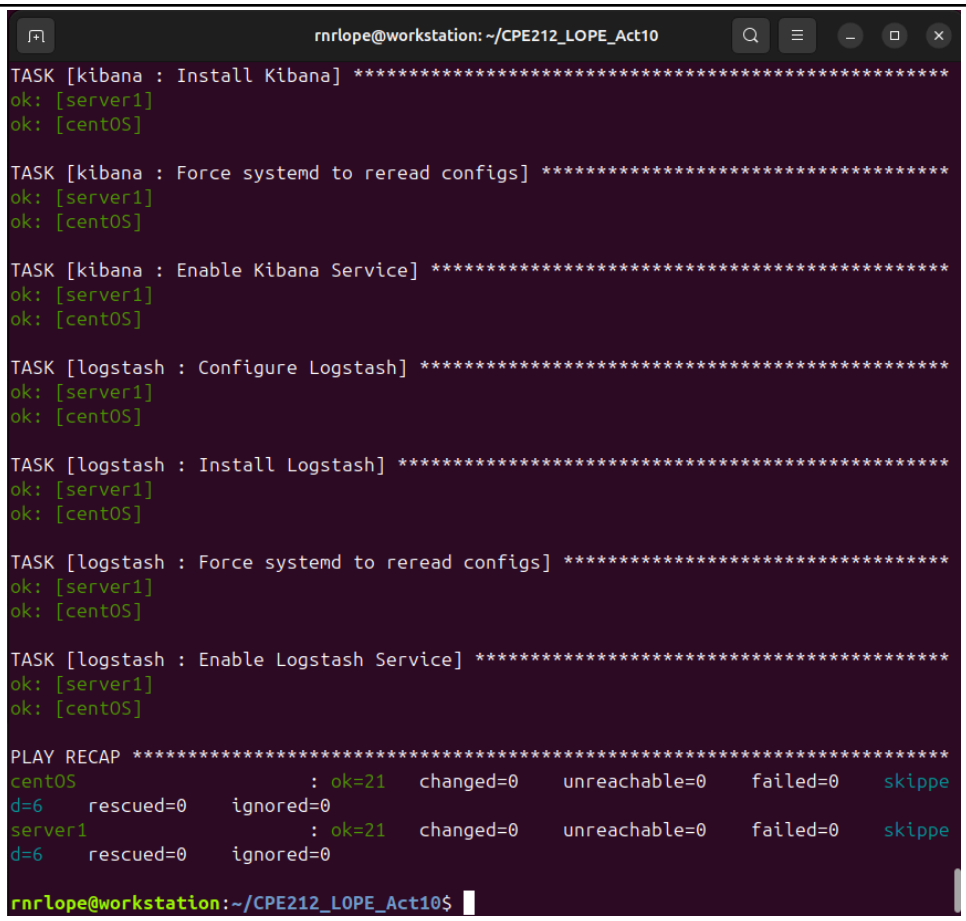


Figure 10.13: Play Recap of executing the playbook "elasticStack.yml".

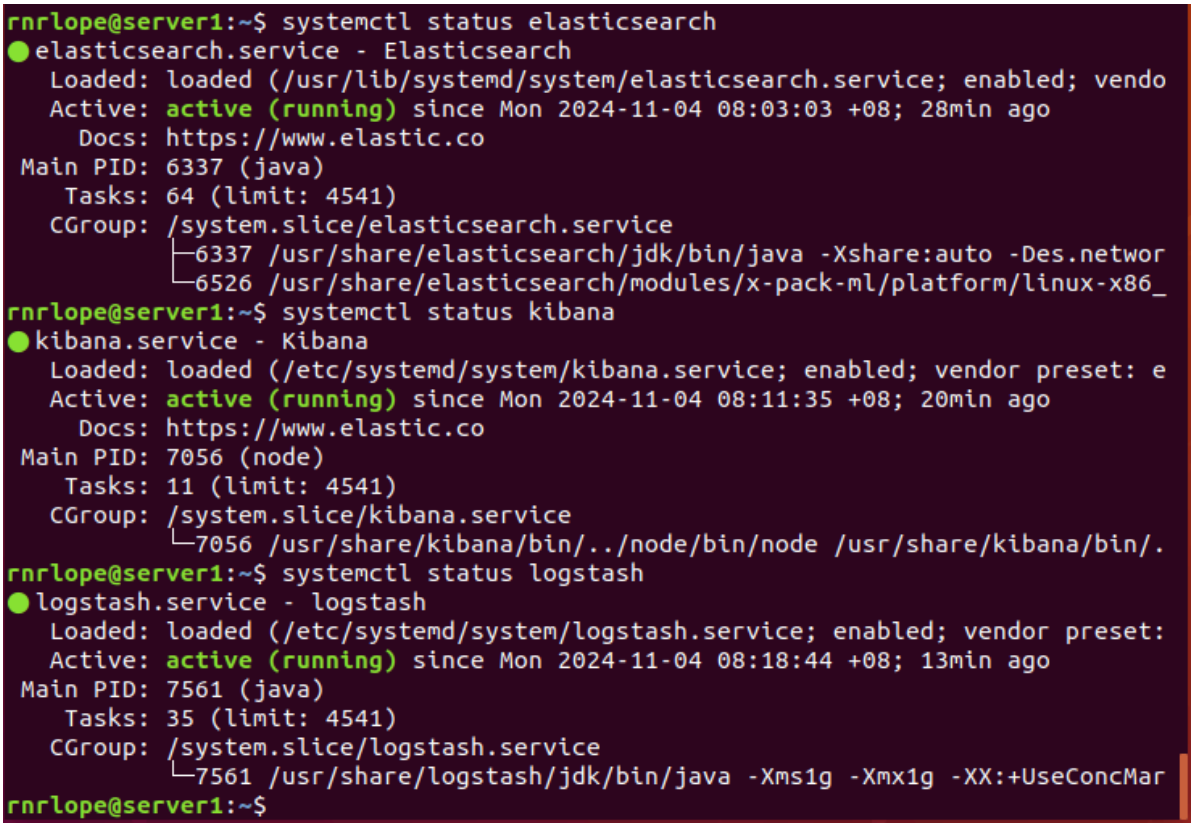


Figure 10.14 - 10.16: Verifying if elasticsearch, kibana, and logstash are installed and enabled on server1.

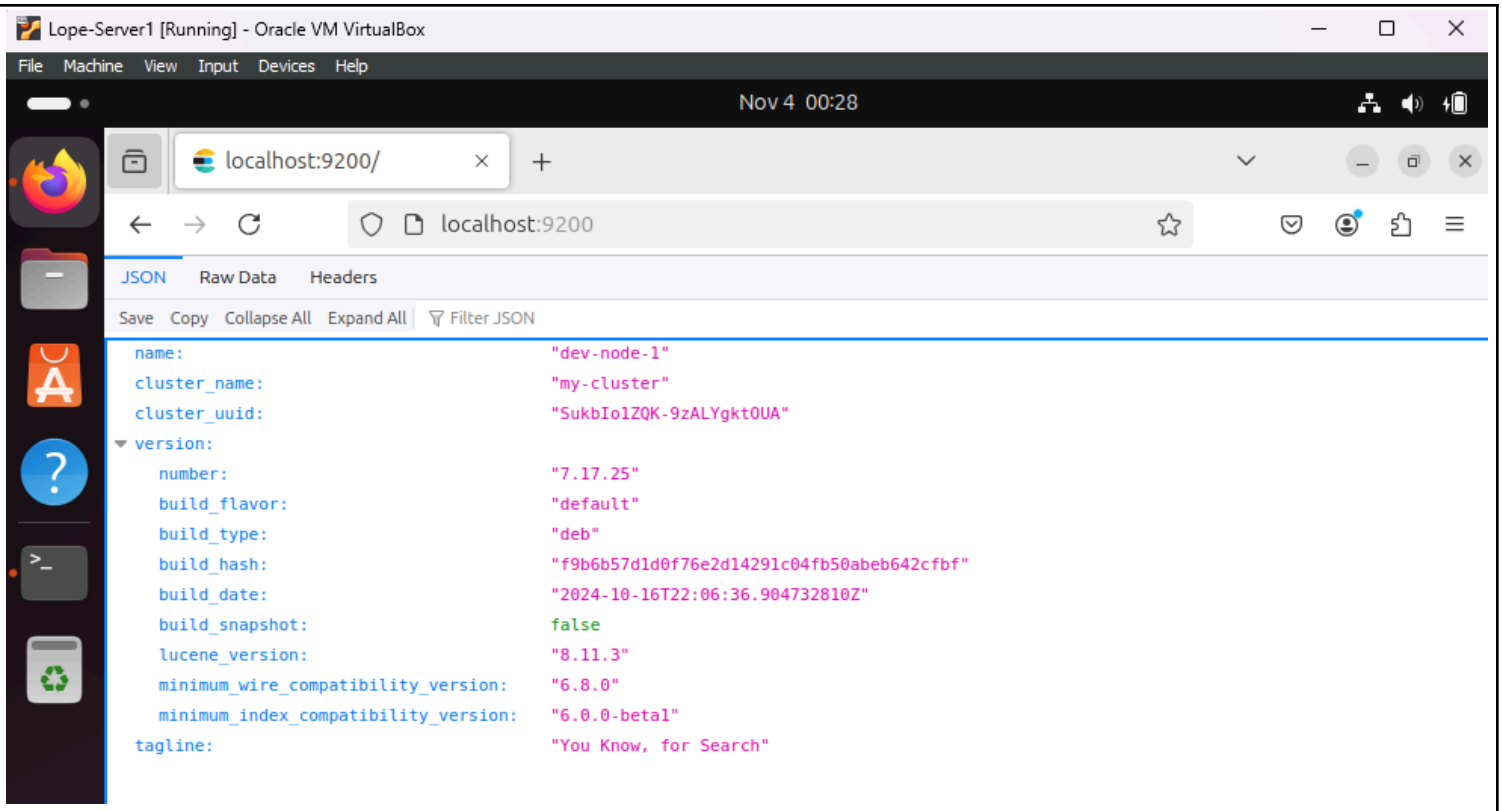
```
[rnrlope@localhost ~]$ systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2024-11-03 19:02:49 EST; 31min ago
     Docs: https://www.elastic.co
 Main PID: 5131 (java)
    Tasks: 66
   CGroup: /system.slice/elasticsearch.service
           ├─5131 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.net...
           └─5332 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x...

Nov 03 19:02:21 localhost.localdomain systemd[1]: Starting Elasticsearch...
Nov 03 19:02:28 localhost.localdomain systemd-entrypoint[5131]: Nov 03, 2024 ...
Nov 03 19:02:28 localhost.localdomain systemd-entrypoint[5131]: WARNING: COMP...
Nov 03 19:02:49 localhost.localdomain systemd[1]: Started Elasticsearch.
Hint: Some lines were ellipsized, use -l to show in full.
[rnrlope@localhost ~]$ systemctl status kibana
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2024-11-03 19:11:33 EST; 22min ago
     Docs: https://www.elastic.co
 Main PID: 6071 (node)
    Tasks: 11
   CGroup: /system.slice/kibana.service
           └─6071 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bi...

Nov 03 19:11:33 localhost.localdomain systemd[1]: Started Kibana.
Nov 03 19:11:33 localhost.localdomain kibana[6071]: Kibana is currently runni...
Hint: Some lines were ellipsized, use -l to show in full.
[rnrlope@localhost ~]$ systemctl status logstash
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2024-11-03 19:18:59 EST; 15min ago
 Main PID: 6850 (java)
    Tasks: 36
   CGroup: /system.slice/logstash.service
           └─6850 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseConc...

Nov 03 19:22:06 localhost.localdomain logstash[6850]: [2024-11-03T19:22:06,47...
Nov 03 19:22:06 localhost.localdomain logstash[6850]: [2024-11-03T19:22:06,53...
Nov 03 19:22:06 localhost.localdomain logstash[6850]: [2024-11-03T19:22:06,76...
Nov 03 19:22:08 localhost.localdomain logstash[6850]: [2024-11-03T19:22:08,51...
Nov 03 19:22:08 localhost.localdomain logstash[6850]: [2024-11-03T19:22:08,55...
Nov 03 19:22:09 localhost.localdomain logstash[6850]: [2024-11-03T19:22:09,56...
```

Figure 10.17 - 10.19: Verifying if elasticsearch, kibana, and logstash are installed and enabled on centOS.



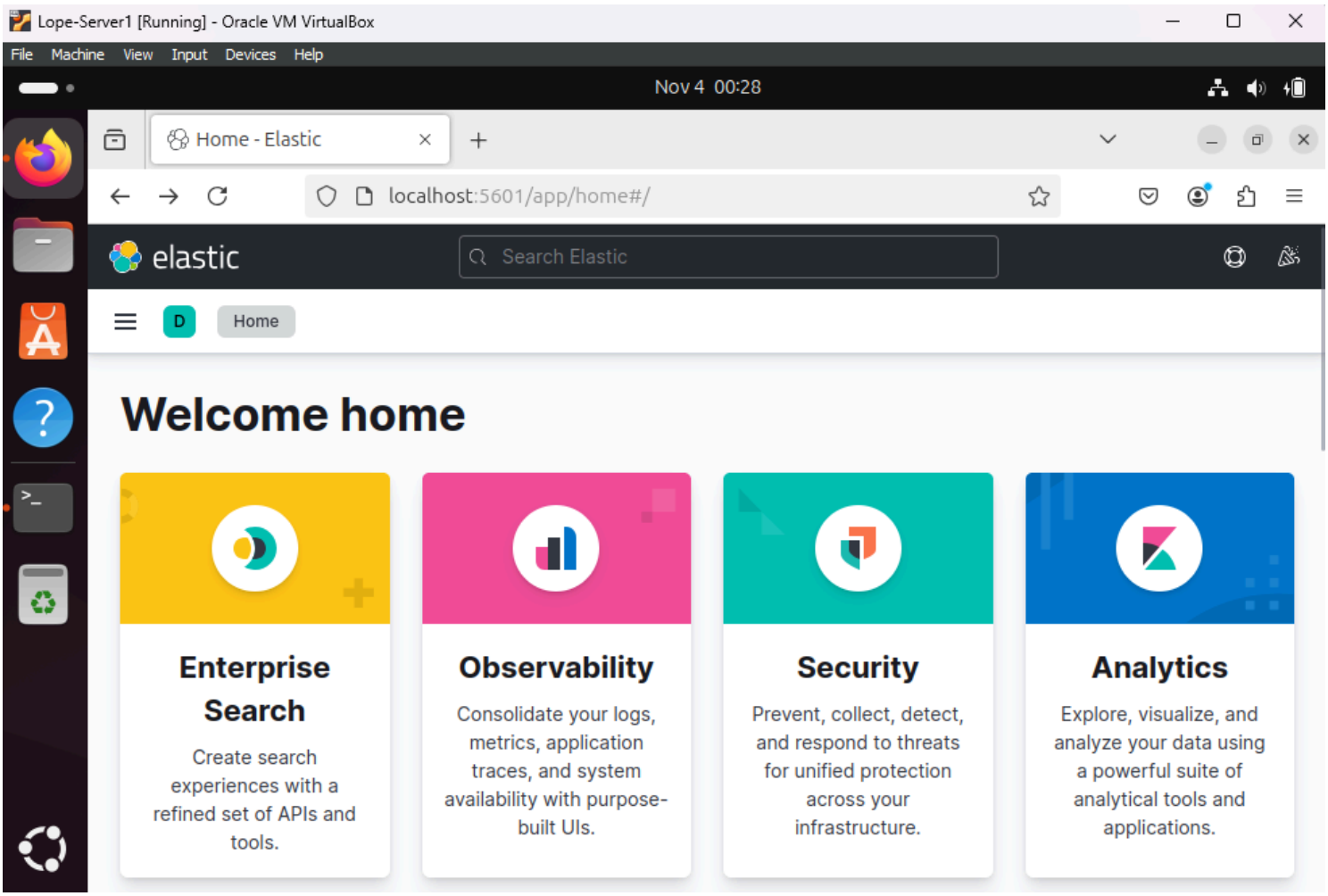


Figure 10.20 & 10.21: Verifying if elasticsearch and kibana are accessible using the web browser on server1.

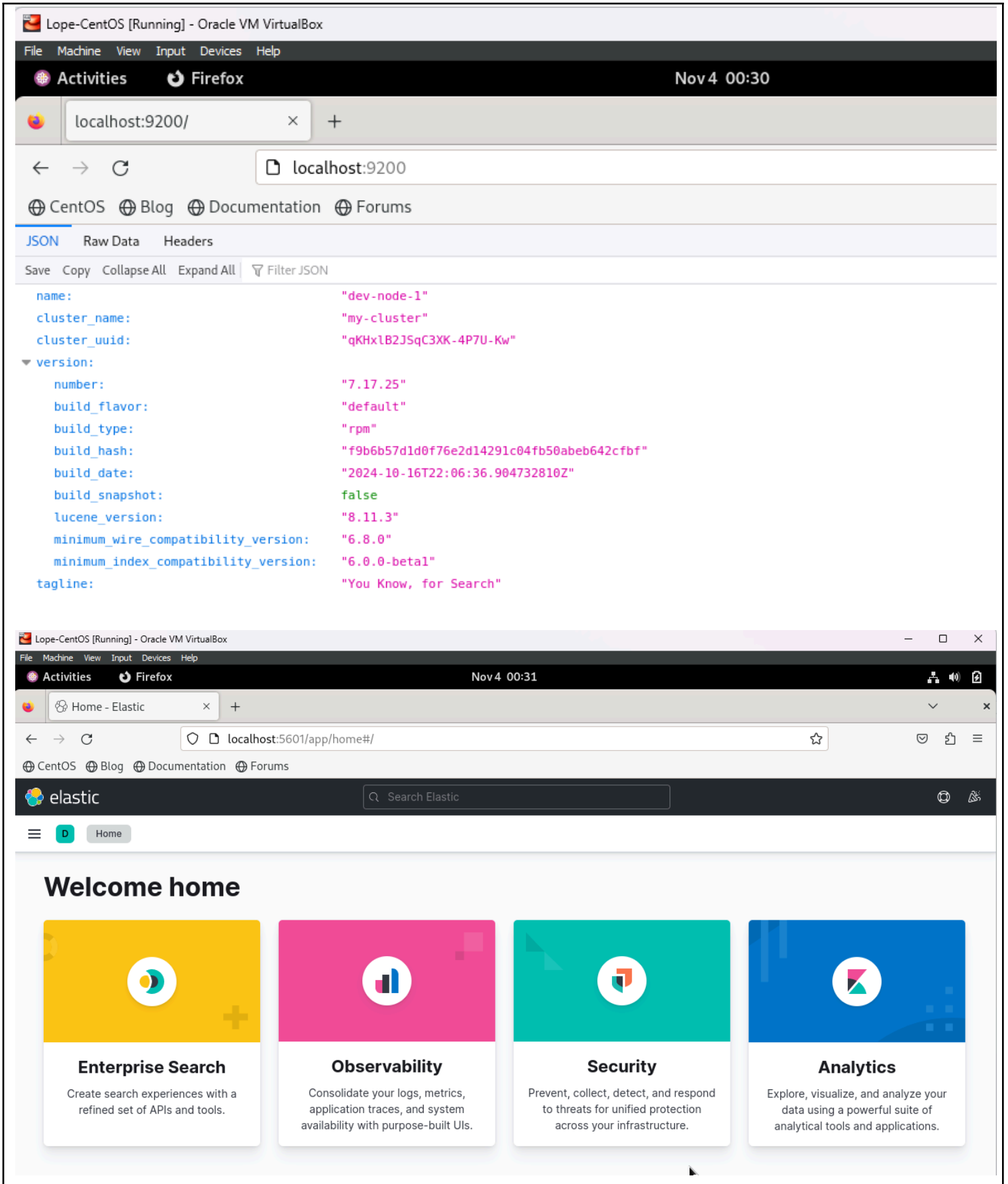


Figure 10.22 & 10.23: Verifying if elasticsearch and kibana are accessible using the web browser on centOS.

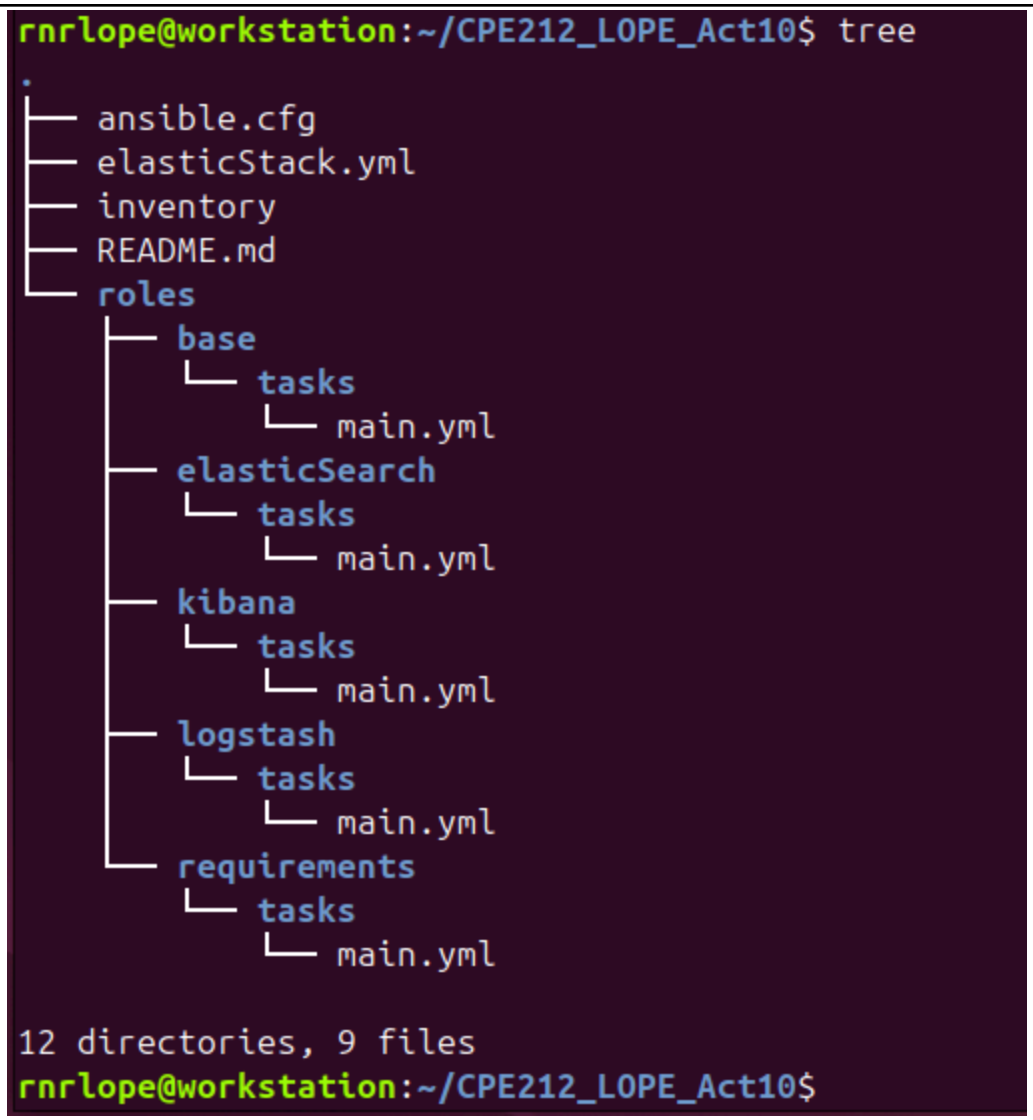
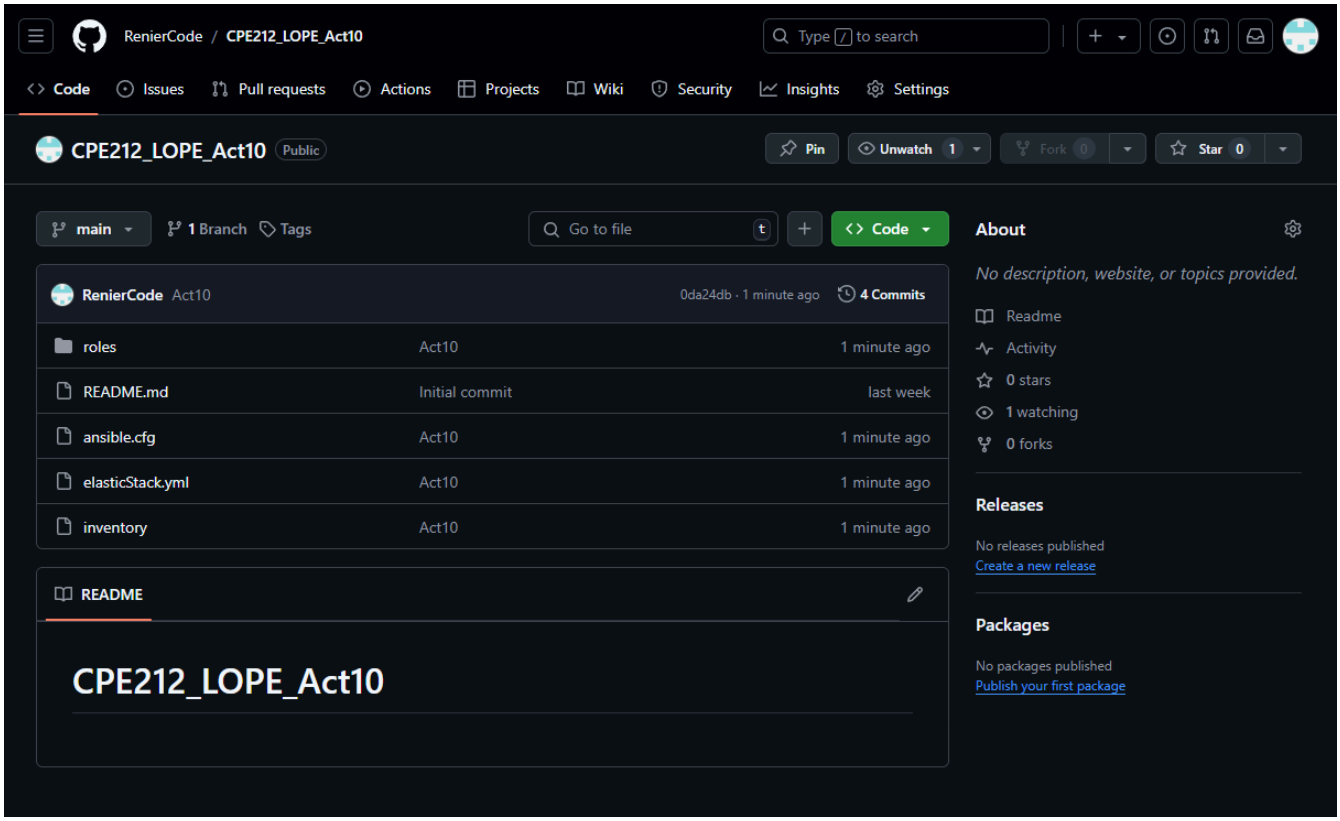


Figure 10.24: Final Contents of the repository.

GIT PUSH:

```
rnrlope@workstation:~/CPE212_LOPE_Act10$ git add --all
rnrlope@workstation:~/CPE212_LOPE_Act10$ git commit -m "Act10"
[main 0da24db] Act10
 10 files changed, 176 insertions(+), 68 deletions(-)
 delete mode 100644 .inventory.swp
 rename elasticstack.yml => elasticStack.yml (64%)
 create mode 100644 roles/elasticSearch/tasks/main.yml
 delete mode 100644 roles/installs/tasks/.main.yml.swp
 delete mode 100644 roles/installs/tasks/main.yml
 create mode 100644 roles/kibana/tasks/main.yml
 create mode 100644 roles/logstash/tasks/main.yml
 create mode 100644 roles/requirements/tasks/main.yml
rnrlope@workstation:~/CPE212_LOPE_Act10$ git push origin main
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 2 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (18/18), 2.78 KiB | 355.00 KiB/s, done.
Total 18 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:RenierCode/CPE212_LOPE_Act10
   2c3d17a..0da24db  main -> main
rnrlope@workstation:~/CPE212_LOPE_Act10$
```



GITHUB LINK:

https://github.com/RenierCode/CPE212_LOPE_Act10.git

Reflections:

Answer the following:

1. What are the benefits of having a log monitoring tool?
    - There are several benefits of having a log monitoring tool in Automating server management such as identifying and blocking unauthorized access attempts ensuring only authorized users can access key systems and data, can be adapted to cloud environments, detecting insider threats such as unusual patterns or behaviours, preventing data leak by identifying instances of data leakage ensuring that sensitive information is well protected, detecting possible threats and vulnerabilities in real-time allowing for a quick response, and many more.

Conclusions:

- In this activity I manage to create and demonstrate a workflow that will set up ElasticStack in both Ubuntu and CentOS server, while utilizing ansible-playbook and applying the concept of creating roles. So to set up a working ElasticStack, I need to install and configure three programs which are elasticsearch, kibana, and logstash.
- First, I created a main playbook named “elasticStack.yml” inside the repository “CPE212_LOPE_Act10” that will run all the tasks inside the desired roles.
- Secondly, I created a new directory named “roles”, then inside created new directories named “base”, “requirements”, “elasticSearch”, “kibana” and “logstash”.
- Thirdly, inside of the new directories I created a new directory named “tasks”.
- Lastly, I created playbooks inside of “tasks”, containing the separated plays based on the names of various roles.
- Overall, I manage to design and create a workflow that setups ElasticStack in both Ubuntu and CentOS server, while utilizing ansible-playbook and applying the concept of creating roles, thereby increasing my knowledge about playbooks.