Name: Gabriel Soriano	Date Performed: October 25, 2022
Course/Section: CPE 232-CPE31S23	Date Submitted: October 26, 2022
Instructor: Engr.Taylar	Semester and SY: 1st Sem-SY 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.

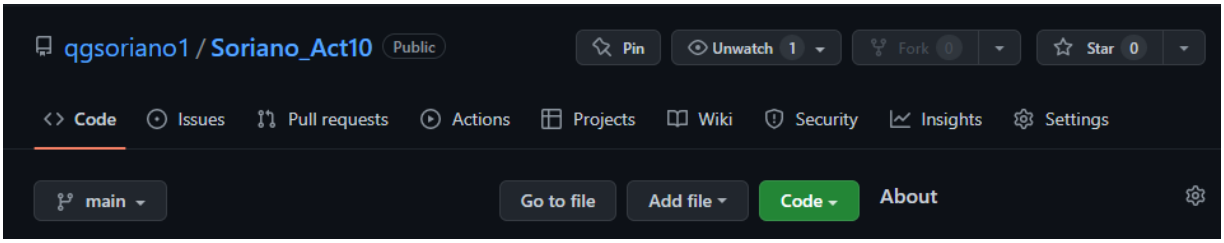
It is based on Elasticsearch, MongoDB, and Scala. Graylog has a main server, which receives data from its clients installed on different servers, and a web interface, which visualizes the data and allows to work with logs aggregated by the main server.

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

SCREENSHOT:



- **This shows the creation of the repository for this specific activity, the repository is named Soriano_Act10**

```
qgsoriano1@cloudshell:~/Soriano_Act10$ tree
.
├── ansible.cfg
├── inventory
├── README.md
├── roles
│   ├── elasticsearch
│   │   └── tasks
│   │       └── main.yml
│   ├── filebeat
│   │   └── tasks
│   │       └── main.yml
│   ├── filebeat-logzio
│   │   ├── defaults
│   │   │   └── main.yml
│   │   ├── tasks
│   │   │   └── main.yml
│   │   └── templates
│   │       └── filebeat.yml.j2
│   ├── install
│   │   └── tasks
│   │       └── main.yml
│   ├── java
│   │   └── tasks
│   │       └── main.yml
│   ├── kibana
│   │   └── tasks
│   │       └── main.yml
│   └── metricbeat
│       └── tasks
│           └── main.yml
└── site.yml

17 directories, 13 files
```

- **This shows the tree directory path of the repository/main folder named Soriano_Act10**

```
qgsoriano1@cloudshell:~/Soriano_Act10$ cat ansible.cfg
[defaults]

inventory = inventory
host_key_checking = False

deprecation_warnings = False

remote_user = qgsoriano1
private_key_file = ~/.ssh/
qgsoriano1@cloudshell:~/Soriano_Act10$ cat inventory
[remote_servers]
192.168.56.105
192.168.56.118
qgsoriano1@cloudshell:~/Soriano_Act10$
```

- **This shows the content of the ansible.cfg file for the ansible configuration, and the content of the inventory file, which contains the ip addresses of the ubuntu and centOS desktops.**

```
qgsoriano1@cloudshell:~/Soriano_Act10$ ls
ansible.cfg  inventory  README.md  roles  site.yml
qgsoriano1@cloudshell:~/Soriano_Act10$ cat site.yml
---
- hosts: all
  remote_user: ubuntu
  become: yes
  become_user: root
  roles:
  - { role: java }
  - { role: elasticsearch }
  - { role: kibana }
  - { role: metricbeat }
qgsoriano1@cloudshell:~/Soriano_Act10$
```

- **This shows the content of the site.yml, which is the main playbook file, this contains the roles specified for the requirements to be installed, which is java, elasticsearch, kibana, and metricbeat.**

```
qgsoriano1@cloudshell:~/Soriano_Act10$ cd roles
qgsoriano1@cloudshell:~/Soriano_Act10/roles$ ls
elasticsearch  filebeat  filebeat-logzio  install  java  kibana  metricbeat
qgsoriano1@cloudshell:~/Soriano_Act10/roles$
```

- **This shows the directories created inside the roles directory.**

```
qgsoriano1@cloudshell:~/Soriano_Act10/roles$ tree
.
├── elasticsearch
│   └── tasks
│       └── main.yml
├── filebeat
│   └── tasks
│       └── main.yml
├── filebeat-logzio
│   ├── defaults
│   │   └── main.yml
│   ├── tasks
│   │   └── main.yml
│   └── templates
│       └── filebeat.yml.j2
├── install
│   └── tasks
│       └── main.yml
├── java
│   └── tasks
│       └── main.yml
├── kibana
│   └── tasks
│       └── main.yml
└── metricbeat
    └── tasks
        └── main.yml

16 directories, 9 files
qgsoriano1@cloudshell:~/Soriano_Act10/roles$
```

- **This shows the tree directory path of the roles directory, which contains the created directories which are the following: elasticsearch, filebeat, filebeat_logzio, install, java, kibana, and metricbeat directories. Each of these sub directories contains their own main.yml playbook file.**

```
---
- name: Add elasticsearch apt key
  apt_key:
   url: "https://packages.elastic.co/GPG-KEY-elasticsearch"
   state: present

- name: Addind elasticsearch repository
  apt_repository:
   repo: deb https://artifacts.elastic.co/packages/5.x/apt stable main
   state: present

- name: Install elasticsearch
  apt:
   name: elasticsearch
   update_cache: yes

- name: UPdating the config file to allow outside access
  lineinfile:
   destfile: /etx/elasticsearch/elasticsearch.yml
   regexp: 'network.host:'
   line: 'network.host: 0.0.0.0'

- name: Updating the port in config file
  lineinfile:
   destfile: /etc/elasticsearch/elasticsearch.yml
   regexp: 'http.port:'
   line: 'http.port: 9200'

- name: Starting elasticsearch
  service:
   name: elasticsearch
   state: started
```

- **This shows the content of the main.yml playbook file of the tasks directory under the elasticsearch directory.**

```
qgsoriano1@cloudshell:~/Soriano_Act10/roles$ cd filebeat
qgsoriano1@cloudshell:~/Soriano_Act10/roles/filebeat$ ls
tasks
qgsoriano1@cloudshell:~/Soriano_Act10/roles/filebeat$ cd tasks
qgsoriano1@cloudshell:~/Soriano_Act10/roles/filebeat/tasks$ ls
main.yml
qgsoriano1@cloudshell:~/Soriano_Act10/roles/filebeat/tasks$ cat main.yml
---

- name: Install filebeat with apt
  apt:
   name: filebeat
   update_cache: yes

- name: Starting filebeat
  service:
   name: filebeat
   state: started
qgsoriano1@cloudshell:~/Soriano_Act10/roles/filebeat/tasks$
```

- **This shows the content of the main.yml playbook file of the tasks directory under the filebeat directory.**

```
qgsoriano1@cloudshell:~/Soriano_Act10/roles$ ls
elasticsearch  filebeat  filebeat-logzio  install  java  kibana  metricbeat
qgsoriano1@cloudshell:~/Soriano_Act10/roles$ cd filebeat-logzio
qgsoriano1@cloudshell:~/Soriano_Act10/roles/filebeat-logzio$ ls
defaults  tasks  templates
qgsoriano1@cloudshell:~/Soriano_Act10/roles/filebeat-logzio$ tree
.
├── defaults
│   └── main.yml
├── tasks
│   └── main.yml
└── templates
    └── filebeat.yml.j2

3 directories, 3 files
qgsoriano1@cloudshell:~/Soriano_Act10/roles/filebeat-logzio$
```

- **This shows the sub directories inside the filebeat-logzio directory. It contains the defaults, tasks, and templates directory.**

```
---

filebeat_create-config: true

filebeat_prospectors:
  - input_type: log
    paths:
      - "/var/log/*.log"
    fields:
     logzio_codec: plain
     token: token
    files_under_root: true
    ignore_older: 3h

filebeat_output_elasticsearch_enabled: false
filebeat_output_elasticsearch_hosts:
  - "localhost:9200"

filebeat_output_logstash_enabled: true
filebeat_output_logstash_hosts:
  - "listener.logz.io:5015"

filebeat_enable_logging: false
filebeat_log_level: warning
filebeat_log_dir: /var/log/mybeat
filebeat_log_filename: mybeat.log

filebeat_ssl_dir: /etc/pki/tls/certs
filebeat_ssl_certificate_file: "etc/pki/tls/certs/COMODORSADomainValidationSecureServerCA.crt"
filebeat_ssl_key_file: ""
filebeat_ssl_insecure: "false"
qgsoriano1@cloudshell:~/Soriano_Act10/roles/filebeat-logzio/defaults$
```

- **This shows the content of the main.yml inside the defaults directory under the filebeat-logzio directory.**

```
---

- name: Download certificate
  shell: wget https://raw.githubusercontent.com?logzio/public-certificate/maste/COMODORSADomainValidationSecureServerCA.crt

- name: Make new directory for cert
  shell: mkdir -p /etc/pkie/tls/certs

- name: Move SSL certification to new folder
  shell: cp COMODORSADomainValidationSecureServerCA.crt /etc/pkie/tls/certs/

- name: Install filebeat with apt
  apt:
   name: filebeat
   update_cache: yes

- name: Replace default filebeat.yml configurations
  template:
    src: filebeat.yml.j2
    dest: /etc/filebeat/filebeat.yml

- name: Starting filebeat
  service:
   name: filebeat
   state: started
qgsoriano1@cloudshell:~/Soriano_Act10/roles/filebeat-logzio/tasks$
```

- **This shows the main.yml playbook content of the tasks directory under the filebeat-logzio directory.**

```
qgsoriano1@cloudshell:~/Soriano_Act10/roles/filebeat-logzio/templates$ cat filebeat.yml.j2
filebeat:
  prospectors:
    {{ filebeat_prospectrs | to_json }}

output:
{% if filebeat_output_elasticsearch_enabled %}
  elasticsearch:
    hosts: {{ filebeat_output_elasticsearch_hosts | to_json }}

{% if filebeat_ssl_certificate_file and filebeat_ssl_key_file %}
    tls:
      certificate: "{{ filebeat_ssl_dir }}/{{ filebeat_ssl_certificate_file | basename }}"

      certificate_key: "{{ filebeat_ssl_dir }}/{{ filebeat_ssl_key_file | basename }}"

      insecure: {{ filebeat_ssl_insecure }}

{% endif %}
{% endif %}

{% if filebeat_output_logstash_enabled %}
  logstash:
    hosts: {{ filebeat_output_logstash_hosts | to_json }}

{%if filebeat_ssl_certificate and filebeat_ssl_key_file %}
   tls:
     certificate: "{{ filebear_ssl_dir }}/{{ filebeat_ssl_certificate_file | basename }}"

     certificate_key: "{{ filebeat_ssl_dir }}/{{ filebeat_ssl_key_file | basename }}"

     insecure: {{ filebeat_ssl_insecure }}
```

```
{%if filebeat_ssl_certificate and filebeat_ssl_key_file %}
   tls:
     certificate: "{{ filebear_ssl_dir }}/{{ filebeat_ssl_certificate_file | basename }}"

     certificate_key: "{{ filebeat_ssl_dir }}/{{ filebeat_ssl_key_file | basename }}"

     insecure: {{ filebeat_ssl_insecure }}

{% endif %}

{%if filebeat_enable_logging %}
logging:
  level: {{ filebeat_log_level }}

     certificate_key: "{{ filebeat_ssl_dir }}/{{ filebeat_ssl_key_file | basename }}"

     insecure: {{ filebeat_ssl_insecure }}

{% endif %}

{% if filebeat_enable_logging %}
logging:
  level: {{ filebeat_log_level }}

  to_files: true

  to_syslog: false

  files:
    path: {{ filebeat_log_dir }}
    name: {{ filebeat_log_filename }}
    keepfiles: 7
{% endif %}
{% endif %}
qgsoriano1@cloudshell:~/Soriano_Act10/roles/filebeat-logzio/templates$
```

- **This shows the content of the playbook file named filebeat.yml.j2**

```
qgsoriano1@cloudshell:~/Soriano_Act10/roles/java/tasks$ cat main.yml
---

- name: Add the java ppa repo
  apt_repository:
   repo: ppa:webupd8team/java

- name: Automatically accept the oravle license
  shell: echo debconf shared/accepted-oracle-license-v1-1 true | sudo debconf-set-selections

- name: Install java 8
  apt:
   name: orcale-java8-installer
   state: present
   update_cache: yes
qgsoriano1@cloudshell:~/Soriano_Act10/roles/java/tasks$
```

- **This shows the content of the main.yml playbook file of the tasks directory under the java directory.**

```
qgsoriano1@cloudshell:~/Soriano_Act10/roles/kibana$ ls
tasks
qgsoriano1@cloudshell:~/Soriano_Act10/roles/kibana$ cd tasks
qgsoriano1@cloudshell:~/Soriano_Act10/roles/kibana/tasks$ ls
main.yml
qgsoriano1@cloudshell:~/Soriano_Act10/roles/kibana/tasks$ cat main.yml
---

- name: Install kibana with apt
  apt:
   name: kibana
   update_cache: yes

- name: Updating the config file to allow outside access
  lineinfile:
   destfile: /etc/kibana/kibana.yml
   regexp: 'server.host:'
   line: 'server.host: 0.0.0.0'

- name: Defining server port
  lineinfile:
   destfile: /etc/kibana/kibana.yml
   regexp: 'server.post:'
   line: 'server.port: 5601'

- name: Defining elasticsearch url
  lineinfile:
   destfile: /etc/kibana/kibana.yml
   regexp: 'elasticsearch.urm:'
   line: 'elasticsearch.url: "http://localhost:9200"'

- name: Starting kibana
  service:
   name: kibana
   state: started
qgsoriano1@cloudshell:~/Soriano_Act10/roles/kibana/tasks$
```

- **This shows the content of the main.yml playbook file of the tasks directory under the kibana directory.**

```
qgsoriano1@cloudshell:~/Soriano_Act10/roles$ ls
elasticsearch  filebeat  filebeat-logzio  install  java  kibana  metricbeat
qgsoriano1@cloudshell:~/Soriano_Act10/roles$ cd metricbeat
qgsoriano1@cloudshell:~/Soriano_Act10/roles/metricbeat$ ls
tasks
qgsoriano1@cloudshell:~/Soriano_Act10/roles/metricbeat$ cd tasks
qgsoriano1@cloudshell:~/Soriano_Act10/roles/metricbeat/tasks$ ls
main.yml
qgsoriano1@cloudshell:~/Soriano_Act10/roles/metricbeat/tasks$ cat main.yml
---

- name: Install metricbeat with apt
  apt:
   name: metricbeat
   update_cache: yes

- name: Starting metricbeat
  service:
   name: metricbeat
   state: started
qgsoriano1@cloudshell:~/Soriano_Act10/roles/metricbeat/tasks$
```

- **This shows the content of the main.yml playbook file of the tasks directory under the metricbeat directory.**

1. Create a playbook that:
    a. Install and configure Elastic Stack in separate hosts (Elastic Search, Kibana, Logstash)

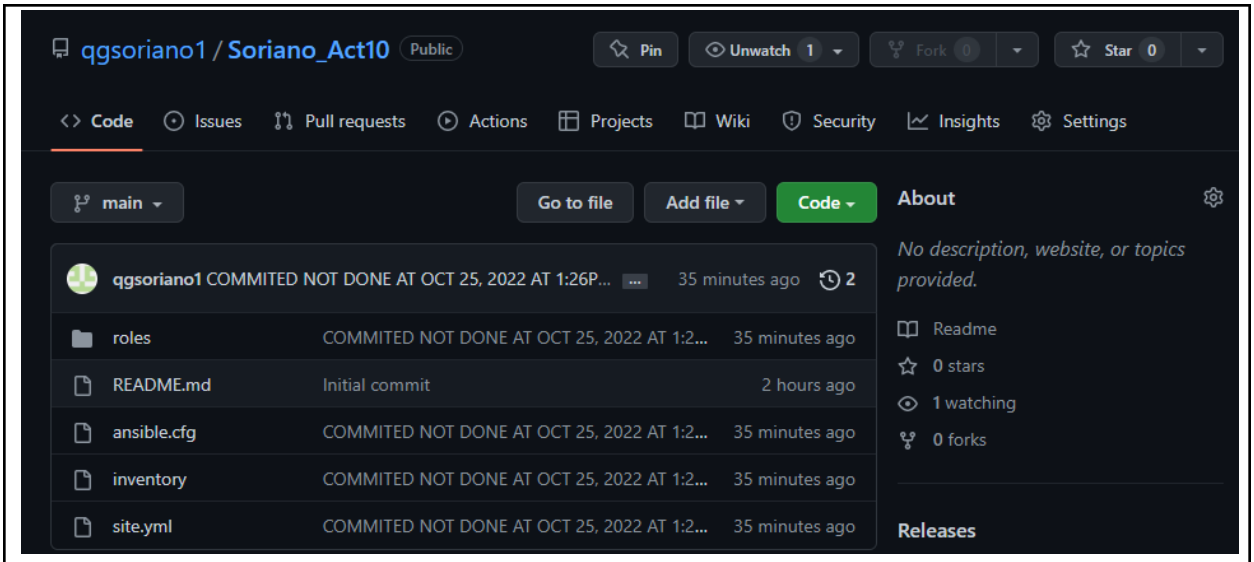
- **Steps are pasted above**

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.)

- **Steps are pasted above**

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.

SCREENSHOT:



- **This shows the successful git adding, committing, and pushing of the performed and created files during this activity. All are uploaded and saved at the github repository name Soriano_Act10.**

4. **Output** (screenshots and explanations)

    - **Steps and outputs are pasted above**

Reflections:

Answer the following:

1. What are the benefits of having log monitoring tool?
    - **Tech experts can more easily identify problem areas, evaluate the health of the application, enhance troubleshooting, and optimize root cause analysis of application performance errors by collecting, analyzing, and monitoring these logs. Increased Business Efficiency. Log analysis tools offer the functionality required to identify critical system errors or trends and address them quickly and effectively because so many departments rely on IT resources to carry out their business-critical tasks and responsibilities. greater, more thorough security. Since the cost of**

cyberattacks has been steadily increasing in recent years, it's more crucial than ever to put in place and keep up strong security procedures. Event log files are crucial for both preventative security measures and forensic investigations when they are deemed necessary. improved allocation of resources and provisioning businesses that use computers and networks need specific resources to operate effectively. These include network bandwidth and hardware drives for storing data, which allow end users to work simultaneously.

Conclusions:

While performing this activity, there are many errors encountered. Mostly of the errors are inside the .yml playbook files, specifically the alignments of the code's syntax. It's been a hard time because some of the .yml playbook files are very long. But all of these similar errors are resolved. One of the errors encountered is the right paths/directory location of the roles. It gave me a bit of a headache because there are a lot of directories under the main directory, after that, there are even sub directories under the directories of the main directory. Headache, right? Well I got over it. Successful encoding of the command is done to install the requirements to their specific OS, which is ubuntu and centOS desktops. After this, there are genuine and honest unachieved requirements for this activity, I am able to code the playbook files name prome.yml and the main.yml, I am pretty sure that those codes will do its job and install the prometheus to the ubuntu and centOS desktops. What I was not able to perform is the accessing of the ubuntu and centOS desktops. This is because I performed this activity at home and my laptop is lacking the ability to run workstations at the virtual box, this is due to lack of memory. I tried to resolve this problem but sadly my laptop's memory is not upgradeable anymore. I will surely reperform this activity back at the cisco lab.