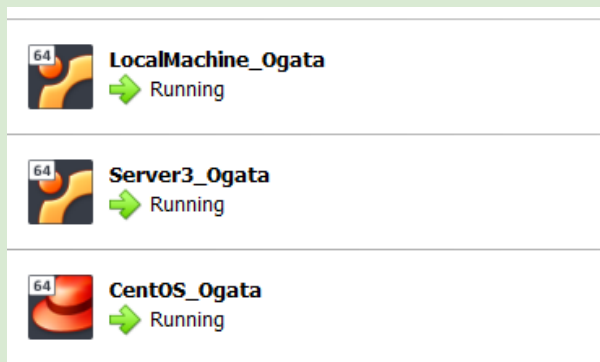


Name: Kazuki A. Ogata	Date Performed: October 27, 2023
Course/Section: CPE 232 - CPE31S5	Date Submitted: October 28, 2023
Instructor: Engr. Roman Richard	Semester and SY: 1st semester S.Y 2023-2024
Activity 10: Install, Configure, and Manage Log Monitoring tools	
1. Objectives	
Create and design a workflow that installs, configure and manage enterprise log monitoring tools using Ansible as an Infrastructure as Code (IaC) tool.	
2. Discussion	
<p>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.</p> <p>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.</p> <p>To qualify for inclusion in the Log Monitoring category, a product must:</p> <ul style="list-style-type: none"> • 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	
<p>ELK suite stands for Elasticsearch, Kibana, Beats, and Logstash (also known as the ELK Stack). Source: https://www.elastic.co/elastic-stack</p> <p>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.</p>	
GrayLog	
<p>Graylog is a powerful platform that allows for easy log management of both structured and unstructured data along with debugging applications.</p> <p>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 it to work with logs aggregated by the main server.</p> <p>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.</p> <p>Source: https://www.graylog.org/products/open-source</p>	

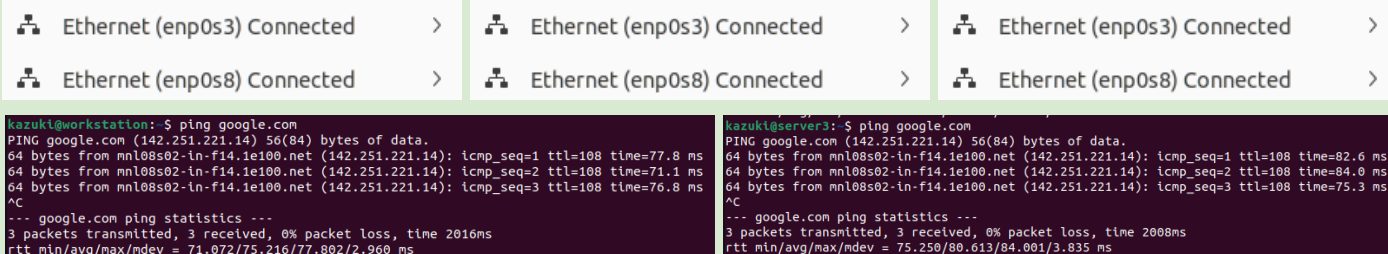
3. Tasks

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)



Before starting this activity, we will be needing 3 virtual machines. 1 control node and 2 managed nodes. I used “LocalMachine_Ogata” as my control node (Ubuntu) and Ubuntu server 3 and CentOS as my managed nodes.



```
kazuki@workstation:~$ ping google.com
PING google.com (142.251.221.14) 56(84) bytes of data.
64 bytes from mnl08s02-in-f14.1e100.net (142.251.221.14): icmp_seq=1 ttl=108 time=77.8 ms
64 bytes from mnl08s02-in-f14.1e100.net (142.251.221.14): icmp_seq=2 ttl=108 time=71.1 ms
64 bytes from mnl08s02-in-f14.1e100.net (142.251.221.14): icmp_seq=3 ttl=108 time=76.8 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2016ms
rtt min/avg/max/mdev = 71.072/75.216/77.802/2.960 ms
```

```
kazuki@server3:~$ ping google.com
PING google.com (142.251.221.14) 56(84) bytes of data.
64 bytes from mnl08s02-in-f14.1e100.net (142.251.221.14): icmp_seq=1 ttl=108 time=82.6 ms
64 bytes from mnl08s02-in-f14.1e100.net (142.251.221.14): icmp_seq=2 ttl=108 time=84.0 ms
64 bytes from mnl08s02-in-f14.1e100.net (142.251.221.14): icmp_seq=3 ttl=108 time=75.3 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2008ms
rtt min/avg/max/mdev = 75.250/80.613/84.001/3.835 ms
```

```
[kazuki@centos ~]$ ping google.com
PING google.com (142.251.221.14) 56(84) bytes of data.
64 bytes from mnl08s02-in-f14.1e100.net (142.251.221.14): icmp_seq=1 ttl=108 time=73.1 ms
64 bytes from mnl08s02-in-f14.1e100.net (142.251.221.14): icmp_seq=2 ttl=108 time=96.6 ms
64 bytes from mnl08s02-in-f14.1e100.net (142.251.221.14): icmp_seq=3 ttl=108 time=93.7 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 73.124/87.839/96.678/10.475 ms
```

Check the internet connection of virtual machines. I learned my lesson from my previous activities, I started doing the activity without checking the internet connection. I used the command “ping google.com” to check the internet connection and manually checked the settings if the Ethernet enp0s3 and enp0s8 were Connected.

```
kazuki@server3:~$ hostname -I
10.0.2.15 192.168.56.129

[kazuki@centos ~]$ hostname -I
10.0.2.15 192.168.56.127 192.168.122.1 172.17.0.1
```

Check the IP Address of managed nodes. I used the command “hostname -I” to show that IP Addresses in that server. 192.168.56.125 is the IP Address of my Ubuntu Server 3. While 192.168.56.127 for my CentOS.

```
kazuki@workstation:~$ ping 192.168.56.129
PING 192.168.56.129 (192.168.56.129) 56(84) bytes of data.
64 bytes from 192.168.56.129: icmp_seq=1 ttl=64 time=1.06 ms
64 bytes from 192.168.56.129: icmp_seq=2 ttl=64 time=1.25 ms
64 bytes from 192.168.56.129: icmp_seq=3 ttl=64 time=0.961 ms
^C
--- 192.168.56.129 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.961/1.090/1.250/0.120 ms

kazuki@workstation:~$ ping 192.168.56.127
PING 192.168.56.127 (192.168.56.127) 56(84) bytes of data.
64 bytes from 192.168.56.127: icmp_seq=1 ttl=64 time=0.828 ms
64 bytes from 192.168.56.127: icmp_seq=2 ttl=64 time=1.12 ms
64 bytes from 192.168.56.127: icmp_seq=3 ttl=64 time=0.813 ms
^C
--- 192.168.56.127 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2042ms
rtt min/avg/max/mdev = 0.813/0.921/1.124/0.143 ms

kazuki@workstation:~$ ssh kazuki@192.168.56.129
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 6.2.0-35-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

kazuki@server3:~$
```

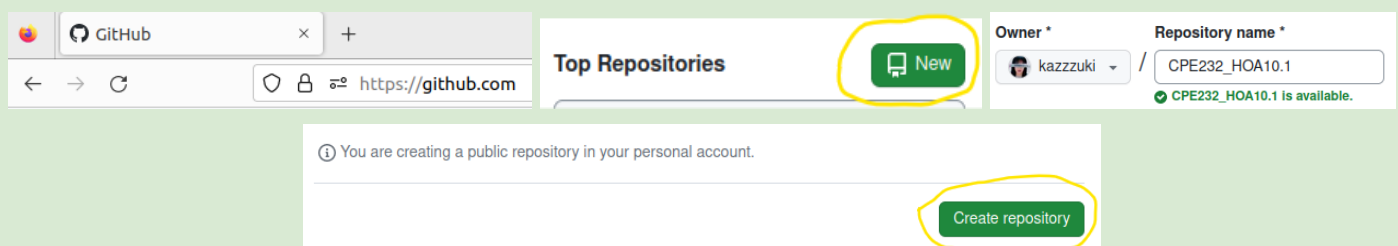
```
kazuki@workstation:~$ ssh kazuki@192.168.56.127
Last login: Fri Oct 27 21:19:14 2023 from 192.168.56.121
[kazuki@centos ~]$
```

After checking the IP address, we need to check if it has a connection to our workstation. I used the command “ping <ip add>” to check the normal connection of two machines. I used the command “ssh user@host” to check the ssh connections. When pinging, you should see a 0% packet loss, and there is no error. While in ssh connection, you should see that the hostname will change into the hostname of the hosts.

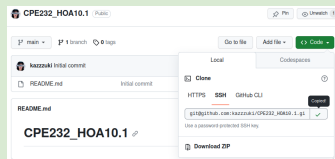
```
kazuki@workstation:~$ which git
/usr/bin/git

kazuki@workstation:~$ git --version
git version 2.34.1
```

Once we are done checking the connection of control nodes to managed nodes. We are now going to check if we have git installed in our workstation. We are checking this, since the activity requires creating a new repository on GitHub. I used the command “which git” to show any “git” related file/directory in my workstation. Then I used the command “git –version” just to make sure I have git installed in my workstation.



After installing or checking git installed in my workstation, I created a new repository named “CPE232_HOA10.1”. In creating a new repository, go to “www.github.com” create an account or sign into your account. After that, you will see a “new repository” there, just click it and then create a new repository. In this repository, I am going to save all my files that I will be creating.



```
kazuki@workstation:~$ git clone git@github.com:kazzzuki/CPE232_HOA10.1.git
Cloning into 'CPE232_HOA10.1'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

On the GitHub repository, I clicked the “<> Code” button and went to “SSH” and copied the link there. Then, I used the command “git clone” to clone my repository in my workstation.

```
kazuki@workstation:~$ ls
CPE232_HOA10.1  CPE232_HOA9.1  Documents  Ogata_PrelimExam  snap
CPE232_HOA6.1  CPE232_KAZUKI  Downloads  Pictures           Templates
CPE232_HOA8.1  Desktop        Music      Public            Videos
```

Verify the cloned GitHub repository. I used the command “ls” to check the cloned git.

```
kazuki@workstation:~$ cd CPE232_HOA10.1
kazuki@workstation:~/CPE232_HOA10.1$
```

Go inside the cloned repository. I used the command “cd” to go inside it.

```
kazuki@workstation:~/CPE232_HOA10.1$ sudo nano inventory
kazuki@workstation:~/CPE232_HOA10.1$ cat inventory

[Hoa10_Ubuntu]
192.168.56.129  ansible_connection=ssh

[Hoa10_CentOS]
192.168.56.127  ansible_connection=ssh
```

I started by creating an inventory file. I put all the remote servers that I will be needing in this activity. I put them both in a group so it will be easy to call them in roles and playbook.

```
kazuki@workstation:~/CPE232_HOA10.1$ sudo nano ansible.cfg
kazuki@workstation:~/CPE232_HOA10.1$ cat ansible.cfg

[defaults]

inventory = inventory
```

Next is creating an ansible.cfg file, I added a default configuration that the inventory file is the inventory.

```
kazuki@workstation:~/CPE232_HOA10.1$ ansible all --list-hosts
hosts (2):
  192.168.56.129
  192.168.56.127
kazuki@workstation:~/CPE232_HOA10.1$ ansible all -m ping
192.168.56.129 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
192.168.56.127 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
```

Now that I already have inventory and ansible.cfg files, I need to check if it has a connection. I used the “ansible all –list-hosts” command to check the IP addresses of the hosts in my workstation. I also used the command “ansible all -m ping” to verify the connection of the IP addresses.

```

kazuki@workstation:~/CPE232_HOA10.1$ mkdir -p roles/Hoa10_Ubuntu/tasks
kazuki@workstation:~/CPE232_HOA10.1$ mkdir -p roles/Hoa10_CentOS/tasks
kazuki@workstation:~/CPE232_HOA10.1$ ls
ansible.cfg  inventory  README.md  roles
kazuki@workstation:~/CPE232_HOA10.1$ cd roles
kazuki@workstation:~/CPE232_HOA10.1/roles$ ls
Hoa10_CentOS  Hoa10_Ubuntu
kazuki@workstation:~/CPE232_HOA10.1/roles$ cd Hoa10_Ubuntu
kazuki@workstation:~/CPE232_HOA10.1/roles/Hoa10_Ubuntu$ ls
tasks
kazuki@workstation:~/CPE232_HOA10.1/roles/Hoa10_Ubuntu$ cd ../Hoa10_CentOS
kazuki@workstation:~/CPE232_HOA10.1/roles/Hoa10_CentOS$ ls
tasks

```

I create a directory for 2 different roles for Ubuntu and CentOS. I used the command “mkdir” in creating the directory. I used the “cd” command to go to the directory. And lastly, I used the command “ls” to show the files inside the directory.

```

kazuki@workstation:~/CPE232_HOA10.1$ sudo nano /home/kazuki/CPE232_HOA10.1/roles/Hoa10_Ubuntu/tasks/main.yml
kazuki@workstation:~/CPE232_HOA10.1$ cat /home/kazuki/CPE232_HOA10.1/roles/Hoa10_Ubuntu/tasks/main.yml

- name: Add Elasticsearch APT repository key
  apt_key:
    url: https://packages.elastic.co/GPG-KEY-elasticsearch
    async: 3600
    poll: 0

- name: Add Elasticsearch APT repository
  apt_repository:
    repo: "deb http://packages.elastic.co/elasticsearch/1.7/debian stable main"
    async: 3600
    poll: 0

- name: Add Kibana APT repository
  apt_repository:
    repo: "deb https://artifacts.elastic.co/packages/7.x/apt stable main"
    async: 3600
    poll: 0

- name: Add Logstash APT repository
  apt_repository:
    repo: "deb http://packages.elasticsearch.org/logstash/1.5/debian stable main"
    async: 3600
    poll: 0

- name: Install Elasticsearch on Ubuntu
  apt:
    name: elasticsearch
    state: present
    async: 3600
    poll: 0

- name: Install Kibana on Ubuntu
  apt:
    name: kibana
    state: present
    async: 3600
    poll: 0

- name: Install Logstash on Ubuntu
  apt:
    name: logstash
    state: present
    async: 3600
    poll: 0

- name: Enable and Start Elasticsearch, Kibana, and Logstash
  systemd:
    name: "{{ item }}"
    enabled: yes
    state: started
  loop:
    - elasticsearch
    - kibana
    - logstash
  async: 3600
  poll: 0

```

In installing Elasticsearch, Kibana, Logstash on Ubuntu, firstly, I created an APT repository key for security measurement to prevent installing malicious software. Then I added tasks that add Elasticsearch, Kibana, Logstash APT repositories. After adding repositories, I then added tasks that install each package. Lastly, I added a task that enables and starts the Elasticsearch, Kibana, and Logstash services.

```

kazuki@workstation:~/CPE232_HOA10.1$ sudo nano /home/kazuki/CPE232_HOA10.1/roles/Hoa10_CentOS/tasks/main.yml
kazuki@workstation:~/CPE232_HOA10.1$ cat /home/kazuki/CPE232_HOA10.1/roles/Hoa10_CentOS/tasks/main.yml

- name: Add Elasticsearch 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
    async: 3600
    poll: 0

- name: Add Kibana YUM repository
  yum_repository:
    name: kibana
    description: Kibana Repository
    baseurl: https://artifacts.elastic.co/packages/7.x/yum
    gpgcheck: yes
    gpgkey: https://artifacts.elastic.co/GPG-KEY-elasticsearch
    async: 3600
    poll: 0

- name: Add Logstash YUM repository
  yum_repository:
    name: logstash
    description: Logstash Repository
    baseurl: https://artifacts.elastic.co/packages/7.x/yum
    gpgcheck: yes
    gpgkey: https://artifacts.elastic.co/GPG-KEY-elasticsearch
    async: 3600
    poll: 0

- name: Install Elasticsearch on CentOS
  dnf:
    name: elasticsearch
    use_backend: dnf
    state: present
  async: 3600
  poll: 0

- name: Install Kibana on CentOS
  dnf:
    name: kibana
    use_backend: dnf
    state: present
  async: 3600
  poll: 0

- name: Install Logstash on CentOS
  dnf:
    name: logstash
    use_backend: dnf
    state: present
  async: 3600
  poll: 0

- name: Enable and Start Elasticsearch, Kibana, and Logstash
  systemd:
    name: "[[ item ]]"
    enabled: yes
    state: started
  loop:
    - elasticsearch
    - kibana
    - logstash
  async: 3600
  poll: 0

```

In installing Elasticsearch, Kibana, Logstash on Ubuntu, firstly, I added tasks that add Elasticsearch, Kibana, Logstash YUM repositories. After adding repositories, I then added tasks that install each package. Lastly, I added a task that enables and starts the Elasticsearch, Kibana, and Logstash services.

```

kazuki@workstation:~/CPE232_HOA10.1$ sudo nano install_EKL.yml
kazuki@workstation:~/CPE232_HOA10.1$ cat install_EKL.yml

---
- hosts: all
  become: true
  roles:
    - Hoa10_Ubuntu
    - Hoa10_CentOS

```

This is my main playbook, I used the “roles” command to include the directories “Hoa10_Ubuntu” and “Hoa10_VentOS” so it will run the tasks “main.yml” inside those directories.


```

kazuki@workstation:~/CPE232_HOA10.1$ ansible-playbook --ask-become-pass install_EKL.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.129]
ok: [192.168.56.127]

TASK [Hoa10_Ubuntu : Add Elasticsearch APT repository key] *****
changed: [192.168.56.129]
changed: [192.168.56.127]

TASK [Hoa10_Ubuntu : Add Elasticsearch APT repository] *****
changed: [192.168.56.129]
changed: [192.168.56.127]

TASK [Hoa10_Ubuntu : Add Kibana APT repository] *****
changed: [192.168.56.129]
changed: [192.168.56.127]

TASK [Hoa10_Ubuntu : Add Logstash APT repository] *****
changed: [192.168.56.129]
changed: [192.168.56.127]

TASK [Hoa10_Ubuntu : Install Elasticsearch on Ubuntu] *****
changed: [192.168.56.129]
changed: [192.168.56.127]

TASK [Hoa10_Ubuntu : Install Kibana on Ubuntu] *****
changed: [192.168.56.127]
changed: [192.168.56.129]

TASK [Hoa10_Ubuntu : Install Logstash on Ubuntu] *****
changed: [192.168.56.127]
changed: [192.168.56.129]

TASK [Hoa10_Ubuntu : Enable and Start Elasticsearch, Kibana, and Logstash] *****
changed: [192.168.56.129] => (item=elasticsearch)
changed: [192.168.56.127] => (item=elasticsearch)
changed: [192.168.56.129] => (item=kibana)
changed: [192.168.56.127] => (item=kibana)
changed: [192.168.56.129] => (item=logstash)
changed: [192.168.56.127] => (item=logstash)

TASK [Hoa10_CentOS : Add Elasticsearch YUM repository] *****
changed: [192.168.56.127]
changed: [192.168.56.129]

TASK [Hoa10_CentOS : Add Kibana YUM repository] *****
changed: [192.168.56.129]
changed: [192.168.56.127]

TASK [Hoa10_CentOS : Add Logstash YUM repository] *****
changed: [192.168.56.129]
changed: [192.168.56.127]

TASK [Hoa10_CentOS : Install Elasticsearch on CentOS] *****
changed: [192.168.56.129]
changed: [192.168.56.127]

TASK [Hoa10_CentOS : Install Kibana on CentOS] *****
changed: [192.168.56.129]
changed: [192.168.56.127]

TASK [Hoa10_CentOS : Install Logstash on CentOS] *****
changed: [192.168.56.129]
changed: [192.168.56.127]

TASK [Hoa10_CentOS : Enable and Start Elasticsearch, Kibana, and Logstash] *****
changed: [192.168.56.129] => (item=elasticsearch)
changed: [192.168.56.129] => (item=kibana)
changed: [192.168.56.127] => (item=elasticsearch)
changed: [192.168.56.129] => (item=logstash)
changed: [192.168.56.127] => (item=kibana)
changed: [192.168.56.127] => (item=logstash)

PLAY RECAP *****
192.168.56.127      : ok=16  changed=15  unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.129      : ok=16  changed=15  unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

After creating roles and playbooks, I executed it using the command “ansible-playbook” and added “--ask-become-pass” so there will be no error like password error or any error. The output shows “ok” in gathering facts, meaning it successfully connects to the managed nodes. The “changed” labels shows that it changed that hosts, meaning it successfully do all the tasks inside the roles.

```
kazuki@server3:~$ sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-10-28 22:43:14 +08; 6min ago
     Docs: https://www.elastic.co
   Main PID: 11936 (java)
    Tasks: 65 (limit: 4602)
   Memory: 1.4G
     CPU: 3min 41.061s
   CGroup: /system.slice/elasticsearch.service
           └─11936 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache...
           └─12825 /usr/share/elasticsearch/modules/x-pack-m/ml/platform/linux-x86_64/bin/controller

Oct 28 22:40:50 server3 systemd[1]: Starting Elasticsearch...
Oct 28 22:41:30 server3 systemd-entrypoint[11936]: Oct 28, 2023 10:41:29 PM sun.util.locale.provider.LocaleProviderAdapter <clinit>
Oct 28 22:41:30 server3 systemd-entrypoint[11936]: WARNING: COMPAT locale provider will be removed in a future release
Oct 28 22:41:16 server3 systemd[1]: Started Elasticsearch.
lines 1-16/16 (END)
```

```
[kazuki@centos ~]$ sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; vendor prese...
   Active: active (running) since Sat 2023-10-28 20:44:26 PST; 2h 6min ago
     Docs: https://www.elastic.co
   Main PID: 18719 (java)
    Tasks: 61
   Memory: 608.4M
   CGroup: /system.slice/elasticsearch.service
           └─18719 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.networkad...
           └─19946 /usr/share/elasticsearch/modules/x-pack-m/ml/platform/linux-x86_64/...

Oct 28 20:41:01 centos systemd[1]: Starting Elasticsearch...
Oct 28 20:42:10 centos systemd-entrypoint[18719]: Oct 28, 2023 8:42:16 PM sun.util....
Oct 28 20:42:10 centos systemd-entrypoint[18719]: WARNING: COMPAT locale provider w...
Oct 28 20:44:26 centos systemd[1]: Started Elasticsearch.
Hint: Some lines were ellipsized, use -l to show in full.
```

I used the command “sudo systemctl status elasticsearch” command to show if the status of elasticsearch in both my Ubuntu and CentOS are running or Active since I included in my playbook a task where it enables and starts elasticsearch service.

```
kazuki@server3:~$ sudo systemctl status kibana
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; disabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-10-28 22:44:16 +08; 7min ago
     Docs: https://www.elastic.co
   Main PID: 14807 (node)
    Tasks: 11 (limit: 4602)
   Memory: 334.1M
     CPU: 1min 20.641s
   CGroup: /system.slice/kibana.service
           └─14807 /usr/share/kibana/bin/./node/bin/node /usr/share/kibana/bin/./src/cli/dist --logging.dest=/var/log/kibana/kibana.log

Oct 28 22:44:16 server3 systemd[1]: Started Kibana.
Oct 28 22:44:18 server3 kibana[14807]: Kibana is currently running with legacy OpenSSL providers enabled! For details and instructions, see https://www.elastic.co/guide/en/kibana/current/openssl.html
lines 1-13/13 (END)
```

```
[kazuki@centos ~]$ sudo systemctl status kibana
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2023-10-28 20:41:02 PST; 2h 11min ago
     Docs: https://www.elastic.co
   Main PID: 18758 (node)
    Tasks: 11
   Memory: 240.2M
   CGroup: /system.slice/kibana.service
           └─18758 /usr/share/kibana/bin/./node/bin/node /usr/share/kibana/bin/./s...

Oct 28 20:41:02 centos systemd[1]: Started Kibana.
Oct 28 20:41:08 centos kibana[18758]: Kibana is currently running with legacy Open...
Hint: Some lines were ellipsized, use -l to show in full.
```

I used the command “sudo systemctl status kibana” command to show if the status of elasticsearch in both my Ubuntu and CentOS are running or Active since I included in my playbook a task where it enables and starts elasticsearch service.

```
kazuki@server3:~$ sudo systemctl status logstash
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; disabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-10-28 22:51:53 +08; 27s ago
     Docs: https://www.elastic.co
   Main PID: 15656 (java)
    Tasks: 22 (limit: 4602)
   Memory: 495.2M
     CPU: 40.986s
   CGroup: /system.slice/logstash.service
           └─15656 /usr/share/logstash/jdk/bin/java -Xmsig -Xmxig -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75

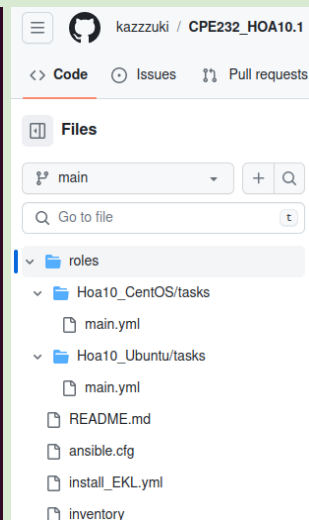
Oct 28 22:51:53 server3 systemd[1]: Started logstash.
Oct 28 22:51:54 server3 logstash[15656]: Using bundled JDK: /usr/share/logstash/jdk
Oct 28 22:51:55 server3 logstash[15656]: OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0
Oct 28 22:52:17 server3 logstash[15656]: Sending Logstash logs to /var/log/logstash which is now configured via log4j2.properties
Oct 28 22:52:17 server3 logstash[15656]: [2023-10-28T22:52:17.424][INFO ][logstash.runner ] Log4j configuration path used
Oct 28 22:52:17 server3 logstash[15656]: [2023-10-28T22:52:17.431][INFO ][logstash.runner ] Starting Logstash ("logstash-...
Oct 28 22:52:17 server3 logstash[15656]: [2023-10-28T22:52:17.438][INFO ][logstash.runner ] JVM bootstrap flags: [-Xmsig,
Oct 28 22:52:19 server3 logstash[15656]: [2023-10-28T22:52:19.389][INFO ][logstash.agent ] Successfully started Logstash
Oct 28 22:52:19 server3 logstash[15656]: [2023-10-28T22:52:19.409][INFO ][logstash.config.source.local.configpathloader] No config
Oct 28 22:52:19 server3 logstash[15656]: [2023-10-28T22:52:19.414][ERROR][logstash.config.sourceloader] No configuration found in t...
lines 1-26/26 (END)
```

```
[kazuki@centos ~]$ sudo systemctl status logstash
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2023-10-28 22:51:37 PST; 2min 24s ago
     Docs: https://www.elastic.co
   Main PID: 7508 (java)
    Tasks: 15
   Memory: 493.3M
   CGroup: /system.slice/logstash.service
           └─7508 /usr/share/logstash/jdk/bin/java -Xmsig -Xmxig -XX:+UseConcMarkSwe...

Oct 28 22:51:37 centos systemd[1]: Started logstash.
Oct 28 22:51:38 centos logstash[7508]: Using bundled JDK: /usr/share/logstash/jdk
Oct 28 22:51:38 centos logstash[7508]: OpenJDK 64-Bit Server VM warning: Option Us...
Hint: Some lines were ellipsized, use -l to show in full.
```

I used the command “sudo systemctl status logstash” command to show if the status of elasticsearch in both my Ubuntu and CentOS are running or Active since I included in my playbook a task where it enables and starts elasticsearch service.

```
kazuki@workstation:~/CPE232_HOA10.1$ git add .
kazuki@workstation:~/CPE232_HOA10.1$ git commit -m "hoa10"
[main 5a67e2d] hoa10
 5 files changed, 146 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 install_EKL.yml
 create mode 100644 inventory
 create mode 100644 roles/Hoa10_CentOS/tasks/main.yml
 create mode 100644 roles/Hoa10_Ubuntu/tasks/main.yml
kazuki@workstation:~/CPE232_HOA10.1$ git push
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 2 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (12/12), 1.37 KiB | 466.00 KiB/s, done.
Total 12 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To github.com:kazzzuki/CPE232_HOA10.1.git
 a090937..5a67e2d main -> main
```



I used the command “git add .” to add all the files in my current directory to my GitHub repository. I used the command “git commit -m “hoa10” command to commit all changes and added a message. Lastly, I push it. GitHub link: https://github.com/kazzzuki/CPE232_HOA10.1

Reflections:

Answer the following:

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

- The benefits of having a log monitoring tool are it helps us in detecting and fixing issues/problems like security breaches or any performance related problems before the problem worsens. For example, when a web server experiences multiple failed login attempts, the monitoring tool can trigger an alert in real time, notifying the system administrator, and then he/she can investigate and block in case it is a potential hacker. So in this example, monitoring tools help in preventing security issue from getting worse.

Conclusions:

In conclusion, this activity involved creating and designing a workflow that install, configure. and manage enterprise monitoring tools using Ansible as IaC tool. This activity taught me the importance of log monitoring tools in maintaining system performance and security. As a system administration student, I will keep this to my mind to always have monitoring tools on the systems I manage. In creating a playbook for installing Elasticsearch, Kibana, and Logstash, I followed all the things I learned from the past activities like using roles. Inside my playbook, I included tasks that add repositories, installing services, and starting the services. I did not encounter any error while doing this activity, the only struggle I experience is my Laptop is lagging when I execute the playbook.