Name: Daniela Marie D. Rabang	Date Performed: 10/23/2023
Course/Section: CPE232/CPE31S4	Date Submitted: 10/31/2023
Instructor: Dr. Jonathan V. Taylar	Semester and SY: 1st Sem 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

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)

Create a repository and clone it into the workstation.

```
daniela@workstation:~$ git clone https://github.com/danielarabang/CPE232_RABANG_HOA10.git
Cloning into 'CPE232_RABANG_HOA10'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
daniela@workstation:~$ cd CPE232_RABANG_HOA10
daniela@workstation:~/CPE232_RABANG_HOA10$
```

- I had created a repository and cloned it into my workstation, so that I can create directories, files, and playbooks that I can push to my repository later on.

Create an inventory file.

```
  GNU nano 2.9.3                                                    inventory

[Ubuntu]
192.168.56.110  ansible_python_interpreter=/usr/bin/python3

[CentOS]
192.168.56.112  ansible_python_interpreter=/usr/bin/python
```

- I had created an inventory file where my Ubuntu and CentOS servers IP address is in.

Create an ansible.cfg file.

```
GNU nano 2.9.3                                                    ansible.cfg

[defaults]

inventory = inventory
host_key checking = False

deprecation_warning = False

remote_user = daniela
private_key_file = ~/.ssh/
```

- I had created a ansible.cfg file where this information is put in.

Create a playbook that is named as install.yml

```
GNU nano 2.9.3                                                    install.yml
- hosts: all
  become: true
  pre_tasks:

  - name: install updates Ubuntu
    apt:
      upgrade: dist
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: install updates CentOS
    yum:
      update_only: yes
      update_cache: yes
    when: ansible_distribution == "CentOS"

- hosts: Ubuntu
  become: true
  roles:
    - Ubuntu

- hosts: CentOS
  become: true
  roles:
    - CentOS
```

- In the main directory I had created a playbook named install.yml where I had assigned the roles for both servers.

Create a directory named roles and inside that directory create two new directories named Ubuntu and CentOS.

```
daniela@workstation:~/CPE232_RABANG_HOA10$ mkdir roles
daniela@workstation:~/CPE232_RABANG_HOA10$ cd roles
daniela@workstation:~/CPE232_RABANG_HOA10/roles$ mkdir CentOS
daniela@workstation:~/CPE232_RABANG_HOA10/roles$ mkdir Ubuntu
```

- I created a directory named roles, this is where I will put the two server directories.

Create a directory inside each role directory named tasks, and create a playbook file named main.yml

```
daniela@workstation:~/CPE232_RABANG_HOA10/roles$ cd CentOS
daniela@workstation:~/CPE232_RABANG_HOA10/roles/CentOS$ mkdir tasks
daniela@workstation:~/CPE232_RABANG_HOA10/roles/CentOS$ cd tasks
daniela@workstation:~/CPE232_RABANG_HOA10/roles/CentOS/tasks$ sudo nano main.yml
```

```
daniela@workstation:~/CPE232_RABANG_HOA10/roles$ cd Ubuntu
daniela@workstation:~/CPE232_RABANG_HOA10/roles/Ubuntu$ mkdir tasks
daniela@workstation:~/CPE232_RABANG_HOA10/roles/Ubuntu$ cd tasks
daniela@workstation:~/CPE232_RABANG_HOA10/roles/Ubuntu/tasks$ sudo nano main.yml
```

- In both server directories I created another directory called tasks and inside that directory I created a playbook called main.yml, this is where all the tasks are.

Create a playbook where it can install the Elastic stack in both roles.

```
daniela@workstation: ~/CPE232_RABANG_HOA10/roles/CentOS/tasks
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                                                    main.yml
---
  - name: install prerequisites (CentOS)
    yum:
      name:
        - java-1.8.0-openjdk
        - epel-release
        - wget
        - which
      state: present
    become: yes

  - name: add Elasticsearch RPM repository (CentOS)
    shell: rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch

  - name: add Elasticsearch YUM repository (CentOS)
    copy:
      content: |
        [elasticsearch-7.x]
        name=Elasticsearch repository for 7.x packages
        baseurl=https://artifacts.elastic.co/packages/7.x/yum
        gpgcheck=1
        gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
        enabled=1
        autorefresh=1
        type=rpm-md
      dest: /etc/yum.repos.d/elasticsearch.repo
    become: yes

  - name: install Elasticsearch (CentOS)
    yum:
      name: elasticsearch
      state: present
    become: yes

  - name: enable and start Elasticsearch service (CentOS)
    systemd:
      name: elasticsearch
      enabled: yes
      state: started
    become: yes

  - name: install Kibana (CentOS)
    yum:
      name: kibana
      state: present
    become: yes

  - name: enable and start Kibana service (CentOS)
    systemd:
      name: kibana
      enabled: yes
      state: started
    become: yes

  - name: install Logstash (CentOS)
    yum:
      name: logstash
      state: present
    become: yes

  - name: enable and start Logstash service (CentOS)
    systemd:
      name: logstash
      enabled: yes
      state: started
    become: yes

  - name: restart Elasticsearch and Kibana (CentOS)
    systemd:
      name: "{{ item }}"
      state: restarted
    loop:
      - elasticsearch
      - kibana
```

- The tasks directory has a main.yml and inside that can install Elastic Stack for the CentOS server.

File Edit View Search Terminal Help

```
GNU nano 2.9.3                                                          main.yml
---
  - name: install prerequisites (Ubuntu)
    apt:
      name:
        - default-jre
        - apt-transport-https
        - curl
        - software-properties-common
      state: present
    become: yes

  - name: add Elasticsearch APT repository key (Ubuntu)
    apt_key:
      url: https://artifacts.elastic.co/GPG-KEY-elasticsearch
    become: yes

  - name: add Elasticsearch APT repository (Ubuntu)
    apt_repository:
      repo: "deb https://artifacts.elastic.co/packages/7.x/apt stable main"
      state: present
    become: yes

  - name: install Elasticsearch (Ubuntu)
    apt:
      name: elasticsearch
      state: present
    become: yes

  - name: enable and start Elasticsearch service (Ubuntu)
    systemd:
      name: elasticsearch
      enabled: yes
      state: started
    become: yes

  - name: install Kibana (Ubuntu)
    apt:
      name: kibana
      state: present
    become: yes

  - name: enable and start Kibana service (Ubuntu)
    systemd:
      name: kibana
      enabled: yes
      state: started
    become: yes

  - name: install Logstash (Ubuntu)
    apt:
      name: logstash
      state: present
    become: yes

  - name: enable and start Logstash service (Ubuntu)
    systemd:
      name: logstash
      enabled: yes
      state: started
    become: yes

  - name: restart Elasticsearch and Kibana (Ubuntu)
    systemd:
      name: "{{ item }}"
      state: restarted
    loop:
      - elasticsearch
      - kibana
```

- The tasks directory has a main.yml and inside that can install Elastic Stack for the Ubuntu server.

Run the playbook install.yml

```
daniela@workstation:~/CPE232_RABANG_HOA10$ ansible-playbook --ask-become-pass install.yml
[DEPRECATION WARNING]: Ansible will require Python 3.8 or newer on the controller starting with Ansible 2.12. Currer
[GCC 8.4.0]. This feature will be removed from ansible-core in version 2.12. Deprecation warnings can be disabled by
BECOME password:

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

TASK [Gathering Facts] **********************************************************
ok: [192.168.56.112]
ok: [192.168.56.110]

TASK [install updates Ubuntu] ***************************************************
skipping: [192.168.56.112]
ok: [192.168.56.110]

TASK [install updates CentOS] ***************************************************
skipping: [192.168.56.110]
ok: [192.168.56.112]

PLAY [Ubuntu] *******************************************************************

TASK [Gathering Facts] **********************************************************
ok: [192.168.56.110]

TASK [Ubuntu : install prerequisites (Ubuntu)] **********************************
ok: [192.168.56.110]

TASK [Ubuntu : add Elasticsearch APT repository key (Ubuntu)] *******************
ok: [192.168.56.110]

TASK [Ubuntu : add Elasticsearch APT repository (Ubuntu)] ***********************
ok: [192.168.56.110]

TASK [Ubuntu : install Elasticsearch (Ubuntu)] **********************************
ok: [192.168.56.110]

TASK [Ubuntu : enable and start Elasticsearch service (Ubuntu)] *****************
ok: [192.168.56.110]

TASK [Ubuntu : install Kibana (Ubuntu)] *****************************************
ok: [192.168.56.110]

TASK [Ubuntu : enable and start Kibana service (Ubuntu)] ************************
ok: [192.168.56.110]

TASK [Ubuntu : install Logstash (Ubuntu)] ***************************************
ok: [192.168.56.110]

TASK [Ubuntu : enable and start Logstash service (Ubuntu)] **********************
ok: [192.168.56.110]

TASK [Ubuntu : restart Elasticsearch and Kibana (Ubuntu)] ***********************
changed: [192.168.56.110] => (item=elasticsearch)
changed: [192.168.56.110] => (item=kibana)

PLAY [CentOS] *******************************************************************

TASK [Gathering Facts] **********************************************************
ok: [192.168.56.112]

TASK [CentOS : install prerequisites (CentOS)] **********************************
ok: [192.168.56.112]

TASK [CentOS : add Elasticsearch RPM repository (CentOS)] ***********************
ok: [192.168.56.112]

TASK [CentOS : add Elasticsearch YUM repository (CentOS)] ***********************
ok: [192.168.56.112]

TASK [CentOS : install Elasticsearch (CentOS)] **********************************
ok: [192.168.56.112]

TASK [CentOS : enable and start Elasticsearch service (CentOS)] *****************
ok: [192.168.56.112]

TASK [CentOS : install Kibana (CentOS)] *****************************************
ok: [192.168.56.112]

TASK [CentOS : enable and start Kibana service (CentOS)] ************************
ok: [192.168.56.112]

TASK [CentOS : install Logstash (CentOS)] ***************************************
ok: [192.168.56.112]

TASK [CentOS : enable and start Logstash service (CentOS)] **********************
ok: [192.168.56.112]

TASK [CentOS : restart Elasticsearch and Kibana (CentOS)] ***********************
changed: [192.168.56.112] => (item=elasticsearch)
changed: [192.168.56.112] => (item=kibana)

PLAY RECAP **********************************************************************
192.168.56.110             : ok=13   changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.112             : ok=13   changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```
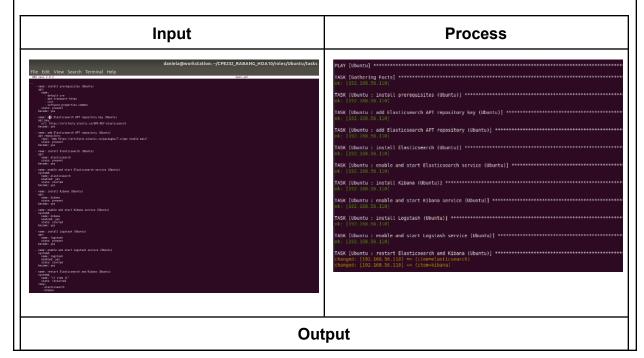
- In this part you can see the results when you run the playbook where it tasks to install an elastic stack for both Ubuntu and CentOS server.

Git add *, Git commit -m, and Git push origin main.


```
daniela@workstation:~/CPE232_RABANG_HOA10$ git push origin main
Username for 'https://github.com': daniela
Password for 'https://daniela@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 359 bytes | 359.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/danielarabang/CPE232_RABANG_HOA10.git
   51f7004..e1f8b6c  main -> main
```
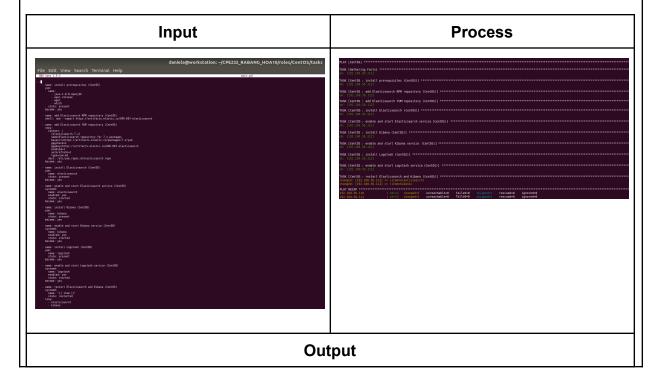
- In this part I had just pushed one file, this is because I already pushed all the files and directories that I had done from my PC in the computer lab but I forgot to screenshot it.

IPO for Ubuntu

Input	Process

Output

```
daniela@server1:~$ sudo systemctl status logstash
[sudo] password for daniela:
● logstash.service - logtash
   Loaded: loaded (etc/systemd/system/logstash.service: enabled; vendor preset
   Active: active (running) since Sat 2023-10-28 14:30:22 PST; 6s ago
 Main PID: 12978 (java)
    Tasks: 15 (limit: 4656)
   CGroup: /system.slice/logstash.service
           └─12978 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseConcM

Oct 28 14:30:22 server1 systemd[1]: Started logstash.
Oct 28 14:30:22 server1 logstash[12978]: Using bundled JDK: /usr/share/logstash
Oct 28 14:30:22 server1 logstash[12978]: OpenJDK 64-Bit Server VM warning: Opti
lines 1-19/19 (END)
```

```
daniela@server1:~$ sudo systemctl status Kibana
[sudo] password for daniela:
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset:
   Active: active (running) since Sat 2023-10-28 14:31:18 PST; 6s ago
     Docs: https://www.elastic.co
 Main PID: 12094 (node)
    Tasks: 11 (limit: 4656)
   CGroup: /system.slice/kibana.service
           └─12094 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bi

Oct 28 14:31:18 server1 systemd[1]: Started Kibana.
Oct 28 14:31:18 server1 kibana[12094]: Kibana is currently running with legacy
lines 1-11/11 (END)
```

- This shows the input, process, and the output of the playbook when it is run.

IPO for CentOS

Input	Process
Output	

```
[daniela@localhost ~]$ systemctl status logstash
● logstash.service - logstash
    Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: disabl
ed)
    Active: active (running) since Sat 2023-10-28 14:40:16 EDT; 9s ago
 Main PID: 12157 (java)
    Tasks: 15
   CGroup: /system.slice/logstash.service
           └─12157 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSw..

Oct 28 14:40:16 localhost.localdomain systemd[1]: Started logstash.
Oct 28 14:40:16 localhost.localdomain logstash[12157]: Using bundled JDK: /usr/shar...k
Oct 28 14:40:16 localhost.localdomain logstash[12157]: OpenJDK 64-Bit Server VM war....
Hint: Some lines were ellipsized, use -l to show in full


[daniela@localhost ~]$ systemctl status kibana
● kibana.service - Kibana
    Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: disa

    Active: active (running) since Sat 2023-10-28 14:41:20 EDT; 9s ago
      Docs: https://www.elastic.co
 Main PID: 11625 (node)
    Tasks: 11
   CGroup: /system.slice/kibana.service
           └─11625 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/../s.

Oct 28 14:41:20 localhost.localdomain systemd[1]: Started Kibana.
Oct 28 14:41:20 localhost.localdomain kibana[11625]: Kibana is currently running wit.
Hint: Some lines were ellipsized, use -l to show in full.
```

- This shows the input, process, and the output of the playbook when it is run.

Reflections:

Answer the following:

1. **What are the benefits of having a log monitoring tool?**
   - There are many benefits of having a log monitoring, but the most benefits is that with this kind of monitoring tools like the one that is used in this hands-on activity is the Logstash, this kind of monitoring tool is used to handle pipelines, collects data that comes from different sources.

Conclusions:

In this hands-on activity I am tasked to create a playbook where I had to accomplished the following tasks where my two servers which is the Ubuntu and the CentOS server that I have will be installed the Elastic Stack where it is consists of the Elasticsearch, Logstash, and the Kibana. This playbook is created into the workstation, using roles so that the following packages can be installed in the right server. First I had created my repository in github so that I can push my work after all. Then cloned it into my workstation so that I can access it. After this in the main directory I had created an inventory file where I put all the Ip address of my servers, and the ansible.cfg file was also created. I also had created an install.yml where I assigned the following roles when the playbook for both servers are run from the separate directories in the roles directory. In the tasks directory I had created a playbook named main.yml in this playbook I put all the packages and everything that can install the following tasks needed to be able to finish this activity. Then I had run the playbook, I had difficulties when I was running the playbooks because of the errors but after all I had fixed it and the following packages are installed in both servers.