Name: Torrecampo, Juan Piolo S.	**Date Performed:** Oct 21, 2022
Course/Section: CPE 232 / CPE31S22	**Date Submitted:** Oct 24, 2022
Instructor: Dr. Jonathan Taylar	**Semester and SY:** 1st Sem, 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).

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.

Source: https://www.elastic.co/elastic-stack

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

Requirements:

- 3 VMs for ElasticSearch Server, Kibana, and Logstash

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)

STEP 1. Create a new repository and clone it into your machine.

Figure 1.1. Github page of the new repository.

```
~/Desktop $ git clone https://github.com/piolotorrecampo/CPE232-Activity_10.git
Cloning into 'CPE232-Activity_10'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
~/Desktop $
```

Figure 1.2. Cloning the recently created repository in the host machine.

STEP 2. Copy the inventory and ansible.cfg from the previous repository in Activity 9.

```
~/Desktop $ cd CPE232-Activity_9/
~/D/CPE232-Activity_9 (main|✓) $ ll
total 12K
-rw-r--r-- 1 papzi papzi  124 Oct 19 14:34 ansible.cfg
drwxr-xr-x 1 papzi papzi   36 Oct 19 22:02 files/
-rw-r--r-- 1 papzi papzi 1.1K Oct 20 01:46 install_prometheus.yml
-rw-r--r-- 1 papzi papzi  120 Oct 20 01:46 inventory
drwxr-xr-x 1 papzi papzi   68 Oct 19 14:38 roles/
~/D/CPE232-Activity_9 (main|✓) $ cp ansible.cfg $HOME/Desktop/CPE232-Activity_10/
~/D/CPE232-Activity_9 (main|✓) $ cp inventory $HOME/Desktop/CPE232-Activity_10/
~/D/CPE232-Activity_9 (main|✓) $ cd ..
~/Desktop $ cd CPE232-Activity_10/ && ll
total 12K
-rw-r--r-- 1 papzi papzi 124 Oct 20 13:28 ansible.cfg
-rw-r--r-- 1 papzi papzi 120 Oct 20 13:28 inventory
-rw-r--r-- 1 papzi papzi  67 Oct 20 13:21 README.md
```

Figure 1.3. Copying the inventory and ansible.cfg from the previous repository.

STEP 3. Create the needed files and directories, as shown in *Figure 1.4.*

```
~/D/CPE232-Activity_10 (main|✓) $ tree
.
├── ansible.cfg
├── files
│   ├── kibana.repo
│   └── logstash.repo
├── install_elastic_stack.yml
├── inventory
├── README.md
└── roles
    ├── elasticsearch_server
    │   └── tasks
    │       └── main.yml
    ├── kibana_server
    │   └── tasks
    │       └── main.yml
    └── logstash_server
        └── tasks
            └── main.yml
```

Figure 1.4. Tree structure of the directory.

```
~/D/CPE232-Activity_10 (main↑1|✹9) $ cd ..
~/Desktop $ cd CPE232-Activity_9/
~/D/CPE232-Activity_9 (main|✓) $ cp ansible.cfg inventory ~/Desktop/CPE232-Activity_10/
~/D/CPE232-Activity_9 (main|✓) $ cd ..
~/Desktop $ cd CPE232-Activity_10/ && ll
total 8.0K
-rw-r--r-- 1 papzi papzi 124 Oct 24 14:48 ansible.cfg
-rw-r--r-- 1 papzi papzi 120 Oct 24 14:48 inventory
~/D/CPE232-Activity_10 (main↑1|✹8) $ mkdir files
~/D/CPE232-Activity_10 (main↑1|✹8) $ mkdir files/kibana.repo files/logstash.repo
~/D/CPE232-Activity_10 (main↑1|✹8) $ touch install_elastic_stack.yml
~/D/CPE232-Activity_10 (main↑1|✹8) $ mkdir roles
~/D/CPE232-Activity_10 (main↑1|✹8) $ mkdir roles/elasticsearch_server/tasks roles/logstash_server/tasks roles/kibana_server/tasks
mkdir: cannot create directory 'roles/elasticsearch_server/tasks': No such file or directory
mkdir: cannot create directory 'roles/logstash_server/tasks': No such file or directory
mkdir: cannot create directory 'roles/kibana_server/tasks': No such file or directory
~/D/CPE232-Activity_10 (main↑1|✹8) [1]$ mkdir roles/elasticsearch_server roles/logstash_server roles/kibana_server
~/D/CPE232-Activity_10 (main↑1|✹8) $ mkdir roles/elasticsearch_server/tasks roles/logstash_server/tasks roles/kibana_server/tasks
~/D/CPE232-Activity_10 (main↑1|✹8) $ touch roles/elasticsearch_server/tasks/main.yml roles/kibana_server/tasks/main.yml roles/logstash_server/tasks/main.yml
~/D/CPE232-Activity_10 (main↑1|✹8) $ tree
```

Figure 1.5. Creating the needed files and directories inside of the repository.

STEP 4. Modifying the inventory file to support three servers. Each server corresponds to the three softwares of the elastic stack.

```
inventory
[logstash_server]
192.168.122.209 ansible_user=logstash-server

[elasticsearch_server]
192.168.122.183 ansible_user=elasticsearch-server

[kibana_server]
192.168.122.234 ansible_user=kibana-server
```

FIgure 1.6. The screenshot above shows the contents of the inventory file after modifying it.

STEP 5. Creating the tasks of install_elastic_stack.yml. This serves as your main control to the whole playbook.

```yaml
---

- hosts: all
  become: true
  pre_tasks:

  - name: Installing dnf
    yum:
      name:
        - dnf
    when: ansible_distribution == "CentOS"

  - name: Update and upgrade remote CentOS server
    dnf:
      update_cache: yes
      name:  "*"
      state: latest
    when: ansible_distribution == "CentOS"

- hosts: kibana_server
  tags: kibana
  become: true
  roles:
    - { role: kibana_server, target_ip: 192.168.122.183, ip_host: 192.168.122.234, gpg_key_kibana: https://artifacts.elastic.c

- hosts: elasticsearch_server
  tags: es
  become: true
  roles:
    - { role: elasticsearch_server, ip_host: 192.168.122.183 }

- hosts: logstash_server
  tags: logstash
  become: true
  roles:
    - role: logstash_server
      gpg_key_logstash: https://artifacts.elastic.co/GPG-KEY-elasticsearch
```

Figure 1.7. Contents of install_elastic_stack.yml.

STEP 6. Adding repo contents to the .repos files. The contents below are the same with logstash.repo.

```
[kibana-8.x]
name=Kibana repository for 8.x packages
baseurl=https://artifacts.elastic.co/packages/8.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
```

Figure 1.8. Contents of kibana.repo file.

```
logstash.repo
[logstash-8.x]
name=Elastic repository for 8.x packages
baseurl=https://artifacts.elastic.co/packages/8.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
```

Figure 1.9. Contents of logstash.repo file.

STEP 7. Adding contents to the roles.

```
main.yml
- name: Downloading the source file of elastic search
  get_url:
    url: https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.4.3-x86_64.rpm
    dest: /tmp/elasticsearch-8.4.3-x86_64.rpm

- name: Installing elasticsearch
  yum:
    name: /tmp/elasticsearch-8.4.3-x86_64.rpm
    state: present

- name: Configuring cluster.name
  replace:
    path: /etc/elasticsearch/elasticsearch.yml
    regexp: '#cluster.name: my-application'
    replace: 'cluster.name: Torrecampo_cluster'

- name: Configuring node.name
  replace:
    path: /etc/elasticsearch/elasticsearch.yml
    regexp: '#node.name: node-1'
    replace: 'node.name: node-1'

- name: Configuring network.host
  replace:
    path: /etc/elasticsearch/elasticsearch.yml
    regexp: '#network.host: 192.168.0.1'
    replace: 'network.host: 192.168.122.183'

- name: Configuring http.port
  replace:
    path: /etc/elasticsearch/elasticsearch.yml
    regexp: '#http.port: 9200'
    replace: 'http.port: 9200'

- name: Configuring network.host
  replace:
    path: /etc/elasticsearch/elasticsearch.yml
    regexp: 'xpack.security.enabled: true'
    replace: 'xpack.security.enabled: false'

- name: Enabling elastic search service
  service:
    name: elasticsearch
    enabled: yes

- name: Modifying service file
  replace:
    path: /usr/lib/systemd/system/elasticsearch.service
```

```
    replace: "TimeoutStartSec=300"

- name: Opening port for elastic search
  shell: |
    sudo firewall-cmd --permanent --zone=public --add-port=9200/tcp
    sleep 10
    sudo firewall-cmd --reload

- name: Enabling elastic search service
  shell: |
    systemctl enable elasticsearch.service
    sleep 10
    systemctl start elasticsearch.service
```

Figure 1.10. The screenshot above shows the installing and configuration script for elastic search where the roles are implemented.

```
main.yml

- name: Downloading and installing public signing key
  rpm_key:
    state: present
    key: "{{ gpg_key_logstash }}"

- name: Creeating a repo file for Logstash
  copy:
    src: logstash.repo
    dest: /etc/yum.repos.d/logstash.repo
    owner: root
    group: root
    mode: 0777

- name: Updating repo
  dnf:
    update_cache: yes

- name: Installing Logstash and its dependencies
  dnf:
    name:
      - logstash
    state: latest

- name: Opening port for Logstash
  shell: |
    sudo firewall-cmd --permanent --zone=public --add-port=9600/tcp
    sleep 10
    sudo firewall-cmd --reload

- name: Making sure that logstash is stared and enabled
  service:
    name: logstash
    state: restarted
    enabled: true
```

Figure 1.11. The screenshot above shows the installing and configuration script for logstash where the roles are implemented.

```
main.yml
- name: Downloading and installing public signing key
  rpm_key:
    state: present
    key: "{{ gpg_key_kibana }}"

- name: Adding kibana to the RPM repository
  copy:
    src: kibana.repo
```

```
    dest: /etc/yum.repos.d/kibana.repo
    owner: root
    group: root
    mode: 777

- name: Updating the repository once again
  yum:
    name:
      - kibana
    state: latest

- name: Cofiguring server.port
  replace:
    path: /etc/kibana/kibana.yml
    regexp: '#server.port: 5601'
    replace: 'server.port: 5601'

- name: Cofiguring server.host
  replace:
    path: /etc/kibana/kibana.yml
    regexp: '#server.host: "localhost"'
    replace: 'server.host: "{{ ip_host }}"'

- name: Cofiguring elasticsearch.hosts
  replace:
    path: /etc/kibana/kibana.yml
    regexp: '#elasticsearch.hosts: ["http://localhost:9200"]'
    replace: '"elasticsearch.hosts: ["http://{{ target_ip }}:9200"]"'

- name: Opening port for elastic search
  firewalld:
    port: 5601/tcp
    zone: public
    permanent: yes
    state: enabled

- name: Making sure that Kibana is started and enabled
  service:
    name: kibana
    state: restarted
    enabled: true
```

Figure 1.12. The screenshot above shows the installing and configuration script for kibana where the roles are implemented.

STEP 8. Checking SSH connection and for syntax errors by running "ansible all -m ping" for the remote connection and "ansible-playbook –syntax-check file.yml" for syntax checking.

```
~/D/CPE232-Activity_10 (main|✔) $ ansible all -m ping
192.168.122.183 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
192.168.122.209 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
```

```
    "ping": "pong"
}
192.168.122.234 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
~/D/CPE232-Activity_10 (main|✔) $
```

Figure 1.13. Conducting a test to make sure the three servers are capable in running ssh.

```
~/D/CPE232-Activity_10 (main|✔) $ ansible-playbook --syntax-check install_elastic_stack.yml

playbook: install_elastic_stack.yml
~/D/CPE232-Activity_10 (main|✔) $
```

Figure 1.14. Verifying that the playbook has no errors.

STEP 9. Run the playbook. The provided tags can be used to install the applications separately.

```
~/D/CPE232-Activity_10 (main|✔) [99]$ ansible-playbook --ask-become-pass install_elastic_stack.yml --tags=es
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.122.209]
ok: [192.168.122.183]
ok: [192.168.122.234]

PLAY [kibana_server] ************************************************************

PLAY [elasticsearch_server] *****************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.122.183]

TASK [elasticsearch_server : Downloading the source file of elastic search] *******
changed: [192.168.122.183]

TASK [elasticsearch_server : Installing elasticsearch] *****************************
changed: [192.168.122.183]

TASK [elasticsearch_server : Configuring cluster.name] *****************************
changed: [192.168.122.183]

TASK [elasticsearch_server : Configuring node.name] ********************************
changed: [192.168.122.183]

TASK [elasticsearch_server : Configuring network.host] *****************************
changed: [192.168.122.183]

TASK [elasticsearch_server : Configuring http.port] ********************************
changed: [192.168.122.183]

TASK [elasticsearch_server : Configuring network.host] *****************************
changed: [192.168.122.183]
```

```
TASK [elasticsearch_server : Enabling elastic search service] *********************
changed: [192.168.122.183]

TASK [elasticsearch_server : Modifying service file] ****************************
changed: [192.168.122.183]

TASK [elasticsearch_server : Opening port for elastic search] *********************
changed: [192.168.122.183]

TASK [elasticsearch_server : Enabling elastic search service] *********************
changed: [192.168.122.183]

PLAY [logstash_server] ***********************************************************

PLAY RECAP ***********************************************************************
192.168.122.183            : ok=13   changed=11   unreachable=0    failed=0    skip
ped=0    rescued=0    ignored=0
192.168.122.209            : ok=1    changed=0    unreachable=0    failed=0    skip
ped=0    rescued=0    ignored=0
192.168.122.234            : ok=1    changed=0    unreachable=0    failed=0    skip
ped=0    rescued=0    ignored=0
```

Figure 1.15. The screenshot above shows the result after running the install_elastic_stack.yml with the tag of es that stands for elastic search.

```
~/D/CPE232-Activity_10 (main|✔) $ ansible-playbook --ask-become-pass install_elasti
c_stack.yml --tags=logstash
BECOME password:

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

TASK [Gathering Facts] ***********************************************************
ok: [192.168.122.209]
ok: [192.168.122.183]
ok: [192.168.122.96]

PLAY [kibana_server] *************************************************************

PLAY [elasticsearch_server] ******************************************************

PLAY [logstash_server] ***********************************************************

TASK [Gathering Facts] ***********************************************************
ok: [192.168.122.209]

TASK [logstash_server : Downloading and installing public signing key] ************
ok: [192.168.122.209]

TASK [logstash_server : Creeating a repo file for Logstash] ***********************
ok: [192.168.122.209]

TASK [logstash_server : Updating repo] *******************************************
ok: [192.168.122.209]

TASK [logstash_server : Installing Logstash and its dependencies] *****************
changed: [192.168.122.209]

TASK [logstash_server : Opening port for Logstash] *******************************
changed: [192.168.122.209]

TASK [logstash_server : Making sure that logstash is stared and enabled] **********
changed: [192.168.122.209]

PLAY RECAP ***********************************************************************
```

```
192.168.122.183            : ok=1    changed=0    unreachable=0    failed=0    skip
ped=0    rescued=0    ignored=0
192.168.122.209            : ok=8    changed=3    unreachable=0    failed=0    skip
ped=0    rescued=0    ignored=0
192.168.122.96             : ok=1    changed=0    unreachable=0    failed=0    skip
ped=0    rescued=0    ignored=0

⏎                                                    ~/D/CPE232-Activity_10 (m
~/D/CPE232-Activity_10 (main|✔) $
```

Figure 1.16. The screenshot above shows the result after running the install_elastic_stack.yml with the tags of kibana.

```
~/D/CPE232-Activity_10 (main|✔) $ ansible-playbook --ask-become-pass install_elasti
c_stack.yml --tags=logstash
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.122.209]
ok: [192.168.122.183]
ok: [192.168.122.96]

PLAY [kibana_server] ***********************************************************

PLAY [elasticsearch_server] ****************************************************

PLAY [logstash_server] *********************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.122.209]

TASK [logstash_server : Downloading and installing public signing key] ************
ok: [192.168.122.209]

TASK [logstash_server : Creeating a repo file for Logstash] ***********************
ok: [192.168.122.209]

TASK [logstash_server : Updating repo] *****************************************
ok: [192.168.122.209]

TASK [logstash_server : Installing Logstash and its dependencies] *****************
changed: [192.168.122.209]

TASK [logstash_server : Opening port for Logstash] *****************************
changed: [192.168.122.209]

TASK [logstash_server : Making sure that logstash is stared and enabled] **********
changed: [192.168.122.209]

PLAY RECAP *********************************************************************
192.168.122.183            : ok=1    changed=0    unreachable=0    failed=0    skip
ped=0    rescued=0    ignored=0
192.168.122.209            : ok=8    changed=3    unreachable=0    failed=0    skip
ped=0    rescued=0    ignored=0
192.168.122.96             : ok=1    changed=0    unreachable=0    failed=0    skip
ped=0    rescued=0    ignored=0

⏎                                                    ~/D/CPE232-Activity_10 (m
~/D/CPE232-Activity_10 (main|✔) $
```

Figure 1.17. The screenshot above shows the result after running the install_elastic_stack.yml with the tags of logstash.

STEP 10. Verifying that the elastic stack is functional.

VERIFYING THE INSTALLATION FOR ELASTIC STACK	
Name	**Screenshot**
Kibana	 [sudo] password for kibana-server: ● kibana.service - Kibana Loaded: loaded (/usr/lib/systemd/system/kibana.service; enabled; vendor prese t: disabled) Active: active (running) since Mon 2022-10-24 03:58:14 EDT; 43s ago Docs: https://www.elastic.co Main PID: 3315 (node) Tasks: 11 CGroup: /system.slice/kibana.service └3315 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bi... Oct 24 03:58:14 localhost.localdomain systemd[1]: Started Kibana. Oct 24 03:58:22 localhost.localdomain kibana[3315]: [2022-10-24T03:58:22.074-... Hint: Some lines were ellipsized, use -l to show in full. [kibana-server@localhost ~]$ Kibana Service Kibana's web page.
Elastic Search	

```
~ [255]$ ssh logstash-server@192.168.122.209
Last login: Sun Oct 23 22:27:44 2022 from gateway
[logstash-server@localhost ~]$ sudo systemctl status logstash
[sudo] password for logstash-server:
● logstash.service - logstash
   Loaded: loaded (/usr/lib/systemd/system/logstash.service; enabled; vendor preset: disabled
)
   Active: active (running) since Mon 2022-10-24 03:41:28 EDT; 1min 15s ago
 Main PID: 1387 (java)
   CGroup: /system.slice/logstash.service
           └─1387 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -Djava.awt.headless=true ...

Oct 24 03:41:28 localhost.localdomain systemd[1]: logstash.service holdoff time over, sc...t.
Oct 24 03:41:28 localhost.localdomain systemd[1]: Stopped logstash.
Oct 24 03:41:28 localhost.localdomain systemd[1]: Started logstash.
Oct 24 03:41:28 localhost.localdomain logstash[1387]: Using bundled JDK: /usr/share/logst...k
Hint: Some lines were ellipsized, use -l to show in full.
[logstash-server@localhost ~]$
```

Elasticsearch Service

```
[elasticsearch-server@localhost ~]$ curl 192.168.122.183:9200
{
  "name" : "node-1",
  "cluster_name" : "Torrecampo_cluster",
  "cluster_uuid" : "_na_",
  "version" : {
    "number" : "8.4.3",
    "build_flavor" : "default",
    "build_type" : "rpm",
    "build_hash" : "42f05b9372a9a4a470db3b52817899b99a76ee73",
    "build_date" : "2022-10-04T07:17:24.662462378Z",
    "build_snapshot" : false,
    "lucene_version" : "9.3.0",
    "minimum_wire_compatibility_version" : "7.17.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "You Know, for Search"
}
[elasticsearch-server@localhost ~]$
```

Verifying if the cluster is running.

```
[elasticsearch-server@localhost ~]$ sudo firewall-cmd --list-ports
9200/tcp
[elasticsearch-server@localhost ~]$
```

Elasticsearch Open Ports

Web view of the elastic search cluster.

Logstash

Logstash Service

Logstash Open Ports

Table 1.1. The table above shows the proof that the software is currently running on a server.

STEP 11. Pushing the updated repo to Github.

Figure 1.18. The screenshot above shows the command used in pushing the files.

Figure 1.19. The screenshot above shows my updated Github page.

References:

https://www.elastic.co/guide/en/logstash/8.4/running-logstash.html
https://www.elastic.co/guide/en/elasticsearch/reference/current/deb.html
https://www.elastic.co/guide/en/kibana/current/deb.html
https://thelionping.com/elasticsearch-and-kibana-installation-and-configuration/

Github Link: https://github.com/piolotorrecampo/CPE232-Activity_10.git

Reflections:

Answer the following:

1. What are the benefits of having a log monitoring tool?
   - The benefits of having a monitoring tool is that a system administrator can monitor multiple servers without physically interacting with it. This activity introduces the concept of elastic stack. The elastic stack is a group of three softwares that focuses on obtaining data from the server and process that data then visualize them. One of the members of the elastic stack is the Logstash. Logstash is used to obtain logs from the servers and filter them and forward them to the elastic search. The elastic search on the other hand is the database wherein the logs are being stored. The visualization of data happens in Kibana. Kibana learns the data and expresses it into a graph form for better understandment.

Conclusions:

The activity achieved its goal to install and configure enterprise log monitoring tools. It introduces the new enterprise log monitoring tools called the elastic stack or ELK stack. The ELK stack is abbreviated from Elasticsearch, Logstash and KIbana. Connecting the softwares is called the elastic stack. The Logstash is used to obtain and filter the data from the servers then store them to the Elasticsearch. Therefore, Elasticsearch is the database where this data can be displayed through Kibana. On the other hand the Kibana is used to visualize the data from the server like a spreadsheet with graphs.

Honor Pledge:

"I affirm that I will not give or receive unauthorized help on this activity and that all will be my own."