

Name: Peñas, Issa Victoria H.	Date Performed: 10/23/2022
Course/Section: CPE232 - CPE31S22	Date Submitted: 10/24/2022
Instructor: Dr. Jonathan V. Taylor	Semester and SY: 1st Semester (SY: 2021 - 2022)
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 <p>Elastic Stack</p> <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> <p>GrayLog</p> <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 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</p>	

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)

1. Create a new repository in GitHub under the name of **HOA-9.1- Prometheus**, and make sure that the repository is **Public**. As a good practice add a **README.md** file and input any related information regarding your inserted repository

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * Repository name *

IVPENAS / HOA-10.1-Elastic ✓

Great repository names are short and memorable. Need inspiration? How about [super-duper-waddle?](#)

Description (optional)

☐ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▼

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: None ▼

This will set `main` as the default branch. Change the default name in your [settings](#).

You are creating a public repository in your personal account.

[Create repository](#)

Figure 1.1. Shows the creation of New Repository in Github

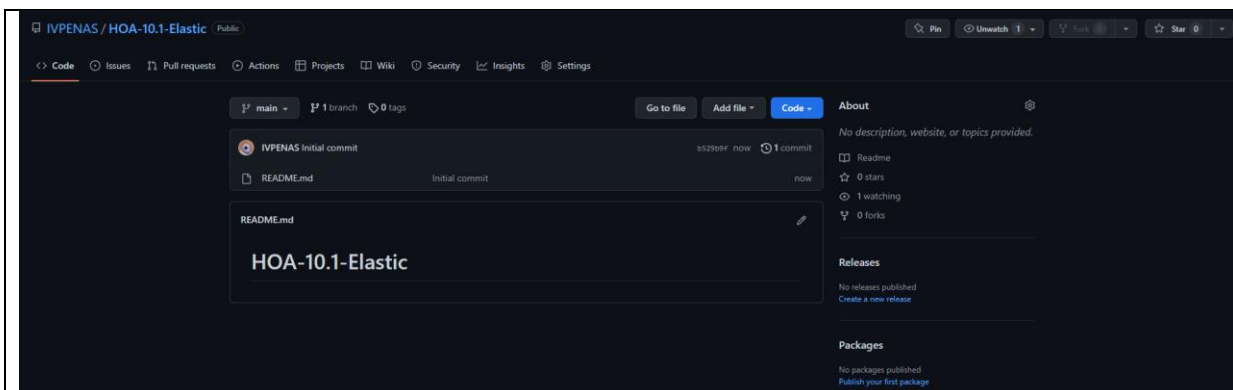


Figure 1.2. Shows the Created Repository named HOA-9.1-Prometheus

2. Make sure that your Local Server was up to date, if not execute the commands: **sudo apt update**, and **sudo apt upgrade**.

```

penas@penas-VirtualBox:~$ sudo apt update
Hit:1 http://ph.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
penas@penas-VirtualBox:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
Try Ubuntu Pro beta with a free personal subscription on up to 5 machines.
Learn more at https://ubuntu.com/pro
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
penas@penas-VirtualBox:~$

```

Figure 1.3. Shows the Local Server was up-to-date

3. Now clone the created repository from GitHub to the Local Server for future synchronization of the Activity.

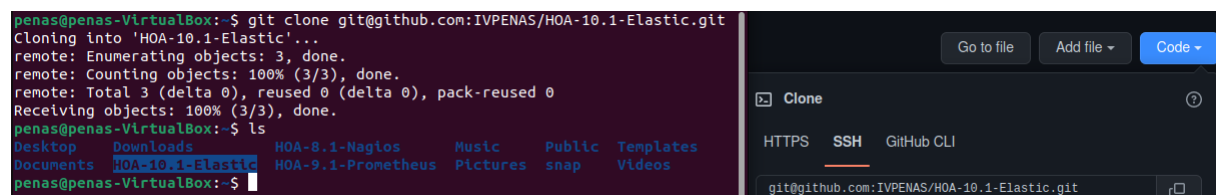


Figure 1.4. Cloning the created Repository from GitHub to the Local Host

4. To ease the workload, you can copy the previous `global_playbook.yml`, `inventory`, and `ansible.cfg` files to the new folder and just modify it, using the command `cp` with the syntax of `cp [filename] [destination]`.

```
penas@penas-VirtualBox:~/HOA-10.1-Elastic$ pwd
/home/penas/HOA-10.1-Elastic
penas@penas-VirtualBox:~/HOA-10.1-Elastic$ cd ..
penas@penas-VirtualBox:~$ cd HOA-9.1-Prometheus
penas@penas-VirtualBox:~/HOA-9.1-Prometheus$ cp ansible.cfg /home/penas/HOA-10.1-Elastic
penas@penas-VirtualBox:~/HOA-9.1-Prometheus$ cp inventory.cfg /home/penas/HOA-10.1-Elastic
cp: cannot stat 'inventory.cfg': No such file or directory
penas@penas-VirtualBox:~/HOA-9.1-Prometheus$ cp inventory /home/penas/HOA-10.1-Elastic
penas@penas-VirtualBox:~/HOA-9.1-Prometheus$ cp global_playbook.yml /home/penas/HOA-10.1-Elastic
penas@penas-VirtualBox:~/HOA-9.1-Prometheus$ cd ..
penas@penas-VirtualBox:~$ cd HOA-10.1-Elastic
penas@penas-VirtualBox:~/HOA-10.1-Elastic$ ls
ansible.cfg  global_playbook.yml  inventory  README.md
penas@penas-VirtualBox:~/HOA-10.1-Elastic$
```

Figure 1.5. Copying the `global_playbook.yml`, `inventory`, and `ansible.cfg` files which to be modified later

5. When initiating roles the admin should create directories using the command `mkdir roles` where we store multiple roles for the main playbook to use. Inside of the `roles` directory, make another directory regarding the types of roles where in this case it'll be named as **Elastic-Ubuntu** and **Elastic-CentOS**. Since we need to download the ELK as by applying roles then the admin will input `Elastic_Search.yml`, `Kibana.yml`, `Logstash.yml` for CentOS Servers, and **PMS-Ubuntu** for Ubuntu Servers

Lastly, inside the respective directory of the role itself create another directory named as **tasks** which consist a `yml` file named as `main.yml`, to easily achieve this last step simply input the command `touch Elastic-CentOS/tasks/main.yml` and `touch Elastic-Ubuntu/tasks/main.yml`

```
penas@penas-VirtualBox:~/HOA-10.1-Elastic$ cd roles
penas@penas-VirtualBox:~/HOA-10.1-Elastic/roles$ mkdir Elastic-Ubuntu
penas@penas-VirtualBox:~/HOA-10.1-Elastic/roles$ cd Elastic-Ubuntu
penas@penas-VirtualBox:~/HOA-10.1-Elastic/roles/Elastic-Ubuntu$ mkdir tasks
penas@penas-VirtualBox:~/HOA-10.1-Elastic/roles/Elastic-Ubuntu$ cd ..
penas@penas-VirtualBox:~/HOA-10.1-Elastic/roles$ mkdir Elastic-CentOS
penas@penas-VirtualBox:~/HOA-10.1-Elastic/roles$ cd Elastic-CentOS
penas@penas-VirtualBox:~/HOA-10.1-Elastic/roles/Elastic-CentOS$ mkdir tasks
penas@penas-VirtualBox:~/HOA-10.1-Elastic/roles/Elastic-CentOS$ cd ..
penas@penas-VirtualBox:~/HOA-10.1-Elastic/roles$ touch Elastic-Ubuntu/tasks/main.yml
penas@penas-VirtualBox:~/HOA-10.1-Elastic/roles$ touch Elastic-CentOS/tasks/main.yml
penas@penas-VirtualBox:~/HOA-10.1-Elastic/roles$ tree
.
├── Elastic-CentOS
│   └── tasks
│       └── main.yml
└── Elastic-Ubuntu
    └── tasks
        └── main.yml

4 directories, 2 files
```

Figure 1.6. Creating Directories for Roles specifically **Elastic-CentOS** and **Elastic-Ubuntu** along with their initial configuration file

6. Change the name of the hostname in the copied inventory and global_playbook.yml files from **PMS-Ubuntu/CentOS** to **Elastic-Ubuntu/CentOS**. These will be only the changes that will be made on the main playbook

```
GNU nano 6.2
[Elastic-Ubuntu]
server_1

[Elastic-CentOS]
Cent-OS

---
- hosts: all
  become: true
  pre_tasks:
    - name: Update and upgrade remote in Ubuntu servers
      apt:
        update_cache: yes
        upgrade: 'yes'
        when: ansible_distribution == "Ubuntu"
    - name: Installing dnf and epel-release
      yum:
        name:
          - epel-release
          - dnf
        when: ansible_distribution == "CentOS"
    - name: Update and upgrade remote CentOS server
      dnf:
        update_cache: yes
        name: "*"
        state: latest
        when: ansible_distribution == "CentOS"
    - name: Dpkg fixing in Ubuntu Servers
      shell: |
        dpkg --configure -a
        when: ansible_distribution == "Ubuntu"
- hosts: Elastic-Ubuntu
  become: true
  roles:
    - Elastic-Ubuntu
- hosts: Elastic-CentOS
  become: true
  roles:
    - Elastic-CentOS
```

Figure 1.7. Changing the name of the hostname in inventory and global_playbook.yml

7. Initiate an initial run of the ansible playbook to check if the Servers and Local Host was working and connected properly.

```
penas@penas-VirtualBox:~/HWA-10.1-Elastic$ ansible-playbook --ask-become-pass global_playbook.yml
BECOME password:
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [Cent-OS]
ok: [server_1]

TASK [Update and upgrade remote in Ubuntu servers] *****
skipping: [Cent-OS]
ok: [server_1]

TASK [Installing dnf and epel-release] *****
skipping: [server_1]
ok: [Cent-OS]

TASK [Update and upgrade remote CentOS server] *****
skipping: [server_1]
ok: [Cent-OS]

TASK [Dpkg fixing in Ubuntu Servers] *****
skipping: [Cent-OS]
changed: [server_1]

PLAY [Elastic-Ubuntu] *****

TASK [Gathering Facts] *****
ok: [server_1]

PLAY [Elastic-CentOS] *****

TASK [Gathering Facts] *****
ok: [Cent-OS]

PLAY RECAP *****
Cent-OS                : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
server_1               : ok=4    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

Figure 1.8. Executing the initial **global_playbook.yml** which is successful without any errors

8. After creating the subdirectories in Roles Directories, proceed to input the commands on your main.yml of Elastic-Ubuntu and Elastic-CentOS

Ubuntu:

Pre-requisites:

```
#Elastic-Ubuntu
- name: Install ELK Dependencies
  apt:
    name:
      - openjdk-11-jdk
      - apt-transport-https
      - curl
      - gpgv
      - gpgsm
      - gnupg-l10n
      - gnupg
      - dirmngr
    state: latest

- name: Get PGP Key
  apt_key:
    url: https://artifacts.elastic.co/GPG-KEY-elasticsearch
    state: present
```

Installing Elastic_Search:

```

#Installation of Elastic Search
- name: Install Elastic_Search packages from the Source
  apt_repository:
    repo: deb https://artifacts.elastic.co/packages/7.x/apt stable main
    state: present

- name: Installing Elastic_Search
  apt:
    name: elasticsearch
    state: latest
    update_cache: yes

- name: Configuring the Cluster name of Elastic_Search
  lineinfile:
    dest: /etc/elasticsearch/elasticsearch.yml
    line: "cluster.name: demo-elk"
    state: present

- name: Configuring the Cluster Descriptive name Elastic_Search
  lineinfile:
    dest: /etc/elasticsearch/elasticsearch.yml
    line: "node.name: elk-1"
    state: present

- name: Adding the network.host in Elastic_Search
  lineinfile:
    dest: /etc/elasticsearch/elasticsearch.yml
    line: "network.host: 0.0.0.0"
    state: present

- name: Adding http.port in Elastic_Search
  lineinfile:
    dest: /etc/elasticsearch/elasticsearch.yml
    line: "http.port: 9200"
    state: present

- name: Adding discovery.type in Elastic_Search
  lineinfile:
    dest: /etc/elasticsearch/elasticsearch.yml
    line: "discovery.type: single-node"
    state: present

- name: Creating an empty file for startup-timeout.conf 1 of 2
  file:
    path: "/etc/systemd/system/elasticsearch.service.d"
    state: directory

- name: Creating an empty file for startup-timeout.conf 2 of 2
  file:
    path: "/etc/systemd/system/elasticsearch.service.d/startup-timeout.conf"
    state: touch

- name: Preventing the systemd service start operation from timing out
  copy:
    dest: "/etc/systemd/system/elasticsearch.service.d/startup-timeout.conf"
    content: |
      [Service]
      TimeoutStartSec=3min

- name: Run daemon-reload for Elastic_Search
  systemd: daemon_reload=yes

- name: Disabling masking of Elastic_Search
  systemd:
    name: elasticsearch
    enabled: yes
    masked: no

- name: Confirmation of Elastic_Search is enabled
  tags: serviceon
  service:
    name: elasticsearch
    state: restarted
    enabled: true

- name: Start Elastic_Search service
  shell: systemctl start elasticsearch

```

Installing Kibana:

```

#Installation of Kibana
- name: Installing Kibana
  apt:
    name: kibana
    state: latest
    update_cache: yes

- name: Adding server.port in Kibana
  lineinfile:
    dest: /etc/kibana/kibana.yml
    line: "server.port: 5601"
    state: present

- name: Adding server.host in Kibana
  lineinfile:
    dest: /etc/kibana/kibana.yml
    line: 'server.host: "0.0.0.0"'
    state: present

- name: Adding server.name in Kibana
  lineinfile:
    dest: /etc/kibana/kibana.yml
    line: 'server.name: "demo-kibana"'
    state: present

- name: Adding elasticsearch.hosts in Kibana
  lineinfile:
    dest: /etc/kibana/kibana.yml
    line: 'elasticsearch.hosts: ["http://0.0.0.0:9200"]'
    state: present

- name: Run daemon-reload for Kibana
  systemd: daemon_reload=yes

- name: Confirmation of Kibana is enabled
  tags: serviceon
  service:
    name: kibana
    state: restarted
    enabled: true

- name: Start Kibana
  shell: systemctl start kibana

```

Installing LogStash:

```

#Installation of LogStash
- name: Installing LogStash
  apt:
    name: logstash
    state: latest
    update_cache: yes

- name: Run daemon-reload for LogStash
  systemd: daemon_reload=yes

- name: Enable service LogStash
  systemd:
    name: logstash
    enabled: yes

- name: Confirmation of LogStash is enabled
  tags: serviceon
  service:
    name: logstash
    state: restarted
    enabled: true

```

Figure 1.9-13. The Playbook of **Elastic-Ubuntu** where it installs the ELK Applications
CentOS:

Pre-requisites:

```
Elastic-CentOS
- name: Install ELK Dependencies
  yum:
    name:
      - java-11-openjdk
      - curl
      - gnupg
    state: latest
```

Installing Elastic_Search:

```
#Installing Elastic_Search
- name: Installing Elastic_Search RPM key
  rpm_key:
    key: https://artifacts.elastic.co/GPG-KEY-elasticsearch
    state: present
    become: true

- name: Installing Elastic_Search 7.x RPM repositories
  yum_repository:
    name: Elastic_7.X_repo
    baseurl: https://artifacts.elastic.co/packages/7.x/yum
    gpgcheck: true
    gpgkey: https://artifacts.elastic.co/GPG-KEY-elasticsearch
    description: Elastic 7.X Repo
    become: true

- name: Installing Elastic_Search
  yum:
    name: elasticsearch
    state: latest
    update_cache: yes

- name: Configuring the Cluster name of Elastic_Search
  lineinfile:
    dest: /etc/elasticsearch/elasticsearch.yml
    line: "cluster.name: demo-elk"
    state: present

- name: Configuring the Cluster Descriptibe name Elastic_Search
  lineinfile:
    dest: /etc/elasticsearch/elasticsearch.yml
    line: "node.name: elk-1"
    state: present

- name: Adding the network.host in Elastic_Search
  lineinfile:
    dest: /etc/elasticsearch/elasticsearch.yml
    line: "network.host: 0.0.0.0"
    state: present

- name: Adding http.port in Elastic_Search
  lineinfile:
    dest: /etc/elasticsearch/elasticsearch.yml
    line: "http.port: 9200"
    state: present
```

```

- name: Creating an empty file for startup-timeout.conf 1 of 2
  file:
    path: "/etc/systemd/system/elasticsearch.service.d"
    state: directory

- name: Creating an empty file for startup-timeout.conf 2 of 2
  file:
    path: "/etc/systemd/system/elasticsearch.service.d/startup-timeout.conf"
    state: touch

- name: Preventing the systemd service start operation from timing out
  copy:
    dest: "/etc/systemd/system/elasticsearch.service.d/startup-timeout.conf"
    content: |
      [Service]
      TimeoutStartSec=3min

- name: Run daemon-reload for Elastic_Search
  systemd: daemon_reload=yes

- name: Disabling masking of Elastic_Search
  systemd:
    name: elasticsearch
    enabled: yes
    masked: no

- name: Confirmation of Elastic_Search is enabled
  tags: serviceon
  service:
    name: elasticsearch
    state: started
    enabled: true

- name: Start Elastic_Search
  shell: systemctl start elasticsearch

```

Installing Kibana:

```

#Installing Kibana
- name: Installing Kibana
  yum:
    name: kibana
    state: latest
    update_cache: yes

- name: Adding server.port in Kibana
  lineinfile:
    dest: /etc/kibana/kibana.yml
    line: "server.port: 5601"
    state: present

- name: Adding server.host in Kibana
  lineinfile:
    dest: /etc/kibana/kibana.yml
    line: 'server.host: "0.0.0.0"'
    state: present

- name: Adding server.name in Kibana
  lineinfile:
    dest: /etc/kibana/kibana.yml
    line: 'server.name: "demo-kibana"'
    state: present

- name: Adding elasticsearch.hosts in Kibana
  lineinfile:
    dest: /etc/kibana/kibana.yml
    line: 'elasticsearch.hosts: ["http://0.0.0.0:9200"]'
    state: present

- name: Run daemon-reload for Kibana
  systemd: daemon_reload=yes

- name: Enable service Kibana
  tags: serviceon
  service:
    name: kibana
    state: restarted
    enabled: true

- name: Start Kibana
  shell: systemctl start kibana

```

Installing LogStash:

```
#Installing Logstash
- name: Install LogStash
  yum:
    name: logstash
    state: latest
    update_cache: yes

- name: Run daemon-reload for LogStash
  systemd: daemon_reload=yes

- name: Enable service LogStash
  systemd:
    name: logstash
    enabled: yes

- name: Confirmation of LogStash is enabled
  tags: serviceon
  service:
    name: logstash
    state: restarted
    enabled: true
```

Figure 1.14-18. The Playbook of Elastic-Ubuntu where it installs the ELK Applications

9. When the playbooks are correctly configured, run the command **ansible-playbook --ask-become-pass global_playbook.yml** which executes the main playbook of this folder.

global_playbook.yml

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

TASK [Gathering Facts] *****
ok: [Cent-OS]
ok: [server_1]

TASK [Update and upgrade remote in Ubuntu servers] *****
skipping: [Cent-OS]
ok: [server_1]

TASK [Installing dnf and epel-release] *****
skipping: [server_1]
ok: [Cent-OS]

TASK [Update and upgrade remote CentOS server] *****
skipping: [server_1]
changed: [Cent-OS]

TASK [Dpkg fixing in Ubuntu Servers] *****
skipping: [Cent-OS]
changed: [server_1]
```

Elastic-Ubuntu-main.yml

```

PLAY [Elastic-Ubuntu] *****

TASK [Gathering Facts] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Install ELK Dependencies] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Get PGP Key] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Install Elastic_Search packages from the Source] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Installing Elastic_Search] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Configuring the Cluster name of Elastic_Search] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Configuring the Cluster Descriptibe name Elastic_Search] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Adding the network.host in Elastic_Search] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Adding http.port in Elastic_Search] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Adding discovery.type in Elastic_Search] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Creating an empty file for startup-timeout.conf 1 of 2] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Creating an empty file for startup-timeout.conf 2 of 2] *****
changed: [server_1]

TASK [Elastic-Ubuntu : Preventing the systemd service start operation from timing out] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Run daemon-reload for Elastic_Search] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Disabling masking of Elastic_Search] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Confirmation of Elastic_Search is enabled] *****
changed: [server_1]

TASK [Elastic-Ubuntu : Start Elastic_Search service] *****
changed: [server_1]

TASK [Elastic-Ubuntu : Installing Kibana] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Adding server.port in Kibana] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Adding server.host in Kibana] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Adding server.name in Kibana] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Adding elasticsearch.hosts in Kibana] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Run daemon-reload for Kibana] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Confirmation of Kibana is enabled] *****
changed: [server_1]

TASK [Elastic-Ubuntu : Start Kibana] *****
changed: [server_1]

TASK [Elastic-Ubuntu : Installing LogStash] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Run daemon-reload for LogStash] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Enable service LogStash] *****
ok: [server_1]

TASK [Elastic-Ubuntu : Confirmation of LogStash is enabled] *****
changed: [server_1]

```

Elastic-CentOS-main.yml

```
PLAY [Elastic-CentOS] *****
TASK [Gathering Facts] *****
ok: [Cent-05]

TASK [Elastic-CentOS : Install ELK Dependencies] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Installing Elastic_Search RPM key] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Installing Elastic_Search 7.x RPM repositories] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Installing Elastic_Search] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Configuring the Cluster name of Elastic_Search] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Configuring the Cluster Descriptibe name Elastic_Search] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Adding the network.host in Elastic_Search] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Adding http.port in Elastic_Search] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Adding discovery.type in Elastic_Search] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Creating an empty file for startup-timeout.conf 1 of 2] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Creating an empty file for startup-timeout.conf 2 of 2] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Preventing the systemd service start operation from tining out] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Run daemon-reload for Elastic_Search] *****
ok: [Cent-05]

TASK [Elastic-CentOS : Disabling masking of Elastic_Search] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Confirmation of Elastic_Search is enabled] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Start Elastic_Search] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Installing Kibana] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Adding server.port in Kibana] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Adding server.host in Kibana] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Adding server.name in Kibana] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Adding elasticsearch.hosts in Kibana] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Run daemon-reload for Kibana] *****
ok: [Cent-05]

TASK [Elastic-CentOS : Enable service Kibana] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Start Kibana] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Install LogStash] *****
changed: [Cent-05]

TASK [Elastic-CentOS : Run daemon-reload for LogStash] *****
ok: [Cent-05]

TASK [Elastic-CentOS : Enable service LogStash] *****
changed: [Cent-05]

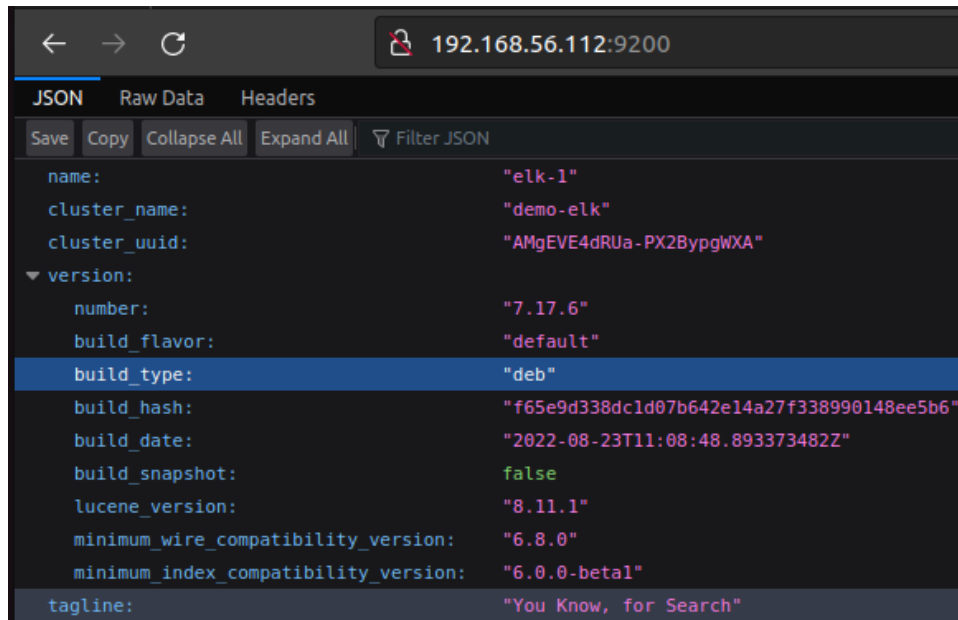
TASK [Elastic-CentOS : Confirmation of LogStash is enabled] *****
changed: [Cent-05]

PLAY RECAP *****
Cent-05 : ok=32 changed=26 unreachable=0 failed=0 skipped=2 rescued=0 ignored=0
server_1 : ok=32 changed=7 unreachable=0 failed=0 skipped=2 rescued=0 ignored=0
```

10. To Verify if the ELK was fully installed within the Servers, Open a Brower and insert the respective IP Address of the server along with the port numbers of the specific Application whereas in Elastic_Search: 9200, Kibana: 5601, LocalStash: 22

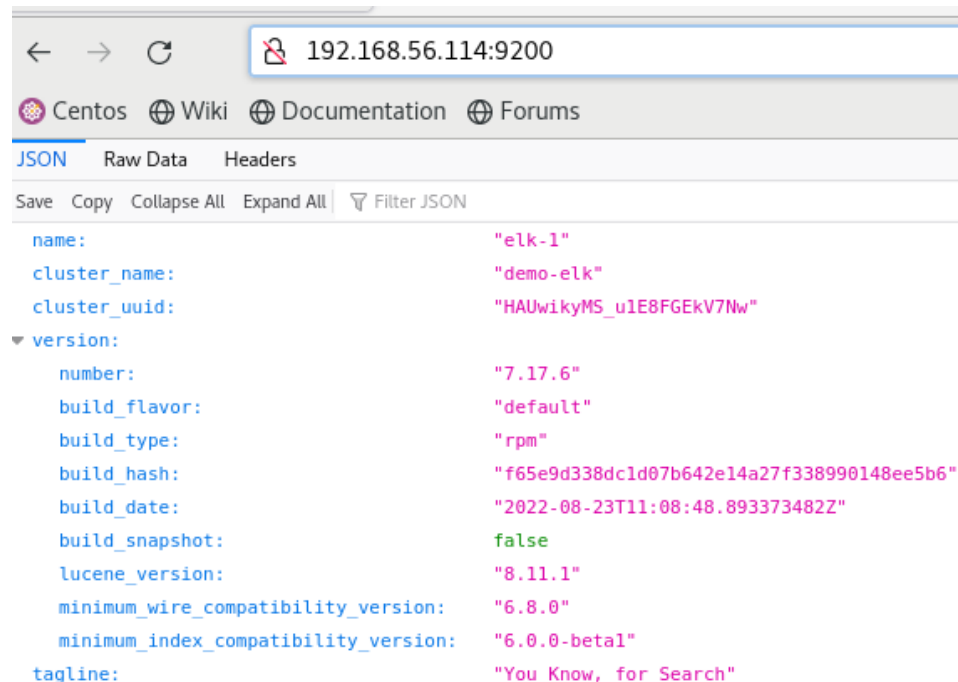
Elastic_Search:

Ubuntu:



```
{
  "name": "elk-1",
  "cluster_name": "demo-elk",
  "cluster_uuid": "AMgEVE4dRUa-PX2BypgwXA",
  "version": {
    "number": "7.17.6",
    "build_flavor": "default",
    "build_type": "deb",
    "build_hash": "f65e9d338dcd07b642e14a27f338990148ee5b6",
    "build_date": "2022-08-23T11:08:48.893373482Z",
    "build_snapshot": false,
    "lucene_version": "8.11.1",
    "minimum_wire_compatibility_version": "6.8.0",
    "minimum_index_compatibility_version": "6.0.0-beta1"
  },
  "tagline": "You Know, for Search"
}
```

CentOS:



```
{
  "name": "elk-1",
  "cluster_name": "demo-elk",
  "cluster_uuid": "HAUwikyMS_u1E8FGEkV7Nw",
  "version": {
    "number": "7.17.6",
    "build_flavor": "default",
    "build_type": "rpm",
    "build_hash": "f65e9d338dcd07b642e14a27f338990148ee5b6",
    "build_date": "2022-08-23T11:08:48.893373482Z",
    "build_snapshot": false,
    "lucene_version": "8.11.1",
    "minimum_wire_compatibility_version": "6.8.0",
    "minimum_index_compatibility_version": "6.0.0-beta1"
  },
  "tagline": "You Know, for Search"
}
```

Kibana:

Ubuntu:

CentOS:

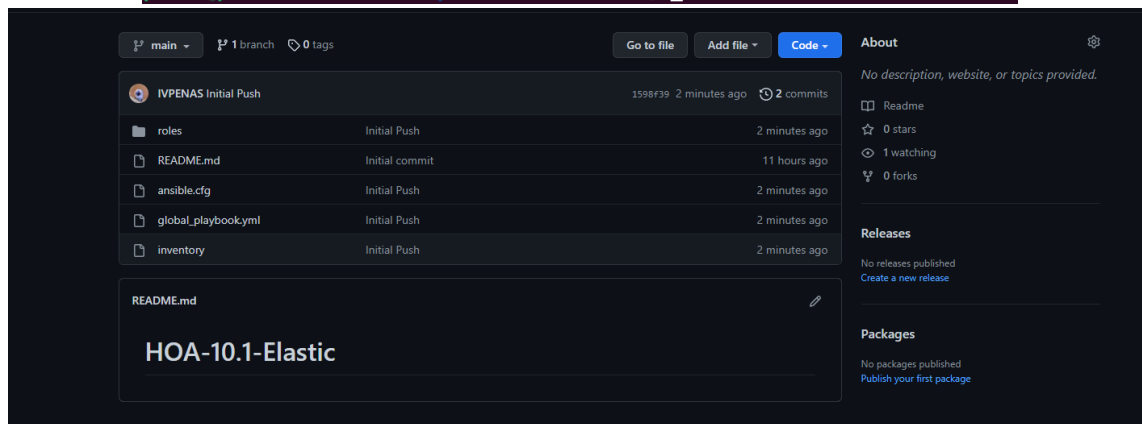
LocalStash:

Ubuntu:

CentOS:

11. When the Verification was done, fully sync the cloned folder to Github using the command '**git add ***' which adds a changed directory, '**git commit -m "ANYMESSG"**' Commits the Changes are made within the repository folder, and '**git push**' to sync and save the folder to the Git Repository.

```
penas@penas-VirtualBox:~/HOA-10.1-Elastic$ git add *
penas@penas-VirtualBox:~/HOA-10.1-Elastic$ git commit -m "Initial Push"
[main 1598f39] Initial Push
 7 files changed, 378 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 global_playbook.yml
 create mode 100644 inventory
 create mode 100644 roles/Elastic-CentOS/tasks/.main.yml.swp
 create mode 100644 roles/Elastic-CentOS/tasks/main.yml
 create mode 100644 roles/Elastic-Ubuntu/tasks/.main.yml.swp
 create mode 100644 roles/Elastic-Ubuntu/tasks/main.yml
penas@penas-VirtualBox:~/HOA-10.1-Elastic$ git push
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Compressing objects: 100% (12/12), done.
Writing objects: 100% (14/14), 2.55 KiB | 2.55 MiB/s, done.
Total 14 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To github.com:IVPENAS/HOA-10.1-Elastic.git
 b529b9f..1598f39  main -> main
penas@penas-VirtualBox:~/HOA-10.1-Elastic$
```



GitHub Link: <https://github.com/IVPENAS/HOA-10.1-Elastic.git>

Reflections:

Answer the following:

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

In short, a log monitoring tool is a type of application that views the logs of a Network or System in order to determine the future events that will occur in each system which are mostly used by Developers and System Administrator as those logs were recorded to analyze its data collected to further improve and maintain the network as a whole. They can operate on the network components of a system both software and hardware, that allows for the professional to be notified when the

network itself was compromised or needed an upgrade. Log Monitoring Tools offer important features to aid our System Administrators, Developers, Information Technology whereas [1] The **Log formats are organized** to determine which system requires need of assistance more accurately resulting in troubleshooting faster and output great results. [2] The tool **stores and monitors the logs in real-time** which are crucial when your network holds sensitive datas that are needed to protect from threat actors, [3] It **alerts the professionals** when the system was compromised or needed some upgrades, [4] It **enhances the system performance**, [5] Since the Professionals handle multiple Systems, using Log Monitoring tool it allows for the Admin to **monitor at once** which **saves time**.

There are types of Log Monitoring Tool in which it target specific parts of a system, [1] **Network Monitoring** which specialized in firewalls or routers specifically which is the backbone of a Company, [2] **Application Monitoring**, specifies the logs and activities of a certain application, [3] **Database Monitoring**, is a platform that most of the System Administrator uses with the help of Database Application like MySQL and more, in this type of Log Monitoring Tool it enhances the Admin's troubleshooting on database errors, and [4] **Cloud Monitoring**, that i think is the most recent and in demand in our Technology which logs the activities from a Cloud Platform.

Conclusions:

Due to the lack of time, the student was not able to fully connect the Kibana and Localstash due to some configuration errors within the main.yml and lack of time but the student was able to accomplish and connect Elasticsearch to the servers by verifying it using the Web Browser inputting the format in the search bar by **[IP]:9200**. Putting that aside, the student was also able to learn more about the Log Monitoring tools like Kibana which was used in previous courses as these type of tools aids Computer Professionals to monitor their Network and Systems in a specific type of Log Monitoring Tool including [1] Network Monitoring, [2] Application Monitoring, [3] Database Monitoring, and [4] Cloud Monitoring at once in real time without wasting necessary time, it also notifies whenever the system was compromised or needed some updates.