

<b>Name:</b> Zamora, Angelo E.	<b>Date Performed:</b> 11 - 01 - 2024
<b>Course/Section:</b> BSCPE - CpE31S2	<b>Date Submitted:</b> 11 - 01 - 2024
<b>Instructor:</b> Engr. Robin Valenzuela	<b>Semester and SY:</b> 1st Semester 2024 - 2025
<b>Activity 10: Install, Configure, and Manage Log Monitoring tools</b>	
<b>1. Objectives</b>	
Create and design a workflow that installs, configure and manage enterprise log monitoring tools using Ansible as an Infrastructure as Code (IaC) tool.	
<b>2. Discussion</b>	
<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"> <li>• Monitor the log files generated by servers, applications, or networks</li> <li>• Alert users when important events are detected</li> <li>• Provide reporting capabilities for log files</li> </ul> <p><b>Elastic Stack</b></p> <p>ELK suite stands for Elasticsearch, Kibana, Beats, and Logstash (also known as the ELK Stack). Source: <a href="https://www.elastic.co/elastic-stack">https://www.elastic.co/elastic-stack</a></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

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)

Step 1: Create a Repository

```
zamora_admin@workstation:~/TIP_HOA-10.1_ZAMORA_Angelo$ ls  
ansible.cfg  inventory  README.md
```

Create a repo for the HOA 10.1 and we will build the needed files for the task.

Here are the contents of the foundational files for the repo the inventory and the ansible.cfg to setup the repo (based on your own manage Nodes setup):

## ansible.cfg

```
zamura@workstation: ~/TIP_HOA-10.1_
GNU nano 6.2 ansible.cfg
[defaults]
inventory = /home/zamura/TIP_HOA-10.1_ZAMORA_Angelo/inventory
remote_user = zamora
host_key_checking = True
```

## inventory

```
GNU nano 6.2
[elasticsearch]
192.168.56.104 ansible_user=azamura
[kibana]
192.168.56.102
[logstash]
192.168.56.103
```

**192.168.56.102** - Server 1

**192.168.56.103** - Server 2

**192.168.56.104** - CentOS Node 1

## Step 2: Create the directory “roles”

Create the directory roles and create three directories inside the “roles” directory. The name for the three is elasticsearch, kibana, and logstash as these will set up the roles approach in ansible playbook. And under each roles create a directory called task to store the main.yml and j2 config file.

```
Processing triggers for libc-bin (2.14-0ubuntu1) ...
zamora@workstation:~/TIP_HOA-10.1_ZAMORA_Angelo$ tree roles
roles
├── elasticsearch
│   └── tasks
│       ├── elasticsearch.yml.j2
│       └── main.yml
├── kibana
│   └── tasks
│       ├── kibana.yml.j2
│       └── main.yml
└── logstash
    └── tasks
        ├── logstash.conf.j2
        └── main.yml

6 directories, 6 files
```

## Step 3: Create a playbook for each roles' tasks folder and config files

Here the contents of each playbook and config files:

elasticsearch (main.yml)

```
Unset
---
- name: Install Java
  yum:
    name: java-11-openjdk
    state: present
    when: ansible_distribution == "CentOS"

- name: Install EPEL repository
  yum:
    name: epel-release
    state: latest
    when: ansible_distribution == "CentOS"

- name: Install Elastic Search 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
    enabled: yes
    when: ansible_distribution == "CentOS"

- name: Install Elastic Search
  dnf:
    name: elasticsearch
    state: present
    when: ansible_distribution == "CentOS"

- name: Configure Elastic Search
  template:
    src: elasticsearch.yml.j2
    dest: /etc/elasticsearch/elasticsearch.yml
    when: ansible_distribution == "CentOS"

- name: Start Elastic Search
  service:
    name: elasticsearch
    state: restarted
    enabled: yes
    when: ansible_distribution == "CentOS"

- name: Allow port 9200 through the firewall
  command: firewall-cmd --zone=public --add-port=9200/tcp --permanent
  register: firewall_result
  ignore_errors: true

```

## elasticsearch(elasticsearch.yml.j2)

Unset

# Elasticsearch Configuration

```

cluster.name: my-cluster
node.name: dev-node-1
network.host: 0.0.0.0
http.port: 9200
discovery.type: single-node
path.data: /var/lib/elasticsearch
path.logs: /var/log/elasticsearch
bootstrap.memory_lock: true

```

## kibana(main.yml)

Unset

```
---
- name: Add GPG key for Elastic APT repository
  tags: kibana
  apt_key:
    url: https://artifacts.elastic.co/GPG-KEY-elasticsearch
    state: present
  when: ansible_distribution == "Ubuntu"

- name: Add Kibana APT repository
  tags: kibana
  apt_repository:
    repo: "deb https://artifacts.elastic.co/packages/7.x/apt stable main"
    state: present
  when: ansible_distribution == "Ubuntu"

- name: Install specific version of Kibana
  tags: kibana
  apt:
    name: kibana
    state: present
  when: ansible_distribution == "Ubuntu"

- name: Create directory for Kibana systemd override
  tags: kibana
  file:
    path: /etc/systemd/system/kibana.service.d
    state: directory
    mode: '0755'
    owner: root
    group: root
  when: ansible_distribution == "Ubuntu"

- name: Check if the directory was created
  tags: kibana
  stat:
    path: /etc/systemd/system/kibana.service.d
  register: kibana_override_dir

- debug:
  msg: "Directory exists: {{ kibana_override_dir.stat.exists }}"

- name: Create Kibana service override configuration
```

```

tags: kibana
file:
  path: /etc/systemd/system/kibana.service.d/override.conf
  state: touch # Ensures the file exists
  owner: root
  group: root
  mode: '0644'
when: ansible_distribution == "Ubuntu"

- name: Configure Kibana (Setting OpenSSL Legacy Provider)
  tags: kibana
  blockinfile:
    path: /etc/systemd/system/kibana.service.d/override.conf
    block: |
      [Service]
      Environment=NODE_OPTIONS=--openssl-legacy-provider
    owner: root
    group: root
    mode: '0644'
  when: ansible_distribution == "Ubuntu"

- name: Configure Kibana
  tags: kibana
  template:
    src: kibana.yml.j2
    dest: /etc/kibana/kibana.yml
  when: ansible_distribution == "Ubuntu"

- name: Reload systemd
  tags: kibana
  command: systemctl daemon-reload
  when: ansible_distribution == "Ubuntu"

- name: Enable Kibana service
  tags: kibana
  service:
    name: kibana
    state: restarted
  become: yes
  when: ansible_distribution == "Ubuntu"

```

## kibana(kibana.yml.j2)

Unset

# Kibana Configuration

# Set the port that the Kibana server will listen on  
server.port: 5601

# Specify the host address that the Kibana server will bind to  
server.host: "192.168.56.102"

# Set the public base URL for Kibana  
server.publicBaseUrl: "http://192.168.56.102:5601"

# Elasticsearch server URL  
elasticsearch.hosts: ["http://192.168.56.104:9200"]

## logstash(main.yml)

Unset

- name: Install dependencies

tags: logstash

apt:

name: gnupg

state: present

update\_cache: yes

become: yes

- name: Add Elastic APT repository key

tags: logstash

apt\_key:

url: https://artifacts.elastic.co/GPG-KEY-elasticsearch

state: present

- name: Add Elastic APT repository

tags: logstash

apt\_repository:

repo: "deb https://artifacts.elastic.co/packages/7.x/apt stable main"

state: present

- name: Install Logstash

tags: logstash

apt:

name: logstash



```

    state: present

- name: Start and Enable Logstash service
  tags: logstash
  systemd:
    name: logstash
    enabled: yes
    state: started

```

logstash(logstash.conf.j2)

```

Unset
nput {
  beats {
    port => 5044
  }
}

filter {
  # Add any filters here
}

output {
  elasticsearch {
    hosts => ["http://192.168.56.104:9200"]
    index => "logstash-%{+YYYY.MM.dd}"
  }
}

```

**Explanation:** In each roles' playbooks, we've acquired the access of the repo through the elastic stack website and each of those will be installed through their package name. After it was installed it will be configured through the .j2 files we've encoded and will allow the necessary port needed in order to run it. After that it will start the service of each tool in our VM's.

Step 4: Make a elk.yml playbook in the repo and run it

elk.yml

```
zamor@workstation: ~/TIP_HOA-10.1_ZAMORA_Angelo
GNU nano 6.2 elk.yml
---
- hosts: all
  become: true
  pre_tasks:
    - name: update repository index / install Updates (CentOS)
      tags: always
      dnf:
        update_cache: yes
        changed_when: false
        when: ansible_distribution == "CentOS"
    - name: update repository index / install Updates (Ubuntu)
      tags: always
      apt:
        update_cache: yes
        changed_when: false
        when: ansible_distribution == "Ubuntu"
- hosts: elasticsearch
  become: true
  roles:
    - elasticsearch
- hosts: kibana
  become: true
  roles:
    - kibana
- hosts: logstash
  become: true
  roles:
    - logstash
```

Running the playbook:

```
zamor@workstation:~/TIP_HOA-10.1_ZAMORA_Angelo$ ansible-playbook --ask-become-pass elk.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.102]
ok: [192.168.56.103]
ok: [192.168.56.104]

TASK [update repository index / install Updates (CentOS)] *****
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [update repository index / install Updates (Ubuntu)] *****
skipping: [192.168.56.104]
ok: [192.168.56.102]
ok: [192.168.56.103]
```

```
PLAY [elasticsearch] *****

TASK [Gathering Facts] *****
ok: [192.168.56.104]

TASK [elasticsearch : Install Java] *****
ok: [192.168.56.104]

TASK [elasticsearch : Install EPEL repository] *****
ok: [192.168.56.104]

TASK [elasticsearch : Install Elastic Search YUM repository] *****
ok: [192.168.56.104]

TASK [elasticsearch : Install Elastic Search] *****
ok: [192.168.56.104]

TASK [elasticsearch : Configure Elastic Search] *****
ok: [192.168.56.104]

TASK [elasticsearch : Start Elastic Search] *****
changed: [192.168.56.104]

TASK [elasticsearch : Allow port 9200 through the firewall] *****
changed: [192.168.56.104]
```

```
PLAY [kibana] *****

TASK [Gathering Facts] *****
ok: [192.168.56.102]

TASK [kibana : Add GPG key for Elastic APT repository] *****
ok: [192.168.56.102]

TASK [kibana : Add Kibana APT repository] *****
ok: [192.168.56.102]

TASK [kibana : Install specific version of Kibana] *****
ok: [192.168.56.102]

TASK [kibana : Create directory for Kibana systemd override] *****
ok: [192.168.56.102]

TASK [kibana : Check if the directory was created] *****
ok: [192.168.56.102]

TASK [kibana : debug] *****
ok: [192.168.56.102] => {
  "msg": "Directory exists: True"
}

TASK [kibana : Create Kibana service override configuration] *****
changed: [192.168.56.102]

TASK [kibana : Configure Kibana (Setting OpenSSL Legacy Provider)] *
ok: [192.168.56.102]

TASK [kibana : Configure Kibana] *****
ok: [192.168.56.102]

TASK [kibana : Reload systemd] *****
changed: [192.168.56.102]
```

```
TASK [kibana : Enable Kibana service]
changed: [192.168.56.102]
```

```
PLAY [logstash] *****
TASK [Gathering Facts] *****
ok: [192.168.56.103]
TASK [logstash : Install dependencies] *****
ok: [192.168.56.103]
TASK [logstash : Add Elastic APT repository key] *****
ok: [192.168.56.103]
TASK [logstash : Add Elastic APT repository] *****
ok: [192.168.56.103]
TASK [logstash : Install Logstash] *****
ok: [192.168.56.103]
TASK [logstash : Start and Enable Logstash service] *****
ok: [192.168.56.103]
PLAY RECAP *****
192.168.56.102      : ok=14   changed=3   unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.103      : ok=8     changed=0   unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.104      : ok=10   changed=2   unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

## Step 5: Checking if installed

To check if it's installed we will prompt ***systemctl status "service name"*** in Ubuntu and CentOS to check if the service is active. Another way to check is by accessing the tool via browser, all you need to do is type ***"ip address": "port number"*** into your browser.

Elasticsearch:

The screenshot shows a web browser window displaying the Elasticsearch configuration page for the cluster 'dev-node-1'. The configuration includes details about the cluster name, UUID, version (7.17.25), build type (rpm), and various compatibility versions. The 'tagline' is 'You Know, for Search'.

Overlaid on the browser window is a terminal window showing the output of the command `systemctl status elasticsearch`. The output indicates that the `elasticsearch.service` is loaded and active (running) since Friday, 2024-11-01 at 15:52:39 PST. It also shows the main PID (4546) and the tasks (75) for the service.

## Kibana:

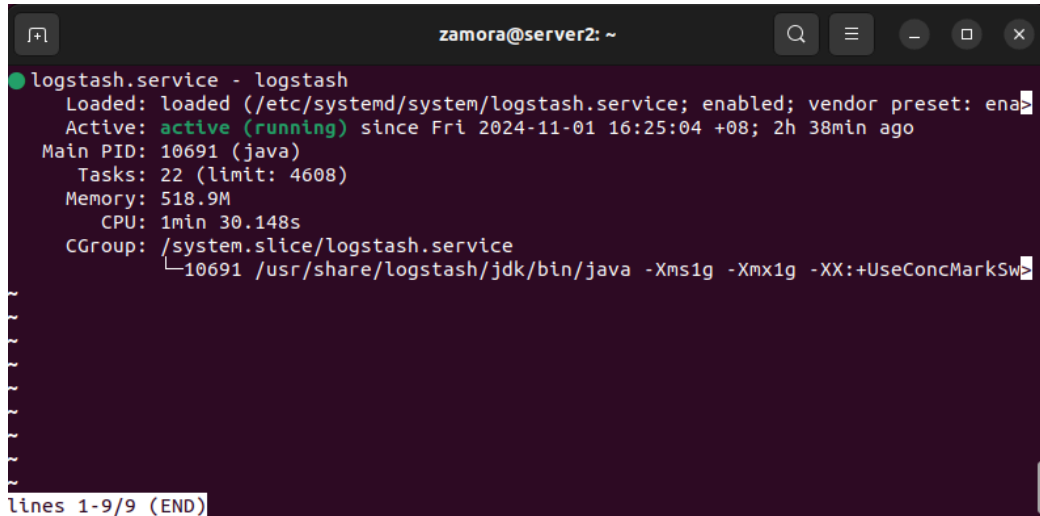
The image shows the Kibana web interface in a browser window. The address bar displays the URL `192.168.56.102:5601/app/home#`. The main content area features a "Welcome to Elastic" message with a colorful graphic and a "Start by adding integrations" section. Below this, there are four colored cards representing different Elastic products: Enterprise Search (yellow), Observability (pink), Security (teal), and Analytics (blue). Each card includes a brief description of its capabilities.

Overlaid on the top right is a terminal window titled `zamora@server1: ~`. It shows the status of the `kibana.service`:

```
kibana.service - Kibana
Loaded: loaded (/etc/systemd/system/kibana.service; vendor preset: enabled)
Drop-In: /etc/systemd/system/kibana.service.d
         override.conf
Active: active (running) since Fri 2024-10-11 14:30:11 CEST; 1min 10s ago
Docs: https://www.elastic.co
Main PID: 7804 (node)
Tasks: 11 (limit: 4608)
Memory: 133.4M
CPU: 6.459s
lines 1-10/12 61%
```

The Kibana interface includes a top navigation bar with the Elastic logo, a search bar labeled "Search Elastic", and a "Home" button. The main heading is "Welcome home".

logstash:

A terminal window titled 'zamora@server2: ~' displays the status of the 'logstash.service'. The output shows it is loaded, enabled, and active (running) since Fri 2024-11-01 16:25:04 +08; 2h 38min ago. It lists the main PID as 10691 (java), with 22 tasks and a limit of 4608. Memory usage is 518.9M and CPU time is 1min 30.148s. The CGroup is /system.slice/logstash.service, and the command line for PID 10691 is /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSwt.

```
logstash.service - logstash
Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: ena
Active: active (running) since Fri 2024-11-01 16:25:04 +08; 2h 38min ago
Main PID: 10691 (java)
Tasks: 22 (limit: 4608)
Memory: 518.9M
CPU: 1min 30.148s
CGroup: /system.slice/logstash.service
└─10691 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSwt
lines 1-9/9 (END)
```

After installing make sure to git push your local repo to your github to save the files.

GitHub Link: [https://github.com/GeloaceRT/TIP\\_HOA-10.1\\_ZAMORA\\_Angelo](https://github.com/GeloaceRT/TIP_HOA-10.1_ZAMORA_Angelo)

### Reflections:

Answer the following:

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

Numerous advantages come with using an availability monitoring solution for Ubuntu server administration, such as preemptive issue detection, real-time performance data, and timely notifications that reduce downtime. These technologies enhance capacity planning and resource usage by studying historical data, which also improves user experience. They offer a comprehensive solution for preserving server performance and dependability. They also facilitate compliance reporting and frequently connect easily with other administration tools.

### Conclusions:

I successfully installed and set up the Elastic Stack—which includes Kibana, Logstash, and Elasticsearch—in this exercise. A complete solution for centralized log and metric management is offered by this potent toolkit.

Large amounts of time-series data may be efficiently indexed and stored using Elasticsearch, the fundamental data store, which makes it perfect for troubleshooting and system performance analysis. With the help of Kibana, the visualization layer, users may produce dynamic dashboards and visualizations to extract insightful

information from the data they have gathered. Before being saved in Elasticsearch, log data is ingested, transformed, and enriched by Logstash, the data pipeline.

I've greatly increased my capacity to track and evaluate system performance, identify irregularities, and take proactive measures to resolve possible problems by utilizing the Elastic Stack. Eventually, more dependable and effective IT operations will result from this improved insight into system behavior. Elasticsearch.