NMAP Installation and Setup

Install nmap on your Ubuntu-based server:

```
sudo apt update
sudo apt install nmap
```

Create Script to run nmap and output the results

Using Shell Script a script (Preffered)

```
#!/bin/bash

NETWORK="172.23.37.31/24"
OUTPUT_FILE="/tmp/nmap_scan_results.json"
NODE_IP="172.23.37.31"  # Replace with the IP of one of your
nodes if different
NODE_PORT="30003"  # The NodePort assigned to your Logstash s
ervice

# Run nmap and format the output as valid JSON
nmap -sn -oX - $NETWORK | xmlstarlet sel -T -t -o '{"hosts":
[' -n \
  -m "//host" -o "{" \
  -o '"ip":"' -v "address/@addr" -o '",' \
  -o '"status":"' -v "status/@state" -o '",' \
  -o '"reason":"' -v "status/@reason" -o '",' \
  -o '"hostname":"' -v "hostnames/hostname/@name" -o '"' \
  -o "}," -n \
| sed '$ s/,$//' | sed '$ s/$/]}/' > $OUTPUT_FILE

# Send the results to Logstash
curl -H "Content-Type: application/json" -XPOST "http://${NOD
E_IP}:${NODE_PORT}" --data-binary @$OUTPUT_FILE
```

```
chmod +x nmap_scan.sh
```

Using Python

```
from scapy.all import ARP, Ether, srp
import json
import requests

NETWORK = "172.23.37.31/24"
NODE_IP = "172.23.37.31"  # Replace with the IP of one of you
r nodes if different
NODE_PORT = "30003"  # The NodePort assigned to your Logstash
service

def scan_network(network):
    arp = ARP(pdst=network)
    ether = Ether(dst="ff:ff:ff:ff:ff:ff")
    packet = ether/arp
    result = srp(packet, timeout=3, verbose=0)[0]
    devices = []
    for sent, received in result:
        devices.append({'ip': received.psrc, 'status': 'up',
'hostname': received.hwsrc})
    return devices

def send_to_logstash(data, node_ip, node_port):
    url = f"http://{node_ip}:{node_port}"
    headers = {"Content-Type": "application/json"}
    response = requests.post(url, headers=headers, json={"hos
ts": data})
    print(f"Status Code: {response.status_code}")
    print(f"Response: {response.text}")

if __name__ == "__main__":
```

```
    devices = scan_network(NETWORK)
    send_to_logstash(devices, NODE_IP, NODE_PORT)
```

Logstash Installation

Follow the Logstash installation from (logstash installation and setup page)

Modiy The Installation

1- Modify Pipeline-Config-py

- change the file name to nmap-pipeline-config and replace the existing code with the following code

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: nmap-pipeline-config
data:
  main_01_input.conf: |
    input {
       http {
        port => 8080
        codec => json
      }

    }
  main_02_filter.conf: |
    filter {
      split {
        field => "hosts"
      }

      mutate {
        rename => {
          "[hosts][ip]" => "ip_address"
          "[hosts][status]" => "host_status"
```

```
          "[hosts][reason]" => "status_reason"
          "[hosts][hostname]" => "hostname"
        }
        remove_field => ["event", "http", "url", "user_agen
t", "hosts"]
      }

      date {
        match => [ "timestamp", "ISO8601" ]
        target => "@timestamp"
      }
    }
  main_03_output.conf: |
    output {
      elasticsearch {
        hosts => ["https://elasticsearch-master:9200"]
        ssl => true
        cacert => "/usr/share/logstash/config/certs/ca.crt"
        user => "elastic"   # Replace with actual username if
authentication is required
        password => "JQvEBqNz5zV4kVOM"
      }
      stdout {
        codec => rubydebug
      }
    }
```

2- Modify Logstash-config.py

Modify the logstash-config.py change the name to nmap-logstash-config.py and replace the existing code with the following code

```
apiVersion: v1
kind: ConfigMap
metadata:
```

```
  name: nmap-logstash-config
data:
  logstash.yml: |
    # Add any configuration you need
    http.host: "0.0.0.0"
    xpack.monitoring.elasticsearch.hosts: [ "https://elastics
earch-master:9200" ]
    xpack.monitoring.elasticsearch.username: elastic
    xpack.monitoring.elasticsearch.password: JQvEBqNz5zV4kVOM
    xpack.monitoring.elasticsearch.ssl.certificate_authority:
/usr/share/logstash/config/certs/ca.crt
    xpack.monitoring.enabled: true

  pipelines.yml: |
    - pipeline.id: main
      path.config: "/usr/share/logstash/pipeline/main*.conf"
```

3- Modify Values.yaml

```
logstash:
  image: "docker.elastic.co/logstash/logstash"
  imageTag: "8.15.0"
  extraVolumes:
    - name: logstash-config
      configMap:
        name: nmap-logstash-config
        items:
          - key: logstash.yml
            path: logstash.yml
          - key: pipelines.yml
            path: pipelines.yml

    - name: pipelines
      configMap:
        name: nmap-pipeline-config
```

```
    - name: ca-cert-volume
      secret:
        secretName: elasticsearch-master-certs

  extraVolumeMounts:
    - name: logstash-config
      mountPath: /usr/share/logstash/config/logstash.yml
      subPath: logstash.yml

    - name: logstash-config
      mountPath: /usr/share/logstash/config/pipelines.yml
      subPath: pipelines.yml

    - name: pipelines
      mountPath: /usr/share/logstash/pipeline

    - name: ca-cert-volume
      mountPath: /usr/share/logstash/config/certs/ca.crt
      subPath: ca.crt
  service:
    type: NodePort
    ports:
      - name: http
        port: 8080
        protocol: TCP
        targetPort: 8080
        nodePort: 30003
```

After these changes, the chart is ready for deployment:

1. Let Helm download logstash subchart: cd `<chart name>/`
2. Install new chart: `helm install logstash-nmap <chart name>/` -n namespace

To send nmap data to logstash-nmap

```
./nmap_scan.sh
```