

ELK Stack Setup Guide

Elasticsearch + Kibana + Logstash — 2-Node Cluster with TLS (PEM)

Overview

This guide installs a secured, two-node Elasticsearch cluster with Kibana (web UI) and Logstash (data pipeline). It uses two Linux machines (physical, virtual, or cloud) that can reach each other over the network.

Machines used in this guide:

- Node 1 ("za-1") — runs Elasticsearch, Kibana, and Logstash
- Node 2 ("za-2") — runs Elasticsearch only

Note: Replace all example IP addresses (192.168.43.30 and 192.168.43.141) with your own machines' real IP addresses throughout this guide.

Prerequisites

- Two Linux machines (Ubuntu/Debian-based), each with at least 2–4 GB RAM and 20+ GB free disk space
- Both machines can reach each other on the network (test with ping)
- A user account with sudo privileges on both machines
- Basic comfort using a terminal

Step 0: Prepare Both Machines

Run this on both Node 1 and Node 2.

```
sudo swapoff -a
sudo sed -i '/ swap / s/^/#/' /etc/fstab

echo "vm.max_map_count=262144" | sudo tee -a /etc/sysctl.conf
sudo sysctl -p

sudo tee -a /etc/hosts <<'EOF'
<NODE1_IP> za-1
<NODE2_IP> za-2
EOF
```

Note: Replace <NODE1_IP> and <NODE2_IP> with the real IP addresses of your two machines.

Step 1: Install Elasticsearch, Kibana, and Logstash

On Node 1, install all three. On Node 2, only Elasticsearch is required.

Run on BOTH nodes:

```
sudo apt-get install -y gnupg apt-transport-https

wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | \
  sudo gpg --dearmor -o /usr/share/keyrings/elastic.gpg

echo "deb [signed-by=/usr/share/keyrings/elastic.gpg]
https://artifacts.elastic.co/packages/8.x/apt stable main" | \
  sudo tee /etc/apt/sources.list.d/elastic-8.x.list

sudo apt-get update
```

On Node 1 only:

```
sudo apt-get install -y elasticsearch kibana logstash
```

On Node 2 only:

```
sudo apt-get install -y elasticsearch
```

Note: During install, Elasticsearch may print an auto-generated password. You can ignore it — this guide sets your own passwords in Step 6.

Step 2: Generate the Certificate Authority and Certificates (Node 1 only)

```
cd /usr/share/elasticsearch
sudo tee instances.yml <<'EOF'
instances:
  - name: "za-1"
    ip: ["<NODE1_IP>"]
  - name: "za-2"
    ip: ["<NODE2_IP>"]
  - name: "kibana"
    ip: ["<NODE1_IP>"]
EOF

sudo ./bin/elasticsearch-certutil ca --pem --out elastic-stack-ca.zip
sudo apt-get install -y unzip zip
sudo unzip -o elastic-stack-ca.zip -d elastic-stack-ca

sudo ./bin/elasticsearch-certutil cert \
  --ca-cert elastic-stack-ca/ca/ca.crt \
  --ca-key elastic-stack-ca/ca/ca.key \
  --pem \
  --in instances.yml \
  --out certs-bundle.zip

sudo unzip -o certs-bundle.zip -d certs-bundle
```

Step 3: Distribute the Certificates

On Node 1:

```
cd /usr/share/elasticsearch
for d in za-1 za-2 kibana; do sudo cp elastic-stack-ca/ca/ca.crt certs-bundle/$d/;
done

sudo mkdir -p /etc/elasticsearch/certs
sudo cp -r certs-bundle/za-1 /etc/elasticsearch/certs/

sudo chown -R $USER:$USER certs-bundle/za-2
scp -r certs-bundle/za-2 <node2_username>@<NODE2_IP>:~/

sudo mkdir -p /etc/kibana/certs
sudo cp -r certs-bundle/kibana/* /etc/kibana/certs/

sudo chown -R root:elasticsearch /etc/elasticsearch/certs/za-1
sudo chmod 750 /etc/elasticsearch/certs/za-1
sudo bash -c 'chmod 640 /etc/elasticsearch/certs/za-1/*'
```

On Node 2 (after the scp transfer completes):

```
sudo mkdir -p /etc/elasticsearch/certs
```

```
sudo mv ~/za-2 /etc/elasticsearch/certs/  
sudo chown -R root:elasticsearch /etc/elasticsearch/certs/za-2  
sudo chmod 750 /etc/elasticsearch/certs/za-2  
sudo bash -c 'chmod 640 /etc/elasticsearch/certs/za-2/*'
```

Step 4: Configure and Start Elasticsearch

On Node 1 — edit /etc/elasticsearch/elasticsearch.yml:

```
sudo tee /etc/elasticsearch/elasticsearch.yml <<'EOF'  
cluster.name: multi-cluster  
node.name: za-1  
node.roles: [ master, data ]  
network.host: <NODE1_IP>  
http.port: 9200  
discovery.seed_hosts: ["<NODE1_IP>", "<NODE2_IP>"]  
cluster.initial_master_nodes: ["za-1", "za-2"]  
xpack.security.enabled: true  
xpack.security.enrollment.enabled: true  
xpack.security.http.ssl:  
  enabled: true  
  verification_mode: certificate  
  key: /etc/elasticsearch/certs/za-1/za-1.key  
  certificate: /etc/elasticsearch/certs/za-1/za-1.crt  
  certificate_authorities: /etc/elasticsearch/certs/za-1/ca.crt  
xpack.security.transport.ssl:  
  enabled: true  
  verification_mode: certificate  
  key: /etc/elasticsearch/certs/za-1/za-1.key  
  certificate: /etc/elasticsearch/certs/za-1/za-1.crt  
  certificate_authorities: /etc/elasticsearch/certs/za-1/ca.crt  
EOF
```

On Node 2 — edit /etc/elasticsearch/elasticsearch.yml:

```
sudo tee /etc/elasticsearch/elasticsearch.yml <<'EOF'  
cluster.name: multi-cluster  
node.name: za-2  
node.roles: [ master, data ]  
network.host: <NODE2_IP>  
http.port: 9200  
discovery.seed_hosts: ["<NODE1_IP>", "<NODE2_IP>"]  
cluster.initial_master_nodes: ["za-1", "za-2"]  
xpack.security.enabled: true  
xpack.security.enrollment.enabled: true  
xpack.security.http.ssl:  
  enabled: true  
  verification_mode: certificate  
  key: /etc/elasticsearch/certs/za-2/za-2.key  
  certificate: /etc/elasticsearch/certs/za-2/za-2.crt  
  certificate_authorities: /etc/elasticsearch/certs/za-2/ca.crt  
xpack.security.transport.ssl:  
  enabled: true  
  verification_mode: certificate  
  key: /etc/elasticsearch/certs/za-2/za-2.key  
  certificate: /etc/elasticsearch/certs/za-2/za-2.crt  
  certificate_authorities: /etc/elasticsearch/certs/za-2/ca.crt  
EOF
```

On BOTH nodes — fix ownership and start the service:

```
sudo chown root:elasticsearch /etc/elasticsearch/elasticsearch.yml
sudo chmod 660 /etc/elasticsearch/elasticsearch.yml
sudo chown -R elasticsearch:elasticsearch /usr/share/elasticsearch
sudo chown -R elasticsearch:elasticsearch /var/log/elasticsearch
sudo chown -R elasticsearch:elasticsearch /var/lib/elasticsearch

sudo systemctl daemon-reload
sudo systemctl enable --now elasticsearch
sudo systemctl status elasticsearch --no-pager
```

Note: If the service uses too much memory on a small machine, cap the JVM heap by creating `/etc/elasticsearch/jvm.options.d/heap.options` with the lines `'-Xms512m'` and `'-Xmx512m'`, then restart the service.

Step 5: Verify the Cluster

On Node 1, copy the CA certificate somewhere readable and check cluster health:

```
sudo cp /etc/elasticsearch/certs/za-1/ca.crt ~/ca.crt
sudo chown $USER:$USER ~/ca.crt

curl --cacert ~/ca.crt -u elastic:<YOUR_ELASTIC_PASSWORD> \
  https://<NODE1_IP>:9200/_cluster/health?pretty
```

A healthy result shows `"status": "green"` and `"number_of_nodes": 2`.

Step 6: Set User Passwords

Run on Node 1:

```
sudo /usr/share/elasticsearch/bin/elasticsearch-reset-password -u elastic
sudo /usr/share/elasticsearch/bin/elasticsearch-reset-password -u kibana_system
sudo /usr/share/elasticsearch/bin/elasticsearch-reset-password -u logstash_system
```

Note: Each command prints a new password. Record all three somewhere safe — they are needed in the next steps and are not shown again.

Step 7: Configure and Start Kibana (Node 1 only)

```
sudo tee /etc/kibana/kibana.yml <<'EOF'
server.port: 5601
server.host: "<NODE1_IP>"
elasticsearch.hosts: ["https://<NODE1_IP>:9200", "https://<NODE2_IP>:9200"]
elasticsearch.username: "kibana_system"
elasticsearch.password: "<KIBANA_SYSTEM_PASSWORD>"
elasticsearch.ssl.certificateAuthorities: ["/etc/kibana/certs/ca.crt"]
server.ssl.enabled: true
server.ssl.certificate: /etc/kibana/certs/kibana.crt
server.ssl.key: /etc/kibana/certs/kibana.key
EOF

sudo chown -R kibana:kibana /etc/kibana/certs
sudo chown root:kibana /etc/kibana/kibana.yml
sudo chmod 660 /etc/kibana/kibana.yml

sudo systemctl daemon-reload
sudo systemctl enable --now kibana
sudo systemctl status kibana --no-pager
```

Once Kibana finishes starting (this can take a few minutes), open a browser and go to:

```
https://<NODE1_IP>:5601
```

Accept the browser's self-signed certificate warning, then log in with username elastic and the elastic password from Step 6.

Step 8: Configure and Start Logstash (Node 1 only)

```
sudo tee /etc/logstash/conf.d/main.conf <<'EOF'
input {
  file {
    path => "/var/log/syslog"
    start_position => "beginning"
  }
}
output {
  elasticsearch {
    hosts => ["https://<NODE1_IP>:9200", "https://<NODE2_IP>:9200"]
    ssl_certificate_authorities => ["/etc/elasticsearch/certs/za-1/ca.crt"]
    user => "logstash_system"
    password => "<LOGSTASH_SYSTEM_PASSWORD>"
    index => "syslog-%{+YYYY.MM.dd}"
  }
}
EOF

sudo /usr/share/logstash/bin/logstash --path.settings /etc/logstash -t
sudo systemctl enable --now logstash
```

In Kibana, go to Discover and create a data view for syslog-* to see incoming log data.

Troubleshooting Tips

- Service fails with a permissions error: re-check file ownership/permissions from Step 4 (files must be owned by root:elasticsearch, folders 750, files 640/660).
- Service is killed with "oom-kill": the machine ran out of RAM. Lower the JVM heap size and/or increase the machine's memory.
- "Cluster state has not been recovered yet": the other node isn't reachable yet. Confirm both machines are powered on, both Elasticsearch services are running, and they can ping each other.
- Browser can't reach Kibana: confirm the machine's IP address hasn't changed (run `ip a`) and that both services show "active (running)".
- curl reports it cannot find the certificate: if using the Snap version of curl, copy the certificate to your home folder first, since Snap cannot read files under `/etc/`.