# Hướng dẫn triển khai ELK Stack 8.13.4

Người viết: Lê Nhựt Anh  
Email: Anh2LN@hcmp.com.vn

Tài liệu này được biên soạn dành cho các kỹ sư phần mềm / DevOps cần thiết lập nhanh bộ ELK Stack (Elasticsearch – Logstash – Kibana) phiên bản 8.13.4 trên hệ điều hành Oracle Linux / CentOS / RHEL 8.

## 1. Tổng quan thành phần

• \*\*Elasticsearch\*\*: cơ sở dữ liệu tìm kiếm phân tán, lưu trữ dữ liệu dưới dạng JSON; cung cấp REST API tốc độ cao, hỗ trợ full‑text, aggregation và security (xpack).

• \*\*Logstash\*\*: pipeline thu thập – chuyển đổi – gửi log/metrics. Cấu trúc pipeline gồm input → filter → output; trong tài liệu này output chính là Elasticsearch qua HTTPS.

• \*\*Kibana\*\*: giao diện trực quan hóa, truy vấn và giám sát dữ liệu; đồng thời cung cấp DevTools, Alerting, Dashboard.

## 2. Cách vận hành tổng thể

Logstash đọc log (ví dụ generator input), áp dụng filter (nếu có) rồi gửi tới Elasticsearch index `test-logs-\*`. Kibana kết nối bằng account `kibana\_system` để truy vấn và hiển thị dashboard. Tất cả lưu lượng REST dùng SSL tự ký được tạo trong script.

## 3. Cài đặt Elasticsearch – Kibana – Logstash trên 1 node và là node master

|  |  |  |
| --- | --- | --- |
| **#** | **Lệnh / khối lệnh chính** | **Mục đích** |
| 1 | id -u $ES\_USER … || sudo useradd … | Tạo tài khoản hệ thống **elasticsearch** nếu chưa có. |
| 2 | sudo mkdir -p $BASE\_DIRsudo tar -xzf "$USER\_HOME/$ES\_TAR" -C $BASE\_DIR | Tạo thư mục gốc /opt (nếu chưa có) và giải nén gói **Elasticsearch**. |
| 3 | sudo chown -R $ES\_USER:$ES\_USER $ES\_DIR | Gán quyền sở hữu thư mục cài Elasticsearch cho user **elasticsearch**. |
| 4 | sudo ln -sfn $ES\_DIR $ES\_SYM | Tạo symlink /opt/elasticsearch trỏ tới phiên bản vừa cài, giúp nâng cấp dễ dàng. |
| 5 | Ghi file /etc/profile.d/elasticsearch.sh rồi source | Đặt biến môi trường **ES\_HOME** và thêm vào **PATH** để chạy lệnh ES ở mọi nơi. |
| 6 | echo "vm.max\_map\_count=262144" …sudo sysctl -w vm.max\_map\_count=262144 | Tăng tham số kernel vm.max\_map\_count – bắt buộc để ES hoạt động ổn định. |
| 7 | Ghi elasticsearch.yml (single-node + security) | Cấu hình ES chạy chế độ 1 nút, bật bảo mật, mở cổng mọi IP. |
| 8 | Ghi /etc/systemd/system/elasticsearch.service | Khai báo service systemd cho Elasticsearch. |
| 9 | systemctl enable & start elasticsearch | Cho ES khởi động cùng hệ thống và chạy ngay. |
| 10 | Vòng until curl -s http://localhost:9200 | Đợi ES sẵn sàng trước khi làm bước kế tiếp. |
| 11 | elasticsearch-setup-passwords auto -b > $ENV\_FILE | Sinh toàn bộ mật khẩu mặc định (elastic, kibana\_system …) và lưu vào file .env. |
| 12 | sed -i … xpack.security.transport.ssl.enabled: true | Bật SSL cho kênh Transport (nội bộ cluster). |
| 13 | Tạo user hệ thống **kibana** | Đảm bảo phân quyền riêng cho dịch vụ Kibana. |
| 14 | Giải nén gói Kibana | Cài đặt Kibana vào /opt. |
| 15 | Phân quyền + symlink | Gán quyền user kibana và trỏ /opt/kibana tới bản cài. |
| 16 | Ghi kibana.yml | Trỏ Kibana tới ES, nạp user/pass và tắt telemetry. |
| 17 | Tạo service systemd cho Kibana | Khởi chạy Kibana dưới user kibana, đặt limit file-handle. |
| 18 | systemctl enable & start kibana | Khởi động và tự chạy lại khi reboot. |
| 19 | Đợi Kibana sẵn sàng (curl :5601) | Xác nhận giao diện Web đã online. |
| 20 | Thông báo URL Kibana | Nhắc admin đường dẫn truy cập. |
| 21 | Tạo user hệ thống **logstash** | Phân tách quyền cho Logstash. |
| 22 | Giải nén Logstash | Thêm thành phần thu-thập log vào stack. |
| 23 | Phân quyền + symlink Logstash | Cho phép user logstash sở hữu thư mục, tạo /opt/logstash. |
| 24 | Gọi API ES tạo **role & user logstash\_writer** | Cấp quyền ghi index & template cho pipeline Logstash. |
| 25 | Viết file pipeline mẫu /etc/logstash/sample.conf | Minh họa input generator → output ES + stdout. |
| 26 | Tạo service systemd cho Logstash | Định nghĩa cách Logstash đọc cấu hình và đường dẫn data. |
| 27 | Tạo /var/lib/logstash & phân quyền | Thư mục lưu trạng thái pipeline, sincedb … |
| 28 | systemctl daemon-reload, enable, start Logstash | Nạp unit mới, bật và chạy ngay Logstash. |
| 29 | Đợi API monitoring Logstash (curl :9600) | Bảo đảm Logstash hoạt động trước khi cấu hình SSL. |
| 30 | Ghi lại **elasticsearch.yml** chi tiết (cluster, SSL) | Chuyển sang mô hình cluster 2 node (master & data) và bật TLS đầy đủ (HTTP + Transport). |
| 31 | elasticsearch-certutil ca/cert … | Tạo CA & keystore elasticsearch.p12 đặt trong config/. |
| 32 | Restart ES & đợi HTTPS (curl -ks https://localhost:9200) | Áp dụng TLS, kiểm tra ES nghe trên HTTPS. |
| 33 | Sinh chứng chỉ cho Kibana, trích cert/key, cập nhật kibana.yml | Cho giao diện Kibana chạy HTTPS và tin cậy CA mới tạo. |
| 34 | Sinh chứng chỉ cho Logstash, chỉnh sample.conf dùng HTTPS | Bảo mật đường truyền Logstash → Elasticsearch. |
| 35 | cp elastic-stack-ca.p12 /home/$USER\_HOME + thông báo | Sao chép CA về thư mục người dùng để tiện phân phối cho node khác; in thông báo hoàn tất & vị trí file mật khẩu. |

## 4. Script cài đặt đầy đủ cho node master

Nội dung bên dưới có thể sao chép trực tiếp thành file `install.sh`. Để cài đặt trên Oracle Linux chỉ cần gõ lệnh “sh install.sh” mặc định sẽ lấy tài khoản root, nếu cài bằng user khác thì lệnh là “sh install.sh <username>“

|  |
| --- |
| #!/bin/bash  set -e  ############################################################  # CÀI ĐẶT ELK STACK 8.13.4 + SSL (giữ nguyên lệnh gốc)  # Tác giả: <YourName> – 04/2025  ############################################################  #################### BIẾN CHUNG ##########################  BASE\_DIR="/opt"  MASTER\_HOST="192.168.56.10"  DATA\_HOST="192.168.56.11"  # Elasticsearch  ES\_VERSION="8.13.4"  ES\_TAR="elasticsearch-${ES\_VERSION}-linux-x86\_64.tar.gz"  ES\_DIR="${BASE\_DIR}/elasticsearch-${ES\_VERSION}"  ES\_SYM="${BASE\_DIR}/elasticsearch"  ES\_USER="elasticsearch"  # Kibana  KIBANA\_VERSION="8.13.4"  KIBANA\_TAR="kibana-${KIBANA\_VERSION}-linux-x86\_64.tar.gz"  KIBANA\_DIR="${BASE\_DIR}/kibana-${KIBANA\_VERSION}"  KIBANA\_SYM="${BASE\_DIR}/kibana"  KIBANA\_USER="kibana"  # Logstash  LOGSTASH\_VERSION="8.13.4"  LOGSTASH\_TAR="logstash-${LOGSTASH\_VERSION}-linux-x86\_64.tar.gz"  LOGSTASH\_DIR="${BASE\_DIR}/logstash-${LOGSTASH\_VERSION}"  LOGSTASH\_SYM="${BASE\_DIR}/logstash"  LOGSTASH\_USER="logstash"  # File chung  ENV\_FILE="/home/vagrant/elk-passwords.env"  CA\_PASS="changeme"  ############################################################  echo "1/35 👉 Tạo user Elasticsearch (nếu chưa có)…"  id -u $ES\_USER &>/dev/null || sudo useradd --system --no-create-home --shell /sbin/nologin $ES\_USER  echo "2/35 📦 Giải nén Elasticsearch…"  sudo mkdir -p $BASE\_DIR  sudo tar -xzf /home/vagrant/$ES\_TAR -C $BASE\_DIR  echo "3/35 🔧 Phân quyền thư mục Elasticsearch…"  sudo chown -R $ES\_USER:$ES\_USER $ES\_DIR  echo "4/35 🔗 Tạo symlink Elasticsearch…"  sudo ln -sfn $ES\_DIR $ES\_SYM  echo "5/35 ⚙️ Khai báo ES\_HOME và PATH…"  echo "export ES\_HOME=$ES\_SYM" | sudo tee /etc/profile.d/elasticsearch.sh  echo "export PATH=\$ES\_HOME/bin:\$PATH" | sudo tee -a /etc/profile.d/elasticsearch.sh  sudo chmod +x /etc/profile.d/elasticsearch.sh  source /etc/profile.d/elasticsearch.sh  echo "6/35 🖥️ Tăng vm.max\_map\_count kernel…"  echo "vm.max\_map\_count=262144" | sudo tee -a /etc/sysctl.conf  sudo sysctl -w vm.max\_map\_count=262144  echo "7/35 📑 Tạo elasticsearch.yml (single-node + security)…"  sudo tee $ES\_DIR/config/elasticsearch.yml >/dev/null <<EOF  discovery.type: single-node  xpack.security.enabled: true  xpack.security.authc.api\_key.enabled: true  xpack.security.http.ssl:  enabled: false  network.host: 0.0.0.0  EOF  echo "8/35 🛠️ Tạo service Elasticsearch…"  sudo tee /etc/systemd/system/elasticsearch.service >/dev/null <<EOF  [Unit]  Description=Elasticsearch $ES\_VERSION  Documentation=https://www.elastic.co  Wants=network-online.target  After=network-online.target  [Service]  Type=simple  User=$ES\_USER  Group=$ES\_USER  ExecStart=$ES\_DIR/bin/elasticsearch  Restart=always  LimitNOFILE=65535  Environment=ES\_JAVA\_HOME=$ES\_DIR/jdk  [Install]  WantedBy=multi-user.target  EOF  echo "9/35 🚀 Bật & khởi động Elasticsearch…"  sudo systemctl enable elasticsearch  sudo systemctl start elasticsearch  echo "10/35 ⏳ Đợi Elasticsearch sẵn sàng (HTTP)…"  until curl -s http://localhost:9200 >/dev/null; do sleep 2; done  echo "11/35 🔐 Sinh mật khẩu mặc định…"  sudo -u $ES\_USER $ES\_DIR/bin/elasticsearch-setup-passwords auto -b > $ENV\_FILE  sudo chown vagrant:vagrant $ENV\_FILE  ES\_PASSWORD=$(grep "PASSWORD elastic" $ENV\_FILE | awk '{print $4}')  echo "12/35 🔧 Bật SSL Transport (dòng thêm)…"  sudo sed -i 's/xpack.security.enabled: true/&\  xpack.security.transport.ssl.enabled: true/' $ES\_DIR/config/elasticsearch.yml  ################### KIBANA ###################  echo "13/35 👉 Tạo user Kibana (nếu chưa có)…"  id -u $KIBANA\_USER &>/dev/null || sudo useradd --system --no-create-home --shell /sbin/nologin $KIBANA\_USER  echo "14/35 📦 Giải nén Kibana…"  sudo tar -xzf /home/vagrant/$KIBANA\_TAR -C $BASE\_DIR  echo "15/35 🔧 Phân quyền & symlink Kibana…"  sudo chown -R $KIBANA\_USER:$KIBANA\_USER $KIBANA\_DIR  sudo ln -sfn $KIBANA\_DIR $KIBANA\_SYM  echo "16/35 📑 Viết kibana.yml (security)…"  sudo tee $KIBANA\_DIR/config/kibana.yml >/dev/null <<EOF  server.host: "0.0.0.0"  elasticsearch.hosts: ["http://localhost:9200"]  elasticsearch.username: "kibana\_system"  elasticsearch.password: "$(grep "PASSWORD kibana\_system" $ENV\_FILE | awk '{print $4}')"  telemetry.optIn: false  xpack.security:  session.idleTimeout: "30m"  session.lifespan: "8h"  encryptionKey: "$(openssl rand -hex 32)"  EOF  echo "17/35 🛠️ Tạo service Kibana…"  sudo tee /etc/systemd/system/kibana.service >/dev/null <<EOF  [Unit]  Description=Kibana $KIBANA\_VERSION  Documentation=https://www.elastic.co  Wants=network-online.target  After=network-online.target elasticsearch.service  [Service]  Type=simple  User=$KIBANA\_USER  Group=$KIBANA\_USER  ExecStart=$KIBANA\_DIR/bin/kibana  Restart=always  LimitNOFILE=65535  Environment=NODE\_OPTIONS="--max-old-space-size=2048"  [Install]  WantedBy=multi-user.target  EOF  echo "18/35 🚀 Bật & khởi động Kibana…"  sudo systemctl enable kibana  sudo systemctl start kibana  echo "19/35 ⏳ Đợi Kibana sẵn sàng (HTTP)…"  until curl -s http://localhost:5601 >/dev/null; do sleep 2; done  echo "20/35 ✅ Kibana đã sẵn sàng tại http://localhost:5601"  ################### LOGSTASH #################  echo "21/35 👉 Tạo user Logstash (nếu chưa có)…"  id -u $LOGSTASH\_USER &>/dev/null || sudo useradd --system --no-create-home --shell /sbin/nologin $LOGSTASH\_USER  echo "22/35 📦 Giải nén Logstash…"  sudo tar -xzf /home/vagrant/$LOGSTASH\_TAR -C $BASE\_DIR  echo "23/35 🔧 Phân quyền & symlink Logstash…"  sudo chown -R $LOGSTASH\_USER:$LOGSTASH\_USER $LOGSTASH\_DIR  sudo ln -sfn $LOGSTASH\_DIR $LOGSTASH\_SYM  echo "24/35 🔐 Tạo role & user logstash\_writer…"  LOGSTASH\_WRITER\_PASSWORD=$(openssl rand -hex 12)  curl -X POST "http://localhost:9200/\_security/role/logstash\_writer" \  -u "elastic:${ES\_PASSWORD}" -H "Content-Type: application/json" -d '{  "cluster":["monitor","manage\_index\_templates"],  "indices":[{"names":["\*"],"privileges":["create\_index","write","delete","index","read"]}]  }'  curl -X POST "http://localhost:9200/\_security/user/logstash\_writer" \  -u "elastic:${ES\_PASSWORD}" -H "Content-Type: application/json" -d '{  "password":"'"${LOGSTASH\_WRITER\_PASSWORD}"'",  "roles":["logstash\_writer"]  }'  echo "PASSWORD logstash\_writer = ${LOGSTASH\_WRITER\_PASSWORD}" >> $ENV\_FILE  echo "25/35 🗄️ Viết pipeline mẫu…"  sudo mkdir -p /etc/logstash  sudo tee /etc/logstash/sample.conf >/dev/null <<EOF  input { generator { lines => ["Hello, world!", "Logstash is awesome!"] count => 10 } }  output {  elasticsearch {  hosts => ["http://localhost:9200"]  index => "test-logs-%{+YYYY.MM.dd}"  user => "logstash\_writer"  password => "${LOGSTASH\_WRITER\_PASSWORD}"  }  stdout { codec => rubydebug }  }  EOF  sudo chown -R $LOGSTASH\_USER:$LOGSTASH\_USER /etc/logstash  echo "26/35 🛠️ Tạo service Logstash…"  sudo tee /etc/systemd/system/logstash.service >/dev/null <<EOF  [Unit]  Description=Logstash ${LOGSTASH\_VERSION}  Documentation=https://www.elastic.co  Wants=network-online.target  After=network-online.target elasticsearch.service  [Service]  Type=simple  User=${LOGSTASH\_USER}  Group=${LOGSTASH\_USER}  ExecStart=${LOGSTASH\_SYM}/bin/logstash --path.settings ${LOGSTASH\_SYM}/config --path.data /var/lib/logstash -f /etc/logstash/  Restart=always  LimitNOFILE=65535  [Install]  WantedBy=multi-user.target  EOF  echo "27/35 📂 Tạo /var/lib/logstash & cấp quyền…"  sudo mkdir -p /var/lib/logstash  sudo chown -R $LOGSTASH\_USER:$LOGSTASH\_USER /var/lib/logstash  echo "28/35 🚀 Bật & khởi động Logstash…"  sudo systemctl daemon-reload  sudo systemctl enable logstash  sudo systemctl start logstash  echo "29/35 ⏳ Đợi API monitoring Logstash…"  until curl -s http://localhost:9600/\_node/pipelines >/dev/null; do sleep 2; done  ################### SSL ####################  echo "30/35 🔒 Sửa elasticsearch.yml để bật SSL HTTP & Transport…"  sudo sed -i '/^xpack\.security\.http\.ssl:/,/^ enabled: false/d' $ES\_SYM/config/elasticsearch.yml  sudo sed -i '/^xpack\.security\.transport\.ssl\.enabled: true/d' $ES\_SYM/config/elasticsearch.yml  sudo rm -f $ES\_SYM/config/elasticsearch.yml  sudo touch $ES\_SYM/config/elasticsearch.yml  sudo chown $ES\_USER:$ES\_USER $ES\_SYM/config/elasticsearch.yml  sudo tee -a $ES\_SYM/config/elasticsearch.yml >/dev/null <<EOF  cluster.name: elk-single  node.name: master-1  node.roles: [master, data]  network.host: 0.0.0.0  transport.bind\_host: $MASTER\_HOST  cluster.initial\_master\_nodes: ["master-1"]  discovery.seed\_hosts: ["$MASTER\_HOST", "$DATA\_HOST"]  network.publish\_host: $MASTER\_HOST  # ---- SSL ----  xpack.security.http.ssl.enabled: true  xpack.security.http.ssl.keystore.type: PKCS12  xpack.security.http.ssl.keystore.path: elasticsearch.p12  xpack.security.http.ssl.keystore.password: $CA\_PASS  xpack.security.http.ssl.truststore.path: elasticsearch.p12  xpack.security.http.ssl.truststore.password: $CA\_PASS  xpack.security.transport.ssl.enabled: true  xpack.security.transport.ssl.verification\_mode: certificate  xpack.security.transport.ssl.keystore.type: PKCS12  xpack.security.transport.ssl.keystore.path: elasticsearch.p12  xpack.security.transport.ssl.keystore.password: $CA\_PASS  xpack.security.transport.ssl.truststore.type: PKCS12  xpack.security.transport.ssl.truststore.path: elasticsearch.p12  xpack.security.transport.ssl.truststore.password: $CA\_PASS  EOF  sudo systemctl stop elasticsearch  sudo systemctl start elasticsearch  echo "31/35 🔒 Sinh CA & keystore elasticsearch.p12…"  sudo rm -f $ES\_DIR/config/elastic-stack-ca.p12 $ES\_DIR/config/elasticsearch.p12  printf '%s\n%s\n' $CA\_PASS $CA\_PASS | \  sudo $ES\_SYM/bin/elasticsearch-certutil ca --silent --pass $CA\_PASS \  --out $ES\_DIR/config/elastic-stack-ca.p12  printf '%s\n%s\n' $CA\_PASS $CA\_PASS | \  sudo $ES\_SYM/bin/elasticsearch-certutil cert --name elasticsearch --ca $ES\_DIR/config/elastic-stack-ca.p12 \  --silent --ca-pass $CA\_PASS --pass $CA\_PASS --out $ES\_DIR/config/elasticsearch.p12  sudo chown $ES\_USER:$ES\_USER $ES\_DIR/config/\*.p12  sudo chmod 640 $ES\_DIR/config/\*.p12  echo "32/35 🔁 Khởi động lại Elasticsearch (HTTPS)…"  sudo systemctl stop elasticsearch  sudo systemctl start elasticsearch  until curl -ks https://localhost:9200 >/dev/null; do sleep 2; done  echo "33/35 🔒 Sinh chứng chỉ Kibana & cấu hình TLS…"  sudo /opt/elasticsearch/bin/elasticsearch-certutil cert --name kibana --ca $ES\_DIR/config/elastic-stack-ca.p12 \  --silent --ca-pass $CA\_PASS --pass $CA\_PASS --out $KIBANA\_DIR/config/kibana.p12  sudo openssl pkcs12 -in $KIBANA\_DIR/config/kibana.p12 -nocerts -nodes -passin pass:$CA\_PASS | \  sudo tee $KIBANA\_DIR/config/kibana.key >/dev/null  sudo openssl pkcs12 -in $KIBANA\_DIR/config/kibana.p12 -clcerts -nokeys -passin pass:$CA\_PASS \  -out $KIBANA\_DIR/config/kibana.crt  sudo openssl pkcs12 -in $ES\_DIR/config/elastic-stack-ca.p12 -nokeys -clcerts -passin pass:$CA\_PASS \  -out $KIBANA\_DIR/config/elastic-stack-ca.pem  sudo chown $KIBANA\_USER:$KIBANA\_USER $KIBANA\_DIR/config/kibana.\* $KIBANA\_DIR/config/elastic-stack-ca.pem  sudo chmod 640 $KIBANA\_DIR/config/kibana.\* $KIBANA\_DIR/config/elastic-stack-ca.pem  sudo sed -i '/^elasticsearch\.hosts:/d' $KIBANA\_DIR/config/kibana.yml  sudo tee -a $KIBANA\_DIR/config/kibana.yml >/dev/null <<EOF  # --- TLS ---  server.ssl.enabled: true  server.ssl.certificate: $KIBANA\_DIR/config/kibana.crt  server.ssl.key: $KIBANA\_DIR/config/kibana.key  elasticsearch.hosts: ["https://localhost:9200"]  elasticsearch.ssl.certificateAuthorities: ["$KIBANA\_DIR/config/elastic-stack-ca.pem"]  elasticsearch.ssl.verificationMode: certificate  EOF  sudo systemctl restart kibana  until curl -ks https://localhost:5601 >/dev/null; do sleep 2; done  echo "34/35 🔒 Sinh chứng chỉ Logstash & chỉnh pipeline SSL & join token"  sudo /opt/elasticsearch/bin/elasticsearch-certutil cert --name logstash --ca $ES\_DIR/config/elastic-stack-ca.p12 \  --silent --ca-pass $CA\_PASS --pass $CA\_PASS --out $LOGSTASH\_DIR/config/logstash.p12  sudo openssl pkcs12 -in $LOGSTASH\_DIR/config/logstash.p12 -nocerts -nodes -passin pass:$CA\_PASS | \  sudo tee $LOGSTASH\_DIR/config/logstash.key >/dev/null  sudo openssl pkcs12 -in $LOGSTASH\_DIR/config/logstash.p12 -clcerts -nokeys -passin pass:$CA\_PASS \  -out $LOGSTASH\_DIR/config/logstash.crt  sudo cp $KIBANA\_DIR/config/elastic-stack-ca.pem $LOGSTASH\_DIR/config/  sudo chown $LOGSTASH\_USER:$LOGSTASH\_USER $LOGSTASH\_DIR/config/logstash.\* $LOGSTASH\_DIR/config/elastic-stack-ca.pem  sudo chmod 640 $LOGSTASH\_DIR/config/logstash.\* $LOGSTASH\_DIR/config/elastic-stack-ca.pem  sudo sed -i -e 's|\(\s\*hosts\s\*=>\s\*\)\["http://|\1["https://|g' \  -e '/^\s\*hosts\s\*=>/a \ ssl\_certificate\_verification => false\n ssl => true\n cacert => "'"$LOGSTASH\_DIR/config/elastic-stack-ca.pem"'"' \  /etc/logstash/sample.conf  sudo systemctl restart logstash  sudo cp /opt/elasticsearch/config/elastic-stack-ca.p12 ~/  echo "35/35 ✅ Hoàn tất! ELK đang chạy HTTPS. Mật khẩu lưu tại $ENV\_FILE" |

## 5. Hình ảnh minh hoạ

A screenshot of a computer screen

Description automatically generated

Hình 1: Hình ảnh hoàn tất cài đặt Elasticsearch, Kibana, Logstash

A screenshot of a computer

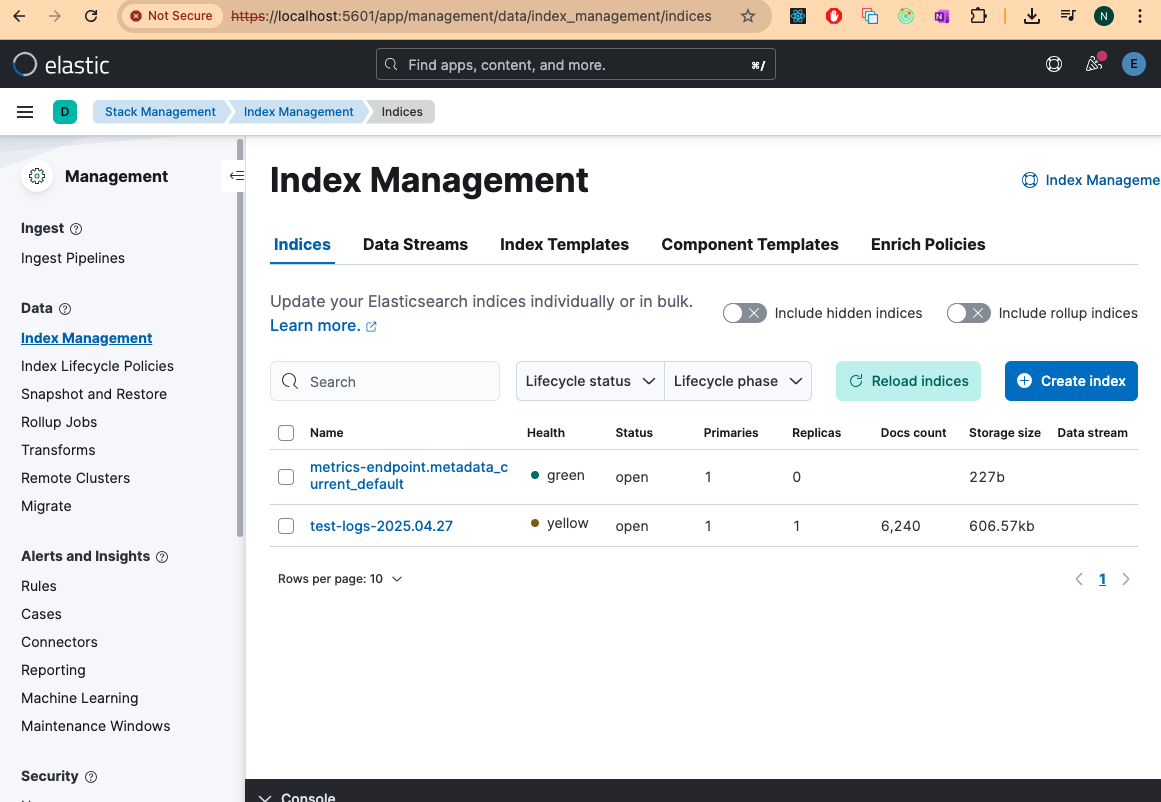
Description automatically generated

Hình 2: Hoàn thành cài đặt elasticsearch trên 1 node

A screenshot of a computer

Description automatically generated

Hình 3: Hình ảnh Kibana đã hoạt động trên https



Hình 4: Logstash tạo log và gửi vào elasticsearch và hiển thị trên Kibana   
(user: elastic, password nằm trong file elk-passwords.env trong thư mục home)



Hình 5: các file có trong thư mục $USER/home

A screenshot of a computer screen

Description automatically generated

Hình 6: Minh hoạ các tài khoản được tạo mật khẩu tự động

Ghi chú:   
1. Các tài khoản kibana\_system, logstash\_system, logstash\_writer, apm\_system, beats\_system, remote\_monitoring\_user là các tài khoản sử dụng để gọi api giữa các phần mềm.  
2. Các tài khoản kibana, logstash, elasticsearch là tài khoản của hệ điều hành và không có quyền shell, chỉ để chạy service   
3. Tài khoản elastic là tài khoản admin dùng để đăng nhập vào api elasticsearch hoặc trang web kibana

## 6. Cài đặt Elasticsearch trên node khác và thêm node vào master

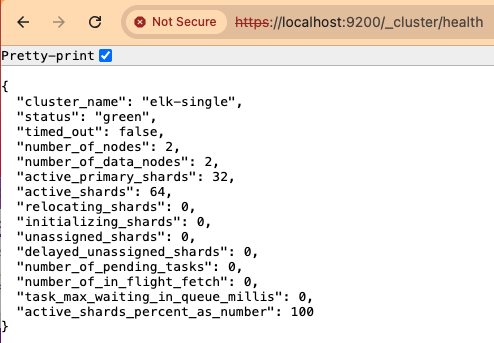
Sau khi cài đặt script trên node master, ta sẽ cài đặt script trên node worker. Trước hết ta sẽ chép file được script tạo sẵn ở thư mục home của máy đã cài script master. File có tên “elastic-stack-ca.p12”. Đây là file dùng để sinh các chứng chỉ cho node worker sử dụng để kết hợp với node master. Ta sẽ copy vào thư mục home của các node worker và chạy script với lệnh “sh install\_node.sh” nếu cài với user khác thì lệnh như sau “sh install\_node.sh <username>”

|  |  |  |
| --- | --- | --- |
| **#** | **Lệnh / khối lệnh** | **Mục đích (giải thích ngắn gọn)** |
| 1 | id -u $ES\_USER … || useradd … | Tạo tài khoản hệ thống **elasticsearch** (nếu chưa tồn tại). |
| 2 | tar -xzf … && ln -sfn … && chown … | Giải nén Elasticsearch vào /opt, tạo symlink cố định /opt/elasticsearch, gán quyền sở hữu cho user **elasticsearch**. |
| 3 | echo "vm.max\_map\_count=262144" …sysctl -w vm.max\_map\_count=262144 | Tăng tham số kernel vm.max\_map\_count – bắt buộc để Elasticsearch hoạt động. |
| 4 | if [[ -f "$HOME\_CA" ]]; then … | Kiểm tra sự tồn tại của **elastic-stack-ca.p12** (CA) đã copy từ master. Nếu có, cấu hình SSL cho node phụ; nếu không, chạy standalone không TLS. |
| 5 | elasticsearch-certutil cert --name "$NODE\_NAME" … | Sinh keystore **PKCS12** riêng cho node (data-1.p12) bằng CA, dùng chung pass phrase $CA\_PASS. |
| 6 | Khối SSL\_BLOCK= và ghi elasticsearch.yml | Tạo block cấu hình bảo mật (HTTP + Transport TLS) và cấu hình tham số cluster (cluster.name, node.name, discovery.seed\_hosts, publish\_host). |
| 7 | Ghi /etc/systemd/system/elasticsearch.service | Định nghĩa service **systemd** để quản lý Elasticsearch, đặt giới hạn file-handle và chỉ định JDK đi kèm. |
| 8 | systemctl daemon-reload, enable, start | Nạp service, cấu hình tự khởi động cùng hệ thống, khởi chạy node Elasticsearch. |
| 9 | Vòng until curl -ks https://localhost:9200 | Chờ tới khi cổng HTTPS 9200 sẵn sàng để xác nhận node đã lên và TLS hoạt động. |

Sau khi chạy, node worker sẽ liên kết với node master để chạy theo vai trò đã được chỉ định trong file cấu hình “elasticsearch.yml” nằm ở đường dẫn “/opt/elasticsearch/config/elasticsearch.yml”  
  
A screenshot of a computer program

Description automatically generated

Hình 7: Log hiển thị khi đã cài thành công script trên node worker

  
Hình 8: Node data đã vào cụm elastic thành công