Installation and Configuration Manual

Installation and Configuration Manual

This manual covers setting up Elasticsearch, Logstash, Kibana (ELK) on your main server (`209.97.172.92`) and Filebeat on an agent server (e.g., `152.42.191.42`). This aligns with our project's goal to "Implement scalable pipelines using Elasticsearch and Logstash" and "Deploy Elastic Fleet agents" (though we are using Filebeat, the principle of agents is similar).

A. Prerequisites (Run on all servers where components will be installed)

1. **Update System:**

   ```
   sudo apt update && sudo apt upgrade -y
   ```

2. **Install Java (Required for Elasticsearch and Logstash):**
   The Elastic Stack typically bundles its own JDK, but having a system Java can be useful. Elasticsearch 8.x bundles Java. If you need to install it for other reasons or for older versions:

   ```
   sudo apt install openjdk-17-jre-headless -y # Or a compatible version
   java -version # Verify installation
   ```

3. **Install `apt-transport-https` and `gnupg` (for adding Elastic repository):**

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

B. Elasticsearch Installation and Configuration (on Main Server: `209.97.172.92`)

1. **Import the Elasticsearch GPG Key:**

   ```
   curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o /usr/share/keyrings/elastic-keyring.gpg
   ```

2. **Add the Elasticsearch Repository:**

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

3. **Install Elasticsearch:**

   ```
   sudo apt update
   sudo apt install elasticsearch # This will install the version matching your repo (e.g., 8.18.1)
   ```

4. **Configure Elasticsearch (`/etc/elasticsearch/elasticsearch.yml`):**

```
sudo nano /etc/elasticsearch/elasticsearch.yml
```

Make these critical changes:

```
cluster.name: my-siem-cluster # Or your preferred name
node.name: node-1 # Or your preferred name

path.data: /var/lib/elasticsearch
path.logs: /var/log/elasticsearch

network.host: 0.0.0.0 # To listen on all interfaces, or your server's specific IP
http.port: 9200

# --- Security Settings (CRUCIAL) ---
xpack.security.enabled: true
xpack.security.enrollment.enabled: true # Important for initial setup if not already done

# SSL/TLS for HTTP (client connections like Kibana, Logstash)
xpack.security.http.ssl:
  enabled: true
  keystore.path: certs/http.p12 # Relative to /etc/elasticsearch

# SSL/TLS for Transport (node-to-node communication, important even for single node)
xpack.security.transport.ssl:
  enabled: true
  verification_mode: certificate
  keystore.path: certs/transport.p12 # Relative to /etc/elasticsearch
  truststore.path: certs/transport.p12 # Relative to /etc/elasticsearch

# For single-node discovery (if it's a single Elasticsearch node)
discovery.type: single-node
cluster.initial_master_nodes: ["node-1"] # Match your node.name
```

- **Certificates:** The paths `certs/http.p12` and `certs/transport.p12` mean Elasticsearch expects these files in `/etc/elasticsearch/certs/`. You would have generated these earlier, likely using:

  Ensure `http_ca.crt` (the CA that signed `http.p12`) is also in `/etc/elasticsearch/certs/` for other components to use.

  ```
  # Run ONCE on the Elasticsearch server if certs don't exist
  # sudo /usr/share/elasticsearch/bin/elasticsearch-certutil ca --out /etc/elasticsearch/certs/elastic-stack-ca.p12 --pass YOUR_CA_PASSWORD
  # sudo /usr/share/elasticsearch/bin/elasticsearch-certutil cert --name http --ca /etc/elasticsearch/certs/elastic-stack-ca.p12 --ca-pass YOUR_CA_PASSWORD --out /etc/elasticsearch/certs/http.p12 --pass YOUR_HTTP_CERT_PASSWORD --dns your-server-ip,localhost
  # sudo /usr/share/elasticsearch/bin/elasticsearch-certutil cert --name transport --ca /etc/elasticsearch/certs/elastic-stack-ca.p12 --ca-pass YOUR_CA_PASSWORD --out /etc/elasticsearch/certs/transport.p12 --pass YOUR_TRANSPORT_CERT_PASSWORD --dns your-server-ip,localhost
  # (And also extract the http_ca.crt: sudo openssl pkcs12 -in /etc/elasticsearch/certs/elastic-stack-ca.p12 -nokeys -out /etc/elasticsearch/certs/http_ca.crt -cacerts -passin pass:YOUR_CA_PASSWORD)
  ```

5. **Start and Enable Elasticsearch:**

```
sudo systemctl daemon-reload sudo systemctl enable elasticsearch.service sudo systemctl start elasticsearch.service sudo systemctl status elasticsearch.service # Check it's active (running)
```

1. **Set Passwords for Built-in Users (if not done during initial setup):**
   This is critical for security.

```
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
```

Save these passwords securely! You'll need them for Kibana and Logstash configurations.

C. Kibana Installation and Configuration (on Main Server: `209.97.172.92`)

1. **Install Kibana (uses the same Elastic repository):**

   ```
   sudo apt install kibana
   ```

2. **Configure Kibana (`/etc/kibana/kibana.yml`):**

   ```
   sudo nano /etc/kibana/kibana.yml
   ```

   Make these critical changes:

   ```
   server.port: 5601
   server.host: "0.0.0.0" # To listen on all interfaces, or your server's IP

   # Elasticsearch connection
   elasticsearch.hosts: ["https://localhost:9200"] # Connect to local ES via HTTPS
   elasticsearch.username: "kibana_system" # Built-in user for Kibana
   elasticsearch.password: "YOUR_KIBANA_SYSTEM_PASSWORD" # Password you set earlier

   # SSL configuration for Kibana to trust Elasticsearch
   elasticsearch.ssl.certificateAuthorities: [ "/etc/elasticsearch/certs/http_ca.crt" ]
   # If Kibana and ES certs have CN/SAN mismatches, you might need:
   # elasticsearch.ssl.verificationMode: certificate # or 'none' (less secure)

   # Optional: If you want Kibana itself to serve HTTPS (if not using Nginx for SSL)
   # server.ssl.enabled: true
   # server.ssl.certificate: /path/to/kibana_server.crt
   # server.ssl.key: /path/to/kibana_server.key
   ```

   - **`elasticsearch.password`**: Use the password you set for the `kibana_system` user.
   - **`elasticsearch.ssl.certificateAuthorities`**: This tells Kibana to trust the CA that signed your Elasticsearch HTTP certificate. Path should be to `http_ca.crt` (or whatever you named your ES HTTP CA file) on the server.

3. **Start and Enable Kibana:**

   ```
   sudo systemctl daemon-reload
   sudo systemctl enable kibana.service
   sudo systemctl start kibana.service
   sudo systemctl status kibana.service # Check it's active (running)
   ```

   Access Kibana at `http://209.97.172.92:5601` (or via Nginx if you set that up).

D. Nginx Reverse Proxy for Kibana (Optional but Recommended for Production)
(As per our earlier discussion, this provides an additional security layer, SSL termination, and easier access). Refer to earlier parts of our conversation for the Nginx setup steps (installing Nginx,
`htpasswd`, self-signed cert or Let's Encrypt, and Nginx config for reverse proxy).

E. Logstash Installation and Configuration (on Main Server: `209.97.172.92`)

1. **Install Logstash (uses the same Elastic repository):**

```
sudo apt install logstash
```

2. **Create Logstash SSL Certificates for Beats Input:**
   These are the certificates Filebeat will use to connect securely to Logstash.

```
# On Logstash server
sudo mkdir -p /etc/logstash/certs
# (Assuming you already have ca.crt, logstash.crt, logstash.key in this directory
# from our previous steps. If not, generate them using OpenSSL as discussed before.
# ca.crt is the CA Filebeat will trust.
# logstash.crt is the server cert Logstash Beats input will use.
# logstash.key is its private key.)
# Ensure correct ownership and permissions:
sudo chown -R logstash:logstash /etc/logstash/certs
sudo chmod 600 /etc/logstash/certs/logstash.key
sudo chmod 644 /etc/logstash/certs/logstash.crt /etc/logstash/certs/ca.crt
```

3. **Configure Logstash Pipeline (`/etc/logstash/conf.d/01-main-pipeline.conf`):**
   This is the file you've been working on with inputs, filters (for privacy techniques), and outputs. Use the latest version you have that includes your desired filters.

   Crucial parts for connection:

```
input {
 beats {
  port => 5044
  ssl_enabled => true # Use ssl_enabled
  ssl_certificate => "/etc/logstash/certs/logstash.crt"
  ssl_key => "/etc/logstash/certs/logstash.key"
 }
}

filter {
 # Your GROK, MUTATE (gsub for redaction), FINGERPRINT (tokenization, integrity) filters here
 # as per your "Balancing Security and Privacy in SIEM logs" [cite: 1] objectives.
}

output {
 elasticsearch {
  hosts => ["https://localhost:9200"]
  user => "elastic" # TEMPORARY - For production, create a less privileged user
  password => "YOUR_ELASTIC_PASSWORD" # Replace with your actual elastic password
  ssl_enabled => true # Use ssl_enabled
  ssl_certificate_authorities => ["/etc/elasticsearch/certs/http_ca.crt"] # ES CA
  index => "filebeat-%{+YYYY.MM.dd}" # Or your preferred index name
 }
 # For debugging:
 # stdout { codec => rubydebug }
}
```

   - **Replace** `YOUR_ELASTIC_PASSWORD` with the actual password for the `elastic` user (or the dedicated Logstash user if you create one).

4. **Test Logstash Configuration:**

```
sudo /usr/share/logstash/bin/logstash --config.test_and_exit -f /etc/logstash/conf.d/01-main-pipeline.conf
```

5. **Start and Enable Logstash:**

```
sudo systemctl daemon-reload
sudo systemctl enable logstash.service
```

```
sudo systemctl start logstash.service
sudo systemctl status logstash.service # Check it's active (running)
```

F. Filebeat Installation and Configuration (on Agent Server, e.g., `152.42.191.42`)

1. **Install Filebeat:**

   ```
   # On Agent Server (e.g., 152.42.191.42)
   curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-8.18.1-amd64.deb # Or your chosen compatible 8.x version
   sudo dpkg -i filebeat-8.18.1-amd64.deb
   ```

2. **Copy Logstash Beats CA Certificate to Agent Server:**

   - **From Logstash Server (`209.97.172.92`):** `cat /etc/logstash/certs/ca.crt` → Copy output.
   - **On Agent Server (`152.42.191.42`):**

     ```
     sudo mkdir -p /etc/filebeat/certs/
     sudo nano /etc/filebeat/certs/logstash-ca.crt # Paste copied CA content
     sudo chmod 644 /etc/filebeat/certs/logstash-ca.crt
     ```

3. **Configure Filebeat (`/etc/filebeat/filebeat.yml` on Agent Server):**
   Key sections:

   ```
   sudo nano /etc/filebeat/filebeat.yml
   ```

   ```
   filebeat.inputs:
   - type: filestream
     id: my-agent-system-logs
     enabled: true
     paths:
       - /var/log/syslog
       - /var/log/auth.log
       # Add other paths as needed

   # Ensure output.elasticsearch is COMMENTED OUT
   # output.elasticsearch:
   #   hosts: ["localhost:9200"]

   output.logstash:
     hosts: ["209.97.172.92:5044"] # Your Logstash server IP
     ssl.enabled: true
     ssl.certificate_authorities: ["/etc/filebeat/certs/logstash-ca.crt"] # Path to Logstash CA on this agent
     ssl.verification_mode: certificate # Important for CN/SAN mismatches
   ```

4. **Test Filebeat Configuration on Agent:**

   ```
   sudo filebeat test config -e
   sudo filebeat test output -e # Should show successful connection to Logstash
   ```

5. **Start and Enable Filebeat on Agent:**

   ```
   sudo systemctl enable filebeat.service
   sudo systemctl start filebeat.service
   sudo systemctl status filebeat.service
   ```

For Fleet server pipeline and Testing in DEV Tool:

code :

```
PUT _ingest/pipeline/log-token-mask-pipeline
{
  "description": "Mask sensitive data like email, SSN, credit card",
  "processors": [
    {
      "script": {
        "description": "Mask email",
        "lang": "painless",
        "source": """
          if (ctx.user?.email != null) {
            String email = ctx.user.email;
            int atIndex = email.indexOf('@');
            if (atIndex > 1) {
              String masked = email.charAt(0) + "***" + email.substring(atIndex);
              ctx.user.email = masked;
            }
          }
        """
      }
    },
    {
      "script": {
        "description": "Mask SSN",
        "lang": "painless",
        "source": """
          if (ctx.user?.ssn != null && ctx.user.ssn.length() >= 11) {
            ctx.user.ssn = "***-**-" + ctx.user.ssn.substring(7);
          }
        """
      }
    },
    {
      "script": {
        "description": "Mask credit card",
        "lang": "painless",
        "source": """
          if (ctx.credit_card?.number != null && ctx.credit_card.number.length() >= 4) {
            ctx.credit_card.number = "****-****-****-" + ctx.credit_card.number.substr
          }
```

```
      """
     }
    },
    {
     "set": {
      "field": "ingested_at",
      "value": "{{_ingest.timestamp}}"
     }
    }
   ]
}
```

Now run our test:

```
POST test-index/_doc?pipeline=log-token-mask-pipeline
{
  "user": {
    "email": "user@example.com",
    "ssn": "123-45-6789"
  },
  "credit_card": {
    "number": "4111111111111111"
  }
}
```

Then Search:

```
GET test-index/_search
```

Output:

```
{
  "_index": "test-index",
  "_id": "_-BJ-JYBAYoxZqVxUpsA",
  "_score": 1,
  "_source": {
    "credit_card": {
      "number": "****-****-****-1111"
    },
    "ingested_at": "2025-05-22T13:57:56.695122994Z",
    "user": {
      "email": "u***@example.com",
      "ssn": "***-**-6789"
    }
  }
},
```