**Elasticsearch Watchers - Step-by-Step Learning Guide**

**🧠 What is a Watcher?**

A **Watcher** in Elasticsearch is a feature provided by the **X-Pack Alerting** module that allows you to **monitor data** and **trigger actions** when certain conditions are met (e.g., send an alert when an error rate goes high).

**✅ Prerequisites**

1. ELK Stack installed and running (Elasticsearch + Kibana).
2. X-Pack enabled (comes bundled with the default Elasticsearch distribution).
3. Kibana access to create and manage watchers visually (or use Dev Tools for API).

**📂 Components of a Watch**

A Watch consists of:

* **Trigger**: When the watch should run (e.g., every 5 minutes).
* **Input**: What data should be queried.
* **Condition**: When to trigger the action (e.g., more than 5 errors).
* **Action**: What to do (e.g., send email, log, webhook).
* **Transform** (optional): Manipulate data before action.

**🧪 Step-by-Step Example: Create a Watcher Using Kibana**

**Step 1: Open Kibana → Stack Management → Watcher**

* Click on “**Create advanced watch**” (for full control).

**Step 2: Define the Watch**

Here’s a sample watch that monitors logs for errors:

json

CopyEdit

{

"trigger": {

"schedule": {

"interval": "1m"

}

},

"input": {

"search": {

"request": {

"indices": [

"filebeat-\*"

],

"body": {

"query": {

"match": {

"log.level": "error"

}

}

}

}

}

},

"condition": {

"compare": {

"ctx.payload.hits.total.value": {

"gt": 0

}

}

},

"actions": {

"log\_error": {

"logging": {

"text": "Found {{ctx.payload.hits.total.value}} error logs in the last 1 minute."

}

}

}

}

**Step 3: Save and Activate**

* Give it a name: e.g., watch\_error\_logs.
* Save and activate the watch.

**Step 4: View Watch Results**

* Go to the “Watcher” UI.
* Click on the watch name → See history.
* You can manually simulate it too.

**🛠️ Simulate a Watch (for Testing)**

Use the **Dev Tools Console**:

json

CopyEdit

POST \_watcher/watch/\_execute

{

"watch": {

"trigger": {

"schedule": {

"interval": "1m"

}

},

"input": {

"simple": {

"status": "testing"

}

},

"condition": {

"always": {}

},

"actions": {

"log\_action": {

"logging": {

"text": "This is a test action. Status: {{ctx.payload.status}}"

}

}

}

}

}

**📤 Adding Email Notifications**

You need to configure elasticsearch.yml:

yaml

CopyEdit

xpack.notification.email.account:

work:

profile: standard

smtp:

auth: true

starttls.enable: true

host: smtp.gmail.com

port: 587

user: your-email@gmail.com

password: your-password

Then add an email action:

json

CopyEdit

"actions": {

"send\_email": {

"email": {

"to": "admin@example.com",

"subject": "Error logs detected",

"body": "There are {{ctx.payload.hits.total.value}} error logs."

}

}

}

**🧼 Cleanup (Optional)**

To delete a watch:

bash

CopyEdit

DELETE \_watcher/watch/watch\_error\_logs

**🧠 Extra Tips for Teaching**

* **Start with simple log alerts** (errors, warnings).
* **Demonstrate both UI and API** usage.
* **Explain real-world scenarios** like CPU threshold alerts, failed jobs, disk space, etc.
* **Simulate logs** using logger or curl so students can see live data changes.
* Ask students to build their own watcher as homework.