**Step 1: Update packages**

sudo apt update -y

sudo apt upgrade -y

**Step 2: Install prerequisites**

sudo apt install -y apt-transport-https ca-certificates curl software-properties-common

**Step 3: Add Docker’s official GPG key**

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

**Step 4: Add Docker repository**

echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

**Step 5: Update packages again**

sudo apt update -y

**Step 6: Install Docker**

sudo apt install -y docker-ce docker-ce-cli containerd.io

**Step 7: Verify Docker installation**

docker --version

Expected output:

Docker version 24.x.x, build ...

**Step 8: Run Docker without sudo (optional)**

1. Add your user to the docker group:

sudo usermod -aG docker $USER

1. Log out and log back in (or restart SSH session).
2. Test:

docker run hello-world

**Step 9: Install Docker Compose**

sudo curl -L "https://github.com/docker/compose/releases/download/v2.23.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

docker-compose --version

Expected output:

Docker Compose version v2.23.1

**Step 1: Create a Filebeat Docker Compose service**

Inside your elk-light directory, edit your docker-compose.yml to add **Filebeat**:

version: '3.8'

services:

elasticsearch:

image: docker.elastic.co/elasticsearch/elasticsearch:8.9.0

container\_name: elasticsearch

environment:

- discovery.type=single-node

- xpack.security.enabled=false

- ES\_JAVA\_OPTS=-Xms512m -Xmx512m

ulimits:

memlock:

soft: -1

hard: -1

volumes:

- es\_data:/usr/share/elasticsearch/data

ports:

- "9200:9200"

kibana:

image: docker.elastic.co/kibana/kibana:8.9.0

container\_name: kibana

environment:

ELASTICSEARCH\_HOSTS: "http://elasticsearch:9200"

ports:

- "5601:5601"

depends\_on:

- elasticsearch

filebeat:

image: docker.elastic.co/beats/filebeat:8.9.0

container\_name: filebeat

user: root

volumes:

- ./filebeat/filebeat.yml:/usr/share/filebeat/filebeat.yml:ro

- /var/log:/var/log:ro # Mount EC2 logs

depends\_on:

- elasticsearch

volumes:

es\_data:

driver: local

**Step 2: Create Filebeat config file**

Create folder:

mkdir filebeat

Create filebeat/filebeat.yml:

filebeat.inputs:

- type: log

enabled: true

paths:

- /var/log/\*.log # Collect all standard EC2 logs

- /var/log/syslog

- /var/log/auth.log

output.elasticsearch:

hosts: ["http://elasticsearch:9200"]

username: "elastic"

password: "" # leave empty if xpack.security.enabled=false

Notes:

* Adjust paths to the logs you want to collect.
* If you enable security in Elasticsearch, set proper username and password.

**Step 3: Run ELK + Filebeat**

docker-compose up -d

Check Filebeat logs:

docker logs -f filebeat

You should see **logs being read and shipped** to Elasticsearch.

**Step 4: Verify in Kibana**

1. Go to http://<EC2-IP>:5601 → **Discover** tab.
2. Look for indices like filebeat-\*.
3. You should see your EC2 logs flowing in real-time.

**Lightweight ELK Docker Compose**

Create a directory:

mkdir elk-light && cd elk-light

Create a file named docker-compose.yml:

version: '3.8'

services:

elasticsearch:

image: docker.elastic.co/elasticsearch/elasticsearch:8.9.0

container\_name: elasticsearch

environment:

- discovery.type=single-node

- xpack.security.enabled=false

- ES\_JAVA\_OPTS=-Xms512m -Xmx512m # Reduce memory

ulimits:

memlock:

soft: -1

hard: -1

volumes:

- es\_data:/usr/share/elasticsearch/data

ports:

- "9200:9200"

kibana:

image: docker.elastic.co/kibana/kibana:8.9.0

container\_name: kibana

environment:

ELASTICSEARCH\_HOSTS: "http://elasticsearch:9200"

ports:

- "5601:5601"

depends\_on:

- elasticsearch

volumes:

es\_data:

driver: local

**Why this is lightweight:**

1. **Elasticsearch 512 MB heap** (-Xms512m -Xmx512m) instead of the default 2 GB.
2. **No Logstash** – Logstash is heavy. For testing, you can skip it or use Filebeat later.
3. **No security enabled** – avoids extra memory overhead.
4. **Single-node Elasticsearch** – minimal setup.

Note: This is suitable for **testing or learning**. For production, use the full ELK stack with proper memory and security.

**Steps to run**

1. Make sure Docker has **enough disk space** (at least 10 GB free).
2. Run Docker Compose:

docker-compose up -d

1. Check status:

docker ps

* Elasticsearch → http://<EC2-IP>:9200
* Kibana → http://<EC2-IP>:5601

1. Test Elasticsearch:

curl http://localhost:9200