# Lab Manual: Centralized Logging with EFK Stack on Kubernetes

## Prerequisites

* Kubernetes cluster with at least **3 nodes**
* kubectl configured to access your cluster
* [Helm](https://helm.sh/) installed (helm version)

## Step 1: Create a Namespace for Logging

kubectl create namespace logging

## Step 2: Deploy Elasticsearch Using Helm

helm repo add elastic https://helm.elastic.co  
helm repo update  
  
helm install elasticsearch elastic/elasticsearch \  
 --namespace=logging \  
 --set replicas=1 \  
 --set resources.requests.cpu="500m" \  
 --set resources.requests.memory="2Gi" \  
 --set resources.limits.cpu="1" \  
 --set resources.limits.memory="4Gi"

**Note:** For production, increase replicas and resources.

## Step 3: Deploy Kibana Using Helm

helm install kibana elastic/kibana \  
 --namespace=logging \  
 --set service.type=NodePort

Check Kibana NodePort:

kubectl get svc -n logging | grep kibana

Example output:

kibana NodePort 10.0.x.x <none> 5601:32501/TCP 1m

Access Kibana at: http://<worker-node-ip>:<NodePort> (e.g., http://<worker-node-ip>:32501)

## Step 4: Deploy Fluentd as a DaemonSet

Fluentd collects logs from every node and forwards to Elasticsearch.

### Fluentd ConfigMap and DaemonSet

Download and apply the [official Fluentd DaemonSet manifest](https://raw.githubusercontent.com/fluent/fluentd-kubernetes-daemonset/master/fluentd-daemonset-elasticsearch-rbac.yaml):

kubectl apply -f https://raw.githubusercontent.com/fluent/fluentd-kubernetes-daemonset/master/fluentd-daemonset-elasticsearch-rbac.yaml -n logging

*Fluentd connects to Elasticsearch via service discovery:*

* elasticsearch-master.logging.svc.cluster.local:9200

(If needed, adjust FLUENT\_ELASTICSEARCH\_HOST and FLUENT\_ELASTICSEARCH\_PORT env vars in the DaemonSet spec.)

## Step 5: Deploy the NGINX Application

**nginx-deployment.yaml**

apiVersion: apps/v1  
kind: Deployment  
metadata:  
 name: nginx  
 namespace: default  
spec:  
 replicas: 3  
 selector:  
 matchLabels:  
 app: nginx  
 template:  
 metadata:  
 labels:  
 app: nginx  
 spec:  
 containers:  
 - name: nginx  
 image: nginx:latest  
 ports:  
 - containerPort: 80  
---  
apiVersion: v1  
kind: Service  
metadata:  
 name: nginx  
 namespace: default  
spec:  
 selector:  
 app: nginx  
 ports:  
 - port: 80  
 targetPort: 80  
 type: NodePort

Apply the deployment and service:

kubectl apply -f nginx-deployment.yaml

Find the NGINX NodePort:

kubectl get svc nginx

## Step 6: Generate Logs

Send traffic to NGINX to generate logs:

curl http://<worker-node-ip>:<nodeport>

## Step 7: Access Kibana

* Open Kibana in your browser:  
  http://<worker-node-ip>:<kibana-nodeport>
* Click **Discover**
* If prompted, set up index pattern: logstash-\* or \*
* Time field: @timestamp

## Step 8: Search and Visualize NGINX Logs

* In **Kibana Discover**, search:  
  kubernetes.labels.app: nginx
* Filter by namespace, pod, severity, etc.

**Sample Queries:**

* All NGINX logs:
* kubernetes.labels.app: nginx
* Logs by severity:
* kubernetes.labels.app: nginx AND log: error
* By pod name:
* kubernetes.pod\_name: nginx-\*

## Step 9: Create a Kibana Dashboard

1. Go to **Dashboard** → “Create new dashboard”
2. Add visualizations: log count per pod, logs over time, logs by level
3. Save dashboard for team use

## Step 10: Set Up Alerts (Optional)

1. In Kibana: “Stack Management” → “Rules and Connectors”
2. Create a rule: log level “error” for NGINX pods
3. Set action: email/Slack/other connector

## Step 11: Clean Up

kubectl delete ns logging  
kubectl delete deploy nginx  
kubectl delete svc nginx

## Sample Fluentd Filter for Tagging (Optional)

<filter kubernetes.\*\*>  
 @type record\_transformer  
 enable\_ruby  
 <record>  
 app\_name ${record["kubernetes"]["labels"]["app"]}  
 namespace ${record["kubernetes"]["namespace\_name"]}  
 pod ${record["kubernetes"]["pod\_name"]}  
 </record>  
</filter>

Edit the Fluentd ConfigMap in the DaemonSet if customization is needed.

## Summary Table

| Step | Task | Key Command / File |
| --- | --- | --- |
| 1 | Create Namespace | kubectl create namespace logging |
| 2 | Deploy Elasticsearch (Helm) | helm install elasticsearch ... |
| 3 | Deploy Kibana (Helm) | helm install kibana ... |
| 4 | Deploy Fluentd DaemonSet | kubectl apply -f fluentd-daemonset-elasticsearch... |
| 5 | Deploy NGINX | kubectl apply -f nginx-deployment.yaml |
| 6 | Generate Logs | curl http://<worker-node-ip>:<nodeport> |
| 7 | Access Kibana | Open browser to http://<node-ip>:<kibana-nodeport> |
| 8 | Search/Visualize in Kibana | Use “Discover”, build queries, dashboards |

# End of Lab