Logstash Installation & Setup

Basic Installation

For the basic Logstash deployment, all we have to do is to use the official Helm chart by Elastic:

1. add the Elastic repo: `helm repo add elastic https://helm.elastic.co`
2. install Logstash chart: `helm install logstash elastic/logstash`
3. verify the Logstash pod is running with `kubectl get po | grep logstash`

The Logstash should be up and running, but it's not doing much — by default, it's configured with a single pipeline called `main` that listens for beats data on port 5044 and outputs them to stdout.

Configuration

We will use Kubernetes config maps to configure our Logstash deployment. For starters, we will create a new Helm chart, that uses the Logstash Helm chart as a subchart:

1. create a new chart: `helm create <chart name>` # Ex logstash_parent
2. clean up pre-created templates: `rm -rf <chart name>/templates/*`
3. open the `<chart name>/Chart.yaml` in your favorite text editor and add the `dependencies` field:

```
apiVersion: v2
name: logstash-parent
description: A parent chart for Logstash deployment

type: application
version: 0.1.0
dependencies:
  - name: logstash
    version: '8.15.0' # this must match the version of docker
```

```
image of ur elasticsearch and kibana
    repository: '@elastic'
```

The `dependencies` section defines we are using a Logstash chart as a subchart.

Now we will add our custom configuration by adding two config maps into the `templates` directory of our newly created chart, first for base logstash configuration that will replace the default config files:

logstash-config.yaml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: logstash-config
data:
  logstash.yml: |

    # Add any configuration you need

    http.host: "0.0.0.0"
    http.port: 9600
    xpack.monitoring.elasticsearch.hosts: [ "https://elastics
earch-master:9200" ]
    xpack.monitoring.elasticsearch.username: elastic
    xpack.monitoring.elasticsearch.password: JWp34zQWi7jlTz5Q
#Change this based on ur password
    xpack.monitoring.elasticsearch.ssl.certificate_authority:
/usr/share/logstash/config/certs/ca.crt
    xpack.monitoring.enabled: true

  pipelines.yml: |

    - pipeline.id: main
      path.config: "/usr/share/logstash/pipeline/main*.conf"
```

```
 # This two lines are for the pfelk dashboard uncomment then
if you want to use that
  #  - pipeline.id: pfelk
  #    path.config: "/etc/pfelk/conf.d/*.pfelk"


    #   pipeline.ecs_compatibility: v8 #Disable if not runnin
g Elastic v8
    #- pipeline.id: audit
    #  path.config: "/usr/share/logstash/pipeline/audit.conf"
```

The second config map will contain the pipeline definition:

pipelines.yaml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: pipeline-config
data:
  main_01_input.conf: |

    input {
      beats {
        port => 5044
      }

      # SNMP part will be unncomented later once we aim to us
e it
    #  snmp {
    #    get => ["1.3.6.1.2.1.1.3.0"]
    #    hosts => [{host => "udp:10.8.9.184/161" community =>
"public"}]
    #  }

    }
```

```
  main_02_filter.conf: |

    filter {
      json {
        id => "message_json_parse"
        source => "message"
        target => "json_log"
      }
    }
  main_03_output.conf: |

    output {
      elasticsearch {
        hosts => ["https://elasticsearch-master:9200"]
        ssl => true
        cacert => "/usr/share/logstash/config/certs/ca.crt"
        user => "elastic"   # Replace with actual username an
d password if authentication is required
        password => "JWp34zQWi7jlTz5Q"
      }
      stdout {
        codec => rubydebug
      }
    }
  audit.conf: |

    input {
      syslog {
        port => 12345
      }
    }
    output {
      stdout {
        codec => rubydebug
```

```
      }
    }
```

Now we have to tell the Logstash subchart to mount these Config maps. To do that, we have to modify `values.yaml` in the chart root directory:

```
# Default values for logstash-parent.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.
replicaCount: 1

image:
  repository: nginx
  pullPolicy: IfNotPresent
  tag: ""

imagePullSecrets: []
nameOverride: ""
fullnameOverride: ""

logstash:
  image: "docker.elastic.co/logstash/logstash"
  imageTag: "8.15.0"
  plugins:
    - logstash-input-snmp
  extraVolumes:
    - name: logstash-config
      configMap:
        name: logstash-config
        items:
          - key: logstash.yml
            path: logstash.yml
          - key: pipelines.yml
            path: pipelines.yml

    - name: pipelines
```

```
    configMap:
      name: pipeline-config

  - name: ca-cert-volume
    secret:
      secretName: elasticsearch-master-certs
      # Uncomment this if you want to use pfelk dashboard
  #- name: logstash-config-volume
  #  persistentVolumeClaim:
  #    claimName: logstash-pvc

extraVolumeMounts:
  - name: logstash-config
    mountPath: /usr/share/logstash/config/logstash.yml
    subPath: logstash.yml

  - name: logstash-config
    mountPath: /usr/share/logstash/config/pipelines.yml
    subPath: pipelines.yml

  - name: pipelines
    mountPath: /usr/share/logstash/pipeline

  - name: ca-cert-volume
    mountPath: /usr/share/logstash/config/certs/ca.crt
    subPath: ca.crt

   # Uncomment this if you want to use pfelk dashboard
 # - name: logstash-config-volume
 #   mountPath: /usr/share/logstash/new-pipeline
 #   mountPath: /etc/pfelk # if you are using pfelk
    #subPath: .

# Manage ports here
service:
  type: NodePort
```

```
    ports:
      - name: beats
        port: 5044
        protocol: TCP
        targetPort: 5044
      - name: monitoring
        port: 9600
        protocol: TCP
        targetPort: 9600
      - name: syslog
        port: 5140
        protocol: TCP
        targetPort: 5140
        nodePort: 30006

  headless:
    annotations: {}
    ports:
      - name: beats
        port: 5044
        protocol: TCP
        targetPort: 5044
      - name: monitoring
        port: 9600
        protocol: TCP
        targetPort: 9600
      - name: syslog
        port: 5140
        protocol: TCP
        targetPort: 5140
  ports:
    - name: beats
      containerPort: 5044
      protocol: TCP
    - name: monitoring
      containerPort: 9600
```

```
      protocol: TCP
    - name: syslog
      containerPort: 5140
      protocol: TCP
serviceAccount:
  create: true
  automount: true
  annotations: {}
  name: ""

podAnnotations: {}
podLabels: {}

podSecurityContext: {}
securityContext: {}

ingress:
  enabled: false
  className: ""
  annotations: {}
  hosts:
    - host: chart-example.local
      paths:
        - path: /
          pathType: ImplementationSpecific
  tls: []

resources: {}

livenessProbe:
  httpGet:
    path: /
    port: http
readinessProbe:
  httpGet:
    path: /
```

```
    port: http

autoscaling:
  enabled: false
  minReplicas: 1
  maxReplicas: 100
  targetCPUUtilizationPercentage: 80

volumes: []
volumeMounts: []

nodeSelector: {}

tolerations: []

affinity: {}
```

Mount Volume for for ca.crt certificate

The logstash require to authorize the certificate

- Display all certificates

```
kubectl get secrets -n elasticsearch
```

- Locate where the certificate is located
  - Usually ends with -certs
  - To display the secrets contents

```
kubectl get secret elasticsearch-master-certs -n elasticsearc
h -o jsonpath='{.data}' | jq -r 'keys[]’
```

Add the Volume & Volume Mount (Note: This being added to the code above)

```
extraVolumes:
    - name: ca-cert-volume
      secret:
        secretName: elasticsearch-master-certs

  extraVolumeMounts:
    - name: ca-cert-volume
      mountPath: /usr/share/logstash/config/certs/ca.crt
      subPath: ca.crt
```

Verify Mounting Done Successfully

```
 kubectl exec -it -n elasticsearch logstash-logstash-0 -- ls
/usr/share/logstash/config/certs
```

This should print out the content of the /usr/share/logstash/config/certs (Certificate must be mounted)

Mount /etc/pfelk folder for PFELK

Create PV and PVC

logstash-pv.yaml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: logstash-pv
  namespace: elasticsearch
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteMany
  hostPath:
    path: /etc/pfelk/
```

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: logstash-pvc
  namespace: elasticsearch
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
```

Apply the PV and PVC

```
kubectl apply -f logstash-pv.yaml
kubectl apply -f logstash-pvc.yaml
kubectl get pv logstash-pv
```

After these changes, the chart is ready for deployment:

1. Uninstall previous logstash chart installation: `helm uninstall logstash`
2. Let Helm download logstash subchart: `helm dep build <chart name>/`
3. Install new chart: `helm install logstash <chart name>/` -n namespace

To verify the health of the logstash pod

```
kubectl port-forward service/logstash-logstash 9600:9600 -n e
lasticsearch
```

Then from other terminal

```
curl -XGET 'http://localhost:9600/?pretty'
```

To change logstash config file and apply change

```
kubectl apply -f pipeline-config.yaml -n elasticsearch
```

```
kubectl rollout restart statefulset/logstash-logstash -n elas
ticsearch
```