Logging and monitoring Kubernetes Cluster using EFK and Prometheus

Table of Contents

Steps describing how to enable logging and monitoring kubernetes cluster using EFK(Elasctisearch Fluentbit Kibana) and Prometheus respectively.

Prerequiste

- Up and running kubernetes cluster.
- We will be using Helm charts, hence helm chart must be available.

Preface

- These steps are validated against kubernetes cluster on VirtualBox, with minimal hardware, hence running Elasticsearch in minimalistic setup.
- Since I'm using VirtualBox, will be using peristent volume of type local.
- Using `stable/elasticsearch` repository instead of elasticsearch official from here
- Validated against elasticsearch version `6.7.2`

Setup EFK stack.

- Update Helm repository.

```
$ helm repo update
```

- We will be running 2 master node (though Ideally quorum would be an odd number), 1 client(coordinator) node and 1 data node. Coordinator node does not require volume, thus we need 3 PV. Create directories for 3 PVs.

```
$ sudo mkdir -p /opt/kubernetes/data/elasticsearch/es0
$ sudo mkdir -p /opt/kubernetes/data/elasticsearch/es1
$ sudo mkdir -p /opt/kubernetes/data/elasticsearch/es2
```

Since I've one worker node cluster, creating all the es nodes in same worker node hence 3 directories in the same server.

- Change permission of the directories

```
$ sudo chmod -R g+w /opt/kubernetes/data
```

- Create a peristent volume and storage class to bind the pvs. from below yaml file, using command `kubectl apply -f <file-name>.yaml`

```
# Source: efk-stack/templates/storage.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: efk-stack-local-storage
  namespace: logging
  labels:
    app: efk-stack
    chart: efk-stack-0.1.0
    release: efk-stack
    heritage: Tiller
provisioner: kubernetes.io/no-provisioner
volumeBindingMode: Immediate
#volumeBindingMode: WaitForFirstConsumer
# Supported policies: Delete, Retain
reclaimPolicy: Delete

---

apiVersion: v1
kind: PersistentVolume
metadata:
  name: efk-stack-pv-0
  namespace: logging
  labels:
    app: efk-stack
    chart: efk-stack-0.1.0
    release: efk-stack
    heritage: Tiller
```

```
spec:
  capacity:
    storage: 4Gi
  # volumeMode field requires BlockVolume Alpha feature gate to be enabled.
  volumeMode: Filesystem
  accessModes:
  - ReadWriteOnce
  persistentVolumeReclaimPolicy: Delete
  storageClassName: efk-stack-local-storage
  local:
    path: /opt/kubernetes/data/elasticsearch/es0
  nodeAffinity:
    required:
      nodeSelectorTerms:
      - matchExpressions:
        - key: disk-available
          operator: In
          values:
          - "true"
---

apiVersion: v1
kind: PersistentVolume
metadata:
  name: efk-stack-pv-1
  namespace: logging
  labels:
    app: efk-stack
    chart: efk-stack-0.1.0
    release: efk-stack
    heritage: Tiller
spec:
  capacity:
    storage: 4Gi
  # volumeMode field requires BlockVolume Alpha feature gate to be enabled.
  volumeMode: Filesystem
  accessModes:
  - ReadWriteOnce
  persistentVolumeReclaimPolicy: Delete
  storageClassName: efk-stack-local-storage
  local:
    path: /opt/kubernetes/data/elasticsearch/es1
  nodeAffinity:
    required:
      nodeSelectorTerms:
      - matchExpressions:
        - key: disk-available
          operator: In
          values:
          - "true"
```

```yaml
---

apiVersion: v1
kind: PersistentVolume
metadata:
  name: efk-stack-pv-2
  namespace: logging
  labels:
    app: efk-stack
    chart: efk-stack-0.1.0
    release: efk-stack
    heritage: Tiller
spec:
  capacity:
    storage: 4Gi
  # volumeMode field requires BlockVolume Alpha feature gate to be enabled.
  volumeMode: Filesystem
  accessModes:
  - ReadWriteOnce
  persistentVolumeReclaimPolicy: Delete
  storageClassName: efk-stack-local-storage
  local:
    path: /opt/kubernetes/data/elasticsearch/es2
  nodeAffinity:
    required:
      nodeSelectorTerms:
      - matchExpressions:
        - key: disk-available
          operator: In
          values:
          - "true"
```

PV cannot be deleted before PVCs are deleted, PVCs need to be deleted manaully.

- Download elasticsearch charts and extract.

```
$  helm fetch stable/elasticsearch --untar
```

- change directory

```
$ cd elasticsearch
```

- Open values.yaml and update values as described below.
  - Make sure image name is `docker.elastic.co/elasticsearch/elasticsearch-oss` if you want to use opensourced version.
  - update elasticsearch tag to `6.7.2`

```
image:
  repository: "docker.elastic.co/elasticsearch/elasticsearch-oss"
  tag: "6.7.2"
  pullPolicy: "IfNotPresent"
```

- Change **client** and **data** node replica count to 1

```
client:
  name: client
  replicas: 1
  serviceType: ClusterIP
```

- (Optional)Comment out CPU count both for **client**, **master** and **data** node

```
  resources:
    limits:
      # cpu: "1"
      # memory: "1024Mi"
    requests:
      cpu: "25m"
      memory: "512Mi"
```

This is inline to my hardware constarint, if there is enough CPU you can leave as is.

- Change Master replica count to 2.

```
master:
  name: master
  exposeHttp: false
  replicas: 2
  heapSize: "512m"
```

- On **data** and **master** node section update disk size to 4Gi as we have created pv with 4Gi and uncomment storageClassName and provide the stoage class name we created earlier.

```
  persistence:
    enabled: true
    accessMode: ReadWriteOnce
    name: data
    size: "4Gi"
    storageClass: "efk-stack-local-storage"
```

- Create elasticsearch cluster using helm command.

```
$  helm install . --name efk-stack-elastic --namespace logging --debug
```

Verification

- Wait untill 4 Pods comes up and state changes to Ready.

```
$ kubectl get po -n logging
```

- Create a busybox pod from below yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: busybox
  namespace: default
spec:
  containers:
  - name: busybox
    image: busybox
    command:
      - sleep
      - "3600"
    imagePullPolicy: IfNotPresent
  restartPolicy: Always
```

- Now execute below command to verify elastic search is up and running.

```
$ kubectl exec busybox -- wget  http://efk-stack-elastic-elasticsearch-
client.logging:9200 \
    -O - \
    -S
```

output:

```
Connecting to efk-stack-elastic-elasticsearch-client.logging:9200 (10.111.238.225
:9200)
  HTTP/1.1 200 OK
  content-type: application/json; charset=UTF-8
  content-length: 539

-                    100% |*******************************|   539   0:00:00 ETA

{
  "name" : "efk-stack-elastic-elasticsearch-client-cf8579b94-zjxxr",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "1n-mUFcJT4C-XPrjb98HHg",
  "version" : {
    "number" : "6.7.2",
    "build_flavor" : "oss",
    "build_type" : "docker",
    "build_hash" : "56c6e48",
    "build_date" : "2019-04-29T09:05:50.290371Z",
    "build_snapshot" : false,
    "lucene_version" : "7.7.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

Setup elasticsearch curator

- Download curator

```
$ helm fetch stable/elasticsearch-curator --untar
```

- change directory.

```
cd elasticsearch-curator/
```

- Add elasticsearch client service as host at `confiMaps` → `config_yml` → `hosts` as shown below in values.yaml

```
  config_yml: |-
    ---
    client:
      hosts:
        - efk-stack-elastic-elasticsearch-client
      port: 9200
```

- Install curator using below command.

```
$ helm install . --name efk-stack-curator --namespace logging
```

Setup Fluentbit

- Download Fluentbit

```
$  helm fetch stable/fluent-bit --untar
```

- change directory.

```
$  cd fluent-bit/
```

- Add elasticsearch client service as host at `backend` → `es` → `host` and `forward` type to `es` as shown below in values.yaml

```
backend:
  type: es
  forward:
    host: fluentd
    port: 24284
    tls: "off"
    tls_verify: "on"
    tls_debug: 1
    shared_key:
  es:
    host: efk-stack-elastic-elasticsearch-client
    port: 9200
    # Elastic Index Name
    index: kubernetes_cluster
    type: flb_type
    logstash_prefix: kubernetes_cluster
    replace_dots: "On"
    time_key: "@timestamp"
    # Optional username credential for Elastic X-Pack access
    http_user:
    # Password for user defined in HTTP_User
    http_passwd:
    # Optional TLS encryption to ElasticSearch instance
    tls: "off"
    tls_verify: "on"
    # TLS certificate for the Elastic (in PEM format). Use if tls=on and
tls_verify=on.
    tls_ca: ""
    # TLS debugging levels = 1-4
    tls_debug: 1
```

- Install fluentbit using below command.

```
$  helm install . --name efk-stack-fluent-bit --namespace logging
```

Verification

Execute below command to check if fluentbit is successfully started. Ready value should be 1

```
$ kubectl get ds -n logging
```

output

```
NAME                    DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR   AGE
efk-stack-fluent-bit    1         1         1       1            1           <none>
14s
```

Setup Kibana

- Download kibana

```
$  helm fetch stable/kibana --untar
```

- change directory.

```
$  cd kibana/
```

- Add elasticsearch client service as host at `files` → `kibana.yml` → `elasticsearch.hosts` and service type `NodePort` as shown below in values.yaml

```
files:
  kibana.yml:
    ## Default Kibana configuration from kibana-docker.
    server.name: kibana
    server.host: "0"
    ## For kibana < 6.6, use elasticsearch.url instead
    elasticsearch.hosts: http://efk-stack-elastic-elasticsearch-client:9200
```

- Install kibana

```
$  helm install . --name efk-stack-kibana --namespace logging
```

- Get the Kibana URL by executing below command

```
export KIBANA_NODE_PORT=$(kubectl get --namespace logging \
    -o jsonpath="{.spec.ports[0].nodePort}" services efk-stack-kibana)
export KIBANA_NODE_IP=$(kubectl get nodes --namespace logging \
    -o jsonpath="{.items[0].status.addresses[0].address}")
echo http://$KIBANA_NODE_IP:$KIBANA_NODE_PORT
```

Configure Kibana UI.

- Open the url obaained in previous section in a browser. Since its fresh installation page will be automatically redirected to management tab to create index Patterns. Add index pattern `kubernetes*` and click next.

  [Create Index Pattern] | *kibana_1.png*

- Select time filter feild as `@timestamp` and click `create index pattern`

  [Additional Index Pattern Settings] | *kibana_2.png*

- Now select `discovery` menu from side nav menu, already logs will start appearing., Now you can add/remove columns to display from Available fields by hovering on field and clicking add button, remove columns can be done by unselecting the fields from selected fields.

  [Manage columns] | *kibana_3.png*

- Log refresh rate and duration to show can be managed by selecting the `Auto refresh` button on top right corner and similarly the duration of the logs to display can be configured on date menu adjacent to `Auto refresh button`.

Setup Prometheus

- Download Prometheus operator

```
$ helm fetch stable/prometheus-operator --untar
```

- change directory.

```
$  cd prometheus-operator/
```

- Install prometheus operator

```
$  helm install . --name prometheus --namespace monitoring
```

- Edit grafana service from cluster ip to node port

```
$ kubectl edit svc -n monitoring  prometheus-grafana
```

- Get the grafan url by executing below command.

```
export GRAFANA_NODE_PORT=$(kubectl get --namespace monitoring \
    -o jsonpath="{.spec.ports[0].nodePort}" services  prometheus-grafana)
export GRAFANA_NODE_IP=$(kubectl get nodes --namespace monitoring \
    -o jsonpath="{.items[0].status.addresses[0].address}")
echo http://$GRAFANA_NODE_IP:$GRAFANA_NODE_PORT
```

Default username/password is admin/prom-operator

TODO

- Add Steps to enable spring boot logs as fields in FluentBit before sending to Elasticsearch.
- Add Steps to enable spring boot metrics to be fetch by Promethus using annotations.