k8s-EFK

EFK日志收集

elasticsearch+fluentd+kibana

收集宿主机日志 /var/log

https://github.com/kubernetes/kubernetes/tree/master/cluster/addons

https://github.com/kubernetes/kubernetes/tree/master/cluster/addons/fluentd-elasticsearch

```
cd /root/install-some-apps/efk
# 下载 https://github.com/kubernetes/kubernetes/tree/master/cluster/addons/fluentd-elasticsearch 这个目录下的
create-logging-namespace.yaml es-service.yaml es-statefulset.yaml fluentd-es-configmap.yaml fluentd-es-ds.yaml kibana-service.yaml kibana-deployment.yaml
```

```
# 1) create ns
kubectl create -f create-logging-namespace.yaml

# 2) create es
kubectl create -f es-service.yaml
kubectl create -f es-statefulset.yaml

# 取消ES健康检查(es-statefulset.yaml)，其他的按需改吧
         # livenessProbe:
         #   tcpSocket:
         #     port: transport
         #   initialDelaySeconds: 5
         #   timeoutSeconds: 10
         # readinessProbe:
         #   tcpSocket:
         #     port: transport
         #   initialDelaySeconds: 5
         #   timeoutSeconds: 10

# view po
kubectl get po -n logging

# view log
kubectl logs -f elasticsearch-logging-0 -n logging

# view service
kubectl get svc -n logging


# 3) create fluentd
```

```
kubectl create -f fluentd-es-configmap.yaml
kubectl create -f fluentd-es-ds.yaml
kubectl get po -n logging

# connect es
curl 10.96.41.190:9200/_cluster/health?pretty



# 4) create kibana
kreate create -f kibana-service.yaml
kubectl get po -n logging
kubectl get svc -n logging
  # web view http://10.4.7.107:31223

  # 如果访问有问题，kibana-deployment.yaml 关掉里面的访问模式proxy
  # 注释以下2行（kibana-deployment.yaml ）
  #- name: SERVER_BASEPATH
  # value: /api/v1/namespaces/kube-system/services/kibana-logging/proxy
```

参考：https://www.cnblogs.com/hsyw/p/14397700.html

Filebeat 收集容器内日志

elasticsearch+Filebeate+Logstash+kibana

zk+ kafka

参考： https://github.com/dotbalo/k8s/tree/master/fklek/7.x

```
# 先启动zk, kafka, 参考上面一章
helm install zookeeper -n public-service .
helm install kafka -n public-service --set zookeeper.enabled=false --set replicaCount=1
--set externalZookeeper.servers=zookeeper .

kubectl get svc -n public-service
```

filebeat-cm.yaml

```
cd /root/install-some-apps/efk
mkdir filebeat && cd filebeat
```

```
vim filebeat-cm.yaml
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: filebeatconf
data:
  filebeat.yml: |-
    filebeat.inputs:
    - input_type: log
      paths:
        - /data/log/*/*.log
      tail_files: true
      fields:
        pod_name: '${podName}'
        pod_ip: '${podIp}'
        pod_deploy_name: '${podDeployName}'
        pod_namespace: '${podNamespace}'
      tags: [test-filebeat]
    output.kafka:
      hosts: ["kafka:9092"]
      topic: "test-filebeat"
      codec.json:
        pretty: false
      keep_alive: 30s
```

```
kubectl create -f filebeat-cm.yaml -n public-service
```

logstash-cm.yaml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: logstash-configmap
data:
  logstash.yml: |
    http.host: "0.0.0.0"
    path.config: /usr/share/logstash/pipeline
  logstash.conf: |
    # all input will come from filebeat, no local logs
    input {
```

```
      kafka {
              enable_auto_commit => true
              auto_commit_interval_ms => "1000"
              bootstrap_servers => "kafka:9092"
              topics => ["test-filebeat"]
              codec => json
          }
    }
    output {
       stdout{ codec=>rubydebug}
       if [fields][pod_namespace] =~ "public-service" {
           elasticsearch {
             hosts => ["elasticsearch-logging:9200"]
             index => "%{[fields][pod_namespace]}-s-%{+YYYY.MM.dd}"
          }
       } else {
          elasticsearch {
             hosts => ["elasticsearch-logging:9200"]
             index => "no-index-%{+YYYY.MM.dd}"
          }
       }
    }
```

```
kubectl create -f logstash-cm.yaml -n public-service
```

logstash-service.yaml

```
kind: Service
apiVersion: v1
metadata:
  name: logstash-service
spec:
  selector:
    app: logstash
  ports:
  - protocol: TCP
    port: 5044
    targetPort: 5044
  type: ClusterIP
```

```
kubectl create -f logstash-service.yaml -n public-service
```

logstash-deploy.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: logstash-deployment
spec:
  selector:
    matchLabels:
      app: logstash
  replicas: 1
  template:
    metadata:
      labels:
        app: logstash
    spec:
      containers:
      - name: logstash
        image: logstash:7.4.2
        ports:
        - containerPort: 5044
        volumeMounts:
          - name: config-volume
            mountPath: /usr/share/logstash/config
          - name: logstash-pipeline-volume
            mountPath: /usr/share/logstash/pipeline
      volumes:
      - name: config-volume
        configMap:
          name: logstash-configmap
          items:
            - key: logstash.yml
              path: logstash.yml
      - name: logstash-pipeline-volume
        configMap:
          name: logstash-configmap
          items:
            - key: logstash.conf
              path: logstash.conf
```

```
kubectl create -f logstash-deploy.yaml -n public-service
```

es & kibana

```
es 和 kibana 用的上面安装的
```

app.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: app
  labels:
    app: app
    env: release
spec:
  selector:
    matchLabels:
      app: app
  replicas: 1
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 0
      maxSurge: 1
  # minReadySeconds: 30
  template:
    metadata:
      labels:
        app: app
    spec:
      nodeSelector:
         kubernetes.io/hostname: k8s-node02
      containers:
        - name: filebeat
          image: elastic/filebeat:7.4.2
          resources:
            requests:
              memory: "100Mi"
              cpu: "10m"
            limits:
              cpu: "200m"
              memory: "300Mi"
          imagePullPolicy: IfNotPresent
          env:
            - name: podIp
              valueFrom:
                fieldRef:
```

```
            apiVersion: v1
            fieldPath: status.podIP
        - name: podName
          valueFrom:
            fieldRef:
              apiVersion: v1
              fieldPath: metadata.name
        - name: podNamespace
          valueFrom:
            fieldRef:
              apiVersion: v1
              fieldPath: metadata.namespace
        - name: podDeployName
          value: app
        - name: TZ
          value: "Asia/Shanghai"
      securityContext:
        runAsUser: 0
      volumeMounts:
        - name: logpath
          mountPath: /data/log/app/
        - name: filebeatconf
          mountPath: /usr/share/filebeat/filebeat.yml
          subPath: usr/share/filebeat/filebeat.yml
    - name: app
      image: alpine:3.6
      imagePullPolicy: IfNotPresent
      volumeMounts:
        - name: logpath
          mountPath: /home/tomcat/target/
        - name: tz-config
          mountPath: /etc/localtime
        - mountPath: /usr/share/zoneinfo/Asia/Shanghai
          name: tz-config
        - mountPath: /etc/timezone
          name: timezone
      env:
        - name: TZ
          value: "Asia/Shanghai"
        - name: LANG
          value: C.UTF-8
        - name: LC_ALL
          value: C.UTF-8
        - name: ENV
          value: k8srelease
        - name: XMS
          value: "2048m"
        - name: XMX
          value: "2048m"
```

```
            - name: MEMORY_LIMIT
              valueFrom:
                resourceFieldRef:
                  resource: requests.memory
                  divisor: 1Mi
          command:
            - sh
            - -c
            - sleep 360000
          ports:
            - containerPort: 8080
              name: tomcat
      volumes:
        - name: tz-config
          hostPath:
            path: /usr/share/zoneinfo/Asia/Shanghai
        - hostPath:
            path: /etc/timezone
            type: ""
          name: timezone
        - name: logpath
          emptyDir: {}
        - name: filebeatconf
          configMap:
            name: filebeatconf
            items:
              - key: filebeat.yml
                path: usr/share/filebeat/filebeat.yml
```

```
# 模拟数据的app

kubectl create -f app.yaml -n public-service

kubectl get po -n public-service
```

使用不同资源名称查询日志

```
kubectl get po -n public-service

# create log for test
kubectl exec -it  app-784784557-p894w -n public-service -c app -- sh
cd /home/tomcat/target/
touch app.log
echo 123 >> app.log

# check
```

```
kubectl exec -it  app-784784557-p894w -n public-service -c filebeat -- sh
cd /data/log/app/
ls
  # 会发现 app.log
  # cat app.log
```

在web view 查看

http://10.4.7.107:31223