本来打算使用ECK（Elastic Cloud Kubernetes）来部署方便扩展的ElasticSearch集群，但是花了2天时间还只是部署了一个简单的集群，主要有以下几个问题没有解决：

- 无法绑定PersistentVolume，每次重启都会丢失数据
- 可以使用NodePort的方式让k8s集群外部访问，但无法指定端口
- 默认开启X-Pack，如果通过修改配置文件的方式关闭X-Pack会导致NodePort无法访问
- 9200端口默认为 HTTPS，无法改为HTTP
- 不能安装自定义插件

所以打算先给集群的每个节点都定义一个StatefulSet来部署集群（共3个节点，都为Master），这样不方便扩展，如果会使用ECK的话不建议这样部署

1. 创建PersistentVolume和PersistentVolumeClaim

本文使用nfs实现数据持久化，使用nfs可以查看http://10.0.10.20:4999/web/#/12?page_id=574

数据

将目录挂载到/data/elasticsearch/data（在elasticsearch.yml中配置path.data可以修改这个目录）实现持久化，这样ElasticSearch重启后就不会丢失数据，本文部署3个节点，所以需要3个PersistentVolume

创建一个`es-data-pv.yml`文件，内容为：

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: es-nfs-pv-0
  labels:
    pv: es-nfs-pv-0
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Recycle
  storageClassName: es-nfs # 要和PersistentVolumeClaim的storageClassName相同
  nfs:
    path: /data/k8s/es/pv-0 # nfs的目录
    server: 10.0.13.240 # nfs的地址

---
apiVersion: v1
kind: PersistentVolume
metadata:
  name: es-nfs-pv-1
  labels:
    pv: es-nfs-pv-1
spec:
  capacity:
```

```
    storage: 5Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Recycle
  storageClassName: es-nfs
  nfs:
    path: /data/k8s/es/pv-1
    server: 10.0.13.240

---
apiVersion: v1
kind: PersistentVolume
metadata:
  name: es-nfs-pv-2
  labels:
    pv: es-nfs-pv-2
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Recycle
  storageClassName: es-nfs
  nfs:
    path: /data/k8s/es/pv-2
    server: 10.0.13.240

---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: es-nfs-pvc-0
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
  storageClassName: es-nfs
  selector:
    matchLabels:
      pv: "es-nfs-pv-0" # 和对应的PersistentVolume的label相同

---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: es-nfs-pvc-1
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
```

```
  storageClassName: es-nfs
  selector:
    matchLabels:
      pv: "es-nfs-pv-1"

---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: es-nfs-pvc-2
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
  storageClassName: es-nfs
  selector:
    matchLabels:
      pv: "es-nfs-pv-2"
```

然后通过文件创建：

```
kubectl create -f es-data-pv.yml
```

插件

通过将要安装的插件所在的目录挂载到/usr/share/elasticsearch/plugins（插件的目录），可以在ElasticSearch启动时自动加载插件，不需要每次启动后手动安装插件或打包一个安装好插件的镜像，插件可以多个节点公用所以只需要一个PersistentVolume

创建一个`es-plugins-pv.yml`文件，内容为：

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: es-nfs-pv-plugins
  labels:
    pv: es-nfs-pv-plugins
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Recycle
  storageClassName: es-nfs
  nfs:
    path: /data/k8s/es/plugins
    server: 10.0.13.240
```

```
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: es-nfs-pvc-plugins
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 5Gi
  storageClassName: es-nfs
  selector:
    matchLabels:
      pv: "es-nfs-pv-plugins"
```

然后通过文件创建：

```
kubectl create -f es-plugins-pv.yml
```

3. 创建ConfigMap

由于集群需要配置所有节点的地址，但是在env中无法设置数组类型的参数，所以需要用ConfigMap来保存配置，然后挂载到/usr/share/elasticsearch/config/elasticsearch.yml（ElasticSearch的配置文件）

创建一个`elasticsearch.yml`文件，内容为：

```
cluster.name: utry-test
node.name: es-0
node.master: true
node.data: true
path.data: /data/elasticsearch/data
http.port: 9200
network.host: 0.0.0.0
discovery.zen.minimum_master_nodes: 2
discovery.zen.ping.unicast.hosts: ["es-inner-svc-0:9300","es-inner-svc-
1:9300","es-inner-svc-2:9300"] # 集群所有Master节点的地址，这里的地址用到了
Service，会在后面创建
```

然后通过文件创建：

```
kubectl create configmap elasticsearch-yml --from-file=elasticsearch.yml
```

4. 创建Service

这里给每个节点的9300端口创建1个Service（这里使用NodePort主要是为了让k8s集群外部的应用通过transport的方式连接ElasticSearch，如果不需要可以改为ClusterIP），9200端口只创建1个Service（REST接口连接只需要调用其中一个节点即可）

创建一个`es-svc.yml`文件，内容为：

```
apiVersion: v1
kind: Service
metadata:
  name: es-svc
  labels:
    app: es-svc
spec:
  type: NodePort
  ports:
  - port: 9200
    name: es-port
    nodePort: 30200
  selector:
    app: es

---
apiVersion: v1
kind: Service
metadata:
  name: es-inner-svc-0
  labels:
    app: es-inner-svc-0
spec:
  type: NodePort
  ports:
  - port: 9300
    name: es-port
    nodePort: 30300
  selector:
    es-node: es-0

---
apiVersion: v1
kind: Service
metadata:
  name: es-inner-svc-1
  labels:
    app: es-inner-svc-1
spec:
  type: NodePort
  ports:
  - port: 9300
    name: es-port
```

```
    nodePort: 30301
  selector:
    es-node: es-1

---
apiVersion: v1
kind: Service
metadata:
  name: es-inner-svc-2
  labels:
    app: es-inner-svc-2
spec:
  type: NodePort
  ports:
  - port: 9300
    name: es-port
    nodePort: 30302
  selector:
    es-node: es-2
```

然后通过文件创建：

```
kubectl create -f es-svc.yml
```

5. 创建StatefulSet

创建一个es-sts.yml文件，内容为：

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: es-sts-0
spec:
  selector:
    matchLabels:
      app: es
      es-node: es-0
  serviceName: "es-svc"
  replicas: 1
  template:
    metadata:
      labels:
        app: es # 所有节点的9200端口都由1个Service转发，所以这个label要相同
        es-node: es-0 # 每个节点的9300端口各自对应1个Service，所以这个label要不同
    spec:
      terminationGracePeriodSeconds: 10
      containers:
      - name: elasticsearch
```

```
        image: elasticsearch:6.8.12
        ports:
        - containerPort: 9200
          name: es-cli
        - containerPort: 9300
          name: es-inner
        resources:
          requests:
            memory: 2Gi
          limits:
            memory: 2Gi
        env:
        - name: node.name
          value: es-0 # 指定节点名称，由于ConfigMap的配置是3个节点共用的，所以节点名称要在这里制定
        volumeMounts:
        - mountPath: "/usr/share/elasticsearch/data"
          name: data-storage # 挂载数据目录
        - mountPath: "/usr/share/elasticsearch/plugins"
          name: plugins-storage # 挂载插件目录
        - name: "es-config-map" # 挂载配置文件
          mountPath: "/usr/share/elasticsearch/config/elasticsearch.yml"
          subPath: elasticsearch.yml
      volumes:
      - name: data-storage
        persistentVolumeClaim:
          claimName: es-nfs-pvc-0
      - name: plugins-storage
        persistentVolumeClaim:
          claimName: es-nfs-pvc-plugins
      - name: "es-config-map"
        configMap:
          name: "es-config-map"
          items:
          - key: "elasticsearch.yml"
            path: "elasticsearch.yml"

---
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: es-sts-1
spec:
  selector:
    matchLabels:
      app: es
      es-node: es-1
  serviceName: "es-svc"
  replicas: 1
  template:
    metadata:
      labels:
        app: es
        es-node: es-1
```

```
    spec:
      terminationGracePeriodSeconds: 10
      containers:
      - name: elasticsearch
        image: elasticsearch:6.8.12
        ports:
        - containerPort: 9200
          name: es-cli
        - containerPort: 9300
          name: es-inner
        resources:
          requests:
            memory: 2Gi
          limits:
            memory: 2Gi
        env:
        - name: node.name
          value: es-1
        volumeMounts:
        - mountPath: "/usr/share/elasticsearch/data"
          name: data-storage
        - mountPath: "/usr/share/elasticsearch/plugins"
          name: plugins-storage
        - name: "es-config-map"
          mountPath: "/usr/share/elasticsearch/config/elasticsearch.yml"
          subPath: elasticsearch.yml
      volumes:
      - name: data-storage
        persistentVolumeClaim:
          claimName: es-nfs-pvc-1
      - name: plugins-storage
        persistentVolumeClaim:
          claimName: es-nfs-pvc-plugins
      - name: "es-config-map"
        configMap:
          name: "es-config-map"
          items:
          - key: "elasticsearch.yml"
            path: "elasticsearch.yml"

---
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: es-sts-2
spec:
  selector:
    matchLabels:
      app: es
      es-node: es-2
  serviceName: "es-svc"
  replicas: 1
  template:
    metadata:
```

```
        labels:
          app: es
          es-node: es-2
      spec:
        terminationGracePeriodSeconds: 10
        containers:
        - name: elasticsearch
          image: elasticsearch:6.8.12
          ports:
          - containerPort: 9200
            name: es-cli
          - containerPort: 9300
            name: es-inner
          resources:
            requests:
              memory: 2Gi
            limits:
              memory: 2Gi
          env:
          - name: node.name
            value: es-2
          volumeMounts:
          - mountPath: "/usr/share/elasticsearch/data"
            name: data-storage
          - mountPath: "/usr/share/elasticsearch/plugins"
            name: plugins-storage
          - name: "es-config-map"
            mountPath: "/usr/share/elasticsearch/config/elasticsearch.yml"
            subPath: elasticsearch.yml
        volumes:
        - name: data-storage
          persistentVolumeClaim:
            claimName: es-nfs-pvc-2
        - name: plugins-storage
          persistentVolumeClaim:
            claimName: es-nfs-pvc-plugins
        - name: "es-config-map"
          configMap:
            name: "es-config-map"
            items:
            - key: "elasticsearch.yml"
              path: "elasticsearch.yml"
```

然后通过文件创建：

```
kubectl create -f es-sts.yml
```