1.安装MetalB

安装手册

2.安装IngressController

参考文档5.2

3.配置kubefate

参考文档6

- 安装kubefate服务

```
cd KubeFATE-master/k8s-deploy
make install
```

- chart包管理

> 当应用安装出问题时，可以自定义修改chart包然后上传

```
#查看chart包
kubefate chart ls

#打包本地chart包
cd KubeFATE-master/helm-charts
make release

#上传本地chart包
kubefate chart upload -f ../helm-charts/fate-v1.6.0-a.tgz
kubefate chart upload -f ../helm-charts/fate-serving-v2.0.4.tgz
kubefate chart upload -f ../helm-charts/fate-exchange-v1.6.0-a.tgz
```

4.配置StorageClass

4.1 NFS

```
#rbac.yaml
[centos@cpu0 ~]$ cat rbac.yaml
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: nfs-provisioner-runner
rules:
  - apiGroups: [""]
    resources: ["persistentvolumes"]
    verbs: ["get", "list", "watch", "create", "delete"]
  - apiGroups: [""]
    resources: ["persistentvolumeclaims"]
    verbs: ["get", "list", "watch", "update"]
  - apiGroups: ["storage.k8s.io"]
```

```
    resources: ["storageclasses"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["events"]
    verbs: ["create", "update", "patch"]
  - apiGroups: [""]
    resources: ["services", "endpoints"]
    verbs: ["get"]
  - apiGroups: ["extensions"]
    resources: ["podsecuritypolicies"]
    resourceNames: ["nfs-provisioner"]
    verbs: ["use"]
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: run-nfs-provisioner
subjects:
  - kind: ServiceAccount
    name: nfs-provisioner
     # replace with namespace where provisioner is deployed
    namespace: default
roleRef:
  kind: ClusterRole
  name: nfs-provisioner-runner
  apiGroup: rbac.authorization.k8s.io
---
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: leader-locking-nfs-provisioner
rules:
  - apiGroups: [""]
    resources: ["endpoints"]
    verbs: ["get", "list", "watch", "create", "update", "patch"]
---
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: leader-locking-nfs-provisioner
subjects:
  - kind: ServiceAccount
    name: nfs-provisioner
    # replace with namespace where provisioner is deployed
    namespace: default
roleRef:
  kind: Role
  name: leader-locking-nfs-provisioner
  apiGroup: rbac.authorization.k8s.io

#deployment.yaml
[centos@cpu0 ~]$ cat deployment.yaml
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: nfs
provisioner: example.com/nfs
mountOptions:
```

```
  - vers=4.1
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: nfs-provisioner
---
kind: Service
apiVersion: v1
metadata:
  name: nfs-provisioner
  labels:
    app: nfs-provisioner
spec:
  ports:
    - name: nfs
      port: 2049
    - name: nfs-udp
      port: 2049
      protocol: UDP
    - name: nlockmgr
      port: 32803
    - name: nlockmgr-udp
      port: 32803
      protocol: UDP
    - name: mountd
      port: 20048
    - name: mountd-udp
      port: 20048
      protocol: UDP
    - name: rquotad
      port: 875
    - name: rquotad-udp
      port: 875
      protocol: UDP
    - name: rpcbind
      port: 111
    - name: rpcbind-udp
      port: 111
      protocol: UDP
    - name: statd
      port: 662
    - name: statd-udp
      port: 662
      protocol: UDP
  selector:
    app: nfs-provisioner
---
kind: Deployment
apiVersion: apps/v1
metadata:
  name: nfs-provisioner
spec:
  selector:
    matchLabels:
      app: nfs-provisioner
  replicas: 1
  strategy:
```

```
    type: Recreate
  template:
    metadata:
      labels:
        app: nfs-provisioner
    spec:
      serviceAccount: nfs-provisioner
      nodeSelector:
        kubernetes.io/hostname: cpu1
      containers:
        - name: nfs-provisioner
          image: k8s.gcr.io/sig-storage/nfs-provisioner:v3.0.0
          ports:
            - name: nfs
              containerPort: 2049
            - name: nfs-udp
              containerPort: 2049
              protocol: UDP
            - name: nlockmgr
              containerPort: 32803
            - name: nlockmgr-udp
              containerPort: 32803
              protocol: UDP
            - name: mountd
              containerPort: 20048
            - name: mountd-udp
              containerPort: 20048
              protocol: UDP
            - name: rquotad
              containerPort: 875
            - name: rquotad-udp
              containerPort: 875
              protocol: UDP
            - name: rpcbind
              containerPort: 111
            - name: rpcbind-udp
              containerPort: 111
              protocol: UDP
            - name: statd
              containerPort: 662
            - name: statd-udp
              containerPort: 662
              protocol: UDP
          securityContext:
            capabilities:
              add:
                - DAC_READ_SEARCH
                - SYS_RESOURCE
          args:
            - "-provisioner=example.com/nfs"
          env:
            - name: POD_IP
              valueFrom:
                fieldRef:
                  fieldPath: status.podIP
            - name: SERVICE_NAME
              value: nfs-provisioner
            - name: POD_NAMESPACE
```

```
              valueFrom:
                fieldRef:
                  fieldPath: metadata.namespace
          imagePullPolicy: "IfNotPresent"
          volumeMounts:
            - name: export-volume
              mountPath: /export
      volumes:
        - name: export-volume
          hostPath:
            path: /srv/k8s
```

4.2 Longhorn

```

```

5.安装serving

- 创建对应的namespace

```
kubectl create namespace fate-10000
kubectl create namespace fate-10000
kubectl create ns fate-serving-9999
kubectl create ns fate-serving-10000
```

- 自定义yaml文件内容

> cluster-serving-10000.yaml

```
name: fate-serving-10000
namespace: fate-serving-10000
chartName: fate-serving
chartVersion: v2.0.4
partyId: 10000
registry: ""
imageTag: ""
pullPolicy:
persistence: true
istio:
  enabled: false
podSecurityPolicy:
  enabled: false
modules:
  - servingProxy
  - servingRedis
  - servingServer
  - servingZookeeper
  - servingAdmin

servingAdmin:
  ingressHost: 10000.serving-admin.kubefate.net
  username: admin
  password: admin
servingProxy:
  type: ClusterIP
```

```
  ingerssHost: 10000.serving-proxy.kubefate.net
  partyList:
   - partyId: 9999
     partyIp: serving-proxy.fate-serving-9999
     partyPort: 8869
servingServer:
  storageClass: "nfs"
  accessMode: ReadWriteOnce
  size: 1Gi
servingRedis:
  password: fate_dev
  storageClass: "nfs"
  accessMode: ReadWriteOnce
  size: 1Gi
servingZookeeper:
  storageClass: "nfs"
  accessMode: ReadWriteOnce
  size: 1Gi
```

cluster-serving-9999.yaml

```
name: fate-serving-9999
namespace: fate-serving-9999
chartName: fate-serving
chartVersion: v2.0.4
partyId: 9999
registry: ""
imageTag: ""
pullPolicy:
persistence: true
istio:
  enabled: false
podSecurityPolicy:
  enabled: false
modules:
  - servingProxy
  - servingRedis
  - servingServer
  - servingZookeeper
  - servingAdmin

servingAdmin:
  ingressHost: 9999.serving-admin.kubefate.net
  username: admin
  password: admin
servingProxy:
  type: ClusterIP
  ingerssHost: 9999.serving-proxy.kubefate.net
  partyList:
    - partyId: 10000
      partyIp: serving-proxy.fate-serving-10000
      partyPort: 8869
servingServer:
  storageClass: "nfs"
  accessMode: ReadWriteOnce
  size: 1Gi
servingRedis:
```

```
  password: fate_dev
  storageClass: "nfs"
  accessMode: ReadWriteOnce
  size: 1Gi
servingZookeeper:
  storageClass: "nfs"
  accessMode: ReadWriteOnce
  size: 1Gi
```

- 安装服务

```
kubefate  cluster install -f ./cluster-serving-10000.yaml
kubefate  cluster install -f ./cluster-serving-9999.yaml
kubefate cluster list
kubefate job list
kubefate describe job ID
```

6.安装fate集群

> cluster-10000.yaml

```
name: fate-10000
namespace: fate-10000
chartName: fate
chartVersion: v1.6.0-a
partyId: 10000
registry: "hub.c.163.com/federatedai"
imageTag: ""
pullPolicy:
imagePullSecrets:
- name: myregistrykey
persistence: true
istio:
  enabled: false
podSecurityPolicy:
  enabled: false
modules:
  - rollsite
  - clustermanager
  - nodemanager
  - mysql
  - python
  - fateboard
backend: eggroll
rollsite:
  exchange:
    ip: rollsite
    port: 9370
    partyList:
    - partyId: 9999
      partyIp: rollsite.fate-9999
      partyPort: 9370
nodemanager:
  count: 2
  sessionProcessorsPerNode: 4
  storageClass: "nfs"
```

```
  accessMode: ReadWriteOnce
  size: 10Gi
python:
  type: ClusterIP
mysql:
  ip: mysql
  port: 3306
  user: fate
  password: fate_dev
  storageClass: "nfs"
  accessMode: ReadWriteOnce
  size: 1Gi
serving:
  useRegistry: true
  zookeeper:
    hosts:
    - serving-zookeeper.fate-serving-10000:2181
    use_acl: false
```

cat cluster-9999.yaml

```
name: fate-9999
namespace: fate-9999
chartName: fate
chartVersion: v1.6.0-a
partyId: 9999
registry: "hub.c.163.com/federatedai"
imageTag: ""
pullPolicy:
persistence: true
istio:
  enabled: false
podSecurityPolicy:
  enabled: false
modules:
  - rollsite
  - clustermanager
  - nodemanager
  - mysql
  - python
  - fateboard
backend: eggroll

rollsite:
  exchange:
    ip: rollsite
    port: 9370
    partyList:
    - partyId: 10000
      partyIp: rollsite.fate-10000
      partyPort: 9370
nodemanager:
  count: 2
  sessionProcessorsPerNode: 4
  storageClass: "nfs"
  accessMode: ReadWriteOnce
  size: 20Gi
```

```
python:
  type: ClusterIP
mysql:
  ip: mysql
  port: 3306
  user: fate
  password: fate_dev
  storageClass: "nfs"
  accessMode: ReadWriteOnce
  size: 1Gi
serving:
  useRegistry: true
  zookeeper:
    hosts:
    - serving-zookeeper.fate-serving-9999:2181
    use_acl: false
```

- 安装fate集群

```
kubefate cluster install -f ./cluster-10000.yaml
kubefate cluster install -f ./cluster-9999.yaml
kubefate cluster install -f cluster-serving-10000.yaml
kubefate cluster install -f cluster-serving-9999.yaml
```

- job管理以及kubefate命令

```
#常用命令
kubefate job list
kubefate job describe
kubefate cluster list

[ppc@cpu7 k8s-deploy]$ kubefate
NAME:
   kubefate - A new cli application

USAGE:
   kubefate [global options] command [command options] [arguments...]

COMMANDS:
   chart      List Charts, create, delete and describe a Chart
   cluster    Manage Cluster install, delete and update
   job        List jobs, describe and delete a job
   namespace  List namespace
   service    Start KubeFATE as service
   user       List all users and describe a user's info
   version    Show kubefate command line and service version
   help, h    Shows a list of commands or help for one command

GLOBAL OPTIONS:
   --help, -h  Show help (default: false)
```

7.进入容器跑任务

```
kubectl exec -n fate-9999 -it svc/fateflow -c python -- bash
kubectl exec -n fate-10000 -it svc/fateflow -c python -- bash
```

8.页面访问

页面访问可以讲各ingress的域名映射写到host表里之后直接访问。