Redis

# 安装

## 通过Docker进行安装

### 单节点

docker run \

-u root \

--name redis \

--restart=always \

-d \

-p 6379:6379 \

redis --requirepass "123456"

### 集群

mkdir -p /usr/local/redis && cd /usr/local/redis

# 创建集群配置文件

vi redis-cluster.tmpl

port ${PORT}

masterauth 123456

requirepass 123456

cluster-enabled yes

cluster-config-file nodes.conf

cluster-node-timeout 5000

cluster-announce-ip 192.168.0.80

cluster-announce-port ${PORT}

cluster-announce-bus-port 1${PORT}

appendonly yes

# 创建各个节点的配置文件

for port in `seq 7001 7006`; do \

mkdir -p ./${port}/conf \

&& PORT=${port} envsubst < ./redis-cluster.tmpl > ./${port}/conf/redis.conf

done

# 创建容器

for port in `seq 7001 7006`; do

docker run -d --net host -v /usr/local/redis/${port}/conf/redis.conf:/etc/redis/redis.conf -v /data/redis/${port}:/data --restart always --name=redis-${port} redis redis-server /etc/redis/redis.conf;

done

# 进入容器完成配置

docker exec -it redis-7001 bash

redis-cli -a 123456 --cluster create 192.168.0.80:7001 192.168.0.80:7002 192.168.0.80:7003 192.168.0.80:7004 192.168.0.80:7005 192.168.0.80:7006 --cluster-replicas 1

## 通过Kubernetes进行安装

cat > /usr/local/kubernetes/redis/redis-cluster.yaml << EOF

apiVersion: v1

kind: ConfigMap

metadata:

  name: redis-cluster

data:

  update-node.sh: |

    #!/bin/sh

    REDIS\_NODES="/data/nodes.conf"

    sed -i -e "/myself/ s/[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}/${POD\_IP}/" ${REDIS\_NODES}

    exec "$@"

  redis.conf: |+

    cluster-enabled yes

    cluster-require-full-coverage no

    cluster-node-timeout 15000

    cluster-config-file /data/nodes.conf

    cluster-migration-barrier 1

    appendonly yes

    protected-mode no

---

apiVersion: apps/v1

kind: StatefulSet

metadata:

  name: redis-cluster

spec:

  serviceName: redis-cluster

  replicas: 6

  selector:

    matchLabels:

      app: redis-cluster

  template:

    metadata:

      labels:

        app: redis-cluster

    spec:

      containers:

      - name: redis

        image: redis:6.0.6

        ports:

        - containerPort: 6379

          name: client

        - containerPort: 16379

          name: gossip

        command: ["/conf/update-node.sh", "redis-server", "/conf/redis.conf"]

        env:

        - name: POD\_IP

          valueFrom:

            fieldRef:

              fieldPath: status.podIP

        volumeMounts:

        - name: conf

          mountPath: /conf

          readOnly: false

        - name: redis-data

          mountPath: /data

          readOnly: false

      volumes:

      - name: conf

        configMap:

          name: redis-cluster

          defaultMode: 0755

volumeClaimTemplates:

- metadata:

name: redis-data

annotations:

volume.beta.kubernetes.io/storage-class: course-nfs-storage

spec:

accessModes: [ "ReadWriteOnce" ]

resources:

requests:

storage: 100Gi

---

apiVersion: v1

kind: Service

metadata:

name: redis

labels:

app: redis

spec:

ports:

- name: redis

port: 6379

targetPort: 6379

clusterIP: None

selector:

app: redis

---

apiVersion: v1

kind: Service

metadata:

name: redis-service #配置外网访问，用于redis客户端管理工具的管理；

labels:

app: redis

spec:

ports:

- name: redis-http

port: 63790

selector:

app: redis

type: NodePort

EOF