

1. 执行命令: `kubectl edit statefulset -n incloud redis-ha-node`, 编辑 redis 的 statefulset。
`dnsPolicy` 值修改为 `ClusterFirstWithHostNet`, 然后在下面增加一行: `hostNetwork: true`。

```
terminationMessagePolicy: File
volumeMounts:
- mountPath: /opt/bitnami/scripts/start-scripts
  name: start-scripts
- mountPath: /health
  name: health
- mountPath: /data
  name: redis-data
- mountPath: /opt/bitnami/redis-sentinel/mounted-etc
  name: config
- mountPath: /opt/bitnami/redis-sentinel/etc
  name: sentinel-tmp-conf
dnsPolicy: ClusterFirstWithHostNet
hostNetwork: true
nodeSelector:
  node-role.kubernetes.io/icmp: enabled
priorityClassName: system-cluster-critical
restartPolicy: Always
schedulerName: default-scheduler
```

更新这个配置之后, redis 的三个 pod 的访问 IP, 不再使用 ClusterIP, 而会变为节点 IP

```
[root@node3 ~]# kubectl get pod -o wide -n incloud | grep redis
redis-ha-node-0          2/2    Running    0          38s    100.2.216.49    node1
redis-ha-node-1          2/2    Running    0          84s    100.2.216.50    node2
redis-ha-node-2          2/2    Running    0          110s   100.2.216.51    node3
```

2. 执行命令: `kubectl edit cm -n incloud icm-configmap-regions`, 编辑 `icm-configmap-regions`, `nodes` 值修改为
`redis://100.2.216.49:26379,redis://100.2.216.50:26379,redis://100.2.216.51:26379` (假设三节点部署, 则这三个 ip 分别对应三个节点的 ip)

```
apiVersion: v1
data:
  application.yml: |
    incloud:
      common:
        db:
          url: 100.2.216.247:33306
        rabbitmq:
          host: rabbitmq
          port: 5672
          username: rabbitmq
          password: '{cipher}0c34999f583d555c3b2e228f0b7f73c0ebafb409608ff53e598da1400b2adcf6729e364481d106aae5adf6ad50e64ce3'
        redis:
          database: 0
          host: redis-ha-headless
          port: 6379
          password: '{cipher}dba92acda9c8af7e8ae363fb302bd53085b09b61fc64190985e6ce05d3105377'
          timeout: '30000'
          sentinel:
            master: mymaster
            nodes: redis://100.2.216.49:26379,redis://100.2.216.50:26379,redis://100.2.216.51:26379
      es:
        url: elasticsearch-es-http.logging:9200
        username: inspur
        password: '{cipher}a7a90b93dae7096397a75e54a524319d55bfa445eb9fb87fbbdc4189f0929bba'
```

3. 执行命令: `kubectl delete po -n incloud `kubectl get po -n incloud | grep config | awk '{print $1}'``, 重启 `incloudmanager-config`。
4. 执行命令: `kubectl delete po -n incloud `kubectl get po -n incloud | grep redis | awk '{print $1}'``, 重启 `redis`。
5. 不需要更改任何配置或代码, 启动本地服务, 可以看到服务已经连接到哨兵, 并可以通

过哨兵获取到主从数据节点。

```
2022-07-08 14:35:35,384 INFO (Version.java:41)- Redisson 3.11.0
2022-07-08 14:35:36,198 INFO (SentinelConnectionManager.java:136)- master: redis://100.2.216.49:6379 added
2022-07-08 14:35:36,217 INFO (SentinelConnectionManager.java:152)- slave: redis://100.2.216.51:6379 added
2022-07-08 14:35:36,218 INFO (SentinelConnectionManager.java:152)- slave: redis://100.2.216.50:6379 added
2022-07-08 14:35:36,232 INFO (SentinelConnectionManager.java:504)- sentinel: redis://100.2.216.50:26379 added
2022-07-08 14:35:36,233 INFO (SentinelConnectionManager.java:504)- sentinel: redis://100.2.216.51:26379 added
2022-07-08 14:35:36,234 INFO (SentinelConnectionManager.java:504)- sentinel: redis://100.2.216.49:26379 added
2022-07-08 14:35:36,234 INFO (SentinelConnectionManager.java:504)- sentinel: redis://100.2.216.49:0 added
2022-07-08 14:35:36,402 INFO (ConnectionPool.java:168)- 1 connections initialized for 100.2.216.51/100.2.216.51:6379
2022-07-08 14:35:36,445 INFO (ConnectionPool.java:168)- 1 connections initialized for 100.2.216.50/100.2.216.50:6379
2022-07-08 14:35:36,446 INFO (ConnectionPool.java:168)- 32 connections initialized for 100.2.216.51/100.2.216.51:6379
2022-07-08 14:35:36,447 INFO (ConnectionPool.java:168)- 32 connections initialized for 100.2.216.50/100.2.216.50:6379
2022-07-08 14:35:36,492 INFO (ConnectionPool.java:168)- 1 connections initialized for 100.2.216.49/100.2.216.49:6379
2022-07-08 14:35:36,492 INFO (ConnectionPool.java:168)- 32 connections initialized for 100.2.216.49/100.2.216.49:6379
2022-07-08 14:35:36,749 ERROR (InspurRedisConfiguration.java:35)- create my redis factory by host redis-ha-headless,port:6379
```

6. 执行命令: `kubectl delete po -n incloud `kubectl get po -n incloud | grep ^i | awk '{print $1}'``, 重启其他服务, 更新各服务 redis 信息。
7. 执行完以上步骤, 本地和远程就都可以连接到哨兵, 并正常使用 redis 集群服务了。

Contributor: 蓝玉杰, 白子玉, 毕亮亮, 杨吉东, 郭立民。