测试环境k8s迁移

1. 测评现有集群迁移的项目

- 节点信息
  - 192.168.13.220
    - role：master
      label：fat-middleware=allow, uat-middleware=allow
  - 192.168.13.221
    - role：node
      label：log-pilot=install
  - 192.168.13.222
    - role：node
  - 192.168.13.223
    - role：node
      label：uat-middleware=allow
- 自动同步名称空间
  - fat-dotnet、fat-frontend、fat-java、fat-middleware、kube-system、kuboard、monitoring、uat-dotnet、uat-frontend、uat-java
- 手动介入名称空间
  - fat-middleware
    - fat-middleware-elk-deployment
      - hostNetwork: true
        nodeSelector: {fat-middleware: allow}
        hostPath: /data/k8s/fat-elk-data
        hostPath: /data/k8s/fat-elk-snapshot
    - fat-middleware-rabbitmq-statefulset
      - nodeSelector: {fat-middleware: allow}
        hostPath: /data/k8s/fat-rabbitmq
    - fat-middleware-redis-deployment
      - nodeSelector: {fat-middleware: allow}
        hostPath: /data/k8s/fat-redis
  - uat-middleware
    - uat-middleware-elk-deployment
      - nodeName: 192.168.13.223
        hostPath: /data/k8s/uat-elk-data
        hostPath: /data/k8s/uat-elk-snapshot
    - fat-middleware-rabbitmq-statefulset
      - nodeSelector: {uat-middleware: allow}
        hostPath: /data/k8s/uat-rabbitmq
    - fat-middleware-redis-deployment
      - nodeSelector: {uat-middleware: allow}
        hostPath: /data/k8s/uat-redis
  - kube-system
    - prometheus-server deployment
      - 运行在192.168.13.223节点之上，需要更改运行节点
    - node-local-dns daemonset
      - 这个在`2. 创建新集群`时开启node-local-dns部署，后续可不用再手动部署
    - configMap: coredns
      - 需要配置coredns，实现内网域名解析
  - default
    - 项目用途：用于测试应用日志收集，优先级最低
      项目：kafka-eagle、kafka、log-pilot、logstash、mysql、zk
      - 所有PV创建在标签为log-pilot=install的节点之上
- 节点部署的systemd服务
  - filebeat，收集系统日志

2. 创建新集群

- 1. 节点信息
  - 192.168.13.170
    - role: master
      label:
      fat-middleware=allow
      uat-middleware=allow
      - kubectl label node 192.168.13.170 fat-middleware=allow uat-middleware=allow
  - 192.168.13.171
    - role: node
      label:
      log-pilot=install
      - kubectl label node 192.168.13.171 log-pilot=install
  - 192.168.13.172
    - role: node
      label:
      uat-middleware=allow
      - kubectl label node 192.168.13.172 uat-middleware=allow
- 2. 创建集群
  - 跟test-k8s ansible部署集群一样，开启node-local-dns功能
- 3. 检查集群是否符合预期
- 4. 复制旧集群的hostPath数据库新集群中
  - # root@test-k8s-master:/data# tar --exclude='k8s/fat-elk-data' --exclude='k8s/fat-elk-snapshot' -czvpf k8s.tar.gz -C /data ./k8s
    # root@test-k8s-master:/data# scp k8s.tar.gz root@192.168.13.170:/root
- 5. 调整新集群的docker limits值（非常重要），否则节点运行的pod将无法运行
  - 无法调度时tasks达到limit值了：
    root@test-k8s02-node02:~# systemctl status docker | grep -i tasks
    Tasks: 4907 (limit: 4915)
  - 配置各节点docker：ansible test-k8s02 -m shell -a 'systemctl set-property docker TasksMax=infinity && systemctl daemon-reload && systemctl restart docker && systemctl status docker | grep -i tasks'
    此时节点没有limits限制 了：
    192.168.13.172 | CHANGED | rc=0 >>
    `-50-TasksMax.conf
    Tasks: 5917

3. 恢复集群项目

- 1. 配置velero bucket和部署velero
  - [root@prometheus velero]# /root/k8s/k8s-test02-env/addon/velero/velero-server-install.sh
    [root@prometheus velero]# velero version
    Client:
    Version: v1.9.7
    Git commit: 9ace4ecbdc08d57415786ab9c896f86dbb6dc0b7
    Server:
    Version: v1.9.7
- 2. 恢复最新的备份数据到新集群
  - [root@prometheus velero]# velero backup get | head -n 3
    [root@prometheus velero]# velero restore create --from-backup test-cluster-daily-20250101170047
    [root@prometheus velero]# velero restore describe test-cluster-daily-20250101170047-20250102195203
- 3. 失败的镜像进行处理
  - [root@prometheus ~]# ansible test-k8s02 -m shell -a 'docker pull harborrepo.hs.com/ops/redis:4.0.11; docker tag harborrepo.hs.com/ops/redis:4.0.11 redis:4.0.11'
    [root@prometheus ~]# ansible test-k8s02 -m shell -a 'docker pull harborrepo.hs.com/k8s/metrics-scraper:v1.0.8; docker tag harborrepo.hs.com/k8s/metrics-scraper:v1.0.8 kubernetesui/metrics-scraper:v1.0.8'
- 4. 配置CoreDNS的Corefile文件，使其它dns解析跟旧集群一样
- 5. 配置prometheus-server deployment
  - 1. 将运行节点配置为nodeName: 192.168.13.172
    2. prometheus挂载的目录权限必须为777
- 5. 监控新集群到prometheus

4. 测试新集群功能

- 1. 检查所有服务并解决问题
  - 1. pod始终启动不起来，查看logs是mysql无法连接，因为devmysql进行了限制，
    mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCES, INDEX, ALTER ON *.* TO 'commonuser'@'192.168.13.170' IDENTIFIED BY 'password';
    mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCES, INDEX, ALTER ON *.* TO 'program_java'@'192.168.13.170' IDENTIFIED BY 'password';
    mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCES, INDEX, ALTER ON `xxl_job`.* TO 'xxljob'@'192.168.13.170' IDENTIFIED BY 'password';
    2. rabbitmq的问题：
    * 有些开发人员写入的是之前集群的IP:PORT，并没有写入域名地址，所以不通，这个不是迁移导致，应该让开发人员修复
    * 因为是全新的rabbitmq，vhost和用户没有创建。（因为之前迁移rabbitmq有重复数据所以没有迁移）
- 2. 执行第`1. 测评现有集群迁移的项目`的`手动介入名称空间`步骤，在执行之前先复制原集群的数据到新集群的对应目录，否则迁移有状态服务数据将丢失
  - 在复制旧redis数据到新的redis时，由于新redis一直是运行的，当数据复制完成后，再重新启动新redis，这时，新redis会将内存数据写入到dump.rdb，结果就是迁移数据失败，解决办法是先停止目标redis，然后迁移数据，最后重新运行redis
    解决方案：
    缩容：kubectl scale deployment fat-middleware-redis-deployment -n fat-middleware --replicas=0
    数据迁移：cp /tmp/dump.rdb /data/k8s/fat-redis
    扩容：kubectl scale deployment fat-middleware-redis-deployment -n fat-middleware --replicas=1
    查看pod状态：kubectl get deploy -n fat-middleware fat-middleware-redis-deployment
- 3. 测试所有服务确保跟旧测试k8s集群服务版本一致
  - 1. 配置新集群kubeconfig到argocd中，使argocd对集群可进行管理
    - 登录argocd: argocd login --grpc-web argocd.k8s.hs.com
      查看集群：argocd cluster list
      添加集群：argocd cluster add context-cluster1 --kubeconfig /root/.kube/config --name test02-cluster
      查看集群：
      [root@prometheus dotnet-appmessage-hs-com]# argocd cluster list
      SERVER NAME VERSION STATUS MESSAGE PROJECT
      https://192.168.13.90:6443 pre-pro-cluster 1.23 Successful
      https://k8s-api-test.hs.com:6443 test-cluster 1.23 Successful
      https://k8s-api-prepro.hs.com:6443 prepro-cluster 1.23 Successful
      https://192.168.13.170:6443 test02-cluster 1.23 Successful
      https://kubernetes.default.svc in-cluster 1.23 Successful
  - 2. 测试argocd构建服务是否正常
    - 挑选一个边缘业务更改Application中的k8sAPI地址，看这个边缘业务是否正常更新为最新版本，测试完毕后将Application中的k8sAPI地址改回去
      1. old cluster:
      root@test-k8s-master:~# kubectl describe deploy java-invoiceapply-service-hs-com-deployment -n fat-java
      Image: harborrepo.hs.com/fat/invoiceapply.service.hs.com:v20250108125650

      2. new cluster:
      root@test-k8s02-master:~# kubectl describe deploy java-invoiceapply-service-hs-com-deployment -n fat-java
      Image: harborrepo.hs.com/fat/invoiceapply.service.hs.com:v20241212115830

      3. change application(java-invoiceapply-service-hs-com) to new cluster(https://192.168.13.170:6443)

      4. new cluster:
      root@test-k8s02-master:~# kubectl describe deploy java-invoiceapply-service-hs-com-deployment -n fat-java
      Image: harborrepo.hs.com/fat/invoiceapply.service.hs.com:v20250108125650

      5. old cluster:
      root@test-k8s-master:~# kubectl describe deploy java-invoiceapply-service-hs-com-deployment -n fat-java
      Image: harborrepo.hs.com/fat/invoiceapply.service.hs.com:v20250108125650

5. 切换测试环境服务指向

- 1. 更改测试nginx配置文件
  - fat-rabbitmq服务、uat-rabbitmq服务
    - 1. 更改nginx.conf配置文件
      grep -E '192.168.13.22[0123]' nginx.conf
      sed -i 's/192.168.13.220/192.168.13.170/g' nginx.conf
      sed -i 's/192.168.13.223:3/192.168.13.172:3/g' nginx.conf
      grep -E '192.168.13.17[012]' nginx.conf

      2. 更改conf.d/*配置文件
      grep -E '192.168.13.22[0123]:4' conf.d/*
      sed -i 's/192.168.13.221:4/192.168.13.171:4/g' conf.d/*
      sed -i 's/192.168.13.223:4/192.168.13.171:4/g' conf.d/*
      grep -E '192.168.13.22[0123]:4' conf.d/*

      3. 重载nginx配置
  - fat-redis服务、uat-redis服务
  - clogui.hs.com、uatlogui.hs.com
  - RabbitMQ.fat.qa.hs.com、RabbitMQ.uat.qa.hs.com
- 2. 更改域名指向
  - clog.hs.com-192.168.13.170
  - uatlog.hs.com-192.168.13.172
  - k8s-api-test.hs.com-192.168.13.170
    - 此步骤需要等较长时间
      1. 等DNS缓存失效
      2. 等Application应用更新为最新，此步骤最为耗时
- 3. 重新配置argocd集群
  - 1. 删除老测试集群
    argocd cluster rm https://k8s-api-test.hs.com:6443

    2. 添加新测试集群，将集群地址https://192.168.13.170:6443 更改为 https://k8s-api-test.hs.com:6443
    argocd cluster add context-cluster1 --kubeconfig ./new-k8s-api-test.hs.com --name test-cluster

    3. 刷新test-homsom项目所有的Application，达到手动同步最新的image版本目的
- 4. kuboard进行配置
  - 1. 更改test-k8s集群kubeconfig文件为新测试集群的kubeconfig
    2. 重新给kuboard用户授权，因为新测试k8s集群资源有rolebinding，但是clusterrole没有还原，所以需要手动创建
    [root@prometheus kuboard]# kubectl apply -f test-clusterrole-homsom-dev.yaml
    clusterrole.rbac.authorization.k8s.io/test-clusterrole-homsom-dev created
    [root@prometheus kuboard]# kubectl apply -f clusterrole-homsom-ops.yaml
    clusterrole.rbac.authorization.k8s.io/clusterrole-homsom-ops created

6. 验证服务是否正常

- 1. 中间件验证
  - 数据要么不迁移，要么迁移最新的数据，否则会有数据重复的风险
- 2. 服务验证
- 3. jenkins发布验证
- 4. kuboard管理验证