[1. 运行环境与文件准备 1](#_Toc718897469)

[1.1 检查集群环境与连接 1](#_Toc766763461)

[1.2 拷贝应用镜像和文档并上传至私有镜像仓库 2](#_Toc2091607027)

[2. 安装Helm (如果环境中已安装helm v3，可略过) 4](#_Toc1479485046)

[3. 部署MinIO 5](#_Toc2139503156)

[3.1 创建storageclass 5](#_Toc1163357524)

[3.2 通过helm安装minio 6](#_Toc1858783580)

[3.3 MinIO配置用户和bucket 10](#_Toc1131016151)

[4. 部署YS1000 11](#_Toc1610690260)

## 1. 运行环境与文件准备

这套方案是基于k8s集群的部署，因此需要一个能够连接k8s集群并有root权限的用户，以及一个私有镜像仓库，把所有需要的文件拷贝到集群所在机器，并将相关image推到私有镜像仓库以方便后续运维。

### 

### 1.1 检查集群环境与连接

检查以下环境准备, 使用 swr.cn-east-3.myhuaweicloud.com 作为私有镜像仓库示例

| Requirment | Example |
| --- | --- |
| k8s | >= v1.15 <= v1.21 |
| docker | >= v19.03.8 |
| swr.cn-east-3.myhuaweicloud.com | 私有镜像地址 |

[root@ys1000-demo2 ~]# kubectl get nodes  
NAME STATUS ROLES AGE VERSION  
ys1000-demo2 Ready control-plane,master 30d v1.20.15  
  
[root@ys1000-demo2 ~]# docker version  
Client: Docker Engine - Community  
 Version: 19.03.14  
 API version: 1.40  
 Go version: go1.13.15  
 ...  
  
[root@ys1000-demo2 ~]# docker login -u xxx -p swr.cn-east-3.myhuaweicloud.com   
...  
Login Succeeded

### 

### 1.2 拷贝应用镜像和文档并上传至私有镜像仓库

第一步，下载软件包并解压至Linux操作环境YS1000-support-main/offline 目录下 **注意**: 解压后的容器镜像文件大小约6GB，请先确保当前运行环境和私有镜像仓库有足够空间

[root@ys1000-demo2 ~]# wget https://ys1000-public.oss-cn-shanghai.aliyuncs.com/v2.7.0/images.tar.gz  
[root@ys1000-demo2 ~]# wget https://ys1000-public.oss-cn-shanghai.aliyuncs.com/v2.7.0/YS1000-support-v2.7.0.zip  
[root@ys1000-demo2 ~]# tar -xvzf images.tar.gz  
[root@ys1000-demo2 ~]# unzip YS1000-support-v2.7.0.zip   
  
[root@ys1000-demo2 ~]# cd YS1000-support-release-2.7/offline/  
# ll  
总用量 13548  
-rw-r--r-- 1 root root 13861119 12月 23 11:30 helm-v3.7.0-linux-amd64.tar.gz  
drwxr-xr-x 2 root root 211 12月 24 10:35 s3-gateway  
-rwxrwxrwx 1 root root 3294 12月 24 11:28 prepare-image.sh  
drwxr-xr-x 2 root root 4096 12月 24 10:36 ys1000  
...  
  
[root@ys1000-demo2 offline]# mv ../../images .  
# ls images/  
total 5.2G  
-rw------- 1 root root 79M Jul 27 09:51 webserver:v2.7.0  
-rw------- 1 root root 185M Jul 27 09:51 velero:v1.7.0-jibu-39a9e6f-202207011049  
-rw------- 1 root root 120M Jul 27 09:50 velero-restic-restore-helper:v1.7.0  
-rw------- 1 root root 106M Jul 27 09:51 velero-plugin-ys1000:v0.4.0  
-rw------- 1 root root 116M Jul 27 09:51 velero-plugin-for-csi:v0.2.0-jibu-2801dcd  
-rw------- 1 root root 65M Jul 27 09:51 velero-plugin-for-aws:v1.3.0  
-rw------- 1 root root 488M Jul 27 09:50 velero-installer:v2.7.0  
-rw------- 1 root root 60M Jul 27 09:51 self-restore:v2.7.0  
-rw------- 1 root root 189M Jul 27 09:51 restic-dm:v2.7.0  
-rw------- 1 root root 175M Jul 27 09:48 qiming-operator:v2.7.0  
-rw------- 1 root root 387M Jul 27 09:52 mysql:8.0.29-debian-10-r23  
-rw------- 1 root root 126M Jul 27 09:52 minio-client:2021.12.10-debian-10-r1  
-rw------- 1 root root 237M Jul 27 09:52 minio:2021.12.10-debian-10-r0  
-rw------- 1 root root 632M Jul 27 09:49 mig-ui:v2.7.0  
-rw------- 1 root root 180M Jul 27 09:49 mig-discovery:v2.7.0  
-rw------- 1 root root 193M Jul 27 09:49 mig-controller:v2.7.0  
-rw------- 1 root root 54M Jul 27 09:52 kube-webhook-certgen:v1.3.0  
-rw------- 1 root root 276M Jul 27 09:52 ingress-nginx-controller:v0.40.2  
-rw------- 1 root root 855M Jul 27 09:50 hook-runner:latest  
-rw------- 1 root root 59M Jul 27 09:51 helm-tool:v2.7.0  
-rw------- 1 root root 174M Jul 27 09:51 dm-agent:v2.7.0  
-rw------- 1 root root 165M Jul 27 09:51 data-mover:v2.7.0  
-rw------- 1 root root 177M Jul 27 09:51 cron:v2.7.0  
-rw------- 1 root root 80M Jul 27 09:52 bitnami-shell:10-debian-10-r275  
-rw------- 1 root root 149M Jul 27 09:51 amberapp:0.0.6

第二步，将私有镜像仓库的地址配置完后，通过运行脚本prepare-image.sh，导入MinIO和YS1000的镜像，并修改tag再上传到私有仓库。

# 设置私有镜像仓库repo地址 (请确保docker login已成功并且具有上传镜像权限)  
[root@ys1000-demo2 offline]# export REPOSITRY\_ID=swr.cn-east-3.myhuaweicloud.com/jibu-dev  
[root@ys1000-demo2 offline]# ./prepare-image.sh -u ys1000  
do image upload only from ./images directory to new repo swr.cn-east-3.myhuaweicloud.com/jibu-dev  
image type: ys1000 only  
docker tag registry.cn-shanghai.aliyuncs.com/jibudata/amberapp:0.0.6 to swr.cn-east-3.myhuaweicloud.com/jibu-dev/amberapp:0.0.6 done!  
The push refers to repository [swr.cn-east-3.myhuaweicloud.com/jibu-dev/amberapp]  
...  
v1.7.0-jibu-2280867-202206231040: digest: sha256:43bb09a4ca27e23a9577c9333c131898d38d4b7727160bb8b110345406716dec size: 1373  
docker push swr.cn-east-3.myhuaweicloud.com/jibu-dev/velero:v1.7.0-jibu-2280867-202206231040 done!

## 

## 2. 安装Helm (如果环境中已安装helm v3，可略过)

第一步，解压helm-v3.7.0-linux-amd64.tar.gz

# tar -zxvf helm-v3.7.0-linux-amd64.tar.gz  
linux-amd64/  
linux-amd64/helm  
linux-amd64/LICENSE  
linux-amd64/README.md

第二步，将二进制文件移至bin目录后查看helm命令

# mv linux-amd64/helm /usr/local/bin/helm  
  
# helm version  
version.BuildInfo{Version:"v3.7.0", GitCommit:"eeac83883cb4014fe60267ec6373570374ce770b", GitTreeState:"clean", GoVersion:"go1.16.8"}

## 

## 3. 部署MinIO

**注意-1:** 推荐在生产环境中使用外部企业级对象存储服务(S3)作为数据备份目标。

**注意-2:** 基于外部S3服务，请准备好备份需要的S3账号和bucket，跳过此步，执行 [4. 部署YS1000](#_4. 部署YS1000)

本文中我们将使用已经打包好的helm chart和docker image通过替换minio-values.yaml的参数实现私有化部署。

### 

### 3.1 创建storageclass

第一步，查看当前环境的storageclass

# kubectl get storageclass  
NAME PROVISIONER RECLAIMPOLICY VOLUMEBINDINGMODE ALLOWVOLUMEEXPANSION AGE  
rook-ceph-block rook-ceph.rbd.csi.ceph.com Delete Immediate true 66d  
test-nfs fuseim.pri/ifs Delete Immediate false 2m54s

第二步，选择目标storageclass (以test-nfs为例)，复制一份test-nfs的yaml文件，修改name和reclaimPolicy并生成一个为备份S3 gateway使用的storageclass

# kubectl get storageclass test-nfs -o yaml > jibu-backup-sc.yaml  
  
# cat jibu-backup-sc.yaml | grep managed-nfs-storage  
 name: managed-nfs-storage  
# cat jibu-backup-sc.yaml | grep Retain  
reclaimPolicy: Retain  
  
# kubectl apply -f ./jibu-backup-sc.yaml  
storageclass.storage.k8s.io/managed-nfs-storage created

第三步，检查storageclass成功创建且参数正确

# kubectl get storageclass  
NAME PROVISIONER RECLAIMPOLICY VOLUMEBINDINGMODE ALLOWVOLUMEEXPANSION AGE  
managed-nfs-storage fuseim.pri/ifs Retain Immediate false 44s  
rook-ceph-block rook-ceph.rbd.csi.ceph.com Delete Immediate true 66d  
test-nfs fuseim.pri/ifs Delete Immediate false 29m

### 

### 3.2 通过helm安装minio

第一步，修改minio-value.yaml 中的值，替换成上一步新建的storageclass和私有镜像仓库的地址，其他参数如resources等根据需求替换（此处为默认值）。

| Key | Value |
| --- | --- |
| global.storageClass | managed-nfs-storage |
| image.registry | swr.cn-east-3.myhuaweicloud.com |
| image.repository | jibu-dev/minio |
| image.tag | 2021.12.10-debian-10-r0 |
| clientImage.registry | swr.cn-east-3.myhuaweicloud.com |
| clientImage.repository | jibu-dev/minio-client |
| clientImage.tag | 2021.12.10-debian-10-r1 |
| volumePermissions.image.registry | swr.cn-east-3.myhuaweicloud.com |
| volumePermissions.image.repository | jibu-dev/bitnami-shell |
| resources.limits.cpu | 100m |
| resources.limits.memory | 64Mi |
| persistence.storageClass | managed-nfs-storage |
| persistence.accessModes | ReadWriteOnce |
| persistence.size | 8Gi |

**注意**: persistence.size 需要根据用户应用数据总量，每日增量以及业务备份频率和过期时间来综合考虑，上述容量仅为示例

[root@ys1000-demo2 offline]# kubectl get storageclasses  
NAME PROVISIONER RECLAIMPOLICY VOLUMEBINDINGMODE ALLOWVOLUMEEXPANSION AGE  
managed-nfs-storage fuseim.pri/ifs Delete Immediate false 127d  
  
[root@ys1000-demo2 offline]# cd s3-gateway  
# 修改 minio-values.yaml 中对应的storageClass 为上述配置的storageClass 和更新私有镜像对应的镜像地址  
[root@ys1000-demo2 s3-gateway]# cat minio-values.yaml  
...  
 storageClass: managed-nfs-storage  
  
image:  
 registry: swr.cn-east-3.myhuaweicloud.com  
 repository: jibu-dev/minio  
 tag: 2021.12.10  
clientImage:  
 registry: swr.cn-east-3.myhuaweicloud.com  
 repository: jibu-dev/minio-client  
 tag: 2021.12.10

第二步，使用helm本地安装minio。

[root@ys1000-demo2 s3-gateway]# helm install minio ./helm-chart-minio-9.2.5.tar.gz --namespace minio --create-namespace -f minio-values.yaml

第三步，按安装完minio后的实际输出命令（每次输出不同，不能直接复制本文！），继续安装minio-client。

[root@ys1000-demo2 s3-gateway]# export ROOT\_USER=$(kubectl get secret --namespace minio minio -o jsonpath="{.data.root-user}" | base64 --decode)

[root@ys1000-demo2 s3-gateway]# export ROOT\_PASSWORD=$(kubectl get secret --namespace minio minio -o jsonpath="{.data.root-password}" | base64 --decode)

[root@ys1000-demo2 s3-gateway]# kubectl run --namespace minio minio-client \

> --rm --tty -i --restart='Never' \

> --env MINIO\_SERVER\_ROOT\_USER=$ROOT\_USER \

> --env MINIO\_SERVER\_ROOT\_PASSWORD=$ROOT\_PASSWORD \

> --env MINIO\_SERVER\_HOST=minio \

> --image swr.cn-east-3.myhuaweicloud.com/jibu-dev/minio-client:2021.12.10-debian-10-r1 -- admin info minio

09:12:27.69

09:12:27.69 Welcome to the Bitnami minio-client container

09:12:27.70 Subscribe to project updates by watching https://github.com/bitnami/bitnami-docker-minio-client

09:12:27.70 Submit issues and feature requests at https://github.com/bitnami/bitnami-docker-minio-client/issues

09:12:27.70

09:12:27.70 INFO ==> \*\* Starting MinIO Client setup \*\*

minio-client 09:12:27.70 INFO ==> Adding Minio host to 'mc' configuration...

Added `minio` successfully.

09:12:27.74 INFO ==> \*\* MinIO Client setup finished! \*\*

● minio:9000

Uptime: 16 seconds

Version: 2021-12-10T23:03:39Z

pod "minio-client" deleted

第四步，上述命令成功完成后，修改minio的service访问方式为nodeport，提供对外S3服务。 **注意**: 本文档以NodePort为例, 其他配置例如 ingress 可根据平台对应信息进行设置

# kubectl get pod -n minio  
NAME READY STATUS RESTARTS AGE  
minio-778c47547c-cns9v 1/1 Running 0 29s

# 修改S3 服务端口9000对应的nodeport访问端口为31900  
# 修改S3 console服务端口9001对应的nodeport访问端口为31901  
# 修改service type为NodePort  
# kubectl -n minio edit svc minio  
...  
spec:  
 clusterIP: 10.98.207.210  
 clusterIPs:  
 - 10.98.207.210  
 externalTrafficPolicy: Cluster  
 ports:  
 - name: minio-api  
 nodePort: 31900 # <<<<<  
 port: 9000  
 protocol: TCP  
 targetPort: minio-api  
 - name: minio-console  
 nodePort: 31901 # <<<<<  
 port: 9001  
 protocol: TCP  
 targetPort: minio-console  
 selector:  
 app.kubernetes.io/instance: minio-1639291135  
 app.kubernetes.io/name: minio  
 sessionAffinity: None  
 type: NodePort # <<<<<  
  
# kubectl -n minio get svc minio  
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE  
minio NodePort 10.98.207.210 <none> 9000:31900/TCP,9001:31901/TCP 42h

### 

### 3.3 MinIO配置用户和bucket

第一步，使用上述命令环境变量中的的用户名和密码 echo $ROOT\_USER; echo $ROOT\_PASSWORD，前端登陆minio。

[root@ys1000-demo2 ~]# echo $ROOT\_USER  
bbIfNCYsAy  
[root@ys1000-demo2 ~]# echo $ROOT\_PASSWORD  
A1SwRaOBM23NwGbK10IByfpS8SrF4XEgjQUyfD65

第二步，浏览器打开

http:// < cluster node ip > :31901/login

输入上述用户名：bbIfNCYsAy，密码：A1SwRaOBM23NwGbK10IByfpS8SrF4XEgjQUyfD65 。

第三步，点击左侧导航栏Users，创建一个user，记录access key和secret key, 并选择权限大于等于readwrite。 **注意**: 用户需要使用额外安全环境保存S3密钥，防止因系统重装等原因造成的密钥丢失, 例如如下用户名和密码示例:

access key：minio secret key：minio123

第四步，点击左侧导航栏Buckets，创建一个bucket， 此处以bucket test 为例。

## 

## 部署YS1000

### 4.1 离线部署YS1000

第一步，进入./ys1000文件夹，修改qiming-value.yaml中的容器镜像地址替换成私有镜像仓库的repositry。

[root@ys1000-demo2 s3-gateway]# cd ../ys1000/  
  
# 修改qiming-values.yaml中所有的镜像配置为私有镜像地址  
# cat qiming-values.yaml  
...  
image:  
 repository: swr.cn-east-3.myhuaweicloud.com/jibu-dev/qiming-operator  
 pullPolicy: Always  
 tag: "v2.7.0"  
...  
imageBase:  
 registry: swr.cn-east-3.myhuaweicloud.com/jibu-dev  
 pullPolicy: Always  
 tag: "v2.7.0"  
...  
componentImages:  
 uiImage:  
 repository: swr.cn-east-3.myhuaweicloud.com/jibu-dev/mig-ui  
 tag: "v2.7.0"  
 discoveryImage:  
 repository: swr.cn-east-3.myhuaweicloud.com/jibu-dev/mig-discovery  
 tag: "v2.7.0"  
 migControllerImage:  
 repository: swr.cn-east-3.myhuaweicloud.com/jibu-dev/mig-controller  
 tag: "v2.7.0"  
 resticHelperImage:  
 repository: swr.cn-east-3.myhuaweicloud.com/jibu-dev/velero-restic-restore-helper  
 tag: "v1.7.0"  
 veleroInstallerImage:  
 repository: swr.cn-east-3.myhuaweicloud.com/jibu-dev/velero-installer  
 tag: "v2.7.0"  
 hookRunnerImage:  
 repository: swr.cn-east-3.myhuaweicloud.com/jibu-dev/hook-runner  
 tag: "latest"  
 cron:  
 repository: swr.cn-east-3.myhuaweicloud.com/jibu-dev/cron  
 tag: "v2.7.0"  
 helmTool:  
 repository: swr.cn-east-3.myhuaweicloud.com/jibu-dev/helm-tool  
 tag: "v2.7.0"  
 selfRestore:  
 repository: swr.cn-east-3.myhuaweicloud.com/jibu-dev/self-restore  
 tag: "v2.7.0"  
 webServer:  
 repository: registry.cn-shanghai.aliyuncs.com/jibudata/webserver  
 tag: "v2.7.0"  
...  
migconfig:  
 ...  
 amberappRegistry: "swr.cn-east-3.myhuaweicloud.com"  
 amberappRepo: "jibu-dev/amberapp"  
 amberappTag: "0.0.6"  
 amberappEnabled: true  
 amberappClusters: "all"  
  
 datamoverRegistry: "swr.cn-east-3.myhuaweicloud.com"  
 datamoverRepo: "jibu-dev/data-mover"  
 datamoverTag: "v2.7.0"  
 datamoverEnabled: true  
 datamoverClusters: "all"  
...  
velero:  
 enabled: true  
 namespace: qiming-backend  
 image: swr.cn-east-3.myhuaweicloud.com/jibu-dev/velero:v1.7.0-jibu-39a9e6f-202207011049

plugins: swr.cn-east-3.myhuaweicloud.com/jibu-dev/velero-plugin-for-aws:v1.3.0,swr.cn-east-3.myhuaweicloud.com/jibu-dev/velero-plugin-for-csi:v0.2.0-jibu-2801dcd,swr.cn-east-3.myhuaweicloud.com/jibu-dev/velero-plugin-ys1000:v0.4.0  
...

第二步，使用helm本地安装YS1000。 **注意**: 本文档以NodePort为例, 其他配置例如 ingress 可根据平台对应信息进行设置

# S3 参数示例  
# http://139.198.27.211:31900 为上面minio配置的S3 服务端口对应的IP地址和对外node port端口   
# minio 和 minio123 分别为上述minio 配置的accessKey 和secretKey

# test是上述minio配置的S3 bucket名称  
#  
[root@ys1000-demo2 ys1000]# helm install ./qiming-operator-2.7.0.tgz --namespace qiming-migration --create-namespace --generate-name -f qiming-values.yaml --set service.type=NodePort --set s3Config.accessKey=minio --set s3Config.secretKey=minio123 --set s3Config.bucket=test --set s3Config.s3Url=http://139.198.27.211:31900

第三步，查看qiming-operator版本和pod运行情况，等待pod就绪

# kubectl -n qiming-migration get pods  
NAME READY STATUS RESTARTS AGE  
cron-79cf8cb8f7-p7dzz 1/1 Running 0 62m  
mig-controller-default-56f88ff77c-f4w5w 1/1 Running 0 62m  
mysql-0 1/1 Running 0 63m  
qiming-operator-1658889927-59569d987b-l76hw 1/1 Running 0 63m  
ui-discovery-default-5c679db89f-vkv8z 2/2 Running 0 62m  
webserver-6f56575f65-97zqt 1/1 Running 0 62m  
  
# kubectl -n qiming-backend get pods  
NAME READY STATUS RESTARTS AGE  
amberapp-controller-manager-76d8fb4998-5p6qz 1/1 Running 0 37d  
data-mover-controller-manager-6f878565bb-px9nt 1/1 Running 0 54m  
restic-djwcd 1/1 Running 0 3d1h  
velero-5586df6449-xgqh4 1/1 Running 0 3d1h  
velero-installer-cdcfc8fd5-l7gkf 1/1 Running 0 58m  
  
# helm list -A  
NAME NAMESPACE REVISION UPDATED STATUS CHART APP VERSION   
...  
qiming-operator-1658889927 qiming-migration 1 2022-07-27 10:45:29.422516501 +0800 CST deployed qiming-operator-2.7.0 2.7.0

第四步，安装成功后根据提示获取访问url和token，并登陆YS1000前端

...  
1. Check the application status Ready by running these commands:  
 NOTE: It may take a few minutes to pull docker images.  
 You can watch the status of by running `kubectl --namespace qiming-migration get migconfigs.migration.yinhestor.com -w`  
 kubectl --namespace qiming-migration get migconfigs.migration.yinhestor.com   
  
2. After status is ready, get the application URL by running these commands:  
 export NODE\_PORT=$(kubectl get --namespace qiming-migration -o jsonpath="{.spec.ports[0].nodePort}" services ui-service-default )  
 export NODE\_IP=$(kubectl get nodes --namespace qiming-migration -o jsonpath="{.items[0].status.addresses[0].address}")  
 echo http://$NODE\_IP:$NODE\_PORT  
  
3. Login web UI with the token by running these commands:  
 export SECRET=$(kubectl -n qiming-migration get secret | (grep qiming-operator |grep -v helm || echo "$\_") | awk '{print $1}')  
 export TOKEN=$(kubectl -n qiming-migration describe secrets $SECRET |grep token: | awk '{print $2}')  
 echo $TOKEN

### 4.2 离线升级YS1000

第一步， YS1000的离线升级步骤和部署步骤类似， 可参考[1.2 拷贝应用镜像和文档并上传至私有镜像仓库](#_1.2 拷贝应用镜像和文档并上传至私有镜像仓库) 下载升级的软件包并解压至Linux操作环境YS1000-support-main/offline 目录下， 通过运行脚本prepare-image.sh，导入YS1000升级后的镜像，并修改tag再上传到私有仓库。

[root@ys1000-demo2 ~]# wget https://ys1000-public.oss-cn-shanghai.aliyuncs.com/v2.7.2/images.tar.gz  
[root@ys1000-demo2 ~]# wget https://ys1000-public.oss-cn-shanghai.aliyuncs.com/v2.7.2/YS1000-support-v2.7.2.zip  
[root@ys1000-demo2 ~]# tar -xvzf images.tar.gz  
[root@ys1000-demo2 ~]# unzip YS1000-support-v2.7.2.zip

...

第二步， 使用命令 helm upgrade 进行软件升级，通过参数 --version=<CHART VERSION> 指定升级版本。

**注意-1：**如果需要在升级过程中修改或者增加部分参数，可以附加参数 --set key=value[,key=value] 来完成，具体参数参照文末 配置   
**注意-2:** 如果安装环境中，之前安装过ys1000 历史版本，需要手动更新crd之后再进行安装或者升级

helm upgrade./qiming-operator-2.7.0.tgz --namespace qiming-migration --create-namespace --generate-name -f qiming-values.yaml --set service.type=NodePort --set s3Config.accessKey=minio --set s3Config.secretKey=minio123 --set s3Config.bucket=test --set s3Config.s3Url=http://139.198.27.211:31900