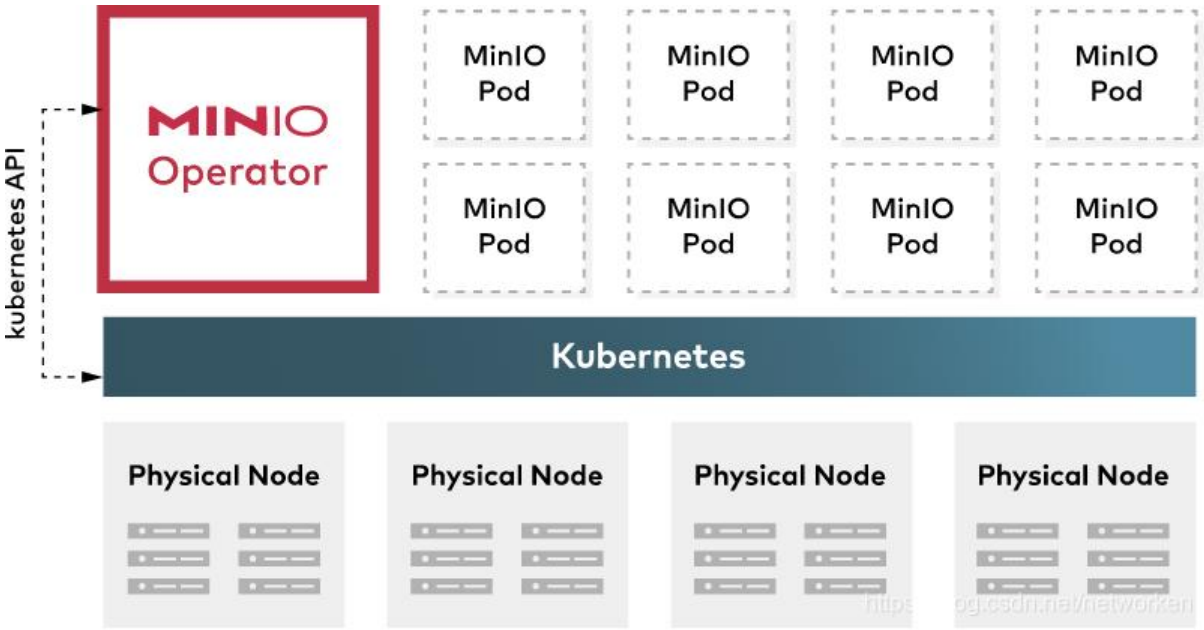
kubernetes 部署 minio 对象存储

作者：wangjunjack
原文链接：https://ld246.com/article/1614926078888
来源网站：链滴

Kubernetes部署MinIO

Kubernetes的部署和状态集提供了在独立，分布式模式下部署MinIO服务器的完美平台。 在Kubernetes上部署MinIO有多种选择，您可以选择最适合您的。



参考：https://github.com/minio/charts

部署前提条件:

1. 需要准备k8s集群，并且包含足够多的节点，如4个节点的集群

```
[root@jenkins ~]# kubectl get nodes
NAME        STATUS   ROLES    AGE   VERSION
k8s-master1   Ready    master   89d   v1.18.8
k8s-master2   Ready    master   89d   v1.18.8
k8s-master3   Ready    master   89d   v1.18.8
k8s-node1     Ready    <none>   89d   v1.18.8
```

2. 需要准备可用的动态存储(longhorn、rook、openebs等)

```
[root@jenkins ~]# kubectl get sc
NAME                 PROVISIONER          RECLAIMPOLICY   VOLUMEBINDINGMODE   ALLOWVO
UMEEXPANSION   AGE
longhorn (default)   driver.longhorn.io   Delete          Immediate           true
                89d
```

独立模式部署

基于官方helm chat进行部署，安装 MinIO chart

```
helm repo add minio https://helm.min.io/
```

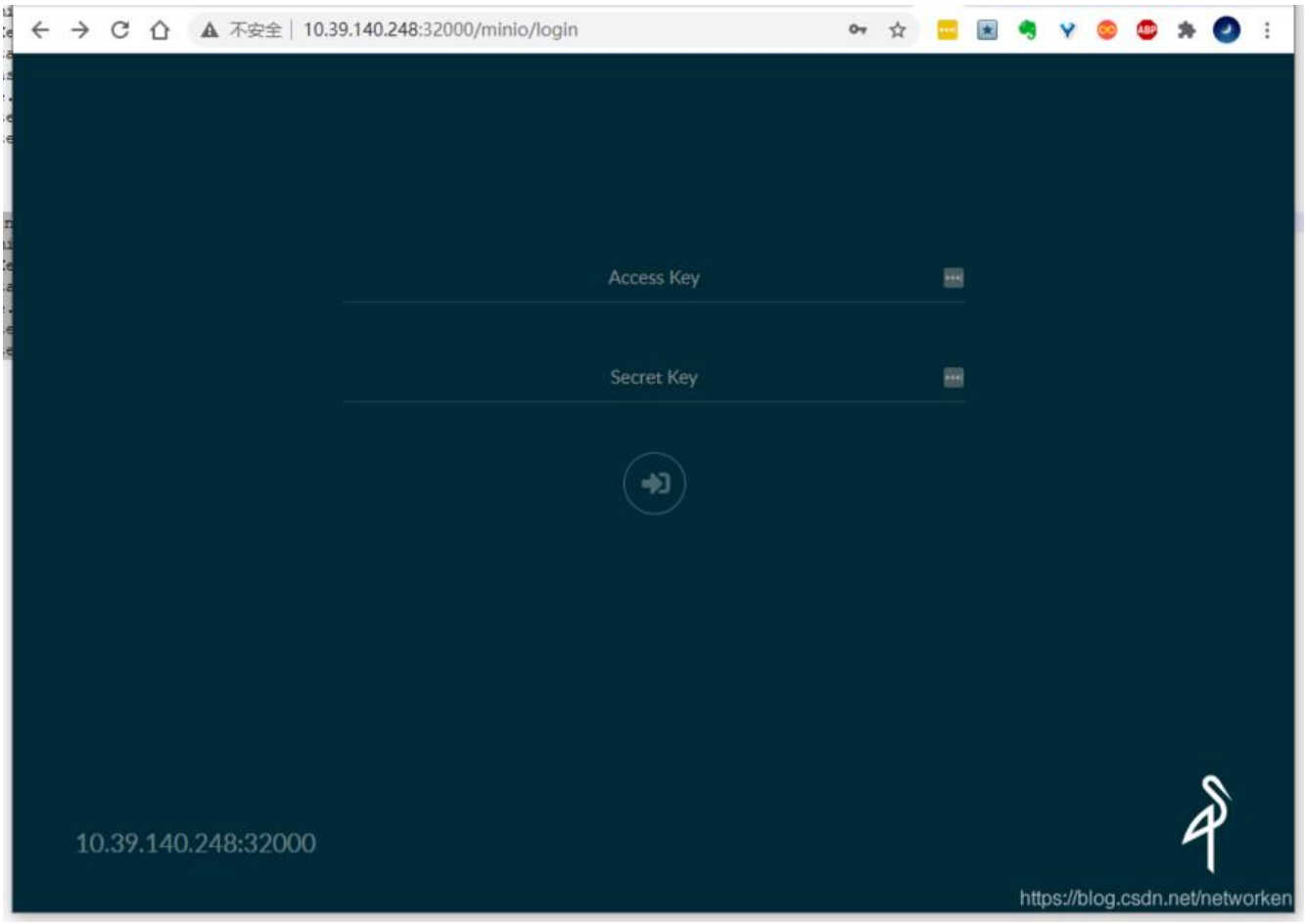
独立模式部署minio，使用deployment方式部署单个pod：

```
helm install minio \
  --namespace minio --create-namespace \
  --set accessKey=minio,secretKey=minio123 \
  --set mode=standalone \
  --set service.type=NodePort \
  --set persistence.enabled=true \
  --set persistence.size=10Gi \
  --set persistence.storageClass=longhorn \
  minio/minio
```

查看创建的资源

```
[root@jenkins ~]# kubectl -n minio get deploy
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
minio   1/1     1            1           12m

[root@jenkins ~]# kubectl -n minio get pods
NAME                     READY   STATUS    RESTARTS   AGE
minio-76dcf6b46c-6lm8c   1/1     Running   0          12m

[root@jenkins minio]# kubectl -n minio get pvc
NAME    STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECL
SS   AGE
minio   Bound    pvc-2f54ab9b-ed5e-4a4d-b528-87be85bf8e6a   10Gi       RWO            longho
n       6m44s

[root@jenkins ~]# kubectl -n minio get svc
NAME    TYPE       CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
minio   NodePort   10.103.145.126   <none>        9000:32000/TCP   12m
```

浏览器访问minio UI:



分布式模式部署

分布式模式部署minio，使用statefulset模式部署多个pod并分布在不同节点：

```
helm install minio \
  --namespace minio --create-namespace \
  --set accessKey=minio,secretKey=minio123 \
  --set mode=distributed \
  --set replicas=4 \
  --set service.type=NodePort \
  --set persistence.size=10Gi \
  --set persistence.storageClass=longhorn \
  minio/minio
```

说明：独立模式下replicas参数不生效，仅对分布式模式生效， 可选值 4 <= x <= 16

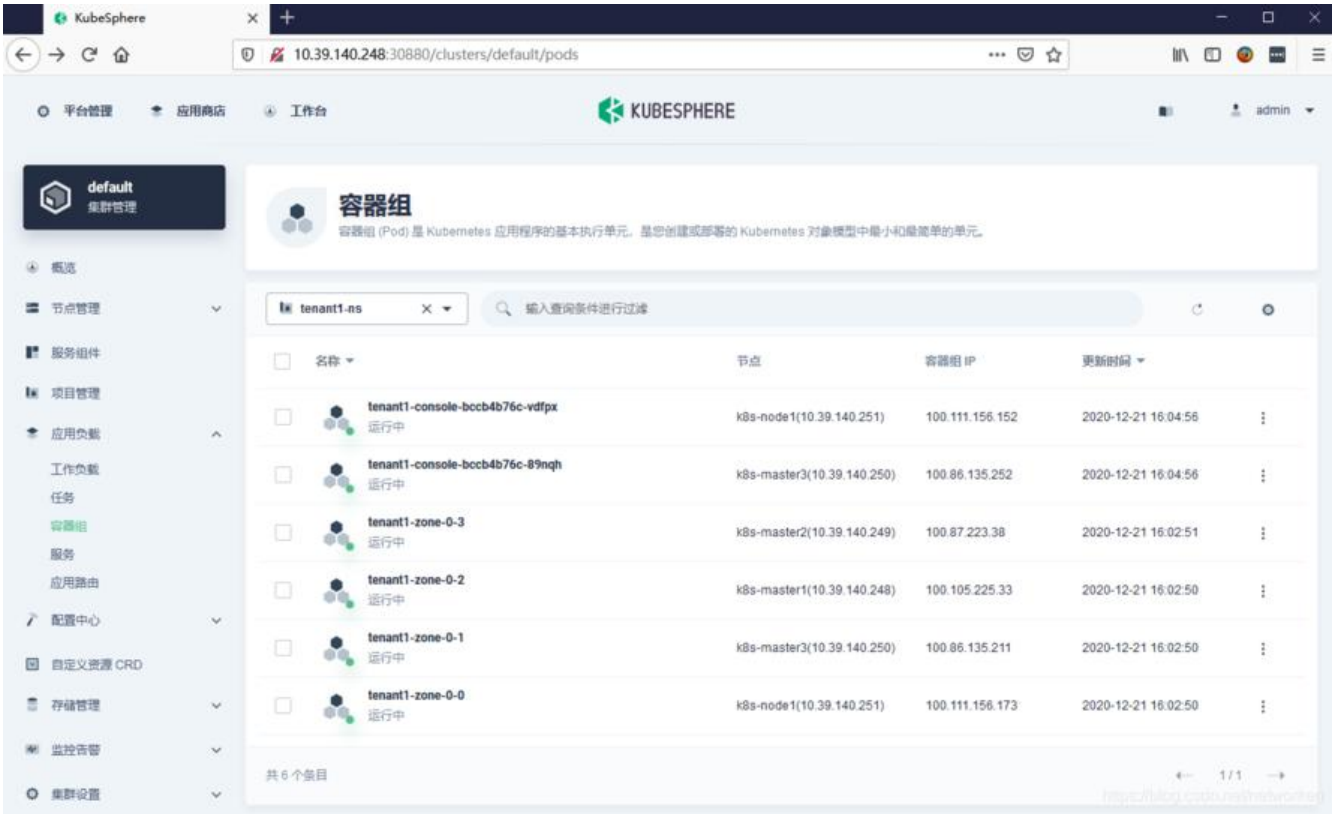
查看创建的资源:

```
[root@jenkins ~]# kubectl -n minio get sts
NAME    READY   AGE
minio   4/4     25m

[root@jenkins ~]# kubectl -n minio get pods -o wide
NAME      READY   STATUS    RESTARTS   AGE   IP              NODE         NOMINATED NODE
EADINESS GATES
```

```
minio-0   1/1   Running   0   26m   100.86.135.210   k8s-master3   <none>   <none
minio-1   1/1   Running   0   26m   100.105.225.44   k8s-master1   <none>   <none
minio-2   1/1   Running   0   26m   100.111.156.179  k8s-node1     <none>   <non
>
minio-3   1/1   Running   0   26m   100.87.223.56    k8s-master2   <none>   <none
```

```
[root@jenkins ~]# kubectl -n minio get pvc
NAME            STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORA
ECLASS   AGE
export-minio-0   Bound    pvc-b48cacbb-d5ab-41d9-8498-6bfb9a385baf   10Gi       RWO
 longhorn       25m
export-minio-1   Bound    pvc-67e9bb47-1f89-4cd8-ac95-ac974941fc0c   10Gi       RWO
longhorn       25m
export-minio-2   Bound    pvc-eb895874-9b96-49c4-8259-b6aefb171bef   10Gi       RWO
 longhorn       25m
export-minio-3   Bound    pvc-959eaf35-d7c7-46a5-8a1f-487203c190bf   10Gi       RWO
longhorn       25m

[root@jenkins ~]# kubectl -n minio get svc
NAME        TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
minio       NodePort    10.106.114.111   <none>        9000:32000/TCP   25m
minio-svc   ClusterIP   None             <none>        9000/TCP         25m
```

在kubesphere容器平台中查看部署的minio资源：



清理minio集群

```
helm -n minio uninstall minio
```

分布式模式下需要手动清理pvc和pv

```
kubectl -n minio delete pvc --all
```

Operator方式部署

MinIO Operator为Kubernetes带来了对MinIO，图形控制台和加密的原生支持。

参考：https://github.com/minio/operator

kubectl安装 krew插件

```
(
  set -x; cd "$(mktemp -d)" &&
  curl -fsSLO "https://github.com/kubernetes-sigs/krew/releases/latest/download/krew.tar.gz"
&&
  tar zxvf krew.tar.gz &&
  KREW=./krew-"$(uname | tr '[:upper:]' '[:lower:]')_$(uname -m | sed -e 's/x86_64/amd64/' -e '
/arm.*$/arm/')" &&
  "$KREW" install krew
)
```

添加环境变量

```
echo 'export PATH="${KREW_ROOT:-$HOME/.krew}/bin:$PATH"' >> /root/.bashrc
```

安装minio插件

```
kubectl krew install minio
```

MinIO Operator提供MinIO租户创建，管理，升级，池添加等功能。Operator可以控制和管理多个MinIO租户。

首先，初始化MinIO Operator部署。这是一个一次性的过程。

kubectl minio初始化，一旦创建了MinIO Operator，便可以继续进行Tenant创建。

```
[root@jenkins ~]# kubectl minio init
```

查看创建的operator

```
[root@jenkins ~]# kubectl get pods |grep minio
minio-operator-78b4f47796-rpnbt   1/1     Running   0          2m35s
```

生成tenant yaml文件，需要提前准备可用的storage-class，指定4个server，每个server一个volume，共40Gi大小。

```
kubectl minio tenant create --name tenant1 \
   --namespace tenant1-ns \
   --storage-class longhorn \
   --servers 4 --volumes 4 --capacity 40Gi -o > tenant.yaml
```

由于默认console镜像版本较低，存在bug，修改tenant.yaml文件，将console镜像修改为新版本:

```
[root@jenkins minio]# cat tenant.yaml  |grep image: | grep console
    image: minio/console:v0.4.6
```

部署yaml文件

```
kubectl apply -f tenant.yaml
```

查看创建的资源

```
[root@jenkins minio]# kubectl -n tenant1-ns get pods
NAME                          READY   STATUS    RESTARTS   AGE
tenant1-console-bccb4b76c-89nqh   1/1     Running   0          23m
tenant1-console-bccb4b76c-vdfpx   1/1     Running   0          23m
tenant1-zone-0-0                  1/1     Running   0          25m
tenant1-zone-0-1                  1/1     Running   0          25m
tenant1-zone-0-2                  1/1     Running   0          25m
tenant1-zone-0-3                  1/1     Running   0          25m
```

修改service类型为NodePort，方便访问minio UI以及tenant-console UI:

```
kubectl -n tenant1-ns patch svc minio -p '{"spec": {"type": "NodePort"}}'
kubectl -n tenant1-ns patch svc tenant1-console -p '{"spec": {"type": "NodePort"}}'
```

查看service，记录minio及tenant1-console中的nodeport

```
[root@jenkins minio]# kubectl -n tenant1-ns get svc
NAME             TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                       AGE
minio            NodePort    10.98.66.158     <none>        443:30143/TCP                 26m
tenant1-console   NodePort    10.101.229.165   <none>        9090:30797/TCP,9443:32372/TCP
 23m
tenant1-hl       ClusterIP   None             <none>        9000/TCP                      26m
```

查看minio UI登录信息

```
# cat tenant.yaml
......
---
apiVersion: v1
data:
  accesskey: ZWM1ZTNmMmYtNzc0NS00MWMwLTlkN2MtODFlMWFkYTczYWU1
  secretkey: NWY1MzYwYmQtNjBlZS00MjY5LTkyYTItYzk4NTNlZDc3Yjk5
kind: Secret
```

```
metadata:
  creationTimestamp: null
  name: tenant1-creds-secret
  namespace: tenant1-ns

---
apiVersion: v1
data:
  CONSOLE_ACCESS_KEY: NjEzNjcyMjAtMDg3ZS00Zjk0LWJkY2QtOWRhODM4YmJmOTE5
  CONSOLE_HMAC_JWT_SECRET: NmEwMWE2N2EtYzM1MS00ZmFkLTkxMDEtMTc1MTBkZmI
ZmU4
  CONSOLE_PBKDF_PASSPHRASE: ZDExYjY2MjAtMWZlNS00Y2ZiLWFlNjUtYWUxZjBmZWRhN2
1
  CONSOLE_PBKDF_SALT: ZmRiZmU5ZmItMWExZi00MmU5LThkNWEtYzJiYWExNmJlN2Jk
  CONSOLE_SECRET_KEY: YWVkZTVjMGYtZDMyMS00MDkxLWFkNDYtMTg3YmNlYWQ3Yzlj
kind: Secret
metadata:
  creationTimestamp: null
  name: tenant1-console-secret
  namespace: tenant1-ns
......
```

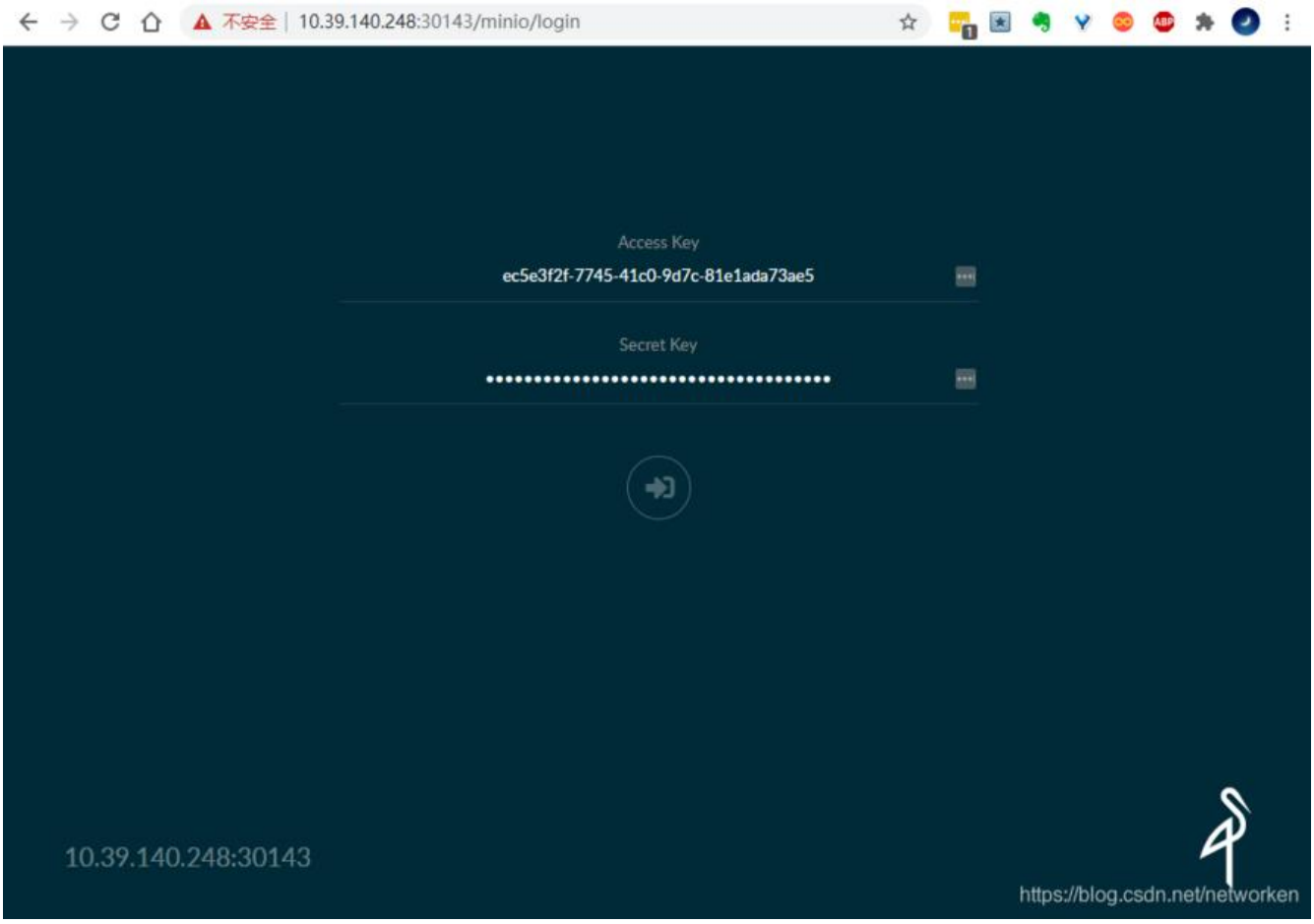
将tenant1-creds-secret中accesskey及secretkey解码

```
echo ZWM1ZTNmMmYtNzc0NS00MWMwLTlkN2MtODFl-e MWFkYTczYWU1 | base64 -d
echo NWY1MzYwYmQtNjBlZS00MjY5LTkyYTItYzk4NTNlZDc3Yjk5 | base64 -d
```

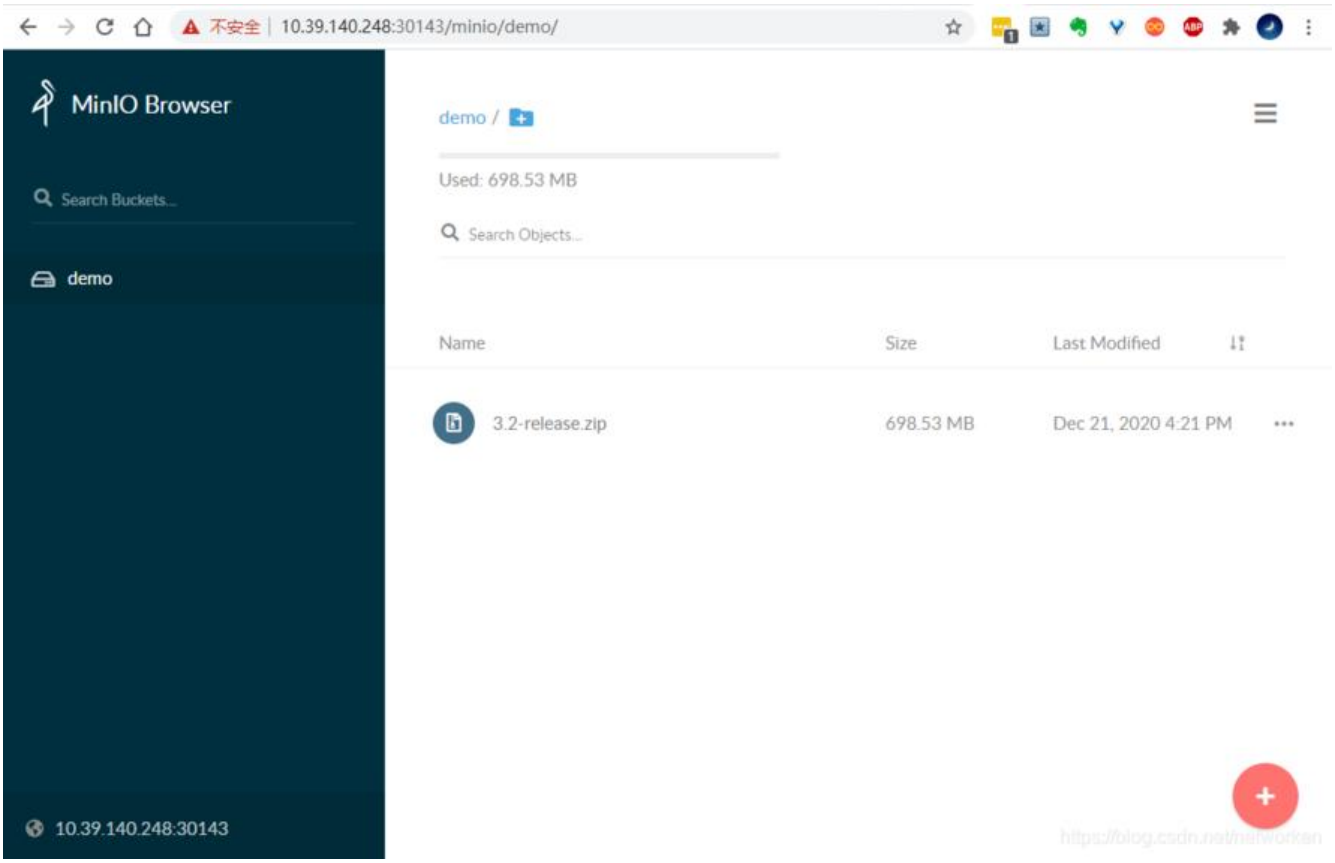
将tenant1-console-secret中CONSOLE_ACCESS_KEY及CONSOLE_SECRET_KEY解码

```
echo NjEzNjcyMjAtMDg3ZS00Zjk0LWJkY2QtOWRhODM4YmJmOTE5 | base64 -d
echo YWVkZTVjMGYtZDMyMS00MDkxLWFkNDYtMTg3YmNlYWQ3Yzlj | base64 -d
```

使用nodeport访问minio UI，使用https方式：

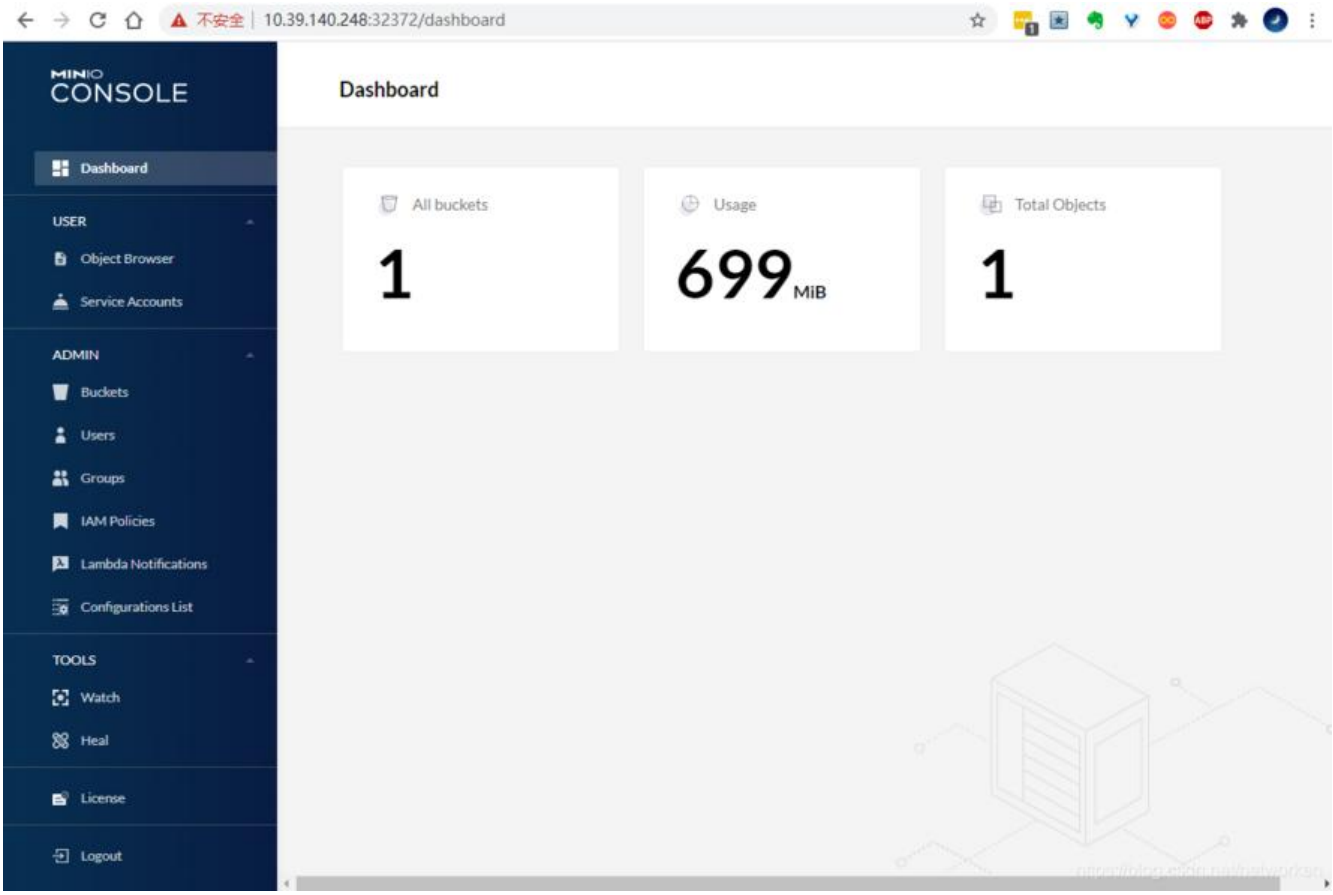


验证创建bucket



原文链接：kubernetes 部署 minio 对象存储

使用nodeport访问tenant-console UI，使用https方式：



minio扩容

您可以使用kubectl minio插件向租户添加容量，如下所示

kubectl minio tenant expand --name tenant1 --servers 8 --volumes 32 --capacity 32Ti

这将为租户增加32个驱动器，这些驱动器均匀分布在8台服务器上tenant1，并具有32Ti的额外容量。

清理operator

[root@jenkins jenkins]# kubectl minio tenant delete --name tenant1 -n tenant1-ns