

设置 storageClass

由于部署是使用的 microk8s 并且这个小集群只有一台服务器，所以存储就保存到本地

storageClass.yaml

```

1 apiVersion: storage.k8s.io/v1
2 kind: StorageClass
3 metadata:
4   annotations:
5     storageclass.kubernetes.io/is-default-class: "false"
6   name: microk8s-localhost
7 provisioner: microk8s.io/hostpath
8 reclaimPolicy: Retain
9 volumeBindingMode: Immediate

```

```

1 $ kubectl apply -f storageClass.yaml
2
3
4 $ kubectl get sc
5 NAME                                PROVISIONER                                RECLAIMPOLICY
6 microk8s-hostpath (default)         microk8s.io/hostpath                       Delete
7 microk8s-localhost                  microk8s.io/hostpath                       Retain

```

NAME	VOLUMEBINDINGMODE	ALLOWVOLUMEEXPANSION	PROVISIONER	AGE	RECLAIMPOLICY
microk8s-hostpath (default)	Immediate	false	microk8s.io/hostpath	4d4h	Delete
microk8s-localhost	Immediate	false	microk8s.io/hostpath	3d1h	Retain

部署数据库

设置 pvc

pg-pvc.yaml

```

1 kind: PersistentVolume
2 apiVersion: v1
3 metadata:
4   name: postgres-data-pv
5   labels:

```

```

6   type: local
7   app: postgres-data
8   spec:
9     storageClassName: microk8s-localhost
10    capacity:
11      storage: 15Gi
12    accessModes:
13      - ReadWriteMany
14    hostPath:
15      path: "/var/local/postgresql/data"
16  ---
17  kind: PersistentVolumeClaim
18  apiVersion: v1
19  metadata:
20    name: postgres-data-pvc
21    namespace: devops
22    labels:
23      app: postgres-data
24  spec:
25    storageClassName: microk8s-localhost
26    accessModes:
27      - ReadWriteMany
28    resources:
29      requests:
30        storage: 10Gi
31

```

```

1  $ kubectl apply -f pg-pvc.yaml
2
3  $ kubectl get pv
4  NAME                 CAPACITY   ACCESS MODES   RECLAIM POLICY
5  postgres-data-pv    15Gi       RWX            Retain
6  Bound                devops/postgres-data-pvc  microk8s-localhost
7  3d
8
9  $ kubectl get pvc -n devops
10 NAME                 STATUS   VOLUME             CAPACITY   ACCESS
11 MODES   STORAGECLASS          AGE
12 postgres-data-pvc   Bound   postgres-data-pv   15Gi       RWX
13         microk8s-localhost  3d

```

configmap.yaml

```

1 apiVersion: v1
2 kind: ConfigMap
3 metadata:
4   name: cm-postgres
5   namespace: devops
6   labels:
7     app: postgres
8 data:
9   POSTGRES_PASSWORD: postgres

```

```

1 $ kubectl apply -f configmap.yaml
2
3 $ kubectl get cm -n devops
4 NAME                DATA  AGE
5 kube-root-ca.crt    1      3d5h
6 cm-postgres         1      2d2h
7

```

deployment.yaml

```

1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: gitea-postgres
5   namespace: devops
6 spec:
7   selector:
8     matchLabels:
9       app: postgres
10  replicas: 1
11  template:
12    metadata:
13      labels:
14        app: postgres
15    spec:
16      containers:
17        - name: gitea-postgres
18          image: postgres:12-alpine
19          ports:
20            - containerPort: 5432
21              name: pg-port
22          envFrom:
23            - configMapRef:
24              name: cm-postgres
25          volumeMounts:
26            - mountPath: /var/lib/postgresql/data
27              name: postgres-data-volume
28  volumes:

```

```

29 |         - name: postgres-data-volume
30 |           persistentVolumeClaim:
31 |             claimName: postgres-data-pvc

```

```

1 | $ kubectl apply -f deployment.yaml
2 |
3 | $ kubectl get po -n devops
4 | NAME                                READY   STATUS    RESTARTS   AGE
5 | gitea-postgres-86d6b8c4c7-rbt45    1/1     Running   0           2d1h
6 |

```

service.yaml

```

1 | apiVersion: v1
2 | kind: Service
3 | metadata:
4 |   name: postgres-service
5 |   namespace: devops
6 |   labels:
7 |     app: postgres
8 | spec:
9 |   type: NodePort
10 | ports:
11 | - port: 5432
12 |   targetPort: 5432
13 |   protocol: TCP
14 |   name: pg-port
15 |   nodePort: 30432
16 | selector:
17 |   app: postgres

```

```

1 | $ kubectl apply -f service.yaml
2 |
3 |
4 | $ kubectl get svc -n devops
5 | NAME                                TYPE           CLUSTER-IP           EXTERNAL-IP
6 | postgres-service                    NodePort       10.152.183.142       <none>
7 | 5432:30432/TCP                       2d1h

```

配置

`postgresql.conf`

```

1
2 # 时区修改
3 log_timezone = 'Asia/Shanghai'
4 timezone = 'Asia/Shanghai'
5
6
7 # 密码加密算法修改
8 password_encryption = scram-sha-256

```

`pg_hba.conf`

```

1 # 本地访问
2 local giteadb gitea
3   scram-sha-256
4
5 # 远程访问
6 host giteadb gitea 0.0.0.0/0
7   scram-sha-256

```

重启数据库

```

1 # 创建用户和数据库
2
3 CREATE ROLE gitea WITH LOGIN PASSWORD 'gitea';
4
5 CREATE DATABASE giteadb WITH OWNER gitea TEMPLATE template0
6   ENCODING UTF8 LC_COLLATE 'en_US.UTF-8' LC_CTYPE 'en_US.UTF-8';

```

部署 Gitea

`gitea-pvc.yaml`

```

1 kind: PersistentVolume
2 apiVersion: v1
3 metadata:

```

```
4   name: gitea-data-pv
5   labels:
6     type: local
7     app: gitea
8   spec:
9     storageClassName: microk8s-localhost
10    capacity:
11      storage: 500Gi
12    accessModes:
13      - ReadwriteMany
14    hostPath:
15      path: "/var/local/gitea/data"
16 ---
17 kind: PersistentVolumeClaim
18 apiVersion: v1
19 metadata:
20   name: gitea-data-pvc
21   namespace: devops
22   labels:
23     app: gitea
24   spec:
25     storageClassName: microk8s-localhost
26     accessModes:
27       - ReadwriteMany
28     resources:
29       requests:
30         storage: 200Gi
```

```

1 $ kubectl apply -f gitea-pvc.yaml
2
3 $ kubectl get pv
4 NAME                CAPACITY  ACCESS MODES  RECLAIM POLICY
   STATUS  CLAIM                STORAGECLASS  REASON
   AGE
5 postgres-data-pv    15Gi      RWX           Retain
   Bound   devops/postgres-data-pvc  microk8s-localhost
   3d
6 gitea-data-pv       500Gi     RWX           Retain
   Bound   devops/gitea-data-pvc    microk8s-localhost
   47h
7 $ kubectl get pvc -n devops
8 NAME                STATUS  VOLUME                CAPACITY  ACCESS
   MODES  STORAGECLASS          AGE
9 postgres-data-pvc   Bound   postgres-data-pv     15Gi      RWX
           microk8s-localhost   3d
10 gitea-data-pvc     Bound   gitea-data-pv        500Gi     RWX
           microk8s-localhost   47h
11

```

configmap.yaml

```

1
2 apiVersion: v1
3 kind: ConfigMap
4 metadata:
5   name: cm-gitea
6   namespace: devops
7   labels:
8     app: gitea
9 data:
10  DB_TYPE: "postgres"
11  DB_HOST: "postgres-service:5432"
12  DB_NAME: "giteadb"
13  DB_USER: "gitea"
14  DB_PASSWD: "gitea"
15

```

```
1 $ kubectl apply -f configmap.yaml
2
3 $ kubectl get cm -n devops
4 NAME          DATA  AGE
5 kube-root-ca.crt  1     3d5h
6 cm-postgres     1     2d2h
7 cm-gitea        6     29h
8
```

`deployment.yaml`

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: gitea
5   namespace: devops
6 spec:
7   selector:
8     matchLabels:
9       app: gitea
10  replicas: 1
11  template:
12    metadata:
13      labels:
14        app: gitea
15    spec:
16      containers:
17        - name: gitea
18          image: gitea/gitea:1.14.6
19          ports:
20            - containerPort: 22
21              name: ssh
22            - containerPort: 3000
23              name: http-port
24          envFrom:
25            - configMapRef:
26                name: cm-gitea
27          volumeMounts:
28            - mountPath: /data
29              name: gitea-data-volume
30      volumes:
31        - name: gitea-data-volume
32          persistentVolumeClaim:
33            claimName: gitea-data-pvc
34
```



```

1 $ kubectl apply -f deployment.yaml
2
3 $ kubectl get po -n devops
4 NAME                                READY   STATUS    RESTARTS   AGE
5 gitea-postgres-86d6b8c4c7-rbt45    1/1    Running   0           2d1h
6 gitea-c4b69d788-mdm7g              1/1    Running   0           6h10m
7

```

service.yaml

```

1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: gitea-service
5   namespace: devops
6   labels:
7     app: gitea
8 spec:
9   type: NodePort
10  ports:
11  - port: 22
12    targetPort: 22
13    protocol: TCP
14    name: ssh
15    nodePort: 30022
16  - port: 3000
17    targetPort: 3000
18    protocol: TCP
19    name: http-port
20    nodePort: 30300
21  selector:
22    app: gitea

```

```

1 $ kubectl apply -f service.yaml
2
3 $ kubectl get svc -n devops
4 NAME                                TYPE           CLUSTER-IP      EXTERNAL-IP
5 postgres-service                    NodePort       10.152.183.142  <none>
6   5432:30432/TCP                      2d1h
7 gitea-service                        NodePort       10.152.183.62   <none>
8   22:30022/TCP,3000:30300/TCP        29h

```

配置

```
1 [server]
2 APP_DATA_PATH = /data/gitea
3 DOMAIN = gitea.free-sun.vip
4 SSH_DOMAIN = gitea.free-sun.vip
5 HTTP_PORT = 3000
6 ROOT_URL = https://gitea.free-sun.vip/
7 DISABLE_SSH = false
8 SSH_PORT = 2022
9 SSH_LISTEN_PORT = 22
10 LFS_START_SERVER = false
11 LFS_CONTENT_PATH = /data/git/lfs
12 LFS_JWT_SECRET = ET6zJ0fRB193bjiHrUAzOxa7xeicpEmY9weiyqwwQqI
13 OFFLINE_MODE = false
14 LANDING_PAGE = explore
15
16
17 [mailer]
18 ENABLED = true
19 FROM = anxinyunwarning@free-sun.com.cn
20 MAILER_TYPE = smtp
21 HOST = smtp.exmail.qq.com:465
22 IS_TLS_ENABLED = true
23 USER = anxinyunwarning@free-sun.com.cn
24 PASSWD = `SGvd7FU7vesjj9su`
25
```

nginx 配置

```
1 # http 增加 server
2
3 upstream gitea-http {
4     server 192.168.0.121:30300;
5 }
6
7
8 server {
9     listen 80;
10    server_name gitea.free-sun.vip;
11    rewrite ^(.*) https://$server_name$1 permanent;
12 }
13
14
15 server {
16     listen 443 ssl;
17     server_name gitea.free-sun.vip;
18     client_max_body_size 5m;
19
```

```
20     ssl_certificate    /etc/nginx/certs/gitea.free-sun.pem;
21     ssl_certificate_key /etc/nginx/certs/gitea.free-sun.key;
22     ssl_session_timeout 5m;
23     ssl_ciphers ECDHE-RSA-AES128-GCM-
SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4;
24     ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
25     ssl_prefer_server_ciphers on;
26
27     location / {
28         client_max_body_size 20m;
29         proxy_redirect off;
30         proxy_set_header Host $host;
31         proxy_set_header X-Real-IP $remote_addr;
32         proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
33         proxy_pass http://gitea-http;
34         index index.html index.htm;
35     }
36 }
37
```

```
1 # stream 增加 server 做 ssh 访问
2 upstream gitea_backend {
3     hash $remote_addr consistent;
4     server 192.168.0.121:30022 max_fails=3 fail_timeout=30s;
5 }
6
7 server {
8     listen 2022 so_keepalive=on;
9     tcp_nodelay on;
10    proxy_pass gitea_backend;
11    proxy_connect_timeout 20s;
12    proxy_timeout 30m;
13    proxy_buffer_size 32k;
14 }
```

Uncle Dragon