

# CT 활용하기#2 - mariadb컨테이너 실행

---

#1 PV를 통해 mariadb컨테이너를 K-ECP의 Container Platform에 등록시키기

본 가이드에서는 K-ECP의 mariadb 이미지를 통해서 db 컨테이너를 K-ECP의 Container Platform에 등록시키는 과정을 안내합니다.

## 관련 안내서

- [Container 시작하기](#)
- [Container Terminal 시작하기](#)

## 목차

[전제 조건](#)

[1단계: PV, PVC 확인](#)

[2단계: mariadb 컨테이너 실행](#)

[3단계: mariadb 컨테이너 접속](#)

[다음단계](#)

---

## 전제 조건

- [Container\\_Terminal 시작하기](#)를 통하여 CT서비스를 할당 받아야 합니다.
- [PV 시작하기](#)를 통해서 PV를 생성해야 합니다.

---

## 1단계: PV, PVC 확인

1. K-ECP User Console에서 [서비스 현황 > 스토리지](#) 로 이동
2. 사전에 PV를 신청한 프로젝트의 돋보기 아이콘🔍 클릭
3. NAS 필드에서 신청한 [스토리지 ID](#) , [스토리지명](#) , [파일경로](#) 를 확인
4. [ContainerTerminal 시작하기](#)를 통해 CT서버에 접속 후 OpenShift계정 로그인

## 5. PVC 상태 확인(STATUS의 상태가 Bound임을 확인)

- 본가이드에서는 PV명 = edupv1, PVC명 = edupv1-claim 으로 진행

```
oc get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
edupv1-claim	Bound	edupv1	10Gi	RWX		15d

## 2단계: mariadb 컨테이너 pod 실행

### 1. CT 서버에서 pod yaml파일 작성

- mariadb-pod.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: mariadbpod
  labels:
    app: mariadbApp
  namespace: edu-ssg-test
spec:
  volumes:
    - name: datavol
      persistentVolumeClaim:
        claimName: edupv1-claim
  containers:
    - name: mariadb
      image: 'registry.redhat.io/rhel8/mariadb-105:latest'
      env:
        - name: MYSQL_ROOT_PASSWORD
          value: "1234"
      volumeMounts:
        - name: datavol
          mountPath: /var/lib/mysql
      ports:
        - containerPort: 3306
          protocol: TCP
```

💡 **안내:** image: 'registry.redhat.io/rhel8/mariadb-105:latest' 의 경우 사용자의 OS버전과 mariadb버전을 확인 후 작성해야 합니다.

💡 **안내:** mariadb의 이미지 정보는 다음과 같습니다. rhscl: Red Hat Software Collections (RHSC)(RHEL6, 7 버전) rhel8: 레드햇 8버전의 기본 컨테이너 이미지 mariadb-103: mariadb 10.3 버전 mariadb-105: mariadb 10.5 버전

- K-ECP의 OpenShift 컨테이너 플랫폼에서 사용 가능한 mariadb 이미지는 다음과 같습니다. (본 가이드의 경우 registry.redhat.io/rhel8/mariadb-105 를 사용합니다.)

```
registry.redhat.io/rhscl/mariadb-103-rhel7 registry.redhat.io/rhel8/mariadb-105
registry.redhat.io/rhscl/mariadb-103-rhel7 registry.redhat.io/rhel8/mariadb-103
registry.redhat.io/rhscl/mariadb-105-rhel7 registry.redhat.io/rhel8/mariadb-103
registry.redhat.io/rhel8/mariadb-105 registry.redhat.io/rhscl/mariadb-105-rhel7
```

## 2. mariadb Pod 생성(yaml파일을 저장한 경로에서 실행)

```
oc create -f mariadb-pod.yaml
```

```
pod/mariadbpod created
```

## 3. Pod 생성 확인

```
oc get pod
```

NAME	READY	STATUS	RESTARTS	AGE
mariadbpod	1/1	Running	0	38s

## 4. Pod 내부 접속 테스트

```
oc rsh mariadbpod
```

```
sh-4.4$
```

## 5. Pod 내부에서 PV가 마운트 되었는지 확인

```
df -h
```

- 2단계에서 작성한 yaml의 mountPath로 PV가 마운트된 것을 확인

Filesystem	Size	Used	Avail	Use%	Mounted on
overlay	446G	64G	383G	15%	/
tmpfs	64M	0	64M	0%	/dev
tmpfs	504G	0	504G	0%	/sys/fs/cgroup
shm	64M	0	64M	0%	/dev/shm
tmpfs	504G	64M	504G	1%	/etc/hostname
/dev/sda4	446G	64G	383G	15%	/etc/hosts
[PV_IP]:/EDU-SSG-TEST/edupv1	10G	334M	9.7G	4%	/var/lib/mysql
tmpfs	2.0G	20K	2.0G	1%	
/run/secrets/kubernetes.io/serviceaccount					
tmpfs	504G	0	504G	0%	/proc/acpi
tmpfs	504G	0	504G	0%	/proc/scsi
tmpfs	504G	0	504G	0%	/sys/firmware

## 6. mariadb 접속 테스트

```
mariadb -u root
```

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 3

Server version: 10.5.16-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>

## 7. database, user 생성

- database 생성

```
create database testdb;
```

- user 생성(본 가이드에서는 user명을 testuser로 지정)

```
create user 'testuser'@'%' identified by '1234';
```

- 생성한 testuser가 testdb로 접근할 수 있도록 권한 설정

```
grant all privileges on testdb.* to 'testuser'@'%';
```

- INSERT, UPDATE, DELETE를 통해 DB 정보를 변경하였을 때, 변경사항을 적용하기 위한 명령어 설정

```
flush privileges;
```

8. root계정 로그아웃 후 생성한 계정정보로 mariadb 접속

```
exit
```

```
mariadb -u testuser -p
```

- 설정한 비밀번호 입력 후 mariadb 접속

Enter password:

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 4

Server version: 10.5.16-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

9. testuser로 접근 가능한 DB 확인

```
show databases;
```

```
+-----+
| Database          |
+-----+
| information_schema |
| testdb            |
+-----+
2 rows in set (0.015 sec)
```

---

## 3단계: mariadb 컨테이너 서비스 생성, 외부 툴 접속

---

## 1. CT 서버에서 서비스 yaml파일 작성

- mariadb-svc.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: mariadb-svc
  namespace: edu-ssg-test
spec:
  selector:
    app: mariadbApp
  type: NodePort
  ports:
    - protocol: TCP
      nodePort: 32306
      port: 3306
      targetPort: 3306
```

## 2. 서비스 생성(yaml파일을 저장한 경로에서 실행)

```
oc create -f mariadb-svc.yaml
```

```
service/mariadbpod created
```

## 3. 생성된 서비스 확인

```
oc get service
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
mariadb-svc	NodePort	172.30.33.187	<none>	3306:32306/TCP	4s

## 4. 서비스를 외부로 노출

```
oc expose service/mariadb-svc
```

```
route.route.openshift.io/mariadb-svc exposed
```

## 5. 외부 툴을 통한 접속 테스트

⚠ **주의사항:** 방화벽 작업이 필요합니다. (32306 포트) 출발지(AP or 개발PC) 목적지 : HA Proxy (10.100.11.120)

---

## 다음단계

---

- CT 활용하기 #2 를 통해 실행중인 Container를 관리할 수 있습니다.