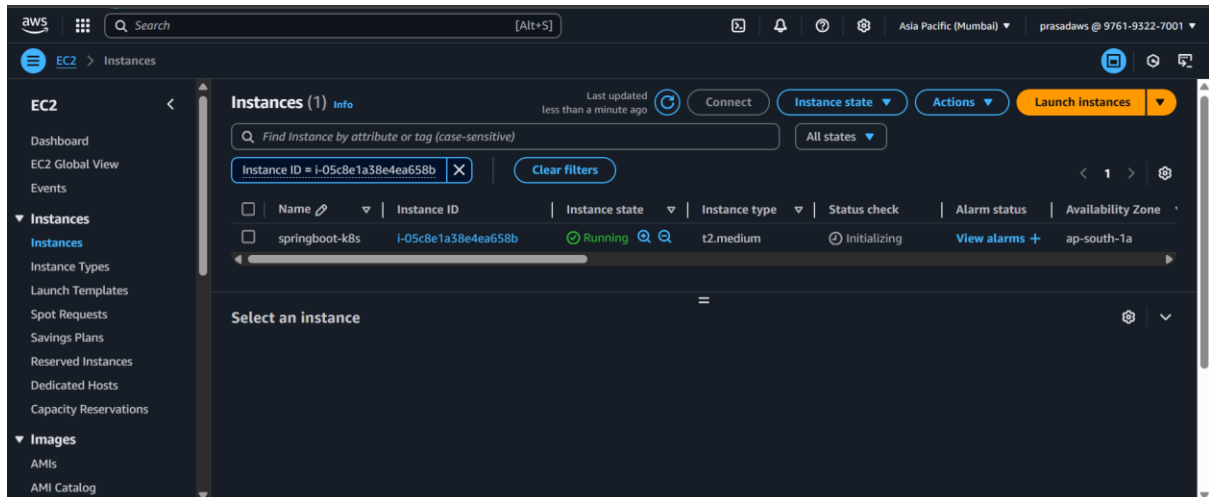


# Deploying Spring Boot application on Kubernetes

## Create EC2 instance:



## Install Docker:

```
sudo apt update -y
```

```
sudo apt install -y docker.io
```

```
sudo systemctl enable docker
```

```
sudo systemctl start docker
```

```
docker --version
```

```
ubuntu@ip-172-31-44-46:~$ docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
ubuntu@ip-172-31-44-46:~$
```

## Install Minikube:

```
ubuntu@ip-172-31-44-46:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload  Total   Spent    Left    Speed
100 119M  100 119M    0     0  11.4M      0  0:00:10  0:00:10 --:--:-- 15.7M
ubuntu@ip-172-31-44-46:~$
```

## Verify Minikube installation:

```
ubuntu@ip-172-31-44-46:~$ minikube version
minikube version: v1.35.0
commit: dd5d320e41b5451cdf3c01891bc4e13d189586ed-dirty
ubuntu@ip-172-31-44-46:~$
```

## Install Kubectl:

```
ubuntu@ip-172-31-44-46:~$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100  138    100  138    0     0    434      0  --:--:-- --:--:-- --:--:--    435
100 54.6M    100 54.6M    0     0  15.6M      0  0:00:03 0:00:03 --:--:--  17.4M
ubuntu@ip-172-31-44-46:~$
```

## Verify kubectl installation:

```
ubuntu@ip-172-31-44-46:~$ kubectl version --client
Client Version: v1.32.3
Kustomize Version: v5.5.0
ubuntu@ip-172-31-44-46:~$
```

## Install Git:

```
ubuntu@ip-172-31-44-46:~$ sudo apt install -y git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.2).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 143 not upgraded.
ubuntu@ip-172-31-44-46:~$
```

## Clone the Repository:

```
ubuntu@ip-172-31-44-46:/opt$ sudo git clone https://github.com/Prasadrasal2002/Spring-Boot-application.git
Cloning into 'Spring-Boot-application'...
remote: Enumerating objects: 102, done.
remote: Counting objects: 100% (102/102), done.
remote: Compressing objects: 100% (71/71), done.
remote: Total 102 (delta 16), reused 73 (delta 4), pack-reused 0 (from 0)
Receiving objects: 100% (102/102), 33.05 MiB | 7.71 MiB/s, done.
Resolving deltas: 100% (16/16), done.
ubuntu@ip-172-31-44-46:/opt$
```

```
ubuntu@ip-172-31-44-46:/opt$ ls
Spring-Boot-application  containerd
ubuntu@ip-172-31-44-46:/opt$ cd Spring-Boot-application/
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-application$
```

## Set Up Database in Kubernetes:

```
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-application$ sudo kubectl apply -f db-pvc.yml
persistentvolumeclaim/mysql-pv-claim created
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-application$ sudo kubectl get pods
```

```
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-application$ sudo kubectl apply -f db-config.yml
configmap/mysql-config created
```

```
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-application$ sudo kubectl apply -f db-secret.yml
secret/mysql-secret created
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-application$ sudo kubectl apply -f db-statefulset.yml
statefulset.apps/mysql created
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-application$ sudo kubectl get pod
NAME      READY   STATUS    RESTARTS   AGE
mysql-0   1/1     Running   0           20s
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-application$
```

## Check the database inside the container:

```
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-applications$ sudo kubectl exec -it mysql-0 -- /bin/bash
bash-4.2# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.44 MySQL Community Server (GPL)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql>
```

## Check the Table Schema:

DESC orders\_tbl;

```
mysql> DESC orders_tbl;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | |
| name | varchar(255) | YES | | NULL | |
| price | double | NO | | NULL | |
| qty | int(11) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

- **Remove the Existing Primary Key:**
- **Command:** ALTER TABLE orders\_tbl DROP PRIMARY KEY;

```
mysql> ALTER TABLE orders_tbl DROP PRIMARY KEY;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

- **Alter the Table to Add AUTO\_INCREMENT:**
- **Command:** ALTER TABLE orders\_tbl MODIFY COLUMN id INT AUTO\_INCREMENT PRIMARY KEY;

```
mysql> ALTER TABLE orders_tbl MODIFY COLUMN id INT NOT NULL AUTO_INCREMENT PRIMARY KEY;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

- **Then try inserting a record again:**
- **Command:** INSERT INTO orders\_tbl (name, price, qty) VALUES ('Test Order', 100, 2);

```
mysql> DESC orders_tbl;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| name | varchar(255) | YES | | NULL | |
| price | double | NO | | NULL | |
| qty | int(11) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> INSERT INTO orders_tbl (name, price, qty) VALUES ('Test Order', 100, 2);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM orders_tbl;
+----+-----+-----+-----+
| id | name | price | qty |
+----+-----+-----+-----+
| 1 | Test Order | 100 | 2 |
+----+-----+-----+-----+
1 row in set (0.00 sec)
```

## Build Docker Image:

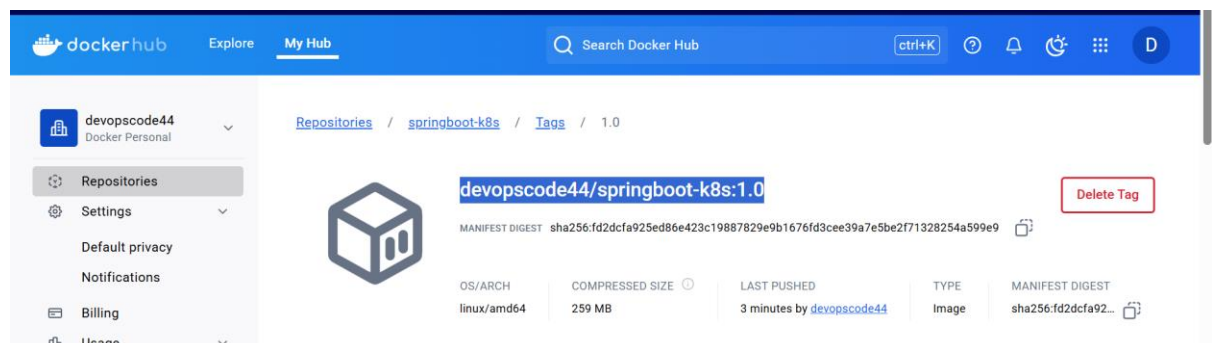
```
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-applications$ sudo docker build -t devopscod44/springboot-k8s:1.0 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with Buildkit.
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 73.36MB
Step 1/4 : FROM openjdk:8
8: Pulling from library/openjdk
001c2a2b6d5: Pull complete
d9d4b9b6e964: Pull complete
2668746827ec: Pull complete
8aef3294350: Pull complete
485151f15666: Pull complete
52a8c426d30b: Pull complete
8754a6060050: Pull complete
Digest: sha256:86e983c57215cfb181bd319736d0haf625fe8f150577f9eb58bd937f5452cb8
Status: Downloaded newer image for openjdk:8
--> b273004037cc
Step 2/4 : EXPOSE 8080
--> Running in b70b7d2cbd8e
--> Removed intermediate container b70b7d2cbd8e
--> c83883a5509
Step 3/4 : ADD target/springboot-crud-k8s.jar springboot-crud-k8s.jar
--> eac0407e164e
Step 4/4 : ENTRYPOINT ["java","-jar","/springboot-crud-k8s.jar"]
--> Running in 49e16c631597
--> Removed intermediate container 49e16c631597
--> 2ea4e9749b10
Successfully built 2ea4e9749b10
Successfully tagged devopscod44/springboot-k8s:1.0
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-applications$ sudo docker images
REPOSITORY          TAG             IMAGE ID          CREATED           SIZE
devopscod44/springboot-k8s   1.0             2ea4e9749b10     20 seconds ago   565MB
gcr.io/k8s-minikube/kicbase   v0.0.46         e72c4cbe9b29     2 months ago     1.31GB
openjdk               8               b273004037cc     2 years ago      526MB
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-applications$
```

## Push the image to Docker Hub:

```
Login Succeeded
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-applications$ sudo docker push devopscod44/springboot-k8s:1.0
The push refers to repository [docker.io/devopscod44/springboot-k8s]
0d01012db8fc: Pushed
0b5aaff44254: Mounted from library/openjdk
53a0b163e995: Mounted from library/openjdk
0026401e1693: Mounted from library/openjdk
8b5156abf26: Mounted from library/openjdk
293d5db30c9f: Mounted from library/openjdk
09127c0b479b: Mounted from library/openjdk
9c742c46c745: Mounted from library/openjdk
1.0: digest: sha256:fd2dcfa925ed86e423c19887829e9b1676fd3cee39a7e5be2f71328254a599e9 size: 2007
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-applications$
```

## Docker Hub:



The screenshot shows the Docker Hub interface for the repository `devopscod44/springboot-k8s:1.0`. The page includes a sidebar with navigation options like Repositories, Settings, and Default privacy. The main content area displays the repository name, a manifest digest, and a table of image details.

OS/ARCH	COMPRESSED SIZE	LAST PUSHED	TYPE	MANIFEST DIGEST
linux/amd64	259 MB	3 minutes by devopscod44	Image	sha256:fd2dcfa92...

## Deploy the Application in Kubernetes:

```
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-applications$ sudo kubectl apply -f app-deployment.yml
deployment.apps/springboot-crud configured

ubuntu@ip-172-31-44-46:/opt/Spring-Boot-applications$ sudo kubectl get pod
NAME                READY   STATUS    RESTARTS   AGE
mysql-0             1/1     Running   0           21m
springboot-crud-684b9c9b4f-g2sr1  1/1     Running   0           36s
springboot-crud-684b9c9b4f-1gthd  1/1     Running   0           34s
springboot-crud-684b9c9b4f-s69kd  1/1     Running   0           55s
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-applications$

ubuntu@ip-172-31-44-46:/opt/Spring-Boot-applications$ sudo kubectl apply -f db-service.yml
service/mysql created
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-applications$ sudo kubectl apply -f app-service.yml
service/springboot-service created
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-applications$ sudo kubectl get svc
NAME                TYPE                CLUSTER-IP      EXTERNAL-IP  PORT(S)          AGE
kubernetes          ClusterIP           10.96.0.1        <none>        443/TCP          37m
mysql               ClusterIP           None             <none>        3306/TCP          36s
springboot-service  NodePort            10.98.150.78     <none>        8080:30008/TCP   19s
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-applications$
```

## Port Forwarding:

```
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-application$ kubectl port-forward --address 0.0.0.0 svc/springboot-service 8080:8080 &
[1] 104186
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-application$ Forwarding from 0.0.0.0:8080 -> 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
ubuntu@ip-172-31-44-46:/opt/Spring-Boot-application$ kubectl proxy --address='0.0.0.0' --accept-hosts='^*$'
Starting to serve on [::]:8001
Handling connection for 8080
```

## Spring Boot application:.



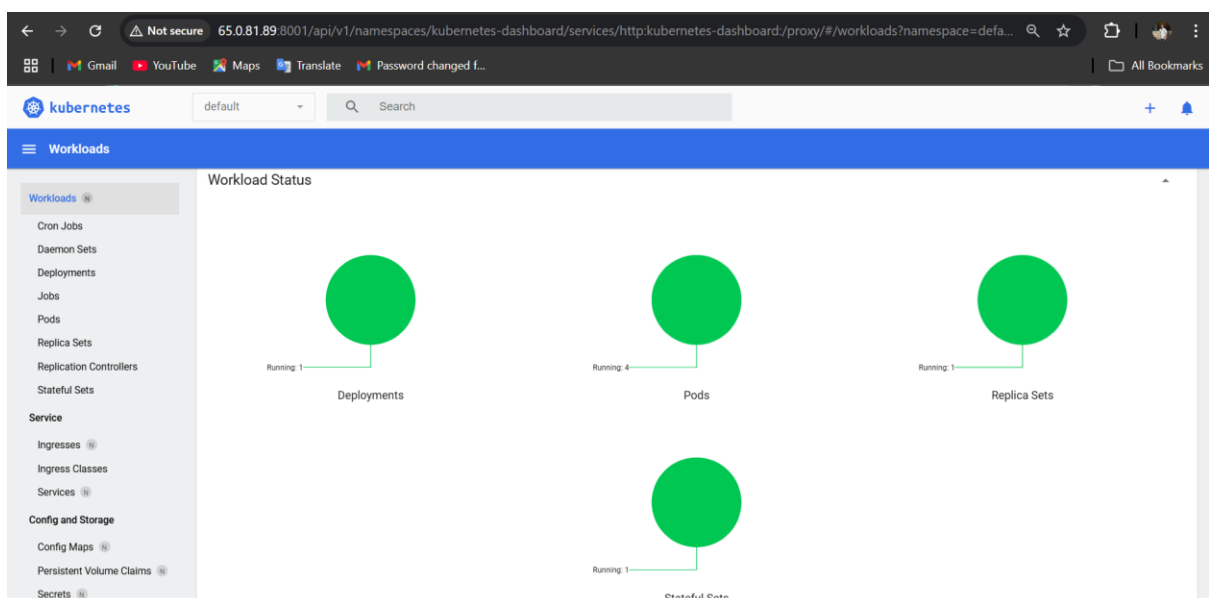
Not secure 65.0.81.89:8080/orders

Pretty print ☐

```
[{"id":1,"name":"Test Order","qty":2,"price":100.0}]
```

## Kubernetes dashboard:

### i)workload:



### ii)Deployments:

← → ↻ Not secure 65.0.81.89:8001/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard/proxy/#/deployment?namespace=de... ☆ 📄 | 👤 : All Bookmarks

📦 Gmail 📺 YouTube 🗺 Maps 🌐 Translate 📧 Password changed f...

kubernetes default 🔍 Search + 🔔

Workloads > Deployments

Workloads ⓘ

- Cron Jobs
- Daemon Sets
- Deployments
- Jobs

Deployments

Name	Images	Labels	Pods	Created ↑
springboot-crud	devopscode44/springboot-k8s:1.0	-	3 / 3	an hour ago

### iii)Pods:

← → ↻ Not secure 65.0.81.89:8001/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard/proxy/#/pod?namespace=default ☆ 📄 | 👤 : All Bookmarks

📦 Gmail 📺 YouTube 🗺 Maps 🌐 Translate 📧 Password changed f...

kubernetes default 🔍 Search + 🔔

Workloads > Pods

Workloads ⓘ

- Cron Jobs
- Daemon Sets
- Deployments
- Jobs
- Pods**
- Replica Sets
- Replication Controllers
- Stateful Sets

Service

- Ingresses ⓘ
- Ingress Classes
- Services ⓘ

Config and Storage

Pods

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
springboot-crud-684b9c9b4f-9hs4s	devopscode44/springboot-k8s:1.0	app: springboot pod-template-hash: 684b9c9b4f	minikube	Running	0	-	-	an hour ago
springboot-crud-684b9c9b4f-msz2x	devopscode44/springboot-k8s:1.0	app: springboot pod-template-hash: 684b9c9b4f	minikube	Running	0	-	-	an hour ago
springboot-crud-684b9c9b4f-yfscd	devopscode44/springboot-k8s:1.0	app: springboot pod-template-hash: 684b9c9b4f	minikube	Running	0	-	-	an hour ago
mysql-0	mysql:5.7	app: mysql apps.kubernetes.io/pod-index: 0 controller-revision-hash: mysql-8987557d9	minikube	Running	0	-	-	an hour ago

Show all

### iv)Replica sets:

← → ↻ Not secure 65.0.81.89:8001/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard/proxy/#/replicaset?namespace=defa... ☆ 📄 | 👤 : All Bookmarks

📦 Gmail 📺 YouTube 🗺 Maps 🌐 Translate 📧 Password changed f...

kubernetes default 🔍 Search + 🔔

Workloads > Replica Sets

Workloads ⓘ

- Cron Jobs
- Daemon Sets
- Deployments
- Jobs
- Pods
- Replica Sets**
- Replication Controllers

Replica Sets

Name	Images	Labels	Pods	Created ↑
springboot-crud-684b9c9b4f	devopscode44/springboot-k8s:1.0	app: springboot pod-template-hash: 684b9c9b4f	3 / 3	an hour ago

## v)Stateful sets:

The screenshot shows the Kubernetes dashboard interface. The top navigation bar includes the Kubernetes logo, a search bar, and a user profile icon. The left sidebar lists various workload types: Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, and Stateful Sets (which is currently selected). The main content area is titled 'Stateful Sets' and contains a table with the following data:

Name	Images	Labels	Pods	Created ↑
mysql	mysql:5.7	-	1 / 1	an hour ago

## vi)Services:

The screenshot shows the Kubernetes dashboard interface. The top navigation bar includes the Kubernetes logo, a search bar, and a user profile icon. The left sidebar lists various workload types: Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, and Service (which is currently selected). The main content area is titled 'Services' and contains a table with the following data:

Name	Labels	Type	Cluster IP	Internal Endpoints	External Endpoints	Created ↑
springboot-service	-	NodePort	10.101.143.18	springboot-service:8080 TCP springboot-service:30008 TCP	-	an hour ago
mysql	-	ClusterIP	None	mysql:3306 TCP mysql:0 TCP	-	an hour ago
kubernetes	component: apiserver provider: kubernetes	ClusterIP	10.96.0.1	kubernetes:443 TCP kubernetes:0 TCP	-	an hour ago

## vii)Config Maps, PVC, Secrets:

←

→

↻

⚠ Not secure

65.0.81.89:8001/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard/proxy/#/config?namespace=default

🔍

☆

📄

👤

⋮

📧 Gmail

📺 YouTube

🗺 Maps

🌐 Translate

🔑 Password changed f...

All Bookmarks

kubernetes

default

🔍 Search

+

🔔

☰ Config And Storage

Cron Jobs

Daemon Sets

Deployments

Jobs

Pods

Replica Sets

Replication Controllers

Stateful Sets

Service

Ingresses ⓘ

Ingress Classes

Services ⓘ

Config and Storage

Config Maps ⓘ

Persistent Volume Claims ⓘ

Secrets ⓘ

Storage Classes

Cluster

Cluster Role Bindings

Cluster Roles

Config Maps

Name	Labels	Created ↑	
<a href="#">mysql-config</a>	-	an hour ago	⋮
<a href="#">kube-root-ca.crt</a>	-	an hour ago	⋮

Persistent Volume Claims

Name	Labels	Status	Volume	Capacity	Access Modes	Storage Class	Created ↑	
<div><div></div><a href="#">mysql-pv-claim</a></div>	<a href="#">app: mysql</a>	Bound	<a href="#">pvc-28e3bdb5-78e3-4167-8ec7-ad4e5b134653</a>	1Gi	<a href="#">ReadWriteOnce</a>	standard	an hour ago	⋮

Secrets

Name	Labels	Type	Created ↑	
<a href="#">mysql-secret</a>	-	Opaque	an hour ago	⋮

Storage Classes

Name	Provisioner	Parameters	Created ↑	
<a href="#">standard</a>	k8s.io/minikube-hostpath	-	an hour ago	⋮