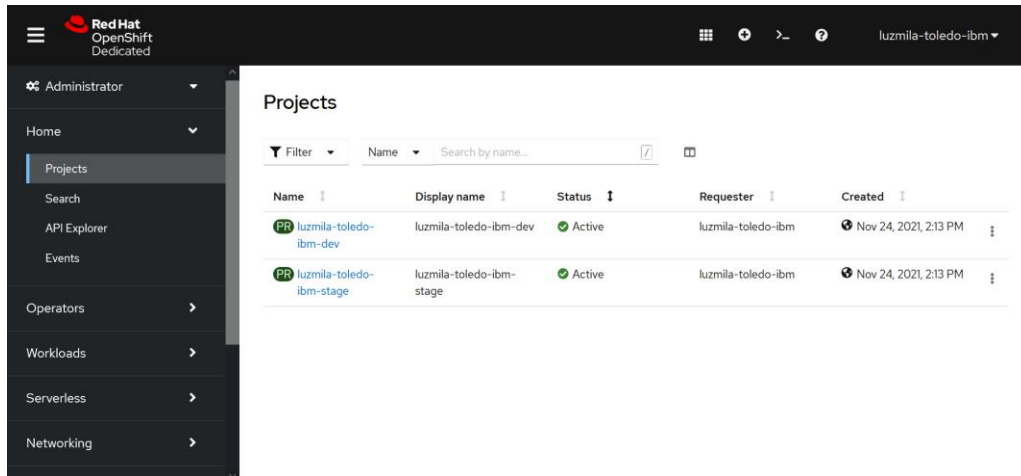


Laboratorio #1: Despliegue de un app via consola web:

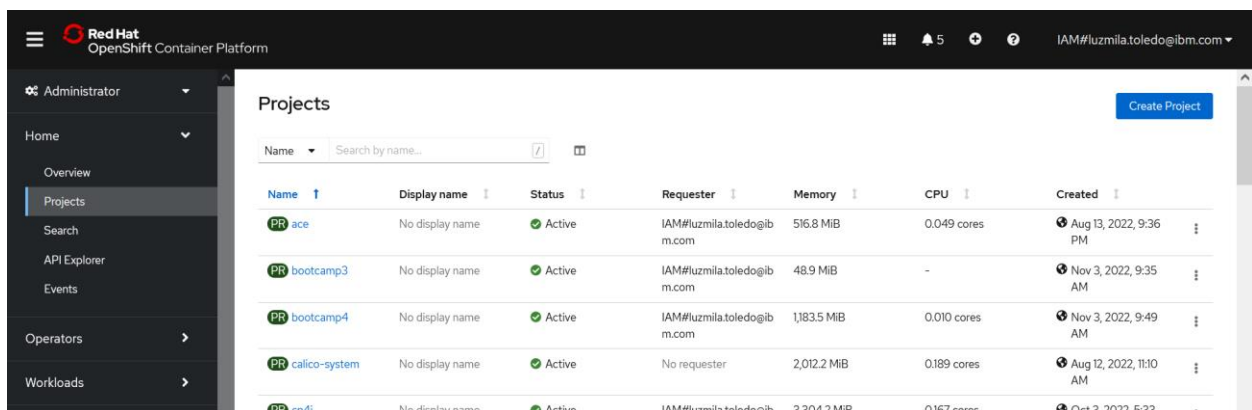
Ingresamos al ambiente de pruebas de Red Hat y veremos la consola principal de Red Hat OpenShift Container Platform



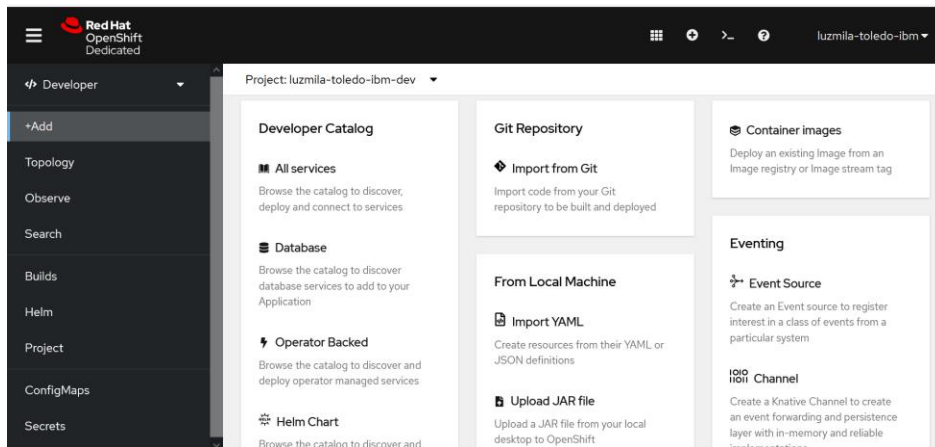
Nos tomamos unos minutos para explorar el cluster desde la vista de Administradores:

- Revisar Proyectos
- Installed Operators
- Workloads
- Storage Classes

Creamos un nuevo proyecto con nuestros nombres. En la vista Administrador > Projects > Create Project

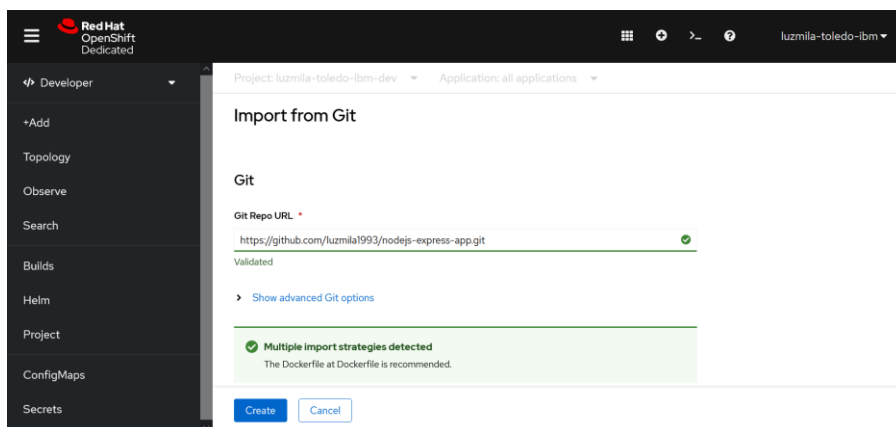
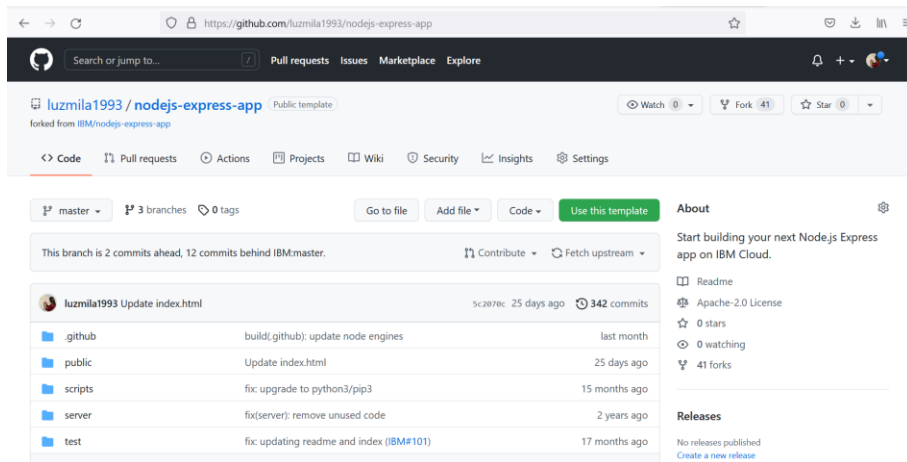


Le damos click a +Add y exploramos las alternativas de creación de aplicaciones que disponemos



Le damos click a “Git repository” > Import from Git y colocamos la URL del siguiente repo:

<https://github.com/luzmila1993/nodejs-express-app.git> (Esta es una aplicación sencilla node.js)



Revisamos que la URL sea validada y nos debería de aparecer un check verde al costado del Git Repo URL.

Ingresamos un nombre para la aplicación:

Application Name: nodejs-demo

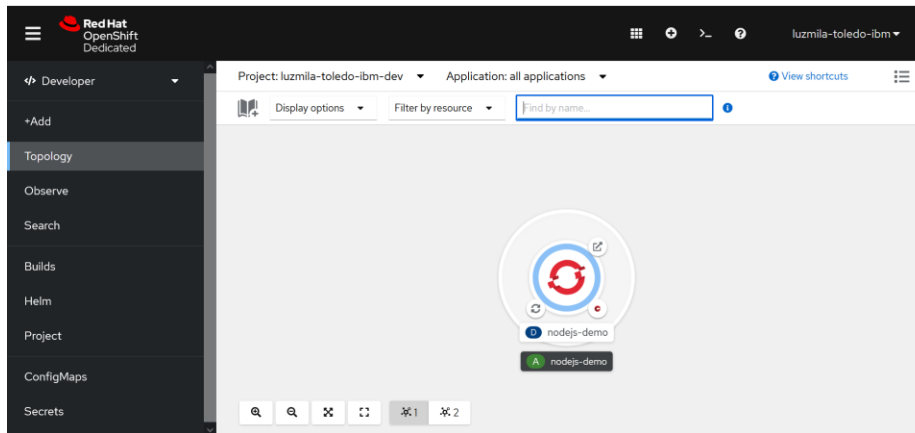
Name: nodejs-demo

Resources: Deployment

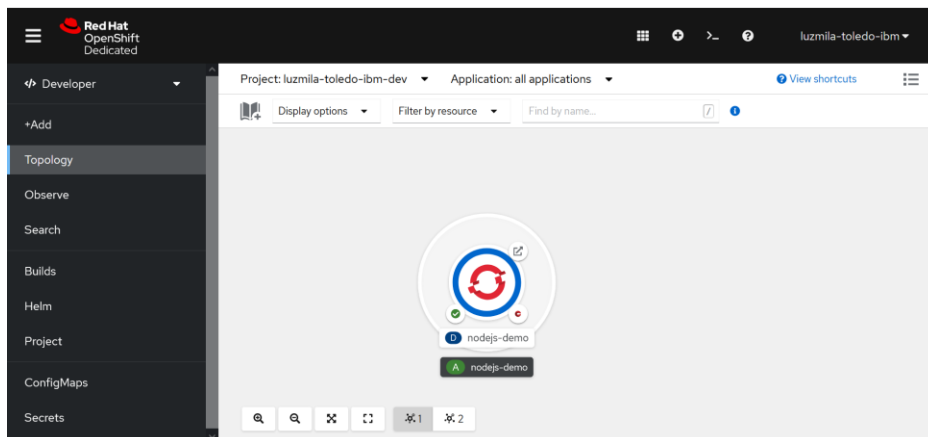
Create a Route to the Application: check

The screenshot shows the Red Hat OpenShift console interface. On the left is a sidebar with navigation options: Developer, +Add, Topology, Observe, Search, Builds, Helm, Project, ConfigMaps, and Secrets. The main panel is titled 'Project: luzmila-toledo-ibm-dev' and 'Application: all applications'. It contains a form for creating a new application. The 'Application name' field is filled with 'Nodejs demo'. Below it, the 'Name' field is filled with 'nodejs-demo'. Under the 'Resources' section, 'Deployment' is selected as the resource type to generate. At the bottom of the form are 'Create' and 'Cancel' buttons.

Le damos click a Create y veremos como la aplicación se empieza a crear:

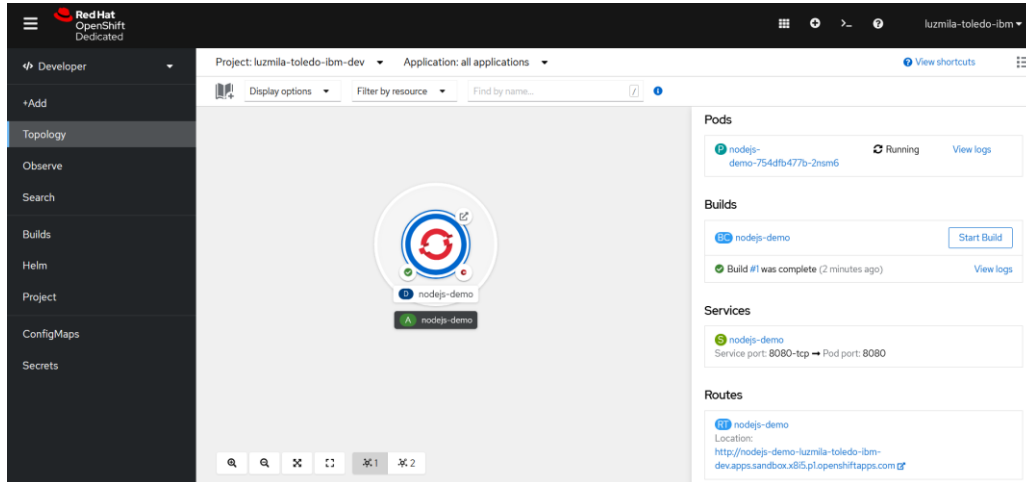


Tenemos que esperar hasta que el círculo esté en azul oscuro para que se encuentre en estado Running

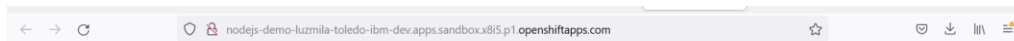


Le damos click y revisamos la información del pod creado, el service y la ruta.

Nota: Para este escenario no es necesario realizar previamente el build, crear la imagen, hacer el push al image registry. Con este escenario uno ahorra todo ese trabajo, únicamente se debe tener acceso al github, Openshift detecta automáticamente el lenguaje de programación, compila y me crea el deployment, servicio y ruta.



Luego, le doy click a la URL de Routes y veremos que obtendremos un error. Este error significa que la aplicación no está corriendo o que se está apuntando a una aplicación que no es la correcta.



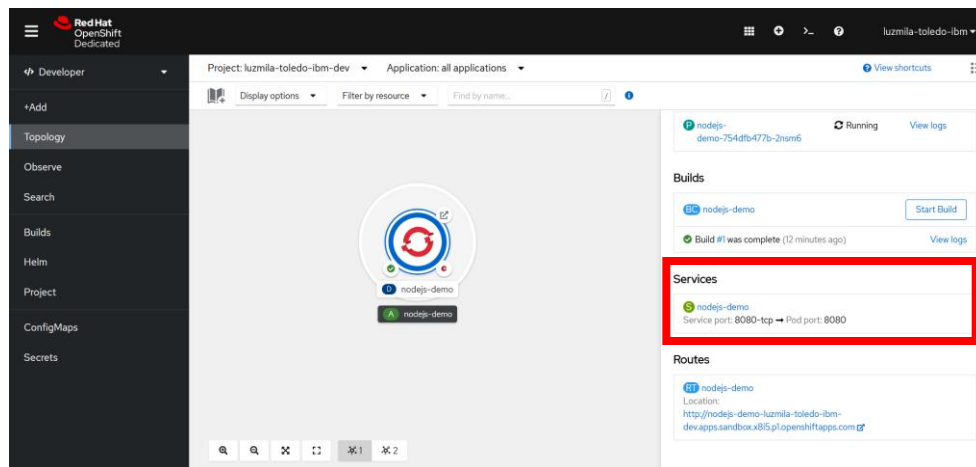
Application is not available

The application is currently not serving requests at this endpoint. It may not have been started or is still starting.

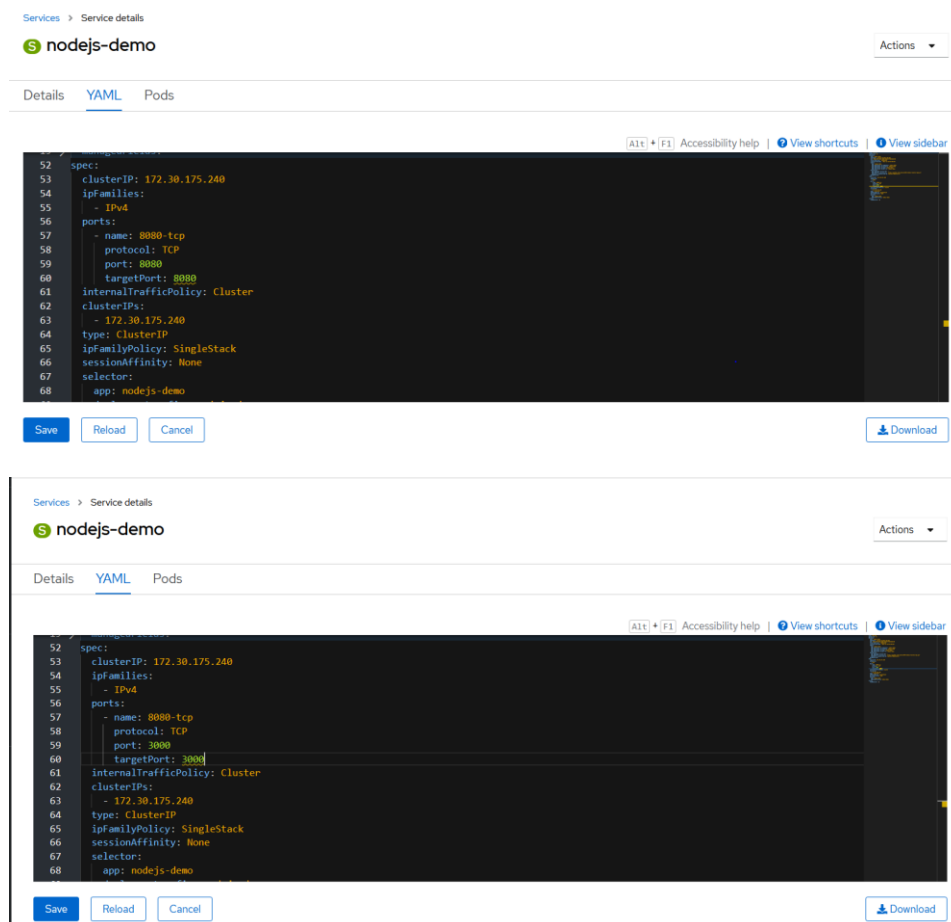
Possible reasons you are seeing this page:

- **The host doesn't exist.** Make sure the hostname was typed correctly and that a route matching this hostname exists.
- **The host exists, but doesn't have a matching path.** Check if the URL path was typed correctly and that the route was created using the desired path.
- **Route and path matches, but all pods are down.** Make sure that the resources exposed by this route (pods, services, deployment configs, etc) have at least one pod running.

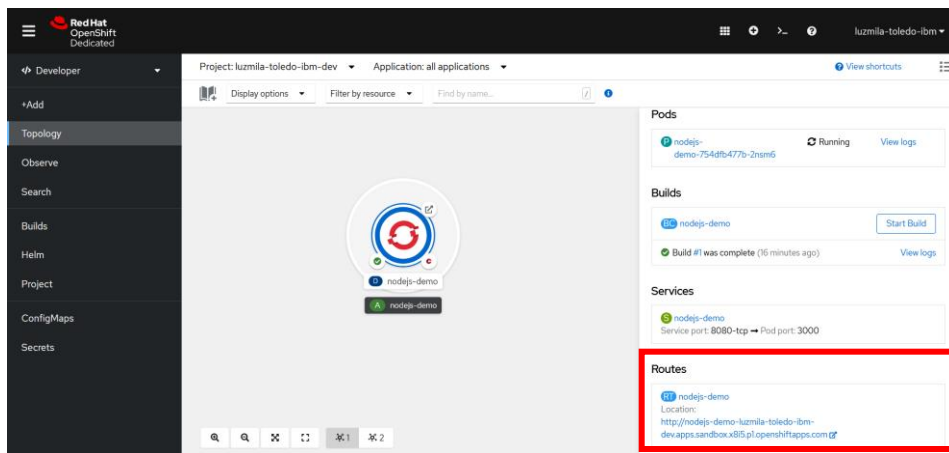
Este error nos está apareciendo porque estamos apuntando a un puerto incorrecto. Se está apuntando al puerto 8080 y el que utiliza nodejs es 3000. Le damos click a Services > nodejs-demo



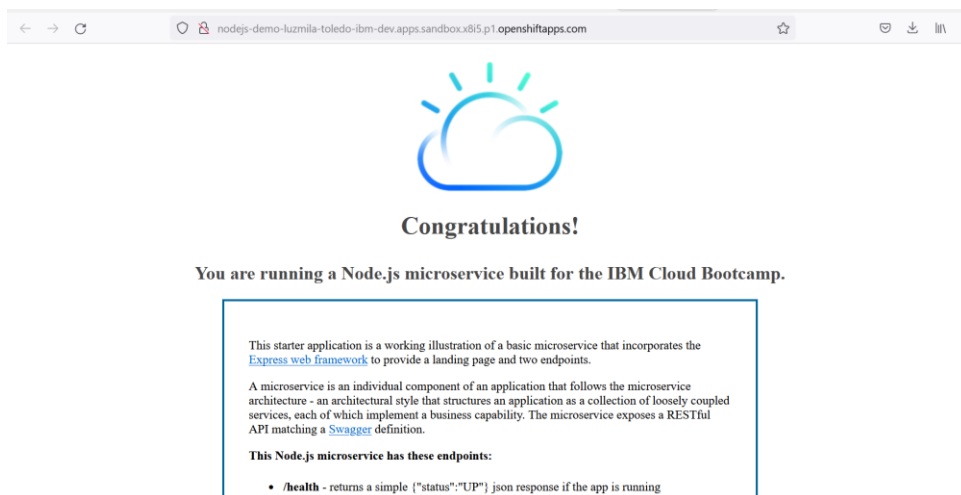
Vamos a la pestaña de YAML y buscamos la sección de ports y cambiamos el puerto 8080 a 3000 y le damos click a Save



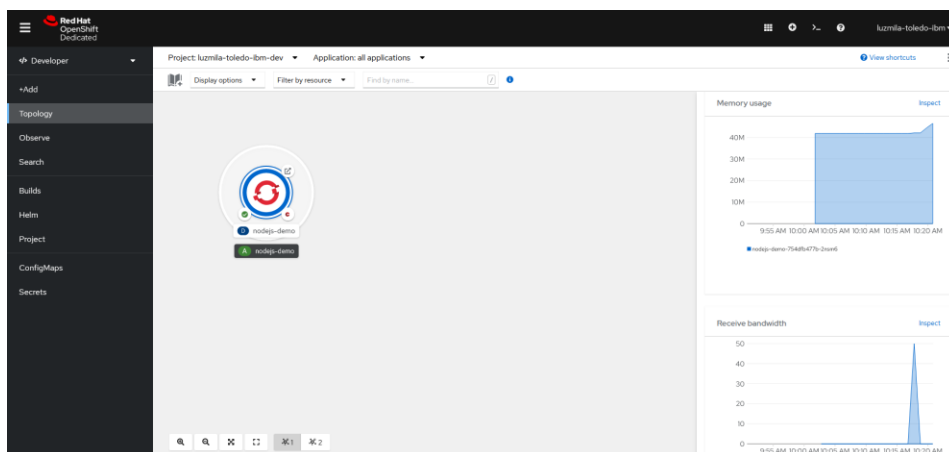
Regresamos a la vista de Topology y nuevamente le damos click a la ruta generada para la aplicación



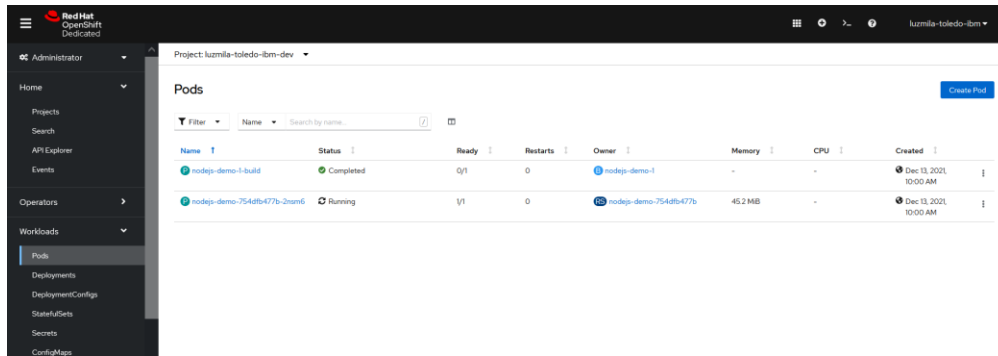
Listo! Tenemos nuestra primera aplicación de Openshift desplegada!!



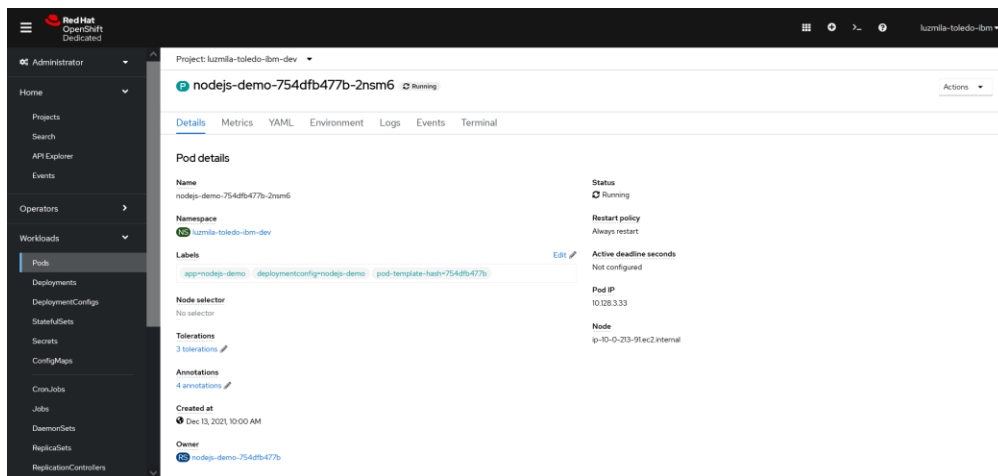
Nos vamos a la pestaña de Observe para ver el consumo de CPU y memoria del pod generado



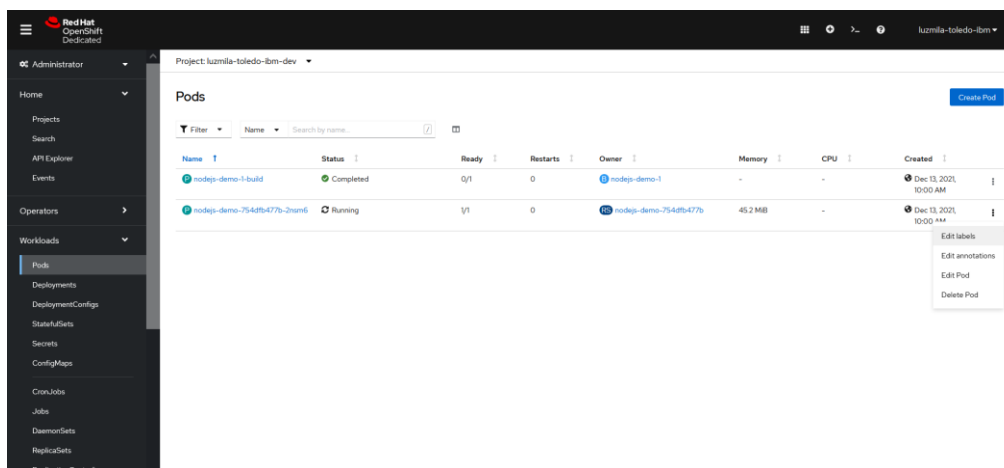
Ahora vamos a la vista de Administrator y vamos a Workloads > Pods



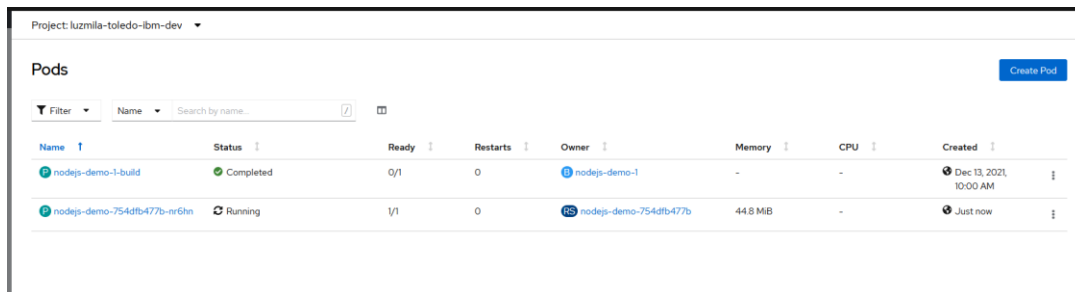
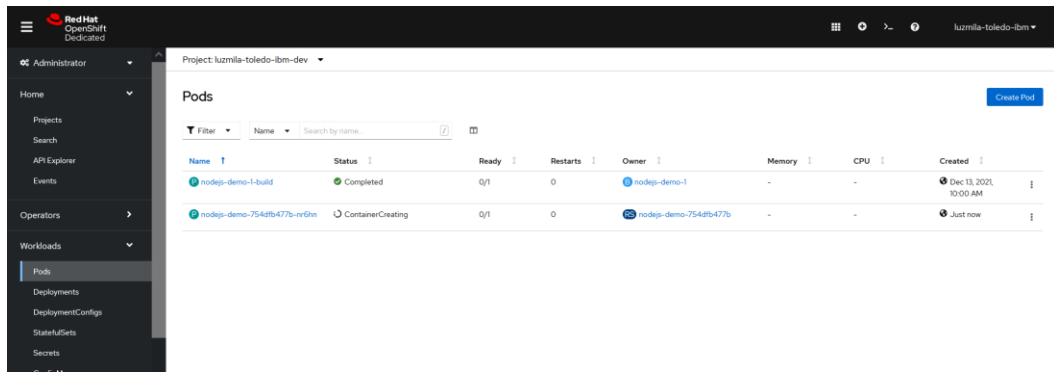
Ingresamos al pod creado en estado Running y visualizamos sus detalles



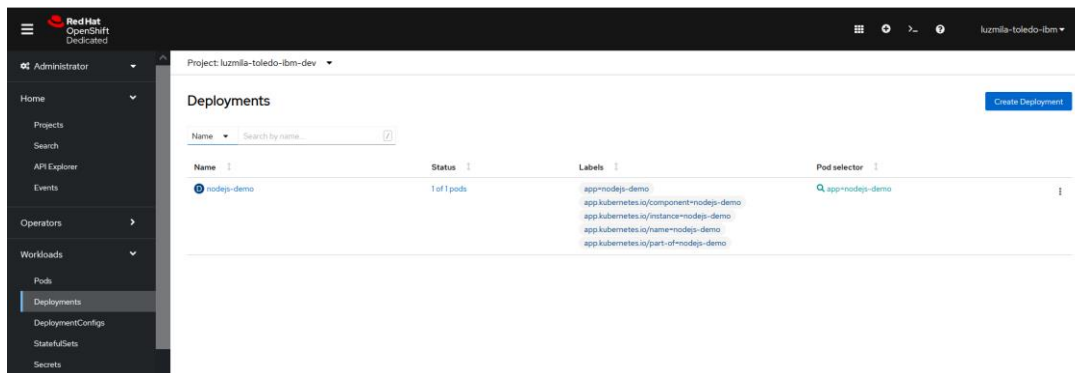
Ahora, procederemos a Eliminar el pod para visualizar como automáticamente se reconstruye. Para ello le damos click en los puntos al final del nombre y seleccionamos Delete Pod



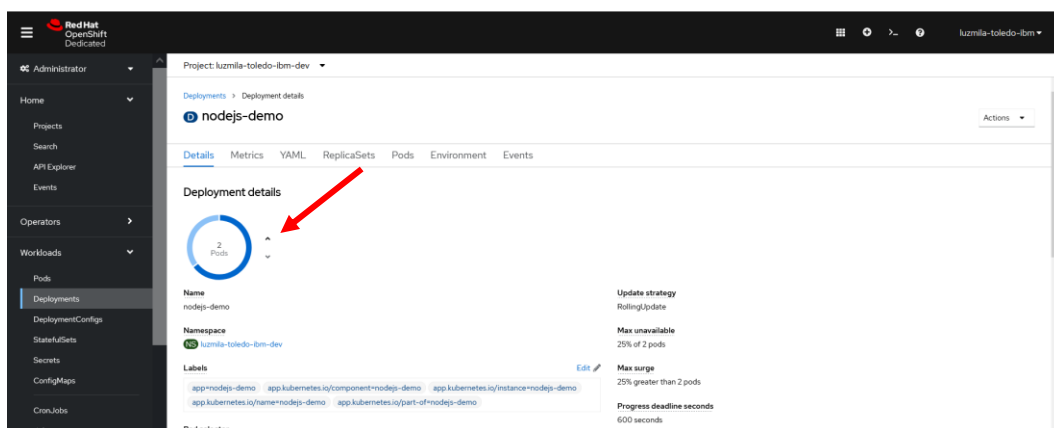
Vemos que el pod se elimina pero automáticamente se vuelve a crear otro (Estado Container Creating y luego Running)



Finalmente, vamos a incrementar el número de replicas de 1 a 2, para lo cual nos vamos a Deployments le damos click a nodejs-demo



El escalamiento lo podemos realizar a través de la adaptación del YAML o dando click en la flecha hacia arriba en Deployment Details



Luego regresamos a Workloads > Pods y vemos que se creo un pod adicional con nuestra aplicación:

Project: luzmila-toledo-ibm-dev

Pods Create Pod

Filter Name Search by name...

Name	Status	Ready	Restarts	Owner	Memory	CPU	Created
nodejs-demo-1-build	Completed	0/1	0	nodejs-demo-1	-	-	Dec 13, 2021, 10:00 AM
nodejs-demo-754dfb477b-j89t	Running	1/1	0	nodejs-demo-754dfb477b	42.6 MB	-	4 minutes ago
nodejs-demo-754dfb477b-nr6hn	Running	1/1	0	nodejs-demo-754dfb477b	46.7 MB	0.000 cores	8 minutes ago

Laboratorio #2

Damos click en Add y seleccionamos la opción de Database

Red Hat OpenShift Container Platform

Project: demos Application: all applications

+Add

Topology

Monitoring

Search

Builds

Helm

Project

ConfigMaps

Secrets

Add

Select a way to create an Application, component or service from one of the options.

Getting started resources

- Create applications using samples
- Build with guided documentation
- Explore new developer features

Developer Catalog

- All services
- Database

Git Repository

- From Git
- From Devfile

Container images

From Local Machine

- Import YAM
- Upload JAR file

Samples

Se desplegará un catálogo de diversas bases de datos. Seleccionamos la opción de MySQL (Ephemeral)

Developer

+Add

Topology

Observe

Search

Builds

Helm

Project

ConfigMaps

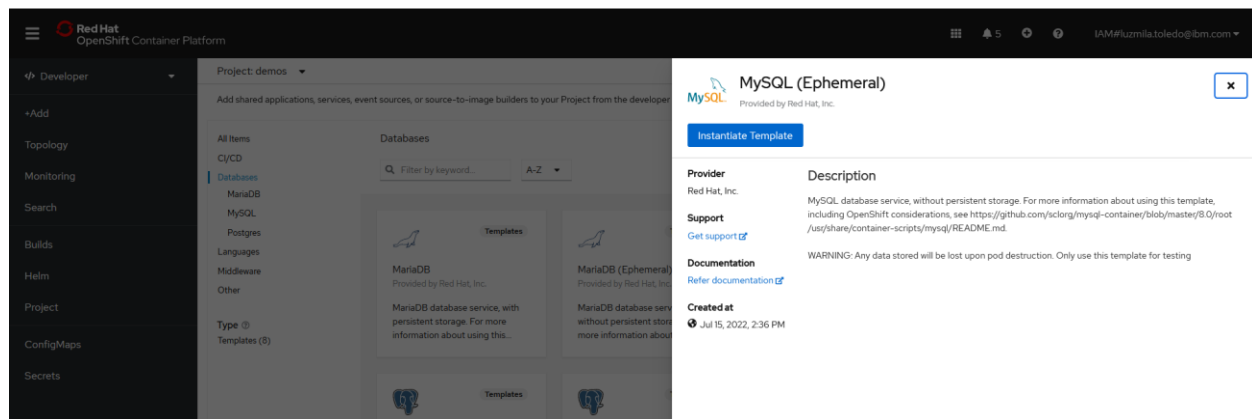
Secrets

Operator Backed (3)

Templates (18)

- Database Role
- MariaDB
- MariaDB (Ephemeral)
- MongoDB
- MongoDB (Ephemeral)
- MongoDB Atlas Cloud Database Service
- MySQL
- MySQL (Ephemeral)

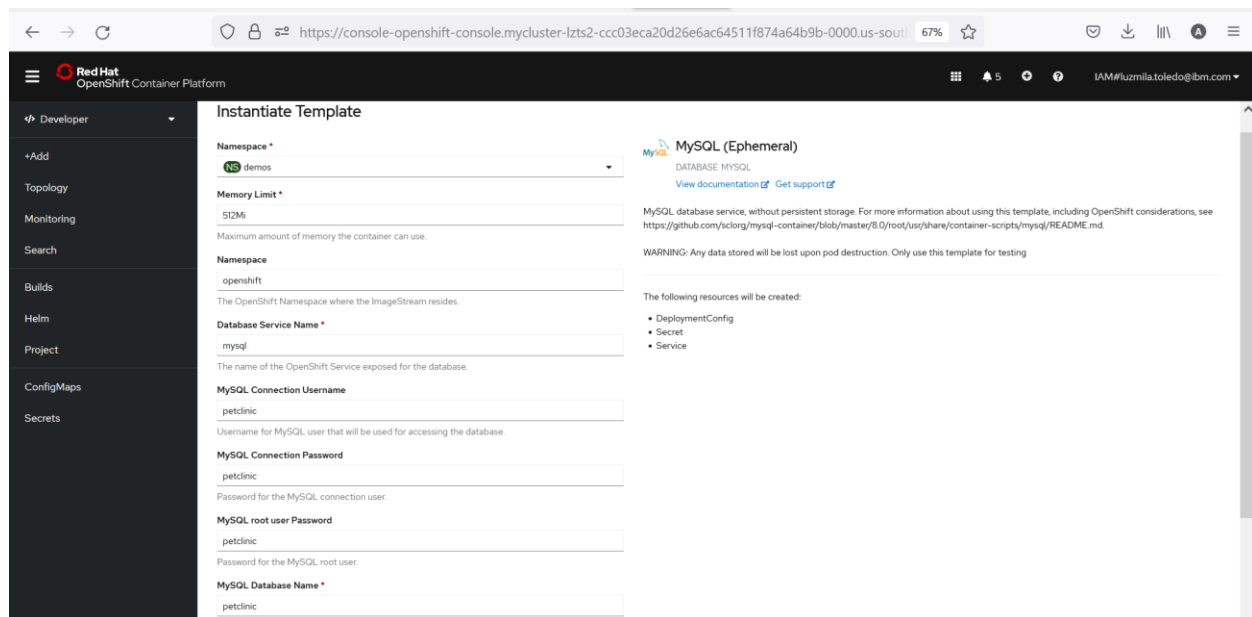
Damos click en el botón de Instantiate Template



Ingresamos los siguientes valores:

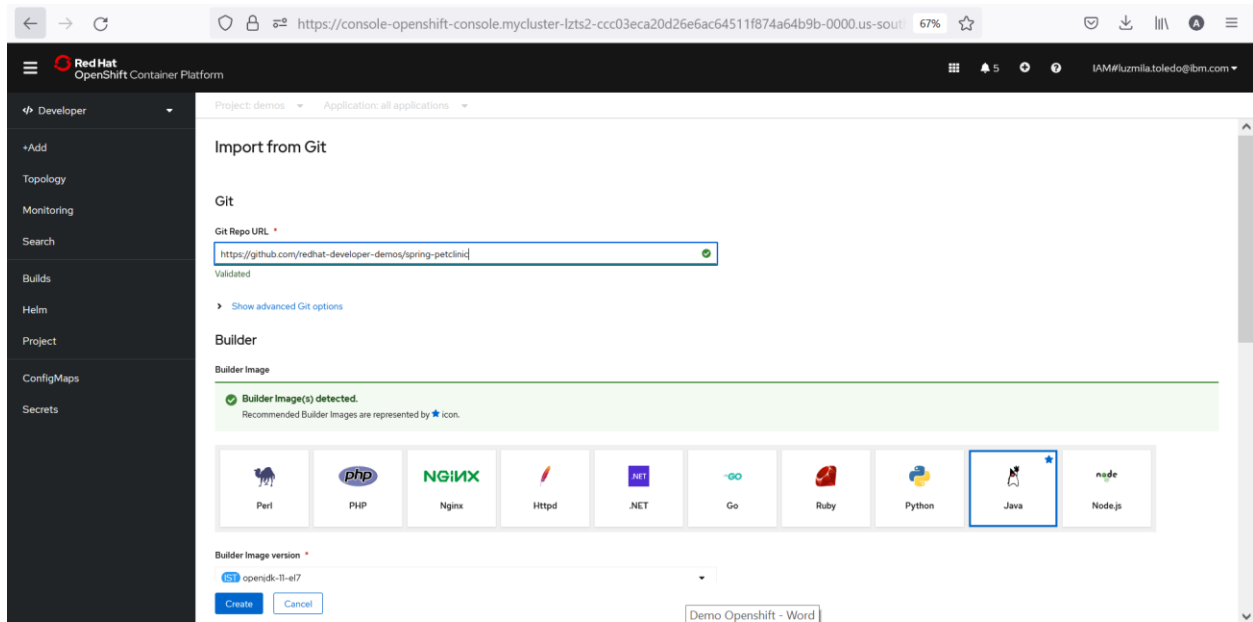
- MySQL Connection Username: **petclinic**
- MySQL Connection Password: **petclinic**
- MySQL root user Password: **petclinic**
- MySQL Database Name: **petclinic**

Damos click en Create



Después de unos minutos, y en segundo plano, se iniciará una instancia efímera de un contenedor de base de datos MySQL. En este punto, tiene un motor de base de datos para ser utilizado por la aplicación. Es hora de seguir adelante y crear la aplicación.

Damos click en +Add y seleccionamos Git Repository > From Git e ingresamos:
<https://github.com/redhat-developer-demos/spring-petclinic>



En la parte inferior seleccionamos Build Configuration

Resources

Select the resource type to generate

- ☒ Deployment
 - apps/Deployment
 - A Deployment enables declarative updates for Pods and ReplicaSets.
- ☐ DeploymentConfig
 - apps.openshift.io/DeploymentConfig
 - A DeploymentConfig defines the template for a Pod and manages deploying new Images or configuration changes.

Advanced options

- ☒ Create a route to the Application
 - Exposes your Application at a public URL

[Show advanced Routing options](#)

Click on the names to access advanced options for [Health check](#), [Build configuration](#), [Deployment](#), [Scaling](#), [Resource limits](#) and [Labels](#).

[Create](#) [Cancel](#)

Añadimos las siguientes variables y luego damos click en Create:

`SPRING_PROFILES_ACTIVE=mysql`
`MYSQL_URL=jdbc:mysql://mysql:3306/petclinic`

Build configuration

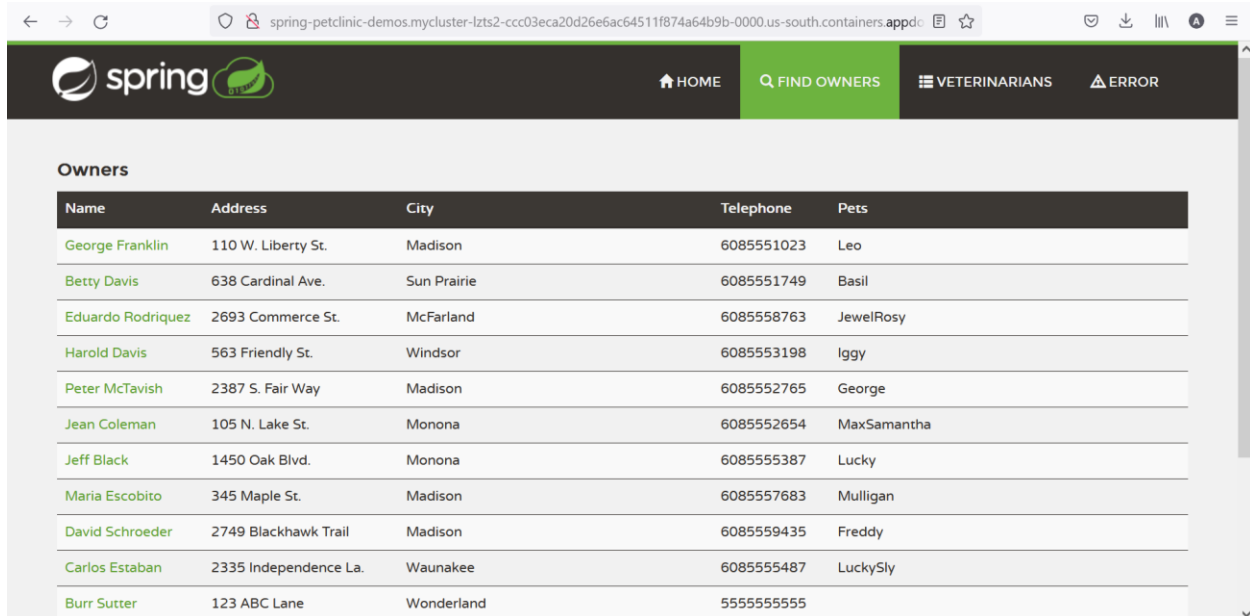
- ☒ Configure a webhook build trigger
- ☒ Automatically build a new Image when the Builder Image changes
- ☒ Launch the first build when the build configuration is created

Environment variables (build and runtime)

Name	Value
SPRING_PROFILES_ACTIVE	mysql
MYSQL_URL	jdbc:mysql://mysql:3306/petclinic

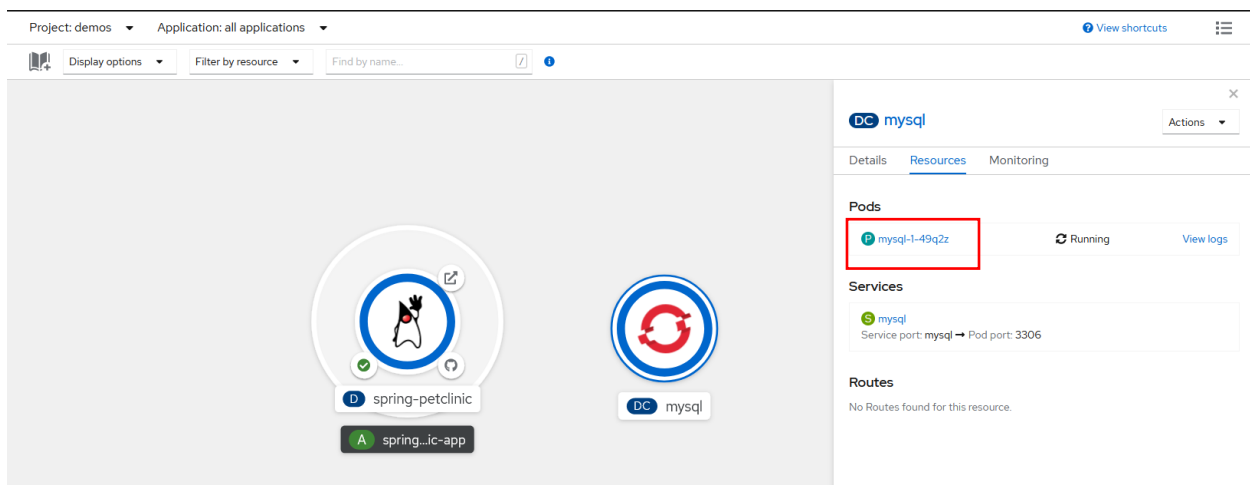
[+ Add value](#) [+ Add from ConfigMap or Secret](#)

Luego revisamos si la creación fue exitosa. En la pestaña de Topology revisamos que el despliegue haya finalizado (círculo en azul) y damos click en Open URL y observamos la aplicación de Petclinic



Name	Address	City	Telephone	Pets
George Franklin	110 W. Liberty St.	Madison	6085551023	Leo
Betty Davis	638 Cardinal Ave.	Sun Prairie	6085551749	Basil
Eduardo Rodriguez	2693 Commerce St.	McFarland	6085558763	JewelRosy
Harold Davis	563 Friendly St.	Windsor	6085553198	Iggy
Peter McTavish	2387 S. Fair Way	Madison	6085552765	George
Jean Coleman	105 N. Lake St.	Monona	6085552654	MaxSamantha
Jeff Black	1450 Oak Blvd.	Monona	608555387	Lucky
Maria Escobito	345 Maple St.	Madison	6085557683	Mulligan
David Schroeder	2749 Blackhawk Trail	Madison	6085559435	Freddy
Carlos Estaban	2335 Independence La.	Waunakee	6085555487	LuckySly
Burr Sutter	123 ABC Lane	Wonderland	5555555555	

Ahora, en la pestaña Topology de la consola de Openshift seleccionamos el Deployment de Mysql, vamos a resources y seleccionamos el pod en estado Running



Project: demos Application: all applications View shortcuts

Display options Filter by resource Find by name...

spring-petclinic spring...ic-app

mysql

mysql-1-49q2z Running View logs

mysql Service port: mysql → Pod port: 3306

No Routes found for this resource.

En la pestaña de terminal ingresamos los siguientes comandos

```
mysql -u root -h mysql -p
```

```
petclinic
```

```
use petclinic;  
show tables;
```

Red Hat

OpenShift Container Platform

5

IAM#luzmila.toledo@ibm.com

Developer

+Add

Topology

Monitoring

Search

Builds

Helm

Project

ConfigMaps

Secrets

Project demos

Pods > Pod details

mysql-1-49q2z Running

Actions

DetailsMetricsYAMLEnvironmentLogsEventsTerminal

Connecting to mysql

Expand

```
sh-4.4$ mysql -u root -h mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 242
Server version: 8.0.26 Source distribution

Copyright (c) 2000, 2021, 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> use petclinic;
Database changed
mysql> show tables;
+-----+
| Tables_in_petclinic |
+-----+
| owners              |
| pets                |
| specialties          |
| types               |
| vet_specialties      |
+-----+
```