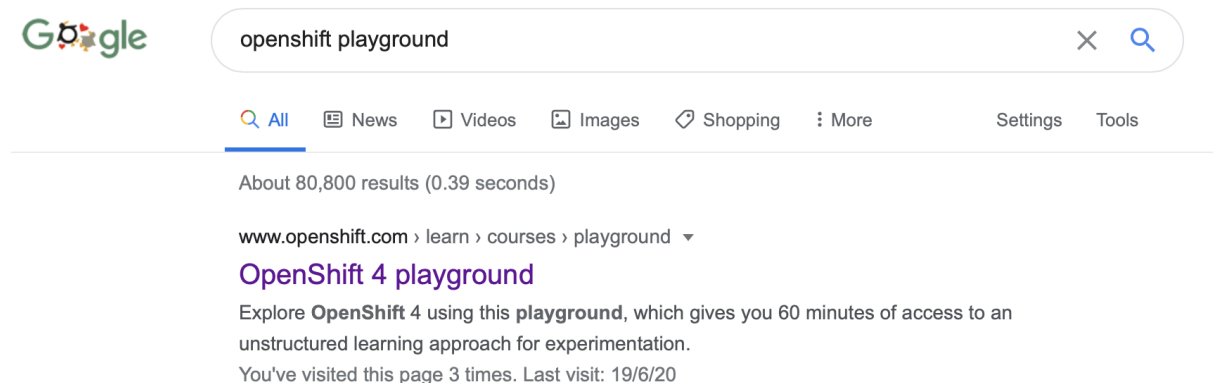


# Deployment using Templates:

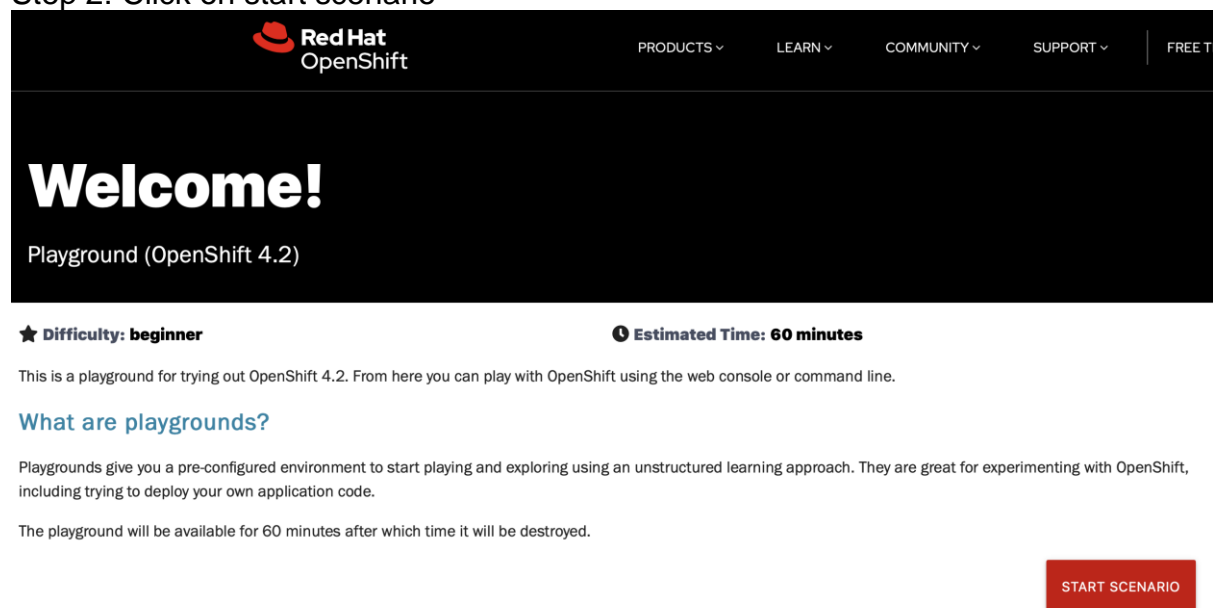
- A. Uploading template through terminal by logging into openshift client, Sampleproject.
- B. Using a template from web console

## A. Uploading template through terminal by logging into openshift client.

Step 1: Type “openshift playground” in google and click on Openshift 4 playground.



Step 2: Click on start scenario



Initially playground cluster is logged in as admin to work with applications we need to login as developer

Step 3: Type ***oc login -u developer -p developer*** in terminal to login as developer.

```
Terminal Console ↻ ↗ +
Your Interactive Learning Environment Bash Terminal

$ launch.sh
OpenShift Ready
OpenShift Ready
$ oc login -u developer -p developer
Login successful.

You don't have any projects. You can try to create a new project, by running

    oc new-project <projectname>

$
```

Step 4: Create a new project by typing ***oc new-project sampleproject***

```
$ oc new-project sampleproject
Now using project "sampleproject" on server "https://openshift:6443".

You can add applications to this project with the 'new-app' command. For example, try:

    oc new-app django-psql-example

to build a new example application in Python. Or use kubectl to deploy a simple Kubernetes application:

    kubectl create deployment hello-node --image=gcr.io/hello-minikube-zero-install/hello-node
```

Step 5: Create an app in the project by using command ***“oc new-app Django-psql-example”***

```
$ oc project sampleproject
Already on project "sampleproject" on server "https://openshift:6443".
$ oc new-app django-psql-example
--> Deploying template "openshift/django-psql-example" to project sampleproject

Django + PostgreSQL (Ephemeral)
-----
An example Django application with a PostgreSQL database. For more information about using this template, including OpenShift considerations, see https://github.com/sclorg/django-ex/blob/master/README.md.

WARNING: Any data stored will be lost upon pod destruction. Only use this template for testing.
```

Step 6: You can access the application via route ***'django-psql-example-sampleproject.2886795275-80-host04nc.environments.katacoda.com'***

Not Secure — django-psql-example-sampleproject.2886795275-80-host04nc.environments.katacoda.com

mycluster-yuk-dal10-b3c.4x16 Cluster - IBM... IBM Webex Bringing OpenShift to IBM Cloud | Chris Rose... Open

Welcome to your Django application on OpenShift

### How to use this example application

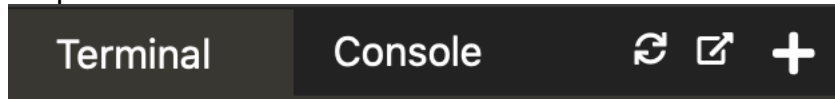
For instructions on how to use this application with OpenShift, start by reading the [Developer Guide](#).

### Managing your application

Documentation on how to manage your application from Line is available at the [Developer Guide](#).

## B. Using a template from web console

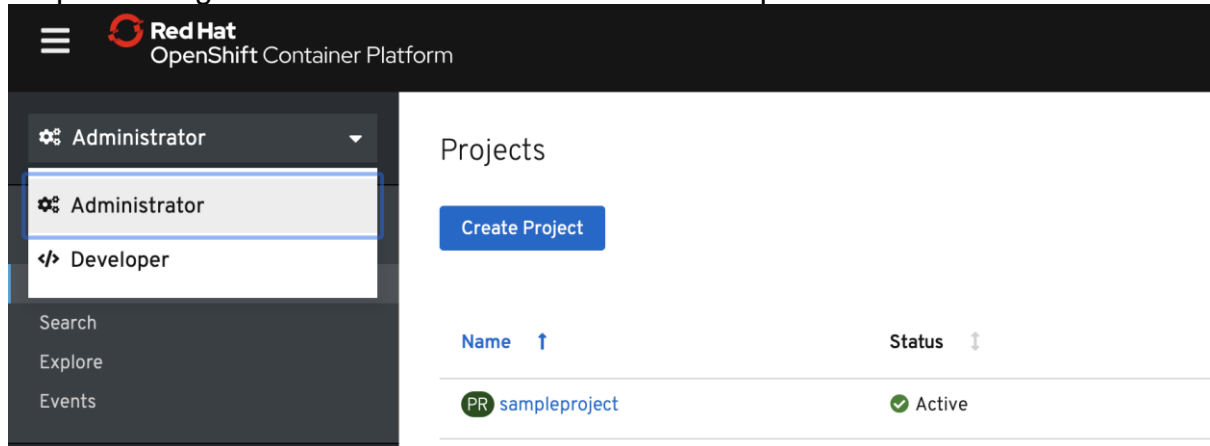
Step 1. Click on web console



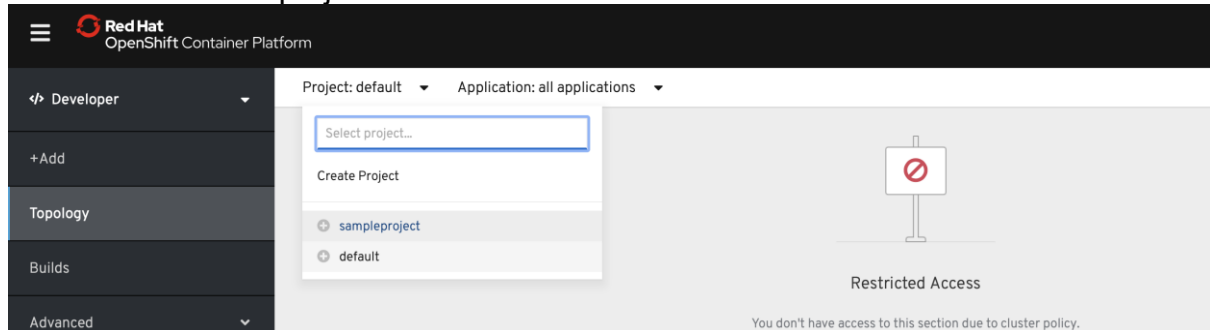
Step 2: Login into web console as developer



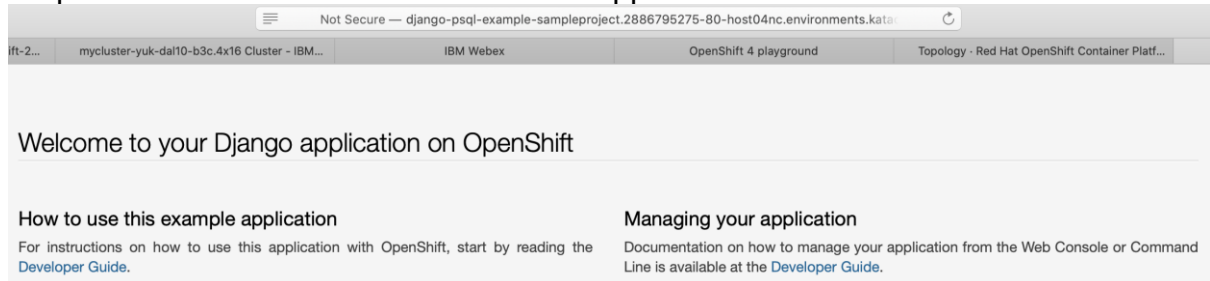
Step 3: Change the user from administrator to developer



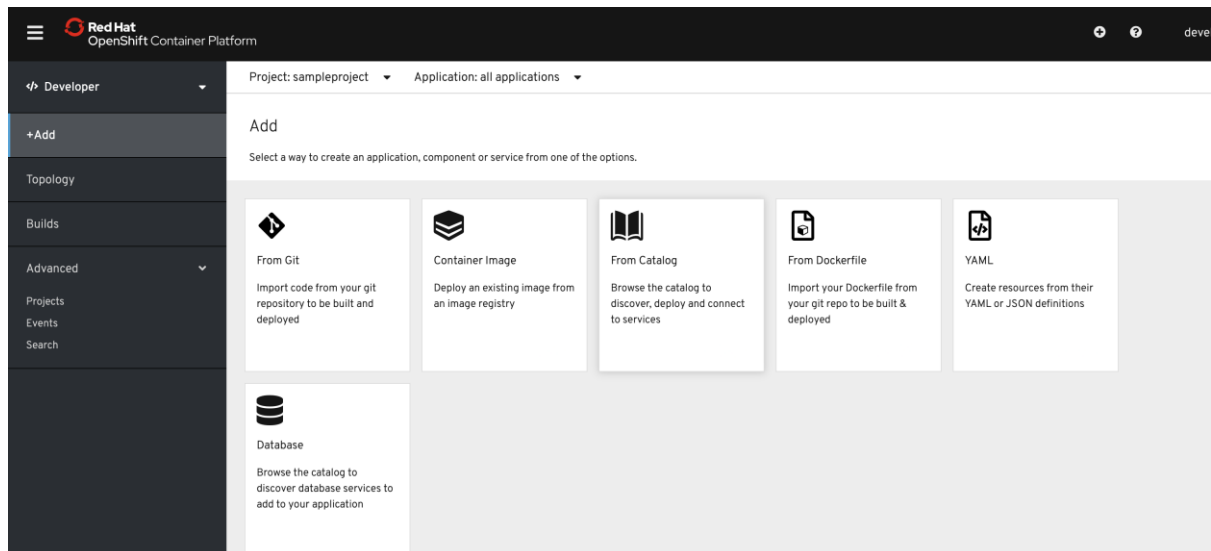
Step 4: Change to our “sampleproject” from “default” as we do not have permission to access “default” project



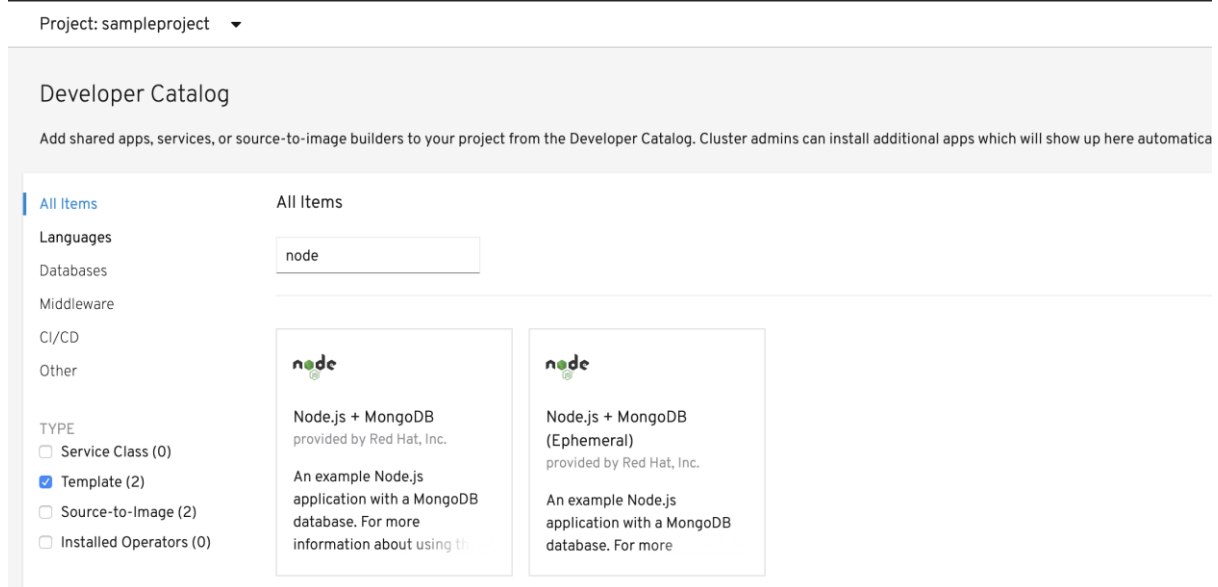
Step 5: Click on arrow icon to access the application



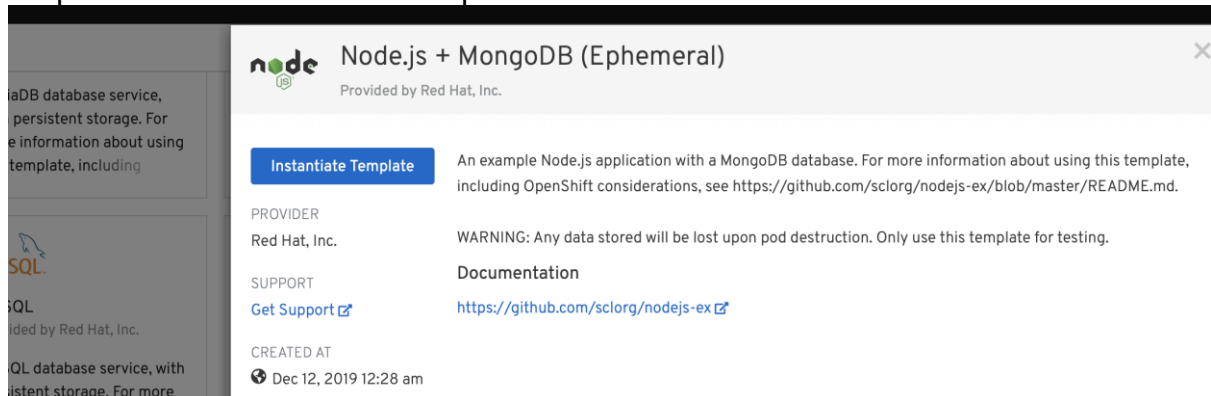
Step 6: Click on Add to add another template to **sampleproject** and choose **From catalog**



## Step 7: create Node.js + MongoDB (Ephemeral)



## Step 8: Click on Instantiate Template



## Step 9: Give MongoDB Username, Password and DBA Password as '**root**' and click on create

Advanced

Database Service Name \*

mongodb

MongoDB Username

root

Username for MongoDB user that will be used for accessing the database.

MongoDB Password

root

Password for the MongoDB user.

Database Name \*

sampledb

Database Administrator Password

root

Password for the database admin user.

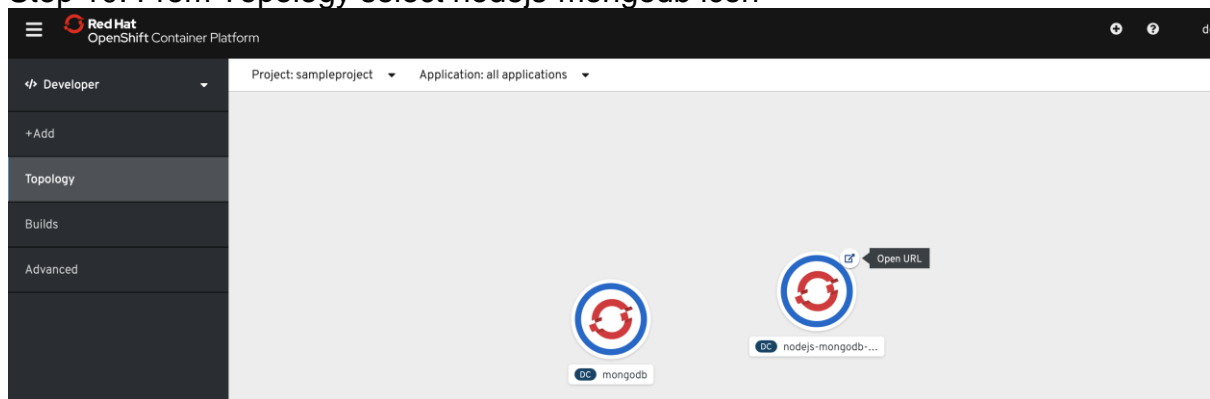
Custom NPM Mirror URL

The custom NPM mirror URL

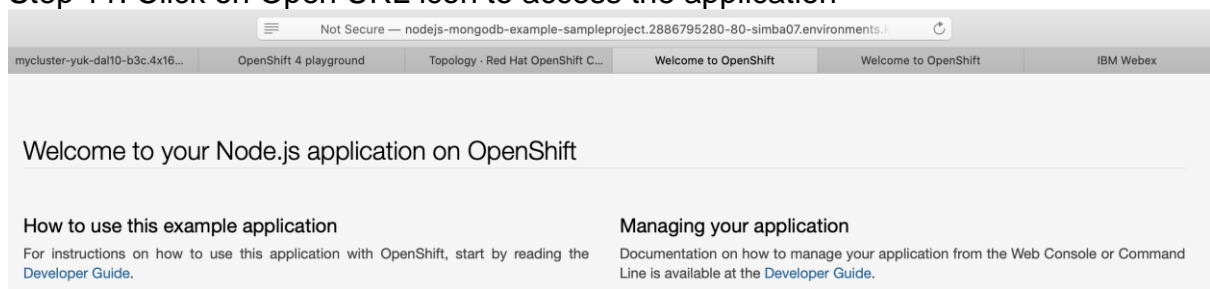
Create

Cancel

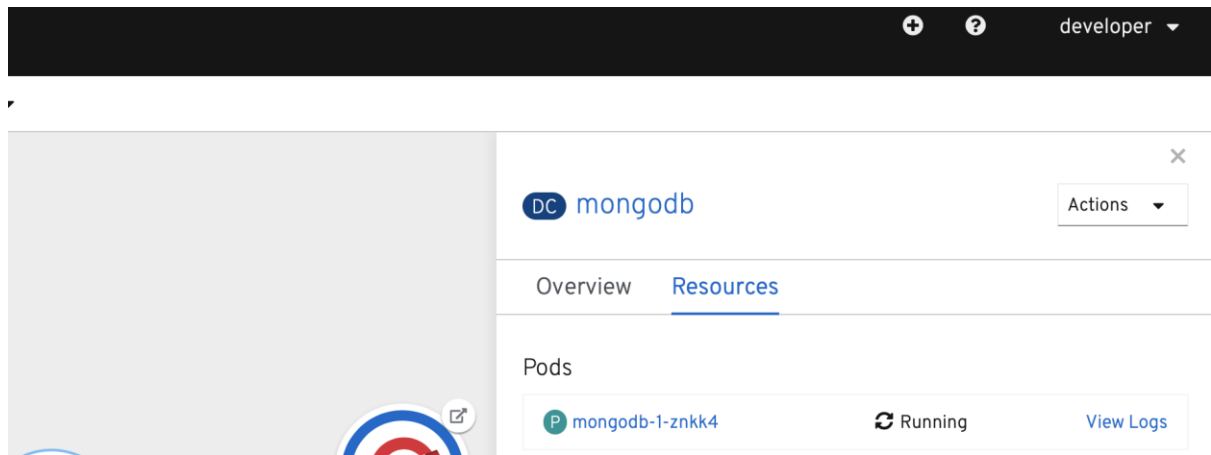
Step 10: From Topology select nodejs-mongodb icon



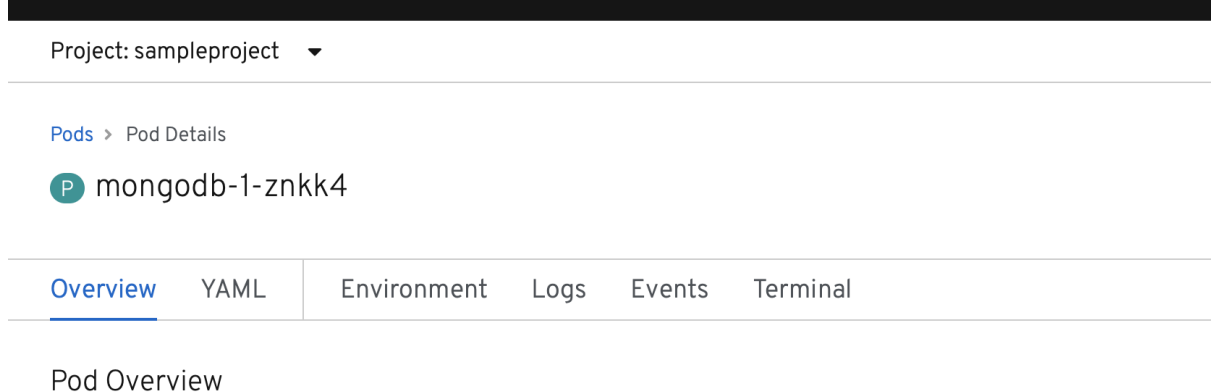
Step 11: Click on Open URL icon to access the application



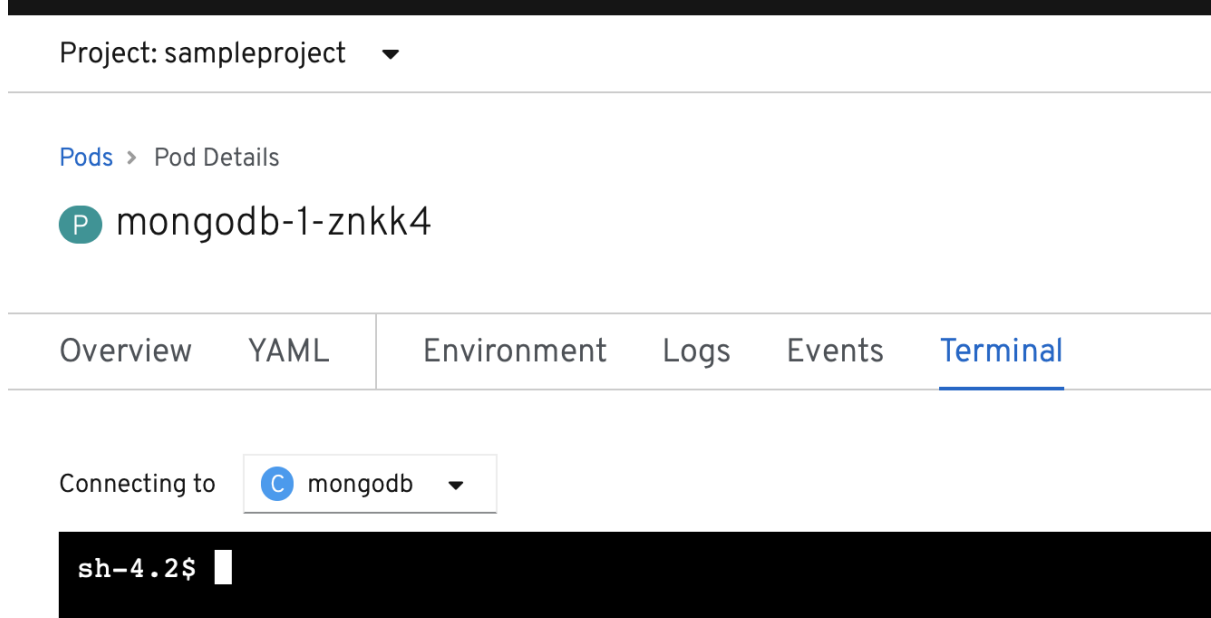
Step 12: Now from Topology select mongodb to access MongoDB database



Step 13: Click on mongodb pod



Step 14: Click on Terminal to access mongodb database




Step 15: Login into mongodb using below command  
Mongo -u root -p root sampled

Topology

Builds

Advanced

OverviewYAMLEnvironmentLogsEventsTerminal

Connecting to  mongodb

```
sh-4.2$ mongo -u root -p root sampledb
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017/sampledb
MongoDB server version: 3.6.3
> 
```

## Step 16: Access counts db

```
> show collections
counts
> db.counts.find()
{ "_id" : ObjectId("5eef0a9d1b35300019267a1d"), "ip" : "10.128.0.1", "date" : 1592724125009 }
{ "_id" : ObjectId("5eef0aa61b35300019267a1e"), "ip" : "10.128.0.1", "date" : 1592724134994 }
{ "_id" : ObjectId("5eef0ab01b35300019267a1f"), "ip" : "10.128.0.1", "date" : 1592724144895 }
```

<https://api.crc.testing:6443/apis/build.openshift.io/v1/namespaces/sampleproject/buildconfigs/nodejs-mongodb-example/webhooks/hqB0PHwmiHxIQvpA6eWnh83qFw2qOXjrOgdwUBUX/github>

hqB0PHwmiHxIQvpA6eWnh83qFw2qOXjrOgdwUBUX