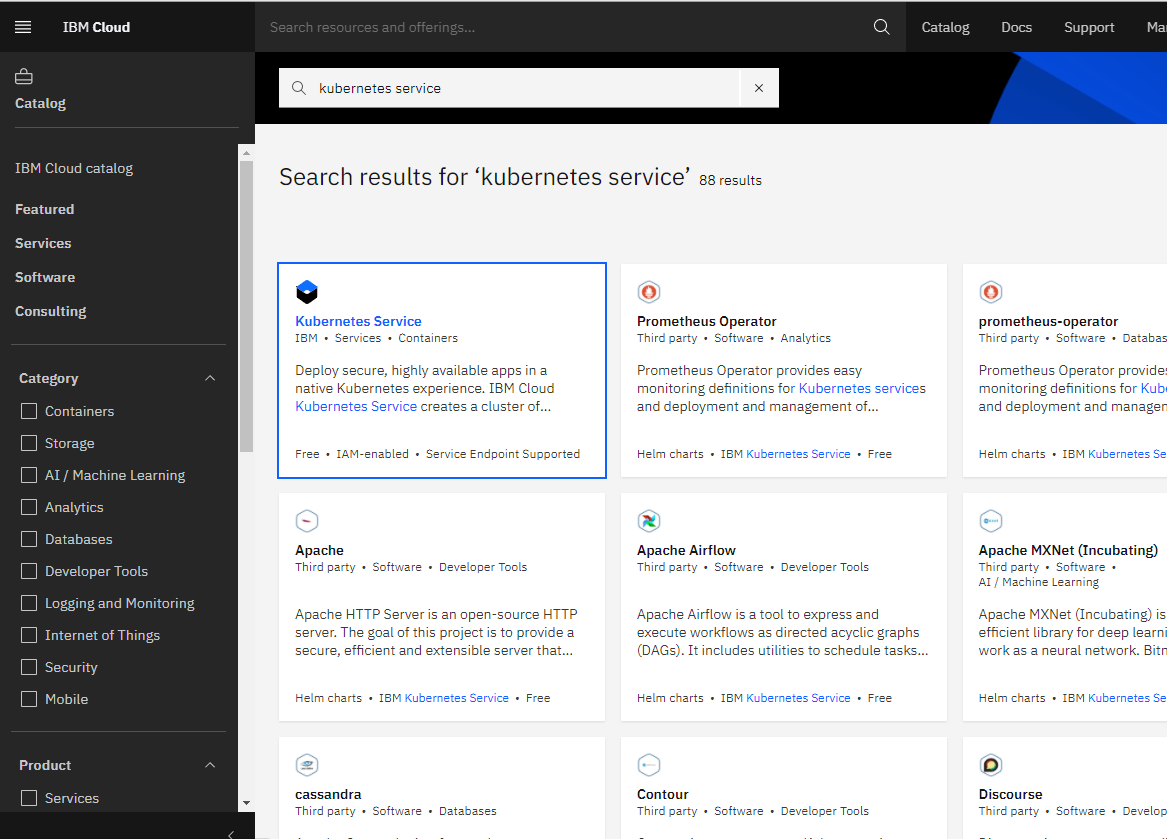
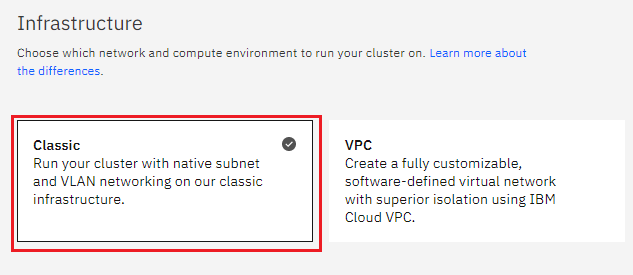
# Installing MongoDB on IBM Cloud

**Step 1 provision Kubernetes Cluster**

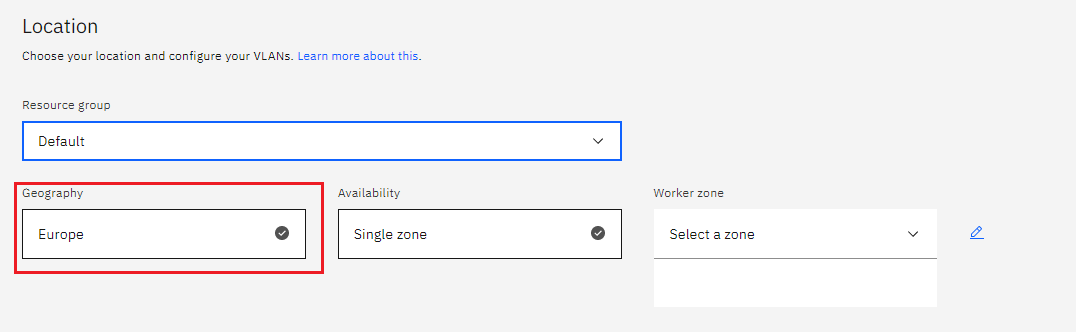
* Click the **Catalog** button on the top
* Select **Service** from the **Catalog**
* Search for **Kubernetes Service** and click on it



* You are now at the Kubernetes deployment page. You need to specify some details about the cluster
* Choose a plan **standard** or **free**. The free plan only has one worker node and no subnet, to provision a standard cluster, you will need to upgrade your account to Pay-As-You-Go
* To upgrade to a Pay-As-You-Go account, complete the following steps:
  + In the console, go to Manage > Account.
  + Select Account settings; and click Add credit card.
  + Enter your payment information, click Next, and submit your information
  + Choose **classic** or **VPC**, read the docs and choose the most suitable type for yourself



* Now choose your location settings,
* Choose **Geography** (continent)



* + Choose Single or Multizone, in single zone your data is only kept on datacentre, on the

other hand with Multizone it is distributed to multiple zones, thus safer in an unforeseen

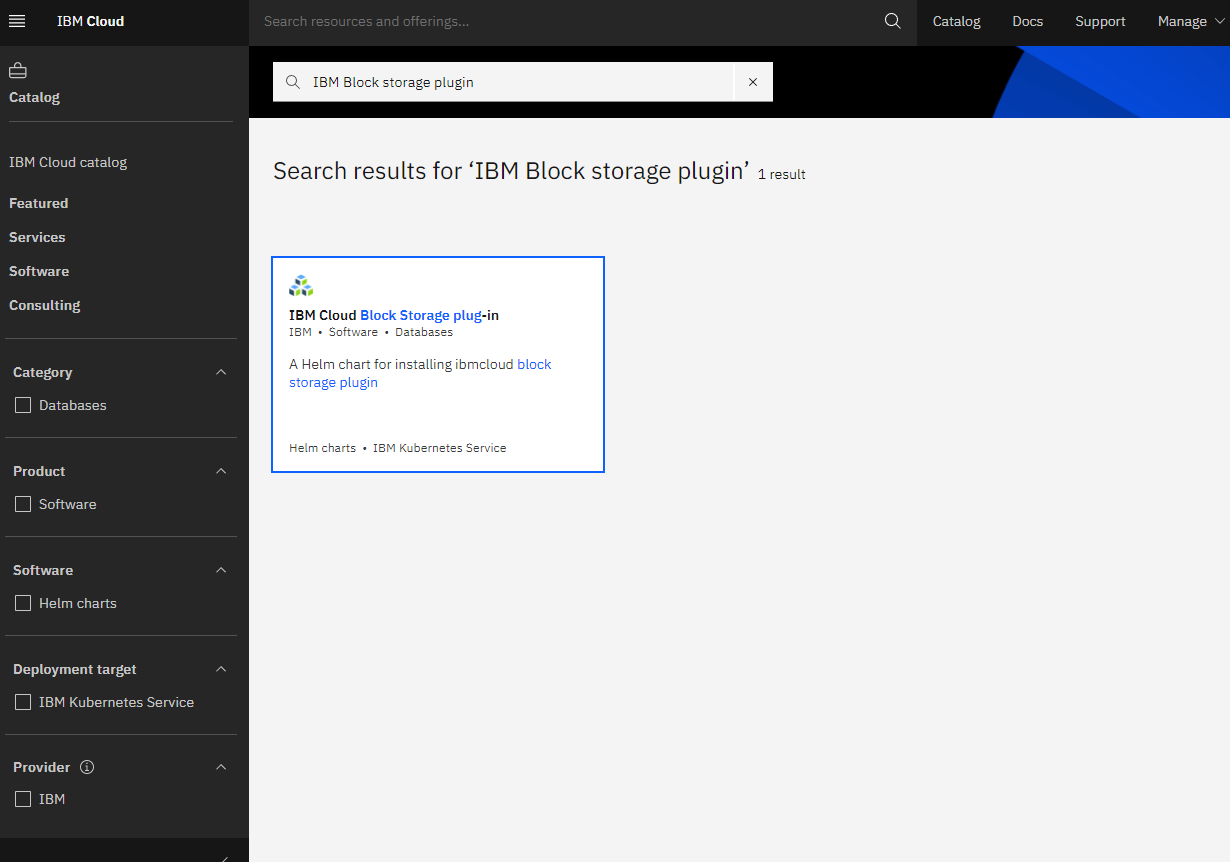
zone failure.

* If you wish to use Multizone, please set up your account with VRF.
* If at your current location selection, there is no available Virtual LAN, a new VLAN will be created for you.
* Choose a Worker node setup or use the preselected one, set Worker node amount per zone
* Choose **Master Service Endpoint** , In VRF-enabled accounts, you can choose private-only to make your master accessible on the private network or via VPN tunnel. Choose public-only to make your master publicly accessible. When you have a VRF-enabled account, your cluster is set up by default to use both private and public endpoints.  
  Give desired **tags** to your cluster, for more information visit tags
* Click **create**  
  • Wait for your cluster to be provisioned  
  • Your cluster is ready for usage

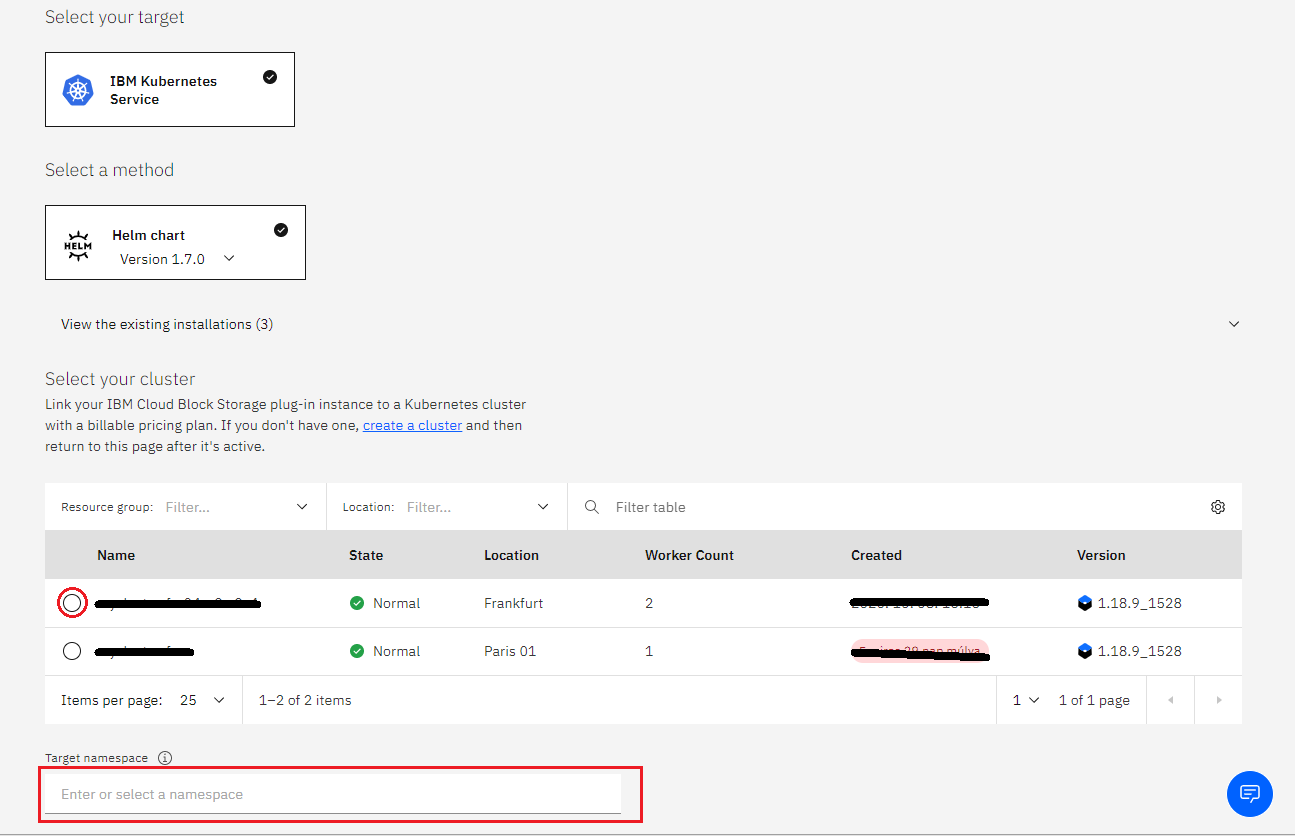
**Step 2 Deploy IBM Cloud Block Storage plug-in**

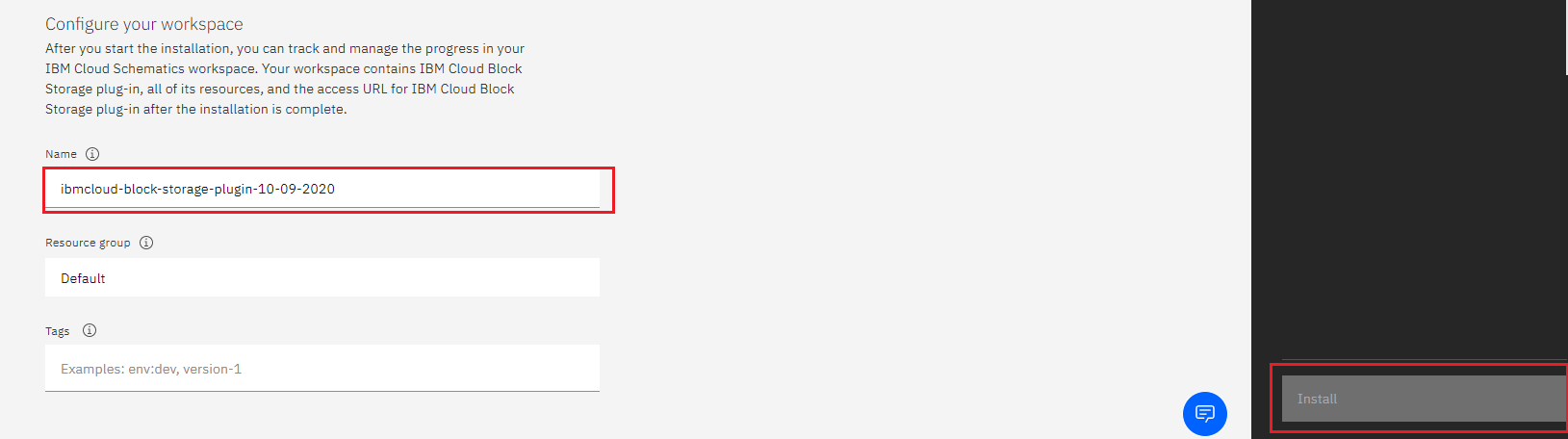
The Block Storage plug-in is a persistent, high-performance iSCSI storage that you can add to your apps by using Kubernetes Persistent Volumes (PVs).

* Click the **Catalog** button on the top
* Select **Software** from the catalog
* Search for **IBM Cloud Block Storage plug-in** and click on it



* On the application page, click on the dot (.) next to the cluster, you wish to use
* Click on Enter or Select Namespace and choose the default Namespace or use a custom  
  one (if you get error please wait for up to 30 minutes for the cluster to finalize)



* Give a **name** to this workspace
* Click **install** and wait for the deployment 

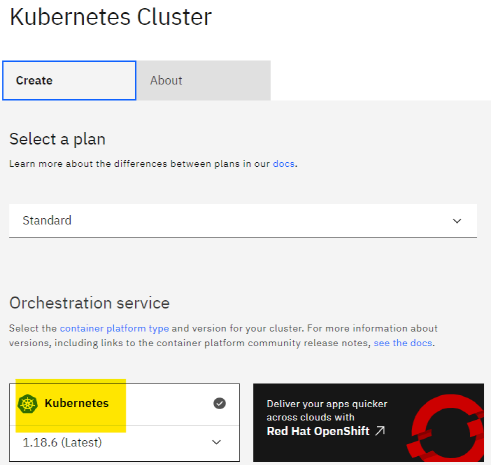
**Step 3 For MongoDB**

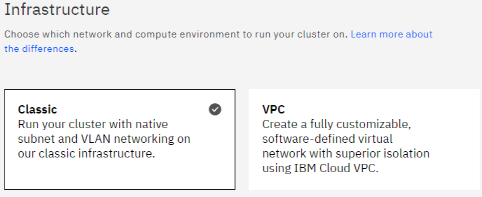
### **Creating a database**

After installing MongoDB, you’ll need to create a directory where your data will be stored. This can be done locally or through public or private cloud storage solutions.

**Steps:**

1. Before starting, you need to have IKS clusters on IBM cloud. If you don’t have the cluster, provision it first at IBM cloud portal h[ttps://cloud.ibm.com/kubernetes/overview](https://cloud.ibm.com/kubernetes/overview) .
2. IKS support 2 types of infrastructure, Classic and VPC. Classic is general one but IBM Cloud Virtual Servers for VPC offer agile scalability for easy business growth, high availability for better customer satisfaction, security for your workloads and lower total cost of ownership. It provides high speed network interface and large amount of cores.





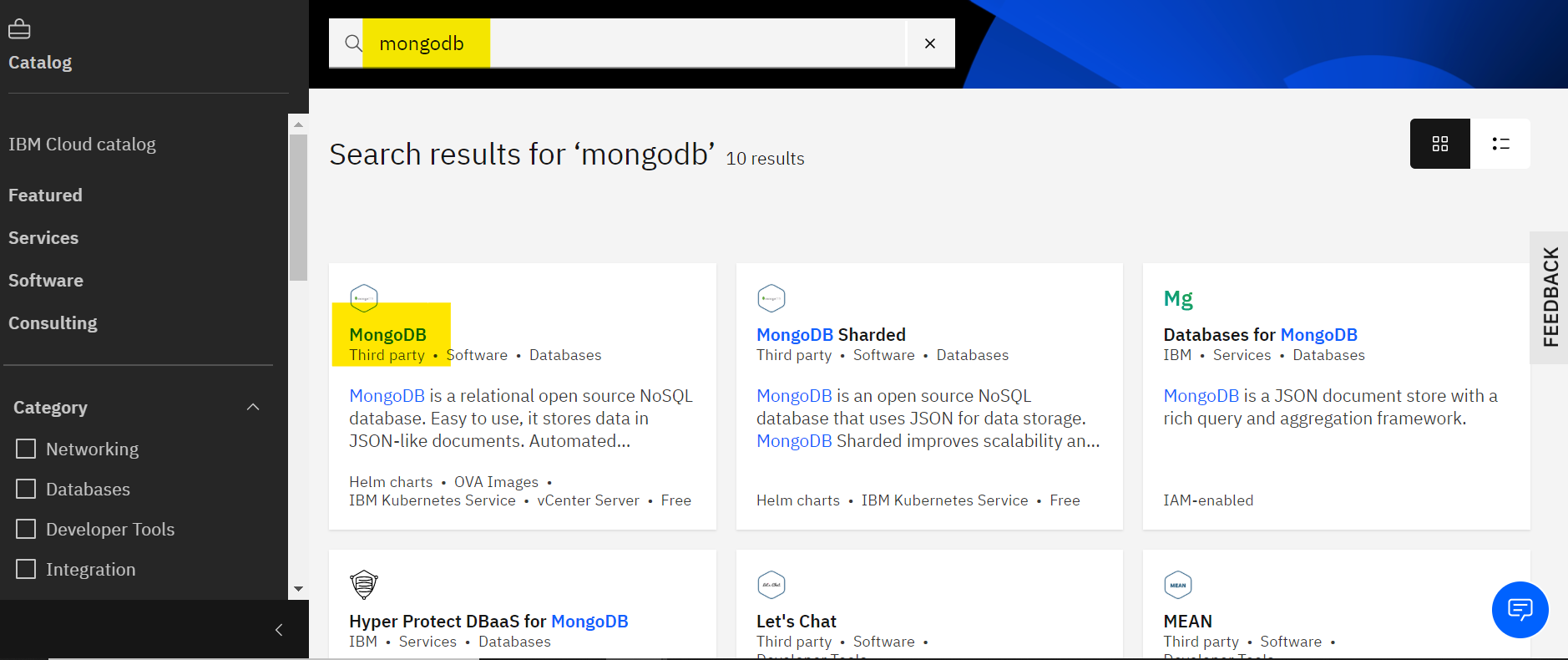
1. Create namespace for mongodb like “mongodb”, but you can use existing one. The easiest way to create namespace is using Web Terminal on cluster overview screen. The shell has an existing authority, so you don’t need to login with the user and password.

$ kubectl get ns  
NAME STATUS AGE  
default Active 21h  
ibm-cert-store Active 21h  
ibm-observe Active 20h  
ibm-operators Active 21h  
ibm-system Active 21h  
kube-node-lease Active 21h  
kube-public Active 21h  
kube-system Active 21h  
$ kubectl create ns mongodb   
namespace/mongodb created

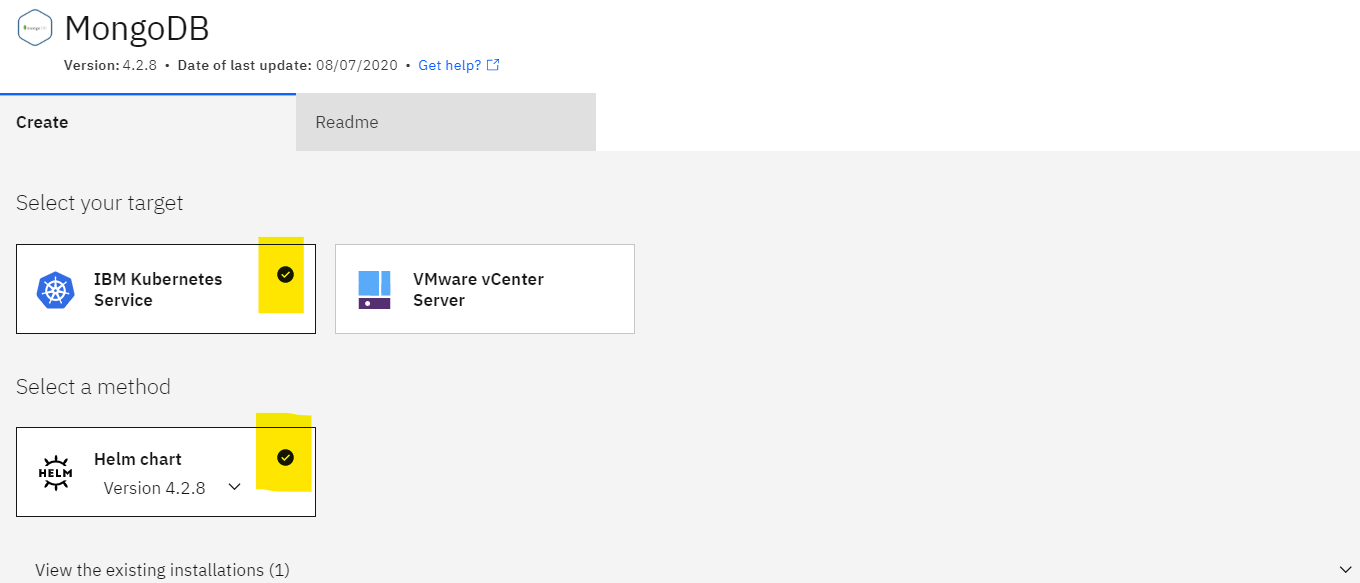
3. Install “IBM Cloud Block storage plug-in” from the catalog. It supports dynamic creation of PV with “ibm-block-storage-driver”.

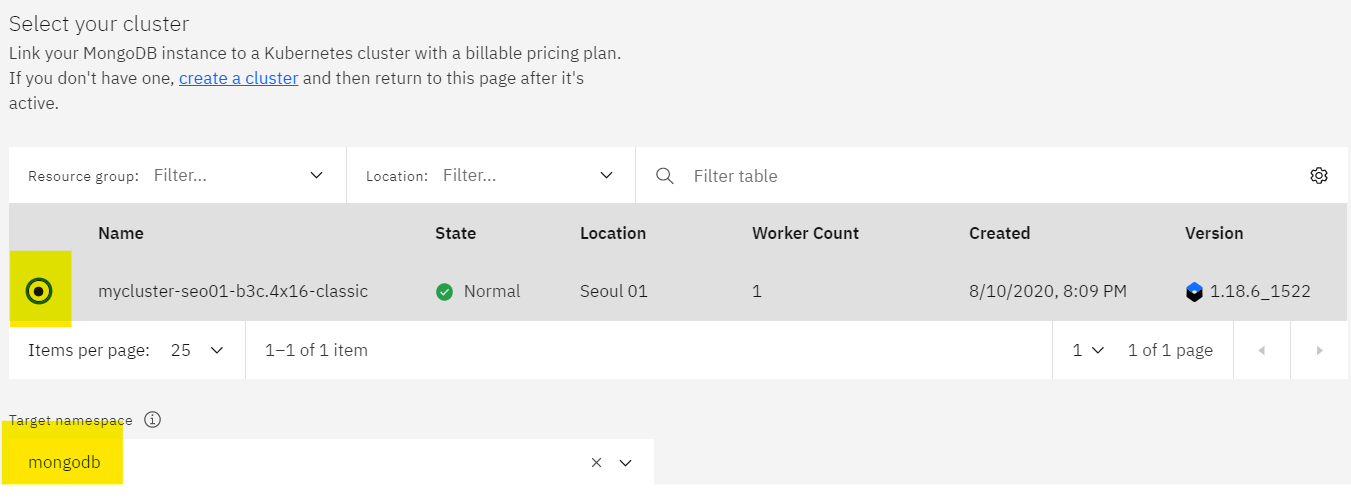
Provisioning MongoDB

From the catalog of IBM cloud portal, you can see 3 kinds of MongoDB. You will work with MongoDB, which is first one.



Click the tile “MongoDB” and then you go to the Helm chart configuration screen. Select the target IKS, not VMware vCenter server.

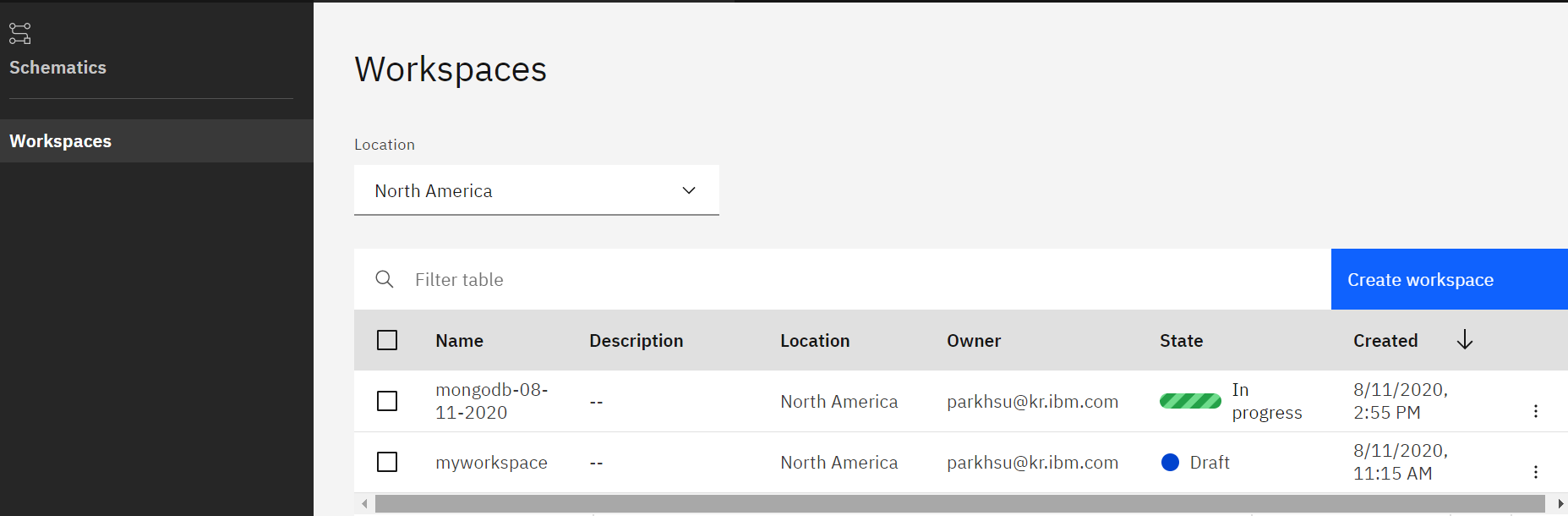
And then select your cluster and namespace, which are created previously.

There are several values, you need to update at “Set the deployment values” as below. It’s just example, you can change it with yours.

auth.username : admin  
auth.password : admxx  
auth.rootPassword : passxx  
auth.database : sample

=

When all configurations are setup and then Click “Install” button at right-bottom of the screen. You can check the installation progress and logs from the “Schematics” workspace for MongoDB.



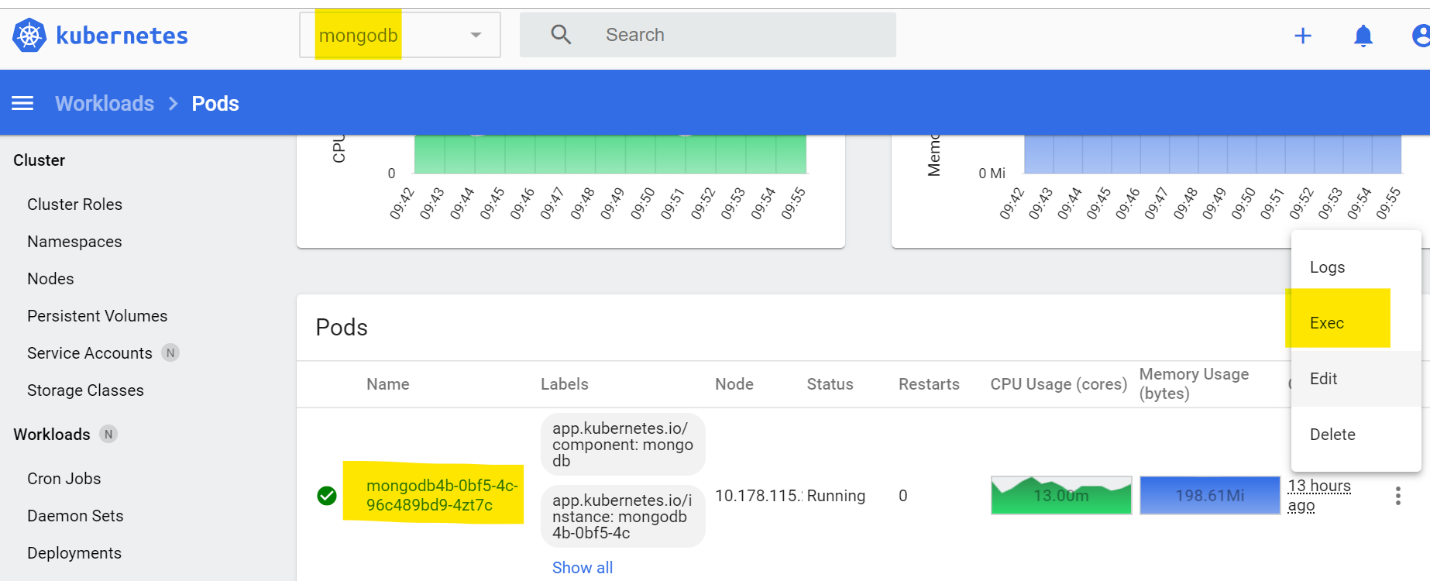
How to Access MongoDB for manipulating

**Option 1) Web terminal**

Previously, creating namespace was explained. When you log in to Web terminal, you have to go to the container, MongoDB

$ kubectl get pods -n mongodb  
NAME READY STATUS RESTARTS AGE  
mongodb4b-0bf5–4c-96c489bd9–4zt7c 1/1 Running 0 13h$ kubectl -n mongodb exec -it mongodb4b-0bf5–4c-96c489bd9–4zt7c bash  
mongodb4b-0bf5–4c-96c489bd9:/$ echo "you are in container shell"

**Option 2) Kubernetes portal**



**Playing with MongoDB inside container**

1. check the parameters, which are defined when you install the MongoDB

$ env | grep MONGO  
MONGODB4B\_0BF5\_4C\_SERVICE\_PORT=27017  
MONGODB\_DISABLE\_SYSTEM\_LOG=no  
MONGODB\_ENABLE\_IPV6=no  
MONGODB\_PASSWORD=admin  
MONGODB4B\_0BF5\_4C\_PORT=tcp://172.21.xx:27017  
MONGODB4B\_0BF5\_4C\_PORT\_27017\_TCP=tcp://172.21.xx:27017  
MONGODB4B\_0BF5\_4C\_SERVICE\_PORT\_MONGODB=27017  
MONGODB4B\_0BF5\_4C\_PORT\_27017\_TCP\_PROTO=tcp  
MONGODB\_DATABASE=sampledb  
MONGODB\_**ROOT**\_PASSWORD=passxx  
MONGODB\_ENABLE\_DIRECTORY\_PER\_DB=no  
MONGODB\_SYSTEM\_LOG\_VERBOSITY=0  
MONGODB4B\_0BF5\_4C\_PORT\_27017\_TCP\_ADDR=172.21.xx  
MONGODB\_USERNAME=admin  
MONGODB4B\_0BF5\_4C\_PORT\_27017\_TCP\_PORT=27017  
MONGODB4B\_0BF5\_4C\_SERVICE\_HOST=172.21.xx

2. running MongoDB cli

$ mongo admin -u root -p passxx  
$ mongo sampledb -u admin -p admin

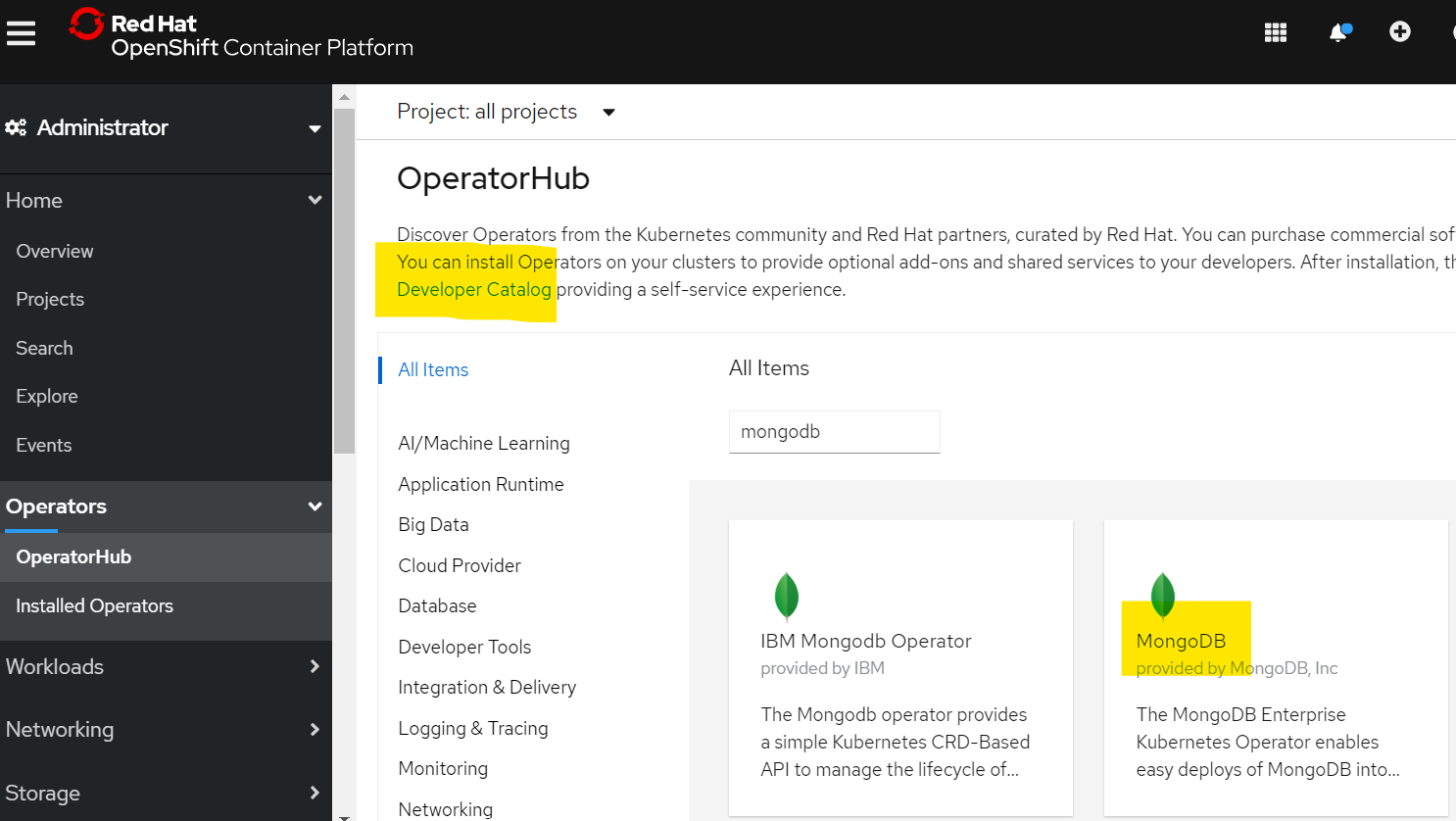
3. sample SQL command

> show dbs   
> use admin  
> show tables> db.mytable.insertOne({ item: "canvas", qty: 100, tags: ["cotton"], size: { h: 28, w: 35.5, uom: "cm" } })  
> db.mytable.find()  
> db.mytable.count()> use admin  
> db.system.users.find({"user":"admin"})> use mytable  
> db.createUser(  
{user: "user02",  
pwd: passwordPrompt(), // or cleartext password  
roles: [  
{ role: "read", db: "admin" },  
{ role: "readWrite", db: "sampledb" }  
]})

PART 2. OpenShift

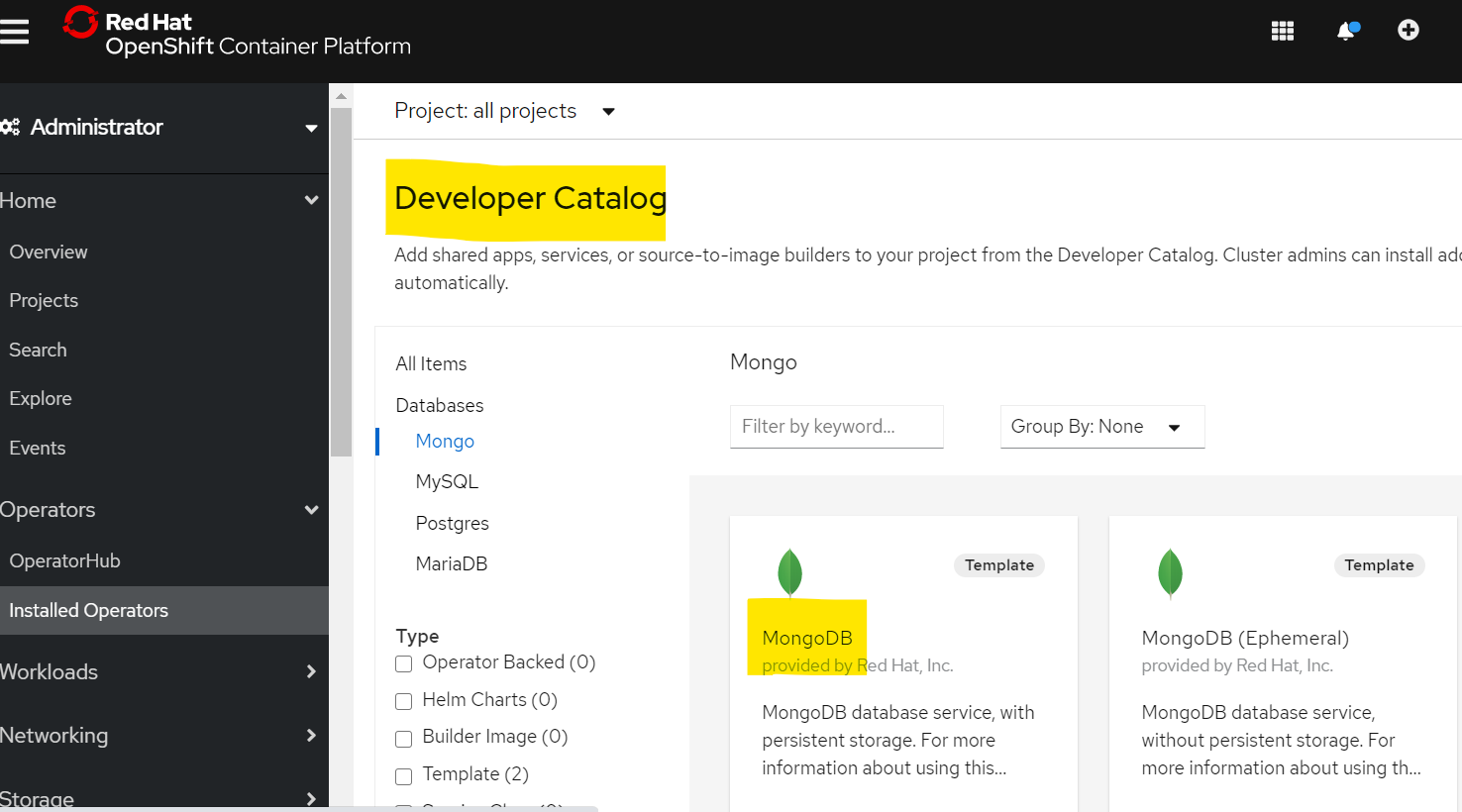
At IKS case, we install it from the catalog of IBM Cloud, but for OpenShift we need can use 2 options, OperatorHub and Template from OpenShift web portal. Two options support different docker images of MongoDB. In my test, I choose the Template, which is easier than OperatorHub’s. Because it requires more prerequisite.

1. OperatorHub



2. Template

Developer Catalog at above screen capture is Template catalog support as below.

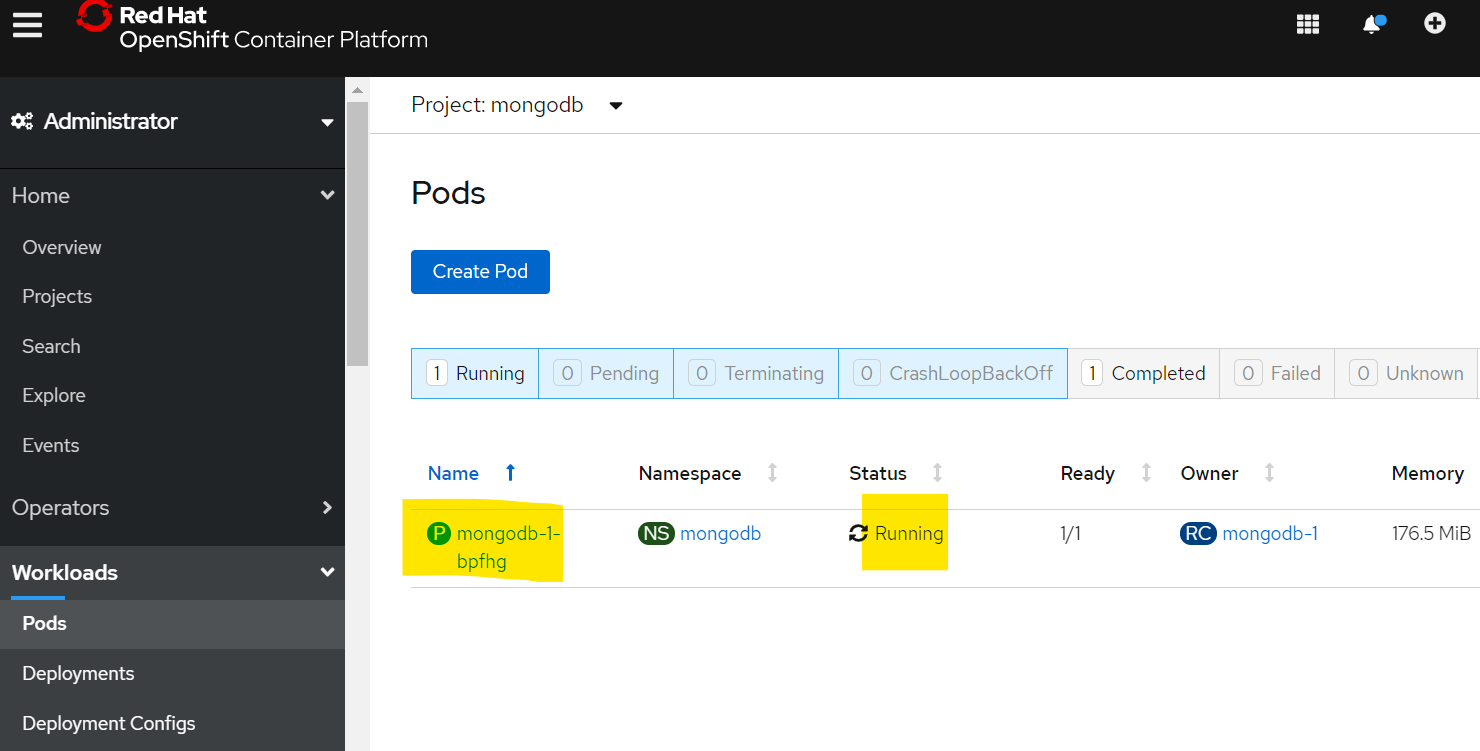


There are several values, you need to update at “Set the deployment values” as below. It’s just example, you can change it with yours.

MongoDB connection username : admin  
MongoDB connection password : admxx  
MongoDB database name : sampledb  
MongoDB admin password : passxx

**Playing with MongoDB inside container**

After install the Template, You need to check the POD and go into the terminal to manipulate MongoDB data.



Database operation is same with IKS case above, but different one is root user id, which is admin.

sh-4.2$ env | grep MONGODB   
MONGODB\_PORT\_27017\_TCP=tcp://172.21.xx:27017  
MONGODB\_PORT=tcp://172.21.xx:27017  
MONGODB\_PORT\_27017\_TCP\_PORT=27017  
MONGODB\_**ADMIN**\_PASSWORD=passxx  
MONGODB\_SERVICE\_HOST=172.21.96.83  
MONGODB\_DATABASE=sampledb  
MONGODB\_PORT\_27017\_TCP\_PROTO=tcp  
MONGODB\_PASSWORD=admxx  
MONGODB\_VERSION=3.6  
MONGODB\_PORT\_27017\_TCP\_ADDR=172.21.xx  
MONGODB\_SERVICE\_PORT=27017  
MONGODB\_USER=admin  
MONGODB\_SERVICE\_PORT\_MONGO=27017sh-4.2$ mongo admin -u admin -p passxx  
sh-4.2$ mongo sampledb -u admin -p admi