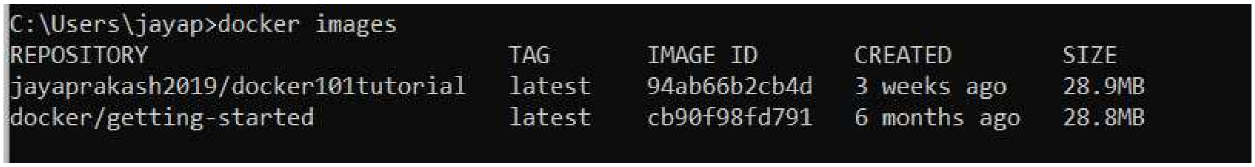
ASSIGNMENT -4

SABARINATHAN B
RAGUL V
PAULSAN S
DHARANIKUMAR K

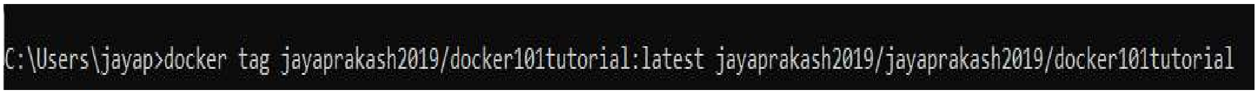
## 1) PULL AN IMAGE FROM DOCKER HUB AND RUN IT IN DOCKER PLAYGROUND.

### STEP-1 : List out the< docker images>



```
C:\Users\jayap>docker images
REPOSITORY                          TAG       IMAGE ID       CREATED         SIZE
jayaprakash2019/docker101tutorial   latest    94ab66b2cb4d   3 weeks ago     28.9MB
docker/getting-started              latest    cb90f98fd791   6 months ago    28.8MB
```

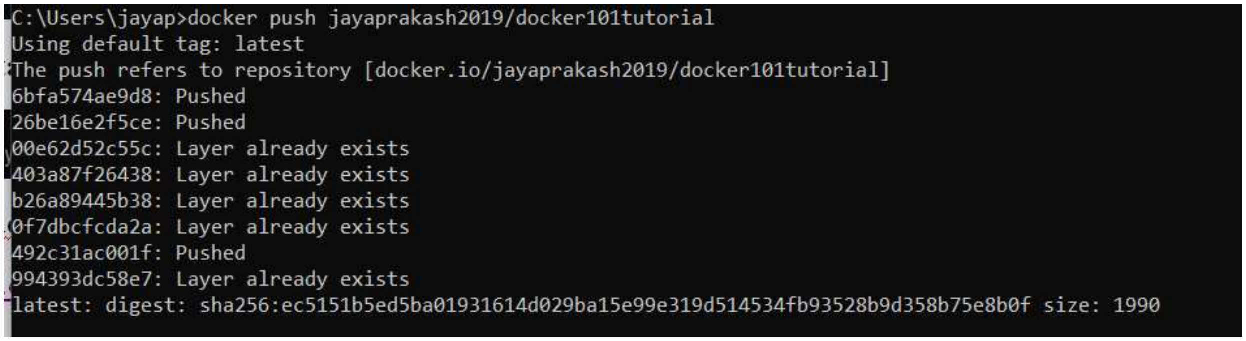
### STEP-2:Add the tag name for existing docker image to push the docker hub



```
C:\Users\jayap>docker tag jayaprakash2019/docker101tutorial:latest jayaprakash2019/jayaprakash2019/docker101tutorial
```

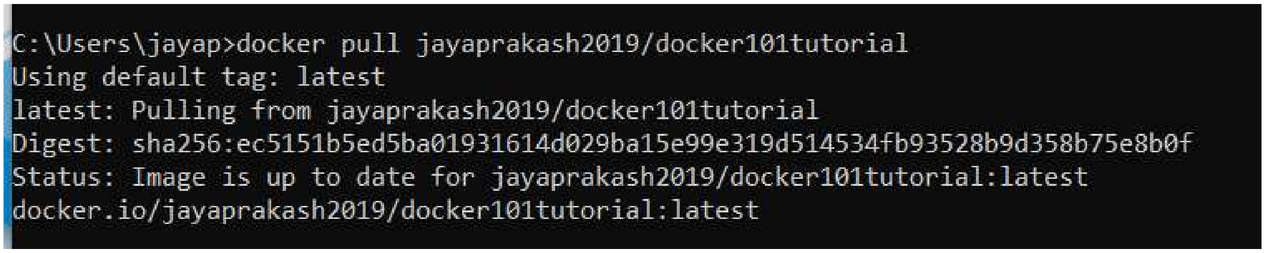
### STEP-3: Push to docker hub

https://hub.docker.com/r/jayaprakash2019/docker101tutorial



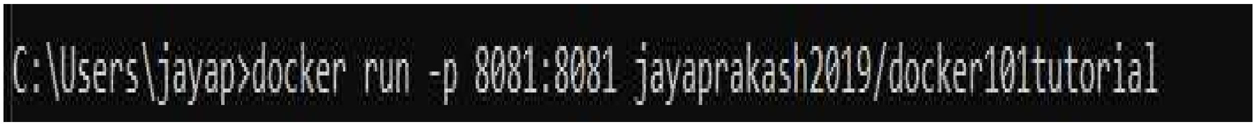
```
C:\Users\jayap>docker push jayaprakash2019/docker101tutorial
Using default tag: latest
The push refers to repository [docker.io/jayaprakash2019/docker101tutorial]
6bfa574ae9d8: Pushed
26be16e2f5ce: Pushed
00e62d52c55c: Layer already exists
403a87f26438: Layer already exists
b26a89445b38: Layer already exists
0f7dbcfcda2a: Layer already exists
492c31ac001f: Pushed
994393dc58e7: Layer already exists
latest: digest: sha256:ec5151b5ed5ba01931614d029ba15e99e319d514534fb93528b9d358b75e8b0f size: 1990
```

## STEP-4: Pull the docker images from the docker hub to docker container

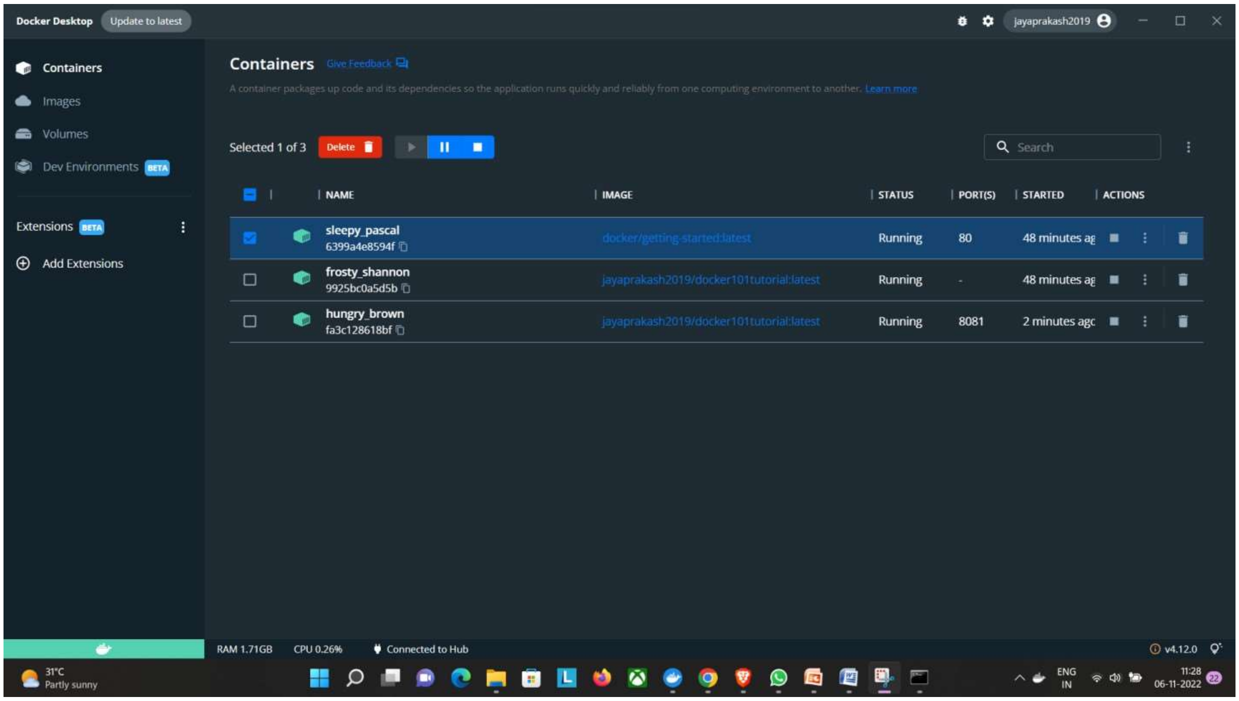


```
C:\Users\jayap>docker pull jayaprakash2019/docker101tutorial
Using default tag: latest
latest: Pulling from jayaprakash2019/docker101tutorial
Digest: sha256:ec5151b5ed5ba01931614d029ba15e99e319d514534fb93528b9d358b75e8b0f
Status: Image is up to date for jayaprakash2019/docker101tutorial:latest
docker.io/jayaprakash2019/docker101tutorial:latest
```

## STEP-5 :Run  the docker container



```
C:\Users\jayap>docker run -p 8081:8081 jayaprakash2019/docker101tutorial
```



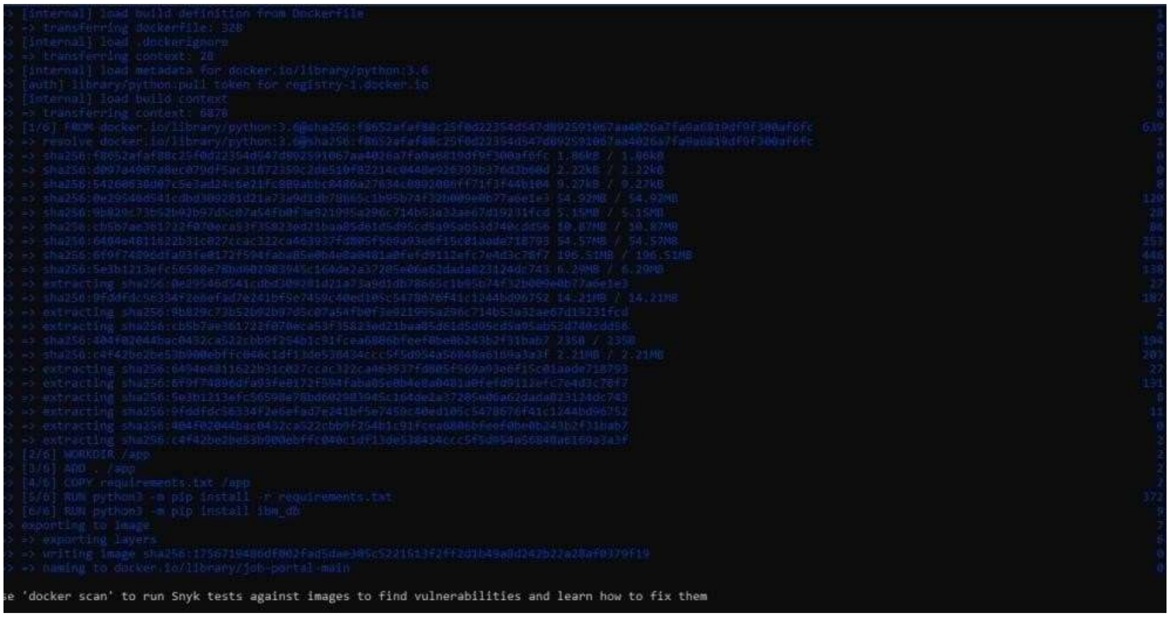
Containers

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more

Selected 1 of 3

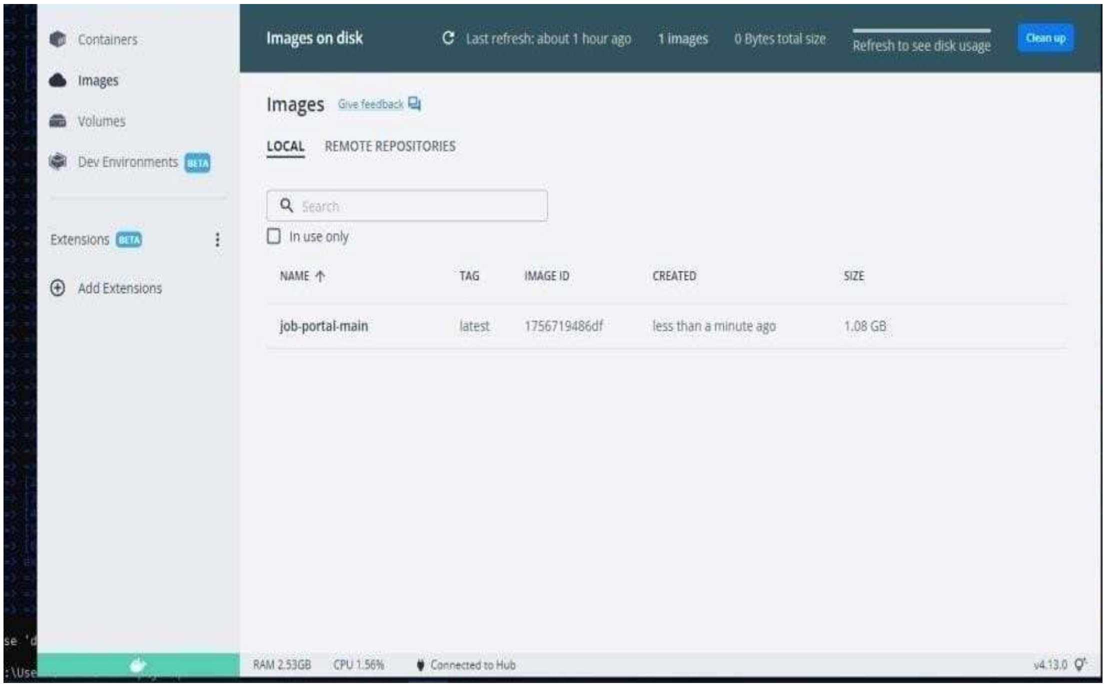
| | NAME | IMAGE | STATUS | PORT(S) | STARTED | ACTIONS |
|---|---|---|---|---|---|---|
| ☑ | sleepy_pascal<br>6399a4e8594f | docker/getting-started:latest | Running | 80 | 48 minutes ag | |
| □ | frosty_shannon<br>9925bc0a5d5b | jayaprakash2019/docker101tutorial:latest | Running | - | 48 minutes ag | |
| □ | hungry_brown<br>fa3c128618bf | jayaprakash2019/docker101tutorial:latest | Running | 8081 | 2 minutes agc | |

# 2) CREATE A DOCKER FILE FOR THE JOBPORTAL APPLICATION AND DEPLOY IT IN DOCKER DESKTOP APPLICATION



```
=> [internal] load build definition from Dockerfile                                                                                    1
=> => transferring dockerfile: 32B                                                                                                     0
=> [internal] load .dockerignore                                                                                                       1
=> => transferring context: 2B                                                                                                         0
=> [internal] load metadata for docker.io/library/python:3.6                                                                           9
=> [auth] library/python:pull token for registry-1.docker.io                                                                           0
=> [internal] load build context                                                                                                       1
=> => transferring context: 687B                                                                                                       0
=> [1/6] FROM docker.io/library/python:3.6@sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc                   639
=> => resolve docker.io/library/python:3.6@sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc                     1
=> => sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc 1.86kB / 1.86kB                                          0
=> => sha256:d097a4907a8ec079df5ac31872359c2de510f82214c0448e926393b376d2b60d 2.22kB / 2.22kB                                          0
=> => sha256:54260638d07c5e3ad24c6e21fc889abbc8486a27634c8892008ff71f3f44b104 9.27kB / 9.27kB                                          8
=> => sha256:0e29546d541cdbd309281d21a73a9d1db78b65c1b95b74f32b009e0b77a6e1e3 54.92MB / 54.92MB                                      120
=> => sha256:9b829c73b52b92b97d5c07a54fb0f3e921995a296c714b53a32ae67d19231fcd 5.15MB / 5.15MB                                         28
=> => sha256:cb5b7ae361722f070eca53f35823ed21baa85d61d5d05cd5a95ab53d740cdd56 10.87MB / 10.87MB                                       86
=> => sha256:6494e4811622b31c027ccac322ca463937fd805f569a93e6f15c81aade718793 54.57MB / 54.57MB                                      251
=> => sha256:6f9f74896dfa93fe8172f594faba85e0b4e8a0481a0fefd9112efc7e4d3c78f7 196.51MB / 196.51MB                                    446
=> => sha256:5e3b1213efc56598e78bd602983945c164de2a37285e06a62dada023124dc743 6.29MB / 6.29MB                                        138
=> => extracting sha256:0e29546d541cdbd309281d21a73a9d1db78b65c1b95b74f32b009e0b77a6e1e3                                              27
=> => sha256:9fddfdc56334f2e6efad7e241bf5e7459c40ed105c5478676f41c1244bd96752 14.21MB / 14.21MB                                      187
=> => extracting sha256:9b829c73b52b92b97d5c07a54fb0f3e921995a296c714b53a32ae67d19231fcd                                               2
=> => extracting sha256:cb5b7ae361722f070eca53f35823ed21baa85d61d5d05cd5a95ab53d740cdd56                                               4
=> => sha256:404f02044bac0432ca522cbb9f254b1c91fcea6806bfeef0be0b243b2f31bab7 235B / 235B                                            194
=> => sha256:c4f42be2be53b900ebffc046c1df13de538434ccc5f5d054a56848a6109a3a3f 2.21MB / 2.21MB                                        203
=> => extracting sha256:6494e4811622b31c027ccac322ca463937fd805f569a93e6f15c81aade718793                                              27
=> => extracting sha256:6f9f74896dfa93fe8172f594faba85e0b4e8a0481a0fefd9112efc7e4d3c78f7                                             131
=> => extracting sha256:5e3b1213efc56598e78bd602983945c164de2a37285e06a62dada023124dc743                                               0
=> => extracting sha256:9fddfdc56334f2e6efad7e241bf5e7459c40ed105c5478676f41c1244bd96752                                              11
=> => extracting sha256:404f02044bac0432ca522cbb9f254b1c91fcea6806bfeef0be0b243b2f31bab7                                               0
=> => extracting sha256:c4f42be2be53b900ebffc046c1df13de538434ccc5f5d054a56848a6109a3a3f                                               2
=> [2/6] WORKDIR /app                                                                                                                  2
=> [3/6] ADD . /app                                                                                                                    2
=> [4/6] COPY requirements.txt /app                                                                                                    2
=> [5/6] RUN python3 -m pip install -r requirements.txt                                                                              372
=> [6/6] RUN python3 -m pip install ibm_db                                                                                             9
=> exporting to image                                                                                                                  7
=> => exporting layers                                                                                                                 6
=> => writing image sha256:1756719486df802fad5dae305c5221613f2ff2d1b49a0d242b22a28af0379f19                                            0
=> => naming to docker.io/library/job-portal-main                                                                                      0

se 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```



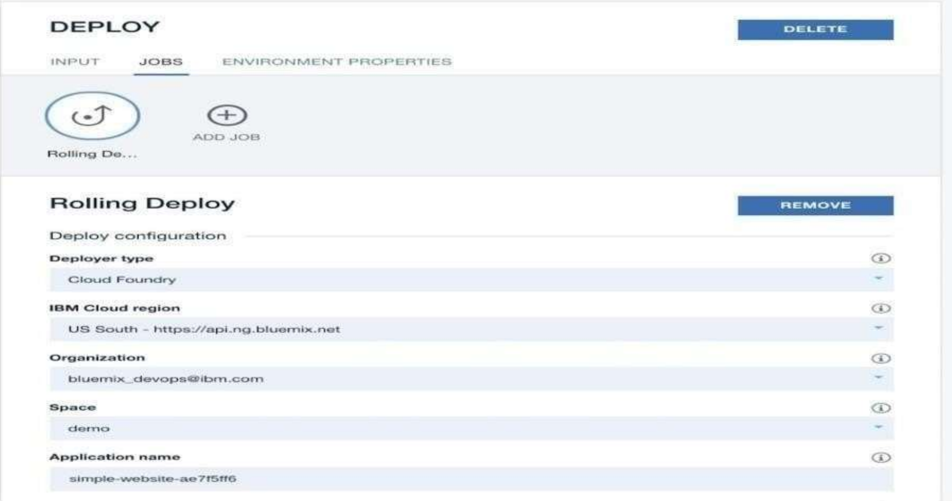
Images

| NAME ↑ | TAG | IMAGE ID | CREATED | SIZE |
|---|---|---|---|---|
| job-portal-main | latest | 1756719486df | less than a minute ago | 1.08 GB |

## 3. CREATE A IBM CONTAINER REGISTRY AND DEPLOY HELLOWORLD APP OR JOBPORTALAPP.

Solution:

```
          <html>
     <body>
        Hello,IBMCloudWorld!
     </body>
</html>---
application
s:
- buildpack: https://github.com/cloudfoundry/staticfile-
  buildpack.githost:simple-website-${random}
  name: simple-website-
  ${random}memory:64M
  stack:cflinuxfs2
```



Rolling Deploy

Deploy configuration

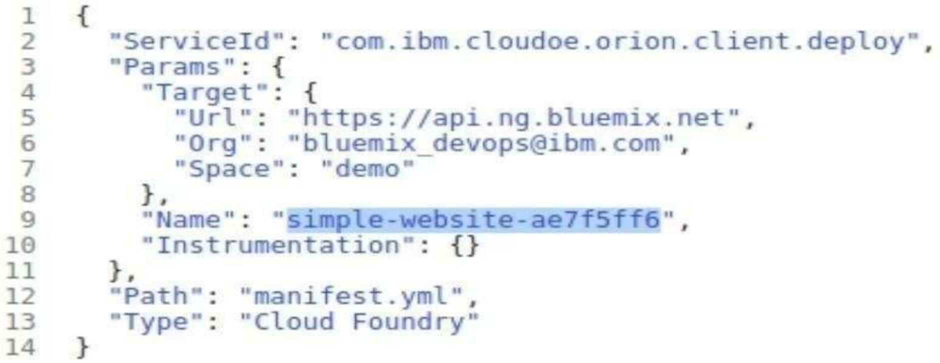
Deployer type: Cloud Foundry

IBM Cloud region: US South - https://api.ng.bluemix.net

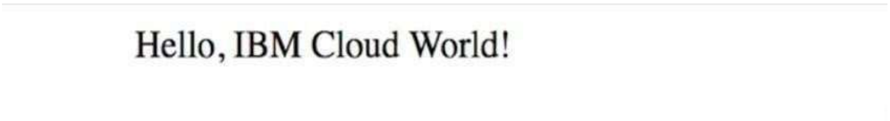
Organization: bluemix_devops@ibm.com

Space: demo

Application name: simple-website-ae7f5ff6



```
{
  "ServiceId": "com.ibm.cloudoe.orion.client.deploy",
  "Params": {
    "Target": {
      "Url": "https://api.ng.bluemix.net",
      "Org": "bluemix_devops@ibm.com",
      "Space": "demo"
    },
    "Name": "simple-website-ae7f5ff6",
    "Instrumentation": {}
  },
  "Path": "manifest.yml",
  "Type": "Cloud Foundry"
}
```



Hello, IBM Cloud World!

## 4) CREATE A KUBERNETES CLUSTER IN IBM CLOUD AND DEPLOY HELLOWORLD IMAGE OR JOBPORTAL IMAGE AND ALSO EXPOSE THE SAME APP TO RUN IN NODEPORT.

Solution:

ibmcloudtarget-g<resource_group_name>ibmcloudcrnishanthc-add <your_nishanthc>ibmcloudresource service-instance-create example-postgresql databases-for-postgresql standard us- southibmcloudks cluster-service-bind mycluster default example-postgresqlgit clone -b node git@github.com:IBM-Cloud/clouddatabases-helloworld-kubernetes-examples.gitspec:

replicas:3name:cloudpostgres-nodejs-app

image:"registry.<region>.bluemix.net/<namespace>/icdpg"#Editme

imagePullPolicy: AlwaysibmcloudcrregionYou are targeting region 'us-south', the registry is'registry.ng.bluemix.net'.ibmcloudcr build -t registry.ng.bluemix.net/<namespace>/icdpg .ibmcloudcrimages

env:

- name:

BINDINGvalue

From:

```
         secretKeyRef:
          name: <postgres-secret-name> # Edit
          mekey:binding
apiVersion:
v1kind:
Servicemeta
data:
 name: cloudpostgres-
 servicelabels:
  run: clouddb-
demospec:
type:
NodePortsele
ctor:
  run: clouddb-
 demoports:
 -
  protocol:TC
  Pport:8080
  nodePort:30081
```