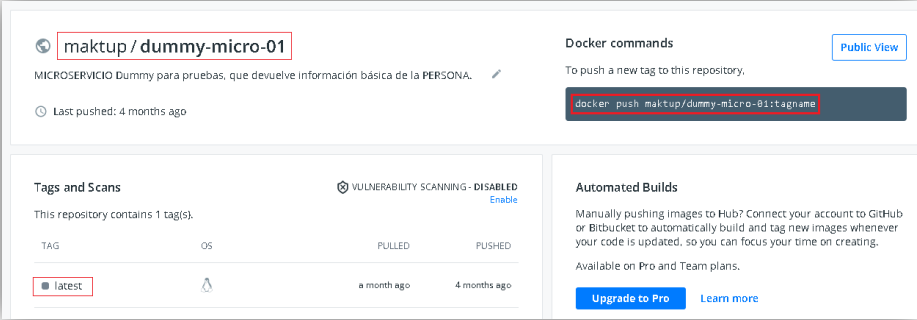
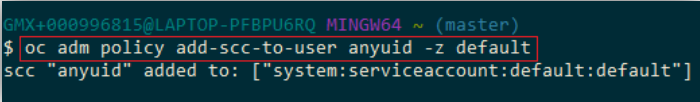


ENTRENAMIENTO: “OPENSIFT v4”

El objetivo del siguiente documento es detallar el manejo de *OPENSIFT*, su funcionamiento & algunas comparativas con *KUBERNETES* propiamente.

DESCRIPCIÓN	DETALLE
DESPLIEGUE APLICACIONES: ‘KUBERNETES’ & ‘OPENSIFT’	
DUMMY: ‘IMPERATIVE COMMANDs’	
<p>Se trabajará con un repositorio <i>IMAGE REGISTRY</i> ya preparado:</p> <p>https://hub.docker.com/repository/docker/maktup/dummy-micro-01</p>	
<p>IMPORTANTE: para crear CONTENEDORES sin problemas que manejen [IMAGENES EXTERNAS], se requiere ejecutar el COMANDO:</p> <p>//Openshift: \$ oc adm policy add-scc-to-user anyuid -z default</p> <p>Luego, los COMANDOS BASE, para el DESPLIEGUE de una aplicación en KUBERNETES serían:</p>	

#[NAMESPACE]:

//Kubernetes:

```
$ kubectl create namespace dummy-csm-kub
```

//Openshift:

```
$ oc new-project dummy-csm-ope
```

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ kubectl create namespace dummy-csm-kub
namespace/dummy-csm-kub created
```

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc new-project dummy-csm-ope
```

Already on project "dummy-csm-ope" on server "https://c100-e.us-south.containers.cloud.ibm.com:31470".

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

```
    oc new-app centos/ruby-25-centos7~https://github.com/sclorg/ruby-ex.git

to build a new example application in Ruby.
```

#[DEPLOYMENT]:

//Kubernetes:

```
$ kubectl create deployment dummy-micro-01 --
image=maktup/dummy-micro-01:latest -n dummy-csm-kub
```

#[DEPLOYMENTCONFIGS]:

//Openshift:

```
$ oc new-app maktup/dummy-micro-01:latest -n dummy-csm-
ope
```

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
```

```
$ kubectl create deployment dummy-micro-01 --image=maktup/dummy-micro-01:latest -n dummy-csm-kub
deployment.apps/dummy-micro-01 created
```

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
```

```
$ oc new-app maktup/dummy-micro-01:latest -n dummy-csm-ope
```

--> Found Docker image 96274bc (4 months old) from Docker Hub for "maktup/dummy-micro-01:latest"

- * An image stream tag will be created as "dummy-micro-01:latest" that will track this image
- * This image will be deployed in deployment config "dummy-micro-01"
- * Port 8080/tcp will be load balanced by service "dummy-micro-01"
- * Other containers can access this service through the hostname "dummy-micro-01"
- * This image declares volumes and will default to use non-persistent, host-local storage. You can add persistent volumes later by running 'volume dc/dummy-micro-01 --add ...'
- * WARNING: Image "maktup/dummy-micro-01:latest" runs as the 'root' user which may not be permitted for all containers

--> Creating resources ...

```
imagestream.image.openshift.io "dummy-micro-01" created
deploymentconfig.apps.openshift.io "dummy-micro-01" created
service "dummy-micro-01" created
```

--> Success

Application is not exposed. You can expose services to the outside world by executing one or more of the following commands:

```
'oc expose svc/dummy-micro-01'
```

Run 'oc status' to view your app.

#[SERVICE]:

//Kubernetes:

```
$ kubectl expose deployment dummy-micro-01 --port=8080 --type=NodePort -n dummy-csm-kub
```

//Openshift:

No require (el **DeploymentConfig** ya lo crea).

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ kubectl expose deployment dummy-micro-01 --port=8080 --type=NodePort -n dummy-csm-kub
service/dummy-micro-01 exposed
```

#[ROUTE]:

//Kubernetes:

No usa, aquí se maneja un recurso: **INGRESS**.

//Openshift:

```
$ oc expose service dummy-micro-01 -n dummy-csm-ope
```

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc expose service dummy-micro-01 -n dummy-csm-ope
route.route.openshift.io/dummy-micro-01 exposed
```

Luego, para consultar los **RECURSOS** creados ejecutar el **COMANDO**:

//Kubernetes:

```
$ kubectl get pod,service,deployment -n dummy-csm-kub
$ kubectl get all -n dummy-csm-kub
```

//Openshift:

```
$ oc get pod,service,deploymentconfig -n dummy-csm-ope
$ oc get all -n dummy-csm-ope
```

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ kubectl get pod,service,deployment -n dummy-csm-kub
NAME                                READY    STATUS    RESTARTS   AGE
pod/dummy-micro-01-89bb48cdf-5nzvd  1/1      Running   0           2m13s

NAME                                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)
service/dummy-micro-01              NodePort    172.21.123.54 <none>         8080:31094/TCP

NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/dummy-micro-01      1/1      1              1             2m14s
```

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ kubectl get all -n dummy-csm-kub
NAME                                READY    STATUS    RESTARTS   AGE
pod/dummy-micro-01-89bb48cdf-5nzvd  1/1      Running   0           5m18s

NAME                                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)
service/dummy-micro-01              NodePort    172.21.123.54 <none>         8080:31094/TCP

NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/dummy-micro-01      1/1      1              1             5m20s

NAME                                DESIRED    CURRENT    READY    AGE
replicaset.apps/dummy-micro-01-89bb48cdf  1          1          1        5m20s
```

Finalmente, para **ELIMINAR** todos los **RECURSOS** creados, ejecutar los **COMANDOS** siguientes:

//Kubernetes:
\$ kubectl delete ns dummy-csm-kub --force --grace-period 0

//Openshift:
\$ oc delete ns dummy-csm-ope --force --grace-period 0

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc get pod,service,deploymentconfig -n dummy-csm-ope
NAME                                READY    STATUS    RESTARTS   AGE
pod/dummy-micro-01-1-6rszd         1/1     Running   0           4m34s
pod/dummy-micro-01-1-deploy        0/1     Completed 0           5m3s

NAME                                TYPE      CLUSTER-IP    EXTERNAL-IP    PORT(S)
service/dummy-micro-01             ClusterIP  172.21.171.142 <none>         8080/TCP

NAME                                REVISION    DESIRED    CURRENT
deploymentconfig.apps.openshift.io/dummy-micro-01  1           1          1
```

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc get all -n dummy-csm-ope
NAME                                READY    STATUS    RESTARTS   AGE
pod/dummy-micro-01-1-6rszd         1/1     Running   0           3m22s
pod/dummy-micro-01-1-deploy        0/1     Completed 0           3m51s

NAME                                DESIRED    CURRENT    READY    AGE
replicationcontroller/dummy-micro-01-1  1          1          1        3m51s

NAME                                TYPE      CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
service/dummy-micro-01             ClusterIP  172.21.171.142 <none>         8080/TCP   3m53s

NAME                                REVISION    DESIRED    CURRENT    TRIGGERED BY
deploymentconfig.apps.openshift.io/dummy-micro-01  1           1          1          config,image(dummy-micro-01:latest)

NAME                                IMAGE REPOSITORY    TAGS    UPDATED
imagestream.image.openshift.io/dummy-micro-01  image-registry.openshift-image-registry.svc:5000/dummy-csm-ope/dummy-micro-01  latest  3 minutes ago

NAME                                HOST/PORT
route.route.openshift.io/dummy-micro-01  dummy-micro-01-dummy-csm-ope.cluster.openshift-cla-cp4-ccc03eca20d26e6ac64511f074a64b9b-0000.us-south.containers.appdomain.cloud
```

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ kubectl delete ns dummy-csm-kub --force --grace-period 0
warning: Immediate deletion does not wait for confirmation that the running resource has been terminated.
namespace "dummy-csm-kub" force deleted
```

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc delete ns dummy-csm-ope --force --grace-period 0
warning: Immediate deletion does not wait for confirmation that the running resource has been terminated.
namespace "dummy-csm-ope" force deleted
```



Imperative-Commands [Kubernetes-Openshift]

DUMMY: 'YAMEL SCRIPTs'

Aquí se mostrarán todos los **SCRIPTs BASE**, para el **DESPLIEGUE** de una aplicación en **KUBERNETES**:

#[NAMESPACE]:

```
apiVersion: v1
kind: Namespace
metadata:
  name: dummy-csm
```

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ cat > Script.yaml
### ----- [NAMESPACE] ----- ###
apiVersion: v1
kind: Namespace
metadata:
  name: dummy-csm
---
```

#[DEPLOYMENT]:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: dummy-micro-deploy
  namespace: dummy-csm
  labels:
    app: dummy-micro-service
    version: v1
spec:
  replicas: 2
  selector:
    matchLabels:
      app: dummy-micro-service
      version: v1
  template:
    metadata:
      labels:
        app: dummy-micro-service
        version: v1
    spec:
      containers:
        - image: maktup/dummy-micro-01:latest
          name: dummy-micro-container
          resources:
            limits:
              cpu: 300m
            requests:
              cpu: 100m
          ports:
            - containerPort: 8080
```

```
### ----- [DEPLOYMENT] ----- ###
apiVersion: apps/v1
kind: Deployment
metadata:
  name: dummy-micro-deploy
  namespace: dummy-csm
  labels:
    app: dummy-micro-service
    version: v1
spec:
  replicas: 1
  selector:
    matchLabels:
      app: dummy-micro-service
      version: v1
  template:
    metadata:
      labels:
        app: dummy-micro-service
        version: v1
    spec:
      containers:
        - image: maktup/dummy-micro-01:latest
          name: dummy-micro-container
          resources:
            limits:
              cpu: 300m
            requests:
              cpu: 100m
          ports:
            - containerPort: 8080
```

<pre> #[POD]: apiVersion: v1 kind: Pod metadata: name: dummy-micro-consumer namespace: dummy-csm labels: run: dummy-micro-consumer spec: containers: - image: nginx name: dummy-micro-consumer ports: - containerPort: 80 </pre>	<pre> #### ----- [POD] ----- #### apiVersion: v1 kind: Pod metadata: name: dummy-micro-consumer namespace: dummy-csm labels: run: dummy-micro-consumer spec: containers: - image: nginx name: dummy-micro-consumer ports: - containerPort: 80 --- </pre>
<pre> #[SERVICE]: apiVersion: v1 kind: Service metadata: name: dummy-micro-service namespace: dummy-csm labels: app: dummy-micro-service spec: type: ClusterIP ports: - port: 8080 protocol: TCP targetPort: 8080 selector: app: dummy-micro-service #[ROUTE]: apiVersion: route.openshift.io/v1 kind: Route metadata: name: dummy-micro-route namespace: dummy-csm labels: app: dummy-micro-service spec: port: targetPort: 8080 to: kind: Service name: dummy-micro-service </pre>	<pre> #### ----- [SERVICE] ----- #### apiVersion: v1 kind: Service metadata: name: dummy-micro-service namespace: dummy-csm labels: app: dummy-micro-service spec: type: ClusterIP ports: - port: 8080 protocol: TCP targetPort: 8080 selector: app: dummy-micro-service --- #### ----- [ROUTE] ----- #### apiVersion: route.openshift.io/v1 kind: Route metadata: name: dummy-micro-route namespace: dummy-csm labels: app: dummy-micro-service spec: port: targetPort: 8080 to: kind: Service name: dummy-micro-service </pre>

Luego, se debe ejecutar el Script **YAMEL** & consultar lo creado ejecutando los **COMANDOS**:

```
//Kubernetes:
$ kubectl create -f script.yaml
$ kubectl get all -n dummy-csm

//Openshift:
$ oc create -f script.yaml
$ oc get all -n dummy-csm
```

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc create -f Script.yaml
namespace/dummy-csm created
deployment.apps/dummy-micro-deploy created
pod/dummy-micro-consumer created
service/dummy-micro-service created
route.route.openshift.io/dummy-micro-route created

GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc get all -n dummy-csm
NAME                                READY   STATUS    RESTARTS   AGE
pod/dummy-micro-consumer           1/1     Running   0           44s
pod/dummy-micro-deploy-547d776df9-vqd5d  1/1     Running   0           44s

NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP  PORT(S)    AGE
service/dummy-micro-service         ClusterIP     172.21.168.228 <none>       8080/TCP    44s

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/dummy-micro-deploy  1/1     1             1           44s

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/dummy-micro-deploy-547d776df9  1         1         1       44s

NAME                                HOST/PORT
route.route.openshift.io/dummy-micro-route  dummy-micro-route-dummy-csm.cluster-openshift-cla-cp4-ccc03eca20d26e6ac64511f074a64b9b-0000.us-south.containers.appdomain.cloud
```

Finalmente, para **ELIMINAR** todos los **RECURSOS** creados, reutilizar el mismo **SCRIPT** ya creado:

```
//Kubernetes:
$ kubectl delete -f script.yaml

//Openshift:
$ oc delete -f script.yaml
```

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc delete -f Script.yaml
namespace "dummy-csm" deleted
deployment.apps "dummy-micro-deploy" deleted
pod "dummy-micro-consumer" deleted
service "dummy-micro-service" deleted
route.route.openshift.io "dummy-micro-route" deleted
```



Script -
[Kubernetes-Opens]

```
$ while true; do curl -s curl http://dummy-micro-route-dummy-  
csm.cluster-openshift-cla-cp4-  
ccc03eca20d26e6ac64511f874a64b9b-0000.us-  
south.containers.appdomain.cloud/dummy-micro-  
01/get/personas; done
```

[illegible]

→ No seguro dummy-micro-route-dummy-csm.cluster-openshift-da-cp4-ccc03eca20d26e6ac64511f874a64b9b-0000.us-south.co... Actualizar

```
{ 'nombre': 'PAOLO GUERRERO', 'edad': 35, 'rol': 'CONSULTOR' }, { 'nombre': 'LUIS GUADALUPE', 'edad': 40, 'rol': 'PROGRAMADOR' }, { 'nombre': 'PEDRO SALAZAR', 'edad': 30, 'rol': 'ARQUITECTO' }
```

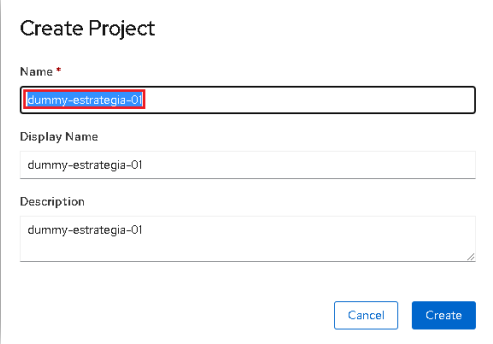
[illegible]

A. DESDE 'CONTAINER IMAGE':

MODALIDAD GRÁFICA:

Estas **ESTRATEGIAS** mostrarán como **DESPLEGAR** el mismo **DUMMY** trabajado previamente, pero sin necesidad del **CÓDIGO** (Todo desde el **DASHBOARD**) de **OPENSIFT**.

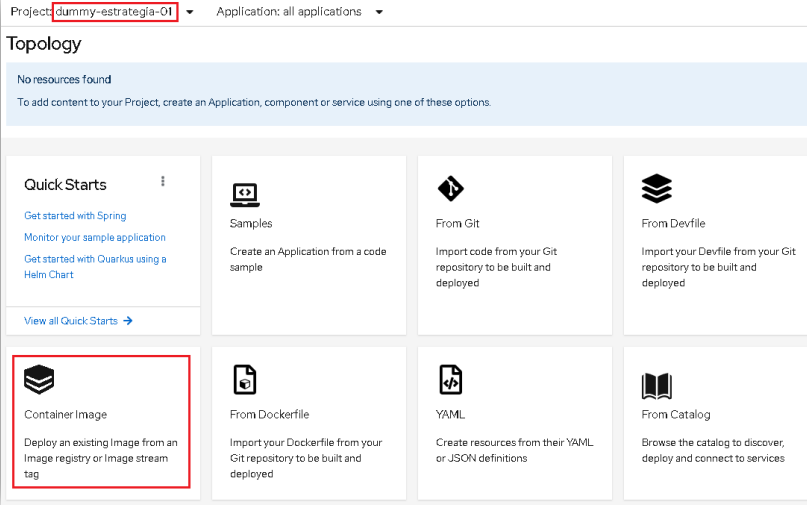
Luego, se procede a crear un **Namespace/Project** llamado: **dummy-estrategia-01**



A 'Create Project' dialog box with a white background and a light gray border. It contains three input fields: 'Name' with the value 'dummy-estrategia-01' (highlighted with a red box), 'Display Name' with the value 'dummy-estrategia-01', and 'Description' with the value 'dummy-estrategia-01'. At the bottom right are two buttons: 'Cancel' and 'Create'.

En este caso, se utilizará una **IMAGE** ya existente desde un **REPOSITORIO** local o desde uno **EXTERNO** (**DOCKERHUB**).

Luego, se seleccionará la opción de **DESPLIEGUE** como: **FROM CONTAINER IMAGE**.



A screenshot of the 'Topology' page in a web application. The page has a header with 'Project: dummy-estrategia-01' and 'Application: all applications'. Below the header, a message states 'No resources found' and provides instructions to add content. The main area displays a grid of deployment options. The 'Container Image' option is highlighted with a red box. Other options include 'Quick Starts', 'Samples', 'From Git', 'From Devfile', 'From Dockerfile', 'YAML', and 'From Catalog'.

Luego, ingresar la información tal como se muestra en el formulario:

El **NOMBRE** de la **IMAGEN** se obtiene del **REPOSITORIO** de **IMÁGENES**:

<https://hub.docker.com/repository/docker/maktup/dummy-micro-01>.

- **Image name:** maktup/dummy-micro-01:latest
- **Allow Images tag from insecure registries:** check
- **Runtime Icon:** openshift
- **Application Name:** dummy-micro-01-app
- **Name:** dummy-micro-01

Image

Deploy an existing Image from an Image Stream or Image registry.

☒ Image name from external registry

maktup/dummy-micro-01:latest

Validated

To deploy an Image from a private repository, you must [create an Image pull secret](#) with your Image registry credentials.

☒ Allow Images from insecure registries

☐ Image stream tag from internal registry

Runtime icon

openshift

The icon represents your Image in Topology view. A label will also be added to the resource defining the icon.

General

Application name

dummy-micro-01-app

A unique name given to the Application grouping to label your resources.

Name *

dummy-micro-01

A unique name given to the component that will be used to name associated resources.

Luego, elegir el tipo de **RECURSO** para el **DESPLIEGUE**:

- ✓ **DEPLOYMENT** => Modalidad **Kubernetes**.
- ✓ **DEPLOYMENTCONFIG** => Modalidad **Openshift**

& dar **check** por si se requiere que dicho **MICROSERVICIO** sea consumido **externamente**. En este caso se le creará un **RECURSO** de tipo **ROUTE**.

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.

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

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

Luego, se verifica que **GRÁFICAMENTE** se muestran los diferentes **RECURSOS** creados en **OPENSIFT**, así mismo, al darle click se detalla a la **DERECHA** opciones como para ingresar al **POD**, a los **LOGs** del **CONTENEDOR**, acceso a la **URL** base del **MICROSERVICIO** desplegado, etc.

Developer

Project: **dummy-estrategia-01** Application: all applications

+Add

Topology

Monitoring

Search

Builds

Helm

Project

ConfigMaps

Secrets

Display options Filter by resource Find by name...

dummy-micro-01

Health checks

Container dummy-micro-01 does not have health checks to ensure your Application is running correctly. [Add health checks](#)

Details **Resources** Monitoring

Pods

dummy-micro-01-1-s27zc Running [View logs](#)

Services

dummy-micro-01

Service port: 8080-tcp → Pod Port: 8080

Routes

dummy-micro-01

Location:

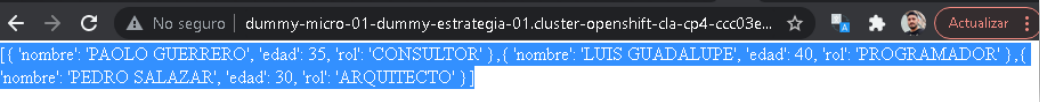
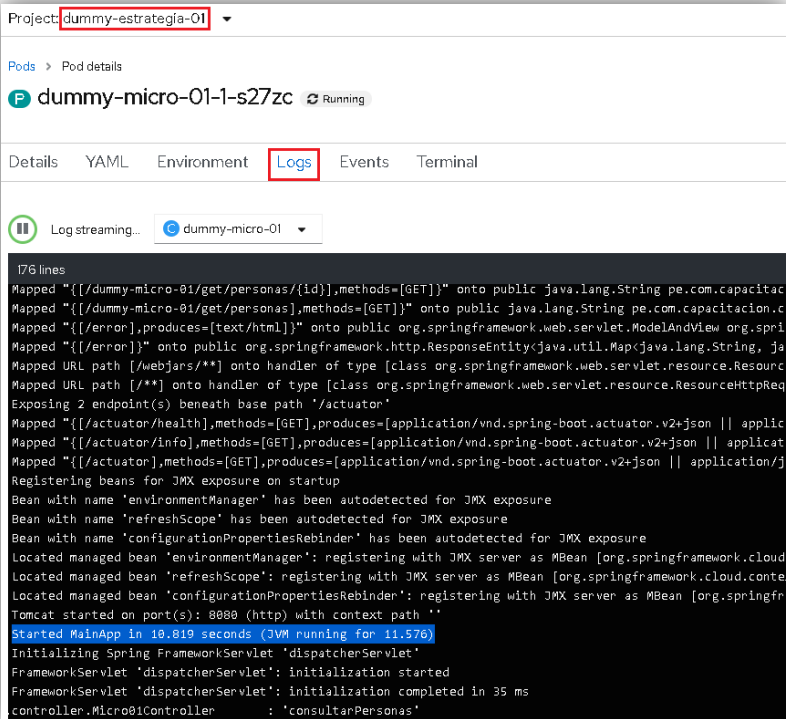
<http://dummy-micro-01-dummy-estrategia-01-cluster-openshift-cla-cp4-ccc03eca20d26e9ac64511f674a64b9b-0000.us-south.containers.appdomain.cloud/>

Finalmente, se verifican los **LOGs** del **CONTENEDOR** dentro del **POD** & se valida que el **MICROSERVICIO** esté ya iniciado.

Luego se accede a la **URL** del **MICROSERVICIO** concatenándole la **URI** que maneja:

```
$ curl http://dummy-micro-01-dummy-estrategia-01.cluster-openshift-cla-cp4-ccc03eca20d26e6ac64511f874a64b9b-0000.us-south.containers.appdomain.cloud/dummy-micro-01/get/personas
```

IMPORTANTE: el **DNS** de la **URL** en este caso es autogenerated & debe ser actualizado.



B. DESDE 'DOCKERFILE':

MODALIDAD GRÁFICA:

Esta **ESTRATEGIA** se aplicará cuando el **ORIGEN** es un archivo de tipo: **DOCKERFILE** en alguna ubicación **EXTERNA**.

Luego, se procede a crear un **Namespace/Project** llamado: **dummy-estrategia-02**

Create Project

Name *

dummy-estrategia-02

Display Name

dummy-estrategia-02

Description

dummy-estrategia-02

Cancel

Create


Luego, se seleccionará la opción de **DESPLIEGUE** como: **FROM DOCKERFILE**

Project: dummy-estrategia-02 Application: all applications

Quick Starts


- [Get started with Spring](#)
- [Monitor your sample application](#)
- [Get started with Quarkus using a Helm Chart](#)

[View all Quick Starts →](#)




Samples

Create an Application from a code sample




From Git

Import code from your Git repository to be built and deployed




From Devfile

Import your Devfile from your Git repository to be built and deployed



Container Image

Deploy an existing Image from an Image registry or Image stream tag



From Dockerfile

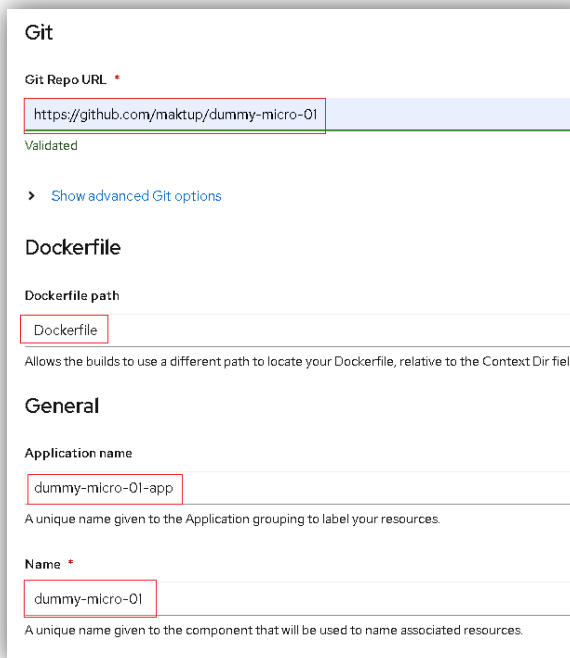
Import your Dockerfile from your Git repository to be built and deployed

Luego, ingresar la **URL** del **REPOSITORIO GIT** donde esté ubicado el **DOCKERFILE**, junto con la fuente del **MICROSERVICIOS** almacenado.

<https://github.com/maktup/dummy-micro-01>

Los datos para ingresar son:

- **Git Repo URL:**
<https://github.com/maktup/dummy-micro-01>
- **Dockerfile Path:** Dockerfile
- **Application Name:** dummy-micro-01-app
- **Name:** dummy-micro-01

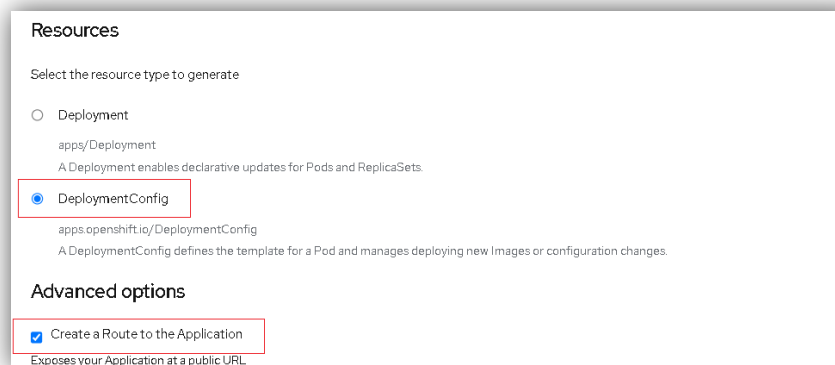


The screenshot shows a 'Git' configuration panel. It has a 'Git Repo URL' field with the value 'https://github.com/maktup/dummy-micro-01' and a 'Validated' status. Below it is a 'Show advanced Git options' link. The 'Dockerfile' section has a 'Dockerfile path' field with the value 'Dockerfile'. The 'General' section has an 'Application name' field with the value 'dummy-micro-01-app' and a 'Name' field with the value 'dummy-micro-01'.

Luego, elegir el tipo de **RECURSO** para el **DESPLIEGUE**:

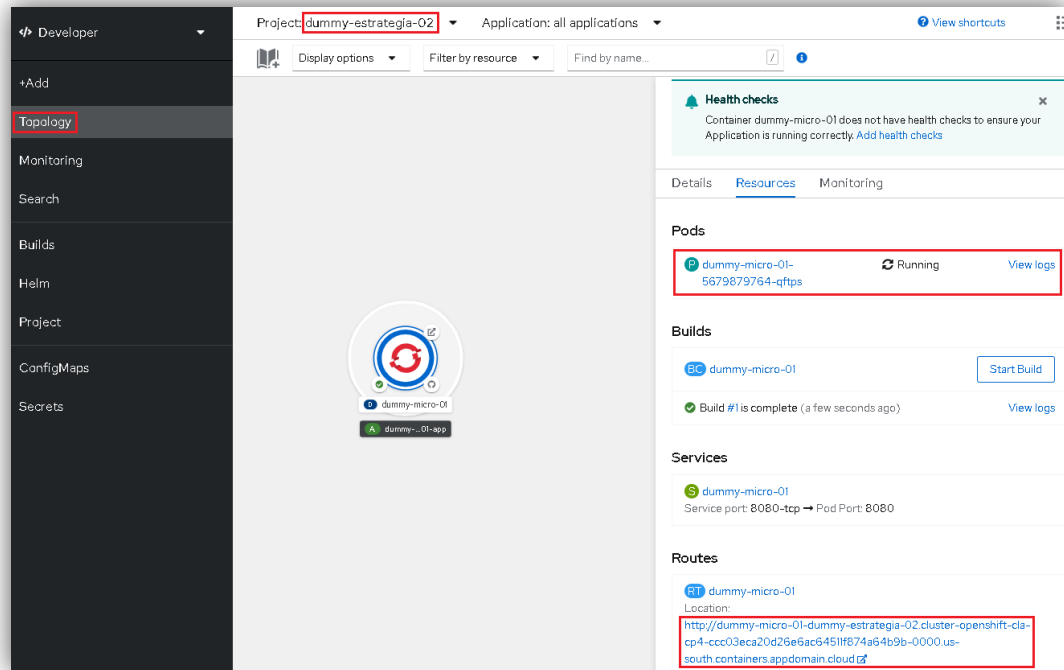
- ✓ **DEPLOYMENT** => Modalidad **Kubernetes**.
- ✓ **DEPLOYMENTCONFIG** => Modalidad **Openshift**

& dar **check** por si se requiere que dicho **MICROSERVICIO** sea consumido **externamente**. En este caso se le creará un **RECURSO** de tipo **ROUTE**.



The screenshot shows a 'Resources' configuration panel. It has a 'Select the resource type to generate' section with two options: 'Deployment' and 'DeploymentConfig'. 'DeploymentConfig' is selected. Below it is an 'Advanced options' section with a checkbox 'Create a Route to the Application' which is checked.

Luego, se verifica que **GRÁFICAMENTE** se muestran los diferentes **RECURSOS** creados en **OPENSIFT**, así mismo, al darle click se detalla a la **DERECHA** opciones como para ingresar al **POD**, a los **LOGs** del **CONTENEDOR**, acceso a la **URL** base del **MICROSERVICIO** desplegado, etc.

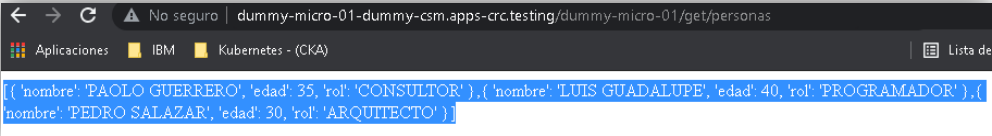
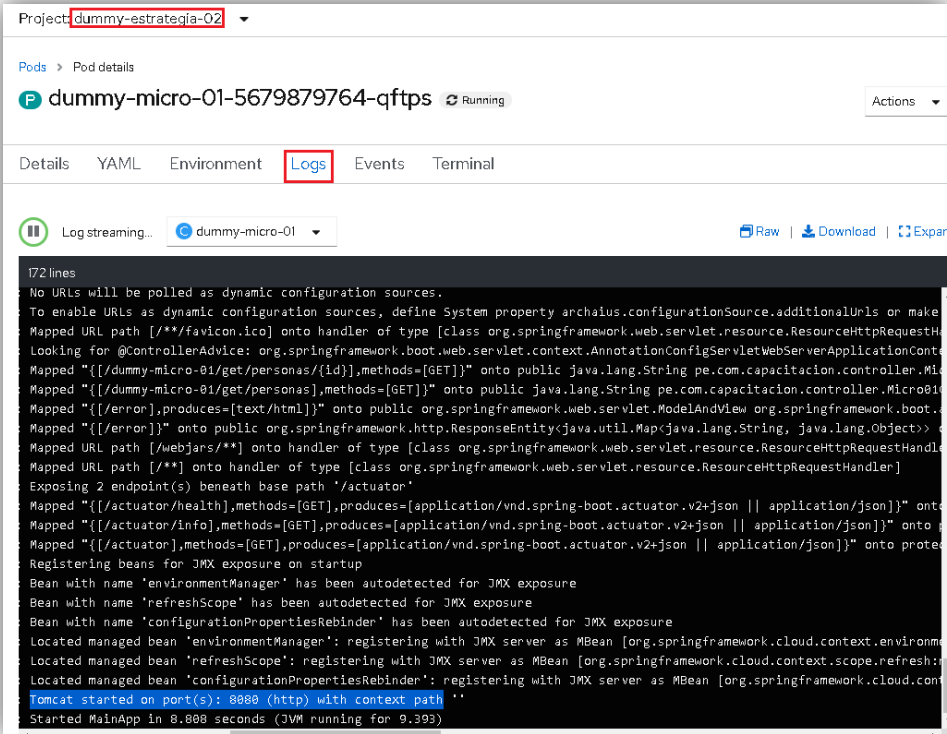


Finalmente, se verifican los **LOGS** del **CONTENEDOR** dentro del **POD** & se valida que el **MICROSERVICIO** esté ya iniciado.

Luego se accede a la **URL** del **MICROSERVICIO** concatenándole la **URI** que maneja:

```
$ curl http://dummy-micro-01-dummy-estrategia-01.cluster-openshift-cla-cp4-ccc03eca20d26e6ac64511f874a64b9b-0000.us-south.containers.appdomain.cloud/dummy-micro-01/get/personas
```

IMPORTANTE: el **DNS** de la **URL** en este caso es autogenerado & debe ser actualizado.



MODALIDAD POR CÓDIGO:

Esta modalidad realiza exactamente lo mismo que la modalidad anterior, solo que por **CÓDIGO**:

Aquí se trabaja en base a la identificación de un **DOCKERFILE** existente, en una ruta específica como una **URL** de un **REPOSITORIO** de **FUENTES** existente.

Ingresar los **COMANDOS** siguientes:

```
$ oc create ns dummy-estrategia-02b
$ oc new-app --name=dummy-micro-01 --strategy=docker https://github.com/maktup/dummy-micro-01.git -n dummy-estrategia-02b
```

En la **IMAGEN** se aprecia que el archivo **DOCKERFILE**, ha sido reconocido & en base a él se empieza a generar la **IMAGEN** & luego la creación de los **RECURSOS**.

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc create ns dummy-estrategia-02b
namespace/dummy-estrategia-02b created

GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc new-app --name=dummy-micro-01 --strategy=docker https://github.com/maktup/dummy-micro-01.git -n dummy-estrategia-02b
warning: Cannot check if git requires authentication.
--> Found Docker image 0b45b6c (3 days old) from Docker Hub for "adoptopenjdk/openjdk8:alpine-slim"

* An image stream tag will be created as "openjdk8:alpine-slim" that will track the source image
* A Docker build using source code from https://github.com/maktup/dummy-micro-01.git will be created
* The resulting image will be pushed to image stream tag "dummy-micro-01:latest"
* Every time "openjdk8:alpine-slim" changes a new build will be triggered
* WARNING: this source repository may require credentials.
  Create a secret with your git credentials and use 'set build-secret' to assign it to the build config.
* This image will be deployed in deployment config "dummy-micro-01"
* Port 8080/tcp will be load balanced by service "dummy-micro-01"
* Other containers can access this service through the hostname "dummy-micro-01"
* WARNING: Image "adoptopenjdk/openjdk8:alpine-slim" runs as the 'root' user which may not be permitted by your cluster administrator

--> Creating resources ...
imagestream.image.openshift.io "openjdk8" created
imagestream.image.openshift.io "dummy-micro-01" created
buildconfig.build.openshift.io "dummy-micro-01" created
deploymentconfig.apps.openshift.io "dummy-micro-01" created
service "dummy-micro-01" created
--> Success
Build scheduled, use 'oc logs -f bc/dummy-micro-01' to track its progress.
Application is not exposed. You can expose services to the outside world by executing one or more of the commands below:
'oc expose svc/dummy-micro-01'
Run 'oc status' to view your app.
```

Luego, se expone & consulta ejecutando el **COMANDO**:

```
$ oc expose servicio dummy-micro-01 -n dummy-estrategia-02b
$ oc get all -n dummy-estrategia-02b
```

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc expose service dummy-micro-01 -n dummy-estrategia-02b
route.route.openshift.io/dummy-micro-01 exposed

GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc get all -n dummy-estrategia-02b
NAME                                READY     STATUS    RESTARTS   AGE
pod/dummy-micro-01-1-4sww4         1/1      Running   0           2m1s
pod/dummy-micro-01-1-build         0/1      Completed 0           3m38s
pod/dummy-micro-01-1-deploy        0/1      Completed 0           2m5s

NAME                                DESIRED   CURRENT   READY     AGE
replicationcontroller/dummy-micro-01-1 1         1         1         2m5s

NAME                                TYPE        CLUSTER-IP    EXTERNAL-IP  PORT(S)    AGE
service/dummy-micro-01              ClusterIP   172.17.29.59  <none>       8080/TCP   3m38s

NAME                                REVISION   DESIRED   CURRENT   TRIGGERED BY
deploymentconfig.apps.openshift.io/dummy-micro-01 1          1         1          config,image(dummy-micro-01:latest)

NAME                                TYPE        FROM        LATEST
buildconfig.build.openshift.io/dummy-micro-01    Docker     Git         1

NAME                                TYPE        FROM        STATUS    STARTED     DURATION
build.build.openshift.io/dummy-micro-01-1        Docker     Git@12dc7d5 Complete  3 minutes ago 1m33s

NAME                                IMAGE REPOSITORY                                TAGS      UPDATED
imagestream.image.openshift.io/dummy-micro-01    image-registry.openshift-image-registry.svc:5000/dummy-estrategia-02b/dummy-micro-01  latest   2 minutes ago
imagestream.image.openshift.io/openjdk8           image-registry.openshift-image-registry.svc:5000/dummy-estrategia-02b/openjdk8        alpine-slim 3 minutes ago

NAME                                HOST/PORT
route.route.openshift.io/dummy-micro-01          dummy-micro-01-dummy-estrategia-02b.cluster-openshift-cla-cp4-ccc03eca20d26e6ac64511f874a64b9b-0000.us-south.containers.appdomain.cloud
```

Luego, se prueba el servicio:

\$ curl http://dummy-micro-01-dummy-estrategia-02b.cluster-openshift-cla-cp4-ccc03eca20d26e6ac64511f874a64b9b-0000.us-south.containers.appdomain.cloud/dummy-micro-01/get/personas

IMPORTANTE: el DNS de la URL en este caso es autogenerado & debe ser actualizado.

```
GPX+000996815@LAPTOP-PF8PU6RQ MINGW64 ~ (master)
$ curl http://dummy-micro-01-dummy-estrategia-01b.cluster-openshift-cla-cp4-ccc03eca20d26e6ac64511f874a64b9b-0000.us-south.containers.appdo
main.cloud/dummy-micro-01/get/personas
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
100    196    100    196     0     0    245     0 --:--:-- --:--:-- --:--:--  447[{ 'nombre': 'PAOLO GUERRERO', 'edad': 35, 'rol': 'CONSULTOR'
},{ 'nombre': 'LUIS GUADALUPE', 'edad': 40, 'rol': 'PROGRAMADOR' }, { 'nombre': 'PEDRO SALAZAR', 'edad': 30, 'rol': 'ARQUITECTO' }]
```

C. DESDE 'GIT-REPOSITORY':

MODALIDAD GRÁFICA:

Esta **ESTRATEGIA** se aplicará cuando el **ORIGEN** es un **REPOSITORIO DE FUENTES** de tipo **GIT** (*Github, GitLab, Bitbucket*) **EXTERNO**.

Se procede a crear un **Namespace/Project** llamado: **dummy-estrategia-03**

Create Project

Name *

dummy-estrategia-03

Display Name

dummy-estrategia-03

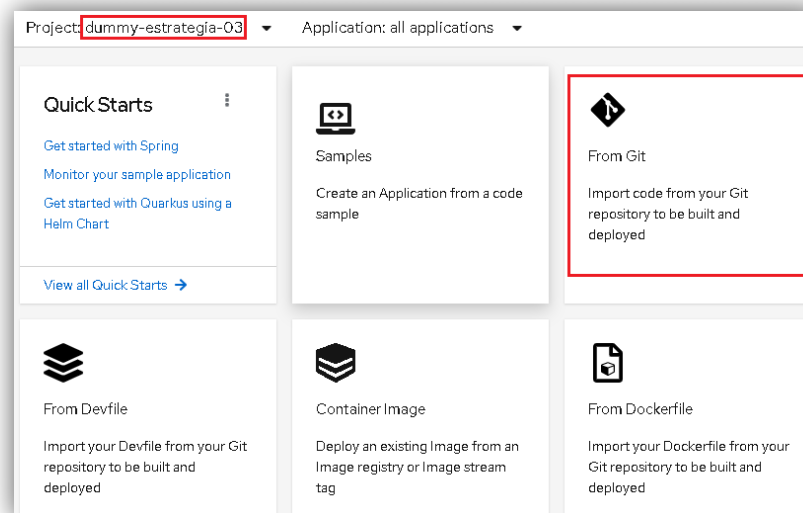
Description

dummy-estrategia-03

Cancel

Create

Luego, se seleccionará la opción de despliegue como:
FROM GIT.



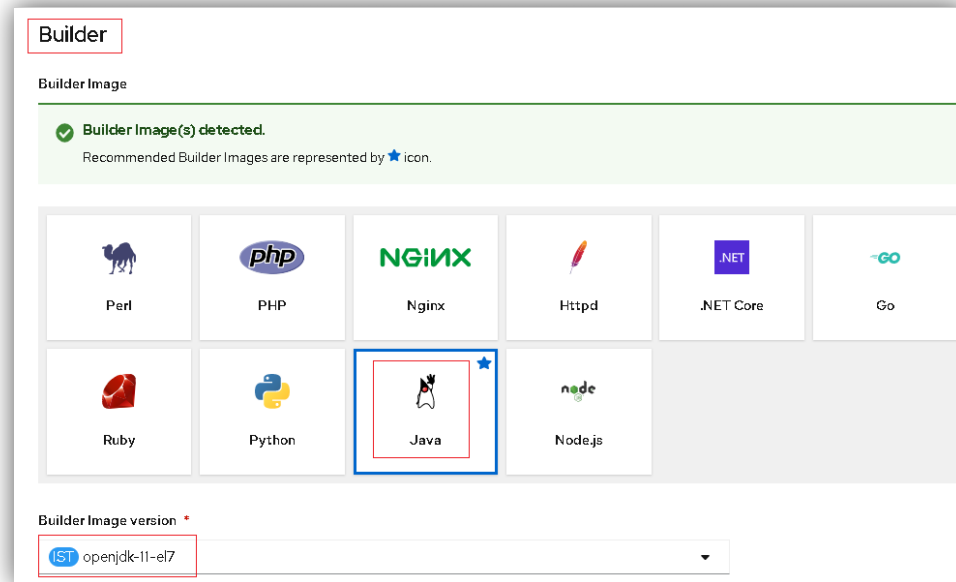
Luego, se debe ingresar la **URL** del **REPOSITORIO GIT** donde esté la fuente del **MICROSERVICIOS** almacenada:
https://github.com/maktup/dummy-micro-01

Así mismo, si existe un **DIRECTORIO** interno donde esté la **FUENTE** ubicada se debe especificar:

- **Git Repo URL:**
https://github.com/maktup/dummy-micro-01
- **Context Dir:**

A screenshot of a 'Git' configuration form. The 'Git Repo URL' field is highlighted with a red box and contains the text 'https://github.com/maktup/dummy-micro-01'. Below it, there's a 'Validated' status. A dropdown menu for 'Git reference' is visible. There's a section for 'Context dir' with a text input field that has a red box around it. To the right of this section, there's a red text note: 'Si existe un directorio interno en el REPOSITORIO GIT EXTERNO.' Below the 'Context dir' section, there's a 'Source Secret' dropdown menu with 'Select Secret name' as the current selection. At the bottom, there's a note: 'Secret with credentials for pulling your source code.'

Luego, automáticamente **OPENSIFT** reconocerá el **BUILDER** en base al **Lenguaje de Programación** del **MICROSERVICIOS** desarrollado. Así como, seleccionará la versión de la **IMAGEN** base identificada.

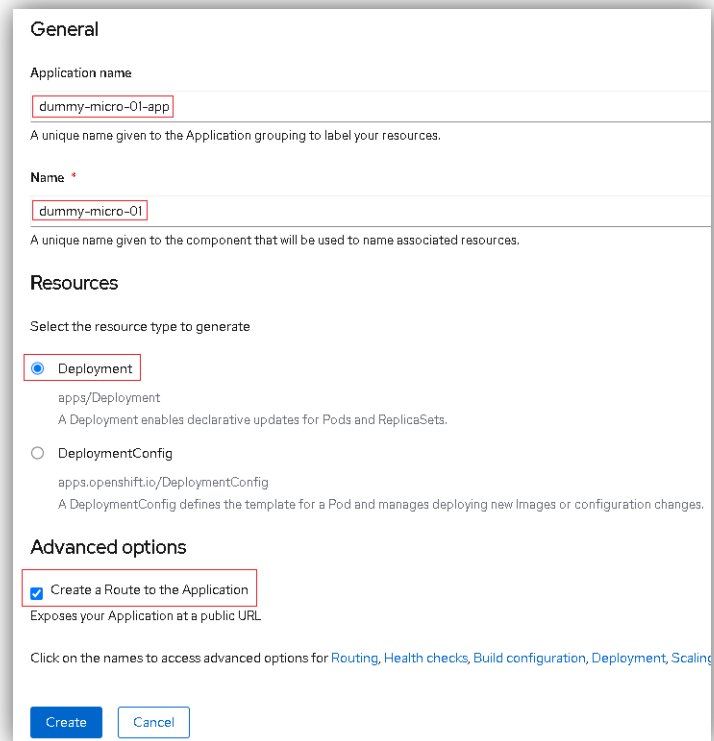


Luego, completar la información del formulario, relacionada a los **RECURSOS** para el **DESPLIEGUE**:

- **Application Name:** dummy-micro-01-app
- **Name:** dummy-micro-01

- ✓ **DEPLOYMENT** => Modalidad **Kubernetes**.
- ✓ **DEPLOYMENTCONFIG** => Modalidad **Openshift**

& dar **check** por si se requiere que dicho **MICROSERVICIO** sea consumido **externamente**. En este caso se le creará un **RECURSO** de tipo **ROUTE**.

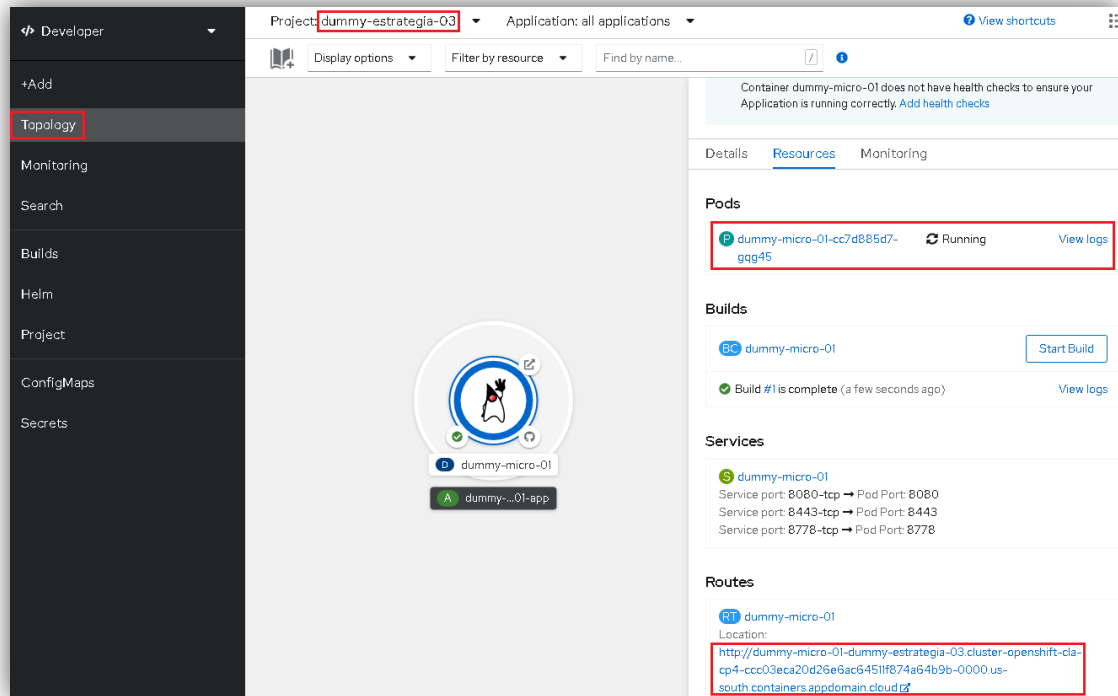


The screenshot shows a 'General' configuration window with the following fields and options:

- Application name:** A text input field containing 'dummy-micro-01-app'.
- Name:** A text input field containing 'dummy-micro-01'.
- Resources:** A section titled 'Select the resource type to generate' with two radio button options:
 - Deployment:** Selected. Description: 'apps/Deployment. A Deployment enables declarative updates for Pods and ReplicaSets.'
 - DeploymentConfig:** Unselected. Description: 'apps.openshift.io/DeploymentConfig. A DeploymentConfig defines the template for a Pod and manages deploying new Images or configuration changes.'
- Advanced options:** A section with a checked checkbox 'Create a Route to the Application'. Description: 'Exposes your Application at a public URL.'

At the bottom, there are 'Create' and 'Cancel' buttons.

Luego, se verifica que **GRÁFICAMENTE** se muestran los diferentes **RECURSOS** creados en **OPENSIFT**, así mismo, al darle click se detalla a la **DERECHA** opciones como para ingresar al **POD**, a los **LOGs** del **CONTENEDOR**, acceso a la **URL** base del **MICROSERVICIO** desplegado, etc.

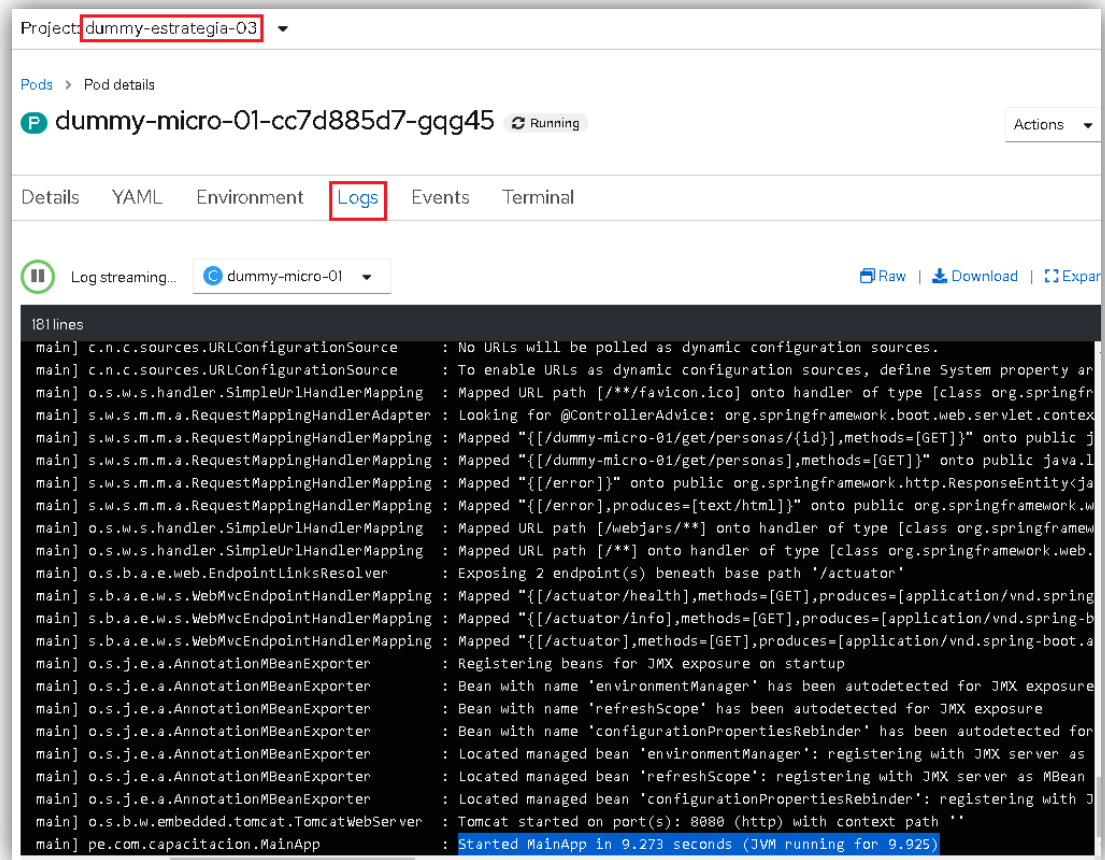


Finalmente, se verifican los **LOGs** del **CONTENEDOR** dentro del **POD** & se valida que el **MICROSERVICIO** esté ya iniciado.

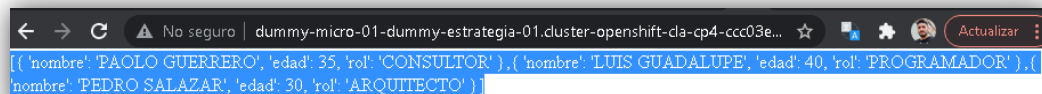
Luego se accede a la **URL** del **MICROSERVICIO** concatenándole la **URI** que maneja:

```
$ curl http://dummy-micro-01-dummy-estrategia-01.cluster-openshift-cla-cp4-ccc03eca20d26e6ac64511f874a64b9b-0000.us-south.containers.appdomain.cloud/dummy-micro-01/get/personas
```

IMPORTANTE: el **DNS** de la **URL** en este caso es autogenerado & debe ser actualizado.



```
Project: dummy-estrategia-03
Pod details
dummy-micro-01-cc7d885d7-gqq45 Running
Details YAML Environment Logs Events Terminal
Log streaming... dummy-micro-01
181 lines
main] c.n.c.sources.URLConfigurationSource : No URLs will be polled as dynamic configuration sources.
main] c.n.c.sources.URLConfigurationSource : To enable URLs as dynamic configuration sources, define System property an
main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**/favicon.ico] onto handler of type [class org.springfr
main] s.w.s.m.m.a.RequestMappingHandlerAdapter : Looking for @ControllerAdvice: org.springframework.boot.web.servlet.context
main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/dummy-micro-01/get/personas/{id}],methods=[GET]}" onto public j
main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/dummy-micro-01/get/personas],methods=[GET]}" onto public java.l
main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error]}" onto public org.springframework.http.ResponseEntity<ja
main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error],produces=[text/html]}" onto public org.springframework.w
main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/webjars/**] onto handler of type [class org.springframework
main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**] onto handler of type [class org.springframework.web.
main] o.s.b.a.e.web.EndpointLinksResolver : Exposing 2 endpoint(s) beneath base path '/actuator'
main] s.b.a.e.w.s.WebMvcEndpointHandlerMapping : Mapped "{[/actuator/health],methods=[GET],produces=[application/vnd.spring
main] s.b.a.e.w.s.WebMvcEndpointHandlerMapping : Mapped "{[/actuator/info],methods=[GET],produces=[application/vnd.spring-b
main] s.b.a.e.w.s.WebMvcEndpointHandlerMapping : Mapped "{[/actuator],methods=[GET],produces=[application/vnd.spring-boot.a
main] o.s.j.e.a.AnnotationMBeanExporter : Registering beans for JMX exposure on startup
main] o.s.j.e.a.AnnotationMBeanExporter : Bean with name 'environmentManager' has been autodetected for JMX exposure
main] o.s.j.e.a.AnnotationMBeanExporter : Bean with name 'refreshScope' has been autodetected for JMX exposure
main] o.s.j.e.a.AnnotationMBeanExporter : Bean with name 'configurationPropertiesRebinder' has been autodetected for
main] o.s.j.e.a.AnnotationMBeanExporter : Located managed bean 'environmentManager': registering with JMX server as
main] o.s.j.e.a.AnnotationMBeanExporter : Located managed bean 'refreshScope': registering with JMX server as MBean
main] o.s.j.e.a.AnnotationMBeanExporter : Located managed bean 'configurationPropertiesRebinder': registering with J
main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
main] pe.com.capacitacion.MainApp : Started MainApp in 9.273 seconds (JVM running for 9.925)
```



```
{('nombre': 'PAOLO GUERRERO', 'edad': 35, 'rol': 'CONSULTOR'), ('nombre': 'LUIS GUADALUPE', 'edad': 40, 'rol': 'PROGRAMADOR'), ('nombre': 'PEDRO SALAZAR', 'edad': 30, 'rol': 'ARQUITECTO') }
```

MODALIDAD POR CÓDIGO:

Esta modalidad realiza exactamente lo mismo que la modalidad anterior, solo que por **CÓDIGO**:

Aquí se trabaja en base a **BUILDERS**, que son componentes que identifican el **TIPO** de fuente existente en un **REPOSITORIO** & en base a ello, se crean los **RECURSOS** & los despliegues requeridos:

- ✓ **BUILDER (JAVA)** => **java**
- ✓ **BUILDER (PYTHON)** => **python**

Ingresar el **COMANDO** siguiente:

PATRÓN: **BUILDER + Repositorio de Fuentes**

```
$ oc create ns dummy-estrategia-03b
$ oc new-app java~https://github.com/maktup/dummy-micro-01 --name=dummy-micro-01 --strategy=source -n dummy-estrategia-03b
```

Luego, se expone & consulta ejecutando el **COMANDO**:

```
$ oc expose service dummy-micro-01 -n dummy-estrategia-03b
$ oc get all -n dummy-estrategia-03b
```

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc create ns dummy-estrategia-03b
namespace/dummy-estrategia-03b created

GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc new-app java~https://github.com/maktup/dummy-micro-01 --name=dummy-micro-01 --strategy=source -n dummy-estrategia-03b
--> [Found image 224e8f2 (2 months old) in image stream "openshift/java" under tag "openjdk-11-ubi8" for "java"]

Java Applications
-----
Platform for building and running plain Java applications (fat-jar and flat classpath)

Tags: builder, java

* A source build using source code from https://github.com/maktup/dummy-micro-01 will be created
* The resulting image will be pushed to image stream tag "dummy-micro-01:latest"
* Use 'start-build' to trigger a new build
* This image will be deployed in deployment config "dummy-micro-01"
* Ports 8080/tcp, 8443/tcp, 8778/tcp will be load balanced by service "dummy-micro-01"
* Other containers can access this service through the hostname "dummy-micro-01"

--> Creating resources ...
imagestream.image.openshift.io "dummy-micro-01" created
buildconfig.build.openshift.io "dummy-micro-01" created
deploymentconfig.apps.openshift.io "dummy-micro-01" created
service "dummy-micro-01" created
--> Success
Build scheduled, use 'oc logs -f bc/dummy-micro-01' to track its progress.
Application is not exposed. You can expose services to the outside world by executing one or more of the commands below:
'oc expose svc/dummy-micro-01'
Run 'oc status' to view your app.
```

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc expose service dummy-micro-01 -n dummy-estrategia-03b
route.route.openshift.io/dummy-micro-01 exposed

GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc get all -n dummy-estrategia-03b
NAME READY STATUS RESTARTS AGE
pod/dummy-micro-01-1-build 1/1 Running 0 32s

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
service/dummy-micro-01 ClusterIP 172.21.159.88 <none> 8080/TCP,8443/TCP,8778/TCP 33s

NAME REVISION DESIRED CURRENT TRIGGERED BY
deploymentconfig.apps.openshift.io/dummy-micro-01 0 1 0 config,image(dummy-micro-01:latest)

NAME TYPE FROM LATEST
buildconfig.build.openshift.io/dummy-micro-01 Source Git 1

NAME TYPE FROM STATUS STARTED DURATION
build.build.openshift.io/dummy-micro-01-1 Source Git@12dc7d5 Running 34 seconds ago

NAME IMAGE REPOSITORY TAGS UPDATED
imagestream.image.openshift.io/dummy-micro-01 image-registry.openshift-image-registry.svc:5000/dummy-estrategia-03b/dummy-micro-01

NAME HOST/PORT
route.route.openshift.io/dummy-micro-01 dummy-micro-01-dummy-estrategia-03b.cluster-openshift-cla-cp4-ccc03eca20d26e6ac6451f874a64b9b-0000.us-south.containers.appdomain.cloud
```


Luego, se prueba el servicio:

\$ curl http://dummy-micro-01-dummy-estrategia-01b.cluster-openshift-cla-cp4-ccc03eca20d26e6ac64511f874a64b9b-0000.us-south.containers.appdomain.cloud/d

IMPORTANTE: el **DNS** de la **URL** en este caso es autogenerated & debe ser actualizado.

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ curl http://dummy-micro-01-dummy-estrategia-03b.cluster-openshift-cla-cp4-ccc03eca20d26e6ac64511f874a64b9b-0000.us-south.containers.appdomain.cloud/dummy-micro-01/get/personas
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 196 100 196 0 0 245 0 --:--:-- --:--:-- --:--:-- 464[{ 'nombre': 'PAOLO GUERRERO', 'edad': 35, 'rol': 'CONSULTOR' }, { 'nombre': 'LUIS GUADALUPE', 'edad': 40, 'rol': 'PROGRAMADOR' }, { 'nombre': 'PEDRO SALAZAR', 'edad': 30, 'rol': 'ARQUITECTO' } ]
```

D. ELIMINACIÓN DE RECURSOS:

Finalmente, para **ELIMINAR** todos los **RECURSOS** creados, ejecutar los **COMANDOS** siguiente:

\$ oc delete ns dummy-estrategia-01 --force --grace-period 0
\$ oc delete ns dummy-estrategia-02 --force --grace-period 0
\$ oc delete ns dummy-estrategia-02b --force --grace-period 0
\$ oc delete ns dummy-estrategia-03 --force --grace-period 0
\$ oc delete ns dummy-estrategia-03b --force --grace-period 0

IMPORTANTE: al eliminar los **RECURSOS: Namespace / Project** automáticamente sus **RECURSOS** internos serán también **ELIMINADOS**.

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc delete ns dummy-estrategia-01 --force --grace-period 0
warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resource may continue to run on the cluster indefinitely.
namespace "dummy-estrategia-01" force deleted

GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc delete ns dummy-estrategia-02 --force --grace-period 0
warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resource may continue to run on the cluster indefinitely.
namespace "dummy-estrategia-02" force deleted

GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc delete ns dummy-estrategia-02b --force --grace-period 0
warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resource may continue to run on the cluster indefinitely.
namespace "dummy-estrategia-02b" force deleted

GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc delete ns dummy-estrategia-03 --force --grace-period 0
warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resource may continue to run on the cluster indefinitely.
namespace "dummy-estrategia-03" force deleted

GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc delete ns dummy-estrategia-03b --force --grace-period 0
warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resource may continue to run on the cluster indefinitely.
namespace "dummy-estrategia-03b" force deleted
```