

MÓDULO 7

Implementación de aplicaciones Simples

EXPLORACIÓN DE LA CONSOLA WEB DE RED HAT OPENSIFT

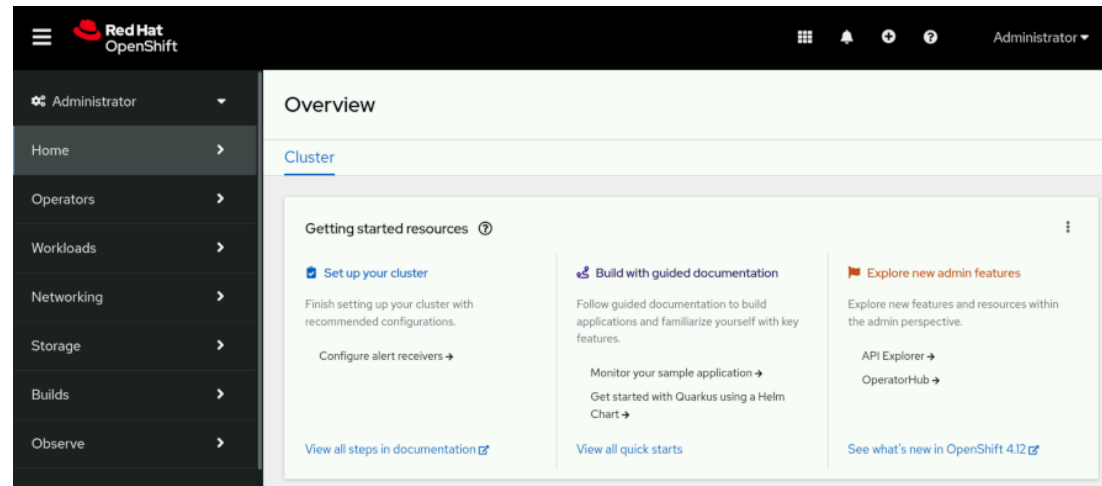


ACCESO A LA CONSOLA WEB DE OPENS SHIFT

- A la consola web se puede acceder mediante cualquier explorador web moderno.
- La URL se puede configurar y se encuentra mediante la CLI oc.

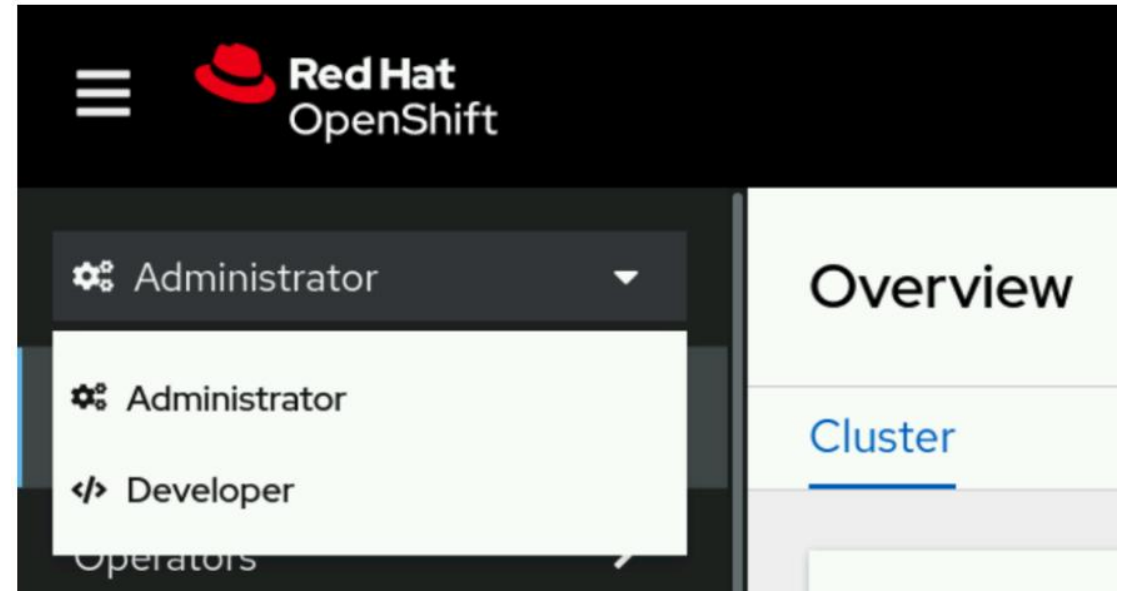
```
[user@host ~]$ oc whoami --show-console  
https://console-openshift-console.apps.ocp4.example.com
```

- Con un explorador web para navegar a la URL, que muestra la página de autenticación.
- El uso de las credenciales para el acceso al clúster lo lleva a la página de inicio de la consola web.



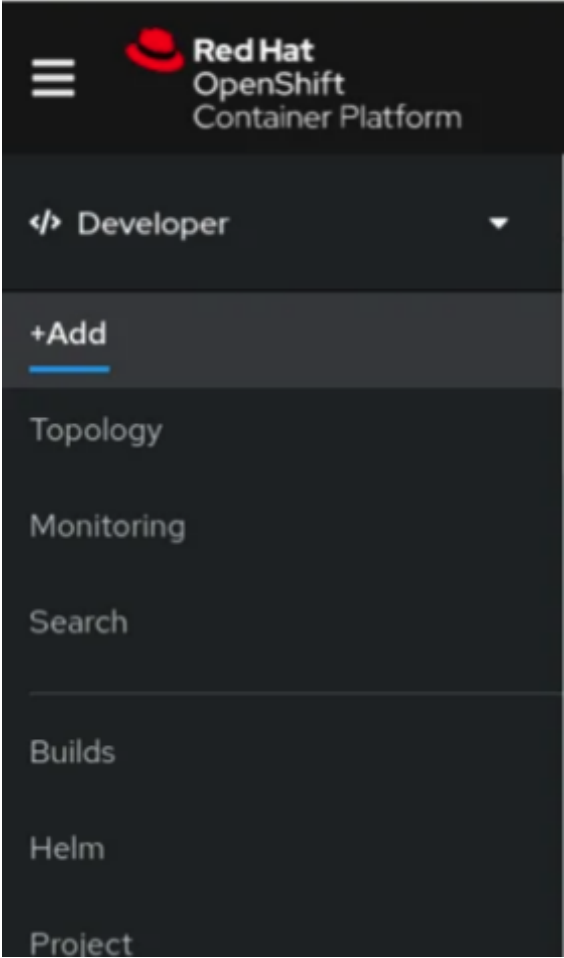
PERSPECTIVAS DE LA CONSOLA WEB

- Dentro de la consola web, el medio principal de navegación es la barra lateral de la izquierda.
- La barra lateral organiza las funciones y las tareas del clúster en varias categorías principales.
- El primer menú de selección en el menú de la barra lateral cambia entre las perspectivas Administrator y Developer.



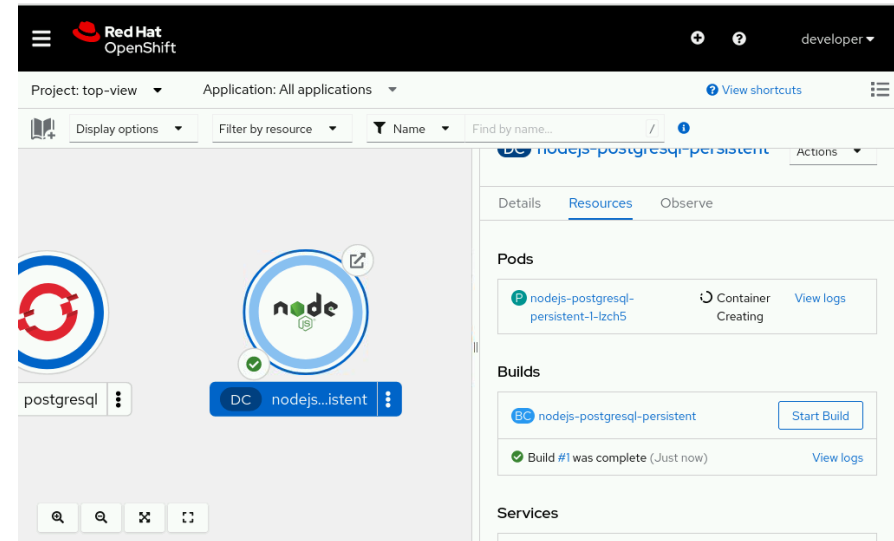
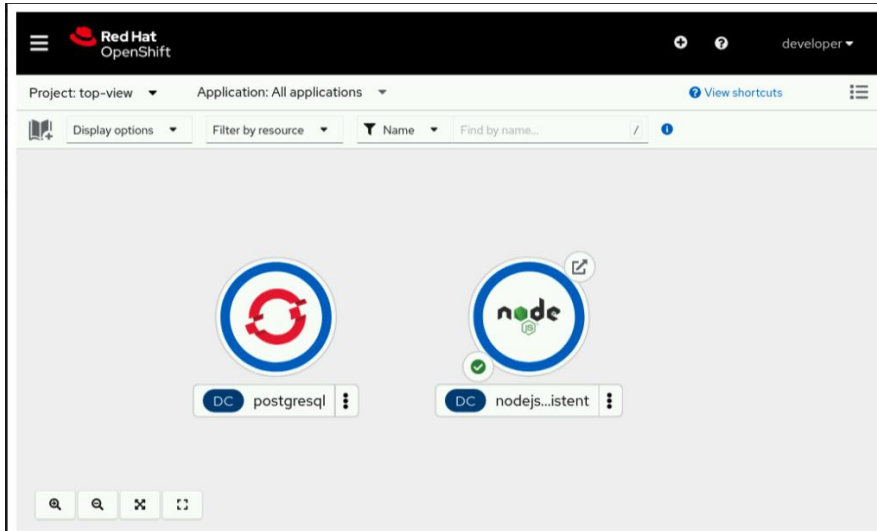
PERSPECTIVA DE DESARROLLADOR

Nombre de la sección	Usos
Topology	Explore de forma interactiva los recursos del proyecto de descripción general de alto nivel.
Observe	Vea información de alto nivel sobre la utilización de recursos del proyecto, como la CPU, la memoria, la red y los límites de almacenamiento.
Search	Construya consultas para buscar recursos según el tipo de recurso y las etiquetas.
Builds	Vea y administre recursos <code>BuildConfig</code> en el proyecto seleccionado.
Helm	Enumere los charts de Helm instalados y agregue gráficos del catálogo de desarrolladores.
Project	Vea una combinación de información relevante para el proyecto seleccionado, como el uso de recursos del clúster y los eventos.
Config Maps	Vea y administre recursos <code>ConfigMap</code> en el proyecto seleccionado.
Secrets	Vea y administre recursos <code>Secret</code> en el proyecto seleccionado.



VISTA DE TOPOLOGÍA

- Vista de topología de la consola web de OpenShift:
- Panel de detalles de la vista de topología:

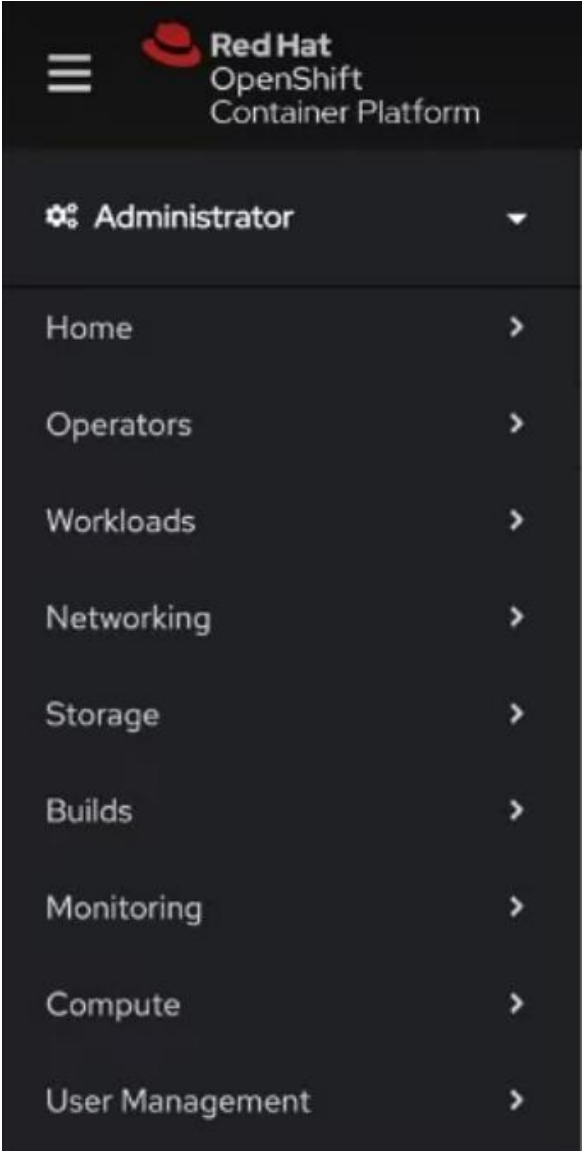


- El indicador de estado de compilación muestra que la compilación más reciente se realizó correctamente:



PERSPECTIVA DE ADMINISTRADOR

Nombre de la sección	Usos
Home	Vea una combinación de información relevante para el proyecto seleccionado, como el uso de recursos del clúster y los eventos.
Operators	Instale y administre operadores de clúster, que amplían las características y funciones de su clúster.
Workloads	Administre recursos dentro de Workload API, que incluye Pods, Deployments, Jobs y otros.
Networking	Administre recursos de red, como Services y Routes, que se utilizan en conjunto para exponer aplicaciones interna y externamente.
Storage	Administre los recursos de almacenamiento, como StorageClasses y PersistentVolumeClaims, que usan los pods para solicitar almacenamiento persistente.
Builds	Vea y administre recursos BuildConfig en el proyecto seleccionado.
User Management	Administre usuarios y asigne roles dentro del clúster.
Administration	Administre ResourceQuotas y LimitRanges.



TIPOS DE RECURSOS

- Con muchas de las secciones en ambas perspectivas, puede navegar más profundamente en las páginas para explorar diversos tipos de recursos.

The screenshot displays the Red Hat OpenShift console interface. At the top, the header shows the Red Hat OpenShift logo, a hamburger menu, and the user 'developer'. Below the header, the breadcrumb navigation indicates 'Project: top-view' and 'Builds > Build details'. The main section shows the build 'nodejs-postgresql-persistent-1' with a 'Complete' status. A tabbed interface below the build name includes 'Details', 'Metrics', 'YAML', 'Environment', 'Logs' (which is selected), and 'Events'. The 'Logs' section shows a 'Log stream ended' message and a search bar. On the right, there are options to 'Wrap lines', 'Raw', 'Download', and 'Expand'. The log content, consisting of 99 lines, shows the successful completion of a build process, including copying blobs, writing the manifest, and pushing the image to the registry.

```
91 Copying blob sha256:c3532032c23424d6d4c608c7a50d4d4cee6f0ea0ebe65cbec4f54fc27839ec8b
92 Copying blob sha256:6208c5a2e205726f3a2cd42a392c5e4f05256850d13197a711000c4021ede87b
93 Copying blob sha256:d6ae08e45d969b916bbc145b18f1a5a261e573ce634b0159862b9145bb3fefe9
94 Copying blob sha256:87b6121ef647e82c2efa8e6489d94c7668d88af38c138236592c6675acdf055a
95 Copying config sha256:bbc720ae3d138fff39c5950ee07721ffb95e0fa9cd76f2f78bac6e8475977882
96 Writing manifest to image destination
97 Storing signatures
98 Successfully pushed image-registry.openshift-image-registry.svc:5000/top-view/nodejs-postgresql-persistent@sha25
99 Push successful
```


IMPLEMENTACIÓN DE APLICACIONES MEDIANTE LA CONSOLA WEB RED HAT OPENSIFT



IMPLEMENTACIÓN DE APLICACIONES CON LA PERSPECTIVA DEL DESARROLLADOR



IMPLEMENTACIÓN DE APLICACIONES CON LA PERSPECTIVA DEL DESARROLLADOR

- La mayoría de los elementos enumerados en la pantalla **Add** desencadenan un proceso de implementación, que incluye una vista de formulario para establecer los valores de configuración de la aplicación o el servicio que desea implementa

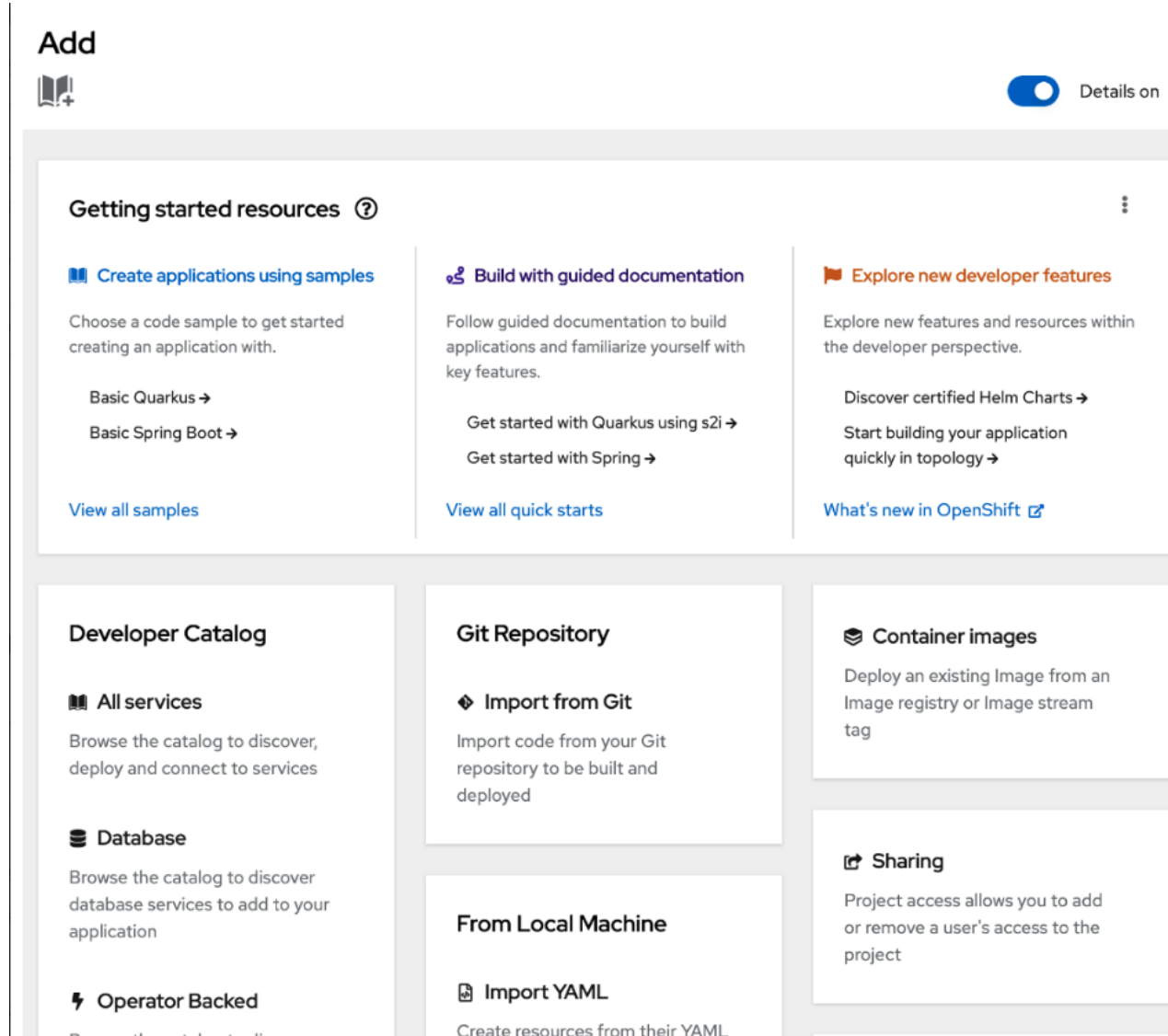
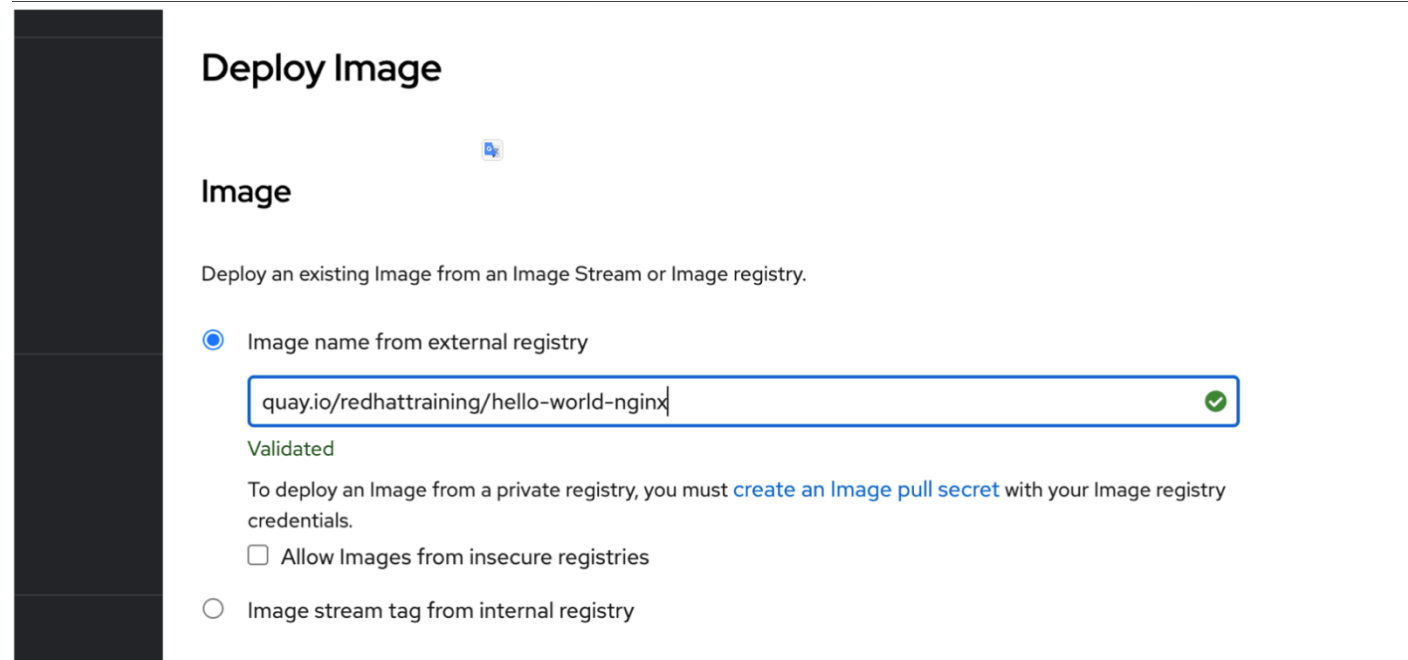



IMAGEN DE CONTENEDOR

- Este método crea una aplicación en el clúster mediante la implementación de una imagen de contenedor existente.
- Para implementar una aplicación desde una imagen de contenedor, abra la pantalla **Add** y luego haga clic en **Container Image**, que abre la pantalla **Deploy Image** correspondiente.




Deploy Image



Image

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

☒ Image name from external registry



Validated

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

☐ Allow Images from insecure registries

☐ Image stream tag from internal registry



IMPORTAR DESDE GIT

- Este método crea una aplicación en el clúster mediante la importación del código fuente de un repositorio de Git, la creación de la imagen del contenedor y la implementación de la imagen compilada.
- Para implementar una aplicación con este método, abra la pantalla **Add** y luego haga clic en **Import from Git**. En la pantalla **Import from Git**, ingrese la URL del repositorio de Git.
- Red Hat OpenShift puede crear automáticamente una imagen de contenedor a partir del código fuente.

Import from Git

Git

Git Repo URL *

Validated

▼ Hide advanced Git options

Git reference

Optional branch, tag, or commit.

Context dir

Optional subdirectory for the source code, used as a context directory for build.

Source Secret

Secret with credentials for pulling your source code.



Builder Image detected.

A Builder Image is recommended.



Red Hat OpenJDK 17 (UBI 8)

[Edit Import Strategy](#)

BUILDER JAVA OPENJDK

Build and run Java applications using Maven and OpenJDK 17.

Sample repository: <https://github.com/jboss-openshift/openshift-quickstarts>

ESTRATEGIAS DE IMPORTACION

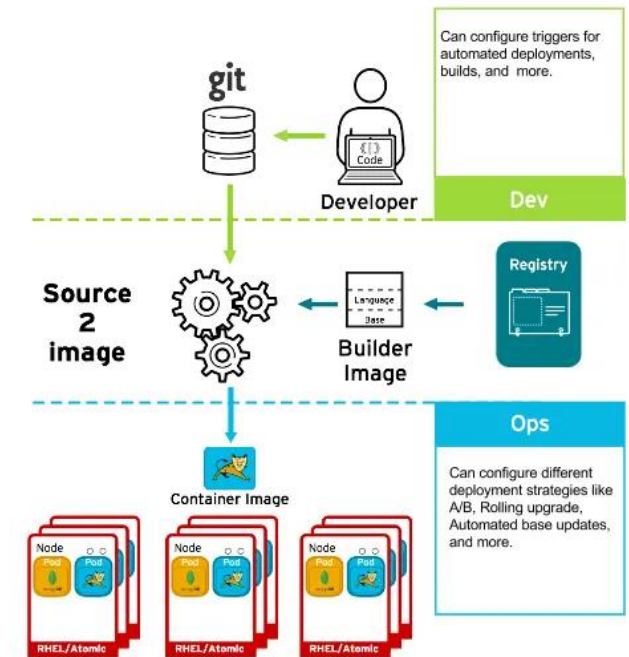
Según el contenido del repositorio, el clúster selecciona una de las siguientes estrategias de importación.

- **Devfile:** El clúster usa un *Devfile* como receta para compilar e implementar la aplicación. Un Devfile es un archivo YAML que define entornos de aplicaciones nativas de la nube contenerizadas.
- **Dockerfile:** El clúster usa un Containerfile para compilar la imagen de la aplicación.
- **Imagen del compilador:** El clúster usa una imagen del compilador para crear la imagen de la aplicación, mediante un proceso denominado *Source-to-Image (S2I)*.

Code

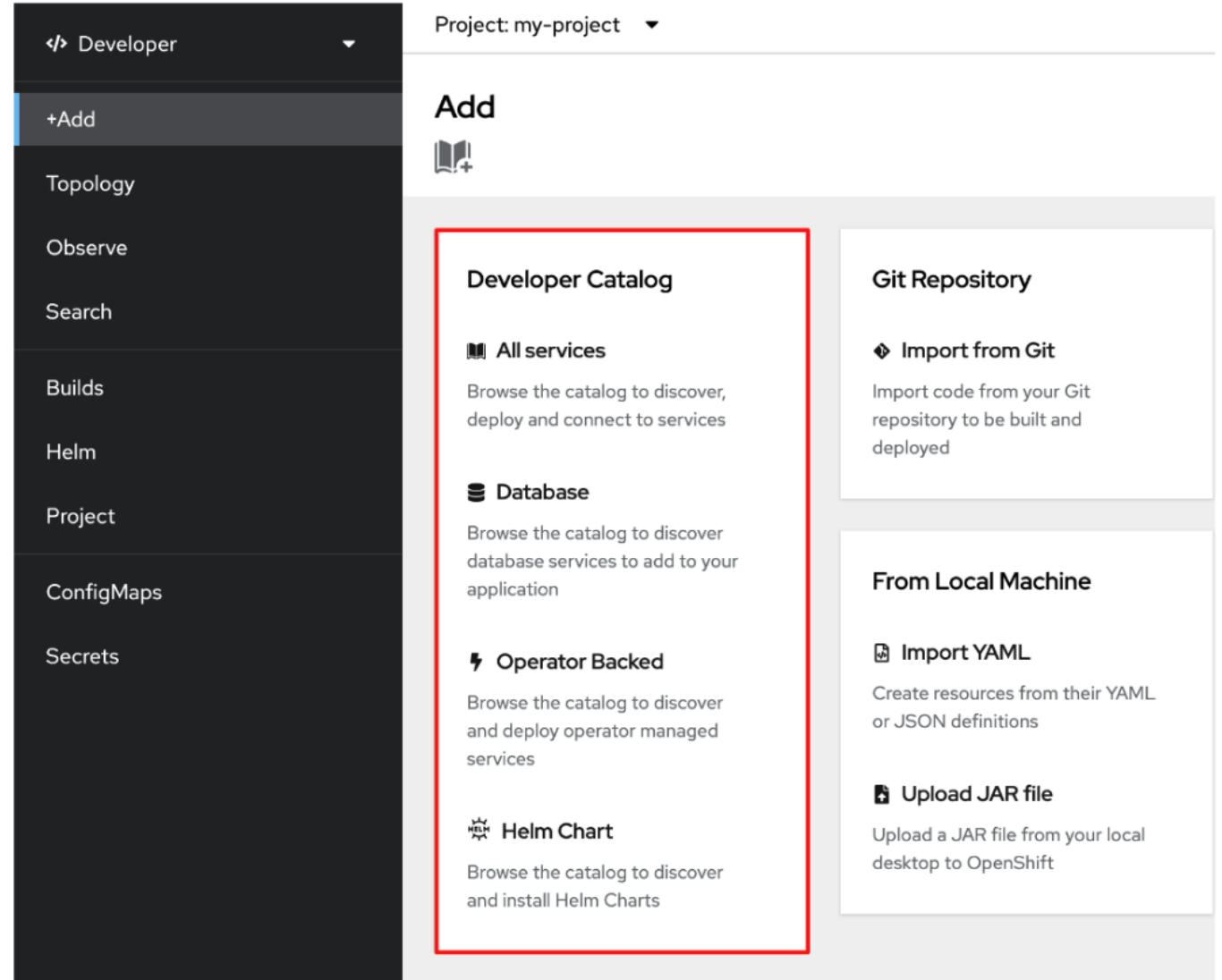
Build

Deploy



DESDE EL CATÁLOGO DE DESARROLLADORES

- El catálogo para desarrolladores ofrece un conjunto de recetas seleccionadas que puede usar para implementar aplicaciones y servicios
- Además de proporcionar accesos directos a formularios preconfigurados para implementar desde repositorios e imágenes, el catálogo también incluye servicios y software adicional, como bases de datos
- Para acceder al catálogo de desarrolladores, abra la pantalla **Add** y luego haga clic en cualquiera de los ítems debajo de la tarjeta **Developer Catalog**.



CATEGORIAS DEL CATALOGO DE DESARROLLADORES

- Tipos de importación disponibles:
 - Templates
 - Builder Images
 - Helm Charts
 - **Operator Backed**

Developer Catalog

Add shared applications, services, event sources, or source-to-image builders to your Project from the developer catalog. Cluster administrators can customize the content made available in the catalog.

All items

CI/CD

Databases

Languages

Middleware

Other

Type ⓘ

Builder Images (12)

Devfiles (6)


Helm Charts (69)

Templates (45)

All items


A-Z ▾

132 items

Builder Images


.NET
Provided by Red Hat

Build and run .NET 7 applications on UBI 8. For more information about using this builder image,...

Helm Charts


.NET
Provided by Red Hat

A Helm chart to build and deploy .NET applications

Helm Charts

a10networks-a10tkc ✓

A Helm chart for A10 Thunder Kubernetes Connector

Templates

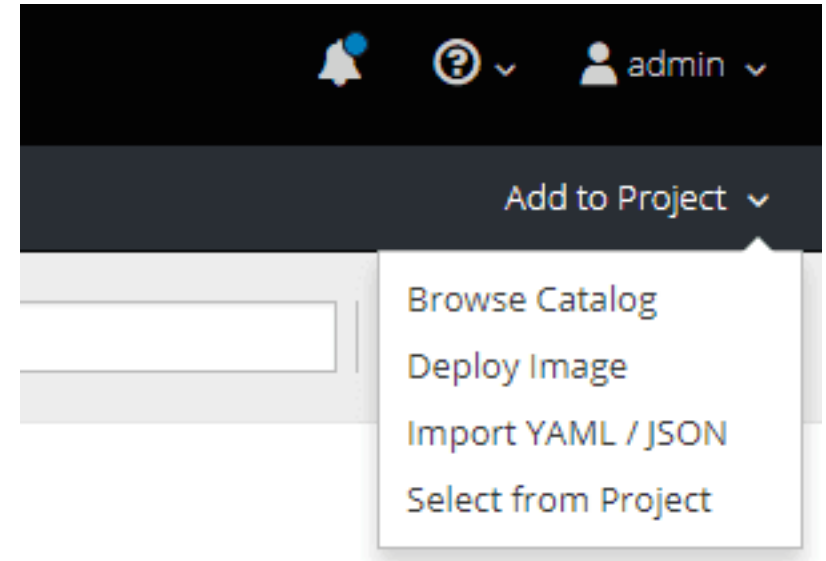
Apache HTTP Server
Provided by Red Hat, Inc.

An example Apache HTTP Server



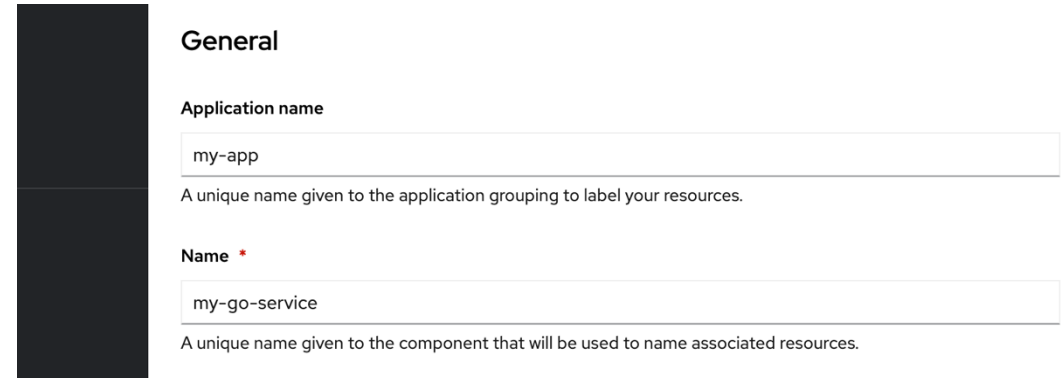
IMPORTACIÓN DESDE LA MAQUINA LOCAL

- Puede implementar aplicaciones importando archivos JAR y manifiestos YAML y JSON directamente en OpenShift.
- Si desea importar un manifiesto, abra la pantalla **Add** y haga clic en **Import YAML**, que abre un editor de texto.
- Si deseas implementar un archivo JAR desde su computadora local, haga clic en **Upload JAR file** en la pantalla **Add**.



PROPIEDADES DE CONFIGURACIÓN COMUNES Y AVANZADAS

- cuando implementa una aplicación, puede definir el nombre de la aplicación y el nombre usado para los recursos asociados
- Si se necesita un control más preciso, se puede personalizar los parámetros en la sección **Advanced options**



General

Application name

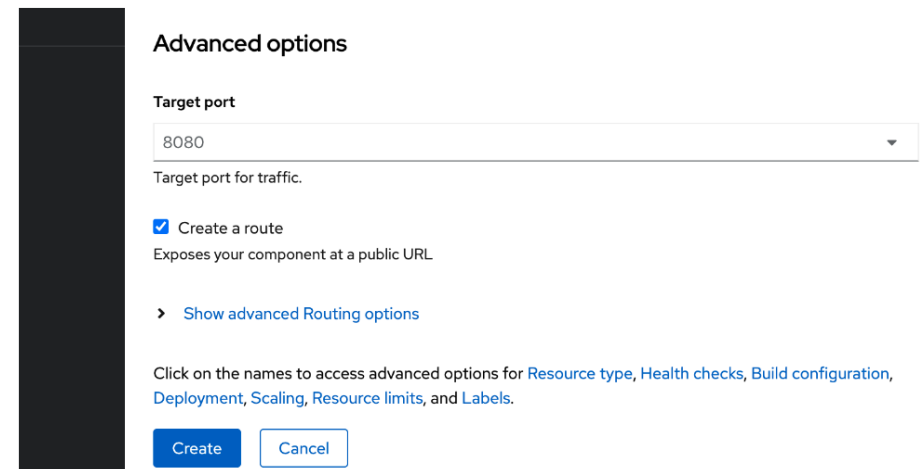
my-app

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

Name *

my-go-service

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



Advanced options

Target port

8080

Target port for traffic.

☒ Create a route

Exposes your component at a public URL

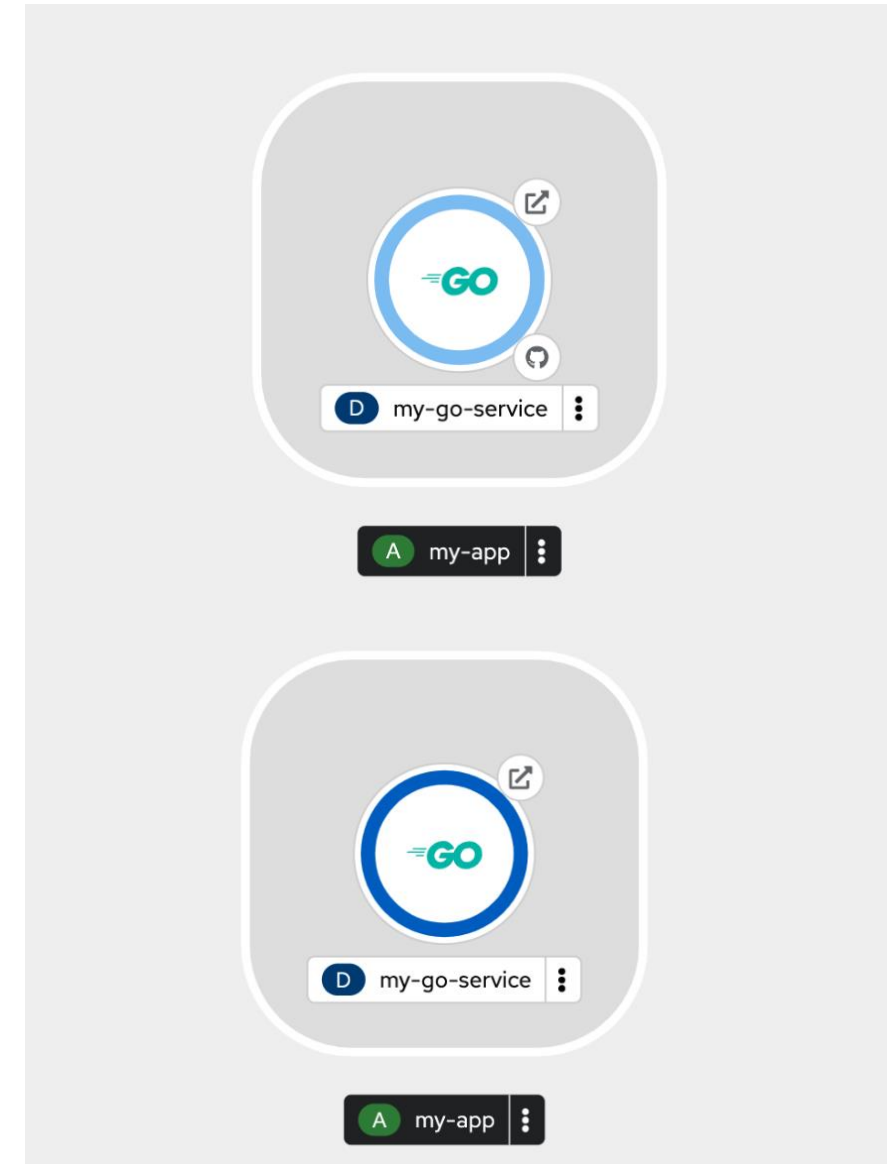
› [Show advanced Routing options](#)

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

Create Cancel

TOPOLOGY (TOPOLOGÍA)

- Después de crear una nueva aplicación con cualquiera de los métodos anteriores, la consola web normalmente lo redirige a la pantalla **Topology**.
- En esta página se puede ver una vista de gráfico que muestra algunos de los recursos de Kubernetes y OpenShift creados cuando implementa una aplicación, como los objetos de implementación.
- Inmediatamente después de la creación de una aplicación, en la pantalla de topología se muestra un borde circular azul claro del ícono de la aplicación.
- El color azul claro indica que el clúster aún está en el proceso de preparar la aplicación y oscuro que los pods de la aplicación se están ejecutando .

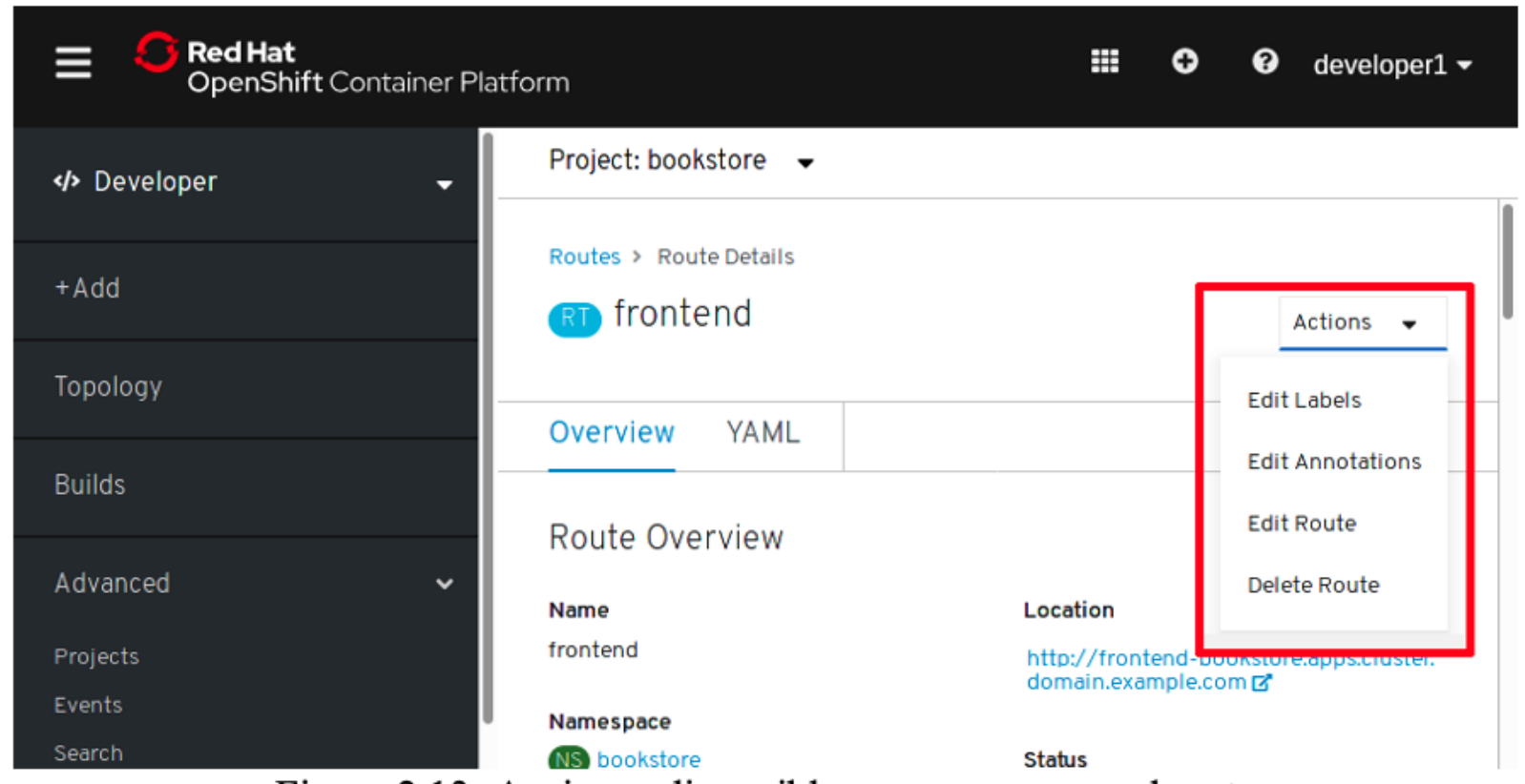


LAB6.2



EDICION DE RECURSOS OPENS SHIFT

- En la mayoría de las páginas de detalles de recursos de la consola web de OpenShift, se proporciona un botón **Action** (Acción) que muestra un menú.

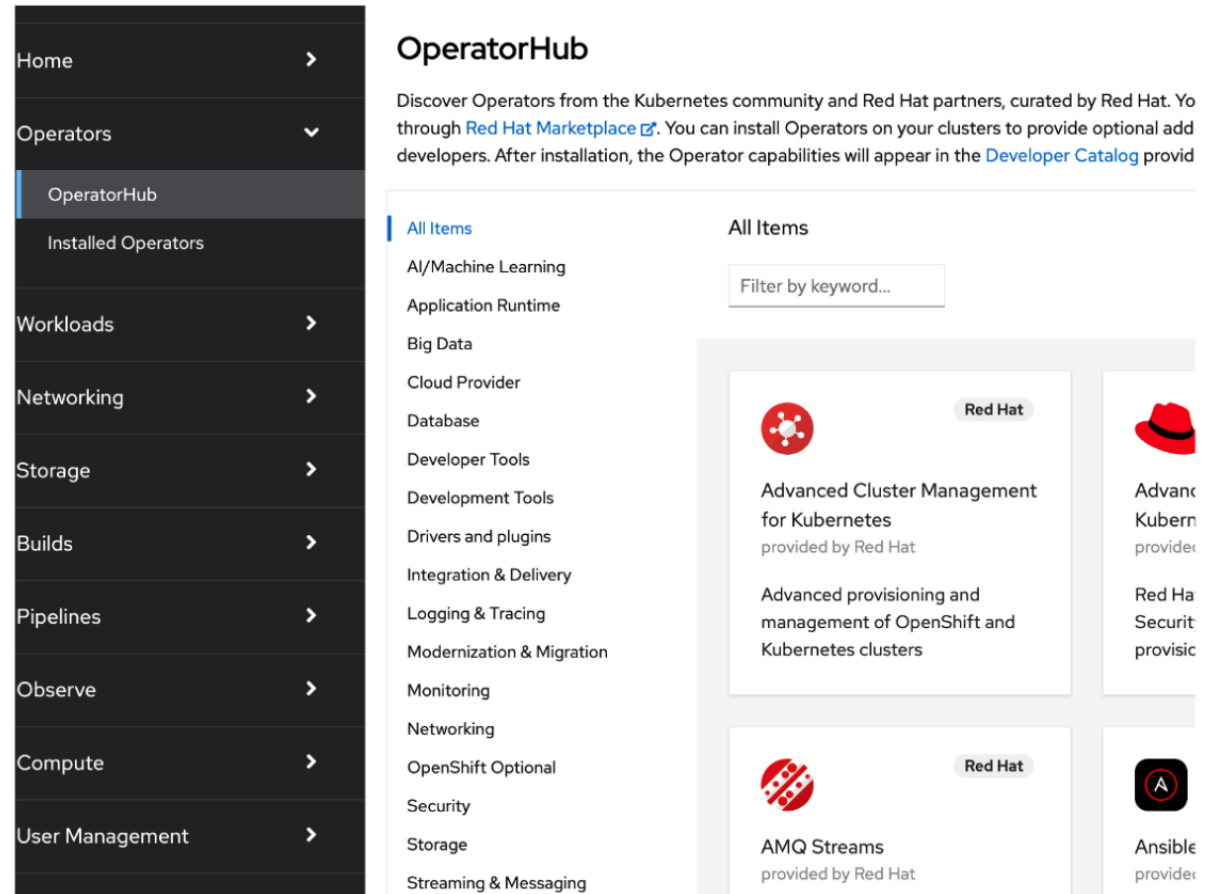


IMPLEMENTACIÓN DE APLICACIONES CON LA PERSPECTIVA DEL ADMINISTRADOR



OPERADORES Y OPERATORHUB

- La instalación de Operadores es una forma útil de proporcionar a los desarrolladores software y servicios adicionales.
- Los operadores son particularmente interesantes cuando un equipo debe agregar capacidades de software complejas a su pila, especialmente si estas capacidades requieren un esfuerzo operativo moderado.
- *OperatorHub* es una forma de descubrir las aplicaciones de Kubernetes que están disponibles para instalar en Red Hat OpenShift.
- Para abrir OperatorHub, desde la perspectiva del administrador, haga clic en **Operators** → **OperatorHub**.



The screenshot displays the OperatorHub interface. On the left is a dark navigation sidebar with the following menu items: Home, Operators (expanded), OperatorHub (selected), Installed Operators, Workloads, Networking, Storage, Builds, Pipelines, Observe, Compute, and User Management. The main content area is titled 'OperatorHub' and includes a descriptive paragraph about discovering operators from the Kubernetes community and Red Hat partners. Below this is a list of categories such as AI/Machine Learning, Application Runtime, Big Data, Cloud Provider, Database, Developer Tools, Development Tools, Drivers and plugins, Integration & Delivery, Logging & Tracing, Modernization & Migration, Monitoring, Networking, OpenShift Optional, Security, Storage, and Streaming & Messaging. A search bar labeled 'Filter by keyword...' is present. The main content area displays a grid of operator cards. Two cards are visible: 'Advanced Cluster Management for Kubernetes' provided by Red Hat, and 'AMQ Streams' provided by Red Hat. Each card includes a Red Hat logo and a brief description of the operator's capabilities.

IMPLEMENTACIÓN DE APLICACIONES MEDIANTE EL USO DE CLI OC



IMPLEMENTACION DE APLICACIONES CON LA CLI DE OPENSIFT

- La CLI de Red Hat OpenShift, **oc**, admite diferentes formas de implementar aplicaciones en un clúster.
- Puedes usar el comando **oc new-app** que crea los recursos requeridos para compilar e implementar una aplicación en OpenShift
- Para aplicaciones complejas, el comando **oc apply** tiene la opción **-f**, que proporciona un archivo que contiene todos los recursos de Kubernetes que su aplicación requiere.

```
[user@host ~]$ oc apply -f YAML_FILE
```



OPCIONES SOPORTADAS CON OC NEW-APP

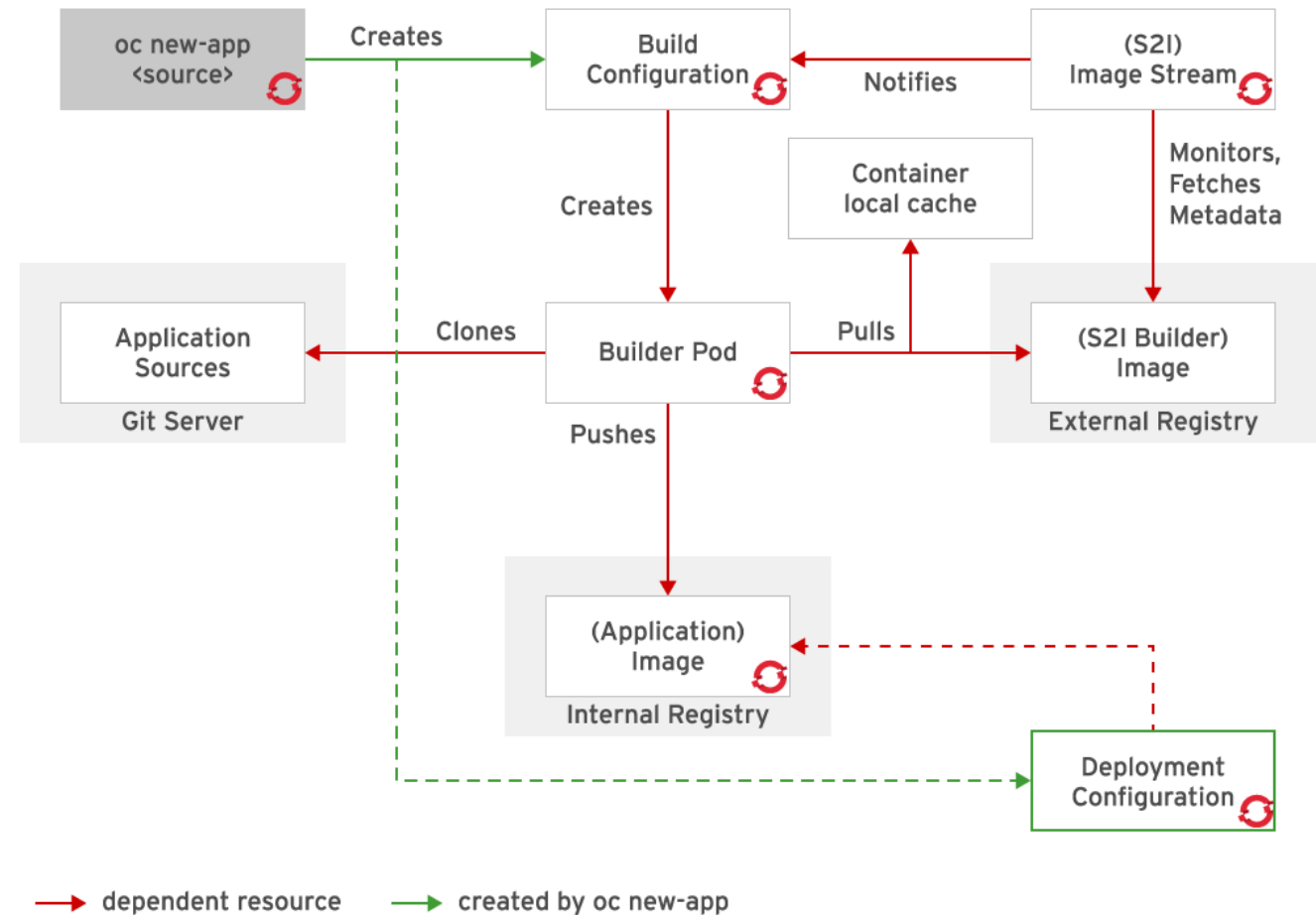
El comando `oc new-app` proporciona las siguientes opciones para personalizar la compilación de la aplicación:

Opción	Descripción
<code>--as-deployment-config</code>	Configura el comando <code>oc new-app</code> para crear un recurso <code>DeploymentConfig</code> en lugar de <code>Deployment</code> .
<code>--image-stream</code> <code>-i</code>	Proporciona el flujo de imágenes que se usará como la imagen de compilador S2I para una compilación S2I o para implementar una imagen de contenedor.
<code>--strategy</code>	<code>docker</code> , <code>pipeline</code> o <code>source</code>
<code>--code</code>	Proporciona la URL a un repositorio de Git que se usará como entrada de una compilación S2I.
<code>--image</code>	Proporciona la URL a una imagen de contenedor que se implementará.
<code>--dry-run</code>	Se establece en <code>true</code> para mostrar el resultado de la operación sin realizarla.
<code>--context-dir</code>	Proporciona la ruta a un directorio dentro del repositorio de Git que se tratará como la raíz de la aplicación.



IMPLEMENTACIÓN DE APLICACIONES CON OC NEW-APP

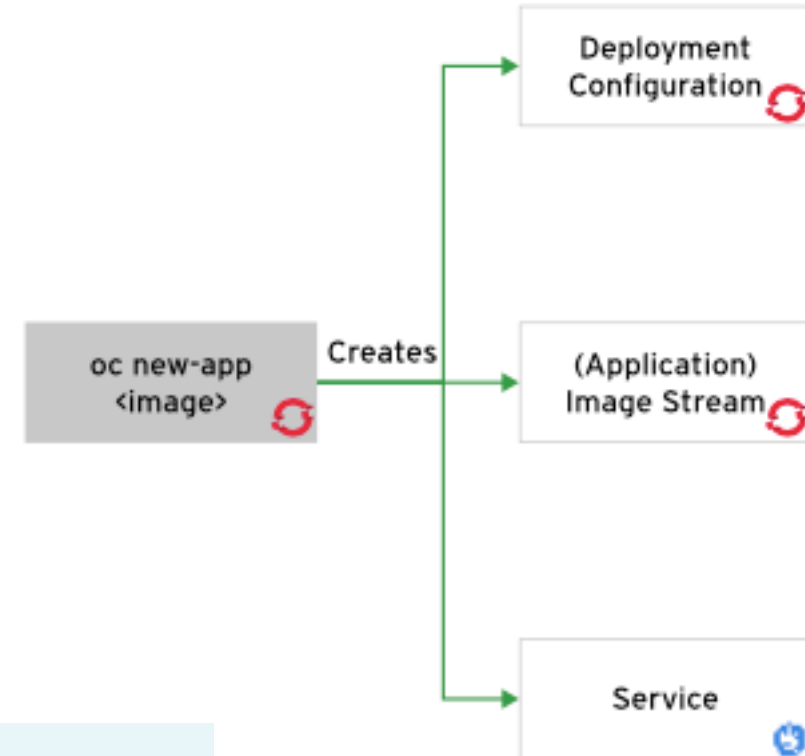
- El comando `oc new-app` crea una aplicación y la pone a disposición dentro del clúster.
- `oc new-app` toma un argumento de URL simple que apunta a un repositorio de Git o una imagen de contenedor.
- `oc` determina la estrategia de compilación de la imagen del contenedor siguiendo este proceso



IMPLEMENTACIÓN DE APLICACIONES CON OC NEW-APP

- El comando **oc new-app** se puede usar con la opción **-o json** o **-o yaml** para crear un archivo de definición de recursos de esqueleto en formato JSON o YAML, respectivamente.
- Este archivo se puede personalizar y utilizar para crear una aplicación mediante el comando **oc create -f <nombre de archivo>**, o combinar con otros archivos de definición de recursos para crear una aplicación compuesta.
- El comando **oc new-app** puede crear pods de aplicaciones para que se ejecuten en OpenShift de muchas formas diferentes.

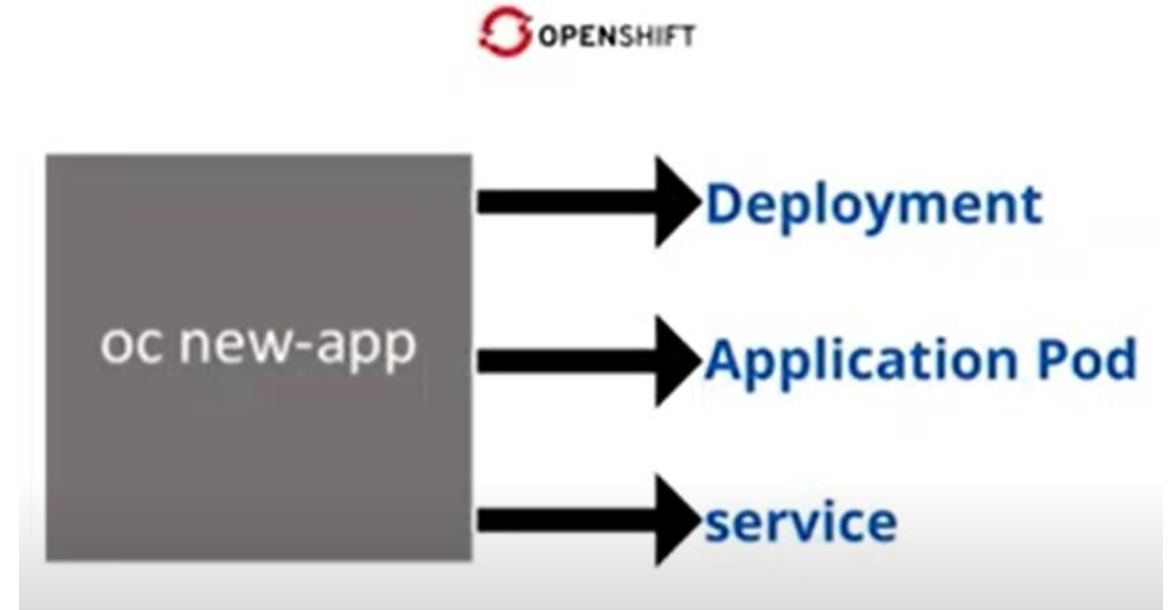
```
[student@workstation ~]$ oc new-app mysql \  
> MYSQL_USER=user MYSQL_PASSWORD=pass MYSQL_DATABASE=testdb -l db=mysql
```



RECURSOS DE APLICACIONES OPENS SHIFT

El comando `oc new-app` agrega los siguientes recursos al proyecto actual para admitir la compilación e implementación de una aplicación:

- Una configuración de compilación para compilar la imagen de contenedor de la aplicación a partir de código fuente o un Dockerfile.
- Un flujo de imágenes que apunta a la imagen generada en el registro interno o a una imagen existente en un registro externo
- Un recurso Deployment o DeploymentConfig que usa el flujo de imágenes como entrada para crear pods de aplicación.
- Un servicio para todos los puertos que expone la imagen de contenedor de la aplicación.



IMPLEMENTACIÓN DE APLICACIONES

- El siguiente comando implementa un servicio MySQL mediante el flujo de imágenes `mysql` y agrega la etiqueta `db=mysql` a los recursos que crea :

```
[user@host ~]$ oc new-app -i mysql \
-e MYSQL_USER=user -e MYSQL_PASSWORD=pass \
-e MYSQL_DATABASE=testdb -l db=mysql
```

- El siguiente comando crea una aplicación del código fuente en el lenguaje de programación PHP :

```
[user@host ~]$ oc new-app -i php \
https://gitserver.example.com/myrepo.git \
--name=hello \
--as-deployment-config
```



OC DESCRIBE RESOURCE_TYPE RESOURCE_NAME

- Si los resúmenes proporcionados por **oc get** son insuficientes, utilice el comando **oc describe** para obtener información adicional.

```
Name:      mysql-openshift-1-glqrp
Namespace:  mysql-openshift
Priority:    0
PriorityClassName:  none
Node:       cluster-worker-1/172.25.250.52
Start Time: Fri, 15 Feb 2019 02:14:34 +0000

Labels:     app=mysql-openshift
            deployment=mysql-openshift-1
            deploymentconfig=mysql-openshift

Annotations: openshift.io/deployment-config.latest-version: 1
            openshift.io/deployment-config.name: mysql-openshift
            openshift.io/deployment.name: mysql-openshift-1
            openshift.io/generated-by: OpenShiftNewApp
            openshift.io/scc: restricted

Status:     Running
IP:         10.129.0.85
```



GESTIÓN DE RECURSOS DE OPENS SHIFT EN LÍNEA DE COMANDOS: OC GET

- El comando `oc get all` para mostrar los recursos en el proyecto actual.

```
[user@host ~]$ oc get all
NAME                                READY    STATUS    RESTARTS   AGE
pod/hello-1-8hm7p                   1/1     Running   0           19m
pod/hello-1-build                   0/1     Completed 0           19m
pod/hello-1-deploy                   0/1     Completed 0           19m

NAME                                DESIRED   CURRENT   READY    AGE
replicationcontroller/hello-1       1         1         1        19m

NAME              TYPE          CLUSTER-IP    EXTERNAL-IP  PORT(S)          AGE
service/hello     ClusterIP     172.30.73.222 <none>       8080/TCP,8443/TCP 19m

NAME                                REVISION  DESIRED   CURRENT   TRIGGERED BY
deploymentconfig.../hello 1          1         1         config,image(hello:latest)

NAME                                TYPE      FROM    LATEST
buildconfig.build.openshift.io/hello Source     Git     1

NAME                                TYPE      FROM    STATUS    STARTED   DURATION
build.build.openshift.io/hello-1 Source     Git@ddfc7fe Complete  19 ...   26s

NAME                                IMAGE REPOSITORY          TAGS    UPDATED
imagestream.image.openshift.io/hello .../project-name/hello latest  16 ...
```

- 1 Pod que ejecuta el contenedor de la aplicación.
- 2 Pod que compila la aplicación a partir del código fuente. Inspeccione los registros de este pod para obtener más información sobre la compilación de la aplicación.
- 3 Pod de implementador que controla el proceso de implementación del recurso DeploymentConfig.
- 4 La configuración del controlador de replicación que administra la configuración de implementación.
- 5 El servicio que crea el comando `oc new-app`, exponiendo el puerto 8080/TCP después del proceso de compilación de S2I.
- 6 La configuración de implementación que crea el comando `oc new-app`.
- 7 La configuración de compilación que crea el comando `oc new-app`.
- 8 La primera compilación se desencadena con el comando `oc new-app`.
- 9 El flujo de imágenes que crea el comando `oc new-app`. Apunta a la imagen de contenedor que creada el proceso S2I.



INSPECCIONAR RECURSOS

Si desea inspeccionar las definiciones de los recursos sin crear los recursos en el proyecto actual, use la opción `-o`:

```
[user1@openshift-2-serverlab1 ~]$ oc new-app -o yaml docker.io/juanlle/images-ubi-greetings:1.0.1
apiVersion: v1
items:
- apiVersion: image.openshift.io/v1
  kind: ImageStream
  metadata:
    annotations:
      openshift.io/generated-by: OpenShiftNewApp
    creationTimestamp: null
    labels:
      app: images-ubi-greetings
      app.kubernetes.io/component: images-ubi-greetings
      app.kubernetes.io/instance: images-ubi-greetings
    name: images-ubi-greetings
  spec:
    lookupPolicy:
      local: false
    tags:
    - annotations:
        openshift.io/imported-from: docker.io/juanlle/images-ubi-greetings:1.0.1
      from:
        kind: DockerImage
        name: docker.io/juanlle/images-ubi-greetings:1.0.1
      generation: null
```

ELIMINAR RECURSOS

- Puede eliminar los recursos que crea el comando `oc new-app` mediante el uso de un solo comando **`oc delete`** con la opción **`--selector`** y la etiqueta .
- El siguiente comando elimina los recursos creados por el comando `oc new-app` anterior:

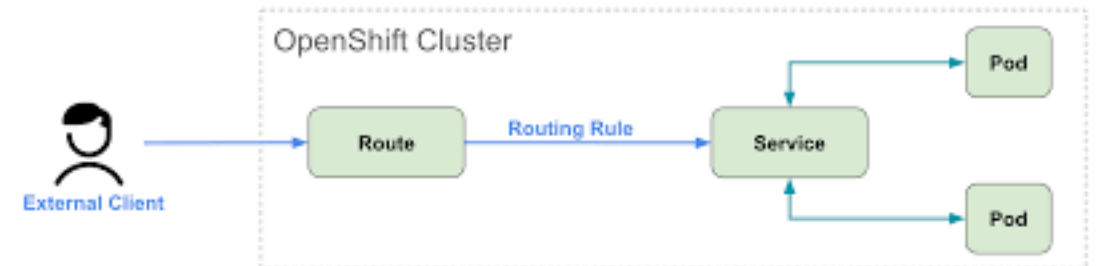
```
C:\Users\MasaAbushamleh\Desktop>oc delete all --selector app=myguestbook
pod "myguestbook-78ff54f76f-nsnxw" deleted
pod "myguestbook-78ff54f76f-qn5vw" deleted
deployment.apps "myguestbook" deleted
```



EXPOSICIÓN DE APLICACIONES FUERA DEL CLÚSTER.

- Para proporcionar acceso externo a su aplicación, puede usar el comando `oc expose`.

```
[user@host ~]$ oc expose service SERVICE_NAME
```



LAB 7

IMPLEMENTACIÓN DE APLICACIONES SIMPLES

