

# Diferencias entre Docker, Kubernetes, Apache Mesos y OpenShift

Computación Tolerante a Fallas

**Alumno:**

**Código:**

**Profesor(a):** Dr. Michel Emanuel López Franco

**Sección:** D06

**Fecha de Entrega:** 21 de noviembre de 2022

## Diferencias

Antes de comenzar, considero útil repasar las definiciones de cada uno de los software que se van a comparar.

**Docker** es una plataforma que empaqueta software en unidades estandarizadas llamadas contenedores las cuales contienen todo lo necesario para que se ejecute, incluidas bibliotecas, herramientas de sistema, código y tiempo de ejecución. De esta forma, es posible ejecutar aplicaciones rápidamente, además de implementar y ajustar la escala de estas rápidamente en cualquier entorno con la certeza de saber que su código se ejecutará. Cada contenedor contiene una serie de instrucciones que indicaran la forma en que debe arrancar.

**Kubernetes** es una plataforma de código abierto diseñada con el objetivo de poder administrar cargas de servicio y trabajos. Automatiza muchos de los procesos manuales involucrados en la implementación, la gestión y el ajuste de las aplicaciones que se alojan en contenedores. Entre sus principales características, se encuentra ser una plataforma de contenedores, de microservicios, y portable en la nube. Orquesta la infraestructura de cómputo, redes y almacenamiento para que las cargas de trabajo de los usuarios no tengan que hacerlo. Dentro de Kubernetes, existen los servicios. Estos son un recurso utilizado para crear un único punto de acceso a los pods, identificándolo con una dirección IP y puerto que nunca cambian. Existen cuatro servicios principales: Ingress, LoadBalancer, ClusterIP, y NodePort.

**Apache Mesos** es un kernel de sistemas distribuidos construido en base al kernel de Linux, solo que a un nivel de abstracción distinto. Es capaz de correr en Linux, MacOS, y Windows y provee aplicaciones con API's para el manejo de recursos y la planificación de tareas a través de todo el centro de datos o la nube. Permite escalabilidad líneal, alta disponibilidad, creación de contenedores, API's, y una interfaz web.

**OpenShift** es un producto de computación en la nube ofrecido por RedHat que permite a los usuarios alojar sus aplicaciones en contenedores de Kubernetes. El despliegue puede ser realizado por medio de GitHub, utilizando Dockerfile, una imagen de lanzamiento, o sus imágenes establecidas por defecto.

**Rancher** es una aplicación para gestionar de una forma centralizada nuestros clústeres de Kubernetes, con una propuesta principal basada en crear una capa de abstracción desde la que poder operar nuestros clústeres a diferentes niveles.

**Istio** es una malla de servicios que ofrece una manera transparente e independiente de cualquier lenguaje de automatizar las funciones de red de una aplicación de forma flexible y sencilla. Permite el intercambio de datos entre los microservicios y su diseño facilita su ejecución en distintos entornos, como contenedores de Kubernetes.

Si nos ponemos a analizar detenidamente estas aplicaciones, podemos notar que no son exclusivas, y algunas utilizan a otras para funcionar. OpenShift hace uso de Docker para desplegar los clústeres de Kubernetes.

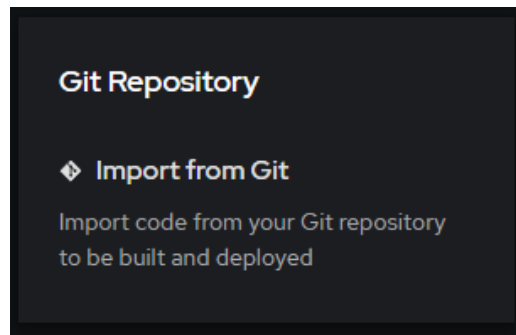
Rancher es una tecnología que sirve para manejar clústeres de Kubernetes en grandes cantidades, permitiendo controlar los accesos a estos y analizar las comunicaciones entre los pods a través de su interfaz gráfica. Istio permite hacer algo similar, pero se diferencia de Rancher en la capacidad de regular quien accede a que clústeres de Kubernetes dentro de una organización.

Docker es una plataforma útil para crear aplicaciones pequeñas dentro de contenedores, los cuales pueden ser compartidos con el resto del mundo en DockerHub. Por su parte, Kubernetes emplea Docker para crear los pods de un clúster, los cuales quizá no pueden ser compartidos directamente, pero poseen gran tolerancia a fallas y posibilidades de configuraciones relativas a la red y el balanceo de cargas.

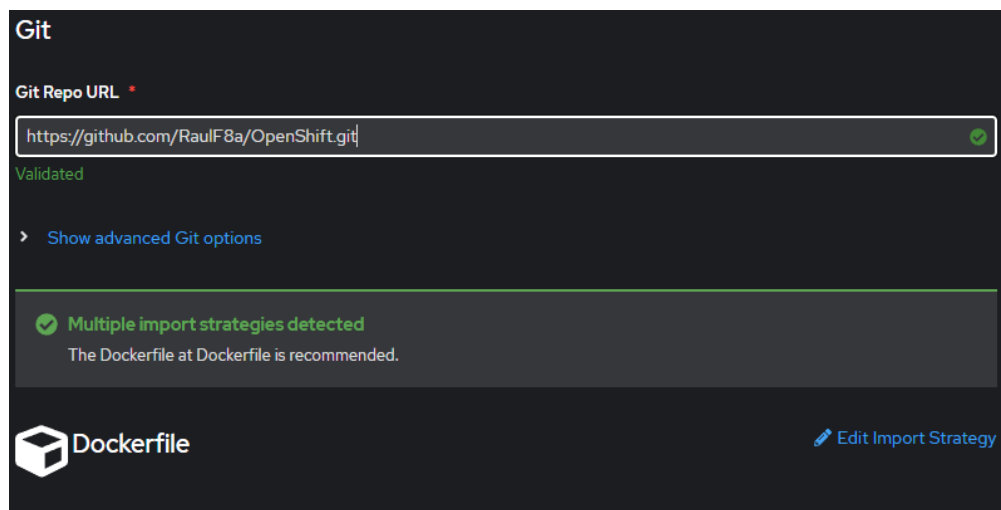
Cuando se tiene una empresa aún más grande y con funciones críticas como los bancos, o aseguradoras, que gestionan muchas aplicaciones y tal vez varias usen Kubernetes, se necesita una capa extra por debajo; una tecnología que aporte la infraestructura y permita gestionar distintas aplicaciones. aquí es cuando se utilizan herramientas como OpenShift y Apache Mesos, las cuales son administradoras de clústeres. Estas herramientas aportan una infraestructura virtualizada también utilizando contenedores. Mientras que Docker y Kubernetes dan soporte a nivel de aplicación, Apache Mesos y OpenShift dan soporte a un conjunto de aplicaciones a nivel de infraestructura.

Una vez definido lo anterior, podemos comenzar a realizar un ejemplo. Elegí utilizar OpenShift, ya que no he trabajado con ella anteriormente y creo me será útil para el proyecto final. Voy a utilizar la aplicación de la práctica de Istio, que consistía en una aplicación con Flask para obtener 5 pasos aleatorios de WikiHow y mostrarlos en pantalla.

Lo primero que hacemos es crear una cuenta de RedHat y suscribirnos al plan gratuito de OpenShift. Este nos permite trabajar con las suficientes herramientas para la práctica por treinta días. Una vez dentro, debemos seleccionar la opción para agregar una aplicación desde Git.



Colocamos el URL del repositorio y seleccionamos que la aplicación sea construida en base al Dockerfile incluido.



Debemos indicar el puerto en que la aplicación de Flask corre, de manera que se pueda ejecutar correctamente.

**Advanced options**

**Target port**

8000

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 [Health checks](#), [Build configuration](#), [Deployment](#), [Scaling](#), [Resource limits](#), and [Labels](#).

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

Inmediatamente se comienza a construir la aplicación dentro de un pod de Kubernetes. Puede tomar algunos minutos, dependiendo de que tan robusto sea el código.

**Pods**

**Waiting for the build**

Waiting for the first build to run successfully. You may temporarily see "ImagePullBackOff" and "ErrImagePull" errors while waiting.

[Show waiting pods with errors](#)

No Pods found for this resource.

**Builds**

BC open-shift-git

[Start Build](#)

Build #1 is running (Just now)

[View logs](#)

**Services**

S open-shift-git

Service port: 4000-tcp → Pod port: 4000

Una vez termina, se muestra el puerto donde esta corriendo, el servicio que se creo, y la ruta en la cual podemos acceder a la aplicación.

### Pods

P

open-shift-git-v5-7854f6bb84-lbm5f

Running

View logs

### Builds

BC

open-shift-git-v5

Start Build

✓

Build #1 was complete (2 minutes ago)

View logs

### Services

S

open-shift-git-v5

Service port: 8001-tcp → Pod port: 8001

### Routes

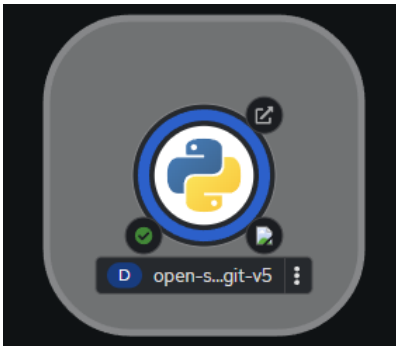
RT

open-shift-git-v5

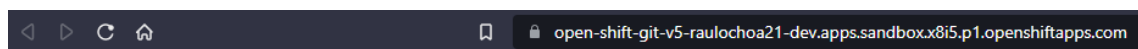
Location:

https://open-shift-git-v5-raulochoa21-dev.apps.sandbox.x8i5.p1.openshiftapps.com

🔗 📄



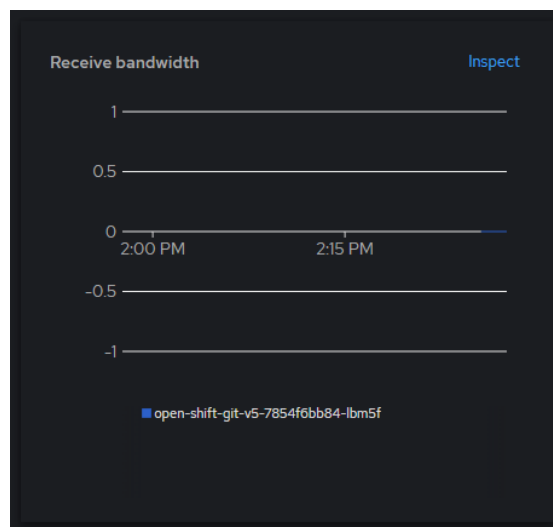
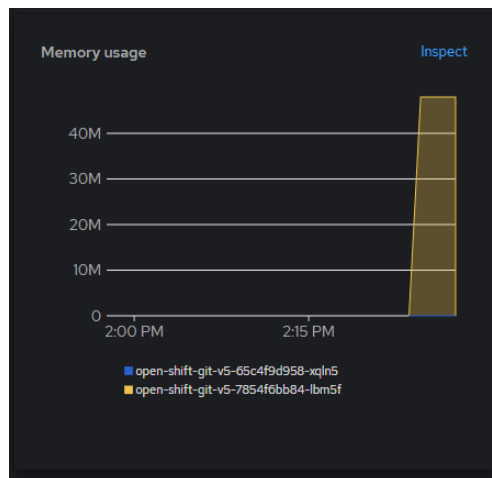
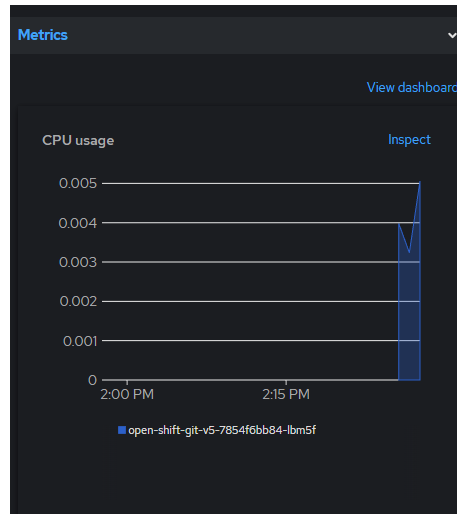
La aplicación se muestra de la siguiente forma:



## Cinco pasos aleatorios de tutoriales de WikiHow

1. Remind the reader about any accomplishments you have achieved while working with the company.
2. Move to the following group of three beads and say three Hail Marys.
3. Describe the scorpion to the medical assistance over the telephone.
4. Copy her makeup.
5. State "no es nada."

OpenShift nos permite ver métricas sobre el uso de CPU, memoria, y ancho de banda en el pod que corre la aplicación.



Es posible aumentar el número de pods, como quiera que necesitemos, ver el archivo utilizado para el despliegue, ver los logs de creación de la aplicación y del pod de Kubernetes.

Deployment details

1 Pod

Name

open-shift-git-v5

Update strategy

RollingUpdate

Namespace

NS raulochoa21-dev

Max unavailable

25% of 1 pod

Labels

app=open-sh...

app.kub...=open-sh...

app.kub...=open-sh...

app.kub...=open-sh...

app.kub...=to-do-lis...

app.ope...=python

app.ope...=3.9-ubi8

Edit

Max surge

25% greater than 1 pod

Pod selector

Q app=open-shift-git-v5

Progress deadline seconds

600 seconds

Node selector

No selector

Min ready seconds

Not configured

PodDisruptionBudgets

No PodDisruptionBudgets

```
D open-shift-git-v5
Details Metrics YAML ReplicaSets Pods Environment Events
AI • F1 Accessibility help • View source

1 kind: Deployment
2 apiVersion: apps/v1
3 metadata:
4   annotations:
5     alpha.image.policy.openshift.io/resolve-names: '*'
6   app.openshift.io/route-disabled: 'false'
7   app.openshift.io/cvs-ref:
8   app.openshift.io/cvs-uri: 'https://github.com/maufda/openShift-git'
9 deployment.kubernetes.io/revision: '2'
10 image.openshift.io/triggerPolicy: 'On'
11 [{"name": "kind", "imageStreamTag", "name": "open-shift-git-v5:latest", "namespace": "raulocha21-dev"}, {"fieldPath": "spec.template.spec.containers[?(@.name==\"open-shift-git-v5\")]"}]
12 openshift.io/generated-by: OpenShiftWebConsole
13 resourceVersion: '16196458'
14 name: open-shift-git-v5
15 uid: 3ef743f7-90ef-4b05-b019-df0a700d5a
16 creationTimestamp: '2022-11-15T20:22:34Z'
17 generation: 2
18 managedFields: ...
19 namespace: raulocha21-dev
20 labels:
21   app: open-shift-git-v5
22   app.kubernetes.io/component: open-shift-git-v5
23   app.kubernetes.io/instance: open-shift-git-v5
24   app.kubernetes.io/name: open-shift-git-v5
25   app.kubernetes.io/part-of: to-do-list-app
26   app.openshift.io/runtime: python
27   app.openshift.io/runtime-version: 3.9-ub18
28 spec:
29   replicas: 1
30   selector:
31     matchLabels:
```

[illegible]



Enlace a la Aplicación: <https://open-shift-git-v5-raulochoa21-dev.apps.sandbox.x8i5.p1.openshiftapps.com/>

## Conclusiones

Para ser la ultima actividad del curso, me pareció una manera muy interesante de consumir los conocimientos adquiridos a través de las practicas pasadas. Hice uso de los Dockerfiles, de los Kubernetes, y de una herramienta útil para monitorear el rendimiento de los contenedores. Aquí se remarca mi punto de que estos software no son mutuamente excluyentes, sino que funcionan uno con ayuda del otro.

Me voy satisfecho con lo aprendido a lo largo de estos meses, y espero poder aplicar estos conocimientos en el proyecto final.

## Bibliografía

¿Qué es Istio? (s. f.). <https://www.redhat.com/es/topics/microservices/what-is-istio>

Istio vs Rancher | What are the differences? (s. f.). StackShare. <https://stackshare.io/stackups/istio-vs-rancher>

Cloud, S. (2021, 10 mayo). ¿Qué es OpenShift? S&M Services. <https://sm-services.es/que-es-openshift/>