



Centro Universitario de Ciencias Exactas  
e Ingenierías.



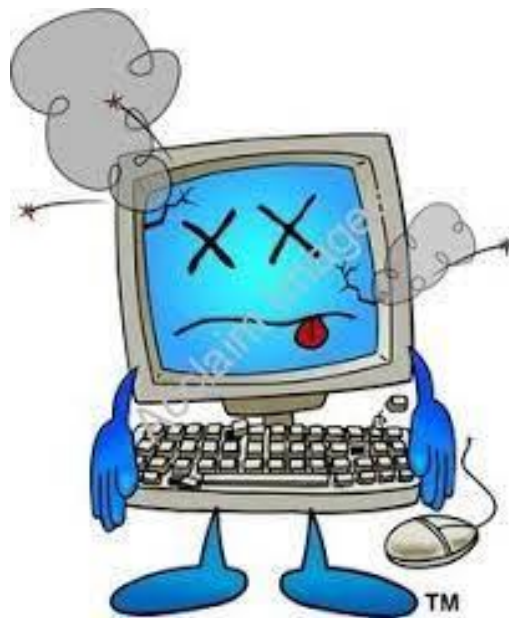
Ingeniería en computación.

Alumno: Vargas López David Guadalupe.

Computación tolerante a fallos.

Profesor: López Franco Michel Emanuel.

Sección: D06.



Guadalajara Jal. mayo del 2022.

## Docker:

### ¿Qué es?

Docker es una herramienta que permite empaquetar una aplicación y sus dependencias en un contenedor muy ligero. Es como si tomaras una aplicación completa con absolutamente todo lo que necesita para funcionar para poder transportarla sin problema a cualquier otro servidor con Docker instalado, ya sea para seguir desarrollándola o para hacer deploy.

### Características:

- Autogestión de los contenedores.
- Fiabilidad.
- Aplicaciones libres de las dependencias instaladas en el sistema anfitrión.
- Capacidad para desplegar multitud de contenedores en un mismo equipo físico.
- Puesta en marcha de los servicios en un abrir y cerrar de ojos.
- Contenedores muy livianos que facilitan su almacenaje, transporte y despliegue.
- Capacidad para ejecutar una amplia gama de aplicaciones (prácticamente cualquier cosa que se nos ocurra podrá ejecutarse en un contenedor Docker).
- Compatibilidad Multi-Sistema, podremos desplegar nuestros contenedores en multitud de plataformas.
- La aplicación base de Docker gestionará los recursos existentes para asignarlos responsablemente entre los contenedores desplegados.
- Podremos establecer una base desde la que comenzar nuestros proyectos, lo que nos ahorrará el tiempo de preparar el entorno para cada uno de ellos.
- Podremos compartir nuestros contenedores para aumentar los repositorios de Docker, así como beneficiarnos de los que compartan los demás.

## Usos:

Podemos encontrar repositorios públicos y totalmente gratuitos o repositorios privados donde tendremos que comprar las imágenes que necesitemos. Estos registros permiten desarrollar o desplegar aplicaciones de forma simple y rápida en base a plantillas, reduciendo el tiempo de creación o implementación de aplicaciones o sistemas.

## Kubernetes:

### ¿Qué es?

Kubernetes es una plataforma portable y extensible de código abierto para administrar cargas de trabajo y servicios. Kubernetes facilita la automatización y la configuración declarativa. Tiene un ecosistema grande y en rápido crecimiento. El soporte, las herramientas y los servicios para Kubernetes están ampliamente disponibles.

### Características:

- Escalado vertical. En función del uso de CPU que hagamos Kubernetes nos permite el escalado vertical de nuestras aplicaciones de manera automática (autoescalado) o manual (comando).
- Reparación automática. En caso de ocurrir un fallo en un contenedor podemos reiniciar automáticamente, así como reemplazarlo o replanificarlo cuando un nodo muere. También nos ofrece la posibilidad de reparar aquellos health checks definidos por el usuario que no respondan.
- Planificación de nodos y contenedores. Kubernetes nos ayuda a decidir en qué nodo se ejecutará cada contenedor, en función de los recursos necesarios y otras restricciones. Además, podemos mezclar cargas de trabajo críticas y best-effort con el objetivo de potenciar el ahorro de recursos.
- Despliegues y rollbacks automáticos. Podemos desplegar los cambios de manera progresiva cuando actualizamos una aplicación o cambiamos su configuración, y así poder realizar un rollback automático en caso de fallo en alguna de las instancias.
- Orquestador de contenedores. Kubernetes permite montar de forma automática el sistema de almacenamiento necesario para los contenedores: en modo local,

en un proveedor de cloud pública, o en un sistema de red como NFS, Flocker o Clúster.

- Almacenamiento persistente. Kubernetes goza de soporte en plataformas como Amazon Web Services o Google Cloud Platform, y los proveedores (Red Hat, Dell EMC, NetApp, etc) le proporcionan almacenamiento persistente.
- Descubrimiento de servicios. Con Kubernetes asignamos a los contenedores sus propias direcciones IP y un nombre DNS específico para cada conjunto de contenedores. Así, no resulta necesario utilizar recursos externos para el descubrimiento de servicios.
- Seguridad. Toda nuestra información sensible, como contraseñas o claves ssh, podemos almacenarla de manera segura en secrets. En este sentido, Kubernetes no expone nuestra información confidencial a la hora de desplegar y configurar nuestras aplicaciones.
- Clusters grandes y heterogéneos. Kubernetes puede desplegarse en clusters muy grandes, incluyendo contenedores Docker. Por otro lado, nos permite crear un clúster como resultado de combinar diferentes máquinas virtuales o servidores locales.

## Usos:

Kubernetes es una plataforma open source para administrar cargas de trabajo y servicios, facilitando la automatización y eliminando muchos de los procesos manuales durante la implementación y escalabilidad de las aplicaciones en contenedores. En este sentido, nos ofrece una mayor simpleza en plataformas como servicio (PaaS) y una mayor flexibilidad en la infraestructura como servicio (IaaS), permitiendo la portabilidad entre proveedores de infraestructura. Se puede desplegar en diversos entornos cloud y soporta múltiples runtimes de contenedores (docker, rkt, cri-o o frakti).

# Apache Mesos:

## ¿Qué es?

Es un administrador de Cluster de código abierto basado en el kernel de Linux, a diferencia de que Mesos puede correr en cualquier plataforma ya sea Linux, Windows o OSX. Además, proporciona aplicaciones con API's para el manejo de recursos y planificación.

## Características:

Escalabilidad Lineal: puedes escalar fácilmente de forma lineal hasta 10.000 nodos.

Alta disponibilidad: réplica de máster y agentes tolerante a fallos usando Zookeeper

Containers: soporta nativamente el uso de containers con Docker y Appc.

Conexiones Aisladas: utiliza Cgroups para el aislamiento de CPU, memoria, disco, puertos, GPU y módulos para el aislamiento de recursos personalizados.

**Dos niveles de Planificación:** soporta ejecutar aplicaciones heredadas y nativas de la nube en el mismo clúster con políticas de programación conectables.

API's: provee aplicaciones HTTP para el desarrollo de nuevas aplicaciones distribuidas para operar el Cluster y monitoreo.

Web UI: interfaz de usuario web incorporada para ver el estado del clúster y navegar por los contenedores

MultiPlataforma: corre en Linux, OSX y Windows.

## Usos:

Mesos se utiliza para correr grandes aplicaciones especialmente aquellas que manejan grandes volúmenes de datos y se hace inviable correr en un único servidor. Mesos permite utilizar el cluster para correr los frameworks sobre los cuales se ejecuta la aplicación, distribuyendo la carga entre los distintos servidores y evitando así

sobrecargas logrando un óptimo desempeño. Mesos se utiliza comúnmente para aplicaciones Java, Pythom, Scala y R.

## open Shift:

### ¿Qué es?

Openshift es una plataforma de desarrollo, con características de Cloud Computing (PaaS) que ofrece la empresa de Red Hat.

Esta plataforma se desarrolla a partir de contenedores kubernetes que los desarrolladores utilizan para desplegar aplicaciones en diferentes lenguajes de programación

### Características:

Multicloud: A través de esta plataforma un desarrollador podrá darle a su aplicación la versatilidad que necesita, ya que podrá adaptarla a las regulaciones de cualquier país, preservar los datos de forma segura, garantizar su funcionamiento y ofrecer un rendimiento más rápido y confiable.

Nube Híbrida: Al trabajar tanto en nube pública, como privada, OpenShift ofrece los beneficios de contar con una nube híbrida, logrando así contar con un mayor control de las cargas de trabajo, una reducción de costos, y gran facilidad para administrar y gestionar todo lo relacionado con su aplicación.

Monitorización y automatización sencilla: Openshift ofrece a los desarrolladores, sistemas automatizados para gestionar el ciclo de vida de nuestra aplicación de manera más sencilla y efectiva, de esta forma su mayor preocupación será, desarrollar.

### Usos:

Como conclusión, Open Shift es una herramienta que nos ofrece una solución muy práctica para los desarrolladores, ya que les abre la posibilidad de tener en un clúster de OpenShift su aplicación y contar con todas las posibilidades de funcionalidad que ofrece kubernetes para la gestión.

# Rancher:

## ¿Qué es?

Rancher es una pila de software que se utiliza para gestionar clústeres Kubernetes. Se trata básicamente de un software que DevOps puede utilizar al adoptar el usuario de contenedores. Rancher incluye una distribución completa de Kubernetes, Docker Swarm y Apache Mesos, lo que facilita la gestión de clústeres de contenedores en cualquier plataforma de nube. Algunas de las empresas populares que utilizan Rancher son: Alibaba travelers, Abeja, Trivago, UseInsider, Starbucks, Oxylabs, yousign, y muchas más.

## Características:

Algunas de las principales características de Rancher son las siguientes

- Aplicación del catálogo Docker
- Distribución de Kubernetes incluida
- Distribución de Docker Swarm incluida
- Distribución de Mesos incluida
- Gestión de infraestructuras
- Algunas de las características clave de Rancher son las siguientes;
- Gestión de hosts, despliegue de contenedores, supervisión de recursos
- Gestión de usuarios y colaboración
- APIs y herramientas nativas de Docker
- Control y registro
- Conectar contenedores, gestionar discos, desplegar balanceadores de carga.

## Usos:

Las soluciones heredadas como OpenShift y Tanzu ofrecen portabilidad limitada, una dependencia máxima sobre el proveedor y modelos de precios elevados por núcleo. Sus pilas integradas verticalmente, desarrolladas con el tiempo a través de la adquisición, son complejas, engorrosas y difíciles de usar.

## Istio:

### ¿Qué es?

Istio es una malla de servicios, una capa de red de servicios modernizada que proporciona una forma transparente que no depende del lenguaje para automatizar con facilidad y flexibilidad las funciones de la red de aplicaciones. Es una solución popular para administrar los diferentes microservicios que componen una aplicación nativa de la nube.

### Características:

- Rotura de circuitos

Si un servicio está sobrecargado o inactivo, las solicitudes adicionales fallarán mientras se continúa sobrecargando el sistema. Debido a que Istio está rastreando errores y retrasos, puede forzar una pausa, permitiendo que un servicio se recupere después de un número específico de solicitudes establecidas por la política. Puede aplicar esta política en todo el clúster mediante la creación de un archivo de texto pequeño y direccionando a Istio para utilizarlo como una nueva política.

- Seguridad

Istio proporciona identidad, política y cifrado de forma predeterminada, junto con autenticación, autorización y auditoría (AAA). Cualquier pod bajo administración que se comunique con otros utilizará el tráfico cifrado, evitando cualquier observación. El servicio de identidad, combinado con el cifrado, asegura que ningún usuario no autorizado pueda falsificar o suplantar la identidad (spoof) de una llamada de servicio. AAA proporciona a los profesionales de seguridad y operaciones las herramientas que necesitan para supervisar, con menos costos generales.

- Administración simplificada

Las applications traditionalist todavía necesitan las características de identificación, política y seguridad que ofrece Istio. Eso implica programadores y administradores trabajando en el nivel equivocado de abstracción, reimplemented las mismas reglas de seguridad una y otra vez para cada servicio. Istio les permite



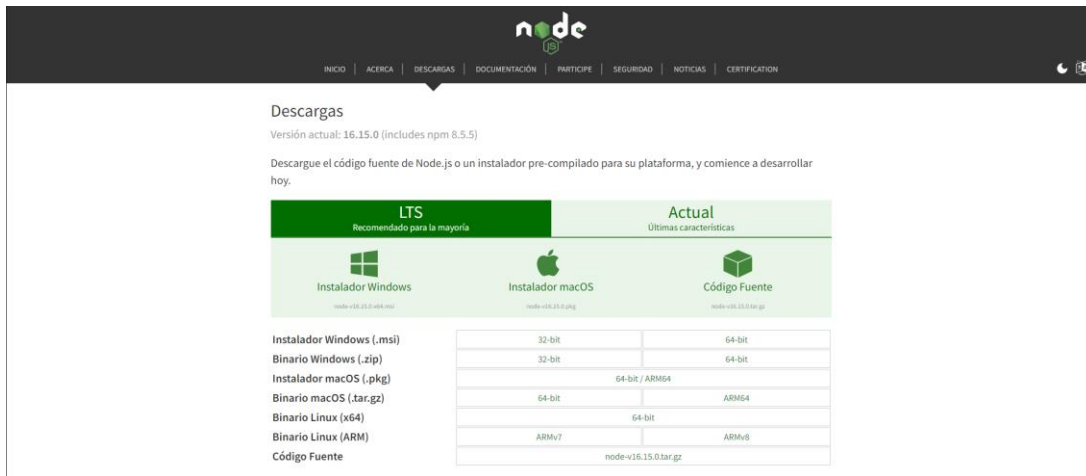
trabajar en la política de ajuste de nivel correcta para el clúster por medio de un único panel de control. Al mismo tiempo, con los controles de acceso de Istio, los paneles de control y las herramientas de depuración que se describen a continuación, puede añadir fácilmente un plug-in en la línea de comando, en lugar de ir a una página web.

### Usos:

Istio permite que las organizaciones entreguen aplicaciones distribuidas a gran escala. Simplifica las operaciones de red de servicio a servicio, como la administración del tráfico, la autorización y la encriptación, así como la auditoría y la observabilidad. Estos son algunos de los casos de uso más comunes en los que se pueden ver los beneficios de usar Istio.

## Aplicación usando openshift.

Instalamos node de acuerdo con el sistema operativo y su distribución más adecuada.



Observamos las versiones tanto de node como de npm para poder continuar usándolas.

```
PS C:\Users\david\OneDrive\Escritorio\NODE JS\Node-Web-Server> node --version
v16.15.0
PS C:\Users\david\OneDrive\Escritorio\NODE JS\Node-Web-Server> node --version
v16.15.0
PS C:\Users\david\OneDrive\Escritorio\NODE JS\Node-Web-Server>
```

Instalamos express y posteriormente levantamos el servidor con node. Dentro del puerto 8080 como se observa.

```
npm WARN old lockfile
npm WARN old lockfile The package-lock.json file was created with an old version of npm,
npm WARN old lockfile so supplemental metadata must be fetched from the registry.
npm WARN old lockfile This is a one-time fix-up, please be patient...
npm WARN old lockfile

added 57 packages, and audited 58 packages in 2s

7 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\david\OneDrive\Escritorio\NODE JS\Node-Web-Server> node server.js
Server started at http://localhost:8080
PS C:\Users\david\OneDrive\Escritorio\NODE JS\Node-Web-Server>
```

Y corroboramos que efectivamente se haya levantado el servidor node. dirigiéndonos a cualquier navegador en el localhost dentro del puerto 8080.



Se sube la aplicación a GitHub, para posteriormente colocarlo dentro de openshift.

```
david@LAPTOP-OMJ7SMPI MINGW64 ~/OneDrive/Escritorio/Nueva carpeta (2) (main)
$ git push -u origin main
Enumerating objects: 195, done.
Counting objects: 100% (195/195), done.
Delta compression using up to 8 threads
Compressing objects: 100% (192/192), done.
Writing objects: 100% (195/195), 2.51 MiB | 1.88 MiB/s, done.
Total 195 (delta 10), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (10/10), done.
To https://github.com/David-1212/openshift.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.

david@LAPTOP-OMJ7SMPI MINGW64 ~/OneDrive/Escritorio/Nueva carpeta (2) (main)
$
```

Se crea una cuenta dentro de red hat openshift para poder añadir una aplicación.

## Necesitamos un poco más de información

Red Hat garantiza la seguridad y responsabilidad de nuestros productos mediante la verificación de información adicional.

\* Campos obligatorios

Número de teléfono \*

3881020569

País \*

Mexico

Línea de dirección 1 \*

Blvd. Gral .Marcelino Garcia Barragan 1421

Línea de dirección 2

Línea de dirección 3

addressLine4

Código postal

44430

Ciudad \*

Guadalajara

Estado/Provincia \*

Jalisco

Nos pide confirmación de número de teléfono para posteriormente colocar el código de seguridad que se manda a dicho número de teléfono.

### Confirm your account via text

Country Code and Phone Number

We require a valid phone number to reduce the creation of fraudulent accounts. Red Hat will not use this information for any other reason.

Cost may apply for receiving texts if abroad. Please consult with your service provider for details.

- 1 Provision your free Red Hat OpenShift development cluster
- 2 Deploy a [sample application](#) or bring your own repo
- 3 Edit code from the integrated [Eclipse Che-based cloud IDE](#)
- 4 Fully explore the Red Hat OpenShift developer experience

Se coloca el código de confirmación de numero de teléfono para seguir con el proceso.

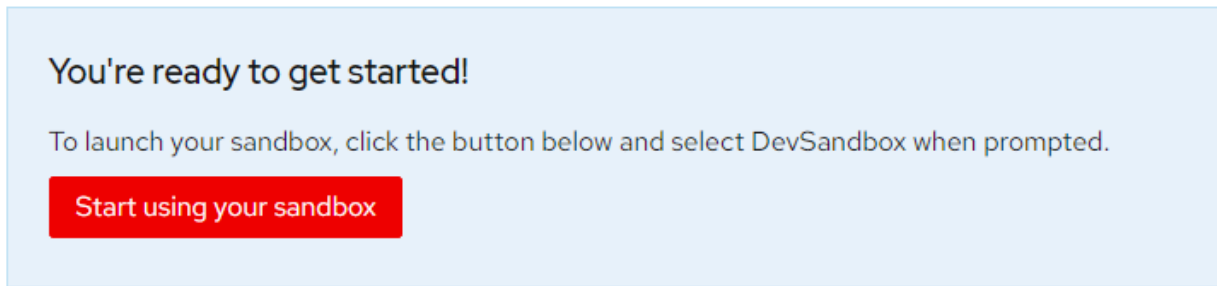
### Confirm your account via text

Enter the code received by text

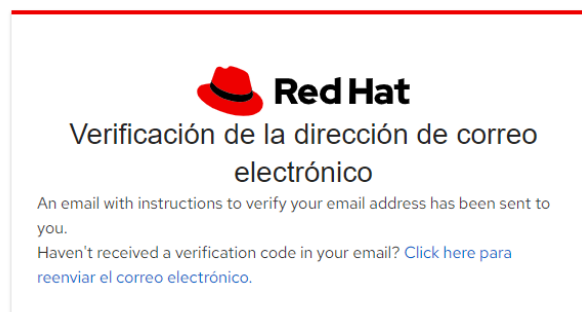
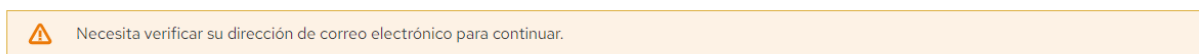
Didn't receive a code? [Resend](#)

Note: verification codes are sent from a US number, if you are outside the US your carrier may block international SMS. If you're having trouble, please email [devsandbox@redhat.com](mailto:devsandbox@redhat.com) to verify your account.

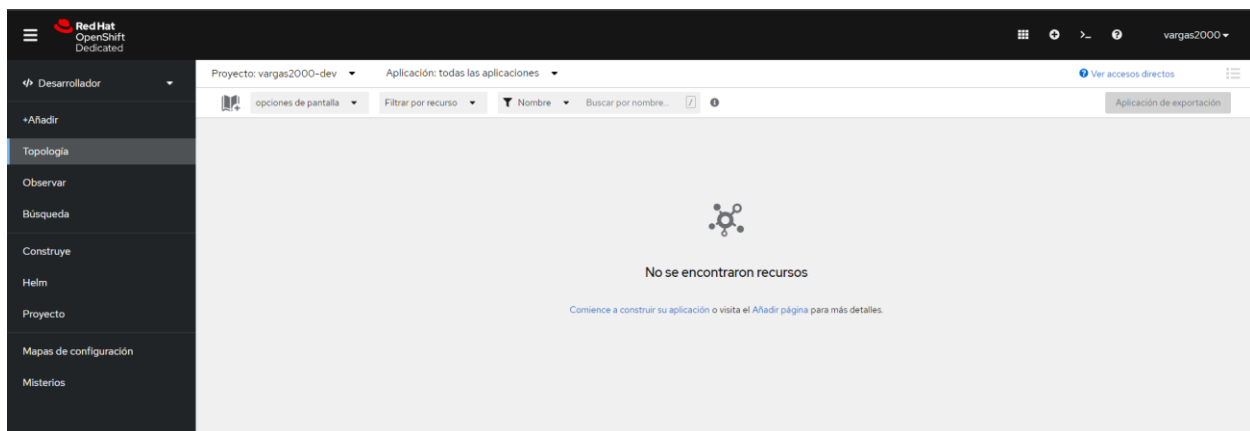
Se presiona el botón rojo para indicar que estamos listos para usarlo.



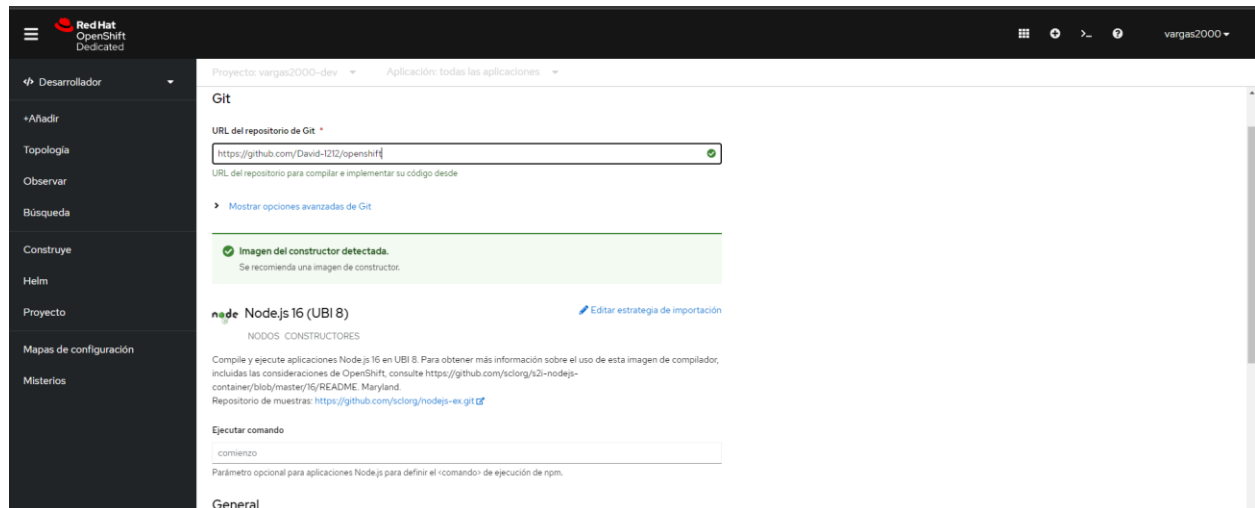
Ahora se pide una verificación de correo electrónico para continuar con el proceso.



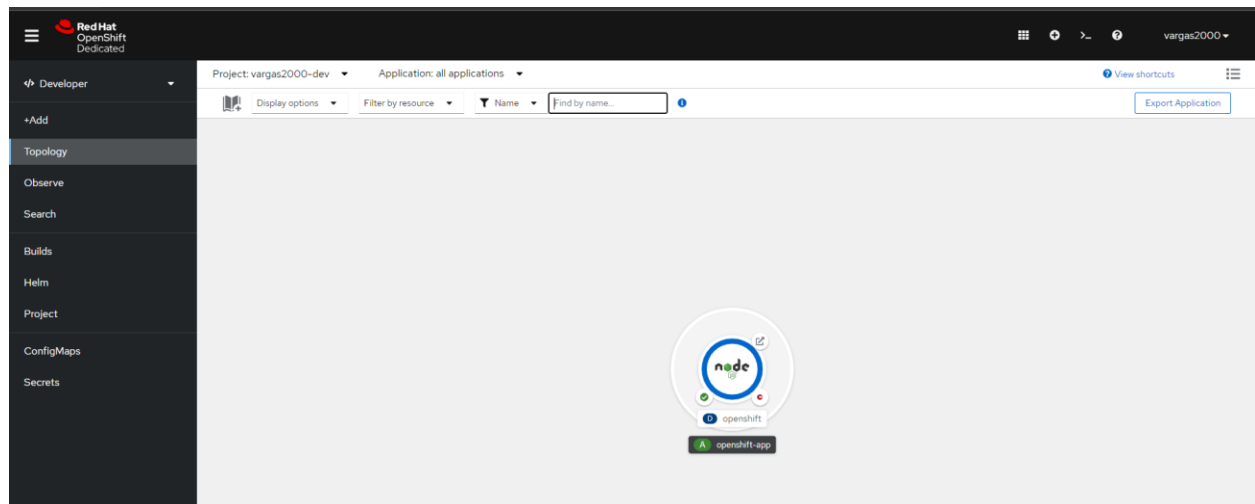
Una vez terminado este proceso, se dirige a la pagina principal, donde esta vacía debido a que aun no se sube ninguna aplicación.



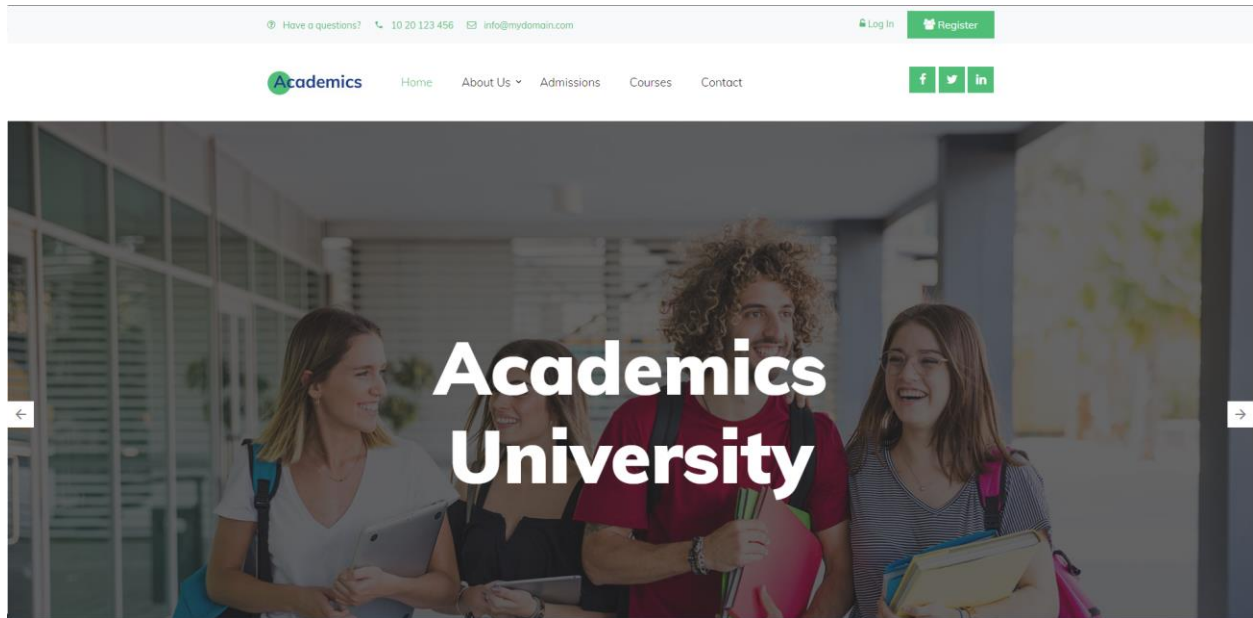
Se coloca el link de GitHub para subir la aplicación y posteriormente presionamos crear.



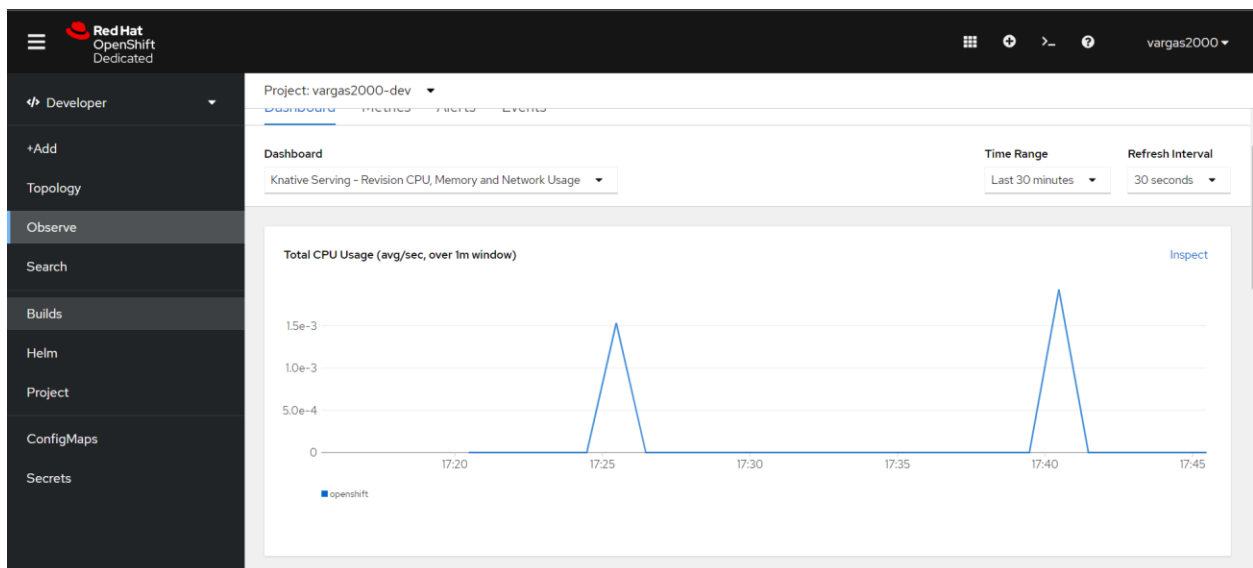
Una vez creada, esperamos a que esta lista para usarse, esto se sabe cuando aparece la palomita en color verde.



Por último observamos el resultado, como se ve se abre la aplicación antes creada.

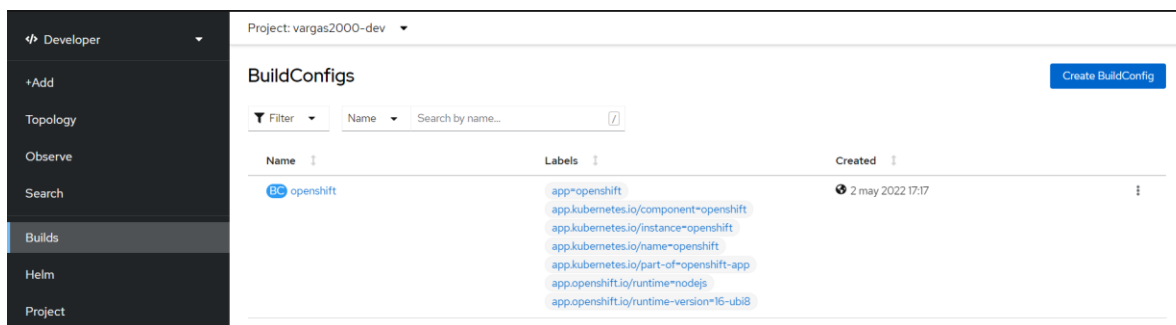


En la parte de observe se puede apreciar el estado de la aplicación además de otros aspectos en forma de gráficos de picos.

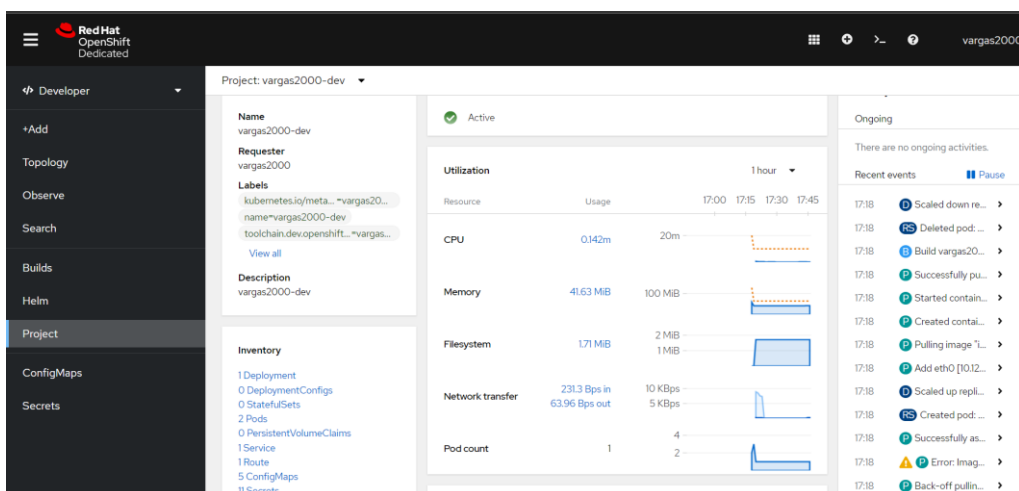




En la parte de build podemos observar que se crearon diferentes archivos, entre ellos una imagen para que se pueda ejecutar la aplicación.



En la parte de project se observan diferentes graficas acerca de la CPU, memoria y demás aspectos.



Por ultimo en la parte de secretos se manejan aspectos de detalles acerca de la app y de su estado al igual que la parte de observe.

The screenshot shows the 'Secrets' page in the OpenShift console for project 'vargas2000-dev'. The left sidebar has 'Secrets' selected. The main area displays a table with the following data:

Name	Type	Size	Created
builder-dockercfg-mmt9	kubernetes.io/dockercfg	1	2 may 2022 16:43
builder-token-9e77v	kubernetes.io/service-account-token	4	2 may 2022 16:43
builder-token-tnvcb	kubernetes.io/service-account-token	4	2 may 2022 16:43
default-dockercfg-thmgz	kubernetes.io/dockercfg	1	2 may 2022 16:43
default-token-nj6q8	kubernetes.io/service-account-token	4	2 may 2022 16:43
default-token-c5gpc	kubernetes.io/service-account-token	4	2 may 2022 16:43
deployer-dockercfg-tthfm	kubernetes.io/dockercfg	1	2 may 2022 16:43
deployer-token-c6qql	kubernetes.io/service-account-token	4	2 may 2022 16:43
deployer-token-cbyj5	kubernetes.io/service-account-token	4	2 may 2022 16:43
openshift-generic-webhook-secret	Opaque	1	2 may 2022 17:17
openshift-github-webhook-secret	Opaque	1	2 may 2022 17:17

## Conclusiones:

En este caso yo llegue a la conclusión de que es interesante ver como se pueden utilizar este tipo de aplicaciones para manejar y hostear apis creadas desde cualquier lenguaje, en este caso se creó una aplicación en node .js desde donde se utiliza esta aplicación simple la cual funciono sin problemas al momento de ejecutar un clúster.

En lo personal esto lo utilizaría para proyectos no tan complejos, y no tan elaborados, aunque es bastante fácil de implementar, aunque sería bastante interesante de observar al momento de realizar.

## Link al repositorio:

<https://github.com/David-1212/openshift>

## Bibliografía:

- Cloud, S. (2021, May 10). *¿Qué es OpenShift? Diferencias con Docker y Kubernetes* • S&M Services. <https://sm-services.es/que-es-openshift/#:%7E:text=Openshift%20es%20una%20plataforma%20de,en%20diferentes%20lenguajes%20de%20programaci%C3%B3n>.
- Peña, A. (2021, January 11). *¿Qué es Kubernetes? ¿Para qué sirve?* Formadores IT. <http://www.formadoresit.es/que-es-kubernetes-para-que-sirve/>
- V. (2021, July 6). *Apache Mesos* –. Guiadev. <https://guiadev.com/apache-mesos/#:%7E:text=3%20Conclusi%C3%B3n-,%C2%BFQu%C3%A9%20es%20Apache%20Mesos%3F,manejo%20de%20recursos%20y%20planificaci%C3%B3n>.

