Diferencias entre Docker, Kubernetes, Apache Mesos y OpenShift

Nombre: Diana Fernanda Castillo Rebolledo

Materia: Computación Tolerante a Fallas

Horario: L-I 11:00 am-1:00 pm

Profesor: Dr. Michel Emanuel López Franco

Sección: D06

Diferencias entre Docker, Kubernetes, Apache Mesos, OpenShift, Rancher e Istio

Todas las anteriores son un conjunto de herramientas de contenedores, Docker es la base de estas tecnologías, pues prácticamente toda la industria utiliza los contenedores de Docker, y para poder montar una arquitectura de cualquiera de estas tecnologías es necesario tener buenos conocimientos sobre Docker porque este se utiliza para crear las imágenes de la aplicación empaquetada que después se van a desplegar. Docker es más que suficiente para pymes y pequeños proyectos. Por su parte, Kubernetes es un orquestador de aplicaciones con Docker (contenedores), pero a diferencia de Docker, en Kubernetes no pueden crearse imágenes ni subirse a Docker Hub, este más bien es un integrador de pods con contenedores para su gestión en un clúster, tiene distintas funcionalidades, como la capacidad de crear tareas programadas, además de que su arquitectura es muy propia pues se tienen los pods, los servicios y demás, por lo que una aplicación tiene que ser adaptada o diseñada pensando en Kubernetes si es que quiere utilizarse esta tecnología, la cual se utiliza en proyectos más grandes que los de Docker, como proyectos de Big Data y Machine Learning. Debido a la complejidad del software, Kubernetes es más apropiado para proyectos más grandes con múltiples años de duración y el presupuesto correspondiente.

Istio tiene otro tipo de arquitectura, la clave para entenderla es conocer Envoy y Kubernetes. A menudo, las tres soluciones se usan a la vez para conseguir que el entorno en contenedores basado en microservicios tenga un mejor rendimiento. Por ejemplo, las mallas de servicio como Istio cuentan con un plano de control y otro de datos. Istio utiliza una versión ampliada de Envoy como plano de datos. Después, Envoy gestiona todo el tráfico entrante y saliente en la malla de servicios de Istio. Por otra parte, Kubernetes automatiza y orquesta las tareas de despliegue y escalado de las aplicaciones en contenedores para minimizar los procesos manuales. Aunque Istio no depende de ninguna plataforma, muchos desarrolladores lo utilizan junto con Kubernetes, y se utiliza en aplicaciones que tengan distintas funciones para poder ser separadas en microservicios.

Cuando se tiene una empresa aún más grande y con funciones críticas como los bancos, aseguradoras o grandes empresas, que gestionan muchas aplicaciones, tal vez varias usando Kubernetes, ya se necesita una capa extra por debajo, es decir, 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.

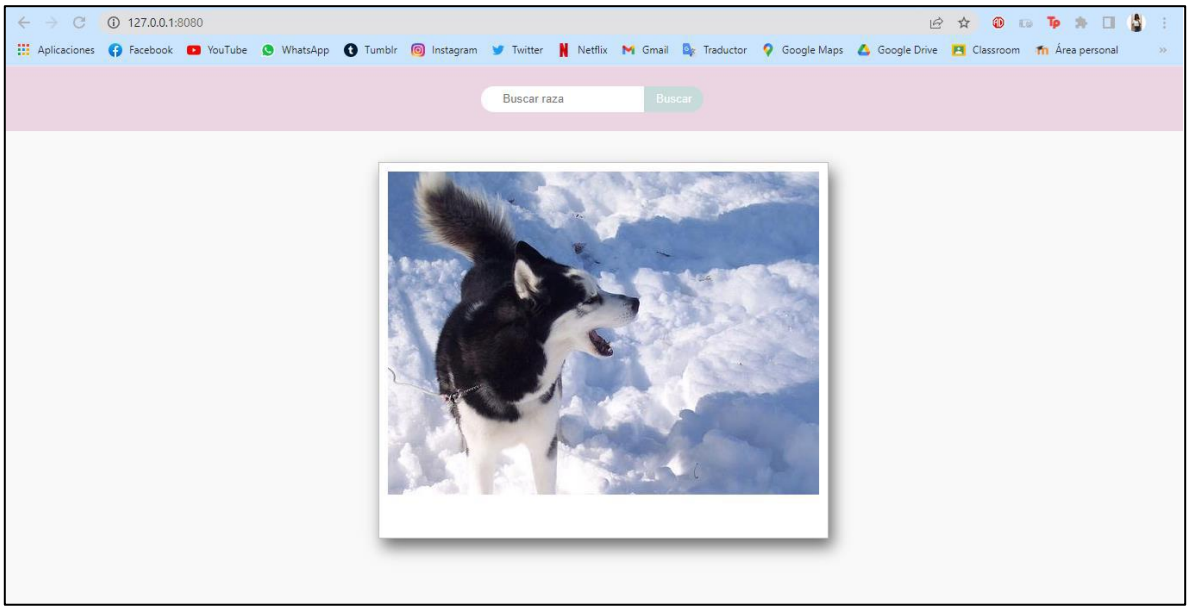
Apache Mesos abstrae los recursos informáticos como la CPU y las memorias de gran capacidad y de trabajo de las máquinas reales, de manera que pueden escalarse clústeres de decenas de miles de nodos. Además, se utilizan nodos manager y agentes redundantes con tolerancia a fallos, que según las necesidades pueden recibir actualizaciones sin interrupciones. El software opera con un nivel de abstracción inferior al de Kubernetes, por

lo que no es tan útil para clústeres pequeños de menos de una docena de nodos. Permite construir sistemas distribuidos elásticos y con tolerancia a fallos. Es particularmente apto para sistemas grandes diseñados para la máxima redundancia.

OpenShift se basa en Kubernetes, esta permite orquestar contenedores y redes, así como gestionar o asignar recursos escalables y distribuidos, es una distribución de Kubernetes, pero esta cuenta con servicios adicionales a Kubernetes, como la capacidad de crear imágenes automáticamente. Puede desplegarse en múltiples entornos en la nube, tanto las nubes privadas, como las infraestructuras en la nube de Amazon AWS y Microsoft Azure. Openshift cuenta con una interfaz web para controlar sus funcionalidades. La gran ventaja de OpenShift frente a Kubernetes yace en que proporciona a los equipos de desarrollares un entorno de trabajo consistente y robusto. Otro punto fuerte de OpenShift son las herramientas de monitoreo y registro incorporadas, como Prometheus, Jaeger y Kali. OpenShift es particularmente adecuado para implementar estrategias de nube híbrida y para construir y escalar las aplicaciones en contenedores necesarias para ello.

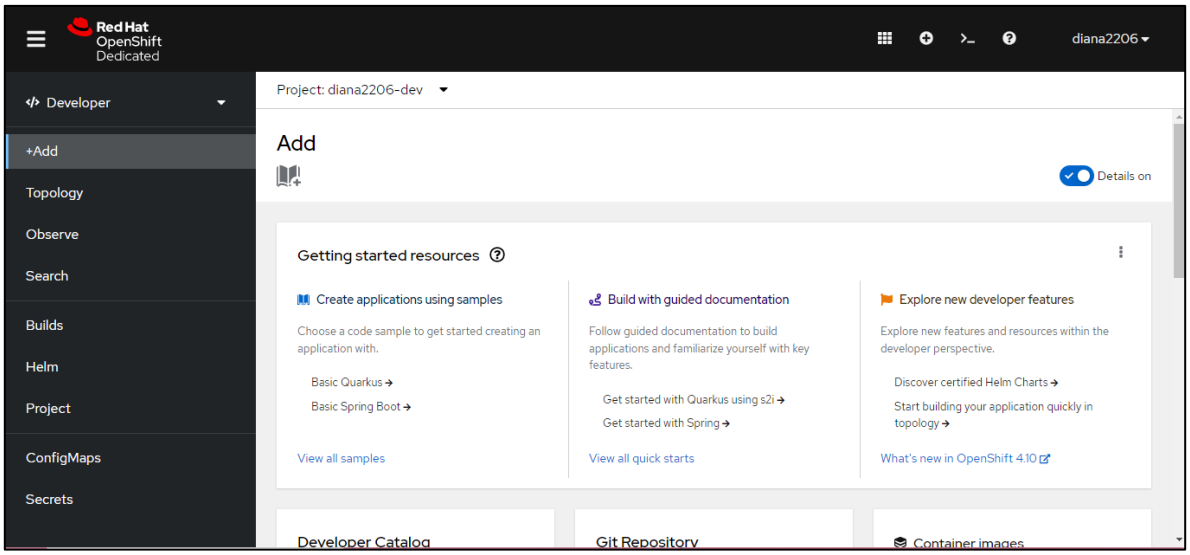
Rancher también está basada en Kubernetes. Es una pila de software completa para los equipos que dependen de los contenedores, se ha integrado para proporcionar soporte de redes y equilibrio de carga. Aborda los desafíos operativos y relevantes para la seguridad que pueden surgir al ejecutar múltiples clústeres de Kubernetes. Rancher pone a disposición una interfaz de control centralizada y unifica la gestión de todos los clústeres de Kubernetes de una organización. Simplifica el despliegue de clústeres de Kubernetes tanto en máquinas físicas como en entornos de nube pública y privada. Los clústeres se protegen mediante políticas globales de seguridad. Utiliza sistemas centralizados de autentificación, control de acceso y rastreo. El software evita fallos de clúster y lleva a cabo medidas de rescate y restauración si es necesario.

Una vez definido lo anterior, se explicará un ejemplo utilizando la herramienta de Openshift para alojar la aplicación de Python utilizada en la actividad anterior, la cual consistía en una aplicación en Flask que obtiene datos a través de una API de perros llamada “Dog API”, que almacena links de imágenes de diversas razas. En el buscador de la aplicación se puede buscar una raza y aparecerán imágenes random de perros de esa raza cada que se recargue la página.

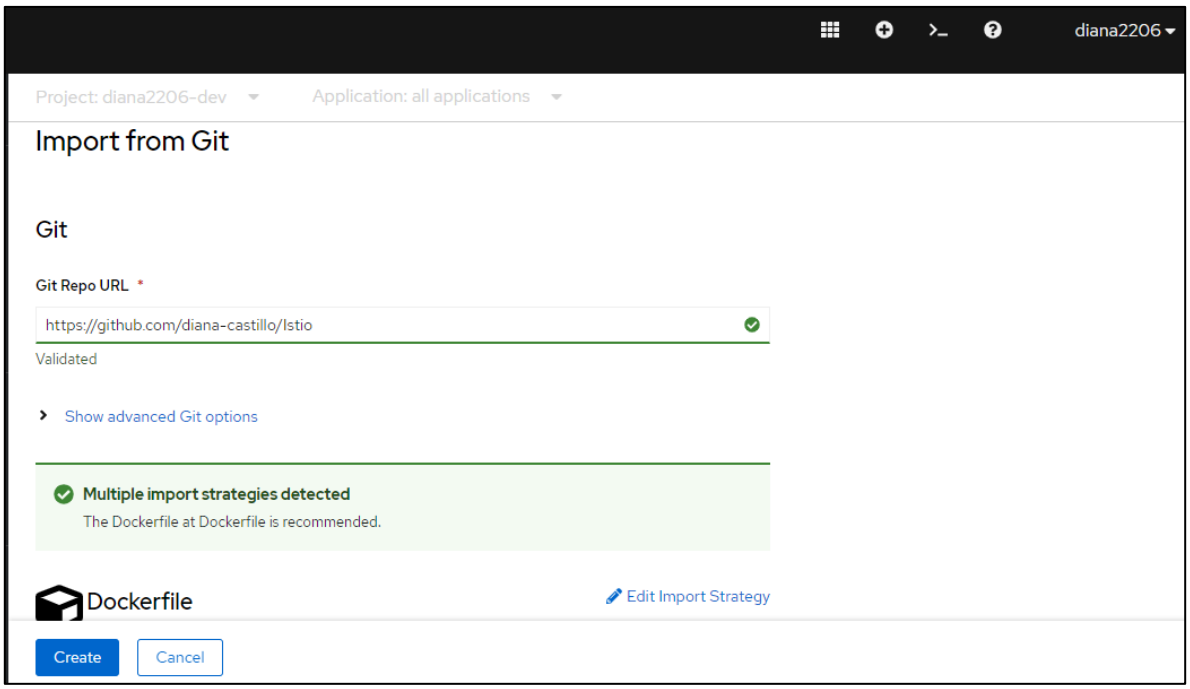


Utilicé Openshift con Developer Sandbox, el cual es un entorno de desarrollo basado en OpenShift que brinda la capacidad de crear rápidamente prototipos de aplicaciones basadas en Kubernetes. Es un entorno privado en un clúster compartido que ya está configurado con un conjunto de herramientas de desarrollo, de modo que la preparación se realiza antes de que los desarrolladores “entren” en el entorno.

Para ello creé una cuenta en Red Hat y entré a la opción de Developer Sandbox, y así se creó un espacio con mi cuenta:

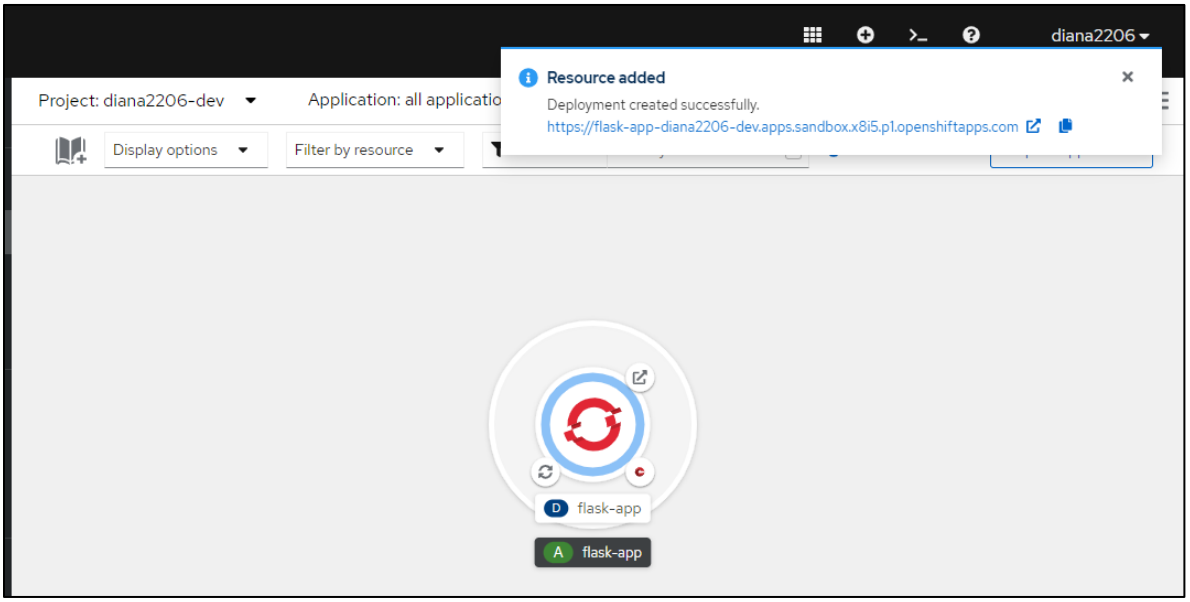


En la opción de +Add hay varias alternativas para lanzar aplicaciones, yo escogí la opción de importar un repositorio de Git para construir y desplegar una aplicación, pues ya tenía listo el repositorio en Github de la actividad anterior. Al colocar el link, se valida que en ese repositorio haya un Dockerfile y un YAML para poder construir la app correctamente:

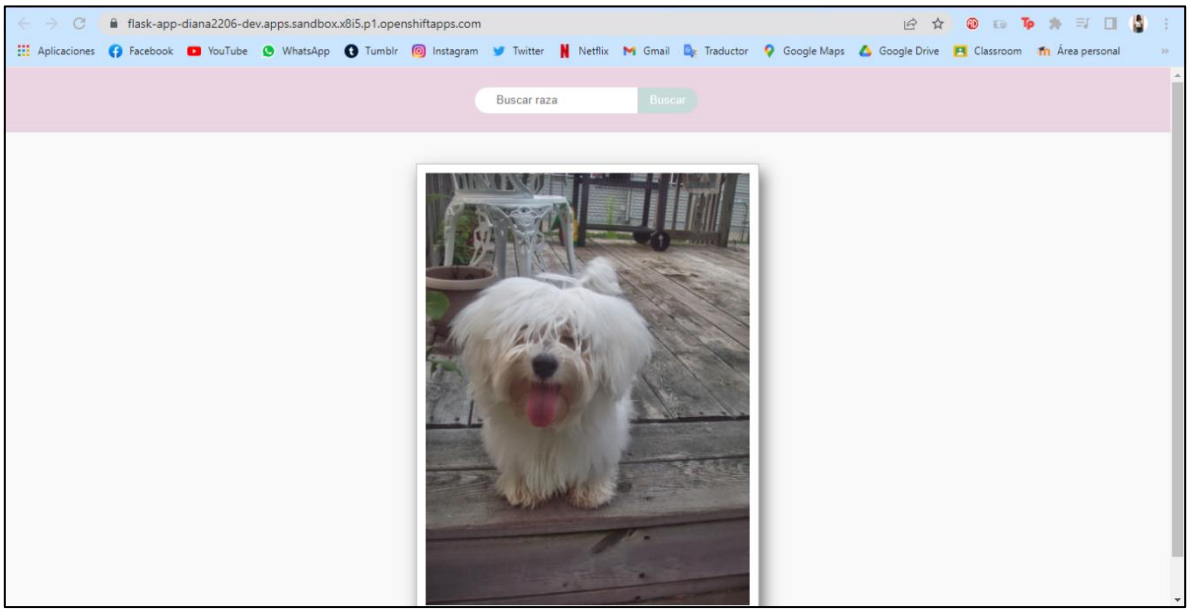


También se pide que se le ponga un nombre a la aplicación y que se seleccione un puerto a utilizar, en este caso se puso el 8080 como predeterminado ya que mi aplicación indica que correrá en el puerto 8080.

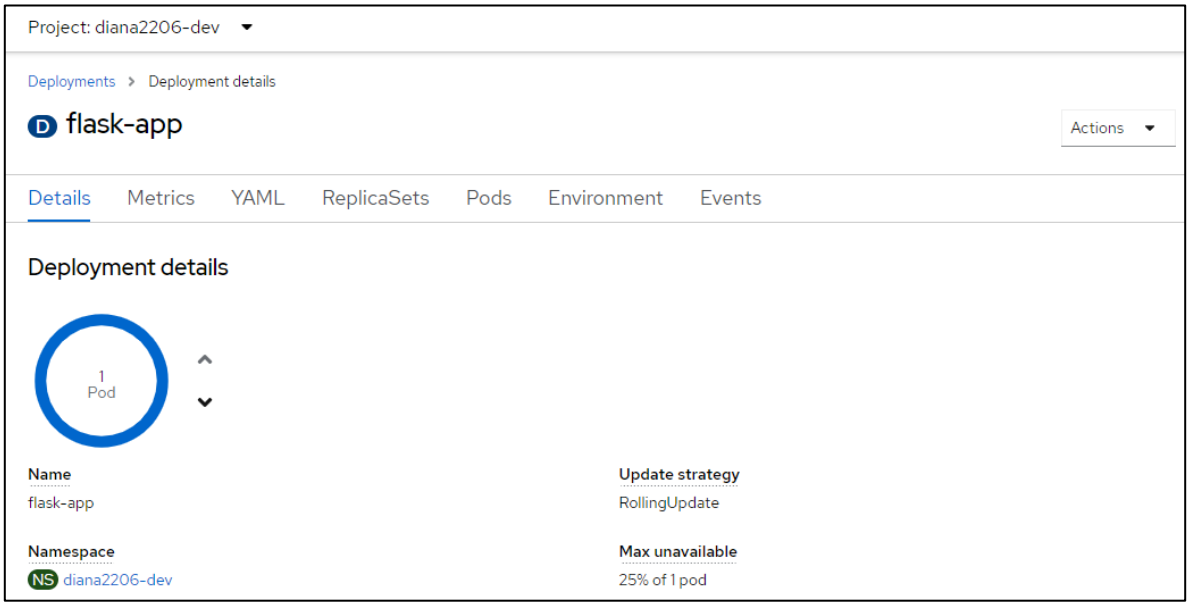
Al crearla aparece lo siguiente en Topology, que indica el estado de la aplicación y también un enlace en el cual se va a mostrar la app:



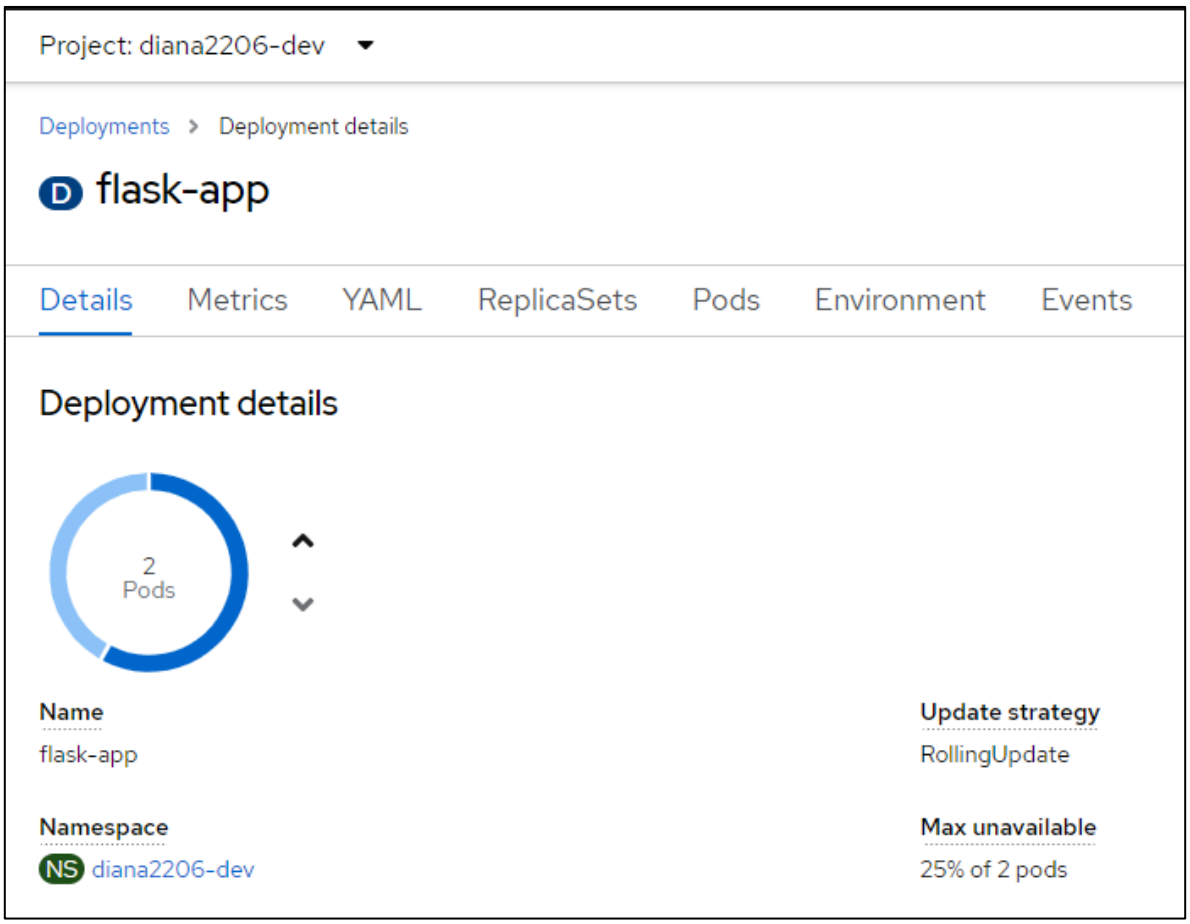
Al entrar al enlace se puede apreciar la aplicación que tenía en el repositorio de Github:



Al dar clic en el círculo de flask-app que tiene una A de Application, aparecen los detalles del Deployment, como el número de pods, en este caso hay uno:

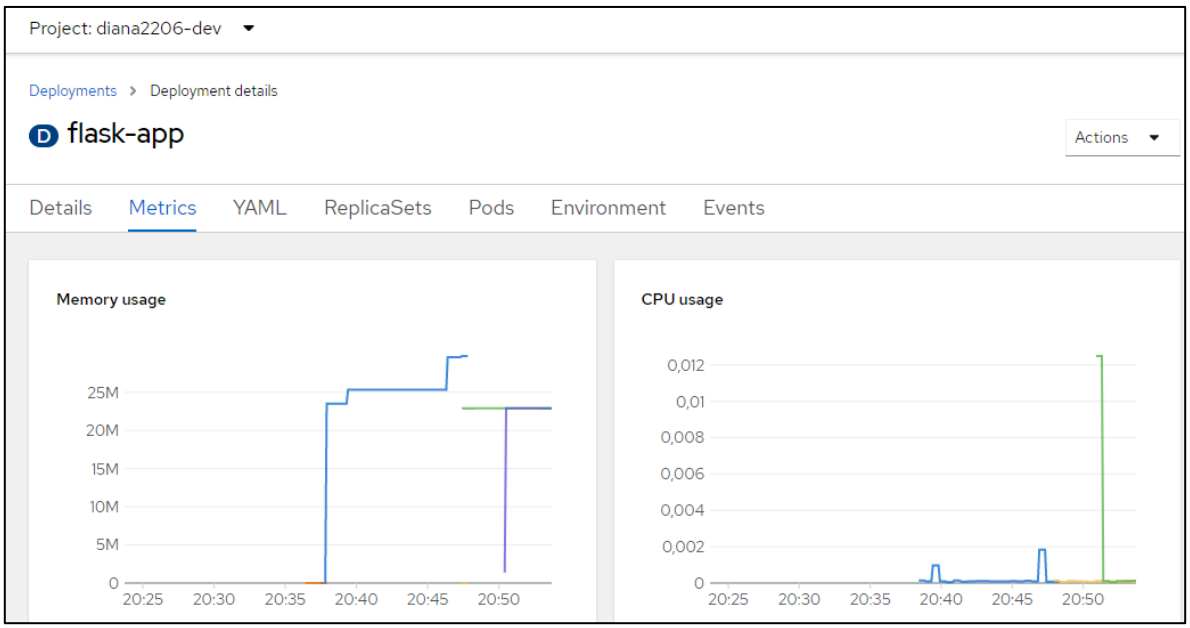


Con las flechas se pueden aumentar o disminuir el número de pods, en este caso los aumenté para que sean dos y cambian los colores conforme se va creando el pod, azul cielo significa que está iniciando y azul fuerte significa que ya está corriendo:

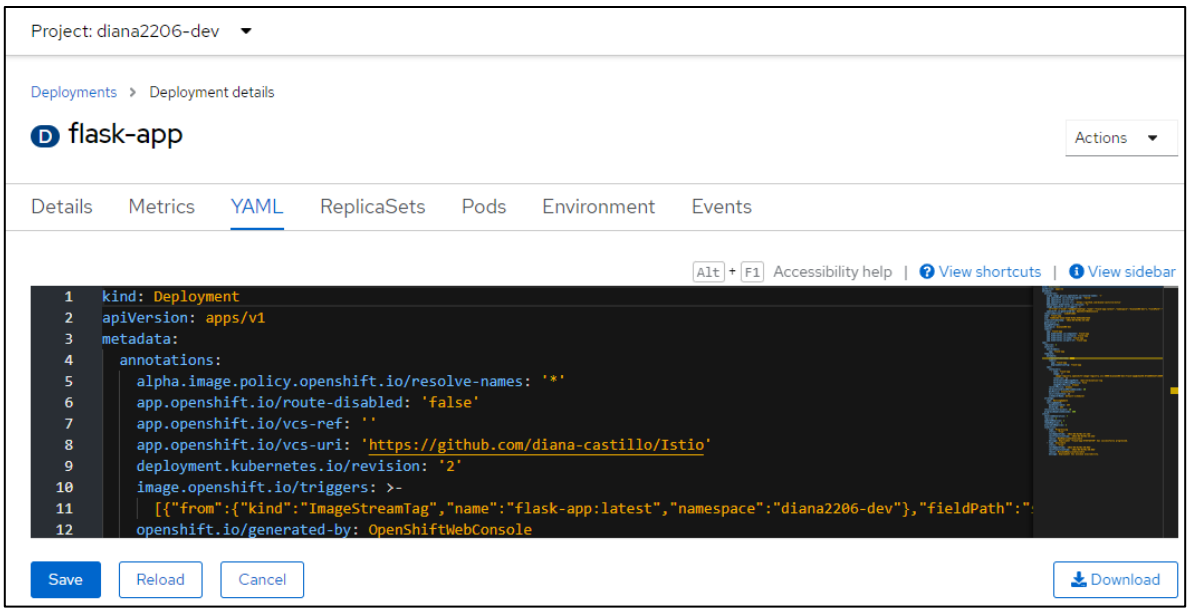


También se muestran los detalles de la imagen creada en openshift:

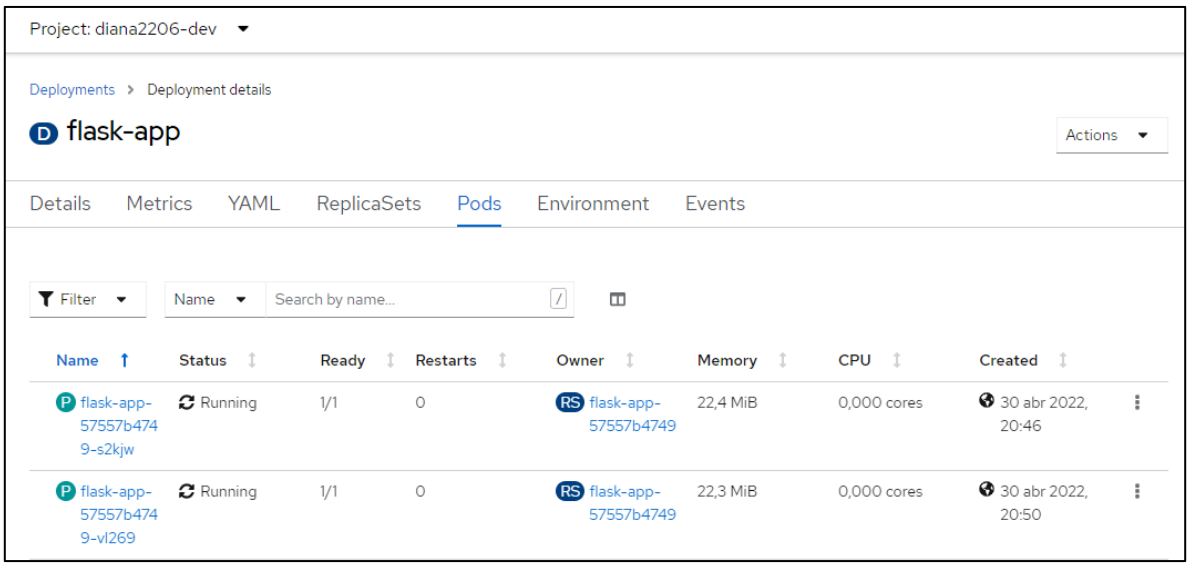
Hay un apartado de métricas en el cual se observan ciertas métricas de los pods como el uso de memoria, uso de CPU, sistema de archivos y red. En la imagen aparece un color distinto por cada pod, y como había agregado dos pods más y luego los eliminé aparecen cuatro colores distintos, pero los que ya eliminé ya no van en aumento en las gráficas:



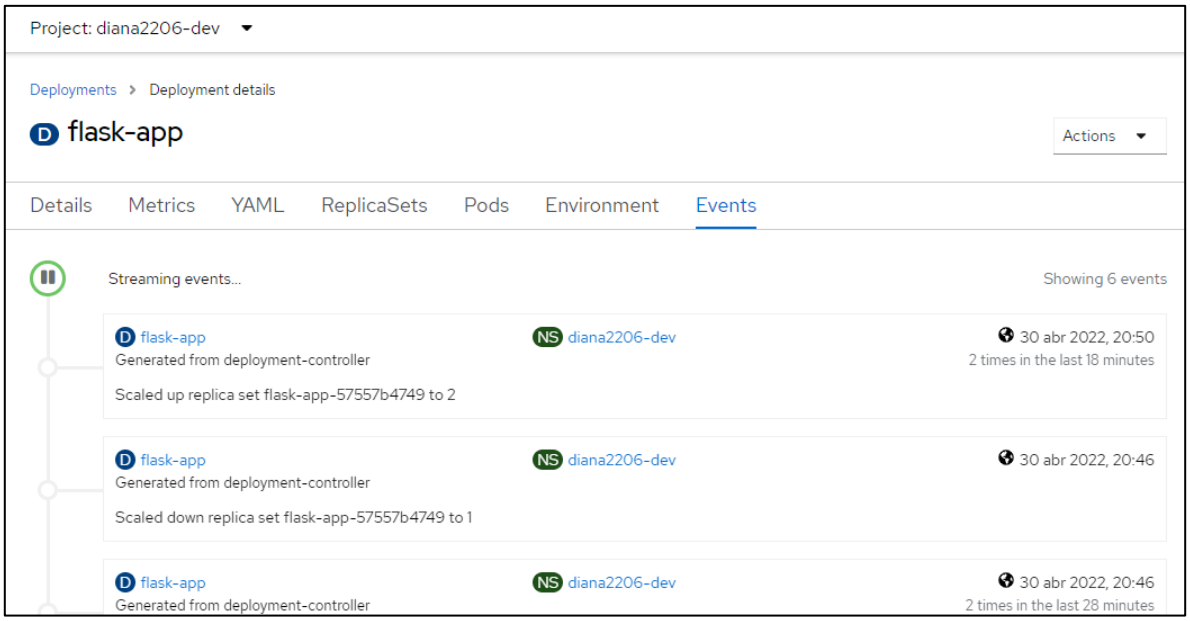
En el siguiente apartado aparece el YAML del Deployment, y se observa que contiene el url de mi repositorio en Github:



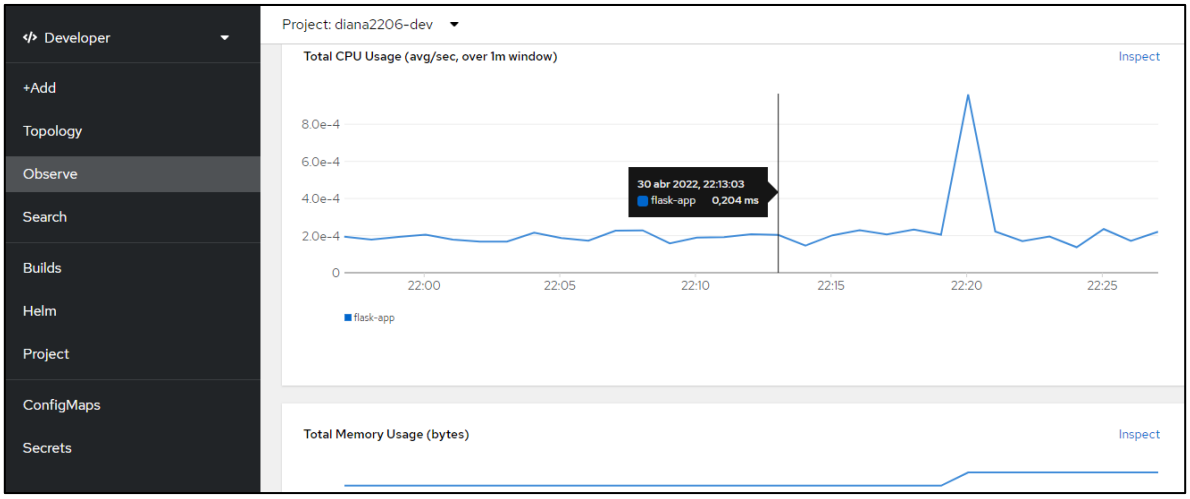
En Pods aparecen los detalles de cada pod de la aplicación:



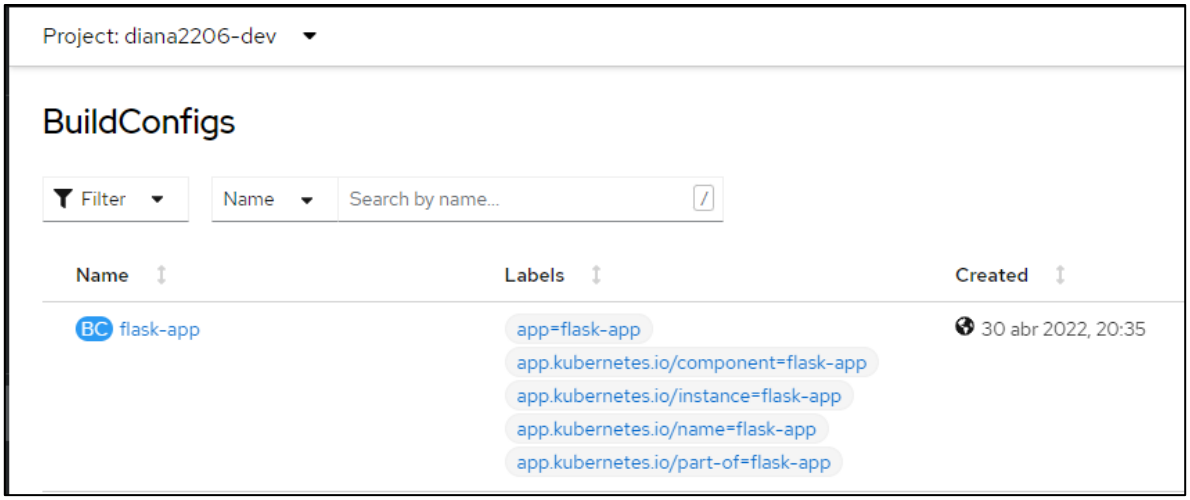
También se enlistan los eventos que van sucediendo:



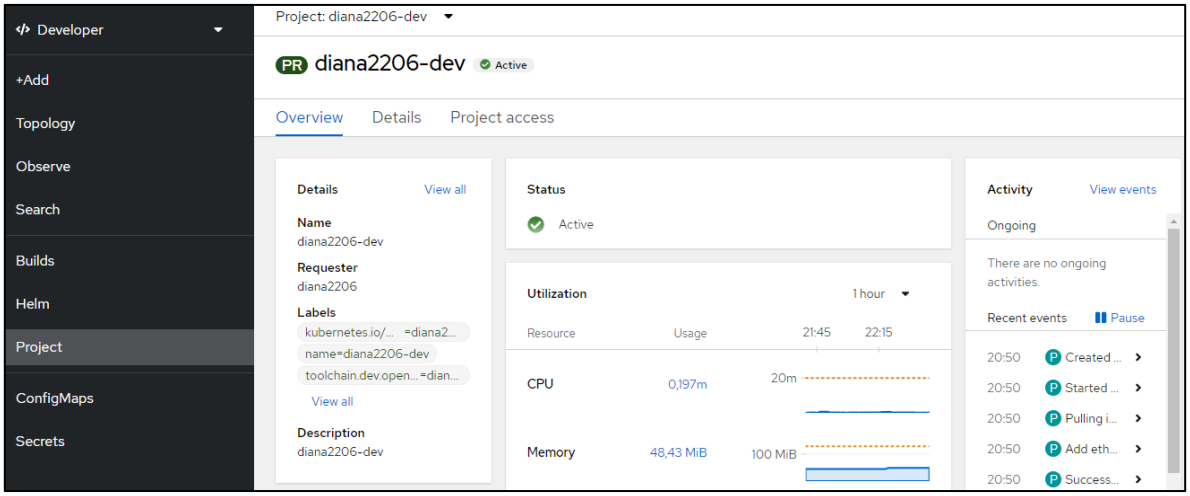
En el apartado de Observe se pueden ver las métricas de la aplicación en general:



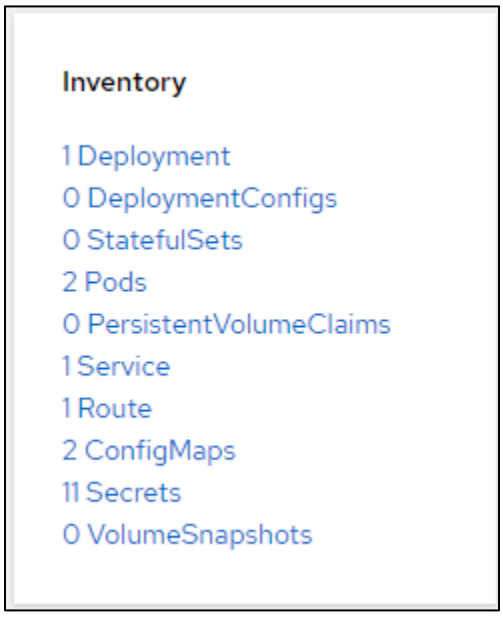
En Builds se enlistan las aplicaciones que se tengan en el proyecto seleccionado, en este caso sólo se tiene la de flask-app:



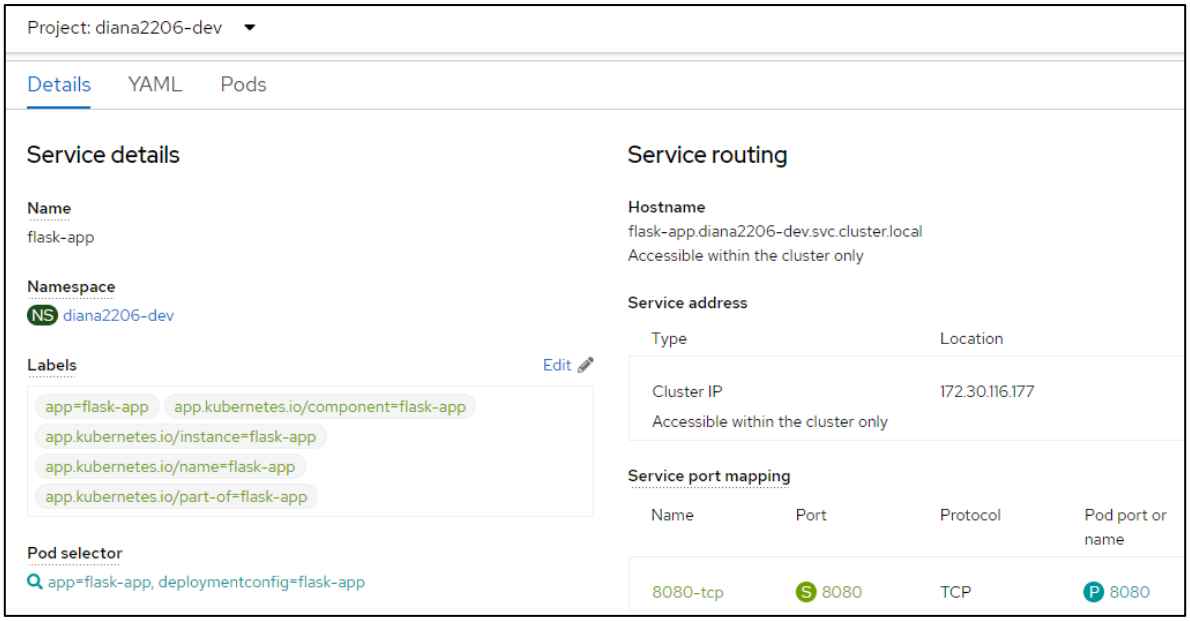
En Project se ven los detalles del namespace donde está almacenada la aplicación:



Así como el inventario de todo lo que contiene:



Al presionar en Service aparecen los detalles del servicio:



Código en GitHub: https://github.com/diana-castillo/Istio.git

Enlace a la aplicación en Openshift:

https://flask-app-diana2206-dev.apps.sandbox.x8i5.p1.openshiftapps.com/

Conclusiones:

Openshift me pareció una herramienta bastante útil y sencilla de utilizar desde su entorno online Sandbox, y tiene múltiples funcionalidades y maneras de observar cada parte de las aplicaciones que contienen los namespace, lo que me pareció de mucha ventaja es la manera tan rápida en la que crea y construye las imágenes a partir de un repositorio de Git que cuente con un Dockerfile y un YAML, eso facilita mucho el despliegue de una aplicación ya existente, además de que te da un enlace en el cual correrá la aplicación. Otro punto interesante es que desde ahí se pueden crear más pods solo con un clic, y que se pueden almacenar más aplicaciones en un proyecto, esto es muy conveniente para las aplicaciones grandes, pues se tiene todo en un mismo entorno y es sencillo de manejar y entender. Sin duda es una muy buena opción de distribución de Kubernetes.

Referencias:

Coronado, A. [Albert Coronado]. (2019, 25 noviembre). *Diferencias entre Docker, Kubernetes, Apache Mesos y OpenShift* [Vídeo]. YouTube. https://www.youtube.com/watch?v=pNlbdrdBxyc

Muñoz, J. D. [OpenWebinars]. (2019, 8 junio). *Cómo empezar a utilizar OpenShift* [Vídeo]. YouTube. https://www.youtube.com/watch?v=HPEPPArfOfk

Alternativas a OpenShift. (2022, 18 febrero). IONOS Digitalguide. Recuperado 30 de abril de 2022, de https://www.ionos.mx/digitalguide/servidores/know-how/openshift-alternatives/

Developer Sandbox un entorno de desarrollo para OpenShift de Red Hat para impulsar el desarrollo de aplicaciones de Kubernetes. (s. f.). Desde Linux. Recuperado 30 de abril de 2022, de https://blog.desdelinux.net/developer-sandbox-un-entorno-de-desarrollo-para-openshift-de-red-hat-para-impulsar-el-desarrollo-de-aplicaciones-de-kubernetes/

Kubernetes alternatives. (2022, 11 enero). IONOS Digitalguide. Recuperado 30 de abril de 2022, de https://www.ionos.mx/digitalguide/servidores/know-how/kubernetes-alternative/