DESPLEGAR APLICACIONES EMPRESARIALES JAVA UTILIZANDO OKD: PLATAFORMA IMPULSADA POR RED HAT OPENSHIFT

POR:

CRISTIAN CAMILO GALLEGO RANGEL

RICARDO BETANCOURTH BOLIVAR

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

FACULTAD DE INGENIERÍA

PEREIRA

2020

DESPLEGAR APLICACIONES EMPRESARIALES JAVA UTILIZANDO OKD: PLATAFORMA IMPULSADA POR RED HAT OPENSHIFT

POR:

CRISTIAN CAMILO GALLEGO RANGEL

RICARDO BETANCOURTH BOLIVAR

TRABAJO DE GRADO

MONOGRAFÍA

DOCUMENTO PROYECTO DE GRADO PRESENTADO COMO REQUISITO PARCIAL PARA OPTAR POR EL TÍTULO DE INGENIERO DE SISTEMAS Y COMPUTACIÓN

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

FACULTAD DE INGENIERÍA

PEREIRA

2020

Nota de aceptación:

_____

_____

_____

_____

_____

Firma del presidente del jurado

_____

Firma jurado

_____

Firma jurado

Pereira 18 de diciembre del 2020

DEDICATORIAS

1. *Para mi madre Margarita María Rangel, a mi padre Juan Manuel Gallego, mis abuelos Luis Alfonso Rangel y Rosalba Tamayo quienes me brindaron su apoyo incondicional y me permitieron llegar hasta cumplir la meta.*

2. *Para mi madre Olga Lucia Bolivar Castro, a mi padre Jhon Betancourth Giraldo y a mis abuelos Amparo Castro y Francisco Bolivar que con su gran apoyo y motivación se logró este reto.*

AGRADECIMIENTOS

1. *A Dios gracias por ayudarnos en el día a día para el cumplimiento de este arduo trabajo.*

2. *Al Ingeniero Cesar Augusto Díaz Arriaga por la disponibilidad, acompañamiento y asesoría en el desarrollo de esta monografía.*

3. *A nuestro director Carlos Augusto Meneses Escobar por su total atención y apoyo en esta monografía.*

4. *A nuestras familias y allegados que aportaron su granito de arena para motivarnos y ver cumplidas nuestras metas.*

5. *Y a la Universidad Tecnológica de Pereira, a la Facultad de Ingeniería de Sistemas y Computación y a todos nuestros maestros por el tiempo dedicado a compartirnos sus conocimientos y que fueron de mucho apoyo en nuestro desarrollo académico.*

TABLA DE CONTENIDO

TABLA DE ILUSTRACIONES

RESUMEN

La presente investigación se refiere a los pasos que las empresas desarrolladoras de software deben seguir para realizar la correcta compilación, pruebas y despliegue automático de sus aplicaciones con el fin de disminuir los tiempos de espera y eliminar el despliegue de manera manual.

Teniendo en cuenta lo anterior en este proyecto se hará uso de la herramienta de OKD Red Hat Open Shift para dar una visión del despliegue de aplicaciones Java en Cloud utilizando el motor de contenerización Docker y Kubernetes lo cual nos ofrece múltiples ventajas, funcionalidades y características como por ejemplo elasticidad, pago por uso, clústeres de servidores, permite desplegar aplicaciones en diferentes entornos, migraciones a la nube pública e híbrida, aplicaciones en contenedores, uso de microservicios y también ofrece un conjunto de herramientas de código abierto basadas en contenedores para habilitar la transformación digital, lo que acelera el desarrollo de aplicaciones y optimiza el uso de la infraestructura.

Se considera que las plataformas de aplicaciones en la nube constituyen la base del portafolio completo de plataformas como servicio (PaaS) sustentado por una gran variedad de servicios en la nube que eliminan la complejidad de desarrollar aplicaciones. A medida que aumenta la demanda de aplicaciones, el principal impulsor de negocio para las soluciones de PaaS es la agilidad con la cual un desarrollador puede tomar un concepto y entregar valor al usuario.

ABSTRACT

This research refers to the steps that software development companies must follow to carry out the correct compilation, testing and automatic deployment of their applications in order to reduce waiting times and eliminate manual deployment.

Taking into account the above, this project will use the OKD Red Hat Open Shift tool to give a vision of the deployment of Java applications in the Cloud using the Docker and Kubernetes containerization engine, which offers us multiple advantages, functionalities and characteristics such as elasticity, pay-per-use, server clusters, allows to deploy applications in different environments, migrations to public and hybrid cloud, applications in containers, use of microservices and also offers a set of open source tools based on containers to enable digital transformation, accelerating application development and optimizing the use of infrastructure.

Cloud application platforms are considered to form the foundation of the complete portfolio of platforms as a service (PaaS) supported by a wide variety of cloud services that eliminate the complexity of developing applications. As the demand for applications increases, the main business driver for PaaS solutions is the agility with which a developer can take a concept and deliver value to the user.

CAPITULO 1 INTRODUCCIÓN

En este capítulo se plantea el problema y se justifica la investigación acerca de los pasos que las empresas desarrolladoras de software deben realizar para poder hacer el despliegue automatizado de aplicaciones empresariales Java en un entorno Cloud para disminuir los tiempos de espera de pasar una aplicación de desarrollo a producción.

1.1. PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA

1.1.1. FORMULACIÓN Y JUSTIFICACIÓN DEL PROBLEMA

En la actualidad las empresas de desarrollo de software debido a las metodologías ágiles y las necesidades del mercado deben reducir los tiempos de pasar la aplicación del código a producción; ya que al momento de hacer el despliegue de sus aplicaciones en la nube, deben realizar la compilación, pruebas, ajuste y despliegue de manera manual.

El problema surge porque al momento de finalizar cualquier cambio en el código fuente por parte de los desarrolladores, se requiere hacer la actualización de la aplicación en un entorno de producción, comúnmente se presentan situaciones en las que el desarrollador sube cambios con la modificación del código y por falta de comunicación entre el equipo de desarrollo y el equipo de operaciones se sube la versión que no es, eso causa que se pierda mucho tiempo y genera malestar en los usuarios finales. Adicionalmente los desarrolladores no están utilizando el tiempo de la forma más eficiente porque no tienen un proceso automatizado de despliegue de sus aplicaciones

El uso de esta herramienta permite una implementación más fácil pues solo se debe hacer clic en un botón o introducir un comando git push; esto reduce o descarta muchos problemas de administración de sistemas relacionados con el diseño y la implementación de aplicaciones en contenedores. La herramienta OKD de Red Hat Open Shift garantiza estandarizar los flujos de trabajo de los desarrolladores, admitir varios entornos y permitir la integración constante junto con la gestión automatizada de las versiones.

1.2. OBJETIVOS

1.2.1. OBJETIVO GENERAL

Crear un documento que ayude a las empresas de desarrollo de software para que puedan realizar el despliegue automatizado de sus aplicaciones empresariales JAVA en la nube utilizando como plataforma OKD.

1.2.2. OBJETIVOS ESPECÍFICOS

1. Mostrar cuáles son los diferentes tipos de aplicaciones empresariales JAVA que podemos subir a la nube.
2. Mostrar las ventajas que nos ofrece la herramienta OKD de RedHat Open Shift.
3. Crear un ambiente de desarrollo para el despliegue de aplicaciones en la plataforma OKD.
4. Crear un ambiente de producción para el despliegue de aplicaciones en la plataforma OKD.
5. Analizar herramientas para ser utilizadas en la fase de pruebas.
6. Analizar herramientas para ser utilizadas en la fase de despliegue.
7. Realizar despliegue de una aplicación de ejemplo en OKD.
8. Elaborar conclusiones sobre el resultado de las pruebas.

1.3. MEDOTODOLOGÍA PROPUESTA PARA EL DESARROLLO DE LA MONOGRAFÍA

A continuación, se desarrollará una metodología en cascada la cual tiene el siguiente orden convirtiéndose en la guía de desarrollo de la monografía.

- Mostrar cuáles son los diferentes tipos de aplicaciones empresariales JAVA que podemos subir a la nube.
- Mostrar las ventajas que nos ofrece la herramienta OKD Open Shift.
- Crear un ambiente de desarrollo para el despliegue de aplicaciones en la plataforma OKD.
- Crear un ambiente de producción para el despliegue de aplicaciones en la plataforma OKD.
- Analizar herramientas para ser utilizadas en la fase de pruebas.
- Analizar herramientas para ser utilizadas en la fase de despliegue.
- Realizar despliegue de una aplicación de ejemplo en OKD.
- Elaborar conclusiones sobre el resultado de las pruebas.

CAPITULO 2 MARCO TEÓRICO

Existen diferentes tipos de aplicaciones empresariales Java las cuales se pueden desplegar en la nube con la ayuda de la herramienta RedHat Open Shift. En este capítulo se conocerán algunas de ellas y también se profundizará sobre las ventajas que nos ofrece la herramienta RedHat Open Shift.

2.1. TIPOS DE APLICACIONES EMPRESARIALES JAVA QUE SE PUEDEN SUBIR A LA NUBE

Actualmente el lenguaje de programación Java proporciona 4 grandes plataformas (JavaEE, 2012, pág. 1) pero solo 3 ediciones son las más importantes como lo muestra la ilustración 1.

Ilustración 1: Principales plataformas del lenguaje de programación Java.

- **Java ME** (*Java Micro Edition*) para el desarrollo de aplicaciones Java en pequeños dispositivos (móviles, tarjetas de crédito, bluetooth, televisores o reproductores blu-ray)
- **Java SE** (*Java Standard Edition*) para el desarrollo de aplicaciones de escritorio en ordenadores personales.

- **Java EE** (Java Enterprise Edition) para el desarrollo de aplicaciones distribuidas (cliente-servidor o con múltiples capas) como aplicaciones web o servicios web.

Todas las plataformas consisten en una Máquina Virtual Java (JVM) y en una interfaz de programación de aplicaciones (API). Todas las plataformas mantienen las ventajas del lenguaje de programación Java: independencia de la plataforma, potencia, estabilidad, facilidad de desarrollo y seguridad.

Sin embargo en este proyecto se tratará con la plataforma Java EE. La plataforma Java EE proporciona un API y un entorno de ejecución para desarrollar y ejecutar aplicaciones en red de gran escala, multi-capa, escalables, fiables y seguras. Estas aplicaciones se denominan *enterprise* (traducido a veces por corporativas o empresarial) porque se diseñan para resolver los problemas de grandes empresas (JavaEE, 2012).

Algunas de las características de la plataforma Java EE son:

1. Java EE facilita el desarrollo de aplicaciones a gran escala
2. Java EE se basa en Java SE. Proporciona funcionalidades como aplicaciones web y Servlets.
3. Java EE es una aplicación estructurada con capas separadas de cliente, negocio y empresa.
4. Se utiliza principalmente para desarrollar aplicaciones web.
5. Adecuado para desarrolladores Java experimentados que crean aplicaciones para toda la empresa.

2.2. VENTAJAS DE LA HERRAMIENTA OKD DE RED HAT OPENSHIFT

OKD es la distribución ascendente de Kubernetes integrada en Red Hat OpenShift. Es una distribución de Kubernetes optimizada para el desarrollo continuo de aplicaciones y la implementación de múltiples clientes (multi-tenant). OKD agrega herramientas centradas en las operaciones y el desarrollador además de Kubernetes para permitir el desarrollo rápido de aplicaciones, la implementación y el escalado sencillo, y el mantenimiento del ciclo de vida a largo plazo para equipos pequeños y grandes (OpenShit, 2020).

Las principales ventajas que nos ofrece la herramienta OKD son:

1. Una distribución completamente automatizada de Kubernetes en todas las principales nubes y bare metal, OpenStack y otros proveedores de virtualización.
    - Cree aplicaciones fácilmente con descubrimiento de servicios integrados y almacenamiento persistente.
    - Escale aplicaciones rápida y fácilmente para manejar períodos de mayor demanda.
    - Soporte para alta disponibilidad automática, equilibrio de carga, verificación de estado y conmutación por error.
    - Acceso al *Operator Hub* para ampliar Kubernetes con nuevas capacidades de ciclo de vida automatizadas.
2. Consola y herramientas centradas en el desarrollador para crear aplicaciones en contenedores en Kubernetes.
    - Envíe el código fuente a su repositorio de Git e implemente automáticamente aplicaciones en contenedores.

- Consola web y cliente de línea de comandos para crear y monitorear aplicaciones.

3. Administración y gestión centralizadas de toda una pila, equipo u organización.
    - Cree plantillas reutilizables para los componentes de su sistema e impleméntelas iterativamente a lo largo del tiempo.
    - Implemente modificaciones a las pilas de software en toda su organización de forma controlada.
    - Integración con sus mecanismos de autenticación existentes, incluidos LDAP, Active Directory y proveedores públicos de OAuth como GitHub.
4. Soporte multi-tenant, incluido el aislamiento de equipos y usuarios de contenedores, compilaciones y comunicación de red.
    - Permita que los desarrolladores ejecuten contenedores de forma segura con controles detallados en producción.
    - Limite, rastree y administre a los desarrolladores y equipos en la plataforma.
5. Registro de imágenes de contenedor integrado, equilibrio automático de carga en el borde y monitoreo de espectro completo con Prometheus (RedHat, 2020).

OKD se ejecuta con la siguiente política de seguridad de forma predeterminada:

1. Los contenedores se ejecutan como un usuario único no root que está separado de otros usuarios del sistema.
    - No pueden acceder a los recursos del host, ejecutar privilegios o convertirse en root.
    - Se les dan límites de CPU y memoria definidos por el administrador del sistema.

- Cualquier almacenamiento persistente al que accedan estará bajo una etiqueta SELinux única, que evita que otros vean su contenido.
- Estas configuraciones son por proyecto, por lo que los contenedores en diferentes proyectos no pueden verse entre sí de forma predeterminada.

2. Los usuarios habituales pueden ejecutar Docker, código fuente y compilaciones personalizadas.
    - De forma predeterminada, las compilaciones de Docker pueden (y a menudo lo hacen) ejecutarse como root. Puede controlar quién puede crear compilaciones de Docker a través de los recursos builds/docker y builds/custom de políticas personalizadas.
3. Los usuarios habituales y los administradores de proyectos no pueden cambiar sus cuotas de seguridad.

OKD está diseñado para ejecutar cualquier carga de trabajo de Kubernetes. También ayuda a crear y desarrollar aplicaciones en contenedores a través de la consola del desarrollador. Para una experiencia más sencilla al ejecutar su código fuente, Source-to-Image (S2I) permite a los desarrolladores simplemente proporcionar un repositorio de fuentes de aplicaciones que contiene código para construir y ejecutar. Funciona combinando una imagen de contenedor habilitada para S2I existente con la fuente de la aplicación para producir una nueva imagen ejecutable para su aplicación.

Algunas de las imágenes disponibles incluyen:

- Ruby
- Python

- Java
- Node.js
- PHP
- Perl
- WildFly
- MySQL
- MongoDB
- PostgreSQL
- MariaDB

La siguiente ilustración muestra el ciclo de vida básico de OKD:

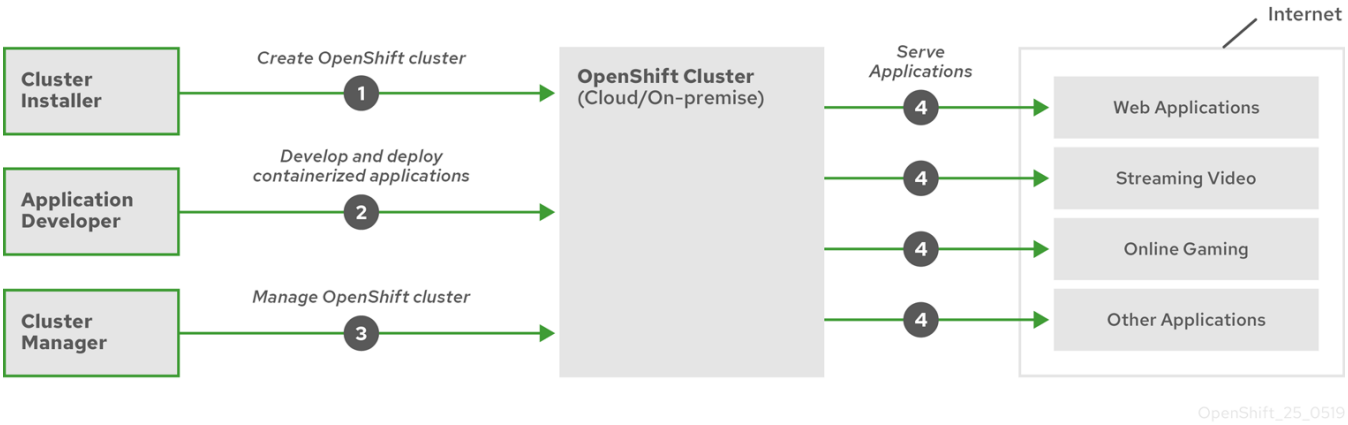

Ilustración 2: Descripción general de OKD de alto nivel

CAPITULO 3 DESARROLLO DE LA MONOGRAFÍA

A continuación se hará el desarrollo de los demás temas de la monografía donde se profundiza sobre cómo crear el ambiente de desarrollo y pruebas, así como las herramientas necesarias para poder ejecutar dichos ambientes.

3.1. AMBIENTES DE DESARROLLO Y PRODUCCIÓN EN OKD

Antes de empezar a crear un ambiente de desarrollo se debe tener unos conocimientos previos a las siguientes herramientas:

- Comandos Linux
- Básico de contenedores
- Conocimientos Docker
- Conocimientos Kubernetes

Se debe tener en cuenta que MiniShift realiza casi todo el trabajo pesado por nosotros pero para poder sacarle todo el provecho a la herramienta se debe tener experiencia en las herramientas mencionadas anteriormente.

Para crear un ambiente de desarrollo/producción y poder usar MiniShift de forma local se debe tener lo siguiente:

- Se puede usar en los siguientes sistemas operativos: Windows, Linux o MAC OS en la página oficial de OKD se encuentra el enlace para descargar MiniShift en su versión más estable: https://github.com/minishift/minishift/releases.
- Minishift necesita una máquina virtual en la que se almacenará el clúster de OpenShift. Verifique que la máquina virtual que elija esté instalada y habilitada en su sistema operativo o en un contenedor Docker antes de configurar Minishift. Una vez que la máquina virtual se encuentre en funcionamiento, se requiere una configuración adicional para que Minishift funcione con la máquina virtual. Según el sistema operativo de este usando el modo local, puede elegir entre las siguientes máquinas virtuales nativas recomendados:

- Mac OS: hiperkit (https://docs.okd.io/3.11/minishift/getting-started/setting-up-virtualization-environment.html#setting-up-hyperkit-driver)
- Linux: KVM (https://docs.okd.io/3.11/minishift/getting-started/setting-up-virtualization-environment.html#setting-up-kvm-driver)
- Windows :Hyper-V (https://docs.okd.io/3.11/minishift/getting-started/setting-up-virtualization-environment.html#setting-up-hyper-v-driver)
- Para cualquier sistema operativo con el VirtualBox (https://docs.okd.io/3.11/minishift/getting-started/setting-up-virtualization-environment.html#setting-up-virtualbox-driver)

En cada enlace se encuentran los pasos de la página oficial de OKD para hacer sus respectivas instalaciones.

- Las características mínimas que debe tener la máquina local debe ser las siguientes:
  - 12 GB de RAM
  - 4 CPU Virtuales
  - 35 GB de disco duro

Al momento de iniciar el MiniShift con el comando *minishift start* muestra por consola los recursos que necesita para poder tener un solo cluster corriendo en la MV como se muestra en la ilustración 3. Luego continua su configuración y ajuste de máquina virtual como se puede ser en la ilustración 4 y por ultimo termina la configuración y debe entregarnos la URL de la consola con algunas credenciales como se muestra en la ilustración 5.

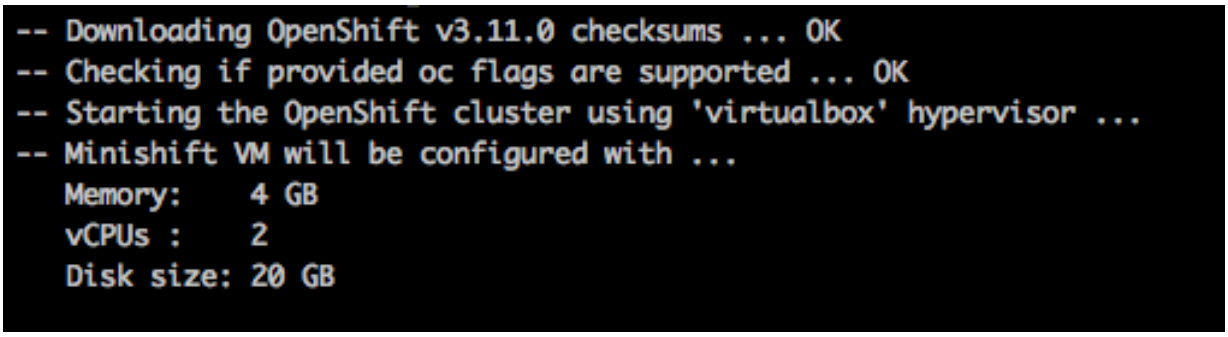

Ilustración 3: Configuración mínima de máquina virtual

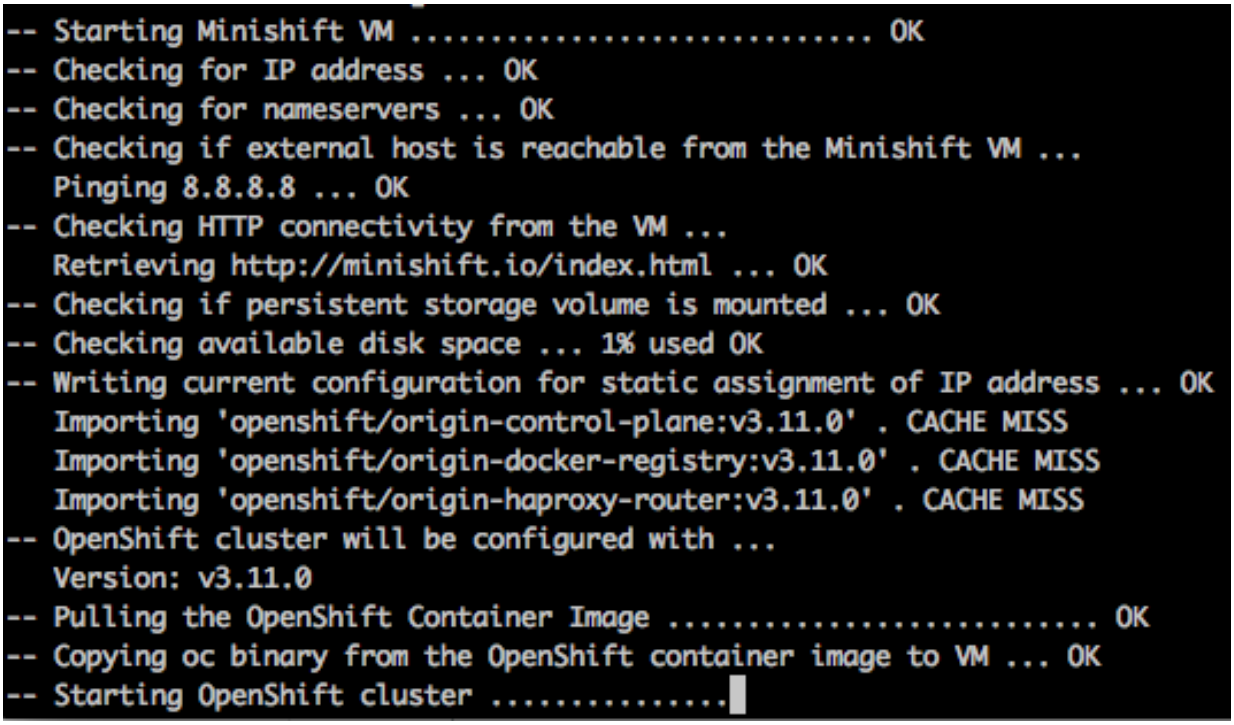

Ilustración 4: Continuación configuración máquina virtual

```
The server is accessible via web console at:
    https://192.168.99.100:8443/console

You are logged in as:
    User:     developer
    Password: <any value>

To login as administrator:
    oc login -u system:admin


-- Exporting of OpenShift images is occuring in background process with pid 6159.
```

Ilustración 5: Finalización configuración máquina virtual

Al momento de ingresar a la URL https://192.168.99.100:8443/console/catalog nos debe salir el tablero de control con los proyectos que tiene pero si es la primera vez solo debe hacer click en Create Project como se puede ver en la ilustración 6.

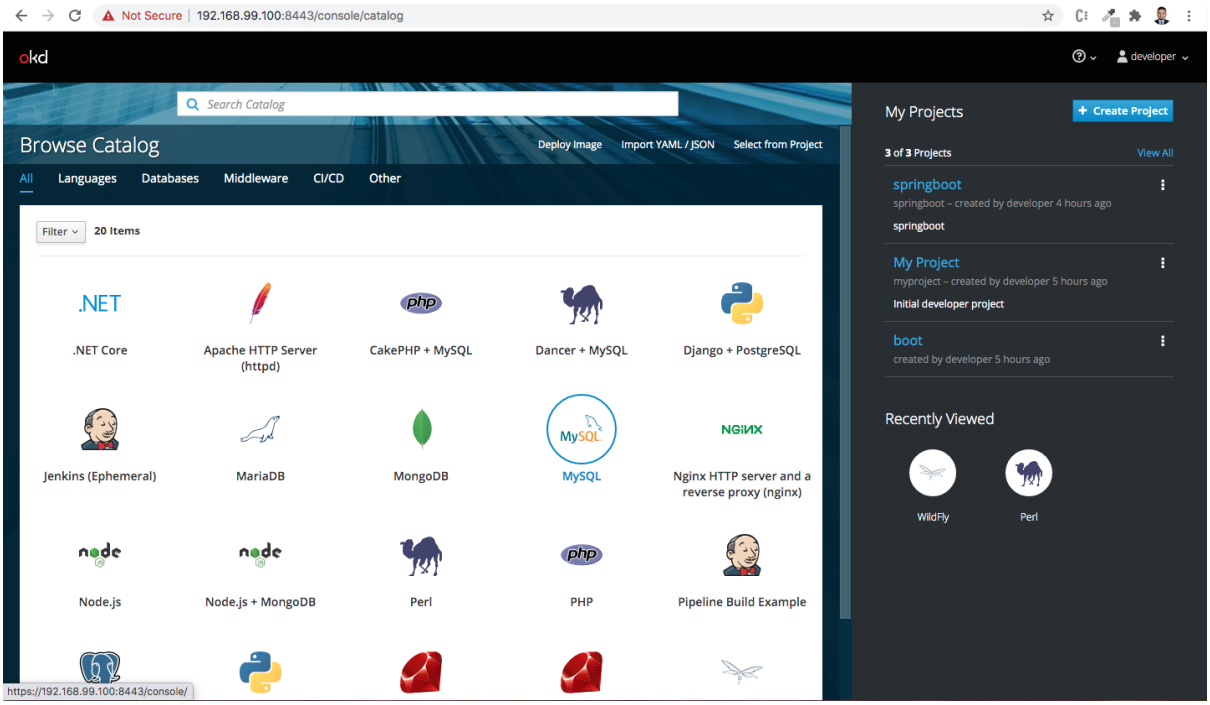

Ilustración 6: Catálogo de OKD - MiniShift

De esta forma ya tenemos todo el ambiente de desarrollo y producción listo para crear proyectos y asignar aplicaciones donde podemos tener el control total de los despliegues.

3.2.ANALISIS DE HERRAMIENTAS A SER UTILIZADAS EN LA FASE DE DESARROLLO Y DESPLIEGUE

La fase de desarrollo y despliegue (o producción) son muy similares en cuanto a las herramientas que se utilizan para el mantenimiento o monitoreo de las aplicaciones que están alojadas en OKD, es por eso que se analizará cada herramienta que será necesario usar en las fases de desarrollo y despliegue.

RedHat ofrece varios tipos de plataformas para empezar con OpenShift, como las versiones gratuitas y las versiones de pago. Una de las versiones gratuitas es la de OKD y una de las opciones para trabajar con OKD es a través de la herramienta MiniShift que es un ejecutable que nos permite instalar un clúster de OpenShift en una sola máquina en local (Muñoz, 2019).

MiniShift

Minishift es una herramienta que le ayuda a ejecutar OKD localmente al lanzar un clúster OKD de un solo nodo dentro de una máquina virtual. Con Minishift puede probar OKD o desarrollar con él, día a día, en su máquina local. Puede ejecutar Minishift en los sistemas operativos Windows, macOS y GNU / Linux (OKD, 2020). Está optimizada para flujos de trabajo en entornos de desarrollo y requiere un hipervisor para iniciar la máquina virtual en la que se proporciona el cluster. Minishift utiliza libmachine para la administración de la máquina, cluster up para el aprovisionamiento del cluster y OpenShift Origin para ejecutar el cluster (Hidalgo, 2017).

Hipervisores o Máquina Virtual

Minishift está soportado en varios hipervisores dependiendo del sistema operativo base que se utilice. Minishift requiere un hipervisor para ejecutar la máquina virtual que va a contener OpenShift. Las 2 opciones que se trabajarón en el desarrollo de la monografía son Oracle Virtual Box y KVM.

- **Oracle Virtual Box:** Oracle VM VirtualBox, el software de virtualización multiplataforma de código abierto más popular del mundo, permite a los desarrolladores entregar código más rápido, ya que pueden ejecutar múltiples sistemas operativos en un solo dispositivo. Los equipos de TI y los proveedores de soluciones usan VirtualBox para reducir los costes operativos y acortar el tiempo necesario para implementar aplicaciones de forma segura on-premises y en la nube (Oracle, 2020).
- **KVM:** es una solución para implementar virtualización completa con Linux. KVM permite ejecutar máquinas virtuales utilizando imágenes de disco que contienen sistemas operativos sin modificar. Cada máquina virtual tiene su propio hardware virtualizado: una tarjeta de red, discos duros, tarjeta gráfica, etc (Wikipedia, 2020).

CAPITULO 4 APLICACIÓN DE EJEMPLO

En este capítulo se mostrará los pasos necesarios para desplegar una aplicación Java EE en la herramienta OKD que es una plataforma como servicio. Se desplegará una aplicación de ejemplo que está alojada en un repositorio remoto GitHub.

4.1. DESPLEGAR UNA APLICACIÓN DE EJEMPLO EN LA PLATAFORMA OKD

Para hacer el ejemplo del despliegue de una aplicación vamos a utilizar un repositorio de GitHub de una calculadora de hipotecas para hacer todo el proceso de inicio a fin.

1. Se debe crear un nuevo proyecto en caso que sea necesario, se encuentra en la esquina superior derecha de la consola web como lo muestra la ilustración 7. O también se puede hacer por consola como se muestra en la ilustración 8.

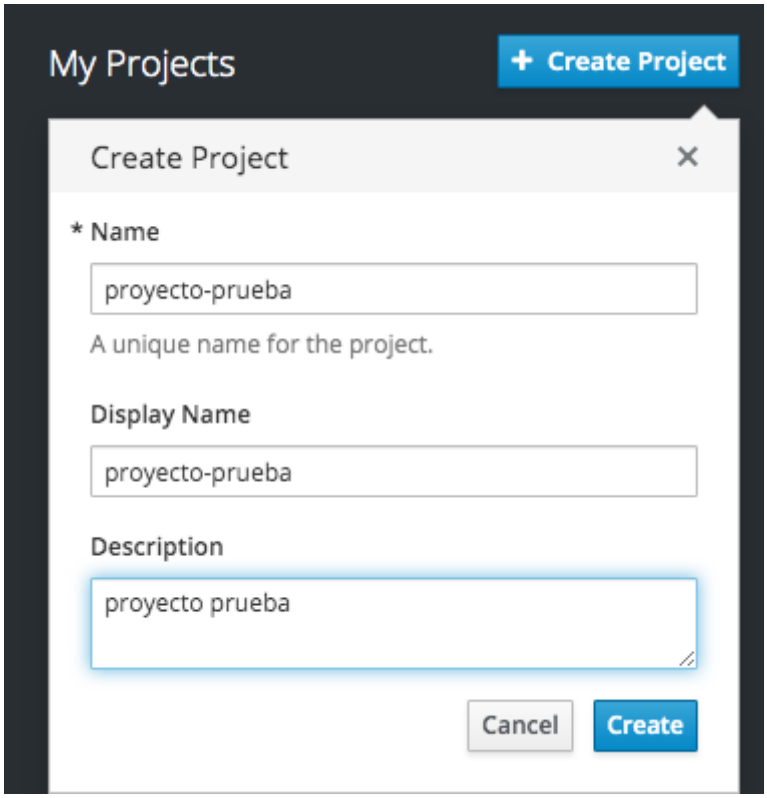

Ilustración 7: Formulario para crear nuevo proyecto

```
(base) ➜ Downloads oc project proyecto-prueba
Now using project "proyecto-prueba" on server "https://192.168.99.100:8443".
(base) ➜ Downloads
```

Ilustración 8: Creación de proyecto por consola cliente

2. Se debe tener una aplicación Java EE en un repositorio remoto como GitHub para poder a partir de ese código fuente crear las imágenes que serán desplegadas en OKD. En este ejemplo trabajaremos con una aplicación Java EE que utiliza el framework SpringBoot que se encuentra en https://github.com/ricardobet12/calculadoraCreditosH.

   Se utilizará el comando *oc new-app* para crear nuevas aplicaciones. Ejecutaremos el siguiente comando para usarlo con Red-Hat-Openjdk y colocar la ruta del Git donde se encuentre almacenado el código fuente de la aplicación de ejemplo como lo muestra la ilustración 9.

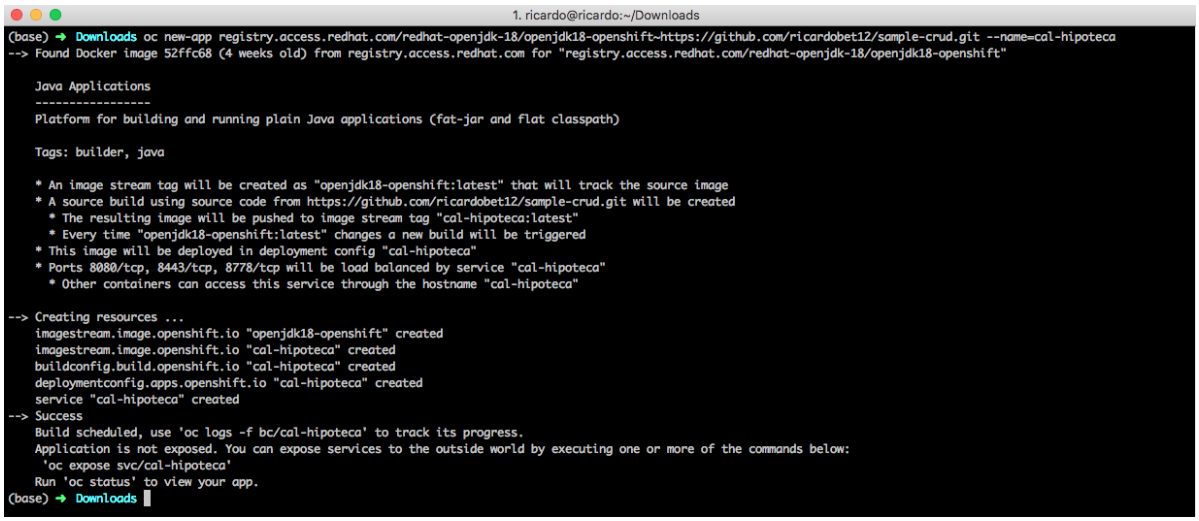

Ilustración 9: Agregando nueva aplicación enlazada con GitHub

Al momento de ir a la consola web de MiniShift se puede ver que ya se encuentra alojada la aplicación y que realiza varias procesos:

- Clonar el código fuente del repositorio de GitHub
- A partir del código fuente, crea una Imagen.
- Lo construye.
- Lo despliega.
- Le crea un servicio y una ruta para poder acceder desde internet.

En la ilustración 10 se puede observar la aplicación en estado Running

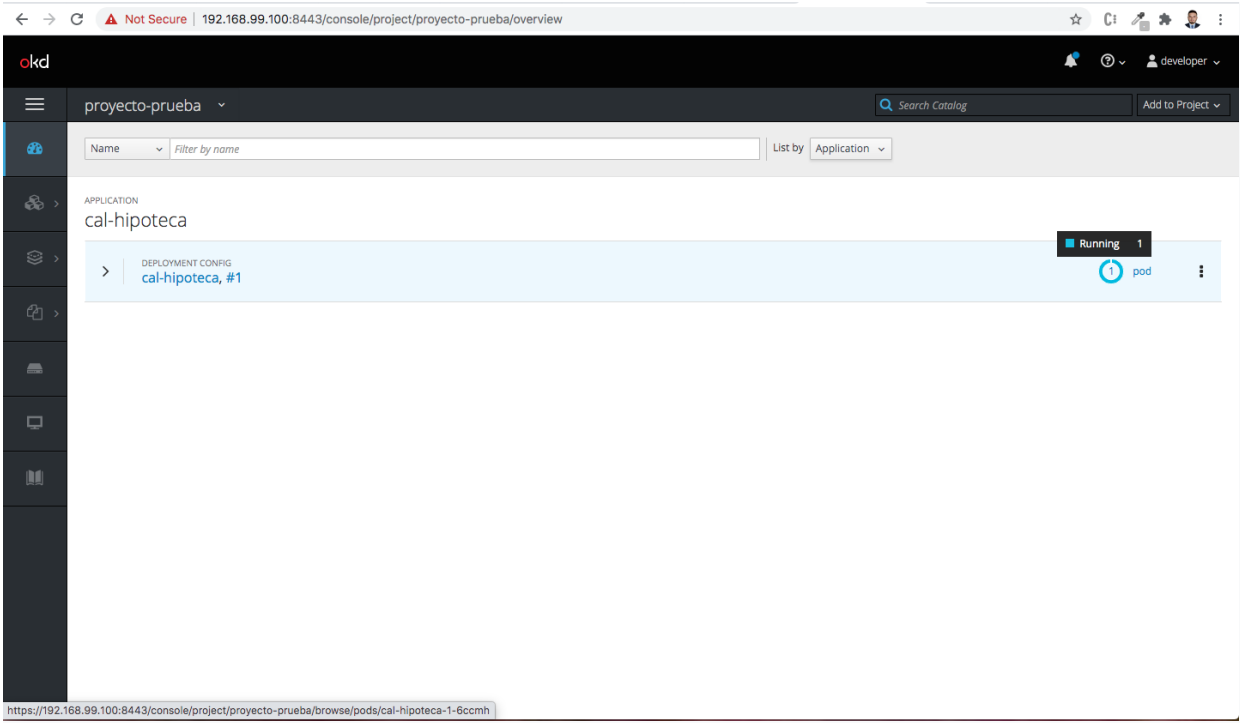

Ilustración 10: Aplicación en estado Running

3. En caso de que no le asigne una ruta pública de la aplicación, se puede crear con el comando que se muestra en la ilustración 11.

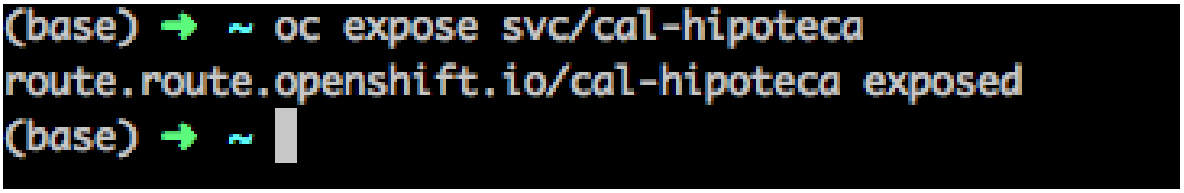

Ilustración 11: Creando una ruta a una aplicación por consola

CONCLUSIONES

Se puede llegar a la conclusión que al usar OKD estamos mejorando mucho no solamente el tiempo de desarrollo sino que también estamos garantizando mejor portabilidad y disponibilidad a las aplicaciones haciendo un seguimiento estricto usando los contenedores para contener la imagen de nuestros sistemas.

No es difícil preparar y cargar la configuración con la base de datos y los microservicios y luego implementarla en el nodo de OpenShift.

BIBLIOGRAFIA

Hidalgo, F. J. (06 de 2017). *OpenShift v3 Minishift*. Recuperado el 14 de 12 de 2020, de Develop Applications Locally in aContainerized OpenShift Cluster: https://dit.gonzalonazareno.org/gestiona/proyectos/2016-17/Minishift_Francisco_Moncayo.pdf

JavaEE, I. a. (2012). *jtech*. Recuperado el 13 de 12 de 2020, de jtech: http://www.jtech.ua.es/j2ee/restringido/pres/sesion01-apuntes.pdf

Muñoz, J. D. (08 de 06 de 2019). *openwebinar*. Recuperado el 14 de 12 de 2020, de openwebinar: https://openwebinars.net/blog/como-empezar-usar-openshift/

OKD, R. (2020). *okd*. Recuperado el 15 de 12 de 2020, de okd: https://docs.okd.io/3.11/minishift/index.html

OpenShit, R. H. (2020). *OKD*. Recuperado el 13 de 12 de 2020, de OKD: https://www.okd.io/

Oracle. (2020). *Oracle*. Recuperado el 15 de 12 de 2020, de Oracle VM VirtualBox: https://www.oracle.com/co/virtualization/virtualbox/

RedHat. (2020). *GitHub OKD Red Hat OpenShift*. Recuperado el 14 de 12 de 2020, de https://github.com/openshift/okd/blob/master/README.md

Wikipedia. (2020). *Kernel-based Virtual Machine*. Recuperado el 15 de 12 de 2020, de Kernel-based Virtual Machine: https://es.wikipedia.org/wiki/Kernel-based_Virtual_Machine