

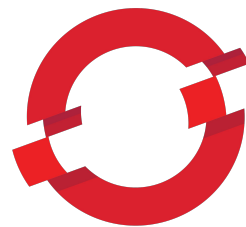
Openshift: PAAS

Autor: Antonio Guzmán Martín

Índice

1. Qué es Openshift
2. Qué es una Plataforma como servicio (PAAS)
3. Openshift y la competencia
4. Openshift Origin
5. La arquitectura de Openshift I
6. La arquitectura de Openshift II
7. Crear una aplicación en Openshift Online. Comandos CLI
8. Estructura del proyecto NodeJS

Qué es Openshift



OPENSIFT

- Es un producto de computación en la nube de plataforma como servicio de Red Hat.
- Los desarrolladores pueden usar Git para desplegar sus aplicaciones Web en los diferentes lenguajes de la plataforma.
- OpenShift se encarga de mantener los servicios subyacentes a la aplicación y la escalabilidad de la aplicación como se necesite.

Qué es una Plataforma como servicio (PaaS)

- Es la **abstracción de un ambiente de desarrollo** y el empaquetamiento de una serie de módulos o complementos que proporcionan, normalmente, una funcionalidad horizontal
- Por ejemplo, podría consistir en un entorno conteniendo una **pila básica de sistemas, componentes o APIs preconfiguradas** y listas para integrarse sobre una tecnología concreta de desarrollo (por ejemplo, un sistema Linux, un servidor web, y un ambiente de programación como Perl o Ruby).
- Las ofertas de PaaS pueden dar servicio a **todas las fases del ciclo de desarrollo y pruebas** del software
















Openshift y la competencia

Google cloud platform: Podemos destacar que está basada principalmente en estándares abiertos y que usa los mismos datacenters de Google

Amazon Web Services: Es la plataforma más extendida y la primera en entrar al mercado. Está muy por encima de la competencia en cuanto a funcionalidades, número de datacenters y número de clientes. Tienen una pegas se basan en tecnologías propias

Windows Azure: Tiene una oferta muy completa no solo de tecnologías Microsoft sino también de tecnologías open source, por las que sorprendentemente los de Redmond están haciendo una gran apuesta en los últimos años.



		Nacimiento Servicio	2006	2010	2011
		Coste	Pago por hora o fracción. Descuentos en contrataciones de 1 o 3 años. Precio servidor pequeño: 39€ Precio servidor mediano: 134€	Pago por minutos. No tiene opciones de ahorro. Precio servidor pequeño: 43€ Precio servidor mediano: 180€	Pago por minutos. Descuentos proporcionales a las horas de consumo Precio servidor pequeño: 35€ Precio servidor mediano: 143€
		Backups	Realiza 3 copias en misma zona geográfica. Posibilidad de replicar copias a otras zonas.	Realiza 3 copias en misma zona geográfica. Posibilidad de replicar copias a otras zonas.	Por defecto realiza las copias en todas las plataformas alrededor del mundo
		Disponibilidad mundial	11 centros de datos. 37 puntos de distribución contenido.	20 centros de datos. 32 puntos de distribución contenido	4 centros de datos. 160 puntos de distribución contenido
		Marketplace	2.400 aplicaciones	707 aplicaciones	160 aplicaciones
		Soporte	Soporte gratuito usando base de conocimientos. Contacto por email en horas de oficina: 49€/mes Contacto en 24x7 con 1 hora de tiempo de respuesta. 10% de la facturación (mín. 100€/mes)	Soporte gratuito usando base de conocimientos. Contacto por web en 24x7. 8 horas de tiempo de respuesta. 24,46€/mes Contacto telefónico 24x7. 2 horas de tiempo de respuesta. 256€/mes	Soporte gratuito usando base de conocimientos. 4 horas laborables de tiempo de respuesta. 150€/mes 1 hora de tiempo de respuesta. 9% de la facturación (mín. 400€/mes)
		Tipos Servidores	53	25	18
		Tipos Discos	Clásicos SSD Se pueden personalizar	Clásicos SSD No se pueden personalizar	Clásicos SSD Se pueden personalizar
		Otros Servicios en la nube	Almacenamiento. Bases de Datos. DNS. VDI.	Almacenamiento. Bases de Datos. Suite Ofimática. Correo electrónico.	Almacenamiento. Bases de Datos. Suite Ofimática. Correo electrónico. Registro dominios y DNS.
		Seguridad	20 certificaciones	25 certificaciones	6 certificaciones

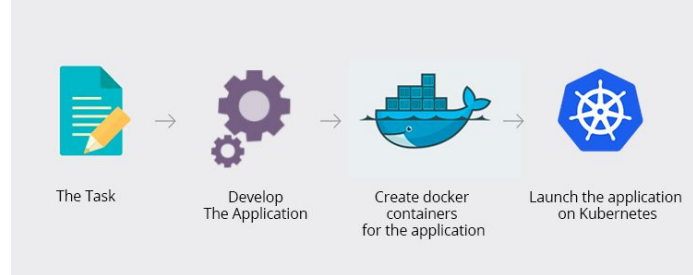
Openshift Origin

Origin es la versión de código abierto de OpenShift. Es ahora mismo una de las opciones más interesantes para **Cloud privadas**, ya que está basada en estándares abiertos, integrada con las tecnologías open source de referencia y tiene un market en constante crecimiento.

Los productos **OpenShift Online**, OpenShift Dedicated y OpenShift Enterprise son implementaciones de Origin.

Origin **utiliza Docker** para la gestión de contenedores **y Kubernetes** para la gestión de grupos de contenedores.

La arquitectura de Openshift I



Kubernetes plataforma comenzada automatiza el desarrollo, escalado y gestión de contenedores. Está construido para trabajar junto a docker. Con Kubernetes podemos ver el conjunto de contenedores como un inmenso clúster de aplicaciones donde desacoplar nuestras aplicaciones.

Docker: basado en crear contenedores portables para ser ejecutados en cualquier máquina con Docker instalado independientemente del SO que tenga por debajo. Un contenedor contiene todo aquello que necesite mi aplicación para ser ejecutada como (java,Maven,tomcat...) y la propia aplicación. La idea es que el programador pueda centrarse en desarrollar su proyecto sin preocuparse si el código funcionará en la maquina que ejecute.

91e54dfb1179	0 B
d74508fb6632	1.895 KB
c22013c84729	194.5 KB
d3a1f33e8a5a	188.1 MB
ubuntu:15.04	

Image

Imagen docker: Para almacenar drivers eficazmente, se debe entender como Docker almacena y compila imágenes. Una imagen Docker contiene una lista de capas que representan diferencias de sistemas de archivos

La arquitectura de Openshift II

Pod: Es un grupo de grupo de uno o más contenedores, el almacenamiento compartido para almacenar dichos contenedores y como ejecutarlos. Siempre se ejecutan en un contexto compartido. En un mundo pre-contenedor, sería ejecutado físicamente como una máquina virtual.

Rutas: una ruta expone un servicio en un hostname como : example.com, así los clientes pueden llegar a ellos a través de su nombre. Openshift usa el balanceo de carga del router. Un router detecta los cambios en las direcciones IP de sus servicios y adapta la configuración de forma acorde.

Desplegando la misma imagen en múltiples contenedores sobre múltiples hosts y cargando y balanceándolos entre ellos Openshift Online puede proveer de redundancia y escalado horizontal para un servicio empacado en una imagen

Crear una aplicación en Openshift Online. Comandos CLI

1. Debemos acceder a About y seleccionar Command Line tools
2. Descargar oc.tar.gz, desempaquetarlo y añadirlo al \$PATH

```
echo $PATH
```

3. Después iniciar sesión y suministrar nombre de usuario y contraseña

```
oc login
```

4. Crear nuevo proyecto nodejs.

```
oc new-project <Project-name> \  
--display-name="nodejs" --description="Sample Node.js app"
```

5. Nos situamos en el proyecto que acabamos de crear

```
oc project <Project-name>
```

6. Creamos una nueva app, escogemos un ejemplo del repositorio de openshift

```
oc new-app https://github.com/openshift/nodejs-ex -l name=myapp
```

La herramienta se encarga de inspeccionar el código, localizar una imagen apropiada en DockerHub, crear una ImageStream (contiene una o más imágenes docker) para esa imagen y crear una configuración de ejecución, de desarrollo y una definición de servicio.

Estructura del proyecto NodeJS

Estructura del proyecto nodejs, también se crean archivos en el directorio openshift > template.

```
nodejs-ex
├── openshift
│   └── templates
│       ├── nodejs.json
│       └── nodejs-mongodb.json
└── nodejs-mongodb-persistent.json
    ├── package.json
    ├── README.md
    ├── server.js
    ├── tests
    │   └── app_test.js
    ├── views
    └── index.html
```

- **Package.json:** fichero en formato json con los metadatos que identifican nuestro proyecto. El nombre y versión son obligatorios.
- **Server.js:** Si hay un archivo server.js npm por defecto lo iniciará. En este archivo se usa el framework Express que nos ayuda a organizar la aplicación web según la arquitectura MVC en el lado del servidor.
- **Directorio de test:** contendrá ficheros para realizar test de la aplicación
- **Directorio Views:** El directorio donde se encuentran los archivos de plantilla. En nuestro caso archivo html
- **Directorio Openshift:** Es una carpeta con templates que contiene nuestro proyecto que podemos para una futura creación de otro proyecto a partir del actual

Haciendo cambios en el proyecto

Exponemos nuestro servicio mongo con la orden.

```
oc expose svc/<nombre_servicio>
```

Ahora podremos acceder al proyecto a través de una url autogenerada.

```
oc describe buildConfig <nombre_proyecto>
```

Copiar webhook a github ,luego realizar cambios en nuestro proyecto y por último hacer commit y push de ellos. Por ejemplo modificando index.html

```
git add views/index.html
```

```
git push origin master
```

```
git commit -am "Ultimo cambio 5-2017"
```

```
git push origin master
```

Debemos hacer redeliver hasta que se produzca el cambio. Después accedemos a la url de nuestro proyecto

Crear una aplicación en Openshift Online. Comandos CLI

Comprobamos que todo se ha creado correctamente con

```
oc status
```

7. Creación base de datos MongoDB ejecutando un contenedor con un volumen efímero

```
oc new-app \  
  -e MONGODB_USER=admin \  
  -e MONGODB_PASSWORD=save \  
  -e MONGODB_DATABASE=save \  
  -e MONGODB_ADMIN_PASSWORD=supersave \  
  mongodb:2.6
```

8. Podremos conectarnos a nuestra base de datos Mongo con el siguiente comando:

```
oc rsh <nombre_bd>
```



¡Muchas gracias!

¿Alguna pregunta?