Deploy

TB022 - Esteban - Riesgo - Kristal

Qué es

Es la acción de "publicar" nuestro servidor para que sea accesible por la Internet.

En el fondo el proceso es muy sencillo: el código fuente se moverá a un host (una computadora) en algún lado, con instrucciones de cómo correrlo. Estas computadores donde se deploya nuestro servidor tienen las configuraciones necesarias para poder ser accedidas por la Internet por cualquier persona.

Técnicamente, cualquier computadora puede ser un host al cual podemos deployar los cambios.
Hay proveedores, llamados web hosting services o *cloud providers*, que proveen el hardware y muchas facilidades para: mantener estos dispositivos corriendo 24/7, controler el acceso y seguridad, las versiones deployadas y su historial, manejo de tráfico, el nombre de dominio y certificados, y muchas cosas más.

Por qué es difícil mantener un servidor

El proceso de desarrollo de software es iterativo. El software desarrollado nunca es fijo, va cambiando con el tiempo (fíjense por ejemplo el TP que hicimos en el cuatrimestre).

Imagínense hacer todos los cambios que fueron haciendo ***mientras*** el servidor está corriendo y hay gente usando la aplicación. Es super importante no tirar abajo el servidor o hacer cambios que hagan que cierta funcionalidad deje de andar bien.

Cosas a tener en cuenta al hacer un deploy:

- los cambios introducidos deben funcionar correctamente
- el resto del código que no debe ser afectado por los cambios debe seguir funcionando igual que antes
- nuevas migraciones de la DB deben poder correrse correctamente sobre el estado actual de la aplicación. El rollback de las mismas debe funcionar correctamente también
-

Cloud Providers

Cuál elegir?

Hoy en día hay decenas de proveedores. Los más grandes y conocidos son:

- Amazon Web Services (AWS) link
- Google Cloud Platform (GCP) link
- Microsoft Azure link

Todos ellos proveen desde soluciones sencillas a ultra complejas y son usados por los sitios más grandes del mundo.

Hay muchos otros proveedores orientados a aplicaciones más pequeñas o que tienen facilidades específicas para cierto tipo de aplicaciones:

- platform.sh (free trial)
- render (free tier)
- porter.run
- heroku
- vercel (free tier)

Nosotros vamos a usar **render** porque tiene un free tier

Render

Vamos a deployar nuestro FastApi BE

Una vez que nos creamos una cuenta en https://dashboard.render.com/ vamos a linkear nuestra cuenta de GH
Le damos permisos a los repos que queramos deployar.
Y seguimos los pasos seleccionando FastAPI

El free tier nos permite configurar todo lo que necesitamos ... salvo ...

No podemos usar Render de esta manera porque no vamos a poder correr nuestras migraciones (al menos con el free tier).

Pero no es un problema, porque tenemos nuestro Dockerfile y podemos deployar un container de Docker genérico

Y el FE?

También se puede deployar, pero ya que estamos vamos a usar Vercel que está orientado a hacer deploys de proyectos basados en js

Vercel

Vamos a Deployar nuestra app de Sveltekit con Vercel

Una vez que nos creamos una cuenta en https://vercel.com/ vamos a linkear nuestra cuenta de GH
Le damos permisos a los repos que queramos deployar. La única limitación es que tiene que no ser de una org y tiene que estar el contenido en la branch main (después se puede cambiar).

Ya al seleccionar el repositorio debería automáticamente seleccionarse que es para Sveltekit. Con las opciones default ya es suficiente para empezar.
Lo único, hay que setear la ENV VAR de la URL del backend a la URL del servicio deployado en Render.
Cada vez que se mergee a main, se hará un deploy.

Ambientes y Ciclos de Desarrollo

Cómo se organiza el desarrollo productivo de un sistema

Es super importante tener procesos claros para evitar introducir problemas y errores inesperados.

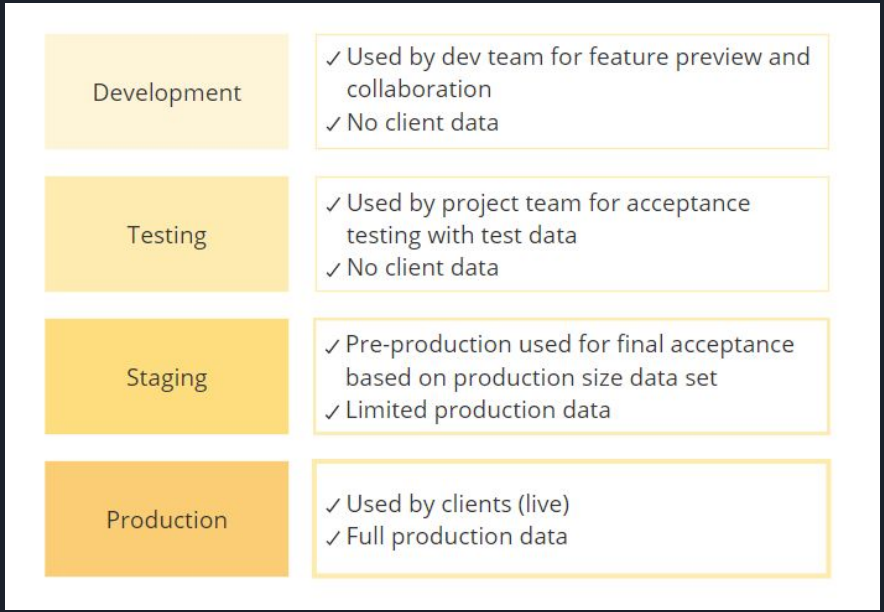
Para mitigar posibles problemas se suelen tener más de un ambiente. Hay ambientes productivos y ambientes de desarrollo.

Los ambientes productivos implican escenarios con usuarios reales. Las aplicaciones que usamos todos los días son ambientes productivos. A veces hay más de un ambiente productivo, si una aplicación tiene versiones Beta o Alfa con updates que todavía no están en la versión general.

Los ambientes de desarrollo son escenario ficticios, generalmente creados para los desarrolladores del software. La función de estos ambientes es de probar los distintos casos de uso de la aplicación y encontrar bugs (y solucionarlos) antes de llegar a un ambiente productivo y llegar a usuarios reales. Generalmente hay más de un ambiente de desarrollo, llegando a haber una gran cantidad en aplicaciones muy grandes y masivas.

Ambientes de Desarrollo

Esta configuración podría ser la algun producto de software cualquiera



Cada empresa, y a veces distintos productos de la misma empresa, tienen distintos flujos de trabajo y distintos ambientes de desarrollo con distintos usos.

Fin