

TWELVE-FACTOR APP

CLASIFICACIÓN DE SERVICIOS

- IaaS (Infraestructura as a Service): Amazon EC2, Microsoft Azure
- PaaS (Platform as a Service): Heroku, Google App Engine
- SaaS (Software as a Service): Google Drive, Facebook, etc

NUEVOS DESAFÍOS

OBJETIVOS DEL MANIFIESTO

- Usar configuraciones automatizables y explícitas, nada manual.
- Dependencias claras con el entorno en el que se ejecutan para máxima compatibilidad.
- Tener en cuenta que la aplicación será desplegada en la nube.
- Despliegue continuo, por tanto requiere coordinación de entornos (test-uat-pre-pro...).
- Crear aplicaciones con la escalabilidad en la mente.

CÓDIGO BASE

1

El código debe estar en repositorios y es desde ahí donde se harán los despliegues a los entornos. Puede haber varios repositorios si la aplicación es compleja, pero no se puede repetir código entre repositorios

DEPENDENCIAS

2

Todas las dependencias de la aplicación deben ser explícitas y existir un sistema automático que las instale si no existen

CONFIGURACIONES

3

El código no puede tener configuraciones, sino que deben ser separadas en otros recursos dependientes del entorno. Así, el código debe hacer referencia a las variables de configuración pero no a los valores. Esto permite instalar el mismo código en diferentes entornos

BACKING SERVICES

4

Son recursos conectables a la aplicación, independientemente de si son locales o de terceros. Si la aplicación necesita una base de datos, será tratada como un recurso independiente, con su URL y su instalación, que deben ser proporcionados a la aplicación

CONSTRUIR, DISTRIBUIR Y EJECUTAR

5

Son las 3 etapas que se deben pasar para transformar el código fuente en un producto utilizable. Deben ser tratadas de forma independiente y con un control de las versiones que se van distribuyendo acorde a un sistema preestablecido.

PROCESOS SIN ESTADO (STATELESS)

6

La aplicación se ejecuta en el destino como un proceso (o varios) en la máquina anfitriona pero la información que existe en su espacio de memoria sólo tiene lo necesario para hacer las transacciones que lleva a cabo en ese momento, no para atender a transacciones futuras: puede reiniciarse la máquina, aparecer otro proceso... La información que sea necesario almacenar para el futuro se hará en un Backing Service

ASIGNACIÓN DE PUERTOS

7

El código no puede considerar a priori la asignación de los puertos. Simplemente sabe que es un proceso que escucha en un puerto arbitrario. Será la configuración la que establezca los puertos de cada componente

CONCURRENCIA

8

Las aplicaciones deben estar desarrolladas teniendo en mente la concurrencia, bien de hilos o bien de procesos. La razón principal es la naturaleza distribuida y la capacidad para escalar levantando nuevas máquinas

DISPONIBILIDAD

9

Las aplicaciones deben ser capaces de iniciar sus procesos rápidamente y de responder de forma controlada a la finalización, aunque esta sea inesperada. Cuando hablamos de escalado, nos referimos al inicio o parada en caliente de nuevas máquinas o al traslado a diferentes máquinas de los procesos.

PARIDAD EN EL DESARROLLO Y PRODUCCIÓN

10

Se refiere a tener los diferentes entornos (des-pre como mínimo) lo más similares posibles, tanto de versiones de código desplegado, como de entornos y servicios. Esto se hace con despliegues continuos, lo que implica automatizar los despliegues lo máximo posible, que el personal de desarrollo sea mucho más consciente del entorno de producción y que incluso se compartan modos de trabajo entre entornos aunque no sea lo más evidente a primera vista: puppet, chef, ansible

HISTORIALES

11

0 más conocidos como “Logs”. Las aplicaciones no se preocupan del direccionamiento y almacenamiento de sus datos de logs. Para eso existen sistemas específicos en cada entorno

ADMINISTRADOR DE PROCESOS

12

Además de los procesos propios de la ejecución de la aplicación, existirán procesos de tareas de administración y mantenimiento que suelen ejecutarse una única vez (ej: migración de bases de datos, ejecuciones de consola) . Estos procesos, aunque puedan parecer secundarios, se ejecutarán con igual importancia que el resto de los procesos de la aplicación

CONCLUSIONES

LAS NUEVAS OPCIONES PARA LA DISTRIBUCIÓN DE LAS APLICACIONES EN INTERNET AFECTAN A LA FORMA DE DESARROLLARLAS Y EN GENERAL A SU CICLO DE VIDA: LA NECESIDAD DE ADAPTARSE A LA DEMANDA, TANTO EN NÚMERO COMO NUEVAS CAPACIDADES, Y HACERLO CON UNA DEMORA CADA VEZ MENOR, NOS FUERZA A ADOPTAR NUEVOS ENFOQUES.