

En esta sección vamos a ver cómo utilizar Docker para mejorar nuestra metodología de entrega continua. Vamos a asumir que el entorno de desarrollo y el de staging han sido fusionados (al entorno resultante le seguiremos llamando staging).

Docker no sólo simplifica enormemente la metodología de entrega continua, si no que la hace mucho más fiable porque el salto del entorno de staging al de producción es menor, y porque las imágenes de Docker, portables e inmutables, han sido validadas por las pruebas unitarias, de integración y de aceptación.

Lo primero que tenemos que hacer es crear un servidor para realizar las labores de integración continua, y que vamos a llamar *builder*. También crearemos un servidor, llamado *staging*, para el entorno de staging. Por último, crearemos varios servidores para nuestro entorno de producción.

A continuación configuramos nuestro repositorio de código (por ejemplo, un repositorio git en Github). El repositorio constará de dos ramas, *master* y *staging*, donde no estará permitido comitear directamente. Cada desarrollador hará un *fork* de este repositorio para hacer desarrollo. Para comitear código al repositorio principal, los desarrolladores crearán *pull requests* que pueden ser revisadas por el resto de desarrolladores.

Nuestros procesos de integración continua validarán el correcto funcionamiento de cada pull request. Cuando una pull request es revisada y *mergeada*, otro proceso de integración continua volverá a correr las pruebas en la rama de staging, y si estas pasan, creará una imagen de Docker con el tag *staging* y el entorno de staging se actualiza automáticamente.

Cuando el equipo de operaciones decida hacer una nueva release se mergea la rama de staging en la rama master, y otro proceso de integración construirá la imagen de Docker con el tag *latest* desde la rama master. Además, a este commit de *master* le añadiremos un tag en Github, y crearemos este mismo tag en la imagen de Docker desde la rama master. También está automatizado el proceso de apagar los contenedores que están corriendo en producción para recrearlos con la nueva imagen. Ante cualquier regresión en producción, podemos fácilmente actualizar los contenedores corriendo en producción para que usen un tag de la imagen de Docker anterior. Por último, todos estos procesos automatizados se notifican al equipo de desarrollo a través de email, o por canales como Slack o Hipchat.