

# CONTINUOUS DEPLOYMENT

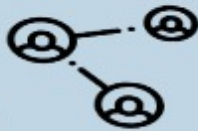
## EN 5 SENCILLOS PASOS.

BY ERIC RIES

INTRODUCCIÓN: TÉCNICA EN DONDE TODO EL CÓDIGO QUE SE ESCRIBE PARA UNA APLICACIÓN ES DESPLEGADO A PRODUCCIÓN INMEDIATAMENTE

### CONTINUOUS INTEGRATION SERVER

- LUGAR CENTRALIZADO DONDE TODAS LAS PRUEBAS AUTOMATIZADAS PUEDAN SER MONITOREADAS DESPUÉS DE CADA COMMIT
- ENCONTRAR PROGRAMA QUE PERMITA CORRER LAS PRUEBAS PARA TODO EL SOFTWARE ESCRITO



### SOURCE CONTROL COMMIT CHECK

- SCRIPT QUE CONTROLE ORIGEN DE CADA COMMIT
- PODER RECHAZAR LOS CAMBIOS DE UN BUILD SI ALGUNO DE LOS TESTS DA UN RESULTADO NEGATIVO
- RETROALIMENTACION
- DISMINUIR DELAYED WASTE



### SIMPLE DEPLOYMENT SCRIPT

- AUTOMATIZAR EL PROCESO A MEDIDA QUE VAMOS GANANDO EXPERIENCIA
- REALIZAR TODOS LOS DESPLIEGUES DE LA MISMA MANERA Y TENER UN PROCESO PÚBLICO



### REAL-TIME ALERTING

- PLATAFORMA DE MONITOREO QUE PERMITA SABER CUÁNDO LAS COSAS EMPEZARON A FUNCIONAR MAL
- ESTAR ATENTOS A LOS PROBLEMAS



### ROOT CAUSE ANALYSIS

- OBTENER LA CAUSA RAÍZ DE UN PROBLEMA INESPERADO DEL SISTEMA PREGUNTANDO PORQUE RECURSIVAMENTE HASTA DESCUBRIR LA FUENTE
- INVERSIÓN PROPORCIONAL AL PROBLEMA PARA PREVENIR UNO PARECIDO EN EL FUTURO



CONCLUSIÓN: SEGUIR ESTOS 5 PASOS NO LE VA A DAR A UN EQUIPO LA CAPACIDAD DE REALIZAR CONTINUOUS DEPLOYMENT DEL DÍA A LA NOCHE, HACE FALTA INVERTIR TIEMPO, PERO CON PACIENCIA Y TRABAJO DURO CUALQUIERA PUEDE USAR ESTA TÉCNICA PARA DEJAR DE GENERAR DESPERDICIO EN EL PROCESO DE DESARROLLO