

Manual para ejecutar el pipeline de Jenkins y cambiar la versión de la aplicación java

Contents

Chapter 1. Continuous Deployment y Continuous Delivery.....	3
Continuous Deployment y Continuous Delivery.....	3
Chapter 2. Empaquetar una Aplicación Java con Maven.....	5
Chapter 3. Análisis de Calidad de Código Fuente.....	6
Realizar Análisis de Calidad de Código con SonarQube.....	6
Chapter 4. Transferir paquete a Sonatype Nexus con Maven (línea de comando).....	7
Chapter 5. Cambio de Versión de Aplicación Java.....	8

Chapter 1. Continuous Deployment y Continuous Delivery

Esta sección describe cómo utilizar la disciplina de Continuous Deployment y Continuous Delivery

Este proceso define cómo entregar continuamente activos de software.

Continuous Deployment y Continuous Delivery

Explicación sobre Continuous Deployment y su relación con Continuous Delivery.

Continuous Deployment

Continuous Deployment es una práctica en la que cada cambio de código que pasa los criterios de pruebas automatizadas es automáticamente desplegado en producción. Esto significa que el código nuevo es implementado sin intervención humana adicional después de las pruebas.

El objetivo principal del Continuous Deployment es proporcionar una forma eficiente y rápida de llevar las nuevas características y correcciones de errores a los usuarios finales.

Continuous Delivery

Continuous Delivery es una práctica relacionada que se centra en asegurar que el software esté siempre en un estado desplegable, es decir, que cualquier versión del software pueda ser implementada en cualquier momento. Esto implica la automatización de procesos de construcción, pruebas y despliegue para garantizar que el software sea siempre estable y listo para su lanzamiento.

Continuous Delivery establece la base para Continuous Deployment, ya que garantiza que el software esté siempre en un estado que pueda ser desplegado, lo que facilita la implementación automática del software en producción.

Relación entre Continuous Deployment y Continuous Delivery

Continuous Deployment y Continuous Delivery están estrechamente relacionados y a menudo se utilizan juntos en los entornos de desarrollo de software ágiles y DevOps. Continuous Delivery proporciona los procesos y herramientas necesarios para garantizar que el software sea desplegable en cualquier momento, mientras que Continuous Deployment lleva esto un paso más allá al automatizar el despliegue de cada cambio de código que pase las pruebas.

Juntos, Continuous Deployment y Continuous Delivery permiten a los equipos de desarrollo y operaciones entregar rápidamente nuevas características y correcciones de errores a los usuarios finales de manera segura y confiable.

La adopción de Continuous Deployment y Continuous Delivery es fundamental en el desarrollo de software moderno por varias razones clave:

- Entrega Rápida de Valor al Cliente
 - Estas prácticas permiten entregar nuevas características y correcciones de errores a los usuarios finales de manera rápida y frecuente. Al automatizar los procesos de construcción, pruebas y despliegue, los equipos pueden liberar cambios de manera eficiente y segura, mejorando la experiencia del usuario y manteniendo la competitividad en el mercado
- Reducción del Riesgo y la Incertidumbre
 - Continuous Delivery garantiza que el software esté siempre en un estado desplegable y probado, lo que reduce el riesgo asociado con la implementación de cambios en producción. Al realizar despliegues pequeños y frecuentes, cualquier problema puede detectarse y corregirse rápidamente, lo que disminuye la incertidumbre y aumenta la confianza en el software
- Iteración y Retroalimentación Continuas
 - Estas prácticas fomentan un ciclo de desarrollo iterativo, donde los equipos pueden experimentar, aprender y mejorar continuamente. Al recibir retroalimentación rápida de los usuarios finales a través de despliegues frecuentes, los equipos pueden adaptar sus productos de manera ágil y responder rápidamente a las necesidades cambiantes del mercado
- Mejora de la Colaboración y la Cultura de Equipo
 - Continuous Deployment y Continuous Delivery promueven una cultura de colaboración y responsabilidad compartida entre los equipos de desarrollo y operaciones. Al automatizar los procesos y trabajar en estrecha colaboración para lograr despliegues confiables y frecuentes, se fortalecen los lazos entre los equipos y se crea un ambiente de trabajo más eficiente y orientado a resultados

Chapter 2. Empaquetar una Aplicación Java con Maven

Tarea que describe cómo empaquetar una aplicación Java utilizando Maven.

Esta tarea assume que tienes Maven instalado en tu sistema y un proyecto Java configurado con un archivo pom.xml.

1. Abre una terminal o línea de comandos.
2. Navega hasta el directorio raíz de tu proyecto Java.
3. Ejecuta el comando `mvn package` para empaquetar la aplicación.
4. Espera a que Maven compile el código y genere el archivo JAR o WAR.
5. Verifica la salida de la consola para confirmar que el proceso de empaquetado se completó correctamente.

Chapter 3. Análisis de Calidad de Código Fuente

Concepto sobre la práctica de realizar análisis de calidad de código fuente en proyectos de software.

El análisis de calidad de código fuente es una actividad importante en el desarrollo de software para evaluar la salud general del código y detectar posibles problemas y áreas de mejora.

Esta práctica implica el uso de herramientas automatizadas que examinan el código fuente en busca de violaciones de estándares de codificación, bugs, vulnerabilidades de seguridad, y otros problemas que puedan afectar la calidad y el rendimiento del software.

El análisis de calidad de código fuente ayuda a los equipos de desarrollo a identificar y corregir problemas tempranamente, mejorar la mantenibilidad del código y reducir el riesgo de errores en el software.

Realizar Análisis de Calidad de Código con SonarQube

Tarea que describe cómo realizar análisis de calidad de código fuente utilizando SonarQube.

Esta tarea asume que tienes configurado un proyecto en SonarQube y acceso a una instancia de SonarQube en tu entorno de desarrollo.

1. Abre un navegador web y accede a la interfaz de SonarQube.
2. Inicia sesión con tus credenciales de usuario.
3. Selecciona el proyecto que deseas analizar.
4. Configura las opciones de análisis según tus necesidades, como reglas de calidad, perfiles de análisis, y configuraciones adicionales.
5. Ejecuta el análisis de calidad de código desde la interfaz de SonarQube.
6. Espera a que se complete el análisis y revisa los resultados obtenidos.
7. Analiza los informes generados por SonarQube y toma las medidas necesarias para abordar los problemas identificados.

Chapter 4. Transferir paquete a Sonatype Nexus con Maven (línea de comando)

Tarea que describe cómo transferir el paquete generado al repositorio de Sonatype Nexus usando Maven a través de la línea de comando.

Esta tarea supone que tienes Maven y acceso al repositorio de Sonatype Nexus configurado correctamente.

1. Abre una terminal o línea de comandos.
2. Navega hasta el directorio raíz de tu proyecto Java.
3. Ejecuta el comando "mvn deploy" para transferir el paquete generado al repositorio Nexus.
4. Espera a que Maven complete la transferencia y verifícala en la consola para asegurarte de que se haya realizado correctamente.

Chapter 5. Cambio de Versión de Aplicación Java

Tarea que describe cómo realizar el cambio de versión de una aplicación Java.

Esta tarea asume que tienes acceso a los servidores de aplicación y al balanceador de carga configurados correctamente.

1. Configura el balanceador de carga para incluir el nuevo servicio en la configuración.
2. Inicia la aplicación nueva en el servidor de aplicación.
3. Redirige el tráfico a través del balanceador de carga hacia la nueva aplicación.
4. Verifica que la nueva aplicación esté funcionando correctamente y respondiendo a las solicitudes.
5. Desactiva el balanceador de carga para redirigir el tráfico hacia la aplicación antigua.
6. Verifica que no haya más tráfico siendo dirigido hacia la aplicación antigua.
7. Detiene la aplicación antigua en el servidor de aplicación.
8. Elimina la aplicación antigua del servidor para liberar recursos.
9. Verifica que el cambio de versión se haya completado exitosamente y que la nueva aplicación esté funcionando según lo esperado.