



Gestión Tecnológica

Integrantes:

Diego Alejandro Parra Daza
Diego Alejandro Rodriguez Ayala
Andres Felipe Gomez Salinas

DevOps Lifecycle

CONCEPTS

- **Integración continua (CI)**

La integración continua es una práctica de desarrollo de software mediante la cual los desarrolladores combinan los cambios en el código en un repositorio central de forma periódica, tras lo cual se ejecutan versiones y pruebas automáticas. Los objetivos clave de la integración continua consisten en encontrar y arreglar errores con mayor rapidez, mejorar la calidad del software y reducir el tiempo que se tarda en validar y publicar nuevas actualizaciones de software.

- **Entrega continua (CD)**

La entrega continua es una práctica de desarrollo de software mediante la cual se crean, prueban y preparan automáticamente los cambios en el código y se entregan para la fase de producción. Amplía la integración continua al implementar todos los cambios en el código en un entorno de pruebas y/o de producción después de la fase de creación. Cuando se la entrega continua se implementa de manera adecuada, los desarrolladores dispondrán siempre de un artefacto listo para su implementación que se ha sometido a un proceso de pruebas estandarizado.

MANAGE - PLAN:

Esta fase se compone de dos cosas: "definir" y "plan". Esta actividad se refiere al valor del negocio y los requisitos de la aplicación. Específicamente las actividades del "plan" incluyen:

- Métricas de producción, objetos y feedback.
- Requerimientos
- Métricas de negocios
- Actualización de métricas de lanzamiento
- Plan de lanzamiento, calendario y caso de negocio.
- Política de seguridad y exigencia.

Documentación para:

- Características estadísticas y analíticas.
- Características de planificación y gestión de proyectos.

Tuleap



Descripción:

Tuleap es una plataforma web Open source y está disponible en dos versiones una empresarial de pago y otra gratuita para ser ejecutada en infraestructuras propias. En su página oficial se describe del siguiente modo:

Tuleap es una suite gratuita y de código abierto para mejorar la administración de proyectos de software y conectarse con los miembros del equipo. Con una única solución basada en la web, el CTO, los gerentes de proyecto, los desarrolladores, los gerentes de calidad y otros usuarios pueden construir, implementar proyectos inteligentes y de calidad con facilidad.

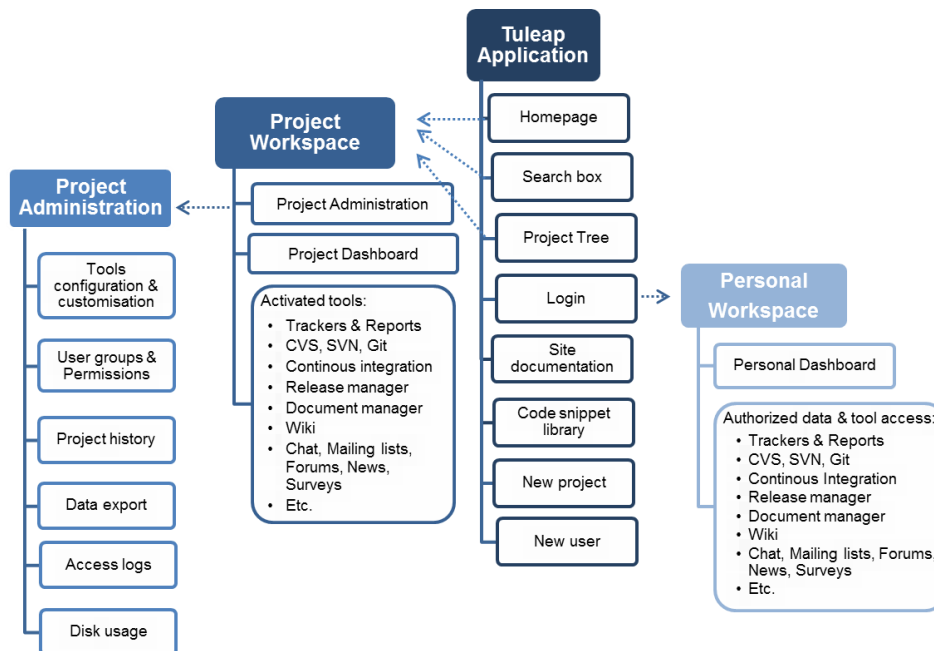
Tuleap proporciona herramientas para gestionar proyectos, tareas, cambios, defectos, documentos, así como control de versiones, integración continua y colaboración social.

Con Tuleap, podrás:

- planificar y monitorear proyectos,

- administrar el ciclo de vida del desarrollo de software: código fuente, compilaciones...,
- Requisitos de seguimiento, tareas, incidencias, etc.
- producir documentos y comunicados,
- Favorecer la colaboración entre los miembros del proyecto.

A continuación se muestra un mapa con los elementos que componen Tuleap:



CREATE:

Esta fase se compone de la creación, codificación y configuración del proceso de desarrollo de software. Las actividades específicas son:

- Diseño del software y configuración.
- Codificación que incluye la calidad y el rendimiento del código.
- Software construir y construir rendimiento
- Candidato de liberación

Documentación para:

- Código fuente y funciones de creación y gestión de datos.

GitLab



Descripción:

GitLab ayuda a los equipos a diseñar, desarrollar y administrar de manera segura los datos de código y proyecto desde un único sistema de control de versiones distribuido para permitir una rápida iteración y entrega de valor comercial. Los repositorios de GitLab proporcionan una única fuente de verdad escalable para colaborar en proyectos y código que permite a los equipos ser productivos sin interrumpir sus flujos de trabajo.

Dispone de las siguientes herramientas:

- **Gestión de código fuente:** La administración del código fuente permite la coordinación, el intercambio y la colaboración en todo el equipo de desarrollo de software. Rastree y fusione sucursales, audite los cambios y habilite el trabajo simultáneo, para acelerar la entrega del software.
- **Revisión del código:** Revise el código, discuta los cambios, comparta el conocimiento e identifique defectos en el código entre los equipos distribuidos a través de la revisión y los comentarios asíncronos. Automatizar, rastrear y reportar revisiones de código.
- **Wiki:** Comparte la documentación y la información de la organización con un wiki incorporado.
- **IDE web:** Un entorno de desarrollo integrado (IDE, por sus siglas en inglés) con todas las funciones integrado en GitLab para que pueda comenzar a contribuir el primer día sin necesidad de pasar todos los días instalando todos los paquetes correctos en su entorno de desarrollo local.

VERIFY:

Esta fase está directamente asociado con garantizar la calidad de la versión del software; las actividades diseñadas para garantizar que la calidad del código se mantiene y la más alta calidad se implementa en la producción. Las principales actividades en esto son:

- Test de aceptación
- Pruebas de regresión
- Análisis de seguridad y vulnerabilidad.

- Actuación
- Pruebas de configuración

Documentación para:

- Pruebas, calidad de código, y características de integración continua.

Jenkins



Jenkins

Descripción:

Jenkins es un servidor de automatización de código abierto, autónomo, que se puede utilizar para automatizar todo tipo de tareas relacionadas con la creación, prueba y entrega o implementación de software.

Jenkins se puede instalar a través de paquetes de sistemas nativos, Docker, o incluso puede ejecutarse de manera independiente en cualquier máquina con un Java Runtime Environment (JRE) instalado.

La base de Jenkins son las tareas, donde indicamos qué es lo que hay que hacer en un build (resultado de una compilación de un código integrado). Por ejemplo, podríamos programar una tarea en la que se compruebe el repositorio de control de versiones cada cierto tiempo, y cuando un desarrollador quiera subir su código al control de versiones, este se compile y se ejecuten las pruebas.

Si el resultado no es el esperado o hay algún error, Jenkins notificará al desarrollador, al equipo de QA, por email o cualquier otro medio, para que lo solucione. Si el build es correcto, podremos indicar a Jenkins que intente integrar el código y subirlo al repositorio de control de versiones.

Pero la cosa no queda ahí. Una de las cosas buenas que tiene Jenkins es que además de poder ayudarte a integrar el código periódicamente, puede actuar como herramienta que sirva de enlace en todo el proceso de desarrollo.

Desde Jenkins podrás indicar que se lancen métricas de calidad y visualizar los resultados dentro de la misma herramienta. También podrás ver el resultado de los tests, generar y visualizar la documentación del proyecto o incluso pasar una

versión estable del software al entorno de QA para ser probado, a pre-producción o producción.

PACKAGE:

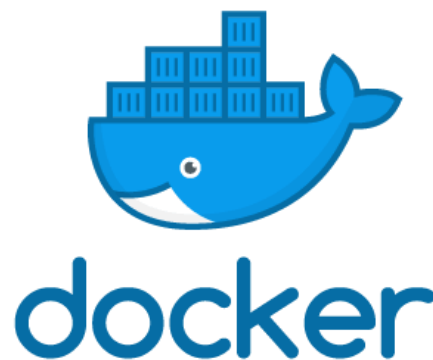
Esta fase se refiere a las actividades involucradas una vez que el lanzamiento está listo para la implementación, a menudo también se denomina puesta en escena o preproducción/"preprod". Esto a menudo incluye tareas y actividades tales como:

- Aprobación / Aprobaciones
- Configuración del paquete
- Lanzamientos disparados
- Liberación de la puesta en escena y la celebración

Documentación para:

- Registro contenedor de Docker.

Docker



Descripción:

Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. Docker utiliza características de aislamiento de recursos del kernel Linux, tales como cgroups y espacios de nombres (namespaces) para permitir que "contenedores" independientes se ejecuten

dentro de una sola instancia de Linux, evitando la sobrecarga de iniciar y mantener máquinas virtuales.

Docker Hub



Descripción:

Docker Hub es un servicio de repositorio basado en la nube en el que los usuarios y socios de Docker crean, prueban, almacenan y distribuyen imágenes de contenedor. Docker Hub es el principal repositorio público de Docker al que van todas las herramientas de docker de forma predeterminada. Ofrece repositorios públicos (gratis) y repositorios privados (por un costo mensual recurrente).

SECURE:

Esta etapa de seguridad se refiere a la construcción de seguridad en las herramientas que existen en DevOps. Esto significa automatización sobre procesos manuales. Significa el uso de herramientas de análisis estático que verifican las porciones de código que se han cambiado, en lugar de escanear todo el código base.

Documentación para:

- Características de la capacidad de seguridad.

SonarQube



Descripción:

SonarQube es una herramienta de código abierto para la inspección continua de la calidad del código utilizando el análisis estático de la composición del software para detectar errores de códigos y vulnerabilidades de seguridad en más de 20 lenguajes de programación. SonarQube ofrece informes sobre códigos duplicados, estándares de codificación, pruebas unitarias, cobertura de códigos, complejidad de códigos, comentarios, errores y vulnerabilidades de seguridad.

RELEASE:

Esta fase actividades relacionadas con la publicación incluye programación (horario), orquestación, aprovisionamiento y despliegue de software en el entorno de producción y específico. Las actividades específicas de lanzamiento incluyen:

- Coordinación de liberación
- Despliegue y promoción de aplicaciones.
- Retrocesos y recuperación
- Lanzamientos programados/cronometrados

Documentación para:

- Características de liberación y entrega de aplicaciones.

Spinnaker



Descripción:

Spinnaker es una plataforma de entrega continua de múltiples fuentes y de código abierto para liberar cambios de software con alta velocidad y confianza. Creado en Netflix, ha sido probado en combate en la producción por cientos de equipos en millones de implementaciones. Combina un sistema de gestión de tuberías potente y flexible con integraciones a los principales proveedores de la nube.

Spinnaker utiliza las funciones de administración de aplicaciones para ver y administrar los recursos de la nube.

Las organizaciones tecnológicas modernas operan colecciones de servicios, a veces denominadas "aplicaciones" o "microservicios". Una aplicación de Spinnaker modela este concepto. Las aplicaciones, los clústeres y los grupos de servidores son los conceptos clave que Spinnaker utiliza para describir sus servicios.

CONFIGURE:

Esta etapa de configuración caen bajo el lado operativo de DevOps. Una vez que se implementa el software, es posible que se requieran actividades adicionales de configuración y aprovisionamiento de infraestructura de TI. Actividades específicas que incluyen:

- Almacenamiento de infraestructura, bases de datos y aprovisionamiento y configuración de redes.
- Provisión y configuración de aplicaciones.

Documentación para:

- Herramientas de configuración de aplicaciones e infraestructura.

Open Source Puppet



Descripción:

Puppet es una herramienta de administración de la configuración y un lenguaje que permite la implementación y el mantenimiento del estado para la infraestructura a gran escala. Puppet se destaca como administrador de infraestructura heredada como servidores físicos y máquinas virtuales. Puppet fue diseñado antes de la adopción generalizada de contenedores y no implementa Kubernetes de forma nativa.

MONITOR:

Esta etapa de monitoreo es un enlace importante en una cadena de herramientas DevOps. Permite a la organización de TI identificar problemas específicos de versiones específicas y comprender el impacto en los usuarios finales. Un resumen de las actividades relacionadas con el Monitor son:

- Rendimiento de la infraestructura de TI
- Respuesta y experiencia del usuario final.
- Métricas y estadísticas de producción.

Documentación para:

- Monitoreo de aplicaciones y características de métricas.

Prometheus



Descripción:

Prometheus es un conjunto de herramientas de monitoreo y alerta de sistemas de código abierto construido originalmente en SoundCloud . Desde su inicio en 2012, muchas empresas y organizaciones han adoptado Prometheus, y el proyecto tiene una comunidad de desarrolladores y usuarios muy activa . Ahora es un proyecto de código abierto independiente y se mantiene independientemente de cualquier compañía. Para enfatizar esto, y para aclarar la estructura de gobierno del proyecto, Prometheus se unió a la Cloud Native Computing Foundation en 2016 como el segundo proyecto hospedado, después de Kubernetes .

Algunas características de Prometheus son:

- Un modelo de datos multidimensional con datos de series de tiempo identificados por nombre de métrica y pares clave / valor
- PromQL, un lenguaje de consulta flexible para aprovechar esta dimensionalidad.
- Colección de series de tiempo a través de un modelo de extracción a través de HTTP.

- Empujar series de tiempo es compatible a través de una pasarela intermedia
- Los objetivos se descubren a través del descubrimiento del servicio o la configuración estática
- Múltiples modos de gráficos y soporte de cuadros de mando.