

GitLab CI/CD, dominar la gestión de su ciclo de vida de desarrollo de software de la integración continua a la implantación continua

Curso práctico de 3 días - 21h

Ref.: GLN - Precio 2025: 1 630€ sin IVA

GitLab no es sólo un gestor de versiones, también es una solución competitiva para la ingeniería DevOps. Este curso práctico le enseñará a utilizar esta herramienta para mejorar la colaboración en proyectos y automatizar un ciclo de desarrollo completo (compilaciones, pruebas, despliegues, etc.).

OBJETIVOS PEDAGÓGICOS

Al término de la formación, el alumno podrá:

Más información sobre GitLab

Practicar la gestión de versiones con Git y trabajar con GitLab

Implantación de la integración continua (IC) y el despliegue continuo (DC) con GitLab

Comprender los componentes básicos de una fábrica de software DevOps

Alternancia de presentaciones teóricas y ejercicios prácticos.

PROGRAMA

última actualización: 04/2024

1) La oferta de GitLab

- GitLab CE (edición comunitaria). GitLab EE (edición Enterprise).
- Las diferentes formas de utilizar GitLab: gitlab.com, on premise, en la nube (AWS, GCP).
- Procedimiento de registro en gitlab.com.

Trabajo práctico : Creación de una cuenta GitLab en gitlab.com. Navegar por la interfaz.

2) Recordatorios sobre Git y cómo utilizarlo con GitLab

- Conceptos de Git: blob, árbol, commit, revisión, rama, etiqueta, etc.
- Gestión local de archivos. Visualización y modificación del historial laboral.
- Gestión de sucursales. Fusiones de sucursales y gestión de conflictos.

Trabajo práctico : Creación de un repositorio remoto y simulación del trabajo en colaboración.

3) Gestión de repositorios con GitLab

- Gestión de los colaboradores del proyecto y sus derechos.
- El sistema de salida y el vínculo con los commits.
- Introducción a la wiki y fragmentos.

Trabajo práctico : Establecimiento de derechos y creación de solicitudes de fusión.

4) GitLab CI/CD

- Presentación de GitLab CI/CD y GitLab runners.
- Introducción a Docker.
- El archivo de manifiesto gitlab-ci.yml, una introducción al lenguaje YAML.

PARTICIPANTES

Desarrolladores, jefes de proyecto, administradores de sistemas, arquitectos.

REQUISITOS PREVIOS

Conocimientos básicos de comandos Linux. Conocimientos básicos de gestión de versiones con Git.

COMPETENCIAS DEL FORMADOR

Los expertos que imparten la formación son especialistas en las materias tratadas. Han sido validados por nuestros equipos pedagógicos, tanto en el plano de los conocimientos profesionales como en el de la pedagogía, para cada curso que imparten. Cuentan al menos con entre cinco y diez años de experiencia en su área y ocupan o han ocupado puestos de responsabilidad en empresas.

MODALIDADES DE EVALUACIÓN

El formador evalúa los progresos pedagógicos del participante a lo largo de toda la formación mediante preguntas de opción múltiple, escenificaciones de situaciones, trabajos prácticos, etc. El participante también completará una prueba de posicionamiento previo y posterior para validar las competencias adquiridas.

MEDIOS PEDAGÓGICOS Y TÉCNICOS

- Los medios pedagógicos y los métodos de enseñanza utilizados son principalmente: ayudas audiovisuales, documentación y soporte de cursos, ejercicios prácticos de aplicación y ejercicios corregidos para los cursillos prácticos, estudios de casos o presentación de casos reales para los seminarios de formación.
- Al final de cada cursillo o seminario, ORSYS facilita a los participantes un cuestionario de evaluación del curso que analizarán luego nuestros equipos pedagógicos.
- Al final de la formación se entrega una hoja de presencia por cada media jornada de presencia, así como un certificado de fin de formación si el alumno ha asistido a la totalidad de la sesión.

MODALIDADES Y PLAZOS DE ACCESO

La inscripción debe estar finalizada 24 horas antes del inicio de la formación.

ACCESIBILIDAD DE LAS PERSONAS CON DISCAPACIDAD

¿Tiene alguna necesidad específica de accesibilidad? Póngase en contacto con la Sra. FOSSE, interlocutora sobre discapacidad, en la siguiente dirección psh-accueil@orsys.fr para estudiar de la mejor forma posible su solicitud y su viabilidad.

- Etiquetas esenciales para describir etapas, trabajos y procesos (cursos, imágenes, guiones, etc.).
- Supervisión de la ejecución de tuberías. Trabajos automáticos, manuales y programados.
- Artefactos y mejora del rendimiento con el almacenamiento en caché.
- La documentación oficial sobre la sintaxis del archivo de manifiesto.
- Gestión medioambiental.

Trabajo práctico : Ejemplos sencillos de creación de empleo y encadenamiento.

5) Más allá con GitLab

- Tipos de corredor (corredor compartido, corredor específico y corredor en grupo).
- Corredores compartidos disponibles en gitlab.com gestionados por el equipo de GitLab.
- Escalabilidad de los corredores con Docker Machine.
- El ejecutor Docker y las imágenes disponibles: MySQL, PostgreSQL, Redis, MongoDB, Ruby, etc.
- Pruebas de integración, pruebas funcionales y pruebas de escalabilidad.
- Despliegue en la nube.
- El ejecutor Kubernetes y la distribución de trabajos en un clúster Kubernetes.
- Integración de la herramienta de supervisión Prometheus.

Trabajo práctico : Configuración de un corredor específico. Ejecución de secuencias de trabajos en diferentes ejecutores.

6) Funciones adicionales de GitLab

- Corredores compartidos para Windows.
- Infraestructura como código con Terraform.
- Auto DevOps.
- Pruebas dinámicas de seguridad (DAST).

FECHAS

Contacto