DevOps

Bootcamp Online

GeeksHubs academy

Adquiere nuevas habilidades y aprende

- **El rol de DevOps**
- **Arquitectura como código**
- **Infraestructuras híbridas**
- **Máquinas virtuales y contenedores**
- **Balanceos de carga**
- **Automatización de tareas**
- **Clustering de contenedores**
- **Continuous delivery**
- **Continuous testing**
- **Log analytics**
- **Orquestación y automatización**

de la mano de **profesionales experimentados y seniors** de grandes empresas y startups.

Programa

BLOQUE 0. Herramientas para DevOps

Durante este primer bloque vamos a revisar las herramientas base con las que trabajamos en el Bootcamp y necesarias para el alumno para poder seguir el aprendizaje del mismo sin problemas.
Empezaremos con Vagrant, Ansible y Docker, aprendiendo los conceptos básicos, cómo instalarlo en nuestro entorno de trabajo y ejecutarlos.

BLOQUE 1. Cultura DevOps. Diseño y metodologías de desarrollo

DevOps no es en sí una cultura, pero sí requiere de un fuerte cambio cultural y organizativo para su implementación. Un cambio cultural hacia la colaboración, la comunicación, y en último término la completa integración entre las antiguas áreas, en lo habitual rabiosamente estancas, de desarrollo y sistemas.

- **¿Qué es DevOps?** Introducción a la colaboración en DevOps. Cambio cultural y herramientas.
- **El rol de DevOps** en las compañías, sus capacidades y la adopción dentro de la empresa y los equipos de desarrollo.

BLOQUE 2.1. Agile Dev. y Herramientas

El Control de versiones con Git permite gestionar archivos de equipo para proyectos grandes y pequeños. Esto permite al equipo mejorar continuamente su producto. Es utilizado por la mayoría de las principales empresas de tecnología, y se ha convertido en esencial en cualquier equipo de desarrollo. Es un componente central de DevOps, canalizaciones de entrega continua y computación nativa de la nube.

- **Introducción a la metodología Agile** en los equipos y en los procesos internos de la compañía.
- **Herramientas y buenas prácticas** para la utilización de la metodología Agile: Repositorios de código, Ethical DevOps
- Comprender los conceptos más importantes de **Git/Gitflow** para el desarrollo de aplicaciones.
- Avanzar en conceptos de Git/Gitflow que mejoren la **eficiencia** del equipo de desarrollo.
- Aplicar **las mejores prácticas de desarrollo** usando Git/Gitflow.
  Diseñar **estrategias de trabajo** con Git/Gitflow dentro de los equipos de desarrollo.

Programa

BLOQUE 2.2. Bases de la infraestructura

Las herramientas de aprovisionamiento brindan formas uniformes de construir, cambiar e implementar infraestructura en múltiples plataformas en las instalaciones y en la nube. La reutilización de los boxes disminuye en gran medida el esfuerzo operativo, sin embargo aún quedan pendientes tareas que son repetibles y no requieren de la intervención de un humano, estas tareas son el foco de los programas de aprovisionamiento.

- Aprovisionar recursos como máquinas virtuales, balanceadores de carga, contenedores y funciones lambda en aws y en una **infraestructura híbrida**.

BLOQUE 2.3. Infraestructura como código

Los proyectos DevOps utilizan procesos estandarizados, permitiendo la automatización de estos para mejorar su fiabilidad y, por extensión, la calidad del SW. De esta forma, la automatización, combinada con otros principios de DevOps, facilita que los equipos puedan focalizarse en proporcionar valor a la entrega del software, siguiendo el primer principio del Manifiesto Agile: “Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor”. La automatización de las tareas de administración de la configuración ayuda a un equipo a ganar velocidad, agilidad y productividad.

- Conocer las **herramientas de aprovisionamiento** para especificar en detalle los ambientes de ejecución, de tal manera que puedan ser replicados de forma automatizada y repetible por medio de scripts.
- Usar la tecnología para **automatizar** estas tareas.

BLOQUE 2.4. Sistemas de gestión de la infraestructura

Vamos aprender a gestionar nuestra infraestructura como código, para ello utilizaremos las diferentes herramientas disponibles para poder crear, modificar o eliminar nuestra infraestructura, así como provisionarla con las herramientas necesarias.

- Usar la tecnología para **automatizar** estas tareas.
- Personalizar la pila de **despliegue del desarrollo** de un equipo con Puppet, Chef, Terraform, Ansible y Packer. Aunque se profundizará en Terraform.

Programa

BLOQUE 2.5. Desplegando contenedores

Gracias a la naturaleza intrínseca de la tecnología los contenedores, los desarrolladores pueden compartir el software y las dependencias fácilmente con los equipos de operaciones de TI y los entornos de producción, lo que pone fin a la típica excusa de "funciona en mi equipo". Los contenedores solucionan los conflictos de las aplicaciones entre distintos entornos. De manera indirecta, los contenedores acercan todavía más a los desarrolladores y los equipos de operaciones de TI, lo que les permite colaborar de forma eficaz.

- Construir contenedores, **Docker.**
- Componer aplicaciones de **múltiples contenedores** para soportar microservicios.
- Adoptar el **flujo de trabajo de contenedor** dentro del equipo.
- Simplificar las **canalizaciones de compilación, prueba e implementación** de DevOps.

BLOQUE 3.1. Desplegando con contenedores

Kubernetes es un gestor de clústeres de contenedores ampliamente adoptado en el mercado. Alguien que hace DevOps necesita un ambiente para crear pipelines de Continuous Integration / Continuous Delivery y muchos otros si quiere conocer la arquitectura al detalle y comenzar a instalarlo para producción.

- Conocer los conceptos básicos y avanzados de **Kubernetes**.
- Implementar aplicaciones en contenedores en los **clústeres de Kubernetes**.
- Crear y administrar con Kubernetes **clúster de grupos de hosts** que ejecutan contenedores.

BLOQUE 3.2. Herramientas de Orquestación

Un equipo efectivo de DevOps aprovecha la tecnología para impulsar la integración continua y un suministro continuo. En la superficie, la diferencia entre automatización y orquestación puede parecer semántica, pero comprender esta diferencia es clave para los equipos de TI que buscan implementar una cultura DevOps y mejorar sus procesos de TI. Tanto la automatización como la orquestación eliminan la carga de administrar las operaciones cotidianas de los equipos de TI para que puedan concentrarse en actividades estratégicas de valor agregado.

- Conocer las herramientas más apropiadas para la automatización y orquestación usando **Jenkins**.
- Configurar Jenkins para ejecutar tuberías, cobertura de código y herramientas de calidad, conjuntos de pruebas y herramientas de implementación y CM.
- **Creación de Pipelines** y procesos de automatización.

Programa

BLOQUE 3.3. Alta disponibilidad

Vamos a aprender como configurar alta disponibilidad en nuestra infraestructura.
Alta disponibilidad es un protocolo de diseño del sistema y su implementación asociada que asegura un cierto grado absoluto de continuidad operacional durante un período de medición dado.

- Conocer los conceptos y definir una infraestructura de **alta disponibilidad**.
- Diferenciar entre **alta disponibilidad y alta concurrencia**.
- Analizar cuales son lo **cuellos de botella** habituales.
- Crear con **kubernetes** una infraestructura de alta disponibilidad.

BLOQUE 3.4. Monitoring and Continuous Testing

El Continuous Testing es el proceso de ejecución de pruebas automatizadas como parte de la distribución de software para obtener feedback sobre los riesgos asociados con el desarrollo y puesta en producción de un software.
Las pruebas continuas van más allá de la automatización y abarcan todas las prácticas, incluidas las herramientas y el cambio cultural, que ayudan a mitigar los riesgos antes de pasar a las siguientes etapas del ciclo de vida de desarrollo de software.

- Integrar el **control de calidad** en los procesos de desarrollo y operaciones.
- Diseñar y ejecutar pruebas de **comportamiento**.
- Analizar los resultados de las pruebas de comportamiento con **jmeter**.
- Monitorizar la Infraestructura con **Cloud Watch.**

BLOQUE 3.5. Log Analytics

El Continuous Monitoring durante todo el ciclo de vida de DevOps puede conducir a una mejor colaboración entre Dev y Ops y ayudarlo a optimizar la experiencia del usuario en cada paso del camino, dejando más tiempo para su próximo sprint.

- Integrar la experiencia en Dev y Ops, abordando las **necesidades de los propietarios** de aplicaciones, profesionales de TI y DevOps.
- Descubrir cómo las **soluciones de monitoreo** ayudan a administrar, identificar, comprender y resolver problemas en sus aplicaciones y servicios web.
- Descubrirá cómo **recopilar, analizar y tomar decisiones** utilizando registros y otros datos generados por el sistema.
- Aprender herramientas como **Splunk**, **la pila de ELK** (Elasticsearch / Logstash / Kibana) o **Grafana**.

Nuestros profesores

Profesionales en activo de primer nivel, con reconocida **experiencia docente** impartiendo conferencias, workshops y cursos de formación en escuelas de negocio, entidades y empresas

Director Bootcamp

Xavi Rodríguez
CTO en GeeksHubs

Rubén Cougil
Software Engineer
en Adobe

Eduardo Escartí
Team Lead Engineer
en XING

Alicia Alcalde
DevOps Lead
en Broker Genius

David Pestana
Tech Advisor

Pedro Díaz
SRE Lead
en Mercadona Tech

Miquel Barceló
DevOps/SRE
en Mercadona Tech

Reinaldo José Leon
CTO
en Nubersia

David Lluna
Site Reliability Engineer
en Flywire

Juan Manuel Mesa
Senior DevOps Engineer
en Letgo

GeeksHubs
academy _

formacion@geekshubs.com