

CURSO DE

Infraestructura como Código y CI/CD Avanzado

Contenido

1. Infraestructura como Código (IaC)

Conceptos fundamentales

- Definición de Infraestructura como Código
- Problemas de la gestión manual de infraestructura
- Beneficios de IaC:
 - o reproducibilidad
 - o automatización
 - o control de versiones
 - o consistencia entre entornos
- Enfoques declarativos vs imperativos
- Herramientas comunes de IaC

Introducción a Terraform

- Arquitectura y funcionamiento de Terraform
- Instalación y configuración inicial
- Estructura básica de un proyecto Terraform
- Archivos principales:
 - o main.tf
 - o variables.tf
 - o outputs.tf
- Comandos básicos:
 - o terraform init
 - o terraform plan
 - o terraform apply
 - o terraform destroy

Aprovisionamiento de infraestructura

- Provisión de máquinas virtuales
- Creación de redes y subredes
- Configuración de grupos de seguridad
- Definición de servicios en la nube
- Uso de proveedores (providers)

Gestión de variables y estados

- Variables de entrada
- Variables por entorno
- Archivos .tfvars
- Manejo del estado de Terraform (terraform state)
- Backend remoto para almacenamiento de estado

Modularización y reutilización

- Creación de módulos reutilizables
- Organización de infraestructura en módulos
- Versionado de módulos
- Buenas prácticas de organización de proyectos Terraform

2. Automatización de configuración con Ansible

Introducción a la automatización de configuración

- Concepto de configuración de servidores
- Problemas de la configuración manual
- Infraestructura mutable vs inmutable
- Arquitectura de Ansible (control node y managed nodes)

Instalación y configuración inicial

- Instalación de Ansible en Linux
- Configuración básica del entorno
- Conexión remota mediante SSH
- Pruebas de conectividad con hosts remotos

Inventarios

- Inventarios estáticos
- Organización de hosts en grupos
- Variables de inventario
- Inventarios dinámicos

Playbooks

- Sintaxis YAML en Ansible
- Estructura de un playbook
- Tareas y módulos
- Ejecución de playbooks
- Idempotencia en la configuración

Roles

- Organización de playbooks mediante roles
- Estructura de directorios de roles
- Variables y dependencias entre roles
- Reutilización de roles en distintos proyectos

Automatización de despliegues

- Instalación automática de software
- Configuración de servicios
- Gestión de archivos y plantillas
- Ejecución remota de tareas administrativas

3. CI/CD avanzado

Conceptos de integración y entrega continua

- Integración continua (CI)
- Entrega continua (CD)
- Diferencias entre entrega y despliegue continuo
- Flujo de trabajo DevOps con pipelines

Pipelines de integración

- Estructura de pipelines
- Etapas (stages) y trabajos (jobs)
- Definición de pipelines como código
- Ejecución automática basada en eventos de Git

Herramientas de CI/CD

- Introducción a GitLab CI
- Introducción a Jenkins
- Configuración básica de pipelines

Automatización de pruebas y builds

- Compilación automática de proyectos
- Ejecución de pruebas automatizadas
- Análisis de calidad de código
- Validación de artefactos antes del despliegue

Despliegue automatizado

- Despliegue en múltiples entornos:
 - o desarrollo
 - o staging
 - o producción
- Gestión de variables de entorno
- Control de versiones de artefactos

Estrategias de despliegue

- Rolling deployment
- Blue/Green deployment
- Control de versiones en despliegues
- Reducción de tiempo de inactividad

Proyecto de módulo: Crear infraestructura en AWS con Terraform, configurarla con Ansible y desplegar la aplicación en Kubernetes.

Requisitos de conocimientos previos

Tener conocimientos básicos de DevOps y CI/CD inicial
Conocer Git y control de versiones (repositorios, commits, branches)
Manejar Docker y Kubernetes a nivel básico (crear imágenes, contenedores y servicios)
Familiaridad con entornos Linux (comandos básicos de terminal)

Preguntas frecuentes

1. *¿Este módulo enseña DevOps desde cero?*
No. El módulo está orientado a estudiantes que ya tienen conocimientos básicos de DevOps y desean profundizar en automatización de infraestructura y despliegues avanzados.
2. *¿Se trabaja con infraestructura real en la nube?*
Sí. Durante el módulo se realizarán prácticas donde se creará infraestructura utilizando servicios de nube pública.
3. *¿Es necesario tener experiencia previa con Terraform o Ansible?*
No es necesario. Ambos se introducen desde sus conceptos básicos antes de trabajar con configuraciones más avanzadas.
4. *¿El curso incluye prácticas?*
Sí. Cada unidad incluye ejercicios prácticos y culmina con un proyecto integrador donde se aplican todas las herramientas.
5. *¿Qué sistemas operativos se utilizarán durante el módulo?*
Las prácticas se realizan principalmente en entornos Linux, utilizando terminal y herramientas de automatización.
6. *¿Es necesario tener conocimientos de programación?*
No es imprescindible, pero es recomendable tener familiaridad con scripts, archivos de configuración y uso de terminal.