

ROADMAP: DEVOPS ENGINEER

Tu guía paso a paso para convertirte en un Ingeniero DevOps competitivo en el mercado laboral actual (fundamentos, nube, automatización y entrega).

FASE 1 : FUNDAMENTOS SÓLIDOS (MESES 1 - 3)

Antes de tocar la nube, necesitas entender la tierra firme. Sin estas bases, todo lo demás se tambalea.

Linux

- Navegación en terminal (`cd`, `ls`, `cat`, `grep`, `find`)
- Permisos de archivos y usuarios (`chmod`, `chown`, `sudo`)
- Gestión de procesos (`ps`, `top`, `kill`, `systemctl`)
- Editor de texto en terminal (`vim` o `nano`)
- Scripting básico en **Bash**
- SSH: conexión remota, llaves públicas/privadas

Redes (Networking)

- Modelo OSI y TCP/IP (al menos las capas clave)
- DNS: cómo se resuelve un nombre de dominio
- HTTP/HTTPS: métodos, códigos de estado, certificados SSL/TLS
- Subnetting y CIDR (imprescindible para VPCs)
- Firewalls, Security Groups y NACLs
- VPN y Peering

Programación Básica

- Un lenguaje de scripting: **Python** o **Bash** (idealmente ambos)
- Variables, condicionales, bucles, funciones
- Lectura/escritura de archivos
- Llamadas a APIs con `requests` o `curl`

FASE 2 : TU PRIMER PROVEEDOR CLOUD (MESES 3 - 6)

Elige **un solo proveedor** para empezar. AWS es el más demandado, pero Azure y GCP son igual de válidos.

Conceptos Core de la Nube

- Regiones y Zonas de Disponibilidad
- Modelo de Responsabilidad Compartida
- Modelos de precios: On-Demand, Reserved, Spot
- IAM: Usuarios, Roles, Políticas (¡esto es CRÍTICO!)

Servicios Esenciales (Ejemplo con AWS)

Categoría	Servicio	Para qué sirve
Cómputo	EC 2	Máquinas virtuales
Almacenamiento	S 3	Objetos (archivos, backups, sitios estáticos)
Base de Datos	RDS	Bases de datos relacionales gestionadas

Categoría	Servicio	Para qué sirve
Red	VPC	Tu red privada en la nube
DNS	Route 53	Gestión de dominios y enrutamiento
Balanceo	ELB/ALB	Distribuir tráfico entre servidores
Serverless	Lambda	Ejecutar código sin gestionar servidores
Contenedores	ECS / EKS	Orquestar contenedores Docker/K8s

Proyecto Práctico Fase 2

Despliega una aplicación web simple (por ejemplo, un sitio en HTML/CSS) en un bucket S3 con CloudFront como CDN y Route 53 como DNS. Documenta todo en GitHub.

FASE 3 : INFRAESTRUCTURA COMO CÓDIGO (IAC) (MESES 6 - 9)

Aquí es donde dejas de hacer clic en consolas y empiezas a **automatizar todo**.

Terraform

- Providers, Resources, Variables, Outputs
- Estado (`terraform.tfstate`): local vs. remoto (S3 + DynamoDB)
- Módulos reutilizables
- Workspaces para múltiples entornos (`dev`, `stg`, `prod`)
- `terraform plan` → `terraform apply` → `terraform destroy`
- Importar recursos existentes con `terraform import`

Alternativas a Conocer

- **CloudFormation** (nativo de AWS)
- **Pulumi** (IaC con lenguajes reales como Python/TypeScript)
- **OpenTofu** (fork open source de Terraform)

Proyecto Práctico Fase 3

Crea un módulo de Terraform que despliegue una VPC con subnets públicas y privadas, un Internet Gateway, NAT Gateway y Security Groups. Usa variables para hacerlo reutilizable.

FASE 4 : CONTENEDORES Y ORQUESTACIÓN (MESES 9 - 12)

Los contenedores son el estándar de la industria. No hay escape.

Docker

- ¿Qué es un contenedor vs. una VM?
- `Dockerfile`: construir imágenes propias
- `docker-compose`: aplicaciones multi-contenedor
- Registries: Docker Hub, ECR, GCR, ACR
- Buenas prácticas: imágenes livianas, multi-stage builds, `.dockerignore`

Kubernetes (K8s)

- Arquitectura: Control Plane, Nodes, Pods
- Objetos esenciales: Deployment, Service, Ingress, ConfigMap, Secret
- `kubectl`: tu mejor amigo
- Namespaces para aislamiento

- Helm Charts para despliegues complejos
- Servicios gestionados: EKS (AWS), AKS (Azure), GKE (GCP)

Proyecto Práctico Fase 4

Dockeriza una app con backend (API en Python/Node) y base de datos (PostgreSQL). Desplégala en un clúster de Kubernetes local (Minikube o Kind) y luego en EKS/AKS/GKE.

FASE 5 : CI/CD Y DEVOPS (MESES 1 2 - 1 5)

Automatizar el ciclo de vida del software: build → test → deploy.

Pipelines de CI/CD

- **GitHub Actions** (el más accesible para empezar)
- **GitLab CI/CD** (muy usado en empresas)
- **Jenkins** (el veterano, aún prevalente)
- **ArgoCD** (GitOps para Kubernetes)

Conceptos Clave

- Integración Continua (CI): tests automáticos en cada commit
- Entrega Continua (CD): deploy automático a staging
- Despliegue Continuo: deploy automático a producción
- Estrategias de despliegue: Blue/Green, Canary, Rolling Update
- GitOps: el repositorio Git como fuente de verdad

Proyecto Práctico Fase 5

Configura un pipeline en GitHub Actions que al hacer push a `main`:
1 . Ejecute tests unitarios
2 . Construya la imagen Docker
3 . La suba a ECR/Docker Hub
4 . Actualice el deployment en Kubernetes via ArgoCD

FASE 6 : OBSERVABILIDAD Y MONITOREO (MESES 1 5 - 1 8)

Si no puedes medirlo, no puedes mejorarlo. Si no puedes verlo, no puedes arreglarlo.

Los Tres Pilares

Pilar	Herramienta	Qué resuelve
Métricas	Prometheus + Grafana	¿Qué tan bien rinde mi sistema?
Logs	ELK Stack / Loki	¿Qué pasó exactamente?
Trazas	Jaeger / OpenTelemetry	¿Dónde se atascó la request?

Alertas y On-Call

- Definir SLIs, SLOs y Error Budgets
- Configurar alertas inteligentes (no ruidosas)
- PagerDuty / Opsgenie para gestión de incidentes
- Runbooks: documentar cómo responder a cada alerta

Proyecto Práctico Fase 6

Instala Prometheus y Grafana en tu clúster K8s con Helm. Crea dashboards para monitorear pods, nodos y tu aplicación. Configura al menos 3 alertas significativas.

Aquí defines tu camino único. Elige una o dos áreas de profundización.

Seguridad Cloud (Cloud Security)

- Gestión de secretos: Vault, AWS Secrets Manager
- Políticas de red: Network Policies en K8s
- Escaneo de vulnerabilidades: Trivy, Snyk
- Compliance y auditoría

Arquitectura y Diseño

- Arquitecturas multi-cuenta / multi-región
- Alta disponibilidad y recuperación ante desastres (DR)
- Migraciones de on-premise a la nube
- Optimización de costos (FinOps)

Platform Engineering

- Internal Developer Platforms (IDPs)
- Backstage / Port para catálogos de servicios
- Golden Paths: caminos estándar para desarrolladores
- Self-service infrastructure

RESUMEN VISUAL DEL ROADMAP



RECURSOS GRATUITOS PARA APRENDER

No necesitas gastar una fortuna para empezar. Estos recursos son 100% **gratuitos** y de altísima calidad:

Recurso	Qué encuentras
 AWS Skill Builder 	Cursos oficiales de AWS, laboratorios y exámenes de práctica
 Microsoft Learn 	Learning paths de Azure con sandbox gratuito
 Google Cloud Skills Boost 	Labs prácticos de GCP
 HashiCorp Learn 	Tutoriales oficiales de Terraform
 Kubernetes.io 	Documentación oficial de K8s

🚀 ACELERA CON CURSOS, LIVES Y SOPORTE (COMUNIDAD SKOOL)

Este roadmap ordena tu ruta; la parte difícil es **practicar** sin quedarte días frenado ante un solo error en consola o en un pipeline. En la comunidad de Skool combinamos **cursos** en la plataforma, **sesiones en vivo** y **soporte técnico** cuando algo no levanta. Además practicas en   **sandboxes temporales AWS, Azure y GCP**: nube real, sin cobros por sorpresa.

 **Entra en la comunidad — skool.com/jemjaf** 

Propuesta práctica para quien estudia **DevOps y cloud**:

-  **Cursos** grabados + **sesiones en vivo** (cloud, infra y herramientas — según calendario del grupo)
-   **Sandboxes temporales en AWS, Azure y GCP**: nube real, reglas claras y sin el susto del cobro fuera de lugar
-  **Soporte directo**: destrabamos labs, mensajes de error y arquitectura pequeña con criterio
-  **Career Center**: feedback de CV y LinkedIn desde el lado técnico que miran reclutadores
-  **Recursos alineados** con lo que enseño arriba: guías, retos breves, portafolio

 **Únete en skool.com/jemjaf**  — Formación y acompañamiento; sin promesa de trabajo mágico. El objetivo es que **lleves proyecto y conversación técnica** que puedas defender.

Sígueme en Redes

-  **jemjaf.com**  — Blog, recursos y portafolio
-  **TikTok — [@jemjaf](https://www.tiktok.com/@jemjaf)**  — Tips rápidos de cloud y DevOps
-  **Instagram — [@jemjaf](https://www.instagram.com/jemjaf)**  — Carruseles, infografías y motivación

TIP

La regla de oro: Por cada tecnología que estudies, construye un proyecto práctico y documéntalo en GitHub. Tu portafolio habla más fuerte que cualquier certificación. **¡Manos a la obra!** 