

Plan intensivo Cloud/DevOps + Python/Go – 12 meses

Objetivo

- Conseguir primer trabajo **Cloud/DevOps mid-level** (tu background IT te da ventaja).
 - Certificaciones reconocidas (AWS/Azure + Kubernetes + Terraform).
 - Portafolio en GitHub con **proyectos prácticos + código real** (Python/Go).
 - Inglés fluido en entrevistas técnicas.
-

Fase 1: Fundamentos + Programación básica (Mes 1–2)

Meta: sentar base en Linux, redes, Git, Cloud y empezar con un lenguaje (Python como base; Go si te sobra tiempo).

◆ Aprendizaje

- **Linux CLI avanzado:** permisos, procesos, logs, networking.
- **Redes:** TCP/IP, DNS, HTTP/HTTPS, VPN, firewalls.
- **Git/GitHub:** ramas, PRs, workflows.
- **Python:**
 - Sintaxis, funciones, scripts CLI.
 - Librerías clave: boto3 (AWS SDK), requests, os/subprocess.
- (Opcional paralelo) **Go (Golang):** fundamentos, CLI apps simples.
- **Cloud:** AWS Certified Cloud Practitioner (CCP).

◆ Proyecto práctico

- Script en **Python** que crea y elimina instancias EC2 en AWS usando boto3.
 - Montar un servidor web en AWS (EC2 con Ubuntu + Nginx/Apache).
 - Añadir dominio y SSL (Let's Encrypt).
 - Documentarlo en GitHub con capturas.
-

Fase 2: Infraestructura como código + Contenedores + Python aplicado (Mes 3–5)

Meta: dominar IaC, Docker y aplicar programación a automatización.

◆ Aprendizaje

- **Terraform:** redes, VMs, S3, RDS.
- **Docker:** imágenes, docker-compose, DockerHub.
- **CI/CD:** GitHub Actions, GitLab CI.

- **Python avanzado:**
 - Scripts para backups en S3, lambdas sencillas.
 - Automación de pipelines con subprocess y yaml.
 - **Go** (si lo incluyes):
 - CLI que hace health-checks de un servicio web.
 - Certificación: **AWS Solutions Architect Associate (SAA)**.
 - ♦ Proyecto práctico
 - Infra con Terraform: VPC + EC2 + RDS + S3.
 - App dockerizada desplegada en AWS.
 - **Pipeline CI/CD** que construya y despliegue.
 - Script en Python/Go que verifique la salud del despliegue.
-

📌 Fase 3: Kubernetes + Observabilidad + Go/Python intermedio (Mes 6–8)

Meta: estar “empleable” como DevOps + añadir skills de código a k8s.

- ♦ Aprendizaje
 - **Kubernetes:** pods, deployments, ingress, configmaps, secrets.
 - **Helm charts.**
 - **Prometheus + Grafana** (monitoring).
 - **Logging:** ELK (Elastic, Logstash, Kibana).
 - **Python:** scripts que interactúan con la API de Kubernetes (kubernetes client).
 - **Go:** CLI simple que despliegue un pod en k8s usando la API.
 - Certificación: **CKA (Certified Kubernetes Administrator)**.
 - ♦ Proyecto práctico
 - Desplegar app de 3 capas (frontend + backend + DB) en Kubernetes.
 - Añadir observabilidad con Prometheus/Grafana.
 - Exponerlo en dominio público con Ingress + SSL.
 - Script en **Python/Go** para automatizar despliegues/rollbacks.
-

📌 Fase 4: Especialización + Portfolio + Entrevistas (Mes 9–12)

Meta: consolidar perfil con proyectos de código + infra y lanzarse al mercado.

- ♦ Aprendizaje

- Seguridad en la nube (IAM, roles, secretos, backups).
 - Cost optimization (AWS billing).
 - Opcional: empezar **Azure AZ-104** o **GCP Associate Engineer**.
 - Preparación de entrevistas (DevOps exercises, system design).
- ◆ Proyecto práctico (portfolio final)
1. **Terraform + AWS**: infra de un e-commerce (S3 + API Gateway + Lambda en Python).
 2. **Kubernetes + Observabilidad**: microservicios + Helm + Prometheus/Grafana.
 3. **Herramienta CLI en Go/Python**: deploy en AWS/EKS + health checks.
- ◆ Acción
- CV en inglés, LinkedIn optimizado.
 - GitHub con **README profesional** + capturas + código.
 - Aplicar a posiciones Cloud/DevOps en España y remoto internacional.
 - **Objetivo salarial inicial**: 40k–50k España / 65k–75k remoto.
-

✿ Horas recomendadas (si vas “all-in”)

- 20–25h semanales si trabajas en paralelo.
 - 30–40h semanales si puedes dedicarte casi full.
 - Resultado: **empleable en 6–8 meses**, competitivo internacional en 12.
-

📊 Plan de Certificaciones

- **Mes 1–2** → AWS CCP ✅
 - **Mes 3–5** → AWS SAA ✅
 - **Mes 6–8** → CKA (Kubernetes) ✅
 - **Mes 9–12** → Opcional: Azure AZ-104 o GCP AE
-

🔗 Conclusión

Con este ajuste:

- **Python** (seguro) y/o **Go** (opcional) te darán **poder extra** para automatizar infra y destacar frente a otros DevOps.
- En tus proyectos de portfolio ya tendrás **infra + pipelines + código real**, lo que te pondrá en ventaja en entrevistas.

- En 12 meses → estarías listo para roles de 40k–60k en España o 65k–90k fuera/remoto.

✿ Proyectos en Python (base obligatoria)

📌 Nivel 1 – Mes 1–2 (básico, scripts útiles)

1. Gestor de instancias EC2 con boto3

- Script que crea, lista y elimina instancias EC2 en AWS.
- Añade opciones con argparse (--start, --stop, --terminate).

2. Health-check de sitios web

- Script que recibe una lista de URLs y devuelve si están activas (HTTP 200) o caídas.
- Guardar resultados en un .json o .csv.

3. Backup automático a S3

- Script que sube ficheros locales a un bucket S3 con fecha en el nombre.
- Añadir logging y control de errores.

📌 Nivel 2 – Mes 3–5 (con infra y CI/CD)

4. Lambda function en Python

- Lambda que se dispara cuando subes un archivo a S3 → convierte a .zip o cambia extensión.
- Infra desplegada con Terraform.

5. CLI para despliegues en Docker

- Python script que construye una imagen Docker y la sube a DockerHub con un solo comando.
- Integrar con un pipeline en GitHub Actions.

📌 Nivel 3 – Mes 6–8 (con Kubernetes)

6. Kubernetes Deployment Manager

- Script que, usando la librería oficial kubernetes, despliega un pod en k8s.
- Opciones: --deploy, --delete, --status.

7. Log collector

- Script que recoge logs de un pod de Kubernetes y los guarda en un bucket S3.
-

Nivel 4 – Mes 9–12 (proyecto de portfolio avanzado)

8. Infra completa con automatización en Python

- Terraform crea la infra (EKS + S3 + RDS).
 - Script en Python automatiza:
 - Deploy de microservicio en EKS.
 - Configuración de buckets.
 - Health-check y rollback si falla.
-

Proyectos en Go (Golang) (opcional, para destacar)

Nivel 1 – Mes 1–2

1. CLI básico “Hello Cloud”

- CLI que imprime mensajes con flags (--hello AWS, --hello DevOps).

2. HTTP health-checker

- CLI que recibe una lista de URLs y muestra status codes en terminal.
-

Nivel 2 – Mes 3–5

3. Mini servidor REST en Go

- API que responde con "status: ok".
 - Dockerizarlo y desplegarlo en AWS.
-

Nivel 3 – Mes 6–8

4. CLI para Kubernetes

- Programa en Go que lista pods y sus estados en un cluster k8s.
 - Usa client-go (SDK oficial).
-

Nivel 4 – Mes 9–12

5. Herramienta DevOps real

- CLI en Go que:

- Despliega manifests en k8s.
 - Hace port-forward a un servicio.
 - Recoge logs.
 - Documentar con README, ejemplos y capturas.
-

Cómo usarlos en tu portfolio

- Sube cada proyecto a **GitHub** en repos separados.
 - Añade un **README claro** con:
 - Descripción (qué problema resuelve).
 - Pasos para ejecutar.
 - Ejemplo de uso (captura o comando).
 - Tecnologías usadas (AWS, Terraform, Docker, k8s).
 - En LinkedIn ponlos como **“Proyectos destacados”**.
-

Conclusión

- **Python**: tu “arma principal” para automatizar AWS, CI/CD y k8s.
- **Go**: extra que te hace destacar en cloud-native (muy valorado en entrevistas).
- Con 6–8 proyectos bien documentados ya puedes competir en entrevistas.