Español

✓ 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.

Pase 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.
 - o 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.
 - o Automación de pipelines con subprocess y yaml.
- Go (si lo incluyes):
 - o 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 \rightarrow AWS CCP \checkmark
- Mes 3–5 → AWS SAA
- Mes 6–8 → CKA (Kubernetes)
- Mes 9–12 → Opcional: Azure AZ-104 o GCP AE

S 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

- o Script que crea, lista y elimina instancias EC2 en AWS.
- o 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.
- o Guardar resultados en un .json o .csv.

3. Backup automático a S3

- o Script que sube ficheros locales a un bucket S3 con fecha en el nombre.
- o 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.
- o 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.
- o Integrar con un pipeline en GitHub Actions.

P Nivel 3 − Mes 6−8 (con Kubernetes)

6. Kubernetes Deployment Manager

o Script que, usando la librería oficial kubernetes, despliega un pod en k8s.

o Opciones: --deploy, --delete, --status.

7. Log collector

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

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

8. Infra completa con automatización en Python

- o 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)

9 Nivel 1 − Mes 1−2

1. CLI básico "Hello Cloud"

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

2. HTTP health-checker

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

P Nivel 2 – Mes 3–5

3. Mini servidor REST en Go

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

Nivel 3 − Mes 6−8

4. CLI para Kubernetes

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

9 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:
 - o Descripción (qué problema resuelve).
 - o Pasos para ejecutar.
 - o Ejemplo de uso (captura o comando).
 - o Tecnologías usadas (AWS, Terraform, Docker, k8s).
- En LinkedIn ponlos como "Proyectos destacados".

65 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.

English

✓ Intensive Cloud/DevOps + Python/Go Plan – 12 months

© Goal

- Land your first mid-level Cloud/DevOps job (your IT background gives you an advantage).
- Earn recognized certifications (AWS/Azure + Kubernetes + Terraform).
- Build a GitHub portfolio with practical projects + real code (Python/Go).
- Fluent English for technical interviews.

Phase 1: Fundamentals + Basic Programming (Months 1–2)

Goal: Build a foundation in Linux, networking, Git, Cloud, and start with one language (Python as base; Go if you have extra time).

Learning

- Advanced Linux CLI: permissions, processes, logs, networking.
- Networking: TCP/IP, DNS, HTTP/HTTPS, VPN, firewalls.
- Git/GitHub: branches, PRs, workflows.
- Python:
 - Syntax, functions, CLI scripts.
 - Key libraries: boto3 (AWS SDK), requests, os/subprocess.
 - (Optional parallel) Go (Golang): basics, simple CLI apps.
 - Cloud: AWS Certified Cloud Practitioner (CCP).

Practical Project

- Python script to create/delete EC2 instances in AWS using boto3.
- Set up a web server on AWS (EC2 with Ubuntu + Nginx/Apache).
- Add domain and SSL (Let's Encrypt).
- Document everything on GitHub with screenshots.

Phase 2: Infrastructure as Code + Containers + Applied Python (Months 3–5)

Goal: Master IaC, Docker, and apply programming for automation.

Learning

- Terraform: networks, VMs, S3, RDS.
- Docker: images, docker-compose, DockerHub.
- CI/CD: GitHub Actions, GitLab CI.
- Advanced Python:
 - Scripts for S3 backups, simple Lambda functions.
 - Automate pipelines with subprocess and YAML.
 - Go (optional):
 - CLI performing health checks for a web service.
 - Certification: AWS Solutions Architect Associate (SAA).

Practical Project

- Infrastructure with Terraform: VPC + EC2 + RDS + S3.
- Dockerized app deployed on AWS.
- CI/CD pipeline to build and deploy.
- Python/Go script to verify deployment health.

Phase 3: Kubernetes + Observability + Intermediate Go/Python (Months 6–8)

Goal: Be "hireable" as DevOps + add coding skills to Kubernetes.

Learning

- Kubernetes: pods, deployments, ingress, configmaps, secrets.
- Helm charts.
- Prometheus + Grafana (monitoring).
- Logging: ELK (Elastic, Logstash, Kibana).

- Python: scripts interacting with Kubernetes API (kubernetes client).
- Go: simple CLI to deploy a pod in k8s using API.
- Certification: CKA (Certified Kubernetes Administrator).

Practical Project

- Deploy a 3-tier app (frontend + backend + DB) in Kubernetes.
- Add observability with Prometheus/Grafana.
- Expose it publicly with Ingress + SSL.
- Python/Go script to automate deployments/rollbacks.

Phase 4: Specialization + Portfolio + Interviews (Months 9–12)

Goal: Consolidate your profile with projects (code + infra) and enter the job market.

Learning

- Cloud security (IAM, roles, secrets, backups).
- Cost optimization (AWS billing).
- Optional: start Azure AZ-104 or GCP Associate Engineer.
- Interview prep (DevOps exercises, system design).

Practical Project (Final Portfolio)

- 1. Terraform + AWS: infra for e-commerce (S3 + API Gateway + Lambda in Python).
- 2. Kubernetes + Observability: microservices + Helm + Prometheus/Grafana.
- 3. CLI tool in Go/Python: deploy on AWS/EKS + health checks.

Action

- CV in English, optimized LinkedIn.
- GitHub with professional README + screenshots + code.
- Apply to Cloud/DevOps positions in Spain and internationally remote.
- Initial salary goal: €40k–50k Spain / \$65k–75k remote.

Recommended hours ("all-in")

- 20–25h/week if studying alongside a job.
- 30-40h/week if mostly full-time.
- Result: employable in 6–8 months, internationally competitive in 12.

Certification Plan

- Months 1–2 → AWS CCP
- Months 3–5 → AWS SAA
- Months 6–8 → CKA (Kubernetes)
- Months 9–12 → Optional: Azure AZ-104 or GCP AE

© Conclusion

With this plan:

- Python (mandatory) and/or Go (optional) gives extra power to automate infra and stand out as DevOps.
- Portfolio projects already include infra + pipelines + real code, giving an advantage in interviews.
- In 12 months → ready for €40k–60k roles in Spain or \$65k–90k remote/international.

Python Projects (mandatory base)

← Level 1 – Months 1–2 (basic, useful scripts)

- 1. EC2 instance manager with boto3
- Script to create, list, and delete EC2 instances in AWS.
- Add argparse options (--start, --stop, --terminate).
- 2. Website health-check
- Script receiving a list of URLs, returning active (HTTP 200) or down.
- Save results in .json or .csv.
- 3. Automatic backup to S3
- Script uploads local files to an S3 bucket with date in filename.
- Add logging and error handling.

Level 2 – Months 3–5 (with infra and CI/CD)

- 4. Python Lambda function
 - Triggered when a file is uploaded to S3 \rightarrow convert to .zip or change extension.
 - Infra deployed with Terraform.
 - 5. Docker deployment CLI
 - Python script builds a Docker image and uploads to DockerHub with one command.
 - Integrate with GitHub Actions pipeline.

Level 3 – Months 6–8 (with Kubernetes)

- 6. Kubernetes Deployment Manager
 - Script using official Kubernetes library to deploy a pod in k8s.
 - Options: --deploy, --delete, --status.
 - 7. Log collector
 - Script collects logs from a Kubernetes pod and stores in S3 bucket.

Level 4 – Months 9–12 (advanced portfolio project)

- 8. Full infra with automation in Python
 - Terraform creates infra (EKS + S3 + RDS).
 - Python script automates:
 - Deploy microservice in EKS.
 - Configure buckets.
 - Health-check and rollback if failure.

So Projects (optional, to stand out)

- 1. Basic CLI "Hello Cloud"
- CLI prints messages with flags (--hello AWS, --hello DevOps).
- 2. HTTP health-checker
- CLI receives a list of URLs and shows status codes in terminal.

P Level 2 − Months 3−5

- 3. Mini REST server in Go
 - API responds with "status: ok".
 - Dockerize and deploy on AWS.

Level 3 – Months 6–8

- 4. Kubernetes CLI
 - Go program lists pods and their status in a k8s cluster.
 - Uses client-go (official SDK).

9 Level 4 − Months 9–12

- 5. Real DevOps tool
 - Go CLI that:
 - Deploys manifests to k8s.
 - Performs port-forwarding to a service.
 - Collects logs.
 - Document with README, examples, and screenshots.

How to use them in your portfolio

- Upload each project to GitHub in separate repos.
- Add clear README with:
 - Description (problem it solves).

- Steps to run.
- Usage examples (screenshots or commands).
- Technologies used (AWS, Terraform, Docker, k8s).
 - Highlight projects on LinkedIn as "Featured Projects".

© Conclusion

- Python: your main tool to automate AWS, CI/CD, and k8s.
- Go: extra skill to stand out in cloud-native environments (highly valued in interviews).
- With 6–8 well-documented projects, you can already compete in technical interviews.