

DEVOPS

ADVANCED

5 DAYS

Advanced GoLang Bootcamp (AWS + EKS + Terraform + GitLab CI/CD)

Build a strong foundation in production-grade Go backend engineering, from clean architecture to secure APIs and enterprise observability. Learn how to build, containerize, test, and deploy microservices with PostgreSQL, authentication, metrics, CI/CD automation, and Kubernetes operations aligned to real production standards.

Why Advanced GoLang Bootcamp?

High-performance services

Build fast, scalable backend systems for modern business workloads.

Improved maintainability

Clean architecture patterns lead to stable, production-ready codebases.

Better cloud-native development

Go is widely used for Kubernetes, tooling, and microservices ecosystems.

Faster product delivery

Teams ship reliable APIs and services with fewer runtime surprises.

Reduced infrastructure cost

Efficient Go services consume fewer resources at scale.

What you'll be able to do

- ✓ Master Go language fundamentals with a backend engineering mindset (Java/.NET transition).
- ✓ Build clean architecture microservices with layered design and interface-driven development.
- ✓ Develop REST APIs with middleware, validation, and clean error handling.
- ✓ Implement JWT authentication, RBAC patterns, and security-focused API practices.
- ✓ Integrate PostgreSQL using repository patterns, migrations, pooling, and transactions.
- ✓ Implement config management using 12-factor patterns and safe secrets handling.
- ✓ Write unit tests, integration tests, and benchmarks for performance validation.
- ✓ Implement structured logging, Prometheus metrics, and production health endpoints.
- ✓ Containerize applications using multi-stage builds and distroless images.
- ✓ Build GitLab CI/CD pipelines for build, test, scan, push, and deploy workflows.
- ✓ Provision AWS infrastructure using Terraform including VPC, RDS, and EKS.
- ✓ Deploy Go services to EKS with ingress, autoscaling, and monitoring.
- ✓ Harden production services using graceful shutdown, rate limiting, retries, and reliability patterns.

Who this is for

- Experienced Java Developers (Spring Boot, backend services)
- Experienced .NET Developers (ASP.NET Core, backend APIs)
- Backend Engineers building production microservices
- DevOps and Platform Engineers supporting backend delivery
- SRE engineers improving reliability, observability, and automation
- Technical leads modernizing backend architecture and tooling

Curriculum

01 Go Fundamentals for Backend Engineers (Java/.NET Transition)

- Go language philosophy and where it fits in backend engineering
- Core syntax and types (structs, slices, maps, pointers)
- Functions, methods, interfaces, and composition
- Concurrency fundamentals (goroutines, channels, context basics)
- **Hands-on:** Lab: Build small CLI utilities to practice Go basics and concurrency

02 Clean Architecture Microservices in Go

- Clean architecture layers (handlers, services/usecases, repositories)
- Interface-driven development and dependency injection patterns
- Project structure for maintainable microservices
- Error handling strategy across layers (domain vs transport)
- **Hands-on:** Lab: Scaffold a clean architecture Go microservice with proper folder structure

03 REST API Development (Middleware + Validation + Errors)

- Building REST endpoints (routing, handlers, request/response patterns)
- Middleware design (logging, auth, request id, timing)
- Validation patterns (schema validation, custom validators)
- Consistent API error responses (status codes, error envelope)
- **Hands-on:** Lab: Build CRUD REST APIs with middleware and validation

04 Authentication and Authorization (JWT + RBAC)

- JWT fundamentals (claims, signing, expiry, refresh basics)
- RBAC patterns (roles, permissions, scopes)
- Security best practices for APIs (rate limiting intro, secure headers concept)
- Protecting endpoints using middleware and policy checks
- **Hands-on:** Lab: Implement JWT auth + RBAC for protected REST endpoints

05 PostgreSQL Integration (Repo Pattern + Migrations + Transactions)

- Repository patterns for clean separation of persistence layer
- DB migrations strategy (versioned schema evolution)
- Connection pooling and configuration tuning basics
- Transactions and consistency patterns (commit/rollback, isolation awareness)
- **Hands-on:** Lab: Integrate PostgreSQL with CRUD + migrations + transactional workflow

06 Configuration Management and Secrets (12-Factor Patterns)

- 12-factor config principles (env-driven configuration)
- Config loading patterns and validation at startup
- Safe secrets handling (no hardcoding, secret injection concepts)
- Environment separation (dev/stage/prod)
- **Hands-on:** Lab: Implement config loader with env overrides and secrets-safe patterns

07 Testing Strategy (Unit, Integration, Benchmarks)

- Unit testing in Go (table-driven tests, mocks, interfaces)
- Integration testing with database dependencies
- Benchmarking and performance testing fundamentals
- Test coverage and CI integration best practices
- **Hands-on:** Lab: Write unit + integration tests and run benchmarks for API and repository layers

08 Production Observability (Logging, Metrics, Health Endpoints)

- Structured logging patterns (JSON logs, correlation ids)
- Prometheus metrics basics (counters, histograms, gauges)
- Health endpoints (liveness/readiness patterns)
- Operational visibility (latency, errors, traffic signals)
- **Hands-on:** Lab: Add structured logs + Prometheus metrics + health endpoints to the Go service

09 Containerization (Multi-Stage Builds + Distroless)

- Multi-stage Docker builds for small secure images
- Distroless images concept and reduced attack surface
- Tagging strategies (versioning + git sha) and immutability
- Container runtime hardening (non-root user, minimal permissions)
- **Hands-on:** Lab: Containerize the Go service with multi-stage build + distroless runtime image

10 GitLab CI/CD for Go Microservices (Build, Test, Scan, Push, Deploy)

- GitLab pipeline structure (stages, jobs, artifacts)
- Automating build and tests with caching
- Security scans in CI (dependency scan concepts, image scan intro)
- Build + push Docker images to registry and deploy workflow patterns
- **Hands-on:** Lab: Build GitLab CI pipeline for test → build → scan → push → deploy

11 Provision AWS Infrastructure with Terraform (VPC, RDS, EKS)

- Terraform workflow (init, plan, apply, state basics)
- Provisioning a VPC with subnets, routing, and security rules
- Provisioning RDS PostgreSQL with secure networking
- Provisioning EKS cluster basics (node groups, IAM roles concept)
- **Hands-on:** Lab: Deploy AWS VPC + RDS + EKS using Terraform modules

12 Deploy Go Services to EKS (Ingress, Autoscaling, Monitoring)

- Kubernetes manifests for Go apps (Deployment, Service, ConfigMap, Secret)
- Ingress setup and routing concepts
- Autoscaling basics (HPA and scaling triggers concept)
- Monitoring integration overview (Prometheus metrics scraping patterns)
- **Hands-on:** Lab: Deploy the Go service to EKS with ingress + autoscaling and validate metrics

13 Production Hardening (Reliability and Safety Patterns)

- Graceful shutdown patterns using context cancellation
- Rate limiting and abuse prevention basics
- Retries and timeouts for outbound dependencies
- Resilience patterns for production stability
- **Hands-on:** Capstone Lab: Harden service reliability and validate behavior under failures and load