

CLOUD

BEGINNER

2 DAYS

Cloud-Native Services and Serverless Starter Fast Track

Build a strong foundation in building cloud-native serverless APIs and event-driven automations, from core concepts to real implementations. Learn how to apply secure and reliable production basics to deliver scalable APIs and automation workflows with minimal operational overhead.

Why Cloud-Native Serverless Fast Track?

Ship faster

Build and deploy without server management overhead.

Reduce operations load

Managed services handle scaling and availability.

Optimize costs

Pay only for execution time and scale down automatically.

Modernize with events

Enable async automation and reliable background processing.

Improve reliability

Built-in retries and DLQs reduce failure impact.

What you'll be able to do

- ✓ Understand cloud-native and serverless fundamentals and where they fit best.
- ✓ Build and deploy a serverless HTTP API with validation and logging.
- ✓ Integrate serverless compute with managed object storage and NoSQL database.
- ✓ Design event-driven workflows using queues or pub/sub triggers.
- ✓ Apply IAM least privilege and safe secrets handling practices.
- ✓ Implement reliability patterns including retries, timeouts, and DLQ usage.
- ✓ Deliver a mini capstone demonstrating API to async processing workflow.

Who this is for

- Developers and Backend Engineers
- DevOps and Cloud Engineers
- Platform and SRE Teams (starter level)
- Solution Architects (foundation learning)
- Engineering Leads planning modernization
- Teams starting cloud-native adoption

Curriculum

01 Cloud-Native and Serverless Fundamentals (Where It Fits Best)

- Cloud-native basics (managed services, elasticity, automation mindset)
- Serverless fundamentals (functions, managed runtimes, pay-per-use)
- Serverless vs containers vs VMs (decision guide)
- Common serverless workloads (APIs, automation, event processing)
- **Hands-on:** Activity: Pick 3 real use cases and decide serverless vs Kubernetes vs VM

02 Build and Deploy a Serverless HTTP API (Validation + Logging)

- Designing HTTP APIs for serverless (routes, versioning, contracts)
- Request validation patterns (schema checks, input guards)
- Structured logging fundamentals (correlation ids, log levels)
- API deployment workflow (stages, environments)
- **Hands-on:** Lab: Build and deploy a serverless HTTP API with validation and request logging

03 Integrate Serverless Compute with Object Storage and NoSQL

- Object storage fundamentals (buckets, keys, metadata patterns)
- NoSQL fundamentals (partition keys, query patterns, consistency concepts)
- Typical API flow (write metadata to NoSQL, store payload in object storage)
- Error handling and idempotency patterns for writes
- **Hands-on:** Lab: API writes object to storage and metadata to NoSQL with safe error handling

04 Event-Driven Workflows with Queues or Pub/Sub Triggers

- Event-driven architecture basics (producers, consumers, decoupling)
- Queue vs pub/sub (when to use which)
- Async processing flow (API → queue → worker function)
- Message payload design and schema versioning
- **Hands-on:** Lab: Trigger async processing using a queue or pub/sub from API events

05 IAM Least Privilege and Safe Secrets Handling

- IAM fundamentals for serverless (roles, policies, service identities)
- Least privilege principle and common mistakes
- Secrets management concepts (do not hardcode secrets)
- Secure access to storage, database, and queues
- **Hands-on:** Lab: Apply least-privilege IAM permissions for API and worker function

06 Reliability Patterns (Retries, Timeouts, and DLQ)

- Timeouts and why serverless needs strict execution control
- Retry behavior (what to retry vs what not to retry)
- Dead Letter Queue (DLQ) usage and failure isolation
- Designing idempotent consumers for safe retries
- **Hands-on:** Lab: Configure retries, timeouts, and DLQ and validate failure recovery

07 Mini Capstone (API → Async Workflow)

- Capstone goal: Build an API-to-async processing workflow
- HTTP API receives request, validates input, logs activity
- Store data in object storage and NoSQL
- Trigger async processing using queue/pubsub and handle failures with DLQ
- **Hands-on:** Capstone Lab: Deliver the full end-to-end workflow with evidence and runbook notes

08 Observability with Prometheus and Grafana (Monitoring Essentials)

- Observability basics (metrics vs logs vs traces)
- Prometheus concepts (scrape, targets, exporters, labels)
- Grafana dashboards and alerting basics
- SLO mindset (latency, errors, throughput) for APIs and workers
- **Hands-on:** Lab: Build dashboards and alerts for API/worker metrics using Prometheus + Grafana