Software Development

Program Brochure · 2026

MICROSERVICES ARCHITECTURE

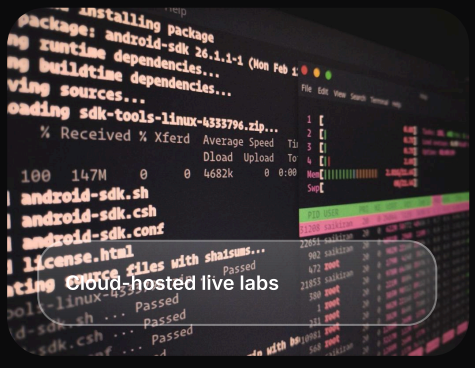
Build systems from small services.

Advanced ↗ | 72 Hours ↗ | SAM-MSA ↗

Software Development

Code that ships.

18 programs in this track



7

modules · hands-on labs · capstone project

Inside this brochure

Contents

Program overview 03

Who it's for 04

Learning outcomes 05

Program runway 06

Detailed curriculum 07

Tools & frameworks 10

Projects & capstone 11

Certification & method 12

Learning path 13

FAQs 14



AT A GLANCE

Level	**Advanced**
Duration	**72 Hours**
Modules	**7**
Certificate	**SAM-MSA**

Program overview

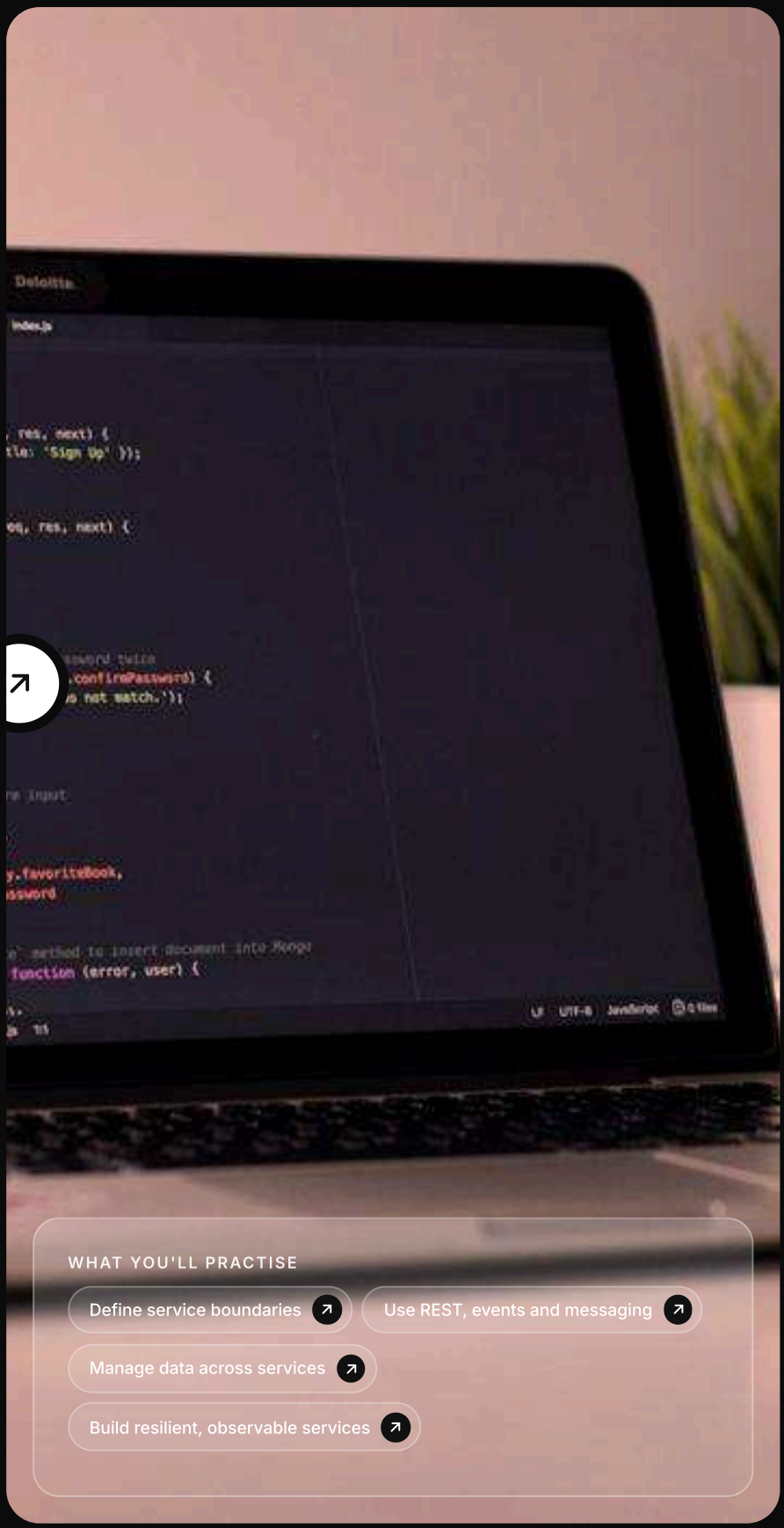
Build systems from small services.

Design, build and operate microservices — boundaries, communication, data and resilience.

As products grow, many teams split monoliths into independently deployable services, but doing it well requires careful design around boundaries, data and failure. This advanced program covers domain-driven decomposition, synchronous and event-driven communication with Kafka, data patterns like sagas and outbox, resilience, observability, security and Kubernetes operations, using hands-on Spring Boot and Docker labs.

DELIVERY

Instructor-led · Live virtual · Blended



WHAT YOU'LL PRACTISE

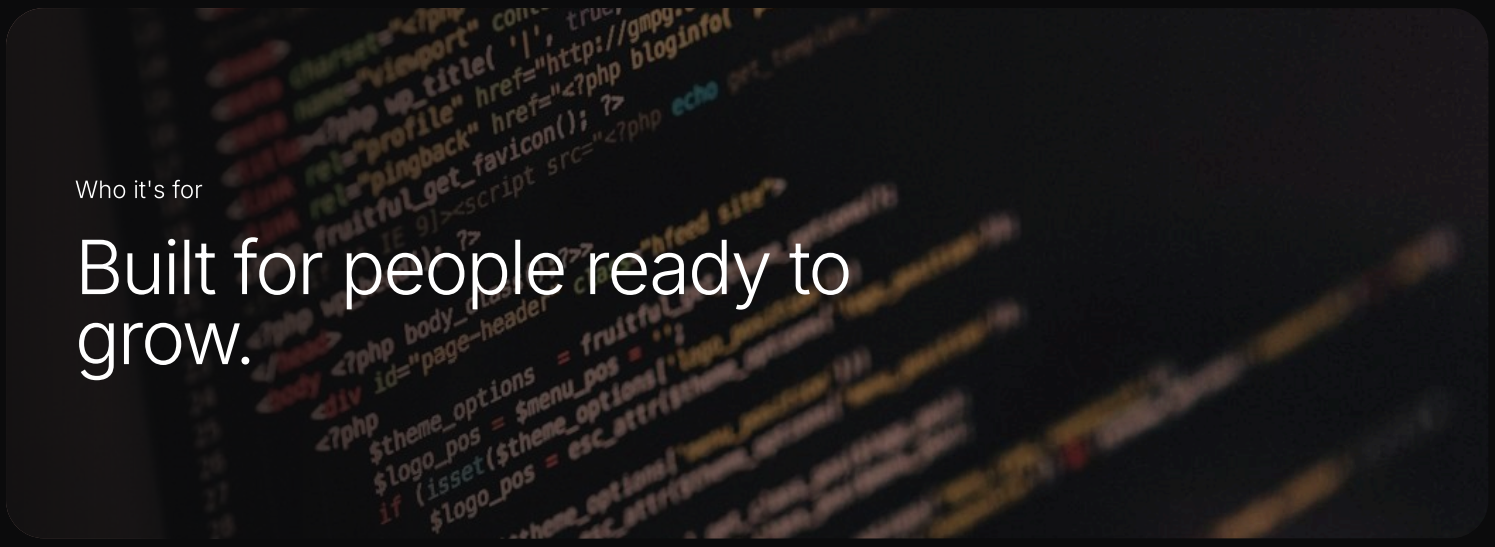
- Define service boundaries
- Use REST, events and messaging
- Manage data across services
- Build resilient, observable services

LEVEL
Advanced

DURATION
72 Hours

CERTIFICATION
SAM-MSA

TRACK
Software Development



Who it's for

Built for people ready to grow.

01

Students & freshers

Python, Java, C/C++ and JavaScript foundations

02

Frontend developers

React, Angular and TypeScript

03

Backend developers

Node, Django, FastAPI and Spring Boot

04

Senior engineers

Microservices, system design, leadership

PREREQUISITES

Experience building backend services in Java, Node.js or Python	01
Understanding of REST APIs and relational databases	02
Basic Docker and Git knowledge	03

ROLES THIS PREPARES YOU FOR

- Backend Engineer ↗
- Microservices Developer ↗
- Software Architect (aspiring) ↗
- Platform Engineer ↗
- DevOps-oriented Developer ↗

Learning outcomes

What you'll be able to do.



01	Decompose domains into services using DDD bounded contexts
02	Choose between REST, gRPC and event-driven communication
03	Implement asynchronous messaging and event streaming with Kafka
04	Manage distributed data using sagas and outbox patterns
05	Build resilient services with timeouts, retries and circuit breakers
06	Deploy and observe services on Kubernetes with OpenTelemetry

Program runway

Module by module.

7
modules · 72 hours

01	**Monoliths to Microservices** Monolith, modular monolith, microservices, Benefits, costs and trade-offs, Conway's law and team topology	8 hrs
02	**Domain-Driven Service Design** Bounded contexts and ubiquitous language, Aggregates and domain events, Event storming workshops	10 hrs
03	**Service Communication Patterns** REST API design and versioning, gRPC and Protocol Buffers, API gateway and BFF patterns	10 hrs
04	**Event-Driven Architecture with Kafka** Messaging versus event streaming, Apache Kafka topics, partitions, consumers, Idempotent consumers	12 hrs
05	**Data Management Across Services** Database per service, Saga orchestration and choreography, Transactional outbox pattern	10 hrs
06	**Resilience and Security** Timeouts, retries and backoff, Circuit breakers with Resilience4j, Bulkheads and rate limiting	8 hrs
07	**Kubernetes, Observability and Delivery** Containerising services with Docker, Kubernetes deployments, services and ingress, Helm charts basics	14 hrs
★	**Capstone · Microservices Ride-Booking Platform** Assessment, capstone review and certification	

The syllabus.

Modules 01–03

Module 01

Monoliths to Microservices

8 hours

Hands-on lab
Assess a sample monolith and propose a migration strategy.

TOPICS COVERED

- Monolith, modular monolith, microservices
- Benefits, costs and trade-offs
- Conway's law and team topology
- Strangler fig migration pattern
- Twelve-factor app principles

Module 02

Domain-Driven Service Design

10 hours

Hands-on lab
Run an event storming session and define service boundaries.

TOPICS COVERED

- Bounded contexts and ubiquitous language
- Aggregates and domain events
- Event storming workshops
- Context mapping patterns
- Defining service contracts

Module 03

Service Communication Patterns

10 hours

Hands-on lab
Build two Spring Boot services communicating via REST and gRPC.

TOPICS COVERED

- REST API design and versioning
- gRPC and Protocol Buffers
- API gateway and BFF patterns
- Service discovery
- Contract testing with Pact

Spring



LEARN BY DOING

Every module ends in a guided hands-on lab — 7 labs across the programme.

Syllabus, continued.

Modules 04–05

Module 04

Event-Driven Architecture with Kafka

12 hours

Hands-on lab
Publish and consume order events between services using Kafka.

TOPICS COVERED

- Messaging versus event streaming
- Apache Kafka topics, partitions, consumers
- Idempotent consumers
- Event schemas and Schema Registry
- Event sourcing and CQRS overview

Apache Kafka

Module 05

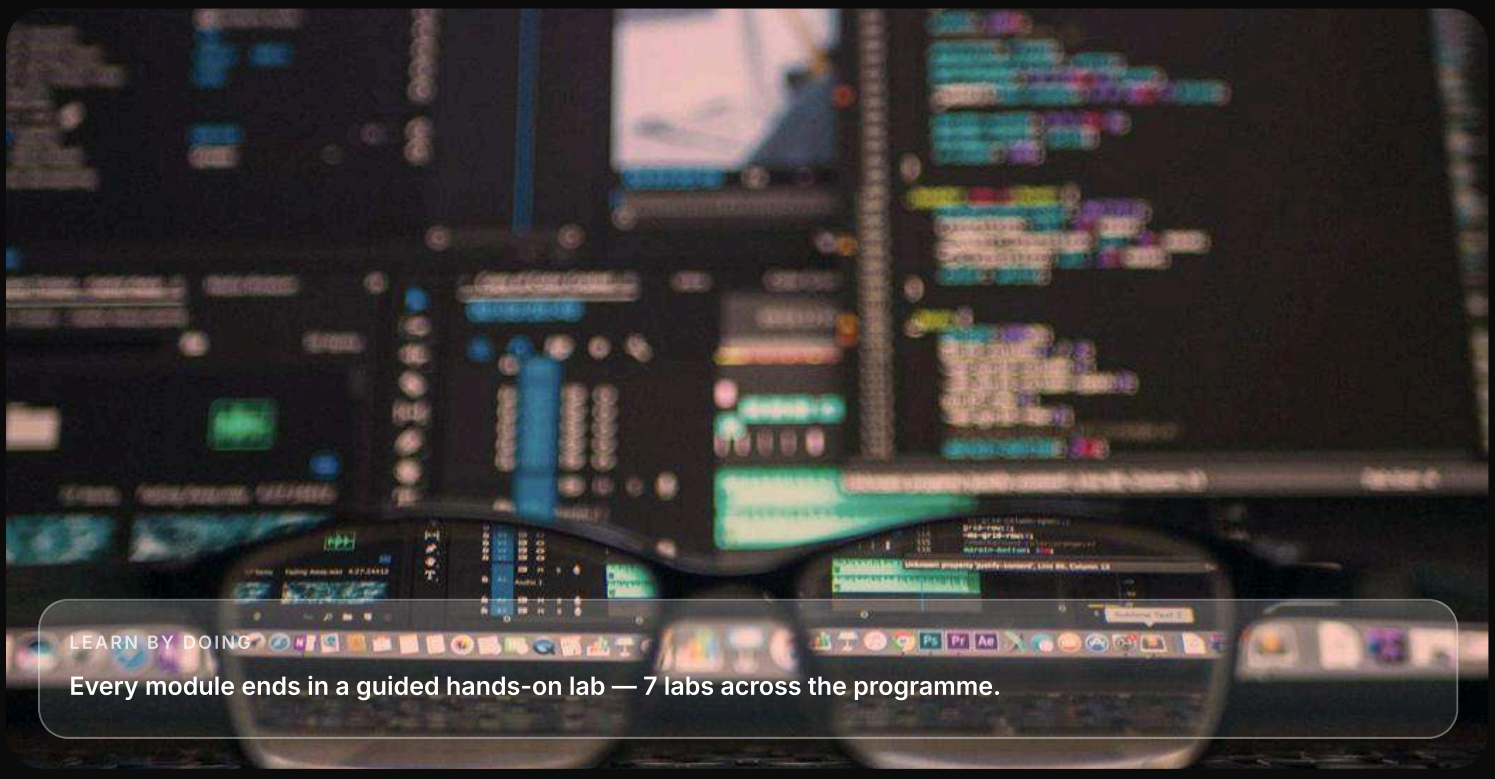
Data Management Across Services

10 hours

Hands-on lab
Implement an order saga with outbox-based event publishing.

TOPICS COVERED

- Database per service
- Saga orchestration and choreography
- Transactional outbox pattern
- Eventual consistency trade-offs
- Change data capture with Debezium



LEARN BY DOING

Every module ends in a guided hands-on lab — 7 labs across the programme.

Syllabus, continued.

Modules 06–07

Module 06

Resilience and Security

8 hours

Hands-on lab
Inject failures and add circuit breakers and retries to services.

TOPICS COVERED

- Timeouts, retries and backoff
- Circuit breakers with Resilience4j
- Bulkheads and rate limiting
- OAuth2 and JWT between services
- mTLS and zero-trust basics

Module 07

Kubernetes, Observability and Delivery

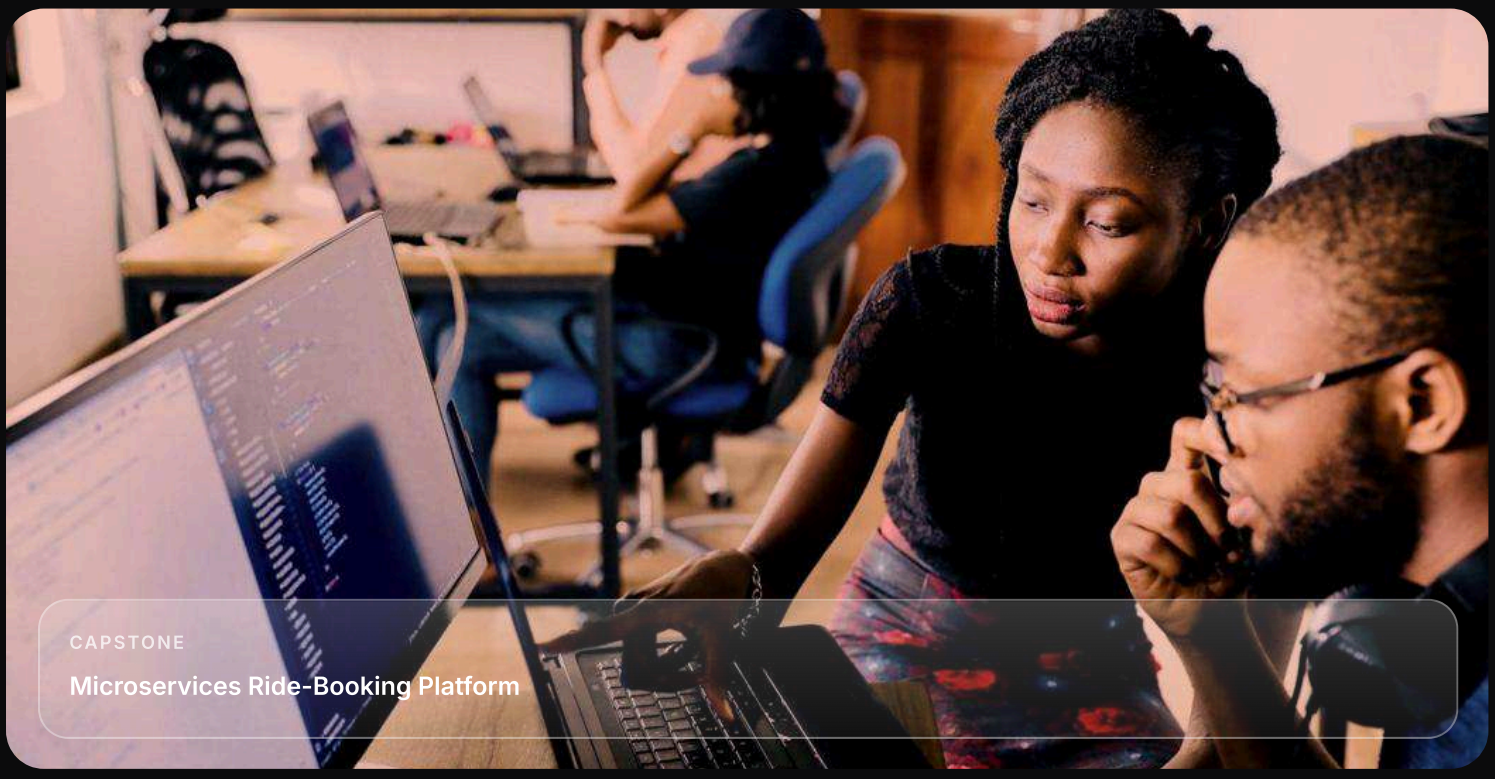
14 hours

Hands-on lab
Deploy services to Kubernetes and trace requests end to end.

TOPICS COVERED

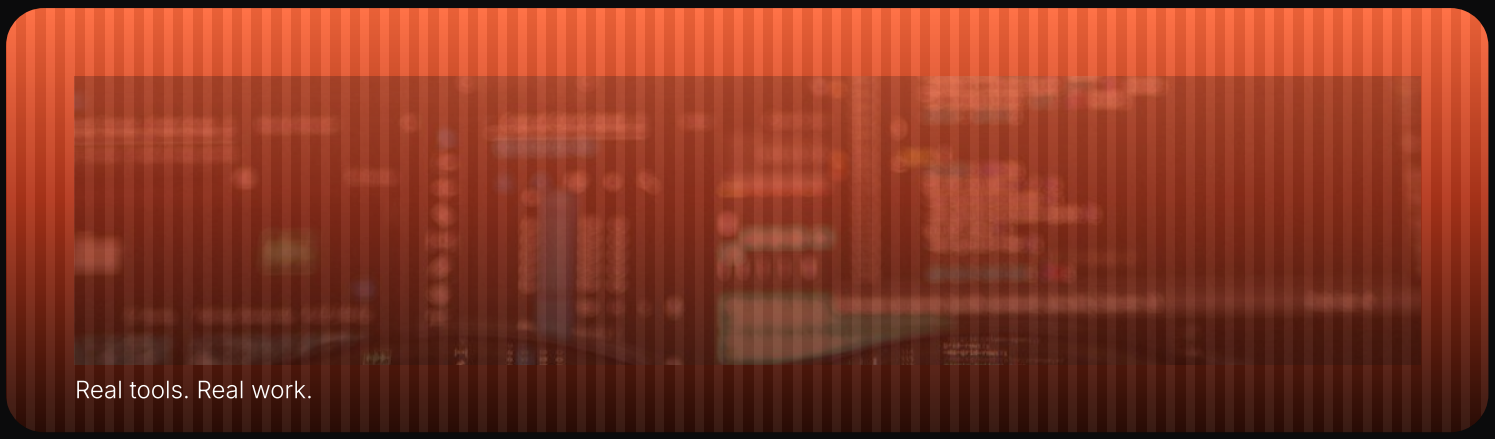
- Containerising services with Docker
- Kubernetes deployments, services and ingress
- Helm charts basics
- Distributed tracing with OpenTelemetry
- Metrics with Prometheus and Grafana
- Per-service CI/CD pipelines
- Final assessment

Docker · Kubernetes · Helm · Prometheus



CAPSTONE

Microservices Ride-Booking Platform



Docker
Containers & images

Kubernetes
Container orchestration

Apache Kafka
Event streaming

Spring
Java enterprise framework

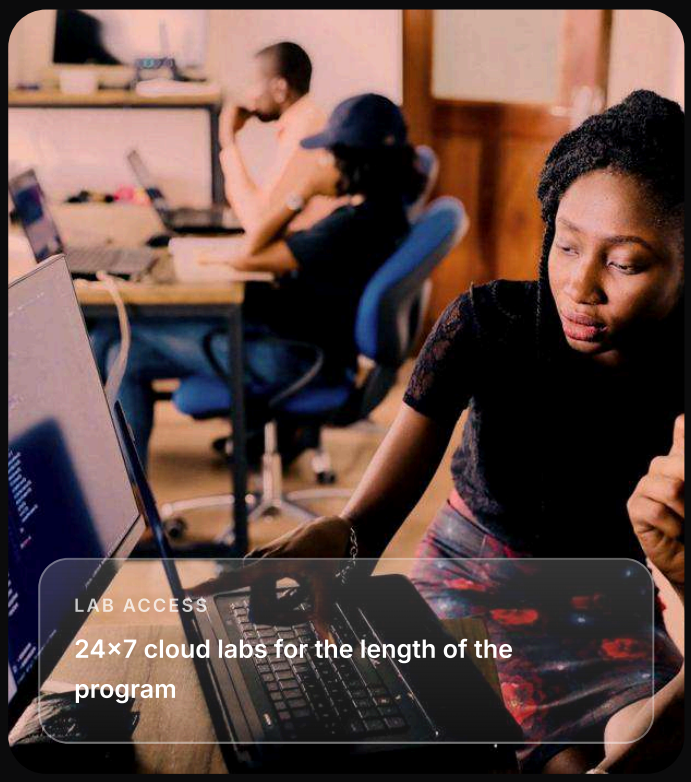
Helm
Kubernetes packaging

Prometheus
Metrics & alerting

Grafana
Observability dashboards

What you will build

Define service boundaries	01
Use REST, events and messaging	02
Manage data across services	03
Build resilient, observable services	04



01

Monolith Decomposition Plan

Analyse a sample monolith, define bounded contexts, and document a phased strangler-fig migration plan with service contracts.

02

Event-Driven Order Pipeline

Order, payment and inventory services coordinated through Kafka events, idempotent consumers and a choreography-based saga.

03

Resilient Service Mesh Lab

Introduce failures into a multi-service system and harden it with retries, circuit breakers, tracing and dashboards.



Capstone project

Microservices Ride-Booking Platform

Design and build a ride-booking system of independent services for riders, drivers, trips and payments, using DDD, REST and Kafka, sagas, resilience patterns, OpenTelemetry tracing and Kubernetes deployment via CI/CD.

Real-world scenario ↗ Mentor review ↗ Portfolio ready ↗

Certification

SAM-MSA

Awarded on successful completion of the program assessments.

- MCQ assessments
- Practical lab evaluation
- Project & capstone review
- Final presentation



The SAM Learning Framework™

Scenario · Apply · Master · Mentor

Phase 01

Scenario

Real-world attack and use-case scenarios open every module.

Phase 02

Apply

Hands-on lab work, CTF challenges and live projects.

Phase 03

Master

Assessment, peer review and certification prep.

Phase 04

Mentor

Post-program career guidance and support.

DELIVERY MODELS

- Instructor-Led (ILT)
- Virtual Instructor-Led
- Self-Paced Online
- E-Learning Licensing
- Blended Learning
- Corporate Onsite
- Bootcamp Intensive

Learning path · Software Development

Where this program fits.

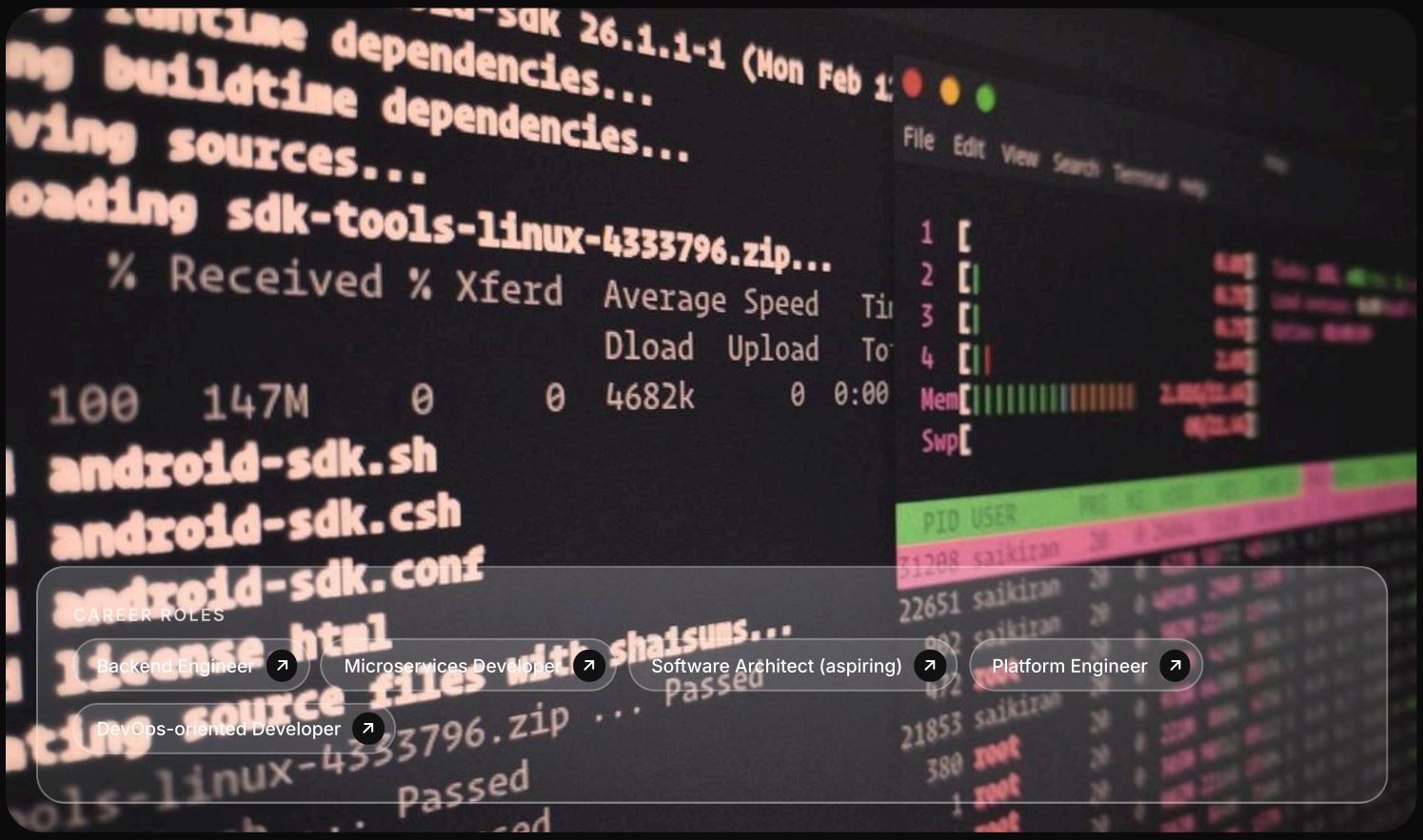
Beginner	Intermediate	Advanced	Expert
4 programs	10 programs	You are here	1 programs

Before this

Python Advanced (OOP, Design Patterns)	56 Hours
TypeScript Mastery	40 Hours
Django Web Framework	56 Hours
FastAPI — Modern Python APIs	48 Hours

Up next

Software Engineering Leadership	48 Hours



Questions, answered.

Q01

Do I need prior experience to join?

Yes. This advanced program expects you to have built backend services, understand REST APIs and databases, and have basic Docker skills. Labs use Spring Boot, but concepts apply equally to Node.js or Python services.

Q02

How are classes scheduled and delivered?

Sessions are instructor-led and run as live virtual or blended batches, with weekday and weekend options where available. Each session combines concept walkthroughs, live coding and guided lab time, with recordings or notes shared for revision.

Q03

Will I receive a certificate?

Yes. Learners who complete the required labs, projects and final assessments receive the Samcommunity Academy SAM-MSA certificate. It is awarded on the basis of assessed work, not attendance alone.

Q04

What tools and lab access will I need?

You need a laptop with at least 16GB RAM, JDK 21, Docker Desktop, kind or minikube, and an IDE. Cloud-based lab environments may be provided where local resources are limited.

Q05

Is career support included?

You get guidance on building a portfolio from your projects, resume and GitHub profile reviews, and mock technical interviews. We help you prepare and present your skills, but we do not guarantee jobs, placements or salary outcomes.



Let's talk

START MICROSERVICES ARCHITECTURE.

academy@samcommunity.in

corporate@samcommunity.in

samcommunity.in

@samcommunity_

Learn practical skills. Grow with mentors.

Xylyn Samcommunity Private Limited · DPIIT-recognized startup · Surat, Gujarat, India