Wholesession

BUILDING PRODUCTION-GRADE MICROSERVICES

COHORT 1 | FEBRUARY 24 - APRIL 16, 2026

TAUGHT BY AKINKUNMI OKUNOLA
SENIOR SOFTWARE ENGINEER
PREVIOUSLY @ INTERSWITCH

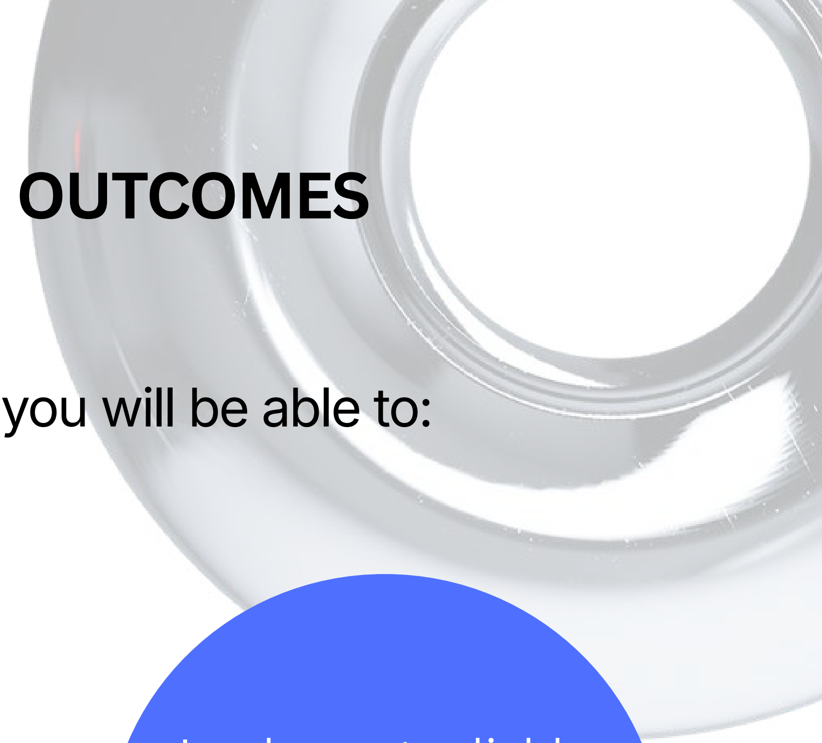
WELCOME TO PRODUCTION MICROSERVICES!

Congratulations on taking this step toward senior-level engineering. Over the next 8 weeks, you'll learn how to design, build, and ship microservices that handle real production traffic. This isn't about toy examples—you'll build systems that scale, fail gracefully, and evolve over time. I've spent 10+ years building fintech systems at Interswitch, my other previous roles, and my current role, serving millions of users.

I've made the mistakes. I've debugged the production fires at 3am. I know what works and what doesn't.

This course is everything I wish someone taught me when I was a mid-level engineer. You're joining 30 ambitious engineers in Cohort 1. Show up. Do the work. Help your peers. You'll be amazed at what you build. Let's go.

— Akinkunmi Okunola



LEARNING OUTCOMES

By the end of this course, you will be able to:

Design and build production-ready microservices using best practices

Implement reliable inter-service communication (REST & messaging)

Deploy, monitor, and operate microservices in real-world environments

Apply Domain-Driven Design (DDD) to real backend systems

Build resilient systems using retries, circuit breakers, and observability

8-WEEK CURRICULUM

WEEK 1:

Introduction to Production Microservices
Feb 24 & 26 | 6:00 PM - 8:00 PM WAT

Understanding the principles behind production microservices and when to use them.

Topics Covered:

→ Monolith vs microservices: trade-offs and when to use each

→ Real-world architectural challenges at scale

→ Understanding bounded contexts and service boundaries

→ Common microservices pitfalls and how to avoid them

→ Case study: Real production architecture patterns

What You'll Build: Identify service boundaries for a sample e-commerce system

Assignment: Design service boundaries for your capstone project idea

WEEK 2:

Service Design & Domain-Driven Design
Mar 3 & 5 | 6:00 PM - 8:00 PM WAT

Designing microservices that work well in production using Domain-Driven Design principles.

Topics Covered:
→ Domain-Driven Design (DDD) fundamentals
→ Identifying and designing bounded contexts
→ Avoiding the distributed monolith trap
→ Service decomposition strategies
→ Designing for independence and autonomy

What You'll Build: Design the domain model and service boundaries for your project

Assignment: Create a service architecture diagram for your capstone

WEEK 3:

API Design & Communication
Mar 10 & 12 | 6:00 PM - 8:00 PM WAT

Learn how to design APIs and reason about distributed communication patterns.

Topics Covered:
→ RESTful API design best practices
→ Synchronous vs asynchronous communication
→ Event-driven architecture patterns
→ Message queues and publish/subscribe
→ API versioning and backwards compatibility

What You'll Build: Design and implement REST APIs for your services

Assignment: Implement API contracts and communication between 2 services

WEEK 4:

Data Management & Consistency
Mar 17 & 19 | 6:00 PM - 8:00 PM WAT

Learn how to manage data and maintain consistency in distributed systems.

Topics Covered:
→ Database per service pattern
→ Distributed transactions and the saga pattern
→ Event sourcing and CQRS (Command Query Responsibility Segregation)
→ Eventual consistency in practice
→ Handling data duplication and synchronization

What You'll Build: Implement data management strategy for your services

Assignment: Design and implement a saga for a distributed transaction

WEEK 5:

Security & Authentication
Mar 24 & 26 | 6:00 PM - 8:00 PM WAT

Learn how to secure microservices and implement authentication and authorization.

Topics Covered:
→ Securing microservices communication
→ OAuth 2.0 and JWT (JSON Web Tokens)
→ API gateway patterns and security
→ Service-to-service authentication
→ Authorization and role-based access control (RBAC)

What You'll Build: Implement authentication and authorization in your system

Assignment: Add JWT-based authentication to your APIs

WEEK 6:

Resilience & Observability
Mar 31 & Apr 2 | 6:00 PM - 8:00 PM WAT

Learn how to make microservices resilient to failures and observable in production.

Topics Covered:
→ Circuit breakers and retry patterns
→ Timeouts and graceful degradation
→ Distributed tracing with OpenTelemetry
→ Logging strategies and centralized logging
→ Metrics and monitoring (Prometheus, Grafana)
→ Health checks and readiness probes

What You'll Build: Add resilience patterns and observability to your services

Assignment: Implement circuit breakers, logging, and metrics

WEEK 7:

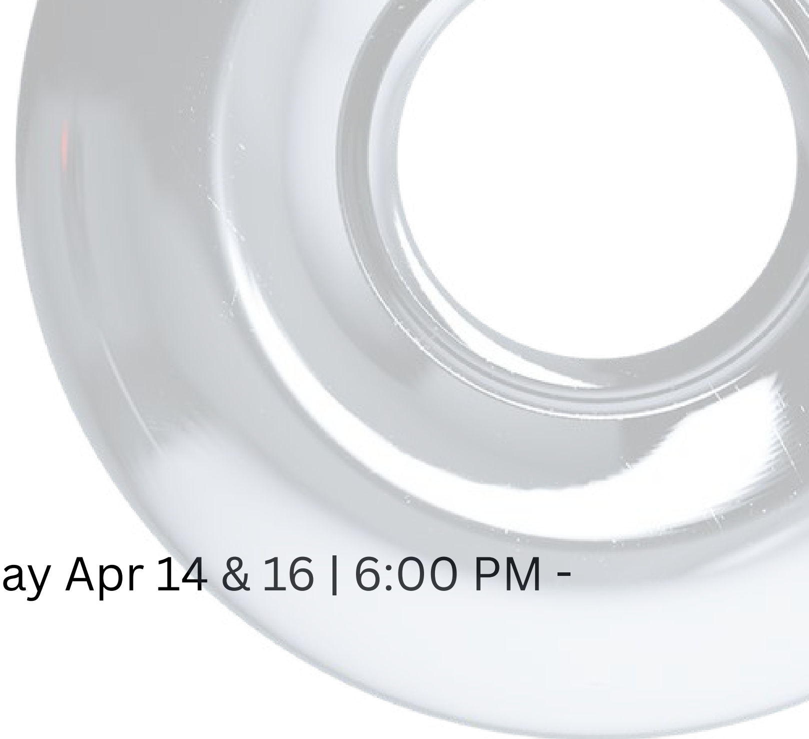
CI/CD & Deployment
Apr 7 & 9 | 6:00 PM - 8:00 PM WAT

Learn how to deploy and operate microservices in production environments.

Topics Covered:
→ Containerization with Docker
→ Container orchestration with Kubernetes basics
→ CI/CD pipelines (GitHub Actions, GitLab CI)
→ Infrastructure as code with Terraform
→ Blue-green and canary deployment strategies
→ Production deployment best practices

What You'll Build: Deploy your microservices to a cloud environment

Assignment: Create a CI/CD pipeline and deploy to AWS/Azure/GCP



WEEK 8:

Capstone Project & Demo Day Apr 14 & 16 | 6:00 PM - 8:00 PM WAT

Final project presentations, code reviews, and production readiness assessment.

Topics Covered:
→ Production readiness checklist
→ Performance testing and optimization
→ Documentation and knowledge sharing
→ Career advice and next steps in your engineering journey

What You'll Present:
Your complete microservices system with:
→ Architecture diagram and design decisions
→ Live demo of your system
→ Code walkthrough and trade-offs explained
→ Production deployment
→ Monitoring and observability setup

Demo Day Deliverables:

→ 10-15 minute presentation
→ Q&A with instructor and guest judges
→ Code review and feedback
→ Production readiness assessment

CAPSTONE PROJECT

You will design and build a complete microservices-based system that demonstrates production-grade architecture.

REQUIREMENTS:

→ Minimum 3 microservices
→ Authentication & authorization
→ Inter-service communication (REST and/or messaging)
→ Database per service
→ Error handling & resilience patterns
→ Monitoring & logging
→ Deployed to cloud (AWS, Azure, or GCP)
→ Documentation and README

EXAMPLE PROJECT IDEAS:

- E-commerce platform (cart, orders, payments, inventory)
- Payment processing system (transactions, wallet, notifications)
- Social media platform (users, posts, comments, notifications) - Booking system (reservations, availability, payments) You choose the domain. You build.

TECH STACK

You'll work with production-grade tools used by real companies:

LANGUAGES (Choose One):
→ Java / Spring Boot
→ C# / .NET Core
→ Node.js
→ Python
→ Go

→ Docker
→ VS Code
→ Postman
→ Redis
→ Relational database
→ Cloud platform
→ GitHub Actions or GitLab CI

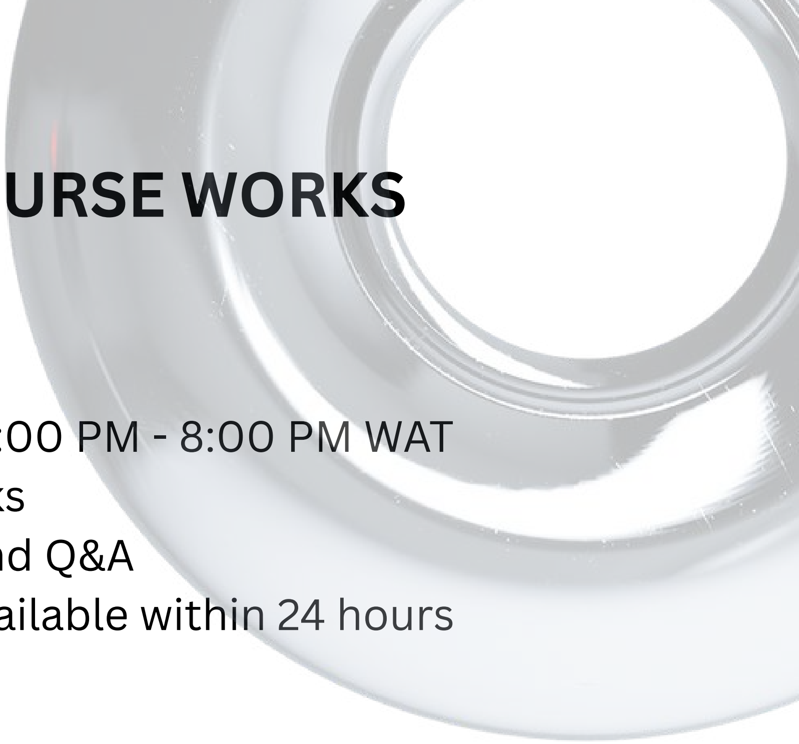
PREREQUISITES

TECHNICAL REQUIREMENTS:
✓ Laptop with minimum 8GB RAM
✓ Stable internet connection
✓ Code editor (VS Code recommended)
✓ Git installed
✓ Docker Desktop installed
✓ Cloud account (AWS/Azure free tier)

KNOWLEDGE PREREQUISITES:
✓ Comfortable with at least one programming language
✓ Built REST APIs before
✓ Basic understanding of databases (SQL or NoSQL)
✓ Familiar with Git and command line

PRE-COURSE READING (Optional but Recommended):
→ "Building Microservices" by Sam Newman (Chapters 1-3)
→ "Domain-Driven Design Distilled" by Vaughn Vernon
→ Martin Fowler's Microservices articles

We'll share these resources in the Discord.



HOW THE COURSE WORKS

LIVE SESSIONS:
→ Every Tuesday & Thursday, 6:00 PM - 8:00 PM WAT
→ 16 sessions total over 8 weeks
→ Mix of lecture, live coding, and Q&A
→ All sessions recorded and available within 24 hours

WEEKLY ASSIGNMENTS:
→ Due before next session
→ Well reviewd
→ Feedback provided within 48 hours

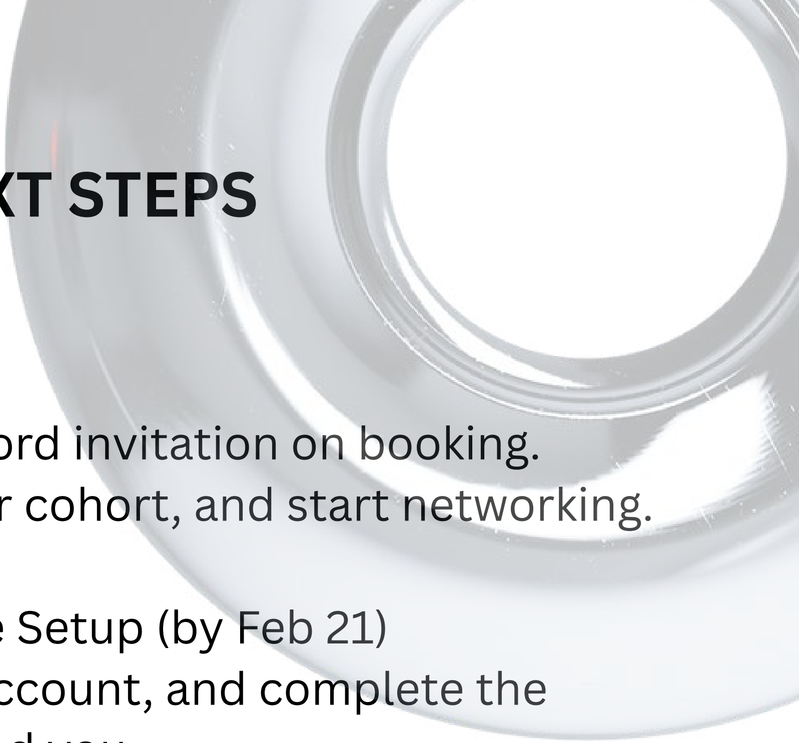
OFFICE HOURS:
→ Weekly optional office hours for questions
→ Time TBA based on cohort availability

DISCORD COMMUNITY:
→ 24/7 access to cohort peers
→ Instructor available for async questions
→ Share resources, debug together, network

TIME COMMITMENT:
→ 4 hours/week: Live sessions (Tue/Thu)
→ 7 hours/week: Assignments and capstone work
→ Total: ~11 hours/week

We don't give formal certificates, but you'll have a production system on your GitHub that proves your skills.

THE NEXT STEPS



STEP 1: Join Discord
You must have received Discord invitation on booking.
Introduce yourself, meet your cohort, and start networking.

STEP 2: Complete Pre-Course Setup (by Feb 21)
Install Docker, set up cloud account, and complete the pre-course checklist we'll send you.

STEP 3: Attend Orientation (Feb 23, 1 day before start)
Optional orientation session to:
→ Meet Akinkunmi and your cohort
→ Review course logistics
→ Set up dev environment
→ Q&A

STEP 4: First Session (Feb 24, 6:00 PM WAT)
Week 1 begins! Come prepared to learn.

IMPORTANT DATES:
→ Feb 23: Orientation session
→ Feb 24: Week 1 starts
→ Apr 16: Demo Day (final presentations)

SUPPORT:
Questions before the course starts?
Email: support@wholesession.com

OUR COMMITMENT

OUR COMMITMENT TO YOU

MONEY-BACK GUARANTEE:
If you attend the first 4 sessions (Week 1 & 2) and feel this course isn't what we promised, email us by end of Week 2 for a full refund. No questions asked.

After Week 2, no refunds are available.

WHAT YOU GET:
✓ 16 live sessions with Akinkunmi
✓ Lifetime access to all session recordings
✓ Weekly assignments with feedback
✓ Capstone project code reviews
✓ Access to private Discord community
✓ Alumni network after graduation

WHAT WE EXPECT FROM YOU:
→ Show up to live sessions
→ Complete weekly assignments
→ Help your peers in Discord
→ Build your capstone project
→ Give honest feedback to improve the course

We succeed when you succeed.

NEED HELP?

NEED HELP?

GENERAL SUPPORT:
Email: support@wholesession.com
Response time: Within 24 hours

TECHNICAL ISSUES:
Discord: #tech-support channel
Response time: Within 12 hours

INSTRUCTOR QUESTIONS:
Discord: Ask in #questions channel
Akinkunmi responds M-F, 6pm - 9pm WAT

COURSE MATERIALS:
All slides, code samples, and resources will be shared in:
→ Discord #resources channel
→ GitHub repository (link shared Week 1)

OFFICE HOURS:
Time TBA (voted by cohort in Discord)

Welcome to Wholesession.
Let's build something real.

WHOLESSION