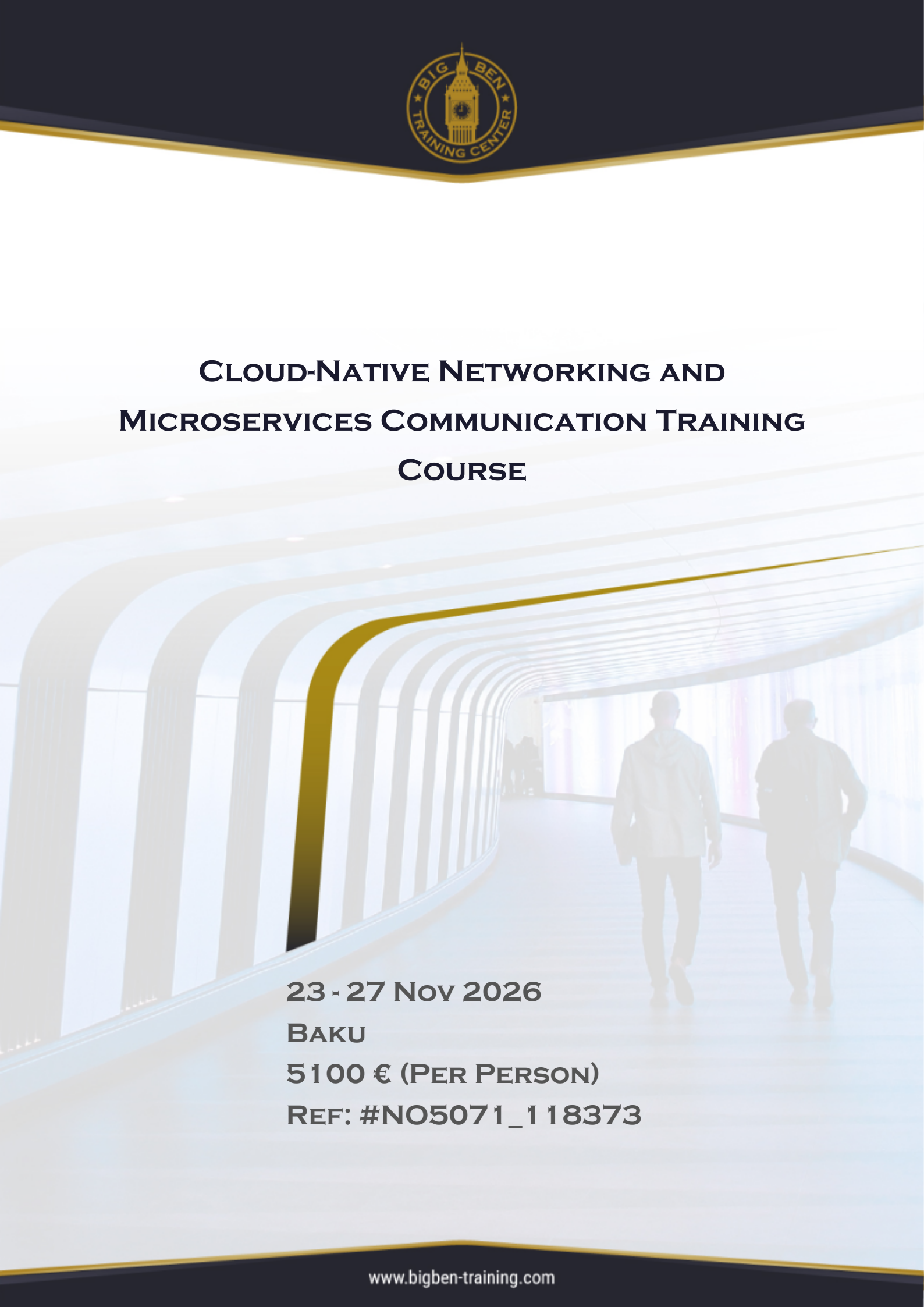
# Cloud-Native Networking and Microservices Communication Training Course



23 - 27 Nov 2026

Baku

5100 € (Per Person)

Ref: #NO5071_118373

## Course Introduction / Overview:

This comprehensive training course provides a deep dive into the critical networking and communication paradigms that underpin modern cloud-native applications. As organizations increasingly migrate from monolithic architectures to distributed microservices, the complexity of inter-service communication, security, and observability presents significant challenges. This program is meticulously designed to equip participants with the knowledge and skills to navigate this complex landscape effectively. Drawing upon foundational principles discussed by experts like Sam Newman in his seminal work, "Building Microservices," the course explores everything from fundamental container networking to advanced service mesh implementations. Participants will gain a robust understanding of how to design, build, and manage resilient, secure, and scalable communication systems for microservices. At BIG BEN Training Center, we focus on providing a holistic view, integrating development, operations, and security concerns to foster a true DevOps and DevSecOps mindset for building next-generation, cloud-native systems. This course is the definitive guide for professionals aiming to master the backbone of modern software architecture.

## Target Audience / This training course is suitable for:

- DevOps Engineers.
- Cloud Architects and Solutions Architects.
- Software Developers and Engineers.
- System and Network Administrators.
- IT Managers and Team Leads.
- Site Reliability Engineers (SREs).
- Cybersecurity Professionals.
- Infrastructure Engineers.

## Target Sectors and Industries:

- Information Technology and Software Services.
- Financial Services and FinTech.
- E-commerce and Retail.
- Healthcare and HealthTech.
- Telecommunications and Media.
- Government and Public Sector Agencies.
- Manufacturing and Industrial IoT.
- Consulting and Professional Services.

## Target Organizations Departments:

- Information Technology (IT) Operations.
- Software Development and Engineering.
- Cloud Engineering and Platform Teams.
- Cybersecurity and Information Security.
- Infrastructure and Network Management.
- Research and Development (R&D).
- Digital Transformation Offices.
- Architecture and Strategy Teams.

## Course Offerings:

By the end of this course, the participants will have able to:

- Design robust and resilient microservice communication patterns.
- Understand and implement various service discovery mechanisms.
- Configure and manage advanced Kubernetes networking concepts like CNI and Ingress.
- Deploy and operate a service mesh for traffic management, observability, and security.
- Secure inter-service communication using mutual TLS and network policies.
- Implement comprehensive monitoring and distributed tracing for microservices.
- Analyze and troubleshoot complex networking issues in a distributed environment.
- Apply event-driven architecture principles for asynchronous communication.
- Evaluate and select appropriate API gateway solutions for specific use cases.
- Integrate security best practices throughout the cloud-native application lifecycle.

## Course Methodology:

The training methodology at BIG BEN Training Center is designed to be highly interactive, engaging, and practical, ensuring participants can immediately apply their learning to real-world challenges. We believe in a blended learning approach that combines expert-led instruction with hands-on application. The course incorporates detailed case studies of successful cloud-native implementations, allowing participants to analyze architectural decisions and outcomes. A significant portion of the training is dedicated to collaborative group exercises and workshops where attendees work in teams to solve complex networking and communication problems, fostering peer-to-peer learning and knowledge sharing. Interactive sessions, including Q&A panels and open discussions, are integrated throughout the five days to encourage active participation and clarify complex topics. Our instructors provide continuous feedback and guidance, ensuring a supportive learning environment. The curriculum emphasizes a problem-solving mindset, moving beyond theoretical knowledge to build tangible skills in designing, securing, and managing modern distributed systems.

## Course Agenda (Course Units):

### Unit One Foundations of Cloud-Native Architectures

- Introduction to cloud-native principles and the 12-Factor App methodology.
- Contrasting monolithic, SOA, and microservices architectures.
- Fundamentals of containerization with Docker.
- Introduction to container orchestration with Kubernetes.
- The role of networking in distributed systems.
- Challenges in microservices communication.
- Overview of the Cloud Native Computing Foundation (CNCF) landscape.

### Unit Two Core Microservices Communication Patterns

- Synchronous communication with REST APIs and best practices.
- High-performance communication with gRPC.
- Asynchronous communication patterns and message brokers.
- Implementing the API Gateway pattern for service aggregation.
- Service discovery strategies (client-side vs. server-side).
- Understanding the Strangler Fig pattern for migration.
- Implementing health checks and readiness probes.

### Unit Three Kubernetes Networking Deep Dive

- The Kubernetes networking model and the Container Network Interface (CNI).
- Understanding Pods, Services, and DNS for service discovery.
- Managing external access with Node Port, Load Balancer, and Ingress controllers.
- Securing cluster traffic with Network Policies.
- Exploring different CNI plugins and their use cases.
- IPv4/IPv6 dual-stack support in Kubernetes.
- Troubleshooting common networking issues in a Kubernetes cluster.

### Unit Four Implementing and Managing a Service Mesh

- Introduction to the service mesh concept (e.g., Istio, Linkerd).
- The role of the control plane and data plane.
- Implementing the sidecar proxy pattern.
- Advanced traffic management: routing, splitting, and mirroring.
- Enhancing observability with distributed tracing, metrics, and logs.
- Securing service-to-service communication with mutual TLS (mTLS).
- Implementing fault tolerance with retries, timeouts, and circuit breakers.

### Unit Five Advanced Observability and Security Topics

- Deep dive into distributed tracing with Jaeger or Zipkin.
- Monitoring and alerting with Prometheus and Grafana.
- Fundamentals of Chaos Engineering for resilience testing.
- Implementing zero-trust security principles in a microservices environment.
- Securing container images and registries.
- Introduction to eBPF for high-performance networking and observability.
- Future trends in cloud-native networking and communication.

## FAQ:

### Qualifications required for registering to this course?

There are no requirements.

### How long is each daily session, and what is the total number of training hours for the course?

This training course spans five days, with daily sessions ranging between 4 to 5 hours, including breaks and interactive activities, bringing the total duration to 20 - 25 training hours.

## Something to think about:

As microservices architectures become more granular and distributed, how does the principle of 'zero trust' networking fundamentally shift the responsibility of security from the perimeter to individual service interactions?

## What unique qualities does this course offer compared to other courses?

This course distinguishes itself by offering a holistic and strategy-oriented perspective on cloud-native networking and microservices communication. While many courses focus narrowly on specific tools, our curriculum emphasizes the underlying architectural principles and patterns that enable effective decision-making, regardless of the technology stack. We bridge the critical gap between software development and network operations, fostering a true DevOps culture by teaching how to design systems that are inherently observable, resilient, and secure from the ground up. The content moves beyond basic implementation to tackle advanced topics like zero-trust security, chaos engineering, and the strategic application of service mesh technology. Participants will not just learn how to configure a tool; they will learn how to architect a robust communication fabric for a complex distributed system. The course's emphasis on real-world case studies and problem-solving workshops ensures that the knowledge gained is practical, applicable, and prepares professionals for the complex challenges they face in modern IT environments.