

Microservices Roadmap_2025

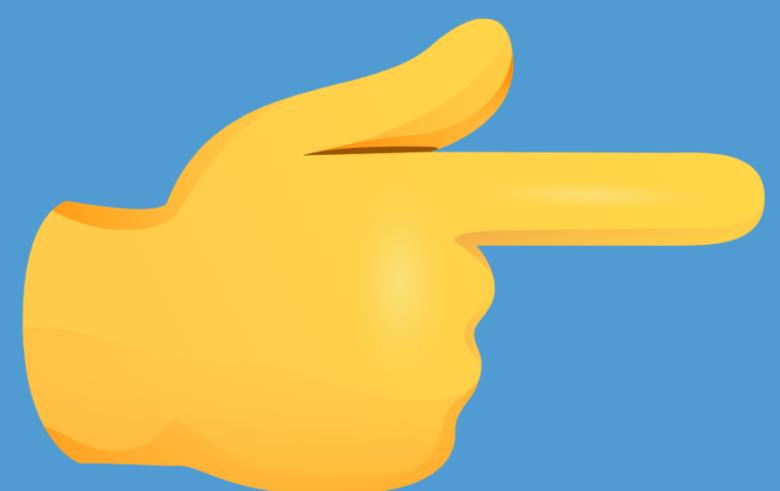
**For Junior, Middle, Senior, Lead developers
and Architects**

This roadmap provides a structured learning path for developers and architects aiming to master microservices architecture.

It is designed to be language-agnostic and is crafted to different experience levels: Junior, Middle, Senior, Lead Developers, and Architects



Anton Martyniuk
antondevtips.com

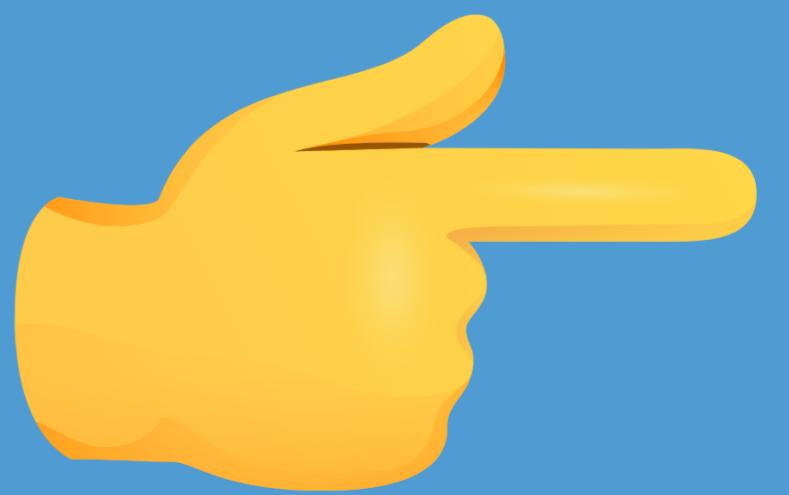


Phase 1: Introduction to Microservices

1. Monolithic vs. Microservices Architecture
 - Understand the differences and when to use each.
2. Benefits and Challenges
 - Learn about scalability, flexibility, complexity, and operational overhead.
3. When to Use Microservices
 - Identify scenarios where microservices are advantageous.



Anton Martyniuk
antondevtips.com



Phase 2: Communication Patterns

1. RESTful APIs

- HTTP Methods: GET, POST, PUT, DELETE, PATCH.
- Status Codes
- API Design Principles: Learn about resource naming, versioning, and HATEOAS.

2. gRPC

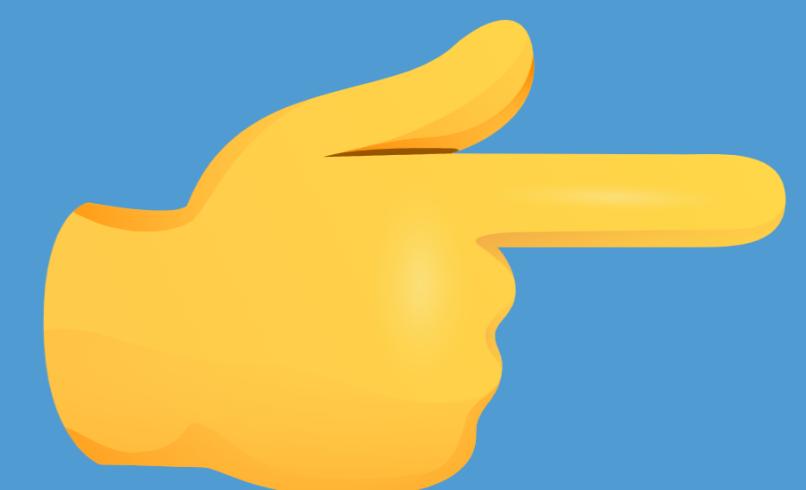
- gRPC communication fundamentals
- Contracts and proto file
- Bidirectional communication

3. GraphQL:

- Queries
- Mutations
- Subscriptions
- Schema Gateways



Anton Martyniuk
antondevtips.com



Phase 3: Data Management Basics

1. Understand data encapsulation in microservices

- Database per Service
- Database schema per Service

2. Database types

- SQL
- NoSQL

3. Data Consistency Basics

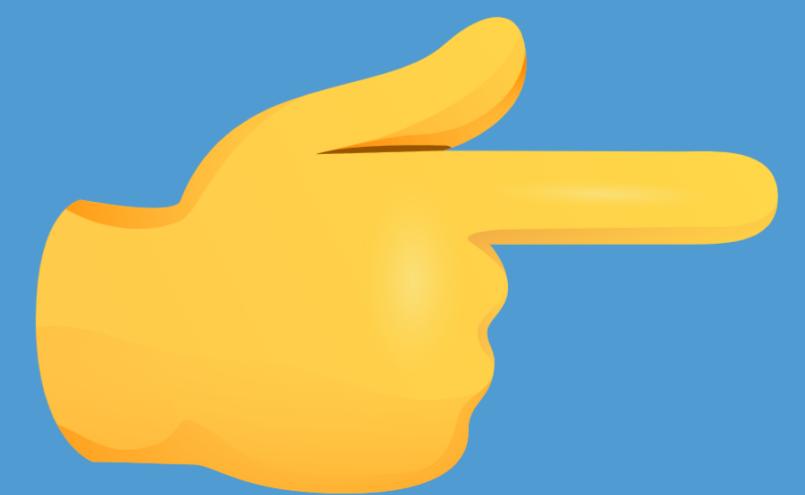
- Learn about eventual consistency vs. strong consistency.

4. Caching

- Implementing distributed caching with tools like Redis



Anton Martyniuk
antondevtips.com



Phase 4: DevOps Basics

1. Continuous Integration/Continuous Deployment (CI/CD)

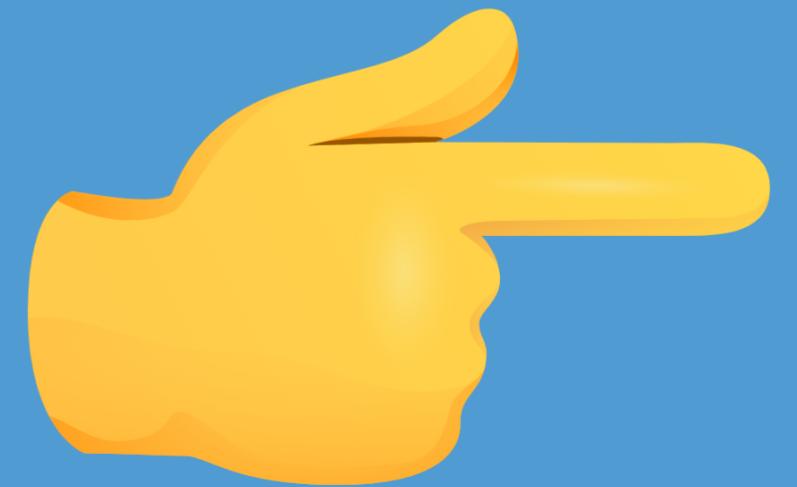
- Introduction to CI/CD pipelines.

2. Containerization

- Basics of Docker and containerization concepts.
- Docker Compose basics



Anton Martyniuk
antondevtips.com



Phase 5: Communication Between Services

1. REST vs. gRPC

- When to use REST or gRPC.

2. Asynchronous Communication

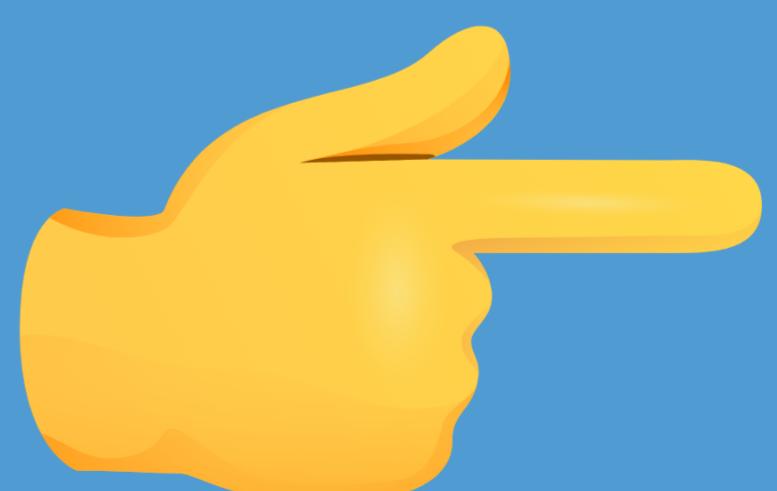
- Introduction to messaging systems.
- Message queues and message brokers
- Introduction to RabbitMQ, Apache Kafka

3. Event-Driven Architecture

- Understanding events, producers, and consumers
- Point to Point and Publish/Subscribe



Anton Martyniuk
antondevtips.com



Level 2: Intermediate Concepts

Phase 6: Data Management Strategies

1. Event Sourcing

- *Storing state changes as a sequence of events.*

2. Command Query Responsibility Segregation (CQRS)

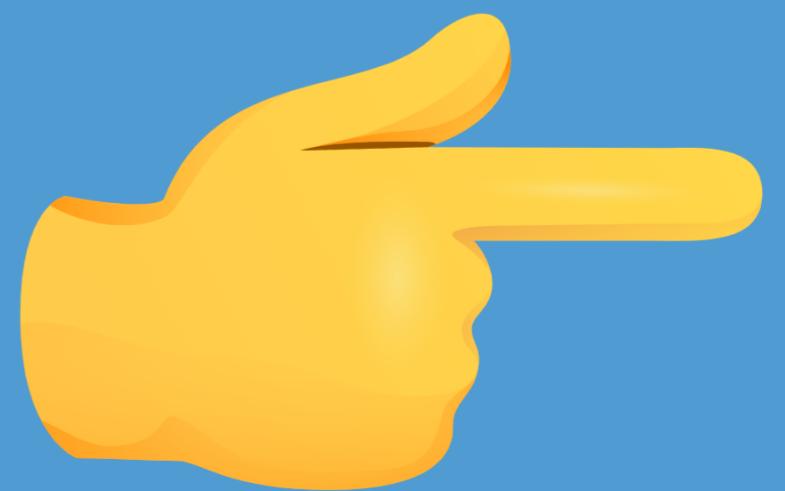
- *Separating read and write models.*

3. Handling Data Consistency

- *Managing data consistency with database transactions inside each service*
- *Implementing eventual consistency across services*



Anton Martyniuk
antondevtips.com



Level 2: Intermediate Concepts

Phase 7: Testing Microservices

1. Unit Testing

- **Testing individual components.**

2. Integration Testing

- **Testing each service integrations with external services**

3. Functional or Contract Testing

- **Ensuring services adhere to agreed contracts.**

4. End to End Testing

- **Ensuring that real services work together as a whole**



Anton Martyniuk
antondevtips.com



Phase 8: Designing Microservices

1. API Versioning

- Strategies for evolving APIs without breaking clients.

2. API Gateway

- Centralized entry point for microservices (e.g., Traefik, Envoy, YARP, Ocelot).

3. Service Registry and Discovery

- Dynamic discovery of services (e.g., Consul, Eureka).

4. Saga Pattern

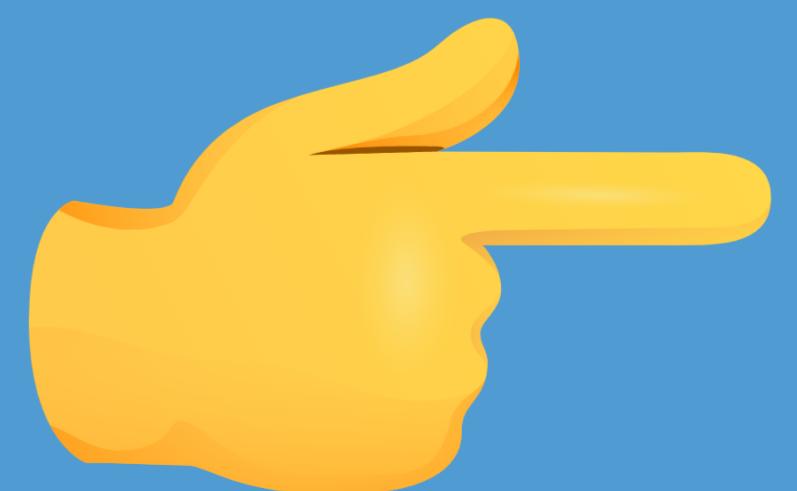
- Managing distributed transactions.

5. Sidecar Pattern

- Enhancing service capabilities without code changes.



Anton Martyniuk
antondevtips.com

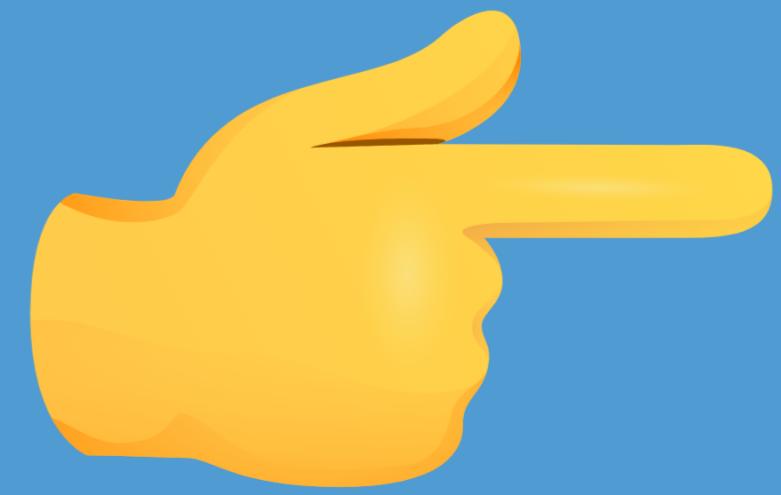


Phase 9: Observability

1. Centralized Logging
 - Tools like Seq, Loki, ELK Stack
2. Distributed Tracing
 - Tools like Jaeger, Seq.
3. Monitoring and Metrics
 - Implementing Prometheus, Grafana.
4. Alerting
5. Setting up proactive notifications.



Anton Martyniuk
antondevtips.com



Phase 10: Scalability and Resilience

1. Load Balancing

- Distributing traffic effectively.

2. Rate Limiting

1. Limiting traefik for certain clients based on some conditions

3. Fault Tolerance

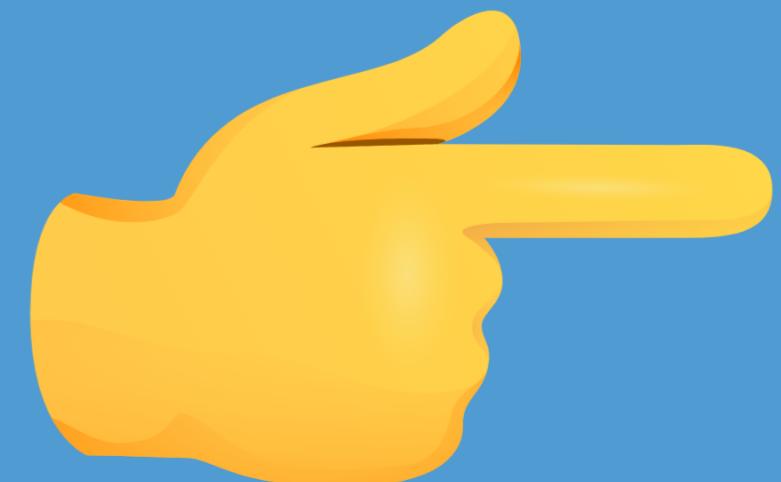
- Designing systems to handle failures gracefully.

4. Auto-Scaling

- Scaling services based on demand.



Anton Martyniuk
antondevtips.com



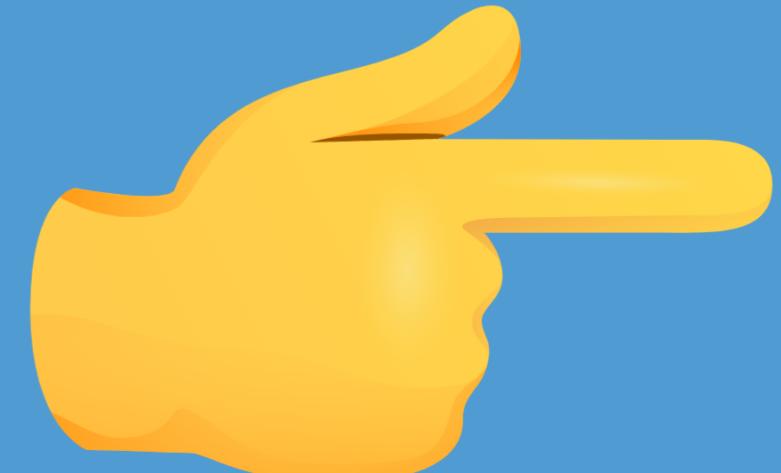
Level 3: Advanced Concepts

Phase 11: Security in Microservices

1. OAuth 2.0 and OpenID Connect
 - Secure authentication protocols.
2. JSON Web Tokens (JWT)
 - Implementing stateless authentication.
3. mTLS (Mutual TLS)
 - Securing communication.
4. Security Best Practices
 - Secure coding standards.
5. Database security
 - TLS
 - Transparent data encryption
 - Always Encrypted



Anton Martyniuk
antondevtips.com



Phase 12: Deployment Strategies

1. Introduction to Kubernetes

- Deploying and managing containers at scale.
- Deployments, services, ingress controllers.

2. Cloud Providers

- Overview of AWS, Azure, GCP services for microservices.

3. Canary and Blue-Green Deployments

- Strategies for zero-downtime deployments.

4. Multi-region deployments.



Anton Martyniuk
antondevtips.com



Phase 13: Performance and Cost Optimization

1. Profiling Microservices

- Identifying performance bottlenecks.

2. Cost Management

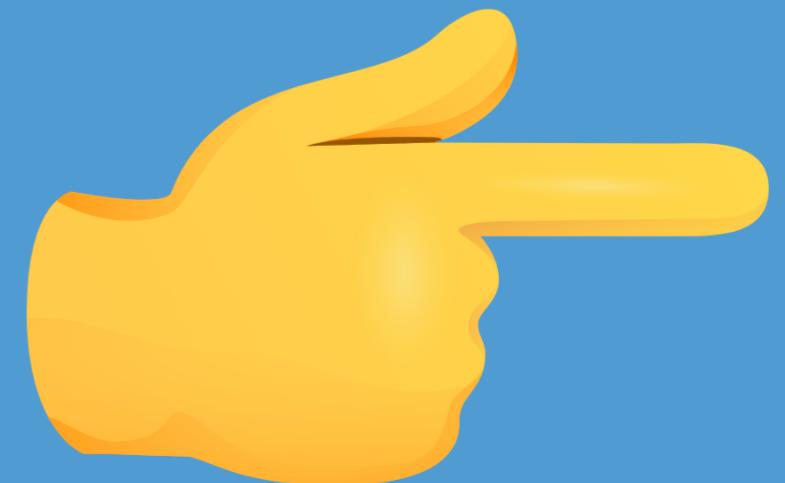
- Optimizing cloud expenditures.

3. Latency Reduction

- Techniques for improving response times.



Anton Martyniuk
antondevtips.com





THANKS FO READING

LETS CONNECT ON LINKED-IN

REPOST TO HELP ME SHARE
USEFUL CONTENT FOR OTHERS

SUBSCRIBE TO ANTONDEVTIPS
TO GET MORE .NET TIPS



Anton Martyniuk
antondevtips.com

