Microservices & DevOps

Understanding Microservices: Building Better Software, Faster

Ivan Malenko, Full-stack developer

What is DevOps? Why Does It Matter?

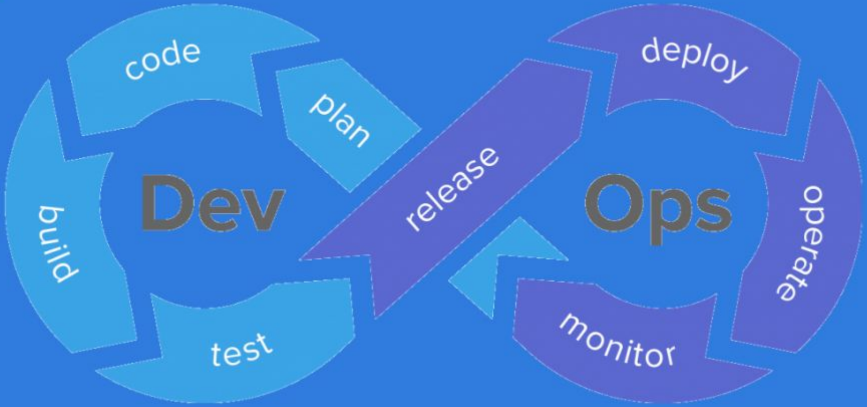
DevOps is a set of practices that automates and integrates the processes between software development and IT operations.

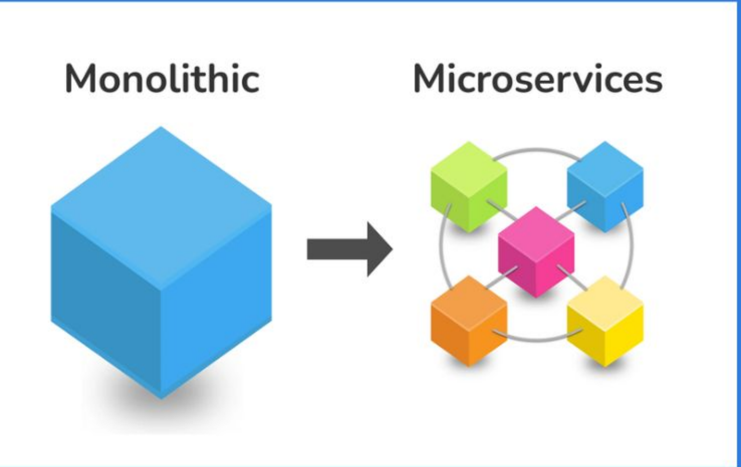
Business Value:

- Faster delivery of features
- Reduced downtime and errors
- Increased customer satisfaction

Bridge Between:

- Developers (who create)
- Operators (who run and support)





Introduction to Microservices

Microservices are an architectural style that structures an application as a collection of loosely coupled services.

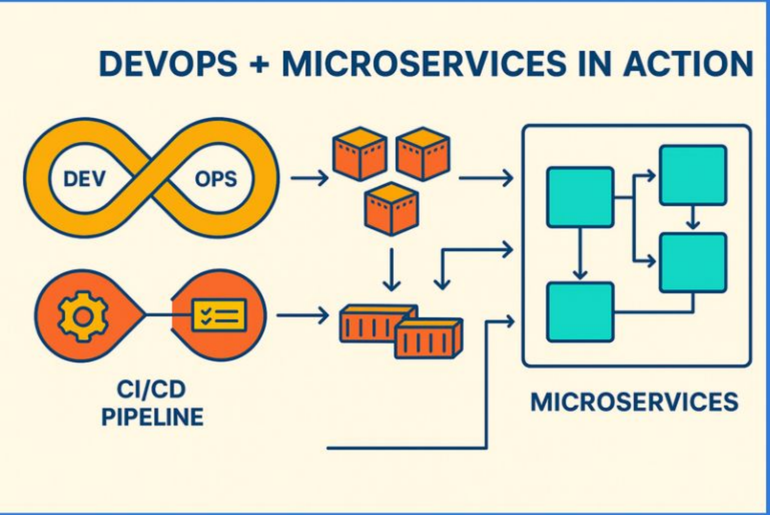
Why Microservices?

- Scalability: Scale individual components, not the entire app
- Flexibility: Deploy independently
- Resilience: Failures isolated to single services
- Faster time-to-market

DevOps + Microservices = Agile, scalable, and fault-tolerant systems

Key DevOps Practices Empowering Microservices

Practice	Role in Pipeline	Efficiency Gains
CI/CD (Continuous Integration/Delivery)	Automates testing and deployment of microservices	Faster releases, fewer integration errors
Infrastructure as Code (IaC)	Automates environment setup using code (e.g., Terraform)	Consistent, repeatable infrastructure
Monitoring & Logging	Real-time insights into microservice performance	Quick issue resolution and continuous feedback
Containerization (Docker + Kubernetes)	Packages microservices for consistent execution	Scalable, portable, isolated deployments



DevOps + Microservices in Action

- Before: Monolithic app, long release cycles, single point of failure
- After:
  - Split into checkout, user auth, and inventory microservices
  - Used Jenkins for CI/CD, Docker for containers, Terraform for IaC
  - Kubernetes for orchestration, Prometheus for monitoring

Result:

- 3x faster deployments
- 40% fewer production incidents
- Instant rollback and zero downtime updates

Adopting DevOps Practices in an Existing Pipeline

Culture and Collaboration

- Promote cross-functional ownership
- Shift-left testing and security

Introduce Incremental Changes

- Begin with CI/CD for one microservice
- Use Docker to standardize dev/prod parity

Automate Infrastructure

- Migrate environments to IaC
- Adopt orchestration tools (e.g., Kubernetes)

Invest in Observability

- Set up centralized logging and alerting
- Use dashboards for KPIs (latency, error rate, etc.)

DevOps + Microservices

✅ Microservices enable agility, DevOps enables velocity

✅ Together, they empower businesses to innovate safely

✅ Start small, scale smart — one service, one practice at a time

✅ Business benefits: Faster releases, happier users, reduced costs

Thank You!

Let’s innovate together today!