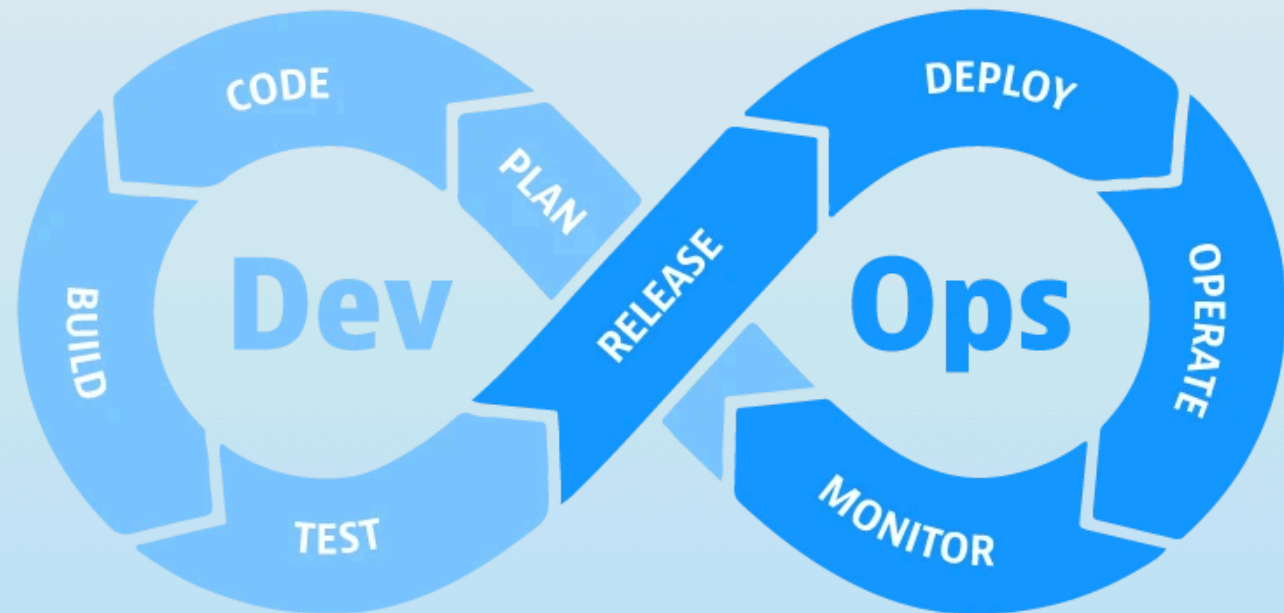


DevOps & Automation



Transforming Modern Software Development

Present by: Lei Chen - Jan, 2025



Introduction



- **Context:** Modern software development cycles are rapid and complex.
- **DevOps:** A cultural and technical approach that unites development and operations (Kim et al., 2016).
- **Objective:** Show how DevOps principles and automation increase **efficiency, reliability, and scalability**.

DevOps Principles



Culture of Collaboration

- Breaks silos between Dev, Ops, QA, and business teams.
- Shared ownership of quality and uptime fosters continuous improvement (Kim et al., 2016).

Continuous Integration & Delivery

- Automated pipelines for builds, tests, and deployments.
- Frequent, small releases that reduce risk and enable fast feedback (Humble & Farley, 2010).

DevOps Principles (Continued)



Infrastructure as Code (IaC)

- Version-controlled, repeatable environment setup (Terraform, Ansible).
- Consistency and elimination of “drift” across environments (HashiCorp, n.d.).

Monitoring & Feedback

- Automated alerts and dashboards catch issues early.
- Data-driven insights guide performance and security enhancements.

Role of Automation in DevOps



- **Streamlined Processes**

- Automated testing (JUnit, Selenium)
- CI/CD tools (Jenkins, GitLab CI)
- Configuration management (Chef, Puppet)

- **Reliability & Scalability**

- Reduced human error via scripted provisioning
- Seamless resource scaling for changing demands

More on Automation



- **Cost & Time Efficiency**

- Fewer manual, repetitive tasks → Faster releases
- Quicker rollbacks reduce downtime

- **Key Metrics**

- Teams practicing DevOps often see >50% faster lead time and improved deployment frequency (Kim et al., 2016).
- Enhanced visibility into systems reduces mean time to recovery (MTTR).

Manual vs. Automated System Administration



Aspect	Manual Tasks	Automated Tasks
Efficiency	Repetitive, slow, error-prone	Scripts handle tasks quickly and consistently
Reliability	Human-dependent; prone to inconsistencies	Automated workflows ensure uniform, predictable execution
Scalability	Requires significant manual intervention when load spikes	On-demand provisioning with IaC or container orchestration
Maintenance & Overhead	Periodic manual checks, updates, and patches	Centralized scripts with version control for easy updates and rollbacks
Learning Curve	Straightforward but time-consuming in large-scale systems	Requires time to learn tools, but yields major efficiency and reliability advantages

Summary & Key Takeaways



- **DevOps** bridges culture and tooling to boost collaboration and software delivery.
- **Automation** underpins reliability, speed, and scalability.
- **CI/CD, IaC, and Monitoring** are pillars of effective DevOps adoption.
- Result: **Faster releases, reduced risks**, and a **continuous innovation cycle**.

Thank You

Appreciate your time and attention!



References



1. Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley.
2. Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press.
3. HashiCorp. (n.d.). *Terraform Documentation*.
<https://developer.hashicorp.com/terraform/docs>
4. Icons8. (n.d.). Icons8. <https://icons8.com/icons>