

Azure DevOps

- Bavatharani S

Introduction to DevOps

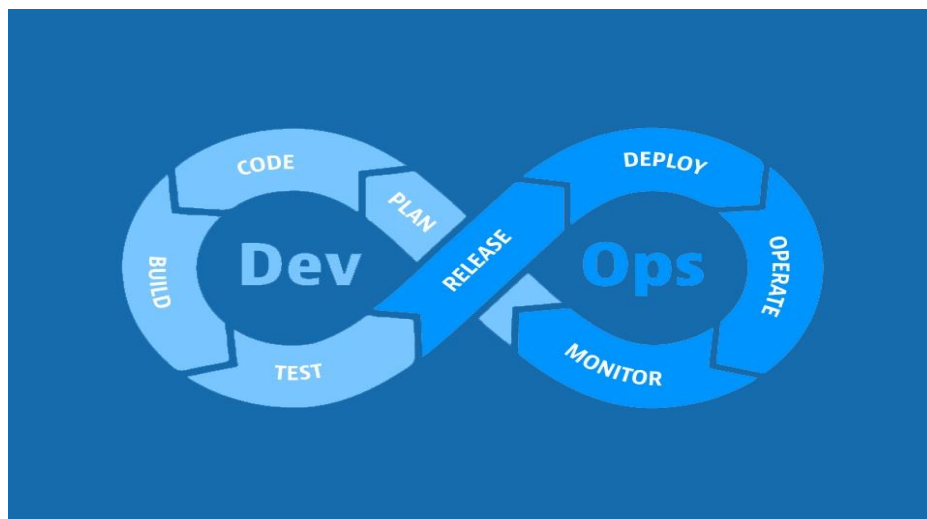
DevOps is a cultural movement and professional practice that bridges the gap between software development (Dev) and IT operations (Ops). The primary focus of DevOps is to improve collaboration, automate repetitive tasks, and enable faster and more reliable delivery of software.

The DevOps approach emphasizes shared responsibility across the software lifecycle. Instead of developers writing code and “throwing it over the wall” to operations, both teams collaborate from the start, ensuring applications are developed with deployment, scalability, and reliability in mind.

What is DevOps?

- DevOps = Development + Operations.
- It is not just a set of tools, but a mindset and methodology.
- Promotes automation, continuous delivery, monitoring, and feedback.
- Focuses on breaking down silos, improving communication, and accelerating delivery.

Example: In traditional IT, code might take weeks to move from development to production. With DevOps practices, the same code can be tested, integrated, and deployed within hours.



Why DevOps?

Traditional software delivery was:

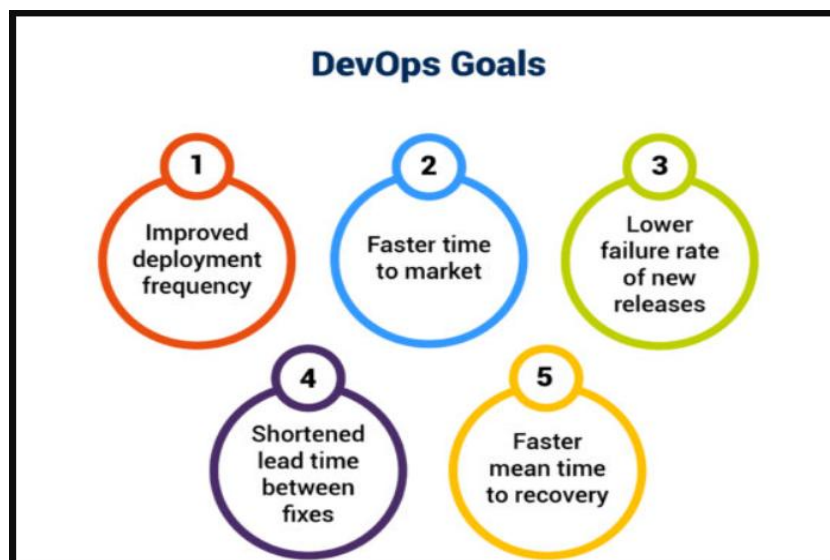
- Slow – Releases happened quarterly or annually.
- Risky – Manual deployments caused frequent outages.
- Siloed – Dev, QA, and Ops worked separately.

DevOps solves these problems by:

- Increasing release frequency.
- Automating builds, tests, and deployments.
- Enabling faster feedback loops.
- Improving customer satisfaction with quick feature delivery.
- Ensuring systems are resilient and scalable.

Goals of DevOps

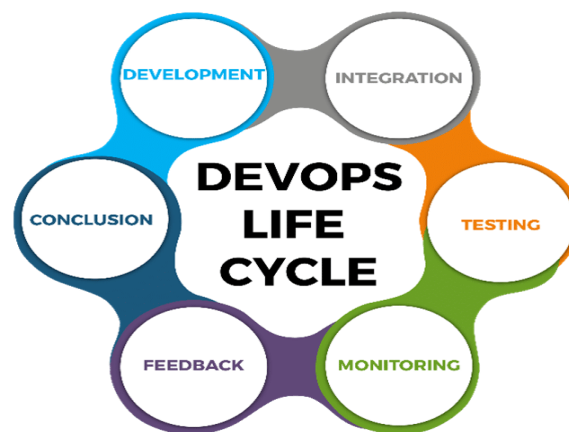
1. Speed: Deliver features faster without sacrificing quality.
2. Automation: Reduce manual intervention to minimize errors.
3. Quality: Ensure every release is stable through continuous testing.
4. Collaboration: Unite development, operations, QA, and business teams.
5. Monitoring & Feedback: Continuously track performance and use customer feedback for improvement.



DevOps Lifecycle

The DevOps lifecycle is often visualized as an infinity loop showing continuous collaboration.

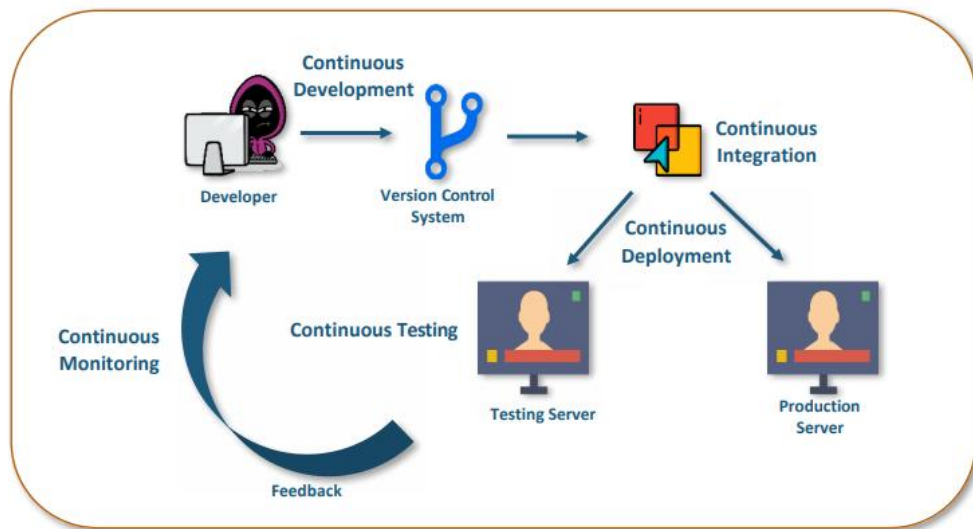
1. Plan: Teams define product requirements, backlog, and architecture.
2. Develop: Developers write, review, and merge code using version control.
3. Build & Integrate: Code is compiled and integrated into a shared repository with automated builds.
4. Test: Automated and manual tests ensure software quality.
5. Deploy: Applications are released to staging and production with automated pipelines.
6. Operate: Systems are managed for uptime, performance, and scalability.
7. Monitor & Feedback: Logs, alerts, and monitoring tools provide feedback for improvements.



Key Concepts & Terminology

- CI (Continuous Integration): Developers frequently integrate code into a central repository; automated builds and tests ensure stability.
- CD (Continuous Deployment/Delivery): Automatically releasing tested code to production, reducing delays.
- IaC (Infrastructure as Code): Infrastructure is managed with code (Terraform, Ansible) instead of manual processes.
- Pipeline: A structured workflow that defines stages from code commit to production deployment.

- Automation: Use of tools to minimize manual work in testing, deployments, monitoring, and scaling.



DevOps Perspective

DevOps is not just about tools but about cultural transformation:

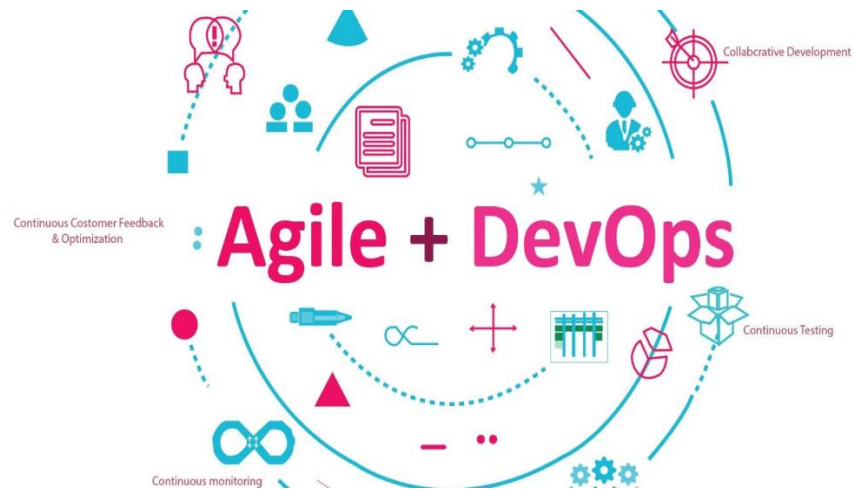
- Developers are responsible for more than just coding — they think about performance, security, and operations.
- Operations engineers are involved earlier in the development cycle to influence architecture and scalability.
- Customers are central — feedback is continuously integrated.
- The culture promotes shared ownership, transparency, and accountability.



DevOps and Agile

- Agile focuses on iterative development and rapid feature delivery.
- DevOps extends Agile by ensuring the delivered features are deployed, monitored, and maintained effectively.

- Together, Agile + DevOps = Fast Development + Reliable Operations.
- Example: Agile teams deliver features every sprint, while DevOps ensures those features reach production smoothly with CI/CD pipelines.



Popular DevOps Tools

- **Version Control:** Git, GitHub, GitLab, Bitbucket.
- **CI/CD Pipelines:** Jenkins, GitHub Actions, GitLab CI, CircleCI.
- **Configuration Management:** Ansible, Puppet, Chef, SaltStack.
- **Containers & Orchestration:** Docker, Kubernetes, OpenShift.
- **Monitoring & Logging:** Prometheus, Grafana, ELK Stack, Splunk.
- **Cloud Platforms:** AWS, Microsoft Azure, Google Cloud Platform.



Configuration Management in DevOps

Configuration management ensures all environments (dev, test, prod) remain consistent and reproducible.

- Automates server setup, patching, and updates.
- Ensures quick rollback in case of failures.
- Provides version control for infrastructure.
- Tools like Ansible, Puppet, and Chef automate repetitive system administration tasks, improving reliability.

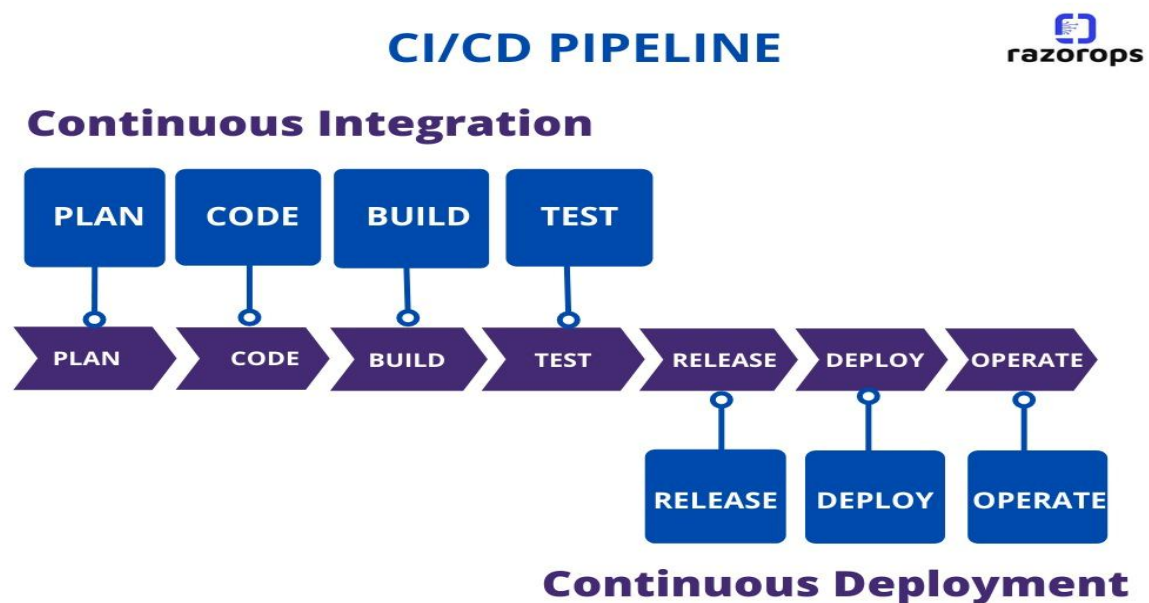
Continuous Integration & Continuous Deployment (CI/CD)

- **Continuous Integration (CI):** Developers frequently push code → automated builds and tests run immediately. This helps detect integration issues early.
- **Continuous Deployment (CD):** Code that passes CI is automatically deployed to production without manual intervention.

Benefits of CI/CD:

- Faster software delivery.
- Reduced integration problems.
- Quick bug detection and resolution.
- Higher quality and customer satisfaction.

Example: An e-commerce company can release new features (discounts, payment options) daily instead of monthly.



Conclusion

DevOps is more than just a set of tools — it is a cultural shift and methodology that transforms how software is built, delivered, and maintained. By fostering collaboration between development and operations, DevOps breaks down silos and ensures faster, more reliable, and higher-quality software releases.

With practices like CI/CD, Infrastructure as Code, automation, and continuous monitoring, organizations can achieve shorter release cycles, reduce failures, and respond quickly to customer needs. The integration of Agile and DevOps practices further strengthens adaptability, allowing teams to innovate and deliver value continuously.

Ultimately, DevOps enables businesses to remain competitive in today's fast-paced digital world by combining speed, reliability, and scalability — ensuring that technology truly delivers on business goals.