21.8.25 – Introduction to DevOps

Introduction to DevOps

DevOps is a cultural and professional movement that emphasizes collaboration between software developers (Dev) and IT operations (Ops).
It aims to shorten the software development lifecycle, deliver high-quality applications, and improve customer satisfaction.

What is DevOps?

- DevOps = Development + Operations.
- It is not just a tool or process but a philosophy that integrates people, processes, and technology.
- It ensures faster, reliable, and automated software delivery.
- Focus areas: automation, collaboration, continuous delivery, monitoring.

DevOps Lifecycle

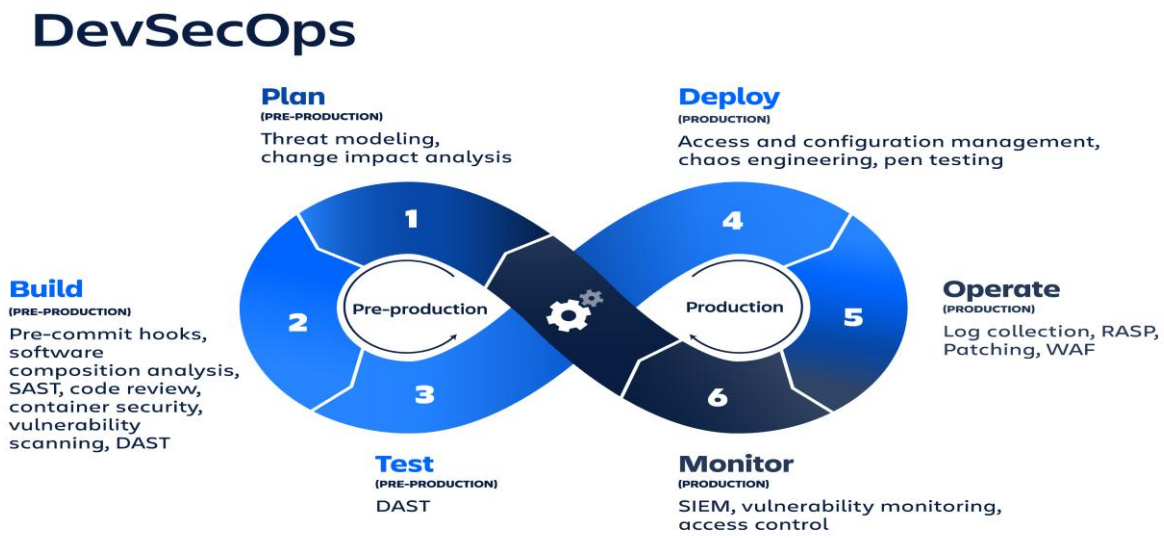

Figure 1: The DevOps lifecycle shown as an infinity loop.

Why DevOps?

Traditional software delivery was slow, error-prone, and siloed. DevOps solves these challenges by:

- Increasing deployment frequency.
- Reducing time to market.
- Improving collaboration.
- Ensuring reliable releases.
- Enhancing customer satisfaction.

DevOps Goals

- Faster delivery with quality.
- Automation across processes.
- Continuous monitoring and feedback.
- Breaking silos between teams.
- Ensuring business value delivery.

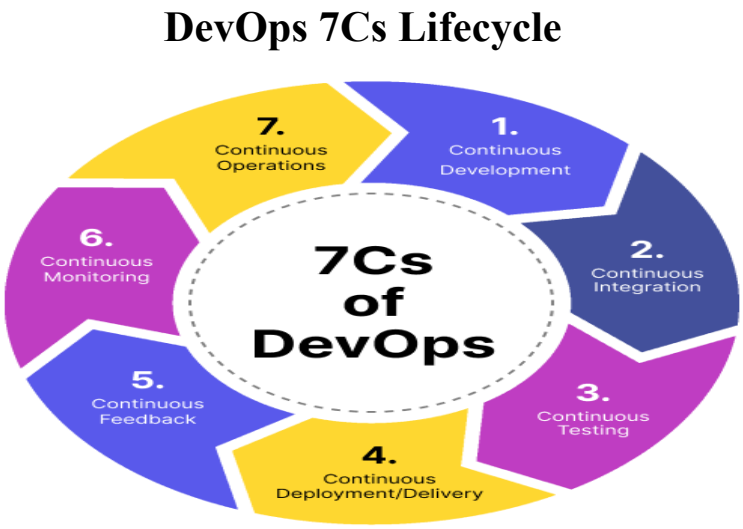

Figure 2: The 7Cs of DevOps lifecycle – Continuous Development, Integration, Testing, Deployment, Feedback, Monitoring, and Operations.

Important Terminology

- CI (Continuous Integration): Automating integration of code and testing.
- CD (Continuous Deployment): Deploying tested code automatically to production.
- IaC (Infrastructure as Code): Managing infrastructure using code.
- Automation: Reducing manual tasks in testing & deployment.
- Pipeline: Workflow from code commit → build → test → deploy.

DevOps Perspective

DevOps is more than tools—it's a mindset shift:

- Developers take ownership beyond coding.
- Ops collaborate early in development.
- Customers stay central to delivery.
- Promotes transparency, trust, and shared ownership.

DevOps and Agile

- Agile: Iterative and incremental development.
- DevOps: End-to-end delivery (development → operations).
- Agile + DevOps = Faster Dev + Faster Ops.
- Together, they ensure flexible and rapid software delivery.

DevOps Lifecycle Stages

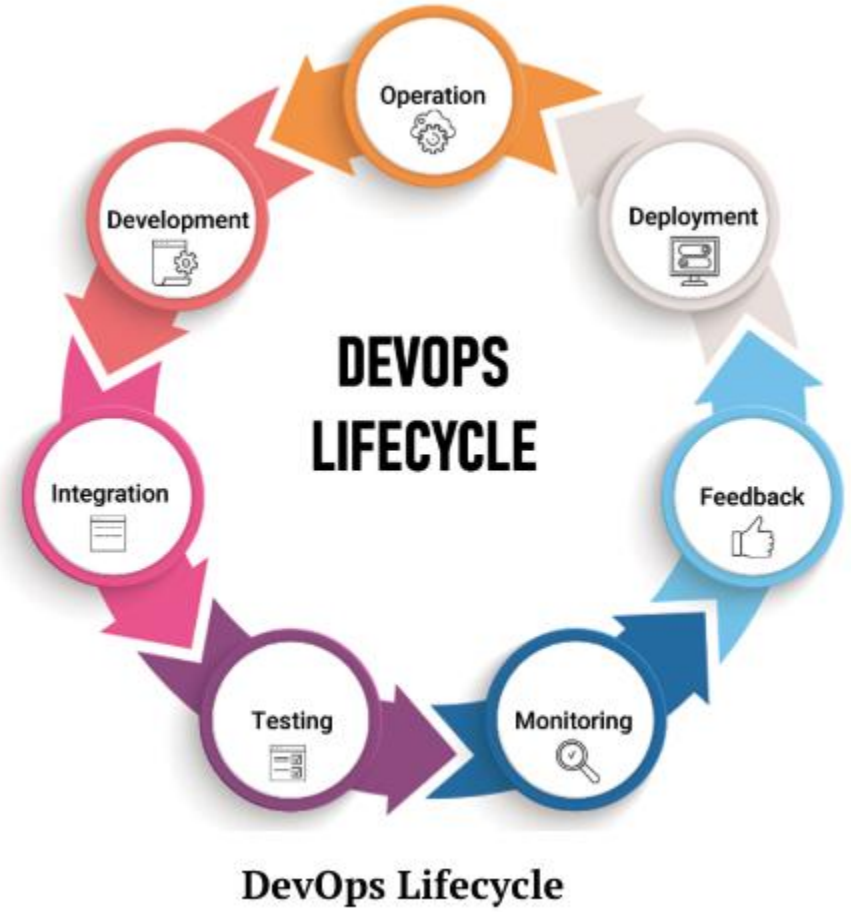

Figure 3: DevOps stages – plan, build, integrate, deploy, operate, monitor.

DevOps Tools

- **Version Control:** Git, GitHub, GitLab, Bitbucket.
- **CI/CD:** Jenkins, GitHub Actions, GitLab CI.
- **Configuration Mgmt:** Ansible, Puppet, Chef.
- **Containers:** Docker, Kubernetes.
- **Monitoring:** Prometheus, Grafana, ELK Stack.
- **Cloud:** AWS, Azure, GCP.

Configuration Management

- Maintains consistency across systems.
- Automates server setup, patching, and updates.
- Ensures quick rollback & versioning.
- Tools: **Ansible, Puppet, Chef**.

Continuous Integration and Deployment (CI/CD)

- **CI:** Developers merge code frequently → automated builds & tests.
- **CD:** Code that passes CI gets auto-deployed to production.

Benefits:

- Faster releases.
- Reduced integration failures.
- Early bug detection.
- Higher customer satisfaction.

CI/CD Pipeline

Figure 4: A simplified CI/CD pipeline from commit → build → test → deploy → monitor.

Final Notes

- DevOps is about people, process, and tools.
- It accelerates delivery while ensuring reliability, scalability, and business value.
- With automation, CI/CD pipelines, and strong collaboration, DevOps transforms software delivery.