Introduction To DevOps

What is DevOps ?

- DevOps combines development (Dev) and operations (Ops) to unite people, process, and technology in application planning, development, delivery, and operations
- DevOps enables coordination and collaboration between formerly siloed roles like development, IT operations, quality engineering, and security.
- DevOps helps teams continually provide value to customers by producing better, more reliable products.

Developer / Programmer

Wall

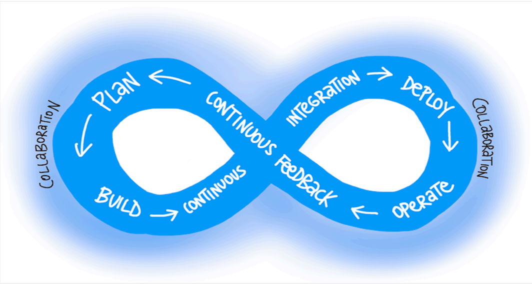
IT Operations Team

The Need for DevOps

- Challenges in traditional software development and operations silos.
- Increased demand for faster and more reliable software delivery.
- Importance of addressing bottlenecks, reducing errors, and improving communication.

Key Principle OF DevOps

1.Collaboration: Emphasizing communication and cooperation between development and operations teams.
2.Automation: Automating repetitive tasks to increase efficiency and reduce human errors.
3.Continuous Integration (CI): Frequent integration of code changes into a shared repository to detect issues early.
4.Continuous Delivery (CD): Automatically deploying code changes to production or staging environments.
5.Monitoring and Feedback: Gathering metrics and user feedback to drive improvements.

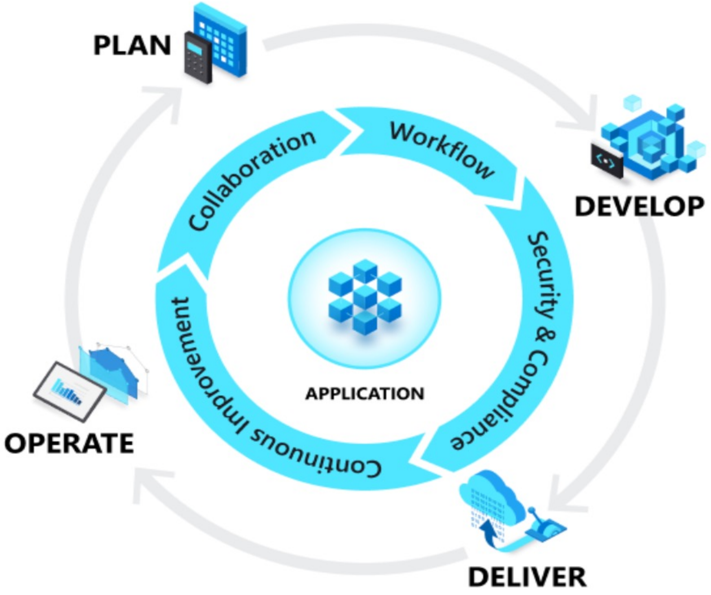


DevOps Goals and Benefits

- **Faster Time-to-Market:** Rapid release cycles and automated processes lead to quicker delivery of features and fixes.
- **Improved Quality:** Continuous testing and integration result in fewer bugs and more stable releases.
- **Enhanced Collaboration:** Cross-functional teams work together, fostering better understanding and cooperation.
- **Reduced Risk:** Smaller, incremental changes reduce the chances of major failures.
- **Scalability:** Easier scaling of infrastructure and applications as needed.
- **Efficiency:** Automation eliminates manual, time-consuming tasks.

DevOps Lifecycle

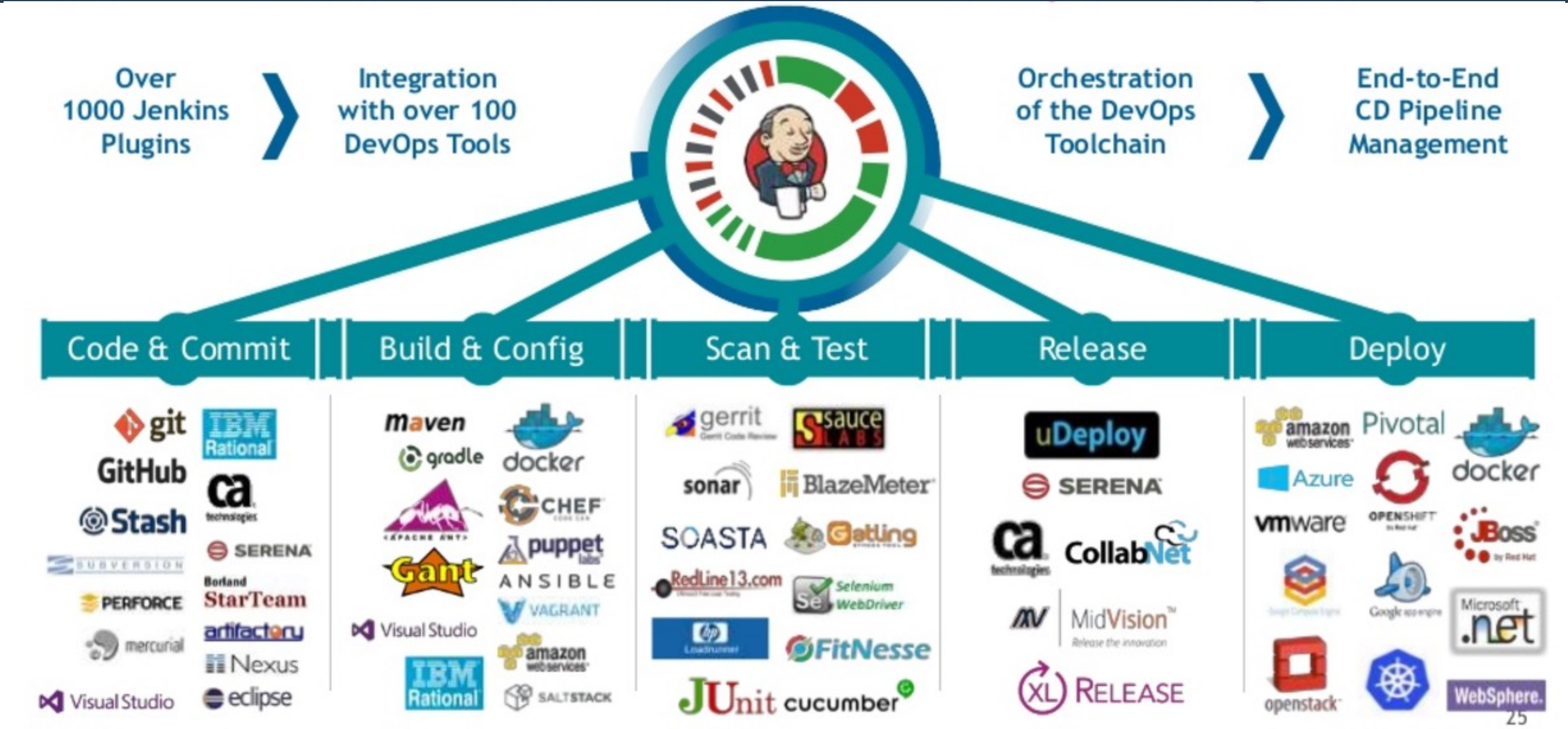
1.Plan: Define goals, prioritize tasks, and plan the development and deployment processes.
2.Code: Develop and review code collaboratively using version control systems.
3.Build: Automate the building and compiling of code into executable artifacts.
4.Test: Conduct automated testing to ensure code quality and functionality.
5.Deploy: Automate deployment to various environments, including production.
6.Operate: Monitor and manage applications in production, responding to issues as they arise.
7.Monitor: Collect performance data, user feedback, and metrics to drive continuous improvement.



DevOps Tools

- Version Control: Git, SVN
- Continuous Integration: Jenkins, CircleCI
- Configuration Management: Ansible, Puppet, Chef
- Containerization: Docker, Kubernetes
- Monitoring: Prometheus, Grafana
- Collaboration: Slack, Jira

DevOps Tools



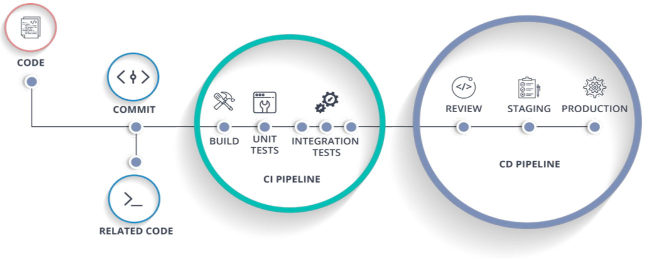
Over 1000 Jenkins Plugins ❯ Integration with over 100 DevOps Tools

Orchestration of the DevOps Toolchain ❯ End-to-End CD Pipeline Management

Code & Commit | Build & Config | Scan & Test | Release | Deploy

Case Study: DevOps Success Story

- Highlight a real-world example of a company's transformation with DevOps.
- Discuss the challenges faced, solutions implemented, and outcomes achieved.



Challenges and Considerations

- **Cultural Shift:** Overcoming resistance to change and fostering a culture of collaboration.
- **Tool Integration:** Selecting and integrating the right tools for your environment.
- **Security:** Balancing speed with security measures in the development pipeline.
- **Continuous Learning:** Keeping up with evolving DevOps practices and technologies.

Recap of Key Points

- **DevOps Objective:** DevOps bridges the gap between software development and operations, promoting collaboration and efficient processes.
- **Core Principles:** DevOps is guided by principles of collaboration, automation, continuous integration, continuous delivery, and monitoring and feedback.
- **Benefits:** Implementing DevOps yields faster delivery, improved quality, enhanced collaboration, reduced risk, scalability, and operational efficiency.
- **Lifecycle:** The DevOps lifecycle involves Planning, Coding, Building, Testing, Deployment, Operation, and Monitoring (PCBTDOM).
- **Tools:** DevOps utilizes tools like Git, Jenkins, Docker, Kubernetes, and more, to automate, monitor, and facilitate collaboration.
- **Challenges:** DevOps adoption faces challenges like cultural transformation, tool integration, security, and the need for continuous learning.

QnA