

DevOps

Anil Joseph

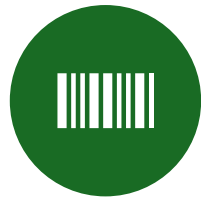
What is DevOps?

- DevOps is a set of practices that combines software development (Dev) and IT operations (Ops) to shorten the systems development life cycle and provide continuous delivery with high software quality.

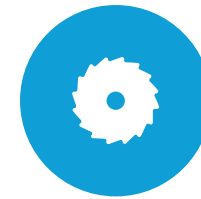
Lifecycle



Plan(Dev)



Code(Dev)



Build(Dev)



Test(Dev)



Release(DevOps)



Deploy(Ops)



Operate(Ops)



Monitor(Ops)

Challenges

- Siloed Teams
 - Development and operations teams work in isolation, leading to poor communication and collaboration.
 - Delays in project timelines, misunderstandings, and lack of shared goals.
- Slow Deployment Cycles
 - Manual processes and lack of automation slow down the deployment of new features and updates.
 - Increased time to market, inability to quickly respond to market demands or customer feedback.

Challenges

- Higher Failure Rates
 - Infrequent releases and lack of continuous testing lead to more bugs and errors in production.
 - Higher risk of downtime, increased troubleshooting time, and lower software quality.
- Inefficient Incident Management
 - Without integrated monitoring and logging, identifying and resolving issues can be time-consuming and complex.
 - Longer recovery times from failures, decreased system reliability, and poor user experience.

Challenges

- **Difficulty in Scaling**
 - Manual configurations and deployments make it hard to scale infrastructure and applications efficiently.
 - Limited ability to handle increased load, potential performance bottlenecks, and increased operational costs.
- **Inconsistent Environments**
 - Differences between development, testing, and production environments can lead to unexpected issues.
 - Increased likelihood of environment-specific bugs, more complex troubleshooting, and deployment failures.

Challenges

- Lack of Continuous Feedback
 - Without continuous integration and delivery, feedback loops between development and operations are slow.
 - Slower identification of issues, delayed improvements, and reduced ability to innovate quickly.
- Security Vulnerabilities
 - Without integrated security practices (DevSecOps), security can be an afterthought rather than a part of the development process.
 - Increased risk of security breaches, higher costs of remediation, and potential legal and reputational damage.

Benefits of DevOps

Improved
collaboration

Faster
deployment

Higher quality

Better
scalability

Consistent
environments

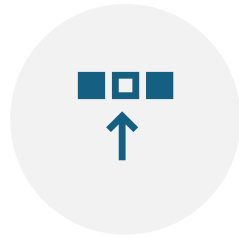
Continuous
feedback

Enhanced
security

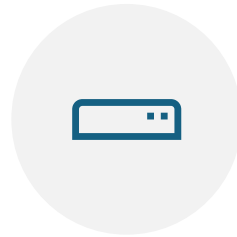
Impact of DevOps



FASTER TIME TO
MARKET



IMPROVED
DEPLOYMENT
FREQUENCY



LOWER FAILURE
RATE OF NEW
RELEASES



SHORTENED LEAD
TIME BETWEEN
FIXES



FASTER MEAN
TIME TO
RECOVERY

DevOps Practices



Continuous
Integration (CI)



Continuous
Delivery (CD)



Infrastructure
as Code (IaC)



Automated
Testing



Monitoring
and Logging

Continuous Integration

- Continuous Integration (CI) is a software development practice where developers frequently integrate their code changes into a shared repository, often multiple times a day.
- Each integration is automatically verified by building the code and running automated tests to detect integration errors as quickly as possible.
- Improved Code Quality:
 - Regular testing helps maintain high code quality.
- Reduced Integration Issues:
 - Early detection of integration problems.
- Faster Feedback:
 - Immediate feedback on code changes.

Continuous Delivery

- Continuous Delivery (CD) is a practice where code changes are automatically prepared for release to production.
- It builds on CI by deploying code changes to testing and production environments after the build stage.
- CD ensures software can be reliably released at any time.

CI/CD



CI: Focuses on integrating and testing code frequently to catch issues early.



CD: Focuses on automating the release process so that the codebase is always ready for deployment.

Infrastructure as Code (IaC):

- A practice where infrastructure is provisioned and managed using code and software development techniques, such as version control and continuous integration.
- This allows for the automation of infrastructure management processes.

DevOps Tools



Version Control:

Git, GitHub, GitLab



CI/CD:

Jenkins, CircleCI, Travis CI, GitHub actions,
Gitlab CI/CD



Configuration Management:

Ansible, Chef, Puppet



Containerization:

Docker, Kubernetes



Monitoring:

Prometheus, Grafana, Nagios

Thank You!

anil.jos@gmail.com