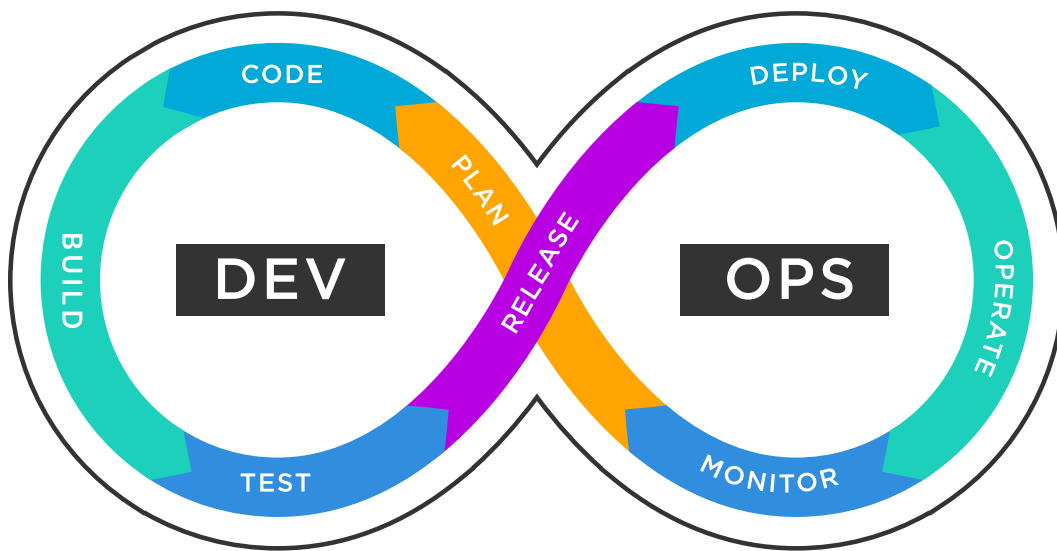


DEVOPS BOOK

PETRO KOLOSOV

1. DEVOPS DEFINITION AND CYCLE

DevOps – is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality [1].



(1) Plan

The plan phase is the foundation of the DevOps lifecycle. It involves setting goals, defining requirements, and creating strategies for developing, deploying, and managing applications and infrastructure. Collaboration between stakeholders, including developers, operations teams, and business units, is critical to align objectives and ensure the successful delivery of projects.

(2) Code

The code phase is where software development occurs. In this phase, developers write, manage, and maintain code collaboratively using version control systems. The focus

is on creating high-quality, maintainable, and secure code while ensuring seamless integration with other phases of the DevOps lifecycle.

(3) **Build**

The build phase is where the source code developed in the previous phase is compiled, tested, and packaged into deployable artifacts. This phase ensures the software is functional, free from major bugs, and ready for deployment to staging or production environments.

(4) **Test**

In the Test phase of the DevOps lifecycle, the main goal is to ensure that the application is functioning as expected and is free from errors before it is released into production. This phase focuses on validating the quality, reliability, and performance of the software using automated testing and other validation techniques. It is essential that tests are run in an environment that closely mirrors production, which is where staging environments and deployment pipelines come into play.

(5) **Release**

In the Release phase of the DevOps lifecycle, the focus is on ensuring that the application is ready for deployment into the production environment. This phase involves activities to package, stage, and deploy the application to production. It ensures that the release is smooth, reliable, and consistent with the desired configurations and quality standards.

(6) **Deploy**

The Deploy phase in DevOps focuses on moving the application from a staging environment to the production environment. It involves releasing and distributing the application, making it accessible to end-users. This phase ensures that the deployment is seamless, consistent, and efficient, without introducing downtime or disruptions.

(7) **Operate**

The operate phase in the DevOps lifecycle focuses on ensuring the smooth and reliable

running of applications in production. This phase involves managing infrastructure, monitoring application performance, responding to incidents, optimizing systems, and maintaining high availability and security. It emphasizes automation, proactive monitoring, and incident response to ensure systems are robust, secure, and scalable.

(8) **Monitor**

In DevOps, the Monitor phase focuses on actively tracking the performance and health of applications, infrastructure, and systems in real-time. It is essential for maintaining operational efficiency, identifying issues, and continuously improving system reliability. Monitoring ensures that potential failures are detected early, and systems can be optimized for peak performance.

2. TECHNOLOGIES PER DEVOPS CYCLE

(1) **Plan**

JIRA, Trello, Confluence, Azure Boards, Microsoft Project, Slack, Google Docs, Microsoft Teams, Notion, Miro, LucidChart

(2) **Code**

GIT, GitHub, GitLab, Bitbucket, Visual Studio Code, IntelliJ IDEA, Eclipse, PyCharm, Sublime Text, Atom, Jenkins, Azure DevOps, GitHub Actions, Azure Pipelines, GitFlow, SonarQube, ESLint, Prettier

(3) **Build**

Jenkins, Azure DevOps, GitLab CI, CircleCI, Travis CI, Bamboo, TeamCity, GitHub Actions, Apache Maven, Gradle, Ant, Nexus, Artifactory, Docker, SonarQube, Snyk, MendBolt, Docker Compose

(4) **Test**

Selenium, Cypress, Postman, Mockito, Cucumber, SoapUI, PyTest, JMeter, Jenkins, GitLab CI, CircleCI, Bamboo, GitHub Actions, Azure DevOps, Terraform, Packer, Snyk, LoadRunner, HELM

(5) **Release**

Jenkins, GitLab CI, Azure DevOps, CircleCI, GitHub Actions, Bamboo, Spinnaker, HELM, Terraform, Ansible, Docker, Artifactory, Nexus, Chef, Puppet, Capistrano, AWS CodePipeline, CloudFormation, GIT, Bitbucket

(6) **Deploy**

Kubernetes, Docker, HELM, AWS CodeDeploy, Azure DevOps, Jenkins, Ansible, Terraform, Puppet, Chef, Spinnaker, GitLab CI, Octopus Deploy, Google Cloud Deployment Manager, Docker Compose, AWS Elastic Beanstalk, Bamboo, Rancher, Capistrano, Azure App Service, CircleCI

(7) **Operate**

Prometheus, Grafana, Ansible, Chef, Puppet, Nagios, Zabbix, Datadog, New Relic,

Splunk, ELK Stack (Elasticsearch, Logstash, Kibana), AWS CloudWatch, Azure Monitor, OpenTelemetry, Fluentd

(8) **Monitor**

Prometheus, Grafana, Nagios, Zabbix, Datadog, New Relic, Splunk, ELK Stack (Elasticsearch, Logstash, Kibana), AWS CloudWatch, Azure Monitor, OpenTelemetry, Fluentd, Fluent Bit

REFERENCES

[1] Len Bass, Ingo Weber, and Liming Zhu. What Software Architects Need to Know About DevOps. 2015.

<https://www.informit.com/articles/article.aspx?p=2424801>.

0.1.26+main.2a9310b