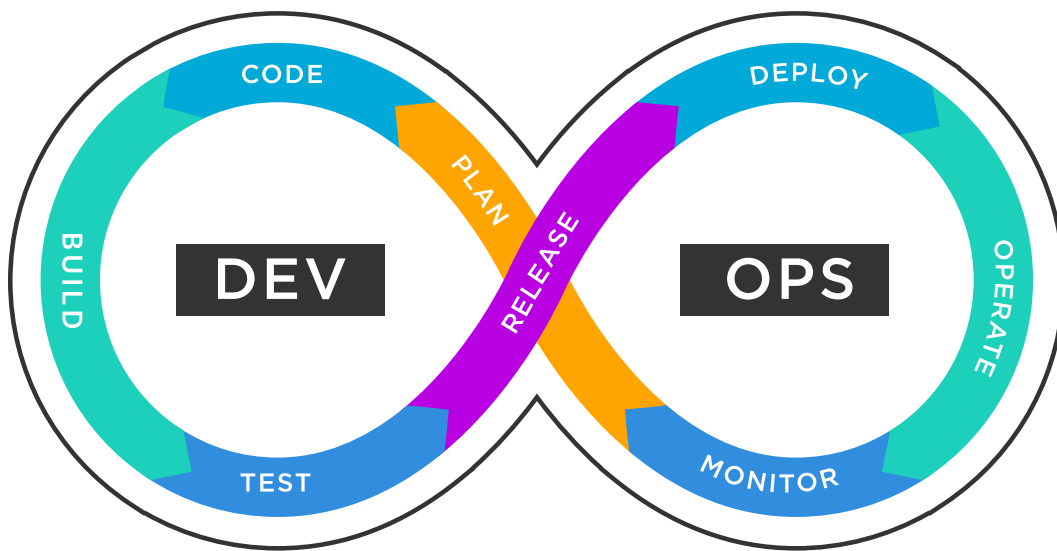


DEVOPS BOOK

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 first stage is the planning of our future product or functionality. At this stage, we need to answer three simple questions:

- **Who are we doing this for?**

By asking this question, we need to determine who our ultimate target audience is – the audience that will use our service.

- **What are we doing?**

By asking this question, we must precisely define the technical specifications for our future product.

- **Why are we doing it this way?**

By asking this question, we need to determine why certain architectural or other decisions were made, and why specific components were chosen. Each decision and its justification should be documented.

(2) **Code**

At this step, developers write, manage, and maintain code collaboratively using version control systems. The focus is on creating high-quality, maintainable, and secure code.

(3) **Build**

At this step, the source code 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**

At this step, the assembled application is tested on environment that closely mirrors production. This involves launching the application and verifying its functionality as well as the functionality of the entire system. In addition to basic testing, integrations with other software components are also checked for proper operation.

(5) **Release**

This step is informational in nature. The notification to the responsible administrator includes information that a new release has been prepared, it has passed all necessary tests, and is ready for deployment on the production platform.

(6) **Deploy**

At this step, release artifact is being deployed to 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.

(7) **Operate**

The step where application management takes place. This includes its configuration, granting access to end users, and managing response to incidents and problems.

(8) **Monitor**

At this step, we collect information about the application's performance: its metrics, logs, and user feedback. After that, we move back to the Plan stage again. Our infinity has come full circle.

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, Snky, MendBolt, Docker Compose

(4) **Test**

Selenium, Cypress, Postman, Mockito, Cucumber, SoapUI, PyTest, JMeter, Jenkins, GitLab CI, CircleCI, Bamboo, GitHub Actions, Azure DevOps, Terraform, Packer, Snky, 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>.

0.1.32+main.80c98fe