

# Methods for Automating Development Processes and Deployment of Microservice Applications

Oleksandr Koshedran<sup>1</sup>

Vitaliy Tkachov<sup>2</sup>

<sup>1</sup> Kharkiv National University of Radio Electronics, 14 Nauky Ave, Kharkiv UA-61166, Ukraine, [oleksandr.koshedran@nure.ua](mailto:oleksandr.koshedran@nure.ua)

<sup>2</sup> Kharkiv National University of Radio Electronics, 14 Nauky Ave, Kharkiv UA-61166, Ukraine, [vitalii.tkachov@nure.ua](mailto:vitalii.tkachov@nure.ua)

**Abstract:** The presented work is devoted to automation CI/CD pipeline for deploying microservices. Main methodologies and practices of software products development are considered: Waterfall, Agile, Scrum, Kanban and DevOps as well as detailed description of them. Developed and deployed a CI / CD process to automate web application deployment using a variety of software tools such as Kubernetes and Jenkins.

**Keywords:** K8S, CI, CD, DEVOPS, JENKINS, AWS.

## I. INTRODUCTION AND PROBLEM STATEMENT

Traditionally, software has been developed using a cascading methodology, but recently Agile and Lean have dominated software development. Today, DevOps practices play a crucial role in implementing this flexible approach to software development

The architecture of microservices offers a number of advantages in comparison with a traditional monolithic design. Building a system of loosely coupled services allows teams to work independently, and individual services can be deployed and scaled independently. As a result, the system becomes more fault-tolerant and reliable, the equipment is used more efficiently, and changes are made faster.

Microservices certainly have several deployment and development advantages, but they also come with their own challenges. Managing microservice repositories and pipelines becomes much more difficult as the number of applications grows.

## II. PROPOSED SOLUTION

In a microservice architecture, continuous integration and continuous delivery CI/CD are critical to creating a flexible environment for making gradual changes to your system. This requires the immediate creation, testing, and deployment of every code change you make to your repository, to your infrastructure.

The goal of any software development process is not (or at least should not be) the perfect implementation and execution of the process itself. Typically, the goal is to benefit users by satisfying a need or solving a problem, ideally so that they are satisfied so that they prefer your product to competitors' product. Many organizations, from small startups to large multinational corporations, use a flexible iterative approach to software development precisely because it helps them benefit users more quickly. Implementing a microservice architecture helps achieve this goal by allowing you to make changes faster, which in turn means you can collect feedback, see what

works and what doesn't, and customize, adjust, or even change in response. By using these short iterations effectively, you can provide your users with a valuable product. Continuous integration CI, delivery CD and deployment (also CD) have been promoted for years as a method and way of thinking to accelerate software release. This approach is not specific to microservices - it is also used to create monolithic systems - but it is an important component in the development of a program based on microservices.. Automated CI/CD needs to be combined with a testing strategy that includes both automated testing and monitoring in the production environment, and works best when organizations adopt a DevOps culture.

## III. CONCLUSIONS

The effectiveness and feasibility of developing and deploying CI / CD as a software development methodology was demonstrated. This significantly reduces the time required for the release and feedback of developers, testers, managers and other team members. The result of the work was the development and deployment of the CI / CD process, a comparison of different methods of software development and conclusions about the relevance of current trends. The built process can be used on real projects, but will require a number of improvements due to a limited budget. Various and modern technologies for deployment were also researched, without which hardly any project will be able to be developed efficiently and expediently.

The choice of technologies such as cloud computing and virtualization at the operating system level, methodologies such as infrastructure such as code, continuous integration and continuous deployment, software tools such as Jenkins, Kubernetes, etc. was justified.

## REFERENCES

- [1] Leszko, R., "Continuous Delivery with Docker and Jenkins: Delivering software at scale," Kindle Edition, 2021, 544 p.
- [2] Tkachev V., Application of the method of preventing collisions during parallel data processing in polling networks for monitoring the state of complex distributed systems / V. Tkachev, A. Kovalenko, V. Lebedev. 90 p.
- [3] Tkachev V., Application of the method of collision prevention in parallel data processing in polling networks for monitoring the state of complex distributed systems / V. Tkachev, A. Kovalenko, V. Lebedev // Third International Scientific and Technical Conference "Problems of Informatization" November 12-13, 2015. - Cherkasy-Baku-Bielsko-Biala-Poltava. - 46 p.
- [4] Surovich, "Kubernetes and Docker - An Enterprise Guide," Apress, 2020, 525 p.