DevOps

Software Architecture

March 23, 2023

Richard Thomas

1 Brief

DevOps is a portmanteau[1] of development and operations. It is intentionally a portmanteau to emphasise that the approach requires a close integration of development and operations behaviours in one team. In a proper DevOps environment the development team is responsible for the infrastructure on which their software runs, not just for the software itself. This can include considering infrastructure costs in their development budget and estimates. This week we will

- provide an overview of DevOps,
- consider what are necessary DevOps practices, and
- let you explore how you might implement a DevOps pipeline.

1.1 Introduction to DevOps

You should be familiar with the concepts of automated testing and continuous integration. We will now extend that to include continuous testing and deployment, and then expand on these to provide a full DevOps process.

For an introduction to DevOps read Amazon's description of "What is DevOps?"[2] [1]. Do not worry about the discussion about microservices. That is an architectural style that will be covered later in the course. DevOps does not require a microservices architecture, though there are some benefits of using it.

Skim through the description of implementing DevOps at Wotif[3] in "DevOps: Making it Easy to Do the Right Thing"[4] [2]. Focus on the section titled "**Making it Easy**." This is where they discuss how they provided the environment to support DevOps.

1.2 DevOps Practices and Tools

You should be familiar with some of the practices and tools used in a DevOps process. Amazon's description of "What is DevOps?"[5] [1] lists what they consider to be required practices. For any practices (aside from microservices) that you are not familiar with, you should follow the links to read a summary of those practices. Amazon naturally describes their tools for implementing a DevOps pipeline. You should skim the description of Amazon's tools at "DevOps and AWS"[6] [3]. You do not need to be familiar with these tools, but should have a general idea of what services the tools provide from their summaries.

[1] https://www.britannica.com/topic/portmanteau-word
[2] https://aws.amazon.com/devops/what-is-devops/
[3] Now Expedia.
[4] https://search.library.uq.edu.au/permalink/f/tbms52/TN_cdi_webofscience_primary_000383092600012CitationCount
[5] https://aws.amazon.com/devops/what-is-devops/
[6] https://aws.amazon.com/devops/

Another view of necessary DevOps practices is that it requires continuous

- development,
- integration,
- testing,
- operations,
- deployment,
- monitoring, and
- feedback.

These can only be achieved through automation.

2 Requirements

Before the Tutorial

- Read the article and web pages indicated above.
- Identify a tool that can be used for each of the seven practices listed above.
- Come to the tutorial with a list of tools and be prepared to give a ten second summary of each tool.

During the Tutorial

- Review and summarise tools that support DevOps practices.
- Discuss how different tools can be used to implement a DevOps pipeline.
- Define a DevOps pipeline for the Sahara eCommerce case study [4].

3 Outline

Introduction (5 minutes)

Introduction to the brief, summarising the idea and value of DevOps and its background.

Small Group Discussion (12 minutes)

In small groups, describe the tools you identified that support each of the DevOps practices. Briefly explain how they can be integrated to deliver a complete DevOps pipeline.

Class Discussion (8 minutes)

With the class, summarise the tools identified by each group and which DevOps practices they support. Consider the following questions:

- How well does each tool support one or more of the DevOps practices (i.e. development, integration, testing, operations, deployment, monitoring, and feedback)?
- Are there advantages to having tools support multiple practices, or is it easier to integrate tools if they only support a single practice?

Pipeline Design (15 minutes)

In a small group, design a DevOps pipeline for the Sahara eCommerce case study [4]. Use the service-based architecture approach for the project.

- What types of tools would be required?
- Which specific tools would you choose?
- On which type of computing infrastructure would you deliver the system?
- What parts of the deployment and operations processes could be automated?

Pipeline Discussion (10 minutes)

With the class, present a few of the pipelines summarising how the tools support delivering an integrated DevOps process. Consider the following questions:

- How well does the entire tool chain support all seven DevOps practices (i.e. development, integration, testing, operations, deployment, monitoring, and feedback)?
- How well would the tool chain support an integrated perception of DevOps as an organisational process?
- Would additional tooling (e.g. scripting) be required to enable smooth integration of the tools? Where would that be required?

4 Challenges

Challenge 1: DevOps in Practice

Read or skim "DevOps Capabilities, Practices, and Challenges: Insights from a Case Study"[7] [5]. Consider the differences between the capabilities and technological enablers mentioned in the article, and the seven DevOps practices listed in section 1.2.

- Do the seven necessary DevOps practices map perfectly to the enablers in the article by Senapathi et al [5]?
- If there are some mismatches, are they important enough that they should be a required practice?

Post your thoughts about these questions in the **pracs-tut** channel on Slack.

Challenge 2: DevOps Best Practice

Scan the literature[8] and identify what is currently considered to be DevOps "best practice." Summarise your findings and post these in the **pracs-tut** channel on Slack. include links to the sources you used in coming to your conclusion.

[7] https://search.library.uq.edu.au/permalink/f/tbms52/TN_cdi_arxiv_primary_1907_10201
[8] This may be formal peer-reviewed sources or published websites or blogs.

Challenge 3: Reuse or Create

Based on your reading and experience, is it better to use an existing DevOps process and pipeline, or is it better for a team to define their own process and create their own pipeline? Pick one of the sides of this point and post a paragraph arguing your view in the **pracs-tut** channel on Slack.

References

[1] AWS, "What is DevOps?." https://aws.amazon.com/devops/what-is-devops/.

[2] M. Callanan and A. Spillane, "DevOps: Making it easy to do the right thing," *IEEE Software*, vol. 33, no. 3, pp. 53–59, 2016.

[3] AWS, "DevOps and AWS: Tooling and infrastructure resources for DevOps practitioners." https://aws.amazon.com/devops/.

[4] R. Thomas, "Service-based architecture slides," March 2022. https://csse6400.uqcloud.net/slides/service-based.pdf.

[5] M. Senapathi, J. Buchan, and H. Osman, "DevOps capabilities, practices, and challenges: Insights from a case study," in *Proceedings of the 22nd International Conference on evaluation and assessment in software engineering 2018*, vol. 137700 of *EASE'18*, pp. 57–67, ACM, 2018.