

DevOps for Developers

An introduction to DevOps for developers and its purpose in software development.

Introduction

Erik Reinert aka "Blackglasses"

- Senior Software Engineer
- Content Creator (@TheAltF4Stream on Twitch & YouTube)
- Obsessed with problem solving

Introduction

Erik Reinert aka "Blackglasses"

- Started my career with frontend (2 years)
- Followed my curiosity with backend (2 years)
- Continued my curiosity with fullstack (2 years)
- Found my passion with DevOps (3+ years)

Introduction

Let's begin our dive into DevOps by starting from scratch.

Introduction

To enter a career in DevOps, it is helpful to have experience in both software development and IT operations. This may include experience with coding and scripting languages, as well as experience with systems administration and infrastructure management.

Having a strong understanding of collaboration and communication tools and methodologies, such as agile and lean principles, can be beneficial. It is also important to have a passion for continuous learning and staying up-to-date with the latest technologies and best practices in the field.

Introduction

Understanding FIRST, tools SECOND. DevOps is a philosophy not a technology.

(first half lectures, second half programming)

What Exists Today

What Exists Today

Poorly Defined

Because DevOps is a broad and somewhat abstract concept, it can be difficult for companies to define and implement in a concrete and consistent way.

What Exists Today

Culture Challenges

Additionally, because DevOps involves cultural shifts within an organization, it can be challenging for companies to adopt and integrate into their existing processes and ways of working.

What Exists Today

Missing Communication

A lack of clear communication and collaboration between these teams can lead to errors and other problems that can compromise the quality of the solutions being produced.

What Exists Today

Suboptimal Solutions

Developers and IT professionals may not be working together as efficiently and effectively as possible, which can result in suboptimal solutions that do not meet the needs of the business or the end users.

Where do we begin?



Start from Scratch

- forget what you've **seen** about DevOps
- forget what you've **heard** about DevOps
- forget what you've **done** with DevOps

What DevOps IS

The goal of DevOps is to improve the speed and reliability of software releases by automating parts of the software development and deployment process, and by fostering a culture of collaboration and experimentation among teams.

What DevOps IS NOT

DevOps is not a specific tool or technology. It is not a job title or a job description, and it is not a standalone practice that can be implemented in isolation.

What DevOps IS NEVER

Tools or Technologies!



DevOps is a Philosophy

DevOps is a philosophy or mindset that emphasizes collaboration, communication, and integration between software developers and IT operations teams.

Define YOUR DevOps

We need to define what DevOps means uniquely to us before we can properly harness its power for an organization.

Three W's of Strategy

The three W's of strategy are:

Why - reasons and motivations behind the strategy.

What - goals and objectives that the strategy is intended to achieve.

Who - people who are benefiting from this.

Together, these three W's help to provide a clear understanding of the strategy and what it aims to accomplish.

The Why, What & Who

- Why do we need DevOps?
- What do we accomplish with DevOps?
- Who do we help with DevOps?

Why do we need DevOps?

DevOps aims to help organizations rapidly and reliably build and deploy software by automating and monitoring the process from end to end.

What do we accomplish with DevOps?

By implementing DevOps practices, organizations can automate and monitor the process of software development from end to end, allowing teams to release and iterate on their products quickly and effectively.

Who do we help with DevOps?

DevOps helps organizations and teams improve their ability to deliver value to their customers and users.

Standards vs. Rules

Standards vs. Rules

Developers don't like rules but will adhere to standards for the greater good of the product.

Standards vs. Rules

DevOps standards are guidelines or best practices that organizations can follow to implement standards successfully. These standards typically cover areas such as communication, collaboration, and automation, and can help organizations to improve their software development process and deliver better products to their customers.

Standards vs. Rules

Automation Standards

Automating the software development and deployment process, using tools such as continuous integration and delivery (CI/CD).

Standards vs. Rules

Planning Standards

Using agile software development methodologies, such as Scrum or Kanban, to increase collaboration and flexibility.

Standards vs. Rules

Quality Standards

Implementing strict version control and configuration management processes to ensure that software is properly tested and deployed

Standards vs. Rules

Communication Standards

Encouraging open communication and collaboration among all members of the development team, including developers, IT operations professionals, and business stakeholders.

Improving Standards

Using metrics and data to track and improve the performance of the software development process.

The Power of TDDs

The Power of TDDs

Technical Design Document

A technical design document (TDD) is a document that describes how a piece of software will be implemented and how it will work. It typically includes detailed information about the system's architecture, modules, interfaces, and data structures, as well as information about any external dependencies or required resources.

The Power of TDDs

Technical Design Document

A technical design document is used to guide the development of a software system, and can be used to communicate the technical aspects of the system to stakeholders and developers.

The Power of TDDs

Your First TDD

There is no standardized way to writing documentation. The best advice I've ever been given was focus only on the specific issue or problem.

Ask yourself what the problem IS and tell the story around that. Set proper boundaries if you need to and include an "out-of-scope" section when you discover items outside of the current issues problem resolution path.

The Power of TDDs

Your First TDD

Apply existing common practices to problem solving that the organization has agreed upon. In this case, we can use the "Three W's of Strategy" to help solve our problem.

The Power of TDDs

Your First TDD

- What problem are we trying to solve? (you should always have this)
- What is the current process? (optional, though normally helpful)
- What are the requirements? (optional, can help with scope)
- How do we solve this problem? (you should always have this)

The Power of TDDs

Your First TDD

Problem

We currently have no way to monitor stream output logs from services and other things running in our Kubernetes cluster.

The Power of TDDs

Your First TDD

Requirements

- We only care about `stdio` - no use for fields advanced indexing, etc
- We don't want to have to manage infrastructure
- We only want to solve simple logging for all of our services in Kubernetes to unblock when we have errors and need to find them, history, etc
- We don't want to deal with metrics right now

The Power of TDDs

Your First TDD

Self Hosted

- Loki (with Grafana)
 - Free and includes most features
 - The query language is very adoptable and simple
- Elastic with Kibana (ELK)
 - Free but limited with features (licensed)
 - Provides a lot of extensibility with searching logs

The Power of TDDs

Your First TDD

Cloud Hosted

- Grafana Cloud [Loki w/ Grafana] (base free - 50GB)
- Solarwinds Papertrail (base \$7 - 1GB)
- Elastic Cloud (base \$95)
- Solarwinds Loggly (base \$79)
- AWS OpenSearch (base \$80 - 50GB + cluster instances)
- New Relic (base free - 100GB)

The Power of TDDs

Your First TDD

Proposed Solution

Looking at the list above there is only one solution that really stands out to solving our problem the quickest with best accuracy:

- Grafana Cloud