simplilearn

DevOps Basics

A Beginner's Guide

TABLE OF CONTENTS

INTRODUCTION

DevOps has witnessed an interesting curve of growth over the last decade. In 2009, Patrick Debois, a project manager, consultant, and agile practitioner, founded the first DevOps conference in Belgium, named DevopsDays. Later in 2012, Alanna Brown created and launched the "State of DevOps" report at **Puppet Labs**, a software and IT automation company headquartered in Portland, USA.

The **2014 "State of DevOps"** annual report showed that the implementation of DevOps is accelerating. In the same year, Janet Gregory and Lisa Crispin released a book, More Agile Testing, giving new impetus to the emerging concept.

Between 2015 and 2016 DevOps adoption increased at the rate of 8 percent, followed by an increase from 10 percent in 2017 to 17 percent in 2018. According to a **MarketsandMarkets report**, the DevOps market will grow from $3.42 Billion in 2018 to $10.31 Billion by the end of 2023, at a CAGR of 24.7 percent.

Experts believe that widespread digitization of businesses, a rapid rise in cloud technology adoption, and the ever-increasing requirement for improved IT collaboration to boost the operational efficiency is driving the DevOps market growth.

DevOps is revolutionizing workplaces in numerous ways. Also, with that, the demand for DevOps experts is skyrocketing each year. DevOps engineers are among the 7th most-coveted and highly-paid IT professionals today, earning an average salary of **US$ 110,000** according to the Robert Half Technology 2019 Salary Guide.

If you are considering a career in DevOps, this **DevOps handbook** will introduce you to the **basics of DevOps**, its industry applications, real-life deployment, key terminologies, and the skills you need to make an impact. Let us begin with the **DevOps Basics**.

A PRIMER ON DEVOPS

The shortened form of software development (Dev) and IT operations (Ops), DevOps combines a variety of tools, philosophies, and practices that augment an organization's capacity to deliver services and applications at an accelerated rate.

DevOps enables organizations to quickly adapt to changing market conditions, helping businesses to improve their products at a much faster pace compared to organizations deploying traditional infrastructure management and software development processes.

In recent times, the processes involved in building and deploying advanced, modern applications underwent a paradigm shift. Activities that were separate and carried out in isolation, are today being merged into a seamless, single process that integrates both IT operations and software development.

This unique methodology, described as DevOps, bridges the gap between operations and development to ensure continuous software delivery more reliably and rapidly.

Under DevOps, operations and development engineers work together across the full life-cycle of an application, from its development to testing, deployment, and operations. The tight integration reduces the friction in automating processes that have been historically manual and sluggish.

Top DevOps benefits

- Speed
- Reliability
- Scalability
- Security

INDUSTRY APPLICATIONS OF DEVOPS

DevOps became mainstream when organizations began to realize that collaboration between operations and development teams helps speedy and efficient project execution while reducing costs.

It is obvious that tech-focused companies, such as Google, Amazon, Apple, Netflix, Uber, LinkedIn, Etsy, and Airbnb were early adopters, but at present, DevOps practices dominate the software development and deployment process in every industrial environment.

Here are some real-life examples of DevOps implementation:

Airlines

The need for a rapid transition to a digital ecosystem is driving every industry to adopt new practices and software tools. The aviation industry, which is a complex sector, has led the digital transformation for many years. When it comes to DevOps, the airline industry was among the early adopters.

Delta Air Lines, one of the top carriers in the United States, serving over three-hundred destinations in 50+ countries, **implemented DevOps** when the concept began taking shape. Jasmine James, IT manager, DevOps Center of Excellence at Delta Air Lines, says the company's DevOps model helped remove the silos between operations and development teams, resulting in continuous integration, improved version control, and a focused approach to business-critical needs.

Banking

The banking industry, worldwide, has been reluctant to uproot its legacy systems. However, to maintain a business advantage, forward-looking institutions are increasingly realizing the importance of transforming and taking advantage of the digital disruption.

British multinational financial services company, **Barclays, implemented DevOps in 2015** to increase its enterprise-wide agility.

In the same year, South Africa based international financial services group, Standard Bank, began its DevOps journey to deliver digital banking solutions. Today, the financial major has over 200 teams implementing DevOps to enhance user experience.

An increasing number of banks are now adopting DevOps practices for boosting their competitive edge over rival Fin-Techs.

Government

The federal government of the United States is highly active in **adopting DevOps,** with the Federal Reserve, the U.S. Customs and Border Protection, and the Department of Justice, all implementing the DevOps method.

Not only the U.S. administration, but also governments around the world are considering adopting DevOps to offer citizens improved e-governance that facilitates efficient management, enhances transparency, encourages citizen participation, and provides easy access to information.

Healthcare

With a significant number of healthcare enterprises using big data analytics to extract crucial insights, DevOps is steadily gaining ground in the medical sector, accelerating positive developments.

While other organizations were exploring the possibility of adopting DevOps, Kaiser Permanente, an Oakland-based American integrated managed care consortium, **implemented DevOps in 2017**.

Kaiser Permanente has 11.8 million end-users and 200,000 employees. Six million customers use their digital channels and the company's website gets over 300 million visits a year.

Prior to deploying DevOps, the company made some progress on customer-centric fronts, but its overall IT operations were deficient in meeting modern healthcare demands.

With the introduction of the DevOps culture, Kaiser Permanente witnessed massive improvements in terms of responding to service and change requests.

The company's response to service requests became 47 percent quicker and response to change requests was a record 53 percent faster.

Hospitality

The advent of the internet has dramatically transformed the hospitality industry, especially the way people look for hotels, book rooms, or hire event managers.

To stay relevant in the hospitality industry, major hotel chains are turning to DevOps to provide convenience to their customers and find solutions for their needs.

DevOps helps hotel brands cope with changing market dynamics, shortening the time to develop, test, and release new offerings.

British multinational hospitality company, InterContinental Hotels Group (IHG), which has 5,656 hotels and 842,749 guest rooms in over 100 countries, adopted DevOps to develop an advanced software delivery model that heightens customer experience.

Robert Kelley, the Director, Delivery Engineering at IHG, explained in a **2018 blog post** how the company is marching ahead in its DevOps journey.

Insurance

Insurance companies have also made major advances in adopting the DevOps culture. Most global organizations within the insurance industry, for instance, Allianz, implemented DevOps models to offer improved digital customer services.

The German multinational company's management felt that implementing DevOps will enhance their ability to better respond to customer preferences without extensive delays.

The operationalization of DevOps practices also facilitated Allianz's transition to the Cloud.

In a separate development, Columbus-based Nationwide Mutual Insurance Company engaged IBM to advance the organization's agile transformation and help enforce an appropriate DevOps approach for application building and delivery. The move, **over a three-year period, led to an improvement of 50% in code quality.**

Retail

The strategic challenges in the retail industry, such as streamlining operations, and optimizing merchandising and marketing, are forcing global retailers to automate their IT processes to make software development and delivery faster and more responsive. So it is no wonder that major retailers are taking the DevOps approach in developing their organization's software.

Amazon was no doubt a pioneer in this field, followed by Walmart.

The digital unit of Walmart, WalmartLabs, which processes the organization's Big Data, and takes care of mobile commerce and e-commerce, **embraced DevOps in 2013** by acquiring OneOps, a company that builds automation tools for operations monitoring and application deployment.

Around the same time, another American retail corporation, Target, adopted DevOps to develop a first-rate mobile application—Cartwheel—which offers discounts to customers earned through Facebook. Target, today, hosts DevOpsDays and also sponsors DevOps meetups.

Rob Cummings, the infrastructure engineer at luxury department store chain, Nordstrom, Inc., says the DevOps approach has helped automate Nordstrom's heterogeneous infrastructure, which included both new and legacy applications. According to Cummings, DevOps empowers organizations to add agility, and its use is becoming increasingly widespread in the retail sector.

A GLOSSARY OF DEVOPS TERMS

If you are aspiring to become a DevOps engineer, knowing the key DevOps terminologies is among the first things to do. Here's a list of some DevOps terms you must familiarize yourself with:

A/B Testing

A/B testing is a technique that compares user behavior and metrics of different feature variants to evaluate which version is performing better.

Acceptance Testing

Also called UAT (User Acceptance Testing), it determines if a system meets business requirements, user needs, and the authorized entity benchmark. Acceptance testing is repeated whenever there is new application development.

Agile

Agile methodology, which helps manage complex operations, is an approach that focuses on customer feedback, collaboration, and rapid, small releases.

Application Release Orchestration

DevOps teams leverage application release orchestration tools to manage application release processes. These tools provide a complete view of the application release pipeline.

Behavior-Driven Development

BDD (Behavior-Driven Development) can help establish standardized communications between development teams and POs (Project Owners). Proper implementation of BDD can prevent misunderstandings and social frictions.

Black Box Testing

A testing technique that emphasis on the functionality of an Application Under Test (AUT), without considering implementation details, the internal structure of code, or the internal software paths.

Blueprint

DevOps blueprint incorporates CCM (Continuous Change Management), CT (Continuous Test), CI (Continuous Integration), and CD (Continuous Delivery) capabilities with the automation of every operation needed for fast-paced product development.

Build Artifact Repository

The Build Artifact Repository manages the entire artifact life-cycle, providing support to various software management systems while ensuring consistency to CI/CT/CD workflow.

Build Automation

Build Automation involves processes for automating software code retrieval from a repository.

Canary Release

A technique used to lower the risks of introducing new versions of software in development by rolling out small changes to user subsets before rolling the entire software out to the infrastructure and making it available to everyone.

Configuration Drift

A phenomenon where hardware and software configurations tend to drift over time, becoming inconsistent because of manual updates, general entropy, and ad-hoc changes.

Configuration Management

A process that ensures the maintenance and consistent functional abilities of servers, computer systems, and software.

Container

A Container refers to a software unit that packages code and other dependencies to make an application run reliably and quickly across different computing environments. Containerized software runs the same, irrespective of the infrastructure, Windows, Linux, or Mac.

Continuous Delivery

An ongoing practice of testing, building, and delivering user environments and software code improvements using automated tools.

Continuous Integration

A software development methodology where developers merge code changes regularly into a repository, following which automated tests and builds are run

Dark Launch

The launching process of production-ready functionalities to a user subset prior to the full launching. This decouples release from deployment, enabling DevOps teams to get user feedback, perform bug tests, and assess infrastructure performance.

DevOps Intelligence

Correlates and analyzes data from every tool in the software delivery process, offering actionable insights for faster, better, and customer-focused delivery.

DevSecOps

Short for development, security, and operations, DevSecOps makes all stakeholders accountable for ensuring security with the objective of deploying security policies at the same speed and scale as operations and development decisions.

Everything as Code

The concept behind Everything as Code is that security, operations, infrastructure, and compliance are treated as application code so that everything follows the same development practices.

Feedback Loops

Feedback loops are relationship sets between entities, where changes in an entity cause changes in another entity, which eventually leads to changes in the first entity.

Functional Testing

A series of processes to ensure that all software components are functioning properly. Functional testing gives priority to interface testing, to check whether user requirements are being met correctly.

Hybrid Cloud

A cloud computing solution that offers the flexibility of both private and public clouds. The hybrid strategy enables businesses to shift workloads between cloud environments as costs and needs fluctuate.

Infrastructure as a Service

Infrastructure as a Service or IaaS refers to cloud computing services where businesses lease or rent servers for storage and compute in the cloud. It adds agility to software development capabilities.

Infrastructure as Code

IaC (Infrastructure as Code) applies to infrastructure management, including connection topology, load balancers, virtual machines, and networks, using the identical versioning that operations and development teams use for source code.

Jenkins

An open-source tool for automation, Jenkins, scripted in Java, includes a variety of plugins for CI (Continuous Integration).

Kubernetes

An open-source container management system developed by Google, Kubernetes makes managing and deploying distributed systems easier for Linux developers.

Microservices

Microservices characterizes the architectural model of developing distributed applications from discretely deployable services that carry out specific functions. DevOps approaches, such as Continuous Delivery and Continuous Integration drive microservice deployments.

Non-functional Requirements

NFRs (Nonfunctional Requirements) specify system attributes, including usability, scalability, maintainability, performance, security, and reliability.

NoOps

NoOps refers to the automation of deployment, application management, and monitoring. It frees IT teams to concentrate on cross-functional collaboration and higher-level tasks.

Orchestration Pipeline

Products or tools enabling automated tasks, which make up the Continuous Delivery (CD) pipeline. These tools record the output and state of each task, visualizing features flow via the pipeline.

Platform as a Service

PaaS (Platform as a Service) is a cloud computing category that offers a platform to developers, allowing them to build, manage, and run applications.

Product Owner

A crucial leadership role, Product Owner is responsible for managing the complete product life-cycle, from beginning to the end.

Regression Testing

Testing that reviews issues introduced by fresh updates and additions to the code base. Regression testing is done to identify new bugs caused by updates.

Release Management

Release management, in DevOps, is about controlling, planning, and scheduling the development and delivery of software.

Software Chain of Custody

In software development, the chain of custody refers to having full knowledge and understanding of the release status at any point in time.

Test-Driven Development

TDD (Test-Driven Development) is a process in software development that uses a short feedback loop, where developers create tests that fail immediately, and then, they write the code to quickly pass the test.

Toolchain

The toolchain is a combination of DevOps tools that help in the development, management, and delivery of applications across the entire development life-cycle.

Unit Testing

It is about testing single functionality pieces in isolation.

Value Stream Mapping

VSM (Value Stream Mapping) is a visual representation that uses key metrics to measure and evaluate success. It helps developers improve continuously.

Virtualization

Virtualization allows DevOps teams to test, develop, and run applications on live, virtual environments. This facilitates development alongside testing in real-time.

Waterfall

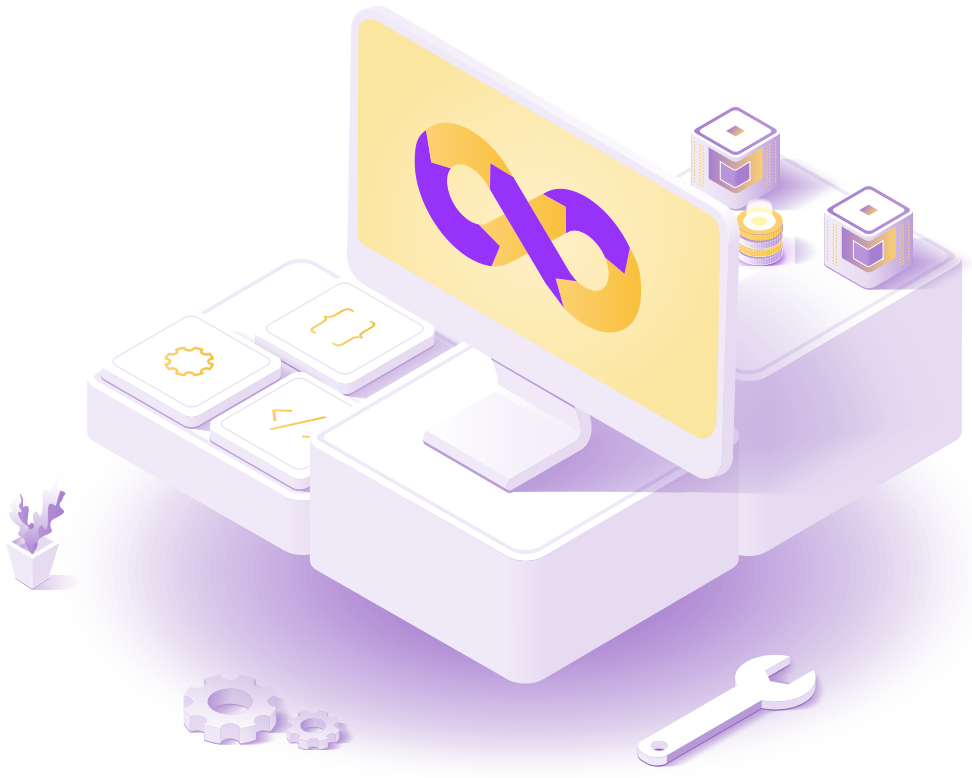
A linear approach to software development, in the waterfall model, the next phase cannot start until the previous phase is over.

White Box Testing

Also called glass box testing, white box testing tests the internal design, structure, and components of a software.

Zero Tolerance

As in, a zero tolerance for service interruptions or failures. Interruptions or failures of any kind can have a devastating impact on an organization. That is why there has to be a zero-tolerance approach to disruptions.



HOW TO BUILD A CAREER IN DEVOPS

In this section, we'll talk about the skills and steps needed to get a sound footing in this field.

The Must-Have DevOps Skills

You can become a DevOps engineer and **earn $100,000 + even without a college degree**. Does it sound unbelievable? Well, in the field of DevOps, there is such a shortage of skilled workforce that employers are considering academic education irrelevant. All you need is the right skill set.

Here are the top skills you must learn to capture the interest of employers:

Scripting Skills

Successful DevOps professionals are experts in scripting code.

Be it Ruby, Perl, Python, or JavaScript, you must have high-level code writing skills to grab the employer's attention.

The perfect DevOps candidate should know how to script manual code or replace specific manual processes, for instance, assigning DNS code or IP addresses.

Decision-Making Skills

DevOps engineers must have exceptional decision-making skills to make quick and confident decisions.

It is one of the basic skills that employers look for when selecting a candidate for a DevOps engineer position.

This is because the ability to make quick decisions can facilitate speedy deployment of new code and fast development.

Knowledge of Infrastructure

DevOps professionals must have a deep understanding of various IT infrastructures and platforms, including web servers, content management systems (CMS), routers, operating systems such as Linux®, data center, network, and the cloud infrastructure.

Most employers see infrastructure-related knowledge as an essential element of DevOps engineers.

The reason being, a comprehensive understanding of infrastructure and platforms will enable DevOps engineers to build and implement applications using the most suitable architecture.

Collaboration Skills

Collaboration is at the heart of the DevOps approach, which brings together teams from both IT operations and software development.

The job of a DevOps engineer is not a one-man show. To become successful in the DevOps field, you must have excellent collaboration and teamwork skills.

Communication Skills

For better collaboration with teams, and to communicate effectively with management and stakeholders, DevOps engineers must have superb communication skills or soft skills.

Fewer errors, faster development, and rapid market deployment will only be possible if DevOps engineers have impeccable communication skills.

Did you find it Useful?

Leave a comment!

Alamin CodePapa

@CodePapa360

FOLLOW FOR MORE

Like Comment Repost