The ABC of DevOps

By Simon McCullough

27 Jun 2019

What do you think about when you hear the term "DevOps"? Abstruse, impenetrable concept or conduit to a brave new world? Something in between? Maybe you haven't even heard about DevOps?Don't worry, you soon will.

Simon McCullough, Major Channel Account Manager, F5 Networks

Back in the day

In the past, software delivery was relatively straightforward. All requirements were defined with the customer before handover to coding and Quality and Assurance (Q&A) testing.

The Ops team then stepped in to deploy everything. So far, so neat and tidy.

Today, the pressure is mounting for businesses to launch new features and services as and when they are needed. If speed and efficiency are not a concern, the old model still works fine. However, the status quo is becoming increasingly incompatible with modern ambitions. Imagine a situation where multiple teams of developers are contributing in parallel. This

happens all the time.

After everything is coded – or even the entire application finalised – they discover that certain components are incompatible. A huge amount of time is wasted if all teams wait to finish their tasks before merging the code into a single application.

Developmental dynamism goes out of the window, and it can become a torturous clean-up and/or retrofitting exercise. Now, let's imagine a second scenario where code integration is successful, and the Q&A team requests a specific environment from Ops to test their application.

Without automation, Ops may take days to provide the required environment. Meanwhile, developers are likely to keep coding away. If a bug is found during the tests, it is possible that developers were coding on top of an existing bug or bugs.

This would potentially require a major coding overhaul. Another common issue is the pace at which Ops can provide the right environment to test the code.

In larger organisations, such delays may take weeks and can involve more than one department. For example, Ops may deploy a new environment that requires special permission from the security team (i.e. another siloed department).

So, how do we stop these frustrations from occurring and keep delivery on track?

DevOps to the rescue!

DevOps work differently. Here's how:

- ### Continuous Integration

  The DevOps approach uses an agile methodology where, typically, smaller functional chunks of code (e.g. a new feature) are regularly and seamlessly integrated into the application's main development branch. Errors are spotted and swiftly corrected.

  This is known as Continuous Integration (CI). The small chunks of code in play here are created within their own isolated (containerised) environment where each (component) typically have a point of communication to 'talk' to other components known as Application Programming Interfaces (API).

  This gives the developer enough flexibility to add or remove components without affecting others. This is known as microservices architecture, wherein each component basically has a plug-and-play capability.

- ### Continuous Delivery

  Following CI, integrated code is automatically tested via several environments, all the way to pre-production where it is either deployable or ready to be deployed. Some companies do not automate further than that and prefer manual deployment. CI and CD interaction is termed as CI/CD.

- ### Continuous Deployment

  The ultimate DevOps goal: Continuous Integration followed by Continuous Delivery and Continuous Deployment (CI/CD²). As a result, applications are automatically deployed to production.

- ### Cloud-centricity

DevOps thrive in the cloud, as it natively supports their tools and unleashes the speed and automation needed for game-changing innovation.

- ## Infrastructure as a Code

  A heavily automated infrastructure is ideal for the Dev team to automatically deploy code at every juncture, from test to deployment environment. Otherwise known as Infrastructure as a Code (IaaS), it is a potent antidote to potential Dev and Ops bottlenecks. A proper IaaS should encompass infrastructure provisioning tools, which build and deploy infrastructure via a click of a button or by quickly filling out a template. Cloud services are a good example. It should also include configuration management tools (e.g. the facility to upgrade, say, 10,000 serves with a single command).

Collaboration

More than most IT disciplines, DevOps-related success is heavily reliant on intensive, intricately coordinated collaboration between customers, developers and IT operations. Developers need to focus on coding. IT operations need to focus on managing automated infrastructure. Both need to talk to each other to uncover new ways to innovate and improve both process and deliverables. Siloed working is yesterday's news.

Fundamentally, DevOps is a practice that can eliminate sources of waste from the application delivery pipeline. It drives efficiency by optimising processes, removing silos, using automation tools, standardising platforms and establishing a strong culture of collaboration. It is a powerful way to bypass the bottlenecks of traditional software development and infrastructure norms, and an unstoppable force for innovation. It is all this and so much more.

DevOps' true influence is only just being felt; stay curious, stay open minded, keep all development teams connected and, whatever you do, don' get left behind!

ABOUT SIMON MCCULLOUGH

Major Channel Account Manager at F5 Networks
- Try to tackle cybersecurity during #RWC2019 - 20 Sep 2019
- Stay safe from cybercrime with what's left of 2019 - 30 Aug 2019
- Multi-cloud's new multiculturism - 21 Aug 2019
- A new phase of cyber warfare has begun - 7 Aug 2019
- The ABC of DevOps - 27 Jun 2019

View my profile and articles...

For more, visit: https://www.bizcommunity.com