

# A Case Study of DevOps at Netflix

## DevOps and its advantages

DevOps, which bridges development and operations, is designed to increase the frequency and quality of code releases. In an ideal setup, you should have a high level of confidence when you go live with code releases in a frequent and highly-automated manner.

High automation leads to time and cost savings and greater development efficiency. These benefits are likely to be seen more and more as applications and development teams scale. Having confidence in fast and agile code releases is key to fostering an efficient and mobile development team.

## In this article:

- I will provide a case study of DevOps at Netflix
- I will be looking at the benefits of growing with cloud-based services, containerisation and building for failure
- I've chosen to look at Netflix because of the scale at which the company operates and because of their strong technical reputation. They were, for example, early adopters of microservices <sup>1</sup>
- I'll finish with a short summary of the benefits of building a business that understands and takes advantage of a positive DevOps culture

## Leveraging existing cloud services

From its roots as a DVD rental business, Netflix introduced its online streaming offering<sup>2</sup> in 2007. Since then, it has grown to a position where in 2015 the service accounted for over 36% of downstream internet traffic in North America<sup>3</sup> in 2015. And in 2017 its users streamed a little over a billion hours of content each week. <sup>4</sup>

To help handle this scale the company started moving to cloud providers in 2008, a process they finished in January 2016.

**“Our journey to the cloud at Netflix began in August of 2008, when we experienced a major database corruption and for three days could not ship DVDs to our members. That is when we realised that we had to move away from vertically-scaled single points of failure, like relational databases in our datacenter, towards highly reliable, horizontally-scalable, distributed systems in the cloud.”**

*Yury Izrailevsky, VP, Cloud Computing and Platform Engineering, Netflix.*

You can achieve horizontal scaling by adding more machines to your resource pool, as opposed to scaling vertically where you boost the performance of your existing machines. Horizontal scaling can provide more options to scale dynamically and should reduce the risks of downtime.

As a company that has to handle large amounts of traffic, Netflix points to the scalability advantages of the cloud as one of the key drivers for their decision to migrate. You could build all of these features from scratch. But this would move the focus of your company away from its business needs and towards the inevitable technical challenges, it would have to tackle to scale effectively and reliably.

“Letting Amazon focus on datacenter infrastructure allows our engineers to focus on building and improving our business.” **John Ciancutti, Co-founder, 60dB.**

Netflix also point towards a certain level of uncertainty around predicted trends in traffic and uptake in new features.<sup>7</sup> Leveraging existing cloud services with growth plans in place takes the guesswork out of scaling. If a company predicts they are going to grow by 50% over the next six months then they will want to be confident their infrastructure can handle this increased traffic. Short peaks in traffic, where traffic goes up for a brief period of time but then returns to its normal rate, should also be handled. With cloud services this is all taken care of, which means that, because you are less concerned with how you will scale, you can focus instead on building a great product.

### **Building with containers**

Containerisation is a method of abstracting away an applications runtime environment so you can run it consistently on different platforms. Containerisation with Docker has become increasingly popular in the past few years. Beyond promoting consistency between environments, a key advantage of containerisation is that containers can be destroyed and created very quickly. This helps with scaling, reliability and efficient rollbacks. In April 2017, Netflix surpassed one million containers launched a week.<sup>8</sup> Scaling with cloud services and containerisation often go hand in hand and there are applications such as Kubernetes which help to automate this process. Netflix have developed their own container management tool called Titus.

“Titus: Netflix’s infrastructural foundation for container-based applications. Titus provides Netflix scale cluster and resource management as well as container execution with deep Amazon EC2 integration and common Netflix infrastructure enablement.” **Andrew Spyker, Andrew Leung, Tim Bozarth, Netflix Technology Blog.**

Titus’ role is to manage containers. Netflix decided to build their own container management software because of their own unique requirements. They also found themselves in a situation where they were migrating existing cloud applications to a containerised environment. Titus allows existing applications to run without modification in a container. It also integrates with

AWS, handles resources sharing and manages capacity.<sup>9</sup> The application thereby reduces the friction and scaling issues that arise when running an application in a containerised environment.

### **Building for failure**

On Christmas Eve 2012 Netflix experienced a partial outage to their service that lasted a number of hours. The cause of this was a fault with AWS.<sup>10</sup> In 2014 it was estimated that an hour of downtime would cost Netflix \$200,000.<sup>11</sup> More recent AWS outages have seen major websites taken offline. However, Netflix's platform can now cope with these kinds of issues.<sup>12</sup>

<sup>13</sup>To help prepare for these scenarios, Netflix builds for failure. This means accepting that at some point parts of your applications are likely not to work as expected. With this expectation in place, you can prepare, in the best way possible, for these eventualities.

The 'Netflix Simian Army' is part of the company's efforts to build for failure. For example, they have a tool they call 'Chaos Monkey' which helps them to test the stability of their production applications.

"A tool that randomly disables our production instances to make sure we can survive this common type of failure without any customer impact. The name comes from the idea of unleashing a wild monkey with a weapon in your data center (or cloud region) to randomly shoot down instances and chew through cables — all the while we continue serving our customers without interruption." **Netflix Technology Blog.**

Teams, working to engineer a solution to protect against potential faults, should be all the more motivated to build a good solution if they know that such problems will be simulated in real production environments. Having control over the timing of these simulations allows you to allocate suitable resources. And by actually simulating the failures that Netflix is building for, the company is able to learn from these experiences and better protect itself against unplanned failures of a similar nature.

### **Creating a DevOps culture**

I've looked at some of the practices Netflix promote in their DevOps culture, as well as briefly looking at some of the tools they have developed as a result of this. At its core, a positive DevOps culture should promote frequent releases, high automation and software reliability. Furthermore, it's advisable to share a high-level understanding of some of the motivations and objectives of a great DevOps culture amongst your larger business team. This will promote the stability and upgradability of applications, and help you to align your development and operations environments with the greater goals of your business as you strive for success in the online world.