Case Study

How Netflix Became A Master of DevOps?

Kapil Laad

DevOps & Cloud Engineer

Swipe to know >>>

Find out how Netflix developed & adopted a DevOps culture and became a “gold standard” in the DevOps World

Netflix’s Move to the Cloud in 2008!

With over **8.4 million** of customers Netflix moved to the cloud and gave their infrastructure a complete makeover. Netflix **chose AWS** as its cloud partner and spent nearly **7 years to complete its cloud migration.**

Netflix didn’t just forklift the systems to AWS, **they chose to rewrite the entire application in the cloud** to become truly cloud-native.

As a significant part of their transformation, Netflix converted its monolithic, data center-based Java application into cloud-based Java microservices architecture.

It brought about the following changes:

- Denormalized data model using NoSQL databases;
- Enabled teams at Netflix to be loosely coupled;
- Allowed teams to build and push changes at the speed that they were comfortable with;
- Centralized release coordination;
- Multi-week hardware provisioning cycles led to continuous delivery;
- Engineering teams made independent decisions using self-service tools.

How Netflix Does DevOps

As a result, it helped Netflix accelerate innovation and stumble upon the DevOps culture. Netflix also gained eight times as many subscribers as it had in 2008. And Netflix's monthly streaming hours also **grew a thousand times** from Dec 2007 to Dec 2015.

The Best Way to Avoid Failure is to Fail Constantly

Migrating to the cloud made Netflix more resilient, but they wanted to be prepared for any unseen errors that could cause them equivalent or worse damage in the future.

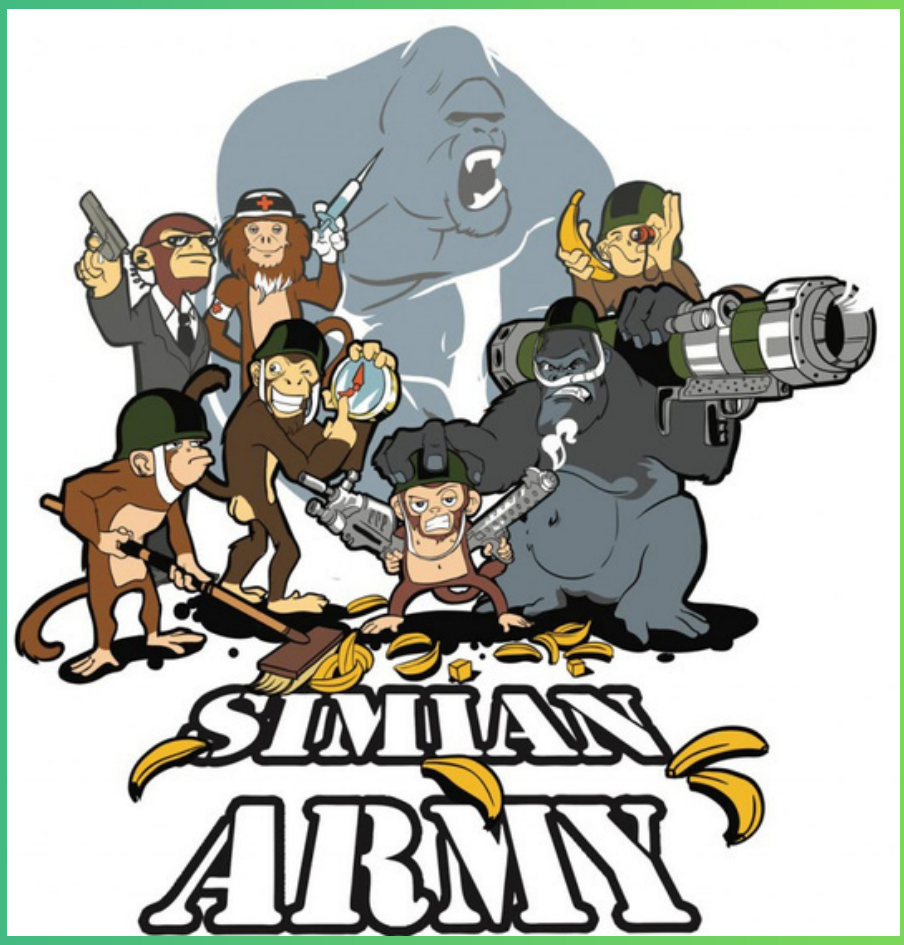
Engineers of Netflix set up best Infrastructure & DevOps practices to make their cloud infrastructure more safe, secure, available and automated.

Picture 1: Netflix’s Infrastructure

Chaos Monkey, The Simian Army, Latency Monkey and more to come!

Netflix created Chaos Monkey, a tool to constantly test its ability to survive unexpected outages without impacting the consumers. Chaos Monkey is a script that runs continuously in all Netflix environments, randomly killing production instances and services in the architecture.

Later, Netflix engineers wanted to test their resilience to all sorts of inevitable failures, detect abnormal conditions. So, they built the **Simian Army**, **a virtual army of tools** as Latency Monkey, Conformity Monkey, Doctor Monkey, Security Monkey, Chaos Gorilla, etc.



Simian Army incorporated DevOps principles of automation, quality assurance, and business needs prioritization.

The Result Was Impressive!

As a result, the tools from Simian Army helped Netflix develop the ability to deal with unexpected failures and minimize their impact on users.

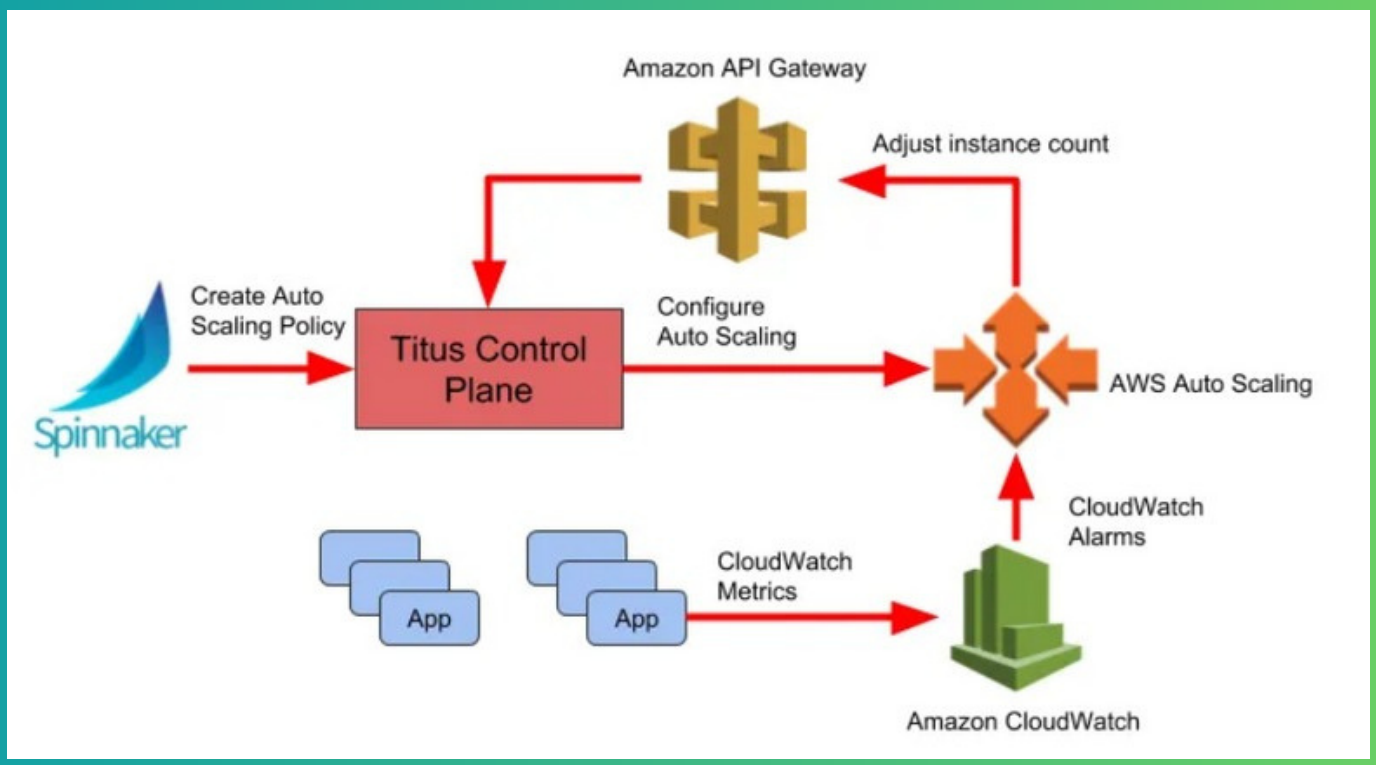
On 21st April 2011, AWS experienced a large outage in the US East region, but Netflix's streaming ran without any interruption.

And on 24th December 2012, AWS faced problems in Elastic Load Balancer(ELB) services, **but Netflix didn't experience an immediate blackout.** Netflix's website was up throughout the outage, supporting most of their services and streaming, although with higher latency on some devices.

Netflix’s Container Journey

Netflix had a cloud-native, microservices-driven VM architecture that was amazingly resilient, CI/CD enabled, and elastically scalable.

It was more reliable, with no SPoFs (single points of failure) and small manageable software components. Also, they adopted container technology being among the first who started using containers and seen tangible benefits. But they faced some challenges such as migrating to containers without refactoring, ensuring seamless connectivity between VMs and containers, and more. Netflix designed a container management platform called Titus, that enabled easy deployment of containerized batches and service applications.

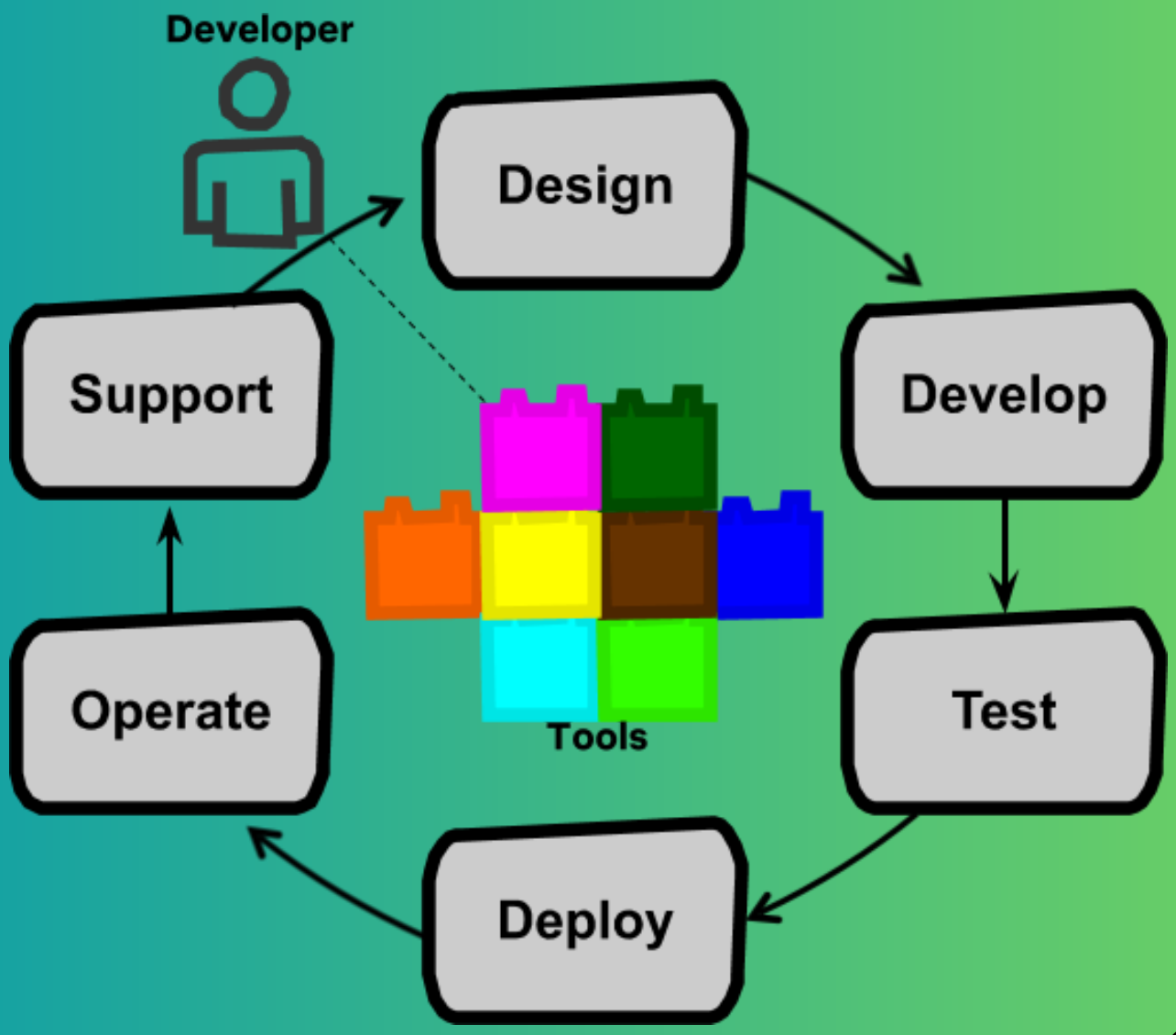


Picture 2: Titus auto scaling workflow

Operate What You Build

To deal with the above challenges and drawing inspiration from DevOps principles, Netflix encouraged **shared ownership of the full SDLC** and broke down silos.

The teams developing a system were responsible for operating and supporting it. Each team owned its own deployment issues, performance bugs, alerting gaps, capacity planning, partner support, and so on.



Picture 3: The empowered full cycle developer

Lessons We Can Learn from Netflix's DevOps Strategy

Netflix practices are unique to their work environment and needs and might not suit all organizations.

But here are a few **lessons to learn from their DevOps strategy**:

Don't build systems that say no to your developers

Netflix has no push schedules or push windows that developers must go through to push their code into production. Instead, every engineer at Netflix has full access to the production environment.

Focus on giving freedom and responsibility to the engineers

Netflix aims to hire intelligent people and provide them with the freedom to solve problems in their own way that they see as best. They hire people who can develop a balance of freedom and responsibility.

Don't think about uptime at all costs

Netflix servers their millions of users with a near-perfect uptime. But it didn't think about uptime when they started chaos testing their environment to deal with unexpected failure.

Lessons We Can Learn from Netflix’s DevOps Strategy

Eliminate a lot of processes and procedures

They limit an organization from moving fast. Instead, they hire people they can trust and have independent decision-making.

- **Don’t do a lot of required standards, but focus on enablement**

Teams at Netflix can work with their choice of programming languages, libraries, frameworks, or IDEs as they see best. In addition, they don’t have to go through any research or approval processes to rewrite a portion of the system.

Don’t do silos, walls, and fences

Netflix teams know where they fit in the ecosystem, their workings with other teams, dependents, and dependencies. There are no operational fences over which developers can throw the code for production.

Lessons We Can Learn from Netflix’s DevOps Strategy

Adopt “you build it, you run it” culture

Netflix focuses on making ownership easy. So it has the “operate what you build” culture but with the enablement idea that we learned about earlier.

Focus on data

Netflix is a data-driven, decision-driven company. It doesn’t do guesses or fall victim to gut instincts and traditional thinking.
It invests in algorithms and systems that combs enormous amounts of data quickly and notify when there’s an issue.

Always put customer satisfaction first

The end goal of DevOps is to make customer-driven and focus on enhancing the user experience with every release.

Don’t do DevOps, but focus on the culture

At Netflix, DevOps emerged as the wonderful result of their healthy culture, thinking and practices.

Netflix has been a gold standard in the DevOps world for years, but copy-pasting their culture might not work for every organization.

Learn from their DevOps culture and mold your processes and organizational structure to continuously improve the software quality and increase business value.

Like, comment or share this post...

Kapil Laad

DevOps & Cloud Engineer