



BHARTIYA VIDYA BHAVANS

SARDAR PATEL INSTITUTE OF TECHNOLOGY

Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai

Software Engineering Lab

Group Members:

Shreya Shetty (2019140059)

Shruti Shetty (2019140060)

Experiment : 10

Aim: Study on Continuous Integration using DevOp

Industry Selected for Case Study : Netflix

DevOps :

DevOps is a set of practices, tools, and a cultural philosophy that automate and integrate the processes between software development and IT teams. It emphasizes team empowerment, cross-team communication and collaboration, and technology automation. It aims to shorten the systems development life cycle and provide continuous delivery with high software quality. It increases an organization's ability to deliver applications and services at high velocity by allowing them to evolve and improve products at a faster pace than traditional software development models and processes which enables them to better serve their customers and compete more effectively in the market.

Advantages of DevOps :

1. *Faster Development* : DevOps allows for faster development with the help of shorter development cycles which results in faster results.
2. *Efficiency* : Use of development acceleration tools allow automation and a good flow of Continuous Integration/Continuous Delivery (CI/CD) pipeline, increasing efficiencies. It has also reduced the software development costs, making it cost efficient.
3. *Improved Quality and Reliability* : Practices like continuous integration and continuous delivery ensure changes are functional and safe, which improves the quality of a software product. Monitoring and logging helps teams keep informed of performance in real-time
4. *Enhanced Security Practices* : Most of the outsourcing companies deploy the DevSecOps model to integrate security with the development. Thus, the magnitude of external security risks decrease, making it more secure.
5. *Improved Collaboration* : Developers and operations teams collaborate closely, share many responsibilities, and combine their workflows. This reduces inefficiencies and saves time.
6. *Better Control of Risks and Recoveries* : With continuous monitoring, testing, and automation, there are less chances for risks to appear. Also, the recovery times are lesser, as there appears no major error.

Disadvantages of DevOps:

1. *Restructuring Work Culture* : DevOps is a complete framework, a work culture that cannot be simply implemented by following a Software Development Life Cycle (SDLC). It needs training and gradually turning into a DevOps environment. Work values need to be remodelled to let DevOps team in.
2. *Requires Software Engineering Expertise* : At Least a little bit of software engineering expertise is needed for DevOps.
3. *Demands Strong Teamwork* : DevOps demands teamwork and strengthened collaboration.
4. *DevOps May Not The Company's Problem* : If the organization only requires software upgrades occasionally, and the speed at which they arrive is not mission-critical, they may have no need to stress operations to implement DevOps.
5. *Speed and Security* : It is critical to the success of DevOps initiative that a separate plan for security at every stage of your DevOps workflow is carefully considered.

COMPARISON BETWEEN AGILE & DEVOPS :

<i>PARAMETER</i>	<i>AGILE</i>	<i>DEVOPS</i>
Definition	Agile refers to an iterative approach which focuses on collaboration, customer feedback, and small, rapid releases.	DevOps is considered a practice of bringing development and operations teams together.
Purpose, Focus & Task	Agile helps to manage complex projects. It focuses on functional and non-function readiness. Agile process focuses on constant changes.	DevOps central concept is to manage end-to-end engineering processes. It focuses more on operational and business readiness. DevOps focuses on constant testing and delivery.
Importance	Developing software is inherent to Agile.	Developing, testing and implementation all are equally important.
Implementation	Agile method can be implemented within a range of tactical frameworks like a sprint, safe and scrum.	The primary goal of DevOps is to focus on collaboration, so it doesn't have any commonly accepted framework.
Duration	Agile development is managed in units of "sprints." This time is much less than a month for each sprint.	DevOps strives for deadlines and benchmarks with major releases. The ideal goal is to deliver code to production DAILY or every few hours.
Target Areas	Software Development	End-to-end business solution and fast delivery.
Emphasis	Agile emphasizes software development methodology for developing software. When the software is developed and released, the agile team will not care what happens to it.	DevOps is all about taking software which is ready for release and deploying it in a reliable and secure manner.
Documentation	Agile method is to give priority to the working system over complete	In the DevOps, process documentation is foremost because it will send the software

documentation. It is ideal when you're flexible and responsive. However, it can hurt when you're trying to turn things over to another team for deployment.

to the operational team for deployment. Automation minimizes the impact of insufficient documentation. However, in the development of complex software, it's difficult to transfer all the knowledge required.

Automation	Agile doesn't emphasize on automation. Though it helps.	Automation is the primary goal of DevOps. It works on the principle to maximize efficiency when deploying software.
Goal	It addresses the gap between customer needs and development & testing teams.	It addresses the gap between development + testing and Ops.
Speed vs. Risk	Teams using Agile support rapid change, and a robust application structure.	In the DevOps method, the teams must make sure that the changes which are made to the architecture never develop a risk to the entire project.
Quality	Agile produces better applications suites with the desired requirements. It can easily adapt according to the changes made on time, during the project life.	DevOps, along with automation and early bug removal, contributes to creating better quality. Developers need to follow Coding and Architectural best practices to maintain quality standards.
Tools used	JIRA, Bugzilla, Kanboard are some popular Agile tools.	Puppet, Chef, TeamCity OpenStack, AWS are popular DevOps tools.
Challenges	The agile method needs teams to be more productive which is difficult to match every time.	DevOps process needs to development, testing and production environments to streamline work.
Advantage	Agile offers shorter development cycle and improved defect detection.	DevOps supports Agile's release cycle.

DEVOPS METHODS:

The following are a few common DevOps methods that organizations can use to speed and improve development and product releases:

1. *Scrum:*

Scrum is an agile development methodology used in the development of Software based on iterative and incremental processes. Scrum is an agile framework that is designed to deliver value to the customer throughout the development of the project. The primary objective of Scrum is to satisfy the customer's need through an environment of transparency in communication, collective responsibility and continuous progress.

2. *Kanban:*

Kanban is a popular framework used to implement agile and DevOps software development. Work items are represented visually on a kanban board, allowing team members to see the state of every piece of work at any time.

3. *Agile:*

Earlier agile software development methods continue to heavily influence DevOps practices and tools. Many DevOps methods, including Scrum and Kanban, incorporate elements of agile programming. Some agile practices are associated with greater responsiveness to changing needs and requirements, documenting requirements as user stories, performing daily standups, and incorporating continuous customer feedback.

DEVOPS PRACTICES:

1. *Version Control For All Production Artifacts*

Both Dev and Ops should use version control for everything. And they should share the same single source of truth.

2. *Continuous Integration and Deployment*

Check in code every day and check into the trunk every day, as opposed to hanging onto private code branches and integrating only at the end of the release.

3. Automated Acceptance Testing

Stop the line not only when the build breaks but also when something breaks. This is true for all software testing, whether it's an automated user test, an integration test, or a system test. This step keeps things in an always-deployable state.

4. Peer Review of Production Changes

Use peer reviews for better quality; leverage your team's familiarity, shared goals, and mutual accountability, as opposed to external change approval (such as a change advisory board).

5. High-Trust Culture

This is both a practice and an outcome resulting from a single source of truth, peer reviews, and shared goals.

6. Proactive Monitoring of the Production Environment

Monitor and communicate across the teams so everyone can see, understand, and affect end results and customer utilization.

7. Win-Win Relationship (and Outcomes) Between Dev and Ops

This approach counters the learned behavior that deployments hurt. By deploying code into production every day, you can change lives in Operations. Deployments don't have to be done at midnight on Friday with Ops working all weekend to get things running. When Ops employees are working the same hours as Dev, there is a sense of teamwork and joint accomplishment.

ABOUT NETFLIX :



Netflix, Inc. is an American subscription streaming service and production company. Launched on August 29, 1997, it offers a library of films and television series through distribution deals as well as its own productions, known as Netflix Originals. It is the world's leading internet television network,

with more than 200 million members in more than 190 countries enjoying 125 million hours of TV shows and movies each day. Netflix builds diversity, inclusion, equity, and a global outlook into everything it does, and by fostering a culture of courage, empathy, and curiosity, Netflix can move faster to develop new stories and better ways of sharing them with its members around the world.

Netflix uses AWS for nearly all its computing and storage needs, including databases, analytics, recommendation engines, video transcoding, and more - hundreds of functions that in total use more than 100,000 server instances on AWS. Netflix relies on AWS to help it innovate with speed and consistently deliver best-in-class entertainment. AWS provides Netflix with compute, storage, and infrastructure that allow the company to scale quickly, operate securely, and meet capacity needs anywhere in the world. Moreover, Netflix, a leading content producer, has used AWS to build a studio in the cloud. This virtual studio enables Netflix to engage top artistic talent, no matter the location, and Netflix artists and partners have the freedom to collaborate without technological or geographical barriers.



The Netflix logo, consisting of the word "NETFLIX" in a bold, white, sans-serif font with a black outline, set against a solid red background.

CONTINUOUS INTEGRATION :

Continuous integration (CI) is the practice of automating the integration of code changes from multiple contributors into a single software project. It's a primary DevOps best practice, allowing developers to frequently merge code changes into a central repository where builds and tests then run. Automated tools are used to assert the new code's correctness before integration.

A source code version control system is the crux of the CI process. The version control system is also supplemented with other checks like automated code quality tests, syntax style review tools, and more. Continuous delivery expands upon continuous integration by deploying all code changes to a testing environment and/or a production environment after the build stage.

NEED FOR CONTINUOUS INTEGRATION :

In the past, developers on a team might work in isolation for an extended period of time and only merge their changes to the master branch once their work was completed. This made merging code changes difficult and time-consuming, and also resulted in bugs accumulating for a long time without correction. These factors made it harder to deliver updates to customers quickly.

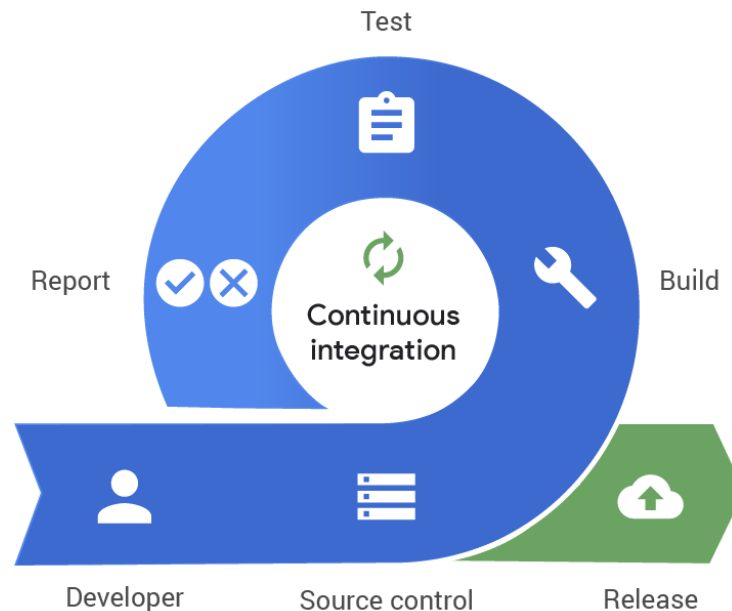
Continuous integration resulted in -

1. Improving Developer Productivity
2. Finding and Addressing Bugs Quicker
3. Faster Delivery Updates
4. Decrease in Code Review Time
5. Automatic Code Deployment

WORKING OF CONTINUOUS INTEGRATION :

1. Developers input code into their private terminals.
2. After that is done, they commit the changes to the shared repository.
3. The CI server monitors the repository and analyzes changes as they occur.
4. Continuous Integration builds the system and runs unit and integration tests.
5. The server also releases deployable artefacts for testing.
6. The CI server assigns a build tag to the version of the code it just built.
7. The Continuous Integration server gives the team reports of the successful build.

8. If the build or tests fail, the server alerts the development team.
9. The team will fix the issues as soon as possible.
10. It continues to integrate and run tests throughout the entire project.



NETFLIX'S USE OF CONTINUOUS INTEGRATION :

Netflix uses AWS CodeBuild for Continuous Integration. AWS CodeBuild is a fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy. Because CodeBuild is a managed service, you don't need to provision any resources such as build servers. As a start a build process, CodeBuild automatically allocates resources for you. Spinnaker is an open-source tool built by Netflix for continuous integration/continuous deployment (CI/CD).

NETFLIX BEFORE THE USE OF AWS CODEBUILD :

Netflix's initial business model included DVD sales and rental by mail, but Hastings abandoned the sales about a year after the company's founding to focus on the initial DVD rental business. Netflix expanded its business in 2007 with the introduction of streaming media while retaining the DVD and Blu-ray rental business.

All the way back in 2008 when the company experienced a major database corruption – an event that hindered their ability to deliver DVDs out to their customers. That's when the idea of a Cloud migration came about. Netflix needed a highly reliable and scalable platform like the Cloud to host their systems.

In the initial phase of Netflix, they were using a development practice called Monolith, which is a system where all the codes are written at once. The monolithic approach is very complex. Software gets bigger and bigger and it turns into a lot more complex coding. Hence, Netflix faced the problem of extremely complex code development.

As Netflix was grabbing more and more subscribers, their system was getting more and more complicated. Whenever a new bug was noticed, it took days to write the proper code and took even more days to go for operation. If the operation failed, again the whole process started to repeat at a similar pace. For a company that is growing rapidly, it is a huge concern as well as a risk to continue with the same approach. Due to such huge complications in the monolithic approach, Netflix started to move to DevOps.

NETFLIX AFTER THE USE OF AWS CODEBUILD :

Turning to DevOps was the key point that helped Netflix deal with such a high volume of subscribers, in a much simpler manner. Today most of the populations of the world are dependent on Netflix.

Moving to the cloud has allowed Netflix to keep its existing members well engaged with overall viewing growing exponentially. It is now the world's leading internet television network, with more than 100 million members in more than 190 countries enjoying 125 million hours of TV shows and movies each day.

Netflix was one of the very first companies to move their infrastructure over to AWS, and have since pioneered many of the common architectural patterns for designing products on AWS. Since moving the majority of their infrastructure to AWS, they have been able to design a system where it is cheaper to serve video as it continues to acquire more and more customers. This results in massive profits for the company. It is able to do this by leveraging AWS services without having to build and manage it themselves.

ADVANTAGES OF DEVOPS WRT NETFLIX :

The advantages of adopting this new approach are :

1. *Speed* : DevOps speeds up the release cycle by increasing the frequency of releases
2. *Efficiency* : DevOps seeks to automate workflows wherever possible
3. *Reliability* : DevOps ensures the quality of application updates and infrastructure changes so organizations can reliably deliver continuous updates while maintaining a positive experience for their customers.
4. *Improved collaboration* : Since, DevOps encourages communication and collaboration, it helps teams become more efficient by reducing inefficiencies.
5. *Improved quality* : It helped the product to have much better quality, due to continuous testing and rechecking.

Netflix's moving to DevOps, or Development + Operation system, merged both of their departments. Because of this, they can develop, check and deploy services together. Here, there is no need to wait for the development team to write another long code again which may take several days to complete. Afterward, it also eradicates repeating scenarios of failure, if any occurs during the deployment. As a result of DevOps, they were able to enhance the pace of the development operation.

DISADVANTAGES OF DEVOPS WRT NETFLIX & NETFLIX'S SOLUTION TO OVERCOME THE PROBLEM :

1. DevOps Security Challenges

A variety of technical and cultural factors impact application security. However, challenges for security in DevOps often stem from the conflict between the differing goals of developers and security teams.

- a. Developers aim to push their software through the pipeline as quickly as possible, while security teams emphasize the elimination of flaws, which can push back development.
- b. Security is often sacrificed for the sake of speed, allowing misconfigurations, unresolved vulnerabilities, and other flaws to remain and exposing the software to breaches or malfunctions.

- c. DevOps typically relies on cloud infrastructure, as well as open-source or immature tools. Some tools can dramatically increase productivity, but they can also carry potential risks for DevOps environments.

Netflix's Solution - Akamai :

Netflix uses Akamai , an American content delivery network, cybersecurity, and cloud service company, providing web and Internet security services. Akamai provides excellent online security for critical commerce applications or applications involving securing data of millions of users.

2. DevOps Service Down Challenge

Netflix software is impressively reliable, however occasionally the available streams of videos change. Sometimes, the 'Recommended Picks' stream may not appear, for example. When this happens it is because the service in AWS that serves the 'Recommended Picks' data is down. However, the Netflix application doesn't crash, it doesn't throw any errors, and it doesn't suffer from any degradation in performance. Netflix software merely omits the stream, or displays an alternate stream, with no hindered experience to the user--exhibiting ideal, elegant failure behavior.

Netflix's Solution - Chaos Monkey :

To achieve this result, Netflix dramatically altered their engineering process by introducing a tool called Chaos Monkey, the first in a series of tools collectively known as the Netflix Simian Army. Chaos Monkey is basically a script that runs continually in all Netflix environments, causing chaos by randomly shutting down server instances. Thus, while writing code, Netflix developers are constantly operating in an environment of unreliable services and unexpected outages. This chaos not only gives developers a unique opportunity to test their software in unexpected failure conditions, but incentivizes them to build fault-tolerant systems to make their day-to-day job as developers less frustrating.

This is DevOps at its finest: altering the development process and using automation to set up a system where the behavioral economics favors producing a desirable level of software quality. With the help of Chao monkey, Netflix had the following benefits -

1. Top class application development
2. High-resilience systems
3. Stable application system

AWS CODEBUILD :

AWS CodeBuild is a DevOps tool in the Continuous Integration category of a tech stack. It is a fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy. With CodeBuild, you don't need to provision, manage, and scale your own build servers. CodeBuild scales continuously and processes multiple builds concurrently, so your builds are not left waiting in a queue. You can get started quickly by using prepackaged build environments, or you can create custom build environments that use your own build tools. With CodeBuild, you are charged by the minute for the compute resources you use. CodeBuild can also create reports for unit, functional, or integration tests. These reports provide a visual view of how many test cases were run and how many passed or failed.

Advantages :

1. Fully managed build service
2. Continuous scaling
3. Extensible
4. Enables continuous integration and delivery
5. Pay as you go
6. Secure

Disadvantages :

Unlike Azure DevOps Services, which is a PaaS (Platform-as-a-Service), AWS is an IaaS (Infrastructure-as-a-Service) solution, so it's tightly linked to the underlying infrastructure. While deploying packages from Azure DevOps Services to another infrastructure such as AWS is possible, the reverse isn't. From AWS DevOps, deployment is possible only to the AWS infrastructure like EC2 (Elastic Compute Cloud) or S3 (Simple Storage Service).

Interface limitations, time to set up and no Self-hosted option are also drawbacks of AWS Codebuild.

SPINNAKER :

Spinnaker is used as a DevOps continuous delivery platform for releasing software changes. It provides application management and deployment to help release software changes with high velocity and confidence and it is open source. Created at Netflix, it has been battle-tested in production by hundreds of teams over millions of deployments. It combines a powerful and flexible pipeline management system with integrations to the major cloud providers.

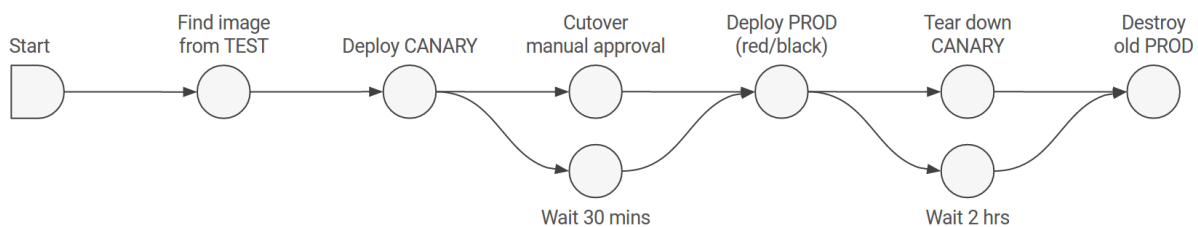
Originally, Netflix's engineering team built Spinnaker to help its internal teams manage their deployments. Spinnaker offers two core sets of features to help developers release software changes with "high velocity and confidence": (i) cluster management and (ii) deployment management.

Spinnaker supports AWS CodeBuild as a continuous integration system. Setting up AWS CodeBuild as a Continuous Integration (CI) system within Spinnaker allows to: trigger pipelines when an AWS CodeBuild build changes its phase or state, add an AWS CodeBuild stage to your pipeline.

STRUCTURE OF NETFLIX WITH DevOps

Spinnaker defines CI/CD workflows as pipelines. A pipeline consists of one or more stages. A stage defines part of the workflow. A pipeline is usually part of an application in Spinnaker.

Pipeline :



The pipeline is the key deployment management construct in Spinnaker. It consists of a sequence of actions, known as stages. You can pass parameters from stage to stage along the pipeline.

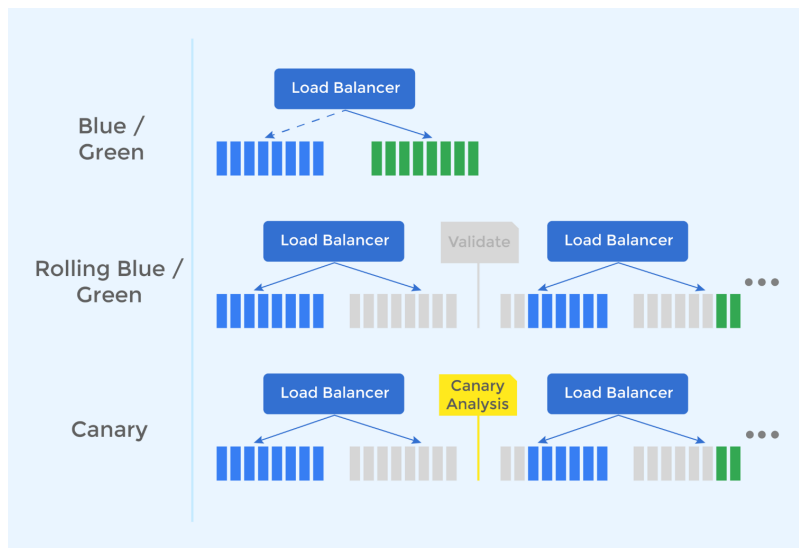
Stage :

A Stage in Spinnaker is a collection of sequential Tasks and composed Stages that describe a higher-level action the Pipeline will perform either linearly or in parallel. You can sequence stages in a Pipeline in any order, though some stage sequences may be more common than others. Spinnaker provides a number of stages such as Deploy, Resize, Disable, Manual Judgment, and many more. You can see the full list of stages and read about implementation details for each provider in the Reference section.

Task :

A Task in Spinnaker is an automatic function to perform.

Deployment strategies :



Spinnaker treats cloud-native deployment strategies as first class constructs, handling the underlying orchestration such as verifying health checks, disabling old server groups and enabling new server groups. Spinnaker supports the red/black (a.k.a. blue/green) strategy, with rolling red/black and canary strategies in active development.

CONCLUSION :

In this experiment we studied the implementation of DevOps in Netflix, its advantages, disadvantages, structure and also the various problems faced by Netflix to implement DevOps and how Netflix overcame these problems.