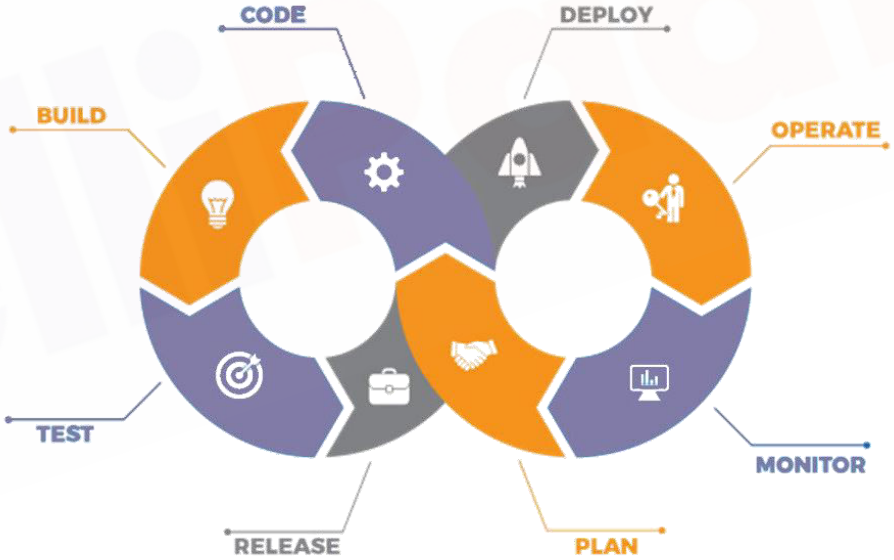
Introduction to DevOps



Agenda

01 WHAT IS SOFTWARE DEVELOPMENT?

02 WATERFALL MODEL

03 AGILE MODEL

04 LEAN MODEL

05 WATERFALL VS AGILE VS LEAN

06 WHY DEVOPS?

07 WHAT IS DEVOPS?

08 DEVOPS LIFECYCLE

09 DEVOPS TOOLS

What is Software Development?

What is Software Development?

Software Development is the process of transforming customer requirements into a complete software product.

Software Development Life Cycle

In broader terms, software development involves the following stages:

- Requirements
- Design
- Implementation
- Verification
- Maintenance

Software Development Life Cycle

- **Requirements**
- Design
- Implementation
- Verification
- Maintenance

This is the most important phase in the software development lifecycle. In this stage, the requirements are gathered from the customers and the requirements are then analysed to ensure their feasibility.

Software Development Life Cycle

- Requirements
- **Design**
- Implementation
- Verification
- Maintenance

Once the requirements are received, the architect transforms these requirements into technical specifications and plan the software components which have to be designed

Software Development Life Cycle

- Requirements
- Design
- **Implementation**
- Verification
- Maintenance

The specifications are then passed on to the developers which create the application based on these specifications

Software Development Life Cycle

- Requirements
- Design
- Implementation
- **Verification**
- Maintenance

Once the development work is done on the application. It is verified by a group of testers to map the application's functionalities with the specification given by customers

Software Development Life Cycle

- Requirements
- Design
- Implementation
- Verification
- **Maintenance**

Once the code is verified, it is pushed to production. Post this, the application is updated with any future enhancements or optimizations, if and when required.

SDLC Models

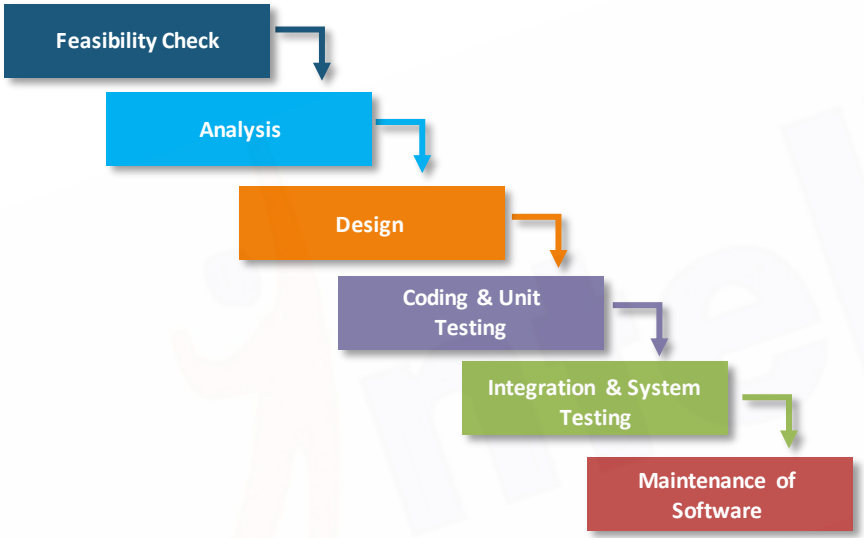
Since the time software development started, various software development models have been curated which implement SDLC. Each of these models solve problems that existed before these models were invented.

Traditionally, there have been 3 major software development models that most companies follow:

- **Waterfall Model**
- **Agile Model**
- **Lean Model**

Waterfall Model

Waterfall Model



- Waterfall Model was among the first development models which followed SDLC
- The Waterfall model follows a linear sequential model of development i.e until the first stage is not finished, the next stage will not start

Advantages of Waterfall Model

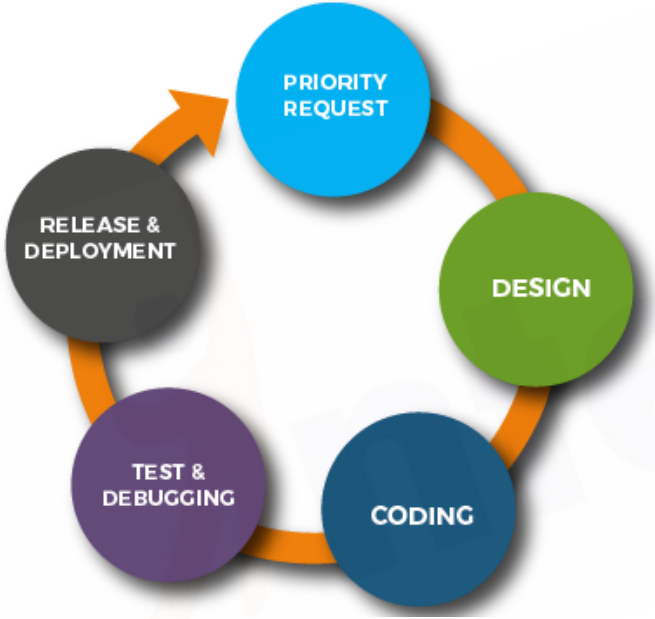
- Clear Objectives
- Specific Deadlines
- No ambiguous requirements
- Well understood milestones
- Process and results are well documented

Disadvantages of Waterfall Model

- Working Product is not available until the later stage in lifecycle
- Poor model for large and complex projects
- Cannot accommodate changing requirements
- High risk and uncertainty

Agile Model

Agile Model



- To overcome the challenges faced in the Waterfall Model, we came up with the Agile Methodology
- Agile Method believes in creating shorter development lifecycles
- Shorter Development Lifecycles are achieved by not releasing all the features at once by following an incremental model of development

Advantages of Agile Model

- Customer Satisfaction is high
- Less Planning Required
- Requirements can be dynamic in nature
- Functionality can be created and tested quickly

Disadvantages of Agile Model

- Not suitable for handling complex dependencies in projects
- Knowledge transfer to colleagues can be difficult since there is little documentation
- Success of the project depends heavily on customer interaction

Lean Model

Lean Model

7 Principles of Lean Methodology

- Eliminate Waste
- Amplify Learning
- Decide as late as possible
- Deliver as fast as possible
- Empower the team
- Build Integrity
- See the whole

- Lean development is a philosophy of increasing quality in software delivery by making use of agile methods
- Ignore the clutter for later and focus on what is required now
- Lean Methodology has it's primary focus on two things – Respect for frontline workers and Continuous Improvement

Advantages of Lean Model

- Carries the same advantages as Agile Methodology
- Creates a positive working environment
- Customer Feedback is given the utmost importance
- Limiting Wastes saves time and money

Disadvantages of Lean Model

- Largely dependent on the skill set of the team, therefore requires a strong team
- No room for error, a missed delivery can be bad for business
- Success of the project depends heavily on customer interaction

Waterfall vs Agile vs Lean

Waterfall vs Agile vs Lean

Summarizing

Problem with Waterfall Model was, the development lifecycle took a lot of time to complete. Therefore, by the time finished product was delivered, the customer requirements were no longer the same.

Customers

Software Company

Summarizing

This problem was fixed by Lean and Agile methodologies. These methodologies strictly focussed on customer feedback and improving the software quality that too in a shorter development lifecycle

Customers

Software Company

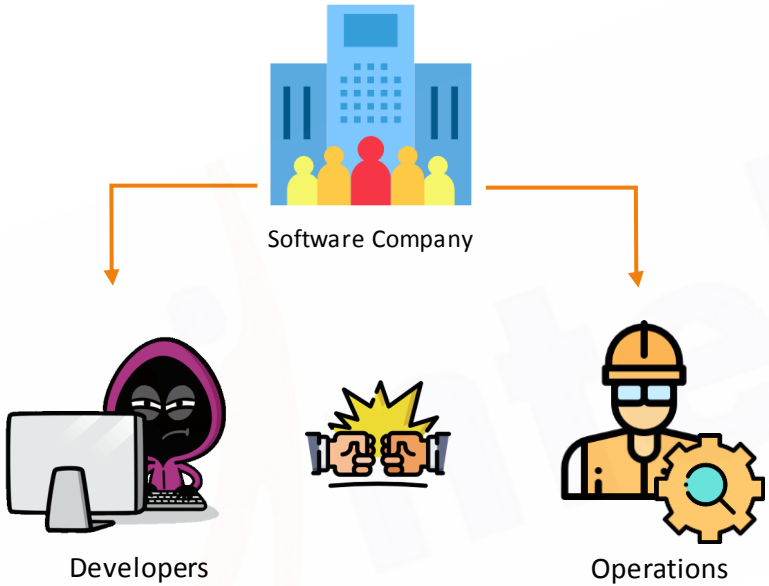
Summarizing

This problem was fixed by Lean and Agile methodologies. These methodologies strictly focussed on customer feedback and improving the software quality that too in a shorter development lifecycle

Why do we need DevOps?

Why DevOps?

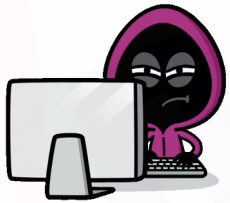
Why DevOps?



Although, the software quality was improved. We still had a lack of efficiency among the development team. A typical software development team consists of Developers and Operations employees. Let us understand their job roles

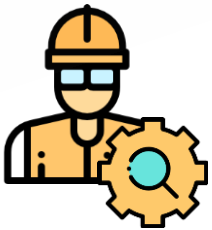
Why DevOps?

A developer's job is to develop applications and pass his code to the operations team



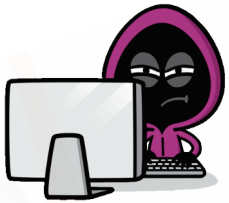
Developer

The operations team job is to test the code, and provide feedback to developers in case of bugs. If all goes well, the operations team uploads the code to the build servers



Operations

Why DevOps?



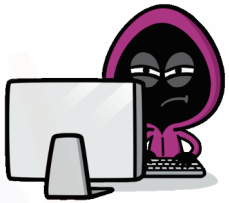
Developer

The developer used to run the code on his system, and then forward it to operations team.

Operations

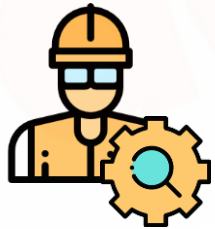
The operations when tried to run the code on their system, it did not run!

Why DevOps?



Developer

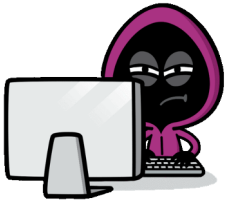
But, the code runs fine on the developer's system and hence he says "It is not my fault!"



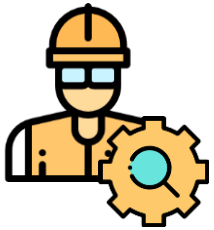
Operations

The operations then marked this code as faulty, and used to forward this feedback to the developer

Why DevOps?



Developer



Operations

This led to a lot of back and forth between the developer and the operations team, hence impacted efficiency.

Why DevOps?

Developer

Operations

This problem was solved using Devops!

Traditional IT vs DevOps

Traditional IT	Devops
Less Productive	More Productive
Skill Centric Team	Team is divided into specialized silos
More Time invested in planning	Smaller and Frequent releases lead to easy scheduling and less time in planning
Difficult to achieve target or goal	Frequent releases, with continuous feedback makes achieving targets easy

What is Devops?

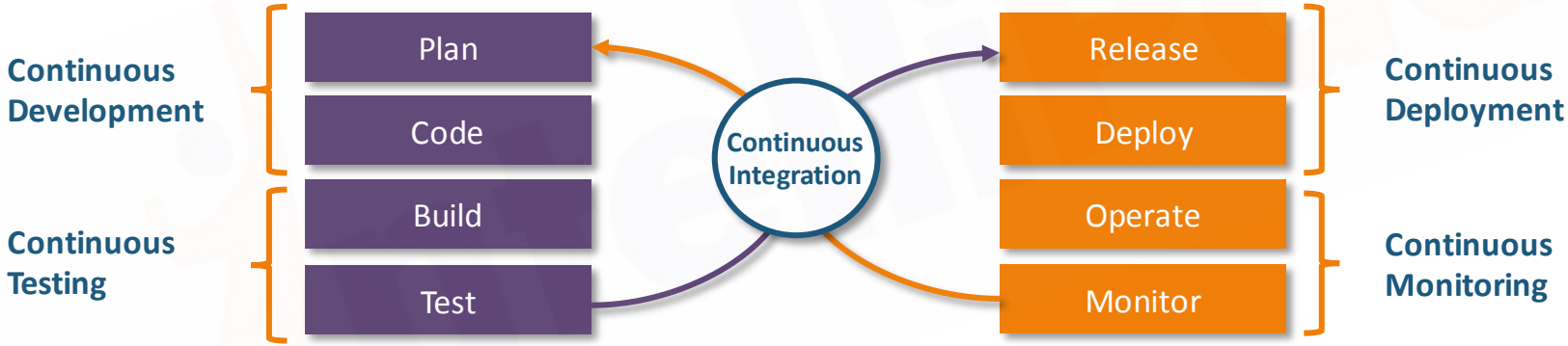
What is DevOps?

Devops is a software development methodology which improves the collaboration between developers and operations team using various automation tools. These automation tools are implemented using various stages which are a part of the Devops Lifecycle

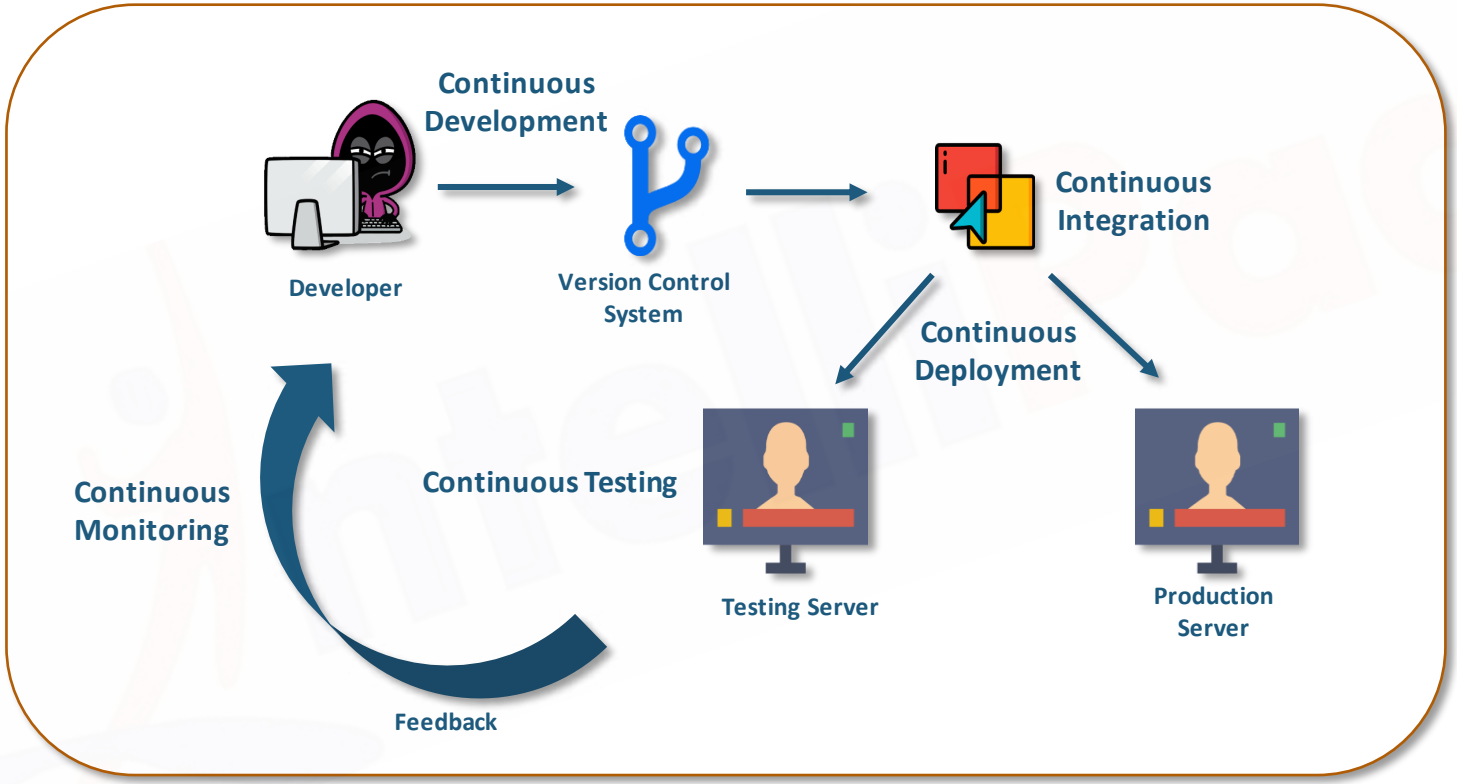
DevOps Lifecycle

How DevOps Works?

The Devops Lifecycle divides the SDLC lifecycle into the following stages:



How DevOps Works?



Automated CI/CD Pipeline

How DevOps Works?

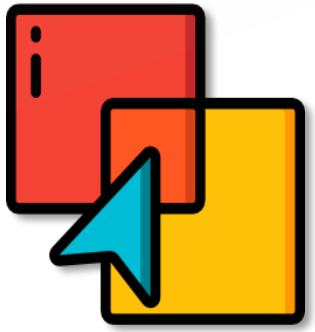
- **Continuous Development**
- Continuous Integration
- Continuous Deployment
- Continuous Testing
- Continuous Monitoring

This stage involves committing code to version control tools such as **Git** or **SVN** for maintaining the different versions of the code, and tools like **Ant**, **Maven**, **Gradle** for building/ packaging the code into an executable file that can be forwarded to the QAs for testing.

How DevOps Works?

- Continuous Development
- **Continuous Integration**
- Continuous Deployment
- Continuous Testing
- Continuous Monitoring

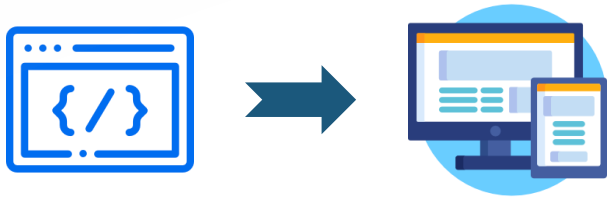
The stage is a critical point in the whole Devops Lifecycle. It deals with integrating the different stages of the devops lifecycle, and is therefore the key in automating the whole Devops Process



How DevOps Works?

- Continuous Development
- Continuous Integration
- **Continuous Deployment**
- Continuous Testing
- Continuous Monitoring

In this stage the code is built, the environment or the application is containerized and is pushed on to the desired server. The key processes in this stage are Configuration Management, Virtualization and Containerization



How DevOps Works?

- Continuous Development
- Continuous Integration
- Continuous Deployment
- **Continuous Testing**
- Continuous Monitoring

The stage deals with automated testing of the application pushed by the developer. If there is an error, the message is sent back to the integration tool, this tool in turn notifies the developer of the error. If the test was a success, the message is sent to Integration tool which pushes the build on the production server

How DevOps Works?

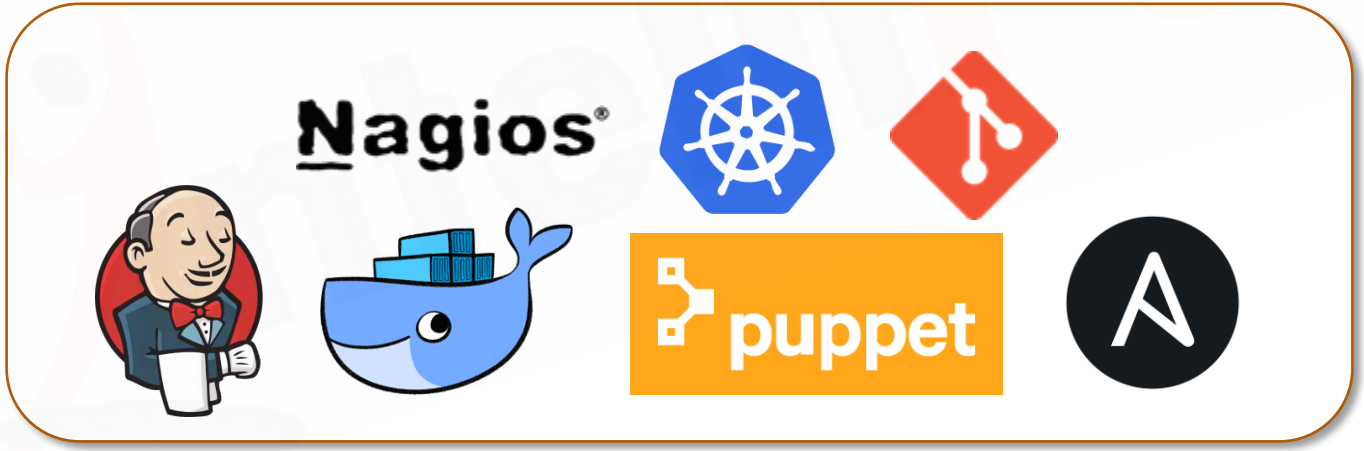
- Continuous Development
- Continuous Integration
- Continuous Deployment
- Continuous Testing
- **Continuous Monitoring**

The stage continuously monitors the deployed application for bugs or crashes. It can also be setup to collect user feedback. The collected data is then sent to the developers to improve the application

Devops Tools

DevOps Tools

We have discussed the Devops Methodology, but this methodology cannot be put into action without it's corresponding tools. Let us discuss the devops tools with their respective lifecycle stages



DevOps Tools

- Continuous Development
- Continuous Integration
- Continuous Deployment
- Continuous Testing
- Continuous Monitoring

Git is a distributed version-control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source-code management in software development, but it can be used to keep track of changes in any set of files

DevOps Tools

- Continuous Development
- **Continuous Integration**
- Continuous Deployment
- Continuous Testing
- Continuous Monitoring

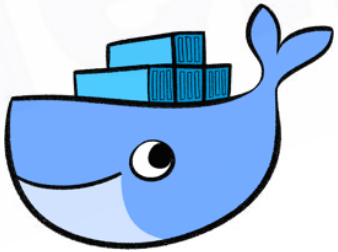
Jenkins is an open source automation server written in Java. Jenkins helps to automate the non-human part of the software development process, with continuous integration and facilitating technical aspects of continuous delivery

DevOps Tools

- Continuous Development
- Continuous Integration
- Continuous Deployment
- Continuous Testing
- Continuous Monitoring

Continuous Deployment

Virtualization & Containerization



Configuration Management

puppet

DevOps Tools

- Continuous Development
- Continuous Integration
- Continuous Deployment
- **Continuous Testing**
- Continuous Monitoring

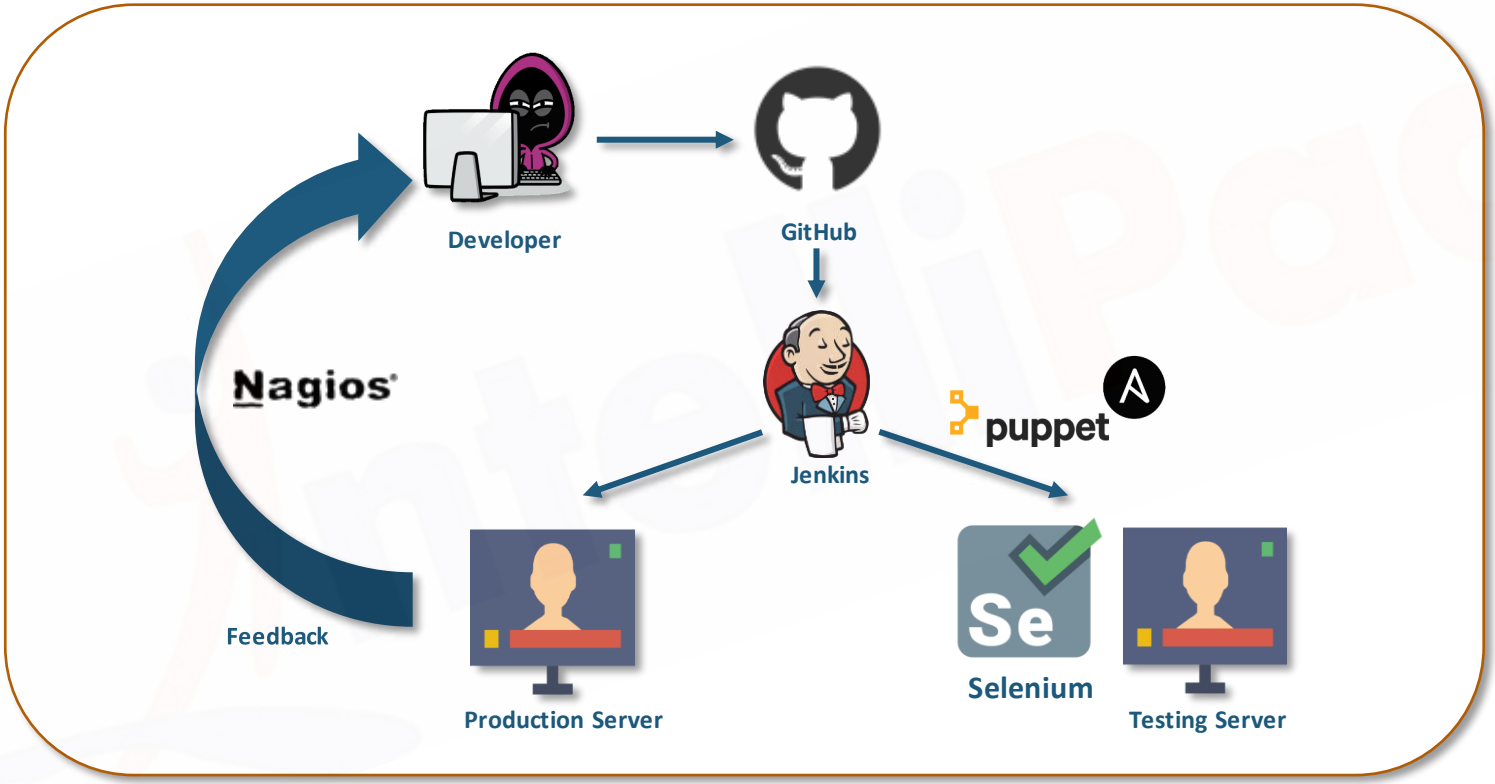
Selenium is a portable software-testing framework used for web applications. It is an open source tool which is used for automating the tests carried out on web browsers (Web applications are tested using any web browser).

DevOps Tools

- Continuous Development
- Continuous Integration
- Continuous Deployment
- Continuous Testing
- **Continuous Monitoring**

Nagios is an open-source devops tool which is used for monitoring systems, networks and infrastructure. It also offers monitoring and alerting services for any configurable event.

DevOps Tools



Quiz

Quiz

1. Which of these Software Development Methodologies are not suitable for large and complex projects?

A. Waterfall Model

B. Devops

C. Agile Methodology

D. None of these

Quiz

1. Which of these Software Development Methodologies are not suitable for large and complex projects?

A. Waterfall Model

B. Devops

C. Agile Methodology

D. None of these

Quiz

2. Devops Methodology was focused on solving the problems between the customers and the software company.

A. True

B. False

Quiz

2. Devops Methodology was focused on solving the problems between the customers and the software company.

A. True

B. False

Quiz

3. Which of these principles are NOT included in Agile Methodologies?

A. Frequent Release Cycles

B. Focus on Customer Feedback

C. Eliminating Waste

D. None of these

Quiz

3. Which of these principles are NOT included in Agile Methodologies?

A. Frequent Release Cycles

B. Focus on Customer Feedback

C. Eliminating Waste

D. None of these

Quiz

4. Which Lifecycle stage in Devops helps in Transition from one stage to another?

A. Continuous Development

B. Continuous Testing

C. Continuous Monitoring

D. Continuous Integration

Quiz

4. Which Lifecycle stage in Devops helps in Transition from one stage to another?

A. Continuous Development

B. Continuous Testing

C. Continuous Monitoring

D. Continuous Integration

Quiz

5. Which tool among the following helps in containerization?

A. Jenkins

B. Git

C. Kubernetes

D. Docker

Quiz

5. Which tool among the following helps in containerization?

A. Jenkins

B. Git

C. Kubernetes

D. Docker

IntelliPaat

India : +91-7847955955

US : 1-800-216-8930 (TOLL FREE)

support@intellipaat.com

24X7 Chat with our Course Advisor