1. Introduction

1) Why Learn DevOps?

- Top skills employers looking in IT industry.
- Market almost $12.5B.
- A corporate term becauase it have direct impact on business merit.
- One of the Highly paid job.
- All companies making software delivery better constantly and continously at an enourmous phase.

2) What exactly is Devops?

- Some say its all about automation/Scripting
- Cloud Computing
- Tools and Culture
- Skills

Software Development Process

1. Requirement (Gathering and Analysis)
   a) Product Features
   b) Users
   c) Usage
   d) User Requirement
   e) Market State
2. Planning (What do we want?)
   a) Cost
   b) Resources
   c) Risk
3. Design Architects
   1. Based on detailed requirements system design documents are created
4. Development
   1. Developers
   2. Software Development Based on Inputs of design documents
5. Testing
   1. Quality Assurance
   2. identify the defects to ensure the quality product is good.
6. Deployment
   1. System Admins

    2. Operation works in deploying the product.
7. Maintenance
    1. Changes and Uptime
    2. System health
    3. Perfomance Uptime
    4. Regular changes

Overview

```
Requirement Gathering
Planning
Designing
Development
Testing
Deploy and Maintain
```

This is software Development life cycle also known as SDLC.

There are different models in SDLC:

- waterfall
- agile
- spiral
- bigbang
- etc,etc

In waterfall model each phase must be completed before another begins. It is difficult to go back to accommodate changing requirements, Also No working software is produced intil late during the life cycle.

In Agile every phases are kept in a cycle and broken into iterations of time cycle. every time the cycle finishes the next cycle begins in which necessity requirements or any other extra things or new ideas can be added onto it. This helps on producing the software in every cycle and helps to have a demo of the product every time the cycle repeats.

Even though agile is fast there may occur certain problems when it is taken into operation. Such as:

- Regular Deploy Request
- No clear instructions
- already occupied
- system uptime
- ITIL process Driven

Dev and Ops are both necessary.
Dev is Agile in all about regular and quick changes are made
Ops is ITIL who writes stable environment for the product.

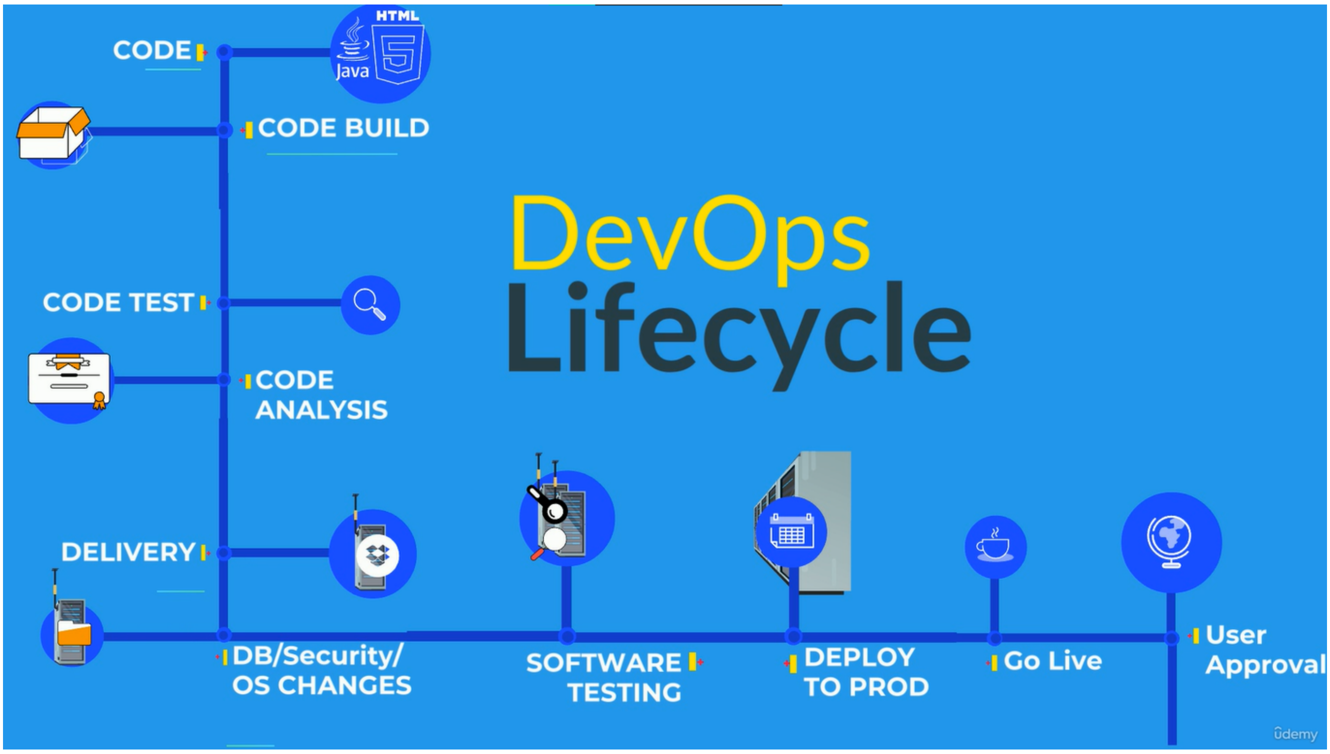
Automation

- Everything code build
- Code testing
- software testing
- infra changes
- deployments
- everything

DevOps Life cycle

- Code
- code build
- code test
- code analysis
- delivery
- DB/Sec Changes
- Software testing
- Deploy to product
- Go live
- User approval

- Keep monitoring



With automation comes integration because it doesnot produce human errors which brings error on the cycle and process.

Continuous Automation

Continuous Automation is an automated process in devops which generates software and its features quickly and efficiently.

While working in a team codes are needed to be stored in a centralised place
This centralised repo is called Version Control System like Github.

Codes will be moved to a Build server on which it will build, tested and evaluated which generates the software or also called artifact. It is stored in a software repository.

artifacts are an archieve of files generated from the build process. Based on the programming language these are packaged in different format. Like ZIP/TAR/WAR/JAR/DLL etc.

Problems
after 3 weeks of work of code when fetching the code to build server various error occurs.
Codes get merged into repo but it doesnt get integrated.

Solution
After every single commit from the developers the code should be built and tested. But this is not humanly possible to do several times a day by which the best solution for this problem would be **Automation**

By this everytime the developer commits the automation will build and test the code and then send a notifcation if there are any errors by which the deveoper can work on that error on the exact same time and commit it again.

In a cycle view:

- code
- fetch
- build
- test
- notify
- feedback

This automation process is call Continuos Integration or C.I in short
The Goal of C,I is to detect defects in very early stages so it doesnt multiply.

IDE is used by developers for coding

- Eclipse
- Visual Studio
- Atom
- Pycharm
- And manymore

These **IDE** is integrated with **Version Control system** is used to store and version the code.

- GIT
- SVN
- TFS
- PERFORCE
- ETC ETC

Build tools based on the programming language.

- MAVEN, ANT, GRADLE
- MSBUILD, VISUAL BUILD
- IBM URBAN CODE
- MAKE
- GRUNT

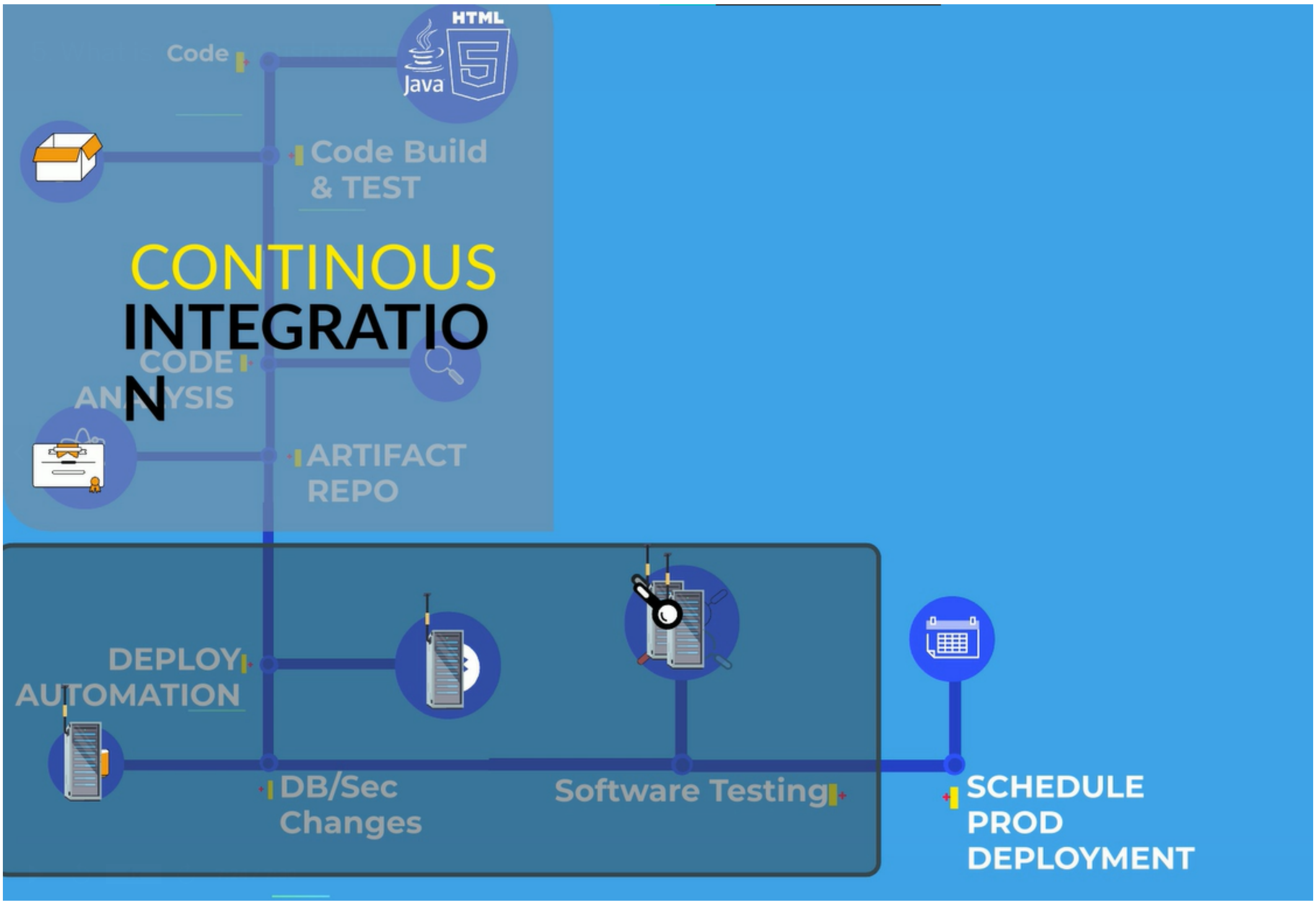
Software Repository to store artifacts (Softwares)

- SONATYPE NEXUS
- JFROG ARTIFACTORY

- ARCHIVA
- CLOUDSMITH PACKAGE
- GRUNT

Continous Integration Tools that integrates everything

- JENKINS
- CIRCLECI
- TEAMCITY
- BAMBOO CI
- CRUISE CONTROL



Continuous Delivery

It is an automated process of delivery code changes to server quickly efficiently at an enourmous pace.

It is the extension of C.I (Continuous Integration)

Deployment

- server provisioning
- dependencies
- configuration changes

- network
- artifact deploy
- many more things

in short automated deployment

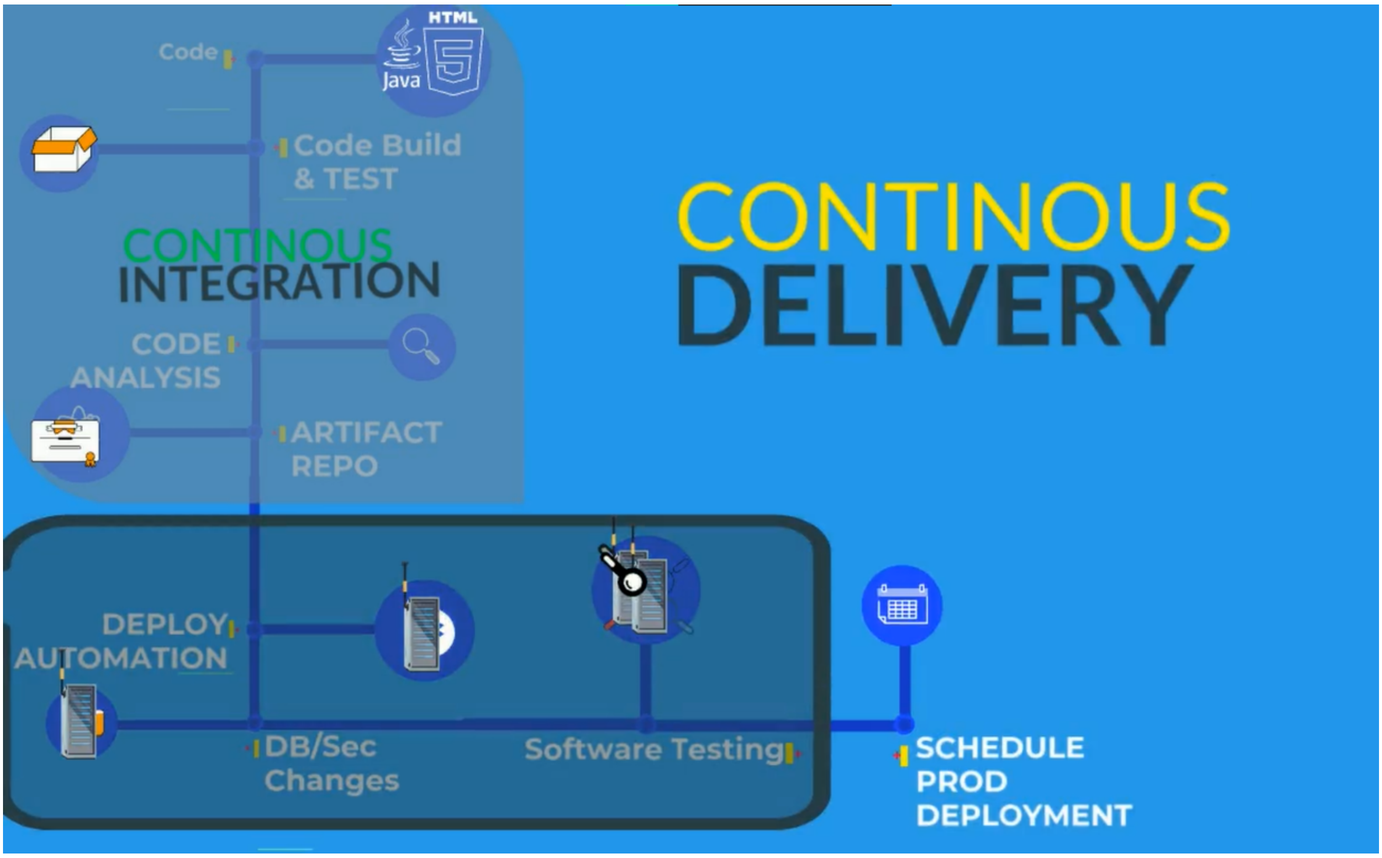
some of the tools are:

- ansible, puppet, chef - system automation
- terraform, cformation - cloud infrastructure automation
- jenkins, octopus deploy - CICD automation
- HELM charts
- Code Deploy

Test Automation

- Functional
- Load
- Performance
- Database
- Security
- Any other test

Dev Team - Integrate CI
Ops team - Automate Deployment
QA Team - Automation Testing



CONTINOUS DELIVERY

Code

HTML

Java

Code Build & TEST

CONTINOUS INTEGRATION

CODE ANALYSIS

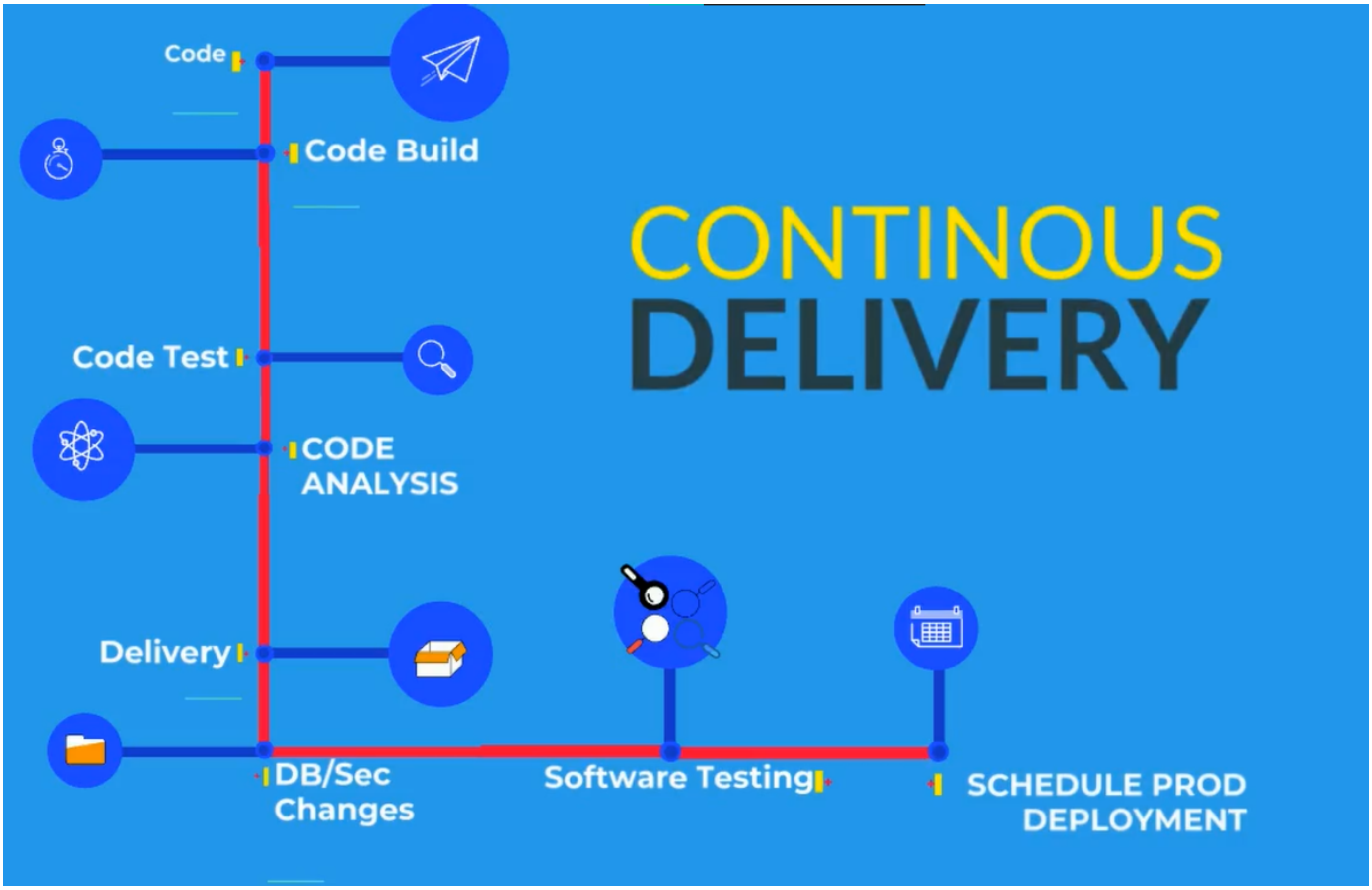
ARTIFACT REPO

DEPLOY AUTOMATION

DB/Sec Changes

Software Testing

SCHEDULE PROD DEPLOYMENT



CONTINOUS DELIVERY

Code

Code Build

Code Test

CODE ANALYSIS

Delivery

DB/Sec Changes

Software Testing

SCHEDULE PROD DEPLOYMENT