**CHAPTER-1**

**INTRODUCTION**

**1.1 INTRODUCTION TO AUTOMATION**

Today in IT sector there is a very tough competition in every domain. None of the company wants to lag behind with any respect to any domain. Say its time, cost or effort. Each day a new problem comes up for the customer and the need for immediate solutions has posed enormous challenges to the IT professionals who have to come up with good solutions to the customers problems.

There is a range of choice for the customers to choose the right vendor who can solve their problems and fulfil their requirements in the most efficient and cost-effective way.

The customer is in search of a vendor who can deliver good solutions in quick time. Hence the concept of automation came up and the companies have tried implementing the concept of automation to fasten their software production process and to speed up the process of delivery to the customer.

A company which delivers faster and good products is always preferred over any other company. The development stage of a product is just one phase of any project and the processes that follow the development stage are equally important to make sure that the product that we have developed is stable, correct and caters to the needs of the customer in the way it is intended to do so. If the processes like software, building, testing, packaging, deployment, delivery and maintenance are automated then a lot of time can be saved and the process of delivering the product becomes faster. Customers always come up with changes in their requirements and the automated system makes it easier for the IT professionals to make the changes in the system that the customer requires.

Setting up an automated system is a big task and as the system gets a considerable amount of shape the problem gets even bigger when integration of different components needs to be done. Automating the processes of software, building, testing, packaging, deployment, delivery and maintenance are very big aspects and there is a need to follow a systematic approach to achieve the process of automating such a big collection of components. Each processes have multiple tools and multiple services running that all need to be integrated and be in sync to achieve completion of the process and to deliver an effective and efficient system that can be trusted upon to deliver rapid and good service to the customer.

If we have to look at it from the aspect of the resources that are needed to complete any software project. We can say that a lot of human resource is required when it comes to software testing, deployment, delivery and maintenance. The cost and time taken to marshal these human resources in enormous. Whereas in the concept of automation of the software delivery process these resources will get cut down in size and the aspect of human interaction will get reduced. Hence the overhead of marshalling and keeping the human resources occupied can be written off. The machine in itself takes care of the processes that were once possible only with the help of direct human interaction.

Automation saves lots of time in the race to faster delivery of software products. Moreover the concept of automation will help the IT professionals involved to deliver faster and in a more efficient way. Automated testing can help them save time and we can expect faster results of the test which in turn will help the software professionals in proceeding with the next task.

**1.2 PROBLEM STATEMENT**

In traditional approach of software development process all the installations of the tools and services required for the development of the product are done manually. The tools, services, hardware components like switches, routers, etc need to be installed and configured manually. This is a time consuming job and if the number of components are more then it takes a lot of manual intervention and time to complete the task. A lot of dependent software’s and tools need to be installed to carry out the development process. And all these installations and configurations will take a lot of time and hence the delivery of the product will be delayed. Hence a system needs to be developed that automatically installs and configures the required tools and services on a large number of servers.

The bundled code packages are stored in an artifactory and these packages need to be automatically deployed onto the server. To accomplish this process, scripts need to be written to automate all the steps that are involved in the deployment process.

**1.3 OBJECTIVES OF THE PROPOSED SYSTEM**

* Installation and setup of DevOps tools.
* Integration of multiple tools within the DevOps chain.
* Deployment Automation of the DevOps tools.
* Automation of installation and configuration of tools.

**1.4 SCOPE OF THE SYSTEM**

This project aims to maximize the predictability, efficiency, and maintainability of operational processes. DevOps describes techniques for automating repetitive tasks within the software development life-cycle (SDLC), such as software builds, testing, and deployments, allowing these tasks to occur more naturally and frequently throughout the SDLC.

DevOps provides continuous integration, continuous deployment, continuous delivery and continuous testing of the software.

**1.4.1 Continuous Integration**

The advantages of Continuous Integration model are,

* Provide continuous compiling, building, packaging of the software.
* On Each check-in, the system triggers the compilation process, runs the unit test and runs the analysis tools to review the build system.
* Developers can provide tagging for each new version of software build.
* The more stable software with less known issues would be deployed to the users.

**1.4.2 Continuous Deployment**

The advantages of Continuous Deployment are,

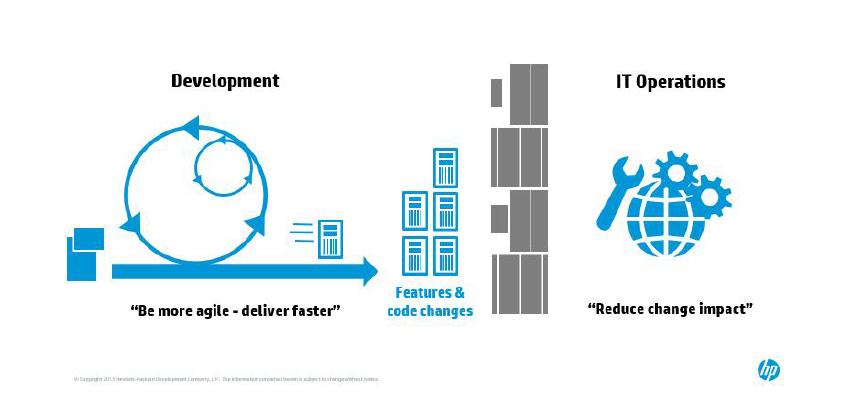
Software Updates are made available more frequently.

* When issues are identified, it will be rectified as soon as possible and new build is deployed after testing.
* Continuous Deployment reduces the cost with the time of software production.

**1.4.3 Continuous Testing**

Continuous Testing enables to achieve the following,

* Testing is executed on every latest software build
* Continuous Testing provides the real-time status of the quality of the software.
* The issues in the software can be identified as early as possible.



1.4 Development Life cycle

**1.5 CONCEPT OF DEVOPS**

In order to achieve the enormous task of implementing the automation process of software production, certain set of practices need to be adapted and incorporated that will help to achieve the goal of automation in a more effective and efficient way. Hence the DEVOPS model is incorporated in the proposed system to make sure that there is a pattern or model that helps to get the desired results.

DevOps is a combination of software Development and information technology Operations. It is used to refer to a set of practices that emphasize the collaboration and communication of both software developers and information technology (IT) professionals while automating the process of software delivery and infrastructure changes. It establishes a culture where development, testing, software release can happen rapidly, often and in a reliable and controlled way and both the Developers and the Operations team work together. They have to be in tandem to make sure that the operations go on smoothly and that the IT professionals know how the code works with the help of the developers.

The key advantage of adopting DevOps in the current business trend is business agility. As the rate of change for business accelerates, companies are less able to predict where business is heading. The top strategic imperative becomes responding rapidly via agility and modularity through DevOps and adaptive IT. When it is unknown what the future holds, it’s important to have the infrastructure in place that can respond to business needs on the fly. When you use the DevOps agile methodology, IT works directly with the business users and delivers exactly what they need and nothing more. By focusing on business needs first the projects get finished and move into production faster.

DevOps allows companies to deliver software that is critical to the production, much faster sometimes 10x to 100x faster. For any business that captures advantage through any type of software, this kind of improvement could mean success or failure. In future, when more companies are successfully deploying DevOps, there will no longer be any room for failure to adopt DevOps. While there are other advantages to DevOps, a 10x to 100x speed advantage is the only one that counts.

If DevOps is adopted properly by the organization, IT’s role changes from being a business cost centre to being that of one that leads the innovation for the business. Product and service offerings can be introduced, tested, and rolled out much more quickly to the market place keeping up with the pace of change in the market place. The principles of DevOps allow IT teams to produce and deliver value quickly, test out hypotheses with real users, and roll out actual services and products that bring in revenue to help grow the business and make an impact to the top line.

**1.6 DEVOPS CHAIN**

The devops chain is a collection of tools that are part of devops model because the IT professionals use these tools to do the required tasks.

The developers develop the code as well as work around with these tools in their projects. Hence the devops tools are an essential part of the devops model.

DevOps is a cultural shift and collaboration (between development, operations and testing), there is no single ”DevOps tool”. It is rather a set of tools or a chain called as ”DevOps toolchain”, consisting of multiple tools. Generally, DevOps tools fit into one or more of these categories, which is reflective of key aspects of the software development and delivery process. The following are the types of DevOps tools that can be integrated in the devops chain.

* Code - Code development and code review, version control tools and
* code merging
* Build - Continuous integration tools, shows the build status
* Test - Regular Tests and results determine the performance
* Package - Artifact repository and application pre-deployment staging
* Release - Change management, release approvals, release automation
* Configure - Infrastructure configuration and its proper management, Infrastructure as Code tools.
* Monitor - Applications performance monitoring, end user experience Automated testing saves a lot of time and we can complete the tasks related to the project quickly. The use of monitoring tools helps us to keep track of all the components in the system and we can easily troubleshoot and find out solutions to problems with the help of these monitoring tools. The build tools will automatically generate the pom.xml file and we need not do the build process by ourselves and hence this process too saves a lot of cost and time.

**1.7 SITE RELIABILITY ENGINEERING**

Site reliability engineering (SRE) is a discipline that incorporates the aspects of software engineering and applies that to operations whose goals are to create ultra-scalable and highly reliable software systems. Site Reliability Engineering was created at Google around 2003 by Ben Treynor. A team of google engineers was given the big task to make Google’s sites run smoothly, efficiently and in a more reliable manner. Google’s large-scale systems required the company to come up with new techniques to manage large systems that have never existed before and at the same time introduce new features in a more continuous manner but also incorporating a very high-quality end user experience.

The word site in this aspect refers to a system which is composed of a collection or a cluster of servers, networking devices, storage devices, set of tools installed on them and the different types of services that are running on the systems to achieve different purposes. A site reliability engineer (SRE) will ideally spend most of his time doing ”ops” related work such as issues, on-call, and manual intervention. Since the software system that an Dev-ops office model handles is expected to be highly automatic and self-healing, the SRE should also spend their time on development tasks such as new features, scaling or automation.

The ideal Devops office model is a coder who has operational and systems knowledge. DevOps plays the main role here and it is a superset of SRE. It refers to a set of practices that enables the collaboration of Software developers and Information technology professionals in a joint environment. The main tasks of a devops office model is to create highly scalable and reliable systems. The SR Engineer has the responsibility of automating the integration process of all the various tool and services running in the system and deployment of all the software services into the live environment. In the wake of emergency the devops model will manually intervene and make specific changes that are required in the system to bring back a state of normalcy in the system and to achieve the goal of high availability to keep the customer happy.

Any new upgrade or launch needs to be scrutinized to evaluate the amount and the type of resources that the proposed system will absorb and what will be the impact on the hardware resources. Hence the devops model are IT professionals who develop the product as well as carry out the operations related to the product. Because they will understand the system better since they have built it and will be managing the related operations.

**1.8 INTRODUCTION TO AGILE METHODOLOGY**

Agile software development describes a set of principles for software development where requirements and solutions evolve through the collaborative work of different teams. It encourages the adaptive planning, evolutionary development, early delivery, and continuous improvement, and it encourages rapid and flexible response to change. The engineers follow the concepts of continuous integration and continuous deployment to deliver the correct system.

In the agile model the requirements can change at any time and the changes to the system need to be made immediately after due diligence. The changes are made rapidly as soon as the requirements are changed. The system needs to be integrated continuously and the deployment process also takes place continuously. Agile speeds up the process of product delivery. At the end of each cycle of development in this methodology there is a fully functional working module that can be deployed. The use of agile has been extended to this automation project and will be instrumental in the entire process.

**1.9 OVERVIEW OF THE PROJECT**

The project aims at automation of the processes involving software building, testing, deployment and monitoring. It initially involves all the manual installations and setup of multiple DevOps tools that range across different aspects such as build tools, testing tools, configuration tools, deployment tools, packaging tools, provisioning tools and monitoring tools. All the installations are done manually in the beginning.

Then the integration phase comes where multiple tools are integrated within the DevOps chain using a tool called Jenkins. The system requires the integration of a lot of tools across different domains like development, Software building, testing, packaging, deployment, delivery and monitoring tools. The Deployment Automation of the DevOps tools is carried out by using a tool called Puppet. This tool automates the installation of multiple tools and sets up the system automatically within seconds for any live environment. Hypervisor is used for the bare metal provisioning and management of the servers. It does the provisioning of servers (nodes) to the services.

Finally the proposed system will be used to automatically build the code, test the code, package the application, deploy the application and monitor it regularly.