Case Studies

edureka!

Problem Statement 1

FinTarget has grown into a big successful organization, and now multiple teams are involved in the delivery cycle. Also, there are hundreds of builds which need to be build every day, so there is a need to automate the build, tests, and deployment process by implementing pipeline. Since it is a critical project, the management team wants to be updated on the build status.

Your manager has given you the responsibility to create an end-to-end CI/CD pipeline for the development team, which will help in faster production. Development teams need CI/CD pipeline comprising of below three important segments:

1. Automated CI job execution form MAVEN build and test cases execution. You need to work on Webhook integration with Jenkins.
2. Perform parallel build with different labels for development and production environment.
3. Sending proper email notifications using Jenkins's pipeline plugin to notify end users.

Considering all these basic requirements, you need to implement this automation so that developers only perform commit and whole automation should trigger automatically. Work with the development team to resolve all build and deployment-related issues.

Step by step guide:

1. Implement Webhook integration between Jenkins and GitHub for automatic build execution.
2. Test build automation with the help of build tools like maven to understand the build process.
3. Work on setting up the initial CI job with the automated trigger.
4. Create parallel build jobs for development and production with labels.
5. Implement an email plugin in Jenkins's pipeline to send notifications post pipeline completion.

Problem Statement – 2

FinTarget has become a leader in the finance category. Financial data is extremely sensitive and private data. So, the management wants to make sure that the data is safe completely and there are less bugs in the system. You are wondering how to give the confidence to the management although the team is performing validation and testing on application source code but still, in the past they have struggled in getting quality source code in production environment. While working as a DevOps Engineer you decided to implement code scan tool in the code to measure and improve the code quality. This shall provide the current level of code coverage which team is able to cover via different test cases. This will provide enough stats and information to the development team to cover the gaps. Also, once code coverage is increased, there will be less chance of getting bugs from the field. As you already have CI/CD pipeline implemented in FinTarget, you need to integrate the code coverage tools with Jenkins. You also need to make sure that all the stakeholders are getting notification for code coverage with each build.

Step by step guide:

1. Integrate Gradle in your Jenkins pipeline and test the code build.
2. Implement plugin into Jenkins to start with integration.
3. Integrate code coverage tool with your application source code.
4. Publish code coverage report either in Jenkins or email so that development team can check.

Problem Statement - 3

You are now working as a Lead DevOps Engineer and responsible for managing a team of developers at FinTarget. They are in the process of implementing modern CI/CD pipelines implementation on Jenkins. While doing this onboarding, one of the important issues you will face is working with authentication and authorization. You must implement certain roles and user management to differentiate between developers on what kind of access they would have on Jenkins. Also, as the team is big now, you want to monitor the Jenkins for specific parameters like CPU usage, memory usage etc. Also, as you are interfacing with the management team, you must showcase the results with automated generated reports. Once these reports are generated, you found out that the build results not very encouraging. Hence one of the actions taken by you was to induce static code analysis using Jenkins with any of the tools - like PMD, CheckStyle, FindBugs. You need to present this data, so you want to have all the details like warnings, warning per build, severity-wise distribution of error/warning. All these reports and analysis are critical, so you would like to configure Jenkins for backup and recovery as a proactive measure.

Step by step guide:

1. Manage developers on Jenkins's instance.
2. Implement roles management plugin on Jenkins.
3. Create respective roles for different users.
4. Now assign respective roles to developers for better access management.
5. Configure roles for different levels of access on Jenkins's job.
6. Enforce job restrictions to ensure that only authorized person will access specific Jenkins resources.
7. Configure Jenkins for monitoring.
8. Generate various graphs and reports to show the build status.
9. Configure static analysis code tool with Jenkins.
10. Generate analyser report.
11. Configure Jenkins for backup and recovery with backup data and make sure no data is lost.

Problem Statement - 4

As a Senior DevOps Engineer, you have successfully deployed CI/CD pipeline for FinTarget's finance application. The development team has started using CI/CD pipeline to create finance application JAVA code builds. The management team decided on the Finance application launch date, and to meet the deadlines development team has started working with full capacity. Due to this, Jenkins single master instance started getting more build request. Now, this standalone Jenkins instance is not capable of handling this increasing load. Also, the current CI/CD pipeline setup lacks the capability to send build alerts. The development team has raised the issue with your Project manager. Your manager has set up a quick meeting with you to discuss the problem solution. So, you have been given the task of implementing distributed build mechanism with build agents to share build load. Also, you need to implement Slack integration for sending build alerts to slack.

Step by step guide:

1. Adding a new slave machine to Jenkins's instance.
2. Allocating Jenkins job to these new slave machines on Jenkins
3. Integrate GITHUB with Jenkins pipeline to trigger the build based on source code change.
4. Setting up Slack webhook for integrating with Jenkins's notifications.
5. Configuring Slack notification with Jenkins to send build alerts while running builds.
6. Run the Jenkins pipeline to test the functionality

Problem Statement - 5

You are working with FinTarget as a Senior Lead now. In the past, you successfully set up the version control system from scratch and have implemented Jenkins with many capabilities. As FinTarget has grown to be a big organization, there are thousands of jobs. Your manager has asked you to implement end-to-end automation of the build process. For that, you need to build the Jenkins pipeline. It shall be an automated process where you need to create branches as well using the scripting file. As there are thousands of jobs, it is practically impossible to track all the status of multiple pipelines. Hence you would also need to provide:

1. Visual representation of pipeline stage execution

2. Build history of the stages

Note: The team stores their code in GitHub repository (example-repo).

Steps by step guide

1. Create example-repo on GitHub.
2. Create a Jenkinsfile with stage info to check out the repository's master branch and run the command git log.
3. Once you are done finalizing the Jenkinsfile, push the file to example-repo's master branch
4. Install Jenkins on a local machine.
5. Install the Pipeline Plugin from the Jenkins plugin management.
6. Create a new Multibranch Pipeline job from New Item in Jenkins with the name Commit history.
7. Configure your Multibranch pipeline. SCM source should point to example-repo URL.
8. Save the configuration.
9. Watch the pipeline run for all the branches of the example-repo.
10. Install the appropriate plugin to show the pipeline status and build history.