**Architecting Jenkins Pipeline for Scale**

Course-end Project 1

Description

Use Jenkins to set up a distributed pipeline that will compile and test a Maven project on two different slave nodes respectively.

**Background of the problem statement:**

You’re a DevOps engineer at Softmax Solutions, a software company that develops image filters for various photo enhancement apps. It is undergoing an infrastructural change to implement DevOps in its development process. The company uses git as their Source Code Management System and AWS for hosting its servers. You’re required to architect a scalable Jenkins Pipeline for building and testing the software stack. You’re tasked with designing a Jenkins architecture that involves one master and two slave nodes, all hosted on various AWS instances. The build jobs should always be triggered by the master and executed on the slaves. You have to set up a pipeline on the master node and write a Groovy script that clearly differentiates the tasks to be run on various slaves.

**The following tools are used in this project:**

* **Git:** As a version control system for the software
* **Jenkins:** To create the build pipeline
* **Spring boot:** To create the Maven app
* **Maven**: To compile the program
* **AWS EC2**: To run the master and slave nodes

**The following requirements should be met:**

* The app should be built with Maven.
* There should be three EC2 instances to run the master and two slave nodes.
* All builds should be triggered and monitored by the master node.
* Compilation and testing should be done on dedicated slave nodes.