

## **Problem Statement:**

AppleBite Co. is using Cloud for one of their products. The project uses modular components, multiple frameworks and want the components to be developed by different teams or by 3<sup>rd</sup> party vendors.

The company's goal is to deliver the product updates frequently to production with High quality & Reliability. They also want to accelerate software delivery speed, quality and reduce feedback time between developers and testers.

As development progressed, they are facing multiple problems, because of various technologies involved in the project.

Following are the problems:

- Building Complex builds is difficult
- Manual efforts to test various components/modules of the project
- Incremental builds are difficult to manage, test and deploy

To solve these problems, they need to implement Continuous Integration & Continuous Deployment with DevOps using following tools:

Git – For version control for tracking changes in the code files

Jenkins – For continuous integration and continuous deployment

Docker – For deploying containerized applications

Puppet/Ansible - Configuration management tools

Selenium - For automating tests on the deployed web application

This project will be about how to do deploy code to dev/stage/prod etc, just on a click of button.

## **Business challenge/requirement**

As soon as the developer pushes the updated code on the GIT master branch, a new test server should be provisioned with all the required software. Post this, the code should be containerized and deployed on the test server.

The deployment should then be tested using a test automation tool, and if the build is successful, it should be pushed to the prod server.

All this should happen automatically and should be triggered from a push to the GitHub master branch.

## **Steps for executing the solution:**

- - Use the Master VM for Jenkins, Ansible, Puppet, GIT etc.
  - Use the Clean Ubuntu VM image provided in the “Edureka Setup Guide” for Jenkins Slave Node (Test Server)
  - Change the IP address of the VMs accordingly
  - Add Build Pipeline Plugin and Post build task plugin to Jenkins on the master VM
  - Install python, openssh server and git on the slave node manually
  - Set up the necessary tools such as git, chromedriver(selenium), chromium browser(selenium) on the slave node through Ansible

- Use the image devopsedu/webapp and add your PHP website to it using a Dockerfile
- Create a Selenium Test for your PHP website. It should click on “About” and verify the text written in it. This will conclude the website is deployed and is running fine.
- Push the PHP website, Dockerfile and Selenium JAR to a git repository

Below tasks should be automated through Jenkins by creating a pipeline:

- 1.Install and configure puppet agent on the slave node (Job 1)
- 2.Sign the puppet certificate on master using Jenkins (Job 2)
- 3.Trigger the puppet agent on test server to install docker (Job 3)
- 4.Pull the PHP website, Dockerfile and Selenium JAR from your git repo and build and deploy your PHP docker container. After this test the deployment using Selenium JAR file. (Job 4)
- 5.If Job 4 fails, delete the running container on Test Server

