# **Using Code pipeline in the project**

# **TechSpace Academy**

Our Project focuses on the Learning platform, and it offers multiple courses in different departments with flexibility of choosing each course based on Customer’s requirement and availability. It provides a few customizations like offline, online, hybrid etc. In this document we are going to document using Code pipeline to make our project automated regarding changes done and automatically pull these changes for deployment. We are assuming to keep our project as Mid-level budget requirement which means as of now, we will go with low-cost services available and in future we may adopt or switch to more convincing and suitable services according to the business needs.

**Tech stack and tools required for Project**

**Framework** – We are not using any framework like Node js or Spring boot because we have decided to go with PHP.

**Project** – Got a Free HTML/CSS template project on Learning platform while searching on the Google platform. We will be utilising that and modify it according to our project requirements.

**Programming language** – to keep it simple we have decided to use html, css and php for the development and modification of the project

**Tools** – GitHub is used for storing, tracking and collaborating the project within the team, JIRA is used to run the scrum every sprint and track down the tasks and its completion, Microsoft doc to store all project related contents to be referred whenever we need or share within the team and instructor. We may use some more tools according to the need soon.

**Approaches to implement CI/CD**

**AWS Code pipeline Service:** This is Infrastructure as Service approach in whichwe go to AWS services either in Lab module or BVC account and choose AWS Code pipeline service. Here we define the pipeline structure by choosing all the configurations which are already available and easy to select like source, destination. Since choices are available, we can customize our code pipeline according to our project requirements. This is easy and fast as we don’t have to remember the configuration. Also, it avoids having wrong setup since we have the options visible. We would know other events to customize the pipeline like creating templates, updating existing pipelines, receiving notifications, giving privileges, registering custom actions etc.

**Using Terraform:** This is Infrastructure as a Code approach in which we create terraform file and add the required configuration to define Code pipeline structure. We define the configuration manually like version, source, destination, providing credentials etc. This approach can be done only by using BVC account as it requires access and secret key. This approach is a little time-consuming as it involves a lot of manual steps to be implemented. This can be adapted to other tools like GitHub and automate the workflow. Better collaboration - shared resources can be managed and used by the same team, avoiding conflict and undetected changes while offering a better review process.

**Implementation of Code pipeline in the Project**

For our project we are choosing to implement the Code pipeline using Terraform approach as it feels Infrastructure as a Code gives more ownership on developer and the team to define or customize the pipeline structure the way we want. First, we are using GitHub to create repository and add our existing project in that repository. We use AWS BVC account for the implementation as it gives no restrictions on the usage of the services and also, we would require access and secret key which are not available in the Lab module. We create terraform files to define and configure EC2, S3 instances and map these instances in code pipeline configuration. So, in the same terraform file we would be creating all the instances and keep the things ready. Open Linux command and open AWS configuration session and provide the keys as it prompts. Now change the terraform folder path and give terraform cycles – terraform init, terraform plan and terraform apply. Once these are executed successfully our EC2, S3, Code pipeline would have been created in BVC AWS account with given configuration.

Why are we considering this approach?

Firstly, we have not done this approach yet and it is an opportunity to explore, learn and know more on IaaC way. Secondly there are few limitations with AWS code pipeline service approach which are - AWS Code Pipeline into one AWS account and AWS Region only. Configuration changes are required for multi-account and multi-Region deployments, and certain IAM privileges restrictions. As previously mentioned, terraform approach gives us more ownership on the code and customize the way team wants. This can be maintained along with other code base and provide variables to create multiple accounts, multiple regions or customize in any level just by providing variables and provide those variables dynamically (storing them in RDS or vault) which create multiple terraform files with different configurations and thereby creating multiple pipeline structures. However, there is a restriction with this approach is we need access/secret keys which are available only in BVC account.

**Future Recommendations**

As we have not done this activity and are exploring this approach as part of our project lifecycle, we may in future switch to other approaches like GitHub actions, GitLab CI/CD, Jenkins, Elastic beanstalk. Jenkins– **an open-source automation server** which enables developers around the world to reliably build, test, and deploy their software. Jenkins also provides code pipeline service. As we use dynamic content and user interaction to provide customer data, we would be using RDS for the Database storage. This also can be created using Terraform and can be connected via SQL workbench and perform database related tasks.

**Team Members : Group2**

**Aditya Mhatre** - a.mhatre623@mybvc.ca

**Geetha Narasimha** - g.narasimha064@mybvc.ca

**Sandeep Parwal** - s.parwal175@mybvc.ca

**Taral Kotwal** - [t.kotwal365@mybvc.ca](mailto:t.kotwal365@mybvc.ca)