**Introduction**

This Project Use Case is for the mid-term project use case display for Publicis Sapient in which the candidates have undergone trainings on multiple technology stack. The candidates are freshers passed out of various engineering branches. This Project Plan is crafted in such a way that the candidates will get the feeling of working on a project in which real-life problem statements are provided, it will evaluate their problem-solving skills and how to implement different tool chains via simulated environment.

**Problem Statement**

The team is provided with a ready code base that is developed in SpringBoot and Java. The code is available at :

<https://github.com/ptabasso2/springblog>

You have a DevOps & Monitoring role and you need to work on the following :

* Use the existing codebase for front-end and backend, defined as springback and springfront in the Github URL
* Two distinct microservices (springfront and springback) communicating with each other through REST. The backend service in turn does a bit of processing and issues two external http calls [Google.fr and an API call]
* Understand the code and use Github for your code management
* If there are issues in the code, try fixing the issues
* Build and execute the code
* Build the application and run it locally – Existing application uses Gradle for build management
* Create the Dockerfile and create the images – springfront and springback
* Push those newly created images to your own remote image registry (ex: dockerhub or any other registry of your choice)
* Scale the application using Docker-compose and create the cluster of 2 Nodes [local + AWS ec2 OR both nodes in AWS Ec2]
* Deploying the Datadog Agent and instrumenting the application
* For the monitoring of the application, you would be using:
* DataDog Free Account
* Infrastructure monitoring using Infrastructure List [depending on the deployment environment]
* Create dashboard to display timeseries and charts
* Monitor CPU and Memory usage
* Monitor Java app using Java APM in DataDog

**Demo:**

You would need to explain the working of the application, any bugs/challenges you came across in the existing code base, how the integration worked. Also, explain the dashboard and the metrics being monitored.

**Outcome**

On working on this Project Case, you will get the following exposure:

1. Working on an existing code base

2. Integrating and monitoring an existing service

3. Demonstrate ability to resolve issues

4. Monitoring

5. Working with 3rd party packages

6. Integration of different DevOps tool chain

7. Code collaboration and management using Github

8. Observability Patterns

9. Development/Update of Code

10. Automation

**Assumptions:**

1. Has access to existing code base

2. You have been trained on the technology stack you are using

3. Application deployment using Docker and scaling using Docker-Compose

4. Basic understanding of configuration files

5. Understand the core concepts of Observability, can do fundamental monitoring with DataDog