## **Project Abstract -**

Participants: Samina, Sushma, Khurram, Jacob and Matan

High level: Our goal is to create an enterprise application for a retailer who sells classic cars. The application will handle the main flows that are required by the retailer, such as: storing customer information, product information, handle sales orders, etc. The application will be web based and will be stored on the cloud.

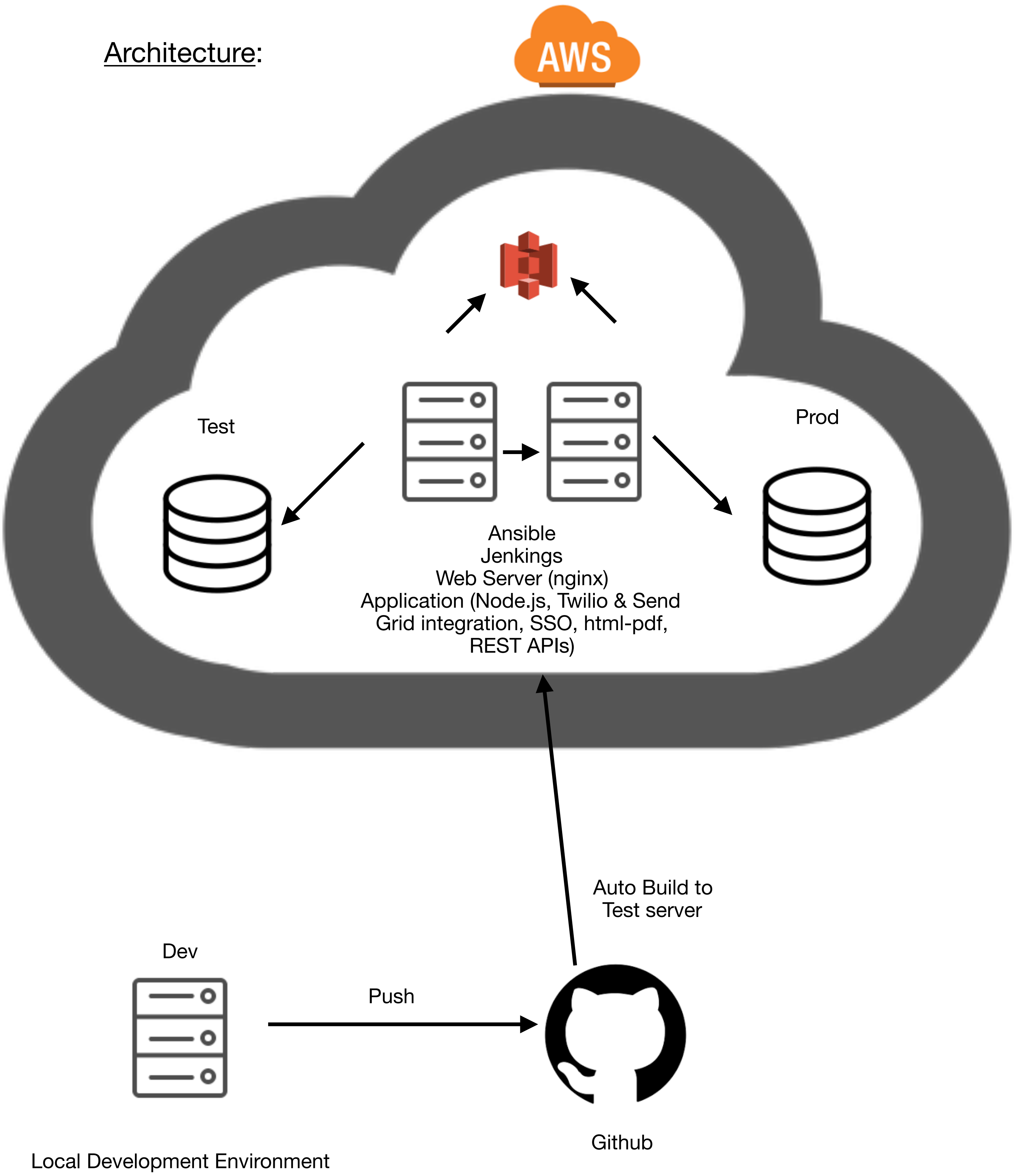
Technologies:

1. Backend Programming Language - Node.js
2. Frontend Programming Language - HTML, JS & CSS
3. AWS Cloud -
    1. EC2 (mini) for different servers we require.
    2. RDS to store data (MySQL)
    3. S3 to store files (pdfs)
4. Jenkins - Build system that will be linked to git project.
5. Github - Git project that will be linked to an auto deployment flow.
6. Ansible - Scripts to deploy and un-deploy the entire application (server/database/storage/etc)
7. SSO - Login with SSO using open source libraries, such as passport.js and Auth0.
8. SFDC Integration - After each build we will run automatic system test before deploying to production.
9. Send grid - Send emails.
10. Twilio - Send SMS.

Extra features:

1. Social media integration - Login with Twitter & Facebook +
2. Notifications via emails / SMS - Whenever there’s a new sales order we will send an email to VP Sales and in case of a new customer the VP Customer Success will get SMS.
3. Alexa Skill to perform orders - AWS Lambda + Skill Configuration + Expose REST API from the application to AWS Lambda.

## Architecture:



AWS

Test

Prod

Ansible
Jenkings
Web Server (nginx)
Application (Node.js, Twilio & Send Grid integration, SSO, html-pdf, REST APIs)

Auto Build to Test server

Dev

Push

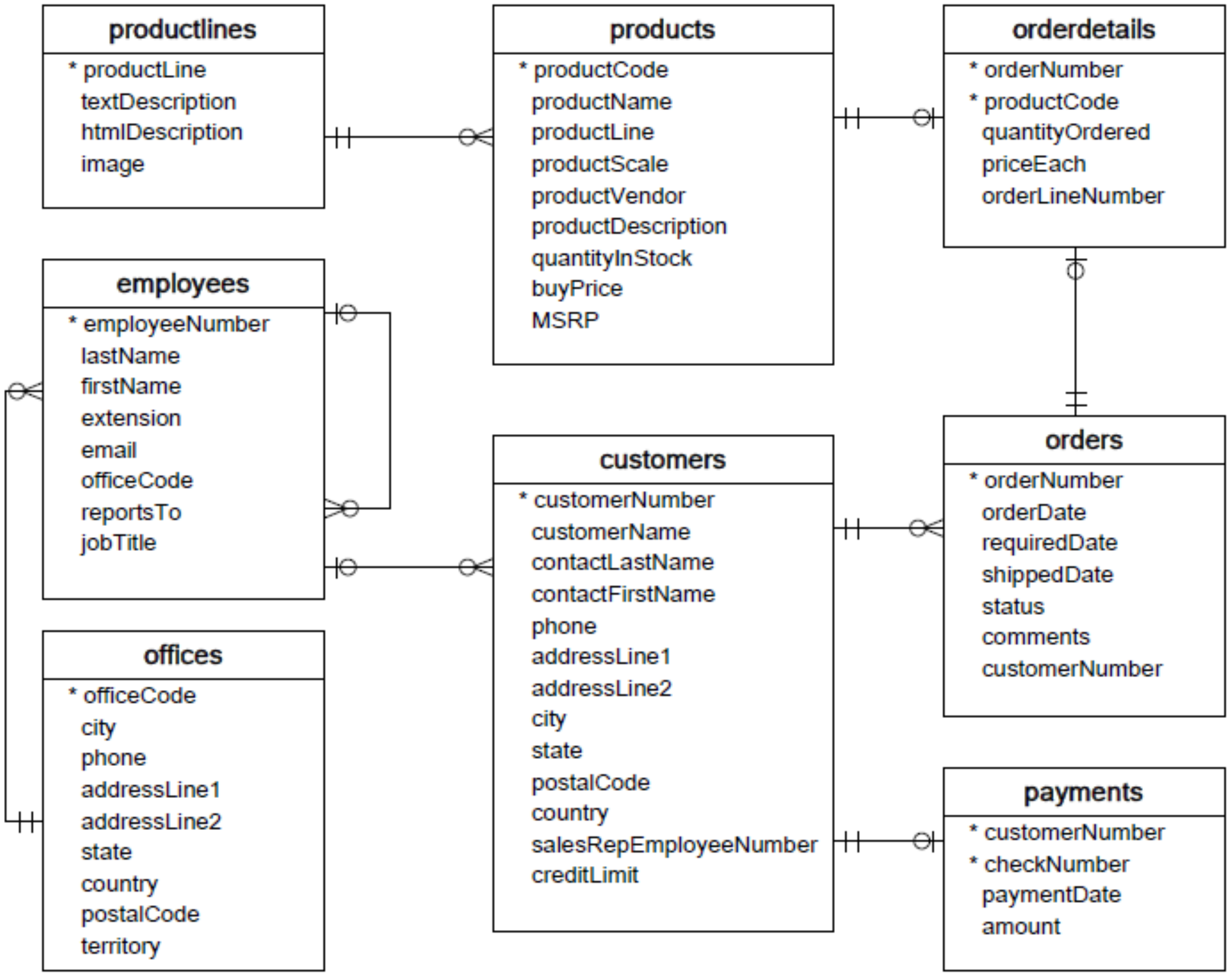
Github

Local Development Environment

Database information: We use the classicmodels database as a MySQL sample database to help us work with MySQL quickly and effectively. The classicmodels database is a retailer of scale models of classic cars database. It contains typical business data such as customers, products, sales orders, sales order line items, etc.

The MySQL sample database schema consists of the following tables:

- **Customers**: stores customer's data.
- **Products**: stores a list of scale model cars.
- **ProductLines**: stores a list of product line categories.
- **Orders**: stores  sales orders placed by customers.
- **OrderDetails**: stores sales order line items for each sales order.
- **Payments**: stores payments made by customers based on their accounts.
- **Employees**: stores all employee information as well as the organization structure such as who reports to whom.
- **Offices**: stores sales office data.



Source - http://www.mysqltutorial.org/mysql-sample-database.aspx