End of Phase 4 Project - Online Test Application

Project and developer details:

Application Name: Capstone Medicare

Developer Details: Emmanuel Zeye

GitHub URL: https://github.com/emmanuel-zeye/capstone-medicare.git

1. **Introduction**

Capstone Medicare application is a web application that provides a store for purchasing medicines and medical supplies, making access to healthcare hassle free. It spots an administrator interface as well as a client interface. The administrator dashboard enables an authenticated administrator to manage the available medical supplies and their metadata, as well as user data and orders. The customer-facing application allows the customer to browse through available products, place orders and checkout items on the store. Below is a breakdown of functionality on each app:

Admin

* Create, view, update and delete medical supplies
* Create, view, update and delete medical supply categories
* View orders created by customers

Customers

* Add to and remove items from cart
* Modify cart item quantity
* Checkout cart items
* Make payment
* View orders

Technologies used for this project are:

* HTML for page layout
* Vanilla CSS and bootstrap (react-bootstrap) for styling
* Spring boot (java) for backend logic
* H2 database for data persistence
* React JS for ui Kit
* Javascript for frontend logic
* Intellij Idea for the Java development and frontend development
* Tomcat server for the backend APIs
* Selenium, Jasmine, and TestNG for automation test.
* GitHub for code repository and versioning
* Jenkins for CI/CD
* Maven for dependency management
* AWS for cloud services

**Sprints Planning**

The application would require three sprints period; each sprint would be two weeks. In all, we will have six weeks to deliver the projects.

The following product backlogs were identified to be part of the sprint planning:

|  |  |
| --- | --- |
| Item number | Activity |
| 1. | Review functional requirements with all stakeholders (client and our team) |
| 2 | Gather all user stories for the requirements |
| 3 | Gather all images for all medicine |
| 4 | Prepare architectural design and integration documents |
| 5 | Produce wireframe for business requirements |
| 6 | Build and test the APIs for the backends |
| 7 | Build the frontend and integrate it to the backend using the API |

Sprint tasks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Assignee | Task | Status | Description | Story Points | Sprint |
| Emmanuel Zeye | create Restful APIfor users authentication and management | completed | API for user authentication | 8 | 1 |
| Emmanuel Zeye | create Restful APIto Add, delete, update and list products | completed | CRUD APIs for managing products | 5 | 1 |
| Emmanuel Zeye | create Restful APIto Add and update product category | completed | CRUD APIs for managing category | 5 | 1 |
| Emmanuel Zeye | create Restful APIto manage cart | completed | CRUD APIs for managing Cart | 5 | 1 |
| Emmanuel Zeye | create APIs for checkout | completed |  | 5 | 2 |
| Emmanuel Zeye | create APIs to manage orders | completed |  | 5 | 2 |
| Emmanuel Zeye | create signup page | completed | sign up page must have username, email, password and register button | 5 | 2 |
| Emmanuel Zeye | create login page | completed | login page should show username and password and submit button. Password must be enrypted | 3 | 2 |
| Emmanuel Zeye | create logout page | completed | when logout page is called it redirect to Home page | 2 | 2 |
| Emmanuel Zeye | CRUD page for products | completed | admin can add, delete, update aand list all product. Fields to consider are product name, price, picture, initial quantity | 5 | 3 |
| Emmanuel Zeye | Add pagination to product listing, show ten product per page | completed | Pagination should display 10 items per page | 5 | 3 |
| Emmanuel Zeye | create customer page to see all products | completed |  | 8 | 3 |
| Emmanuel Zeye | add a add to cart to the customer products page | completed | make sure customers can add product to cart | 5 | 3 |
| Emmanuel Zeye | create a cart page for the customer to view all items in his cart | completed | login customer can only see items in his cart only | 5 | 3 |
| Emmanuel Zeye | Add a functionality to the cart page to increase or decrease quantity per product in the cart | completed |  | 2 | 4 |
| Emmanuel Zeye | Allow user to delete product in their cart | completed |  | 1 | 4 |
| Emmanuel Zeye | Add a button to the cart page where use can checkout orders. The product in the cart should be added to the orders | completed |  | 1 | 4 |
| Emmanuel Zeye | create a page for admin where he sees all orders and their statuses | completed |  | 5 | 4 |
| Emmanuel Zeye | create a page for customers to see his historical orders | completed |  | 5 | 4 |
| Emmanuel Zeye | Writing unit tests | completed |  | 5 | 5 |
| Emmanuel Zeye | Deploying to AWS | completed |  | 8 | 5 |

Database Diagram

A screenshot of a computer

Description automatically generated

Screenshots from application

General

A screenshot of a login form

Description automatically generated

Figure : Login

A screenshot of a login form

Description automatically generated

Figure : Account creation

Admin Dashboard

A screenshot of a computer

Description automatically generated

Figure : Product category

A screenshot of a computer

Description automatically generated

Figure : Product Listing

A screenshot of a computer

Description automatically generated

Figure : User Orders

A screenshot of a computer

Description automatically generated

Figure : User Management

Customer portal

A screenshot of a phone

Description automatically generated

Figure : Storefront

A screenshot of a computer

Description automatically generated

Figure : Cart

A screenshot of a computer

Description automatically generated

Figure : Checkout

A screenshot of a computer

Description automatically generated

Figure : Customer Orders

AWS Setup

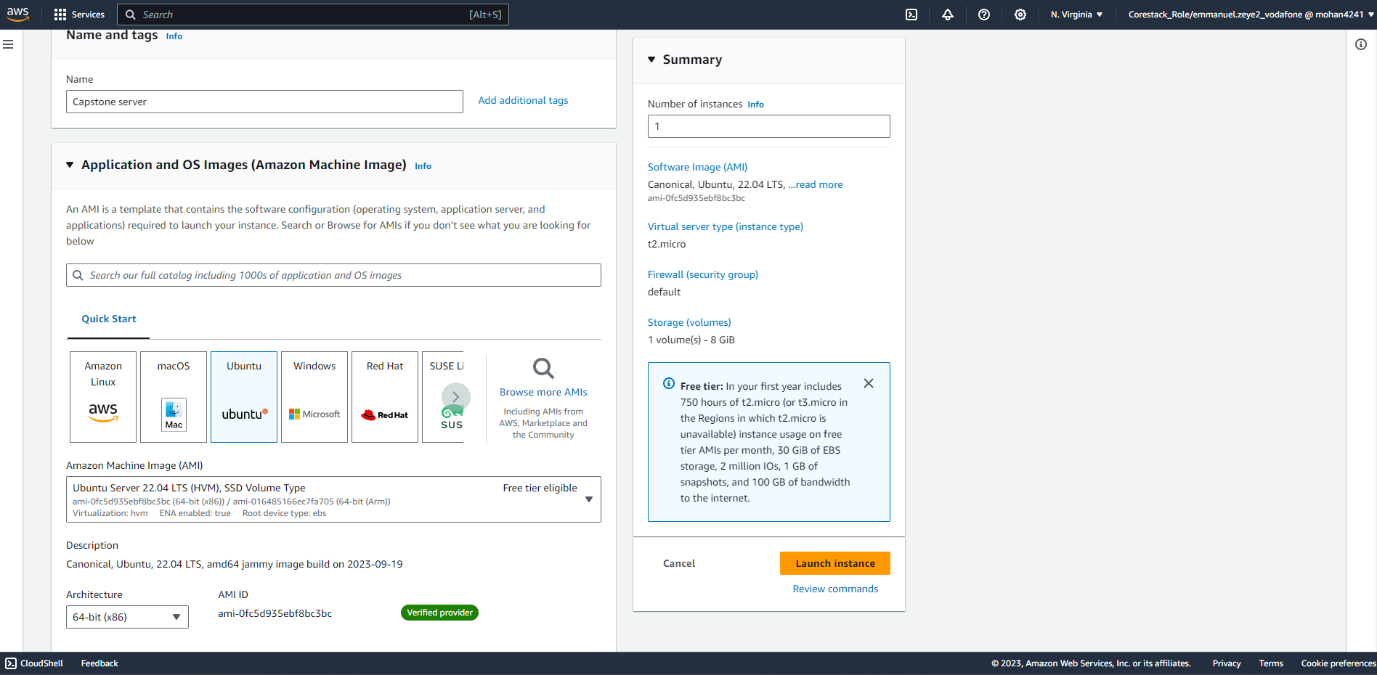


Figure : Creating an EC2 Instance with Ubuntu Image

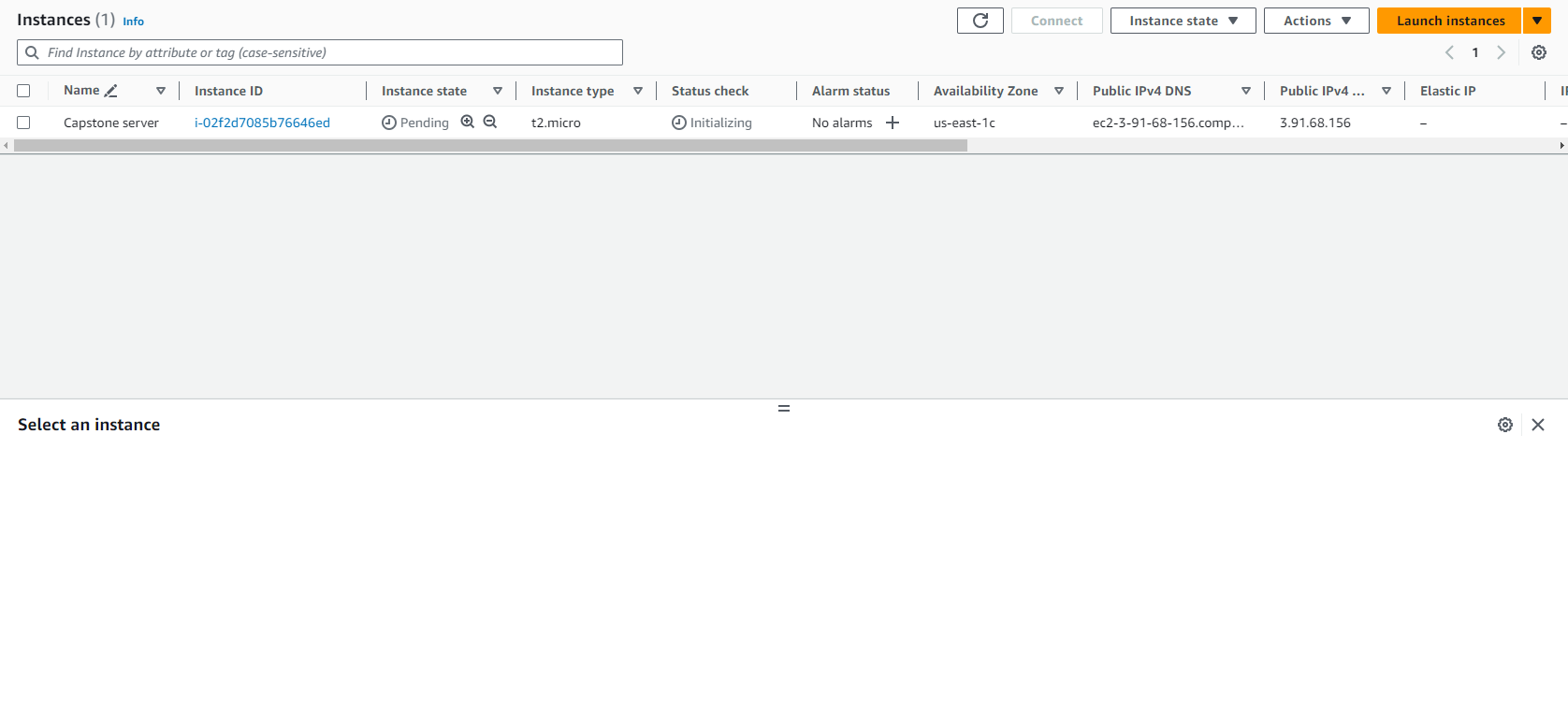


Figure : EC2 Instance created

CI/CD Setup with Jenkins

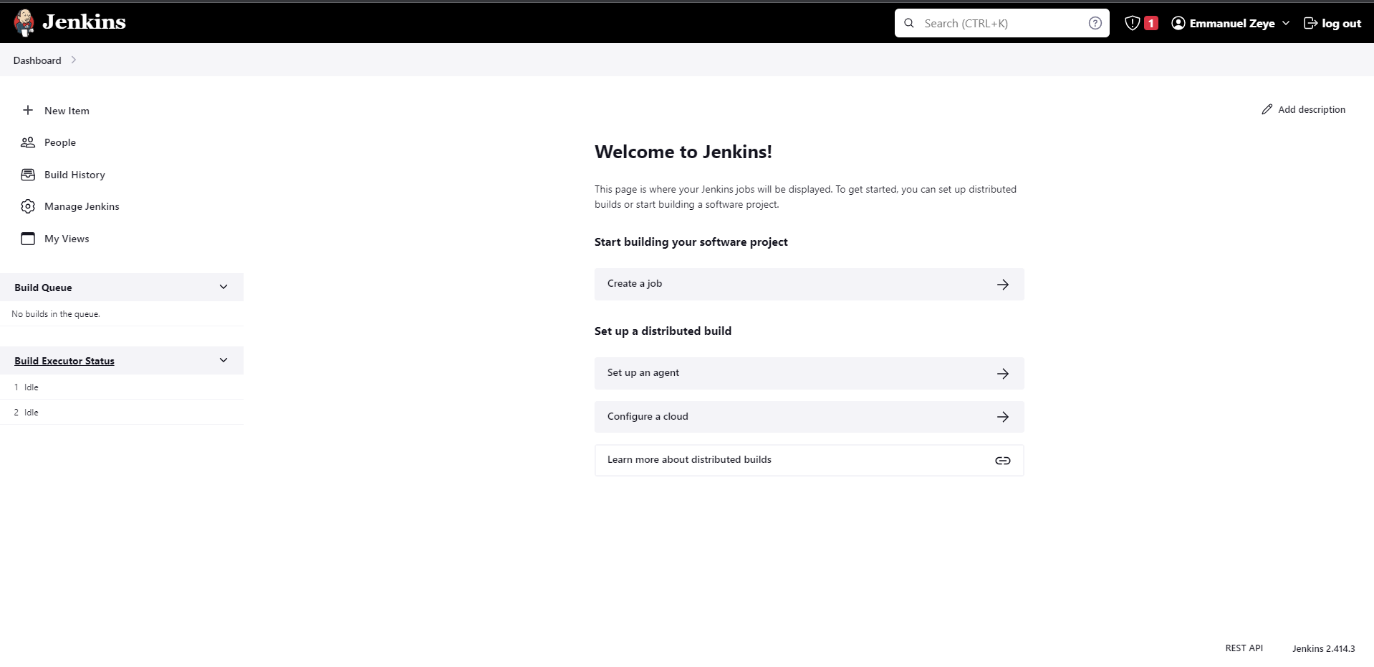


Figure : CI/CD Setup with Jenkins

Jenkins Pipeline Script

A screenshot of a computer

Description automatically generated

Figure : Pipeline script