

# Café Cloud

**By**  
**Team Members & Respective CRN's**

Sai Smruthi Siddemsetty(**CRN 13898**) – 700754105

Jyothirlatha Perla (**CRN 13898**) – 700759284

Srirama Sphoorthy (**CRN 12463**) – 700756577

## TABLE OF CONTENTS

Title	Page Nos.
Purpose of study	i
Intended Audience	ii
Chapter 1: Introduction	1-3
Chapter 2: Architectural Design	4-6
Chapter 3: Detailed Design	7-8
Chapter 4: User Interface Design	9-10
Chapter 5: Conclusion	11
Results	12-17

## **Purpose of study**

Our project aims to simplify and automate the various processes involved in planning and managing orders at a café. The platform will offer customers an easy-to-use interface to place online orders, thus enhancing their overall café experience. The project involves setting up the website on AWS and configuring two environments (development and production) in different AWS regions while ensuring the website's functionality and accessibility. The café aims to improve customer experience by providing online ordering capabilities. Through the enhanced café website with online ordering, the café can meet customer demand and increase customer satisfaction. The robust web application enables controlled testing of new features, with multi-region deployment for complete disaster recovery.

## **Intended Audience**

The target audience for this document includes project team members such as developers, project leads, operations managers, and database administrators.

### **TEAM MEMBERS:**

Sai Smruthi Siddemsetty (Project Lead & Developer)

Jyothirlatha Perla (Operations Manager)

Srirama Sphoorthy (Database Administrator)

# INTRODUCTION

The Café Cloud project aims to simplify and automate the various processes involved in managing orders and planning in a café. It aims to provide a user-friendly interface for customers to place online orders and enhance their overall café experience. The project involves setting up the website on AWS, configuring two different environments (development and production) in distinct AWS Regions, and ensuring the website's functionality and accessibility.

## 1.1 Project Overview: Café Cloud

The Café Cloud project was initiated to meet the evolving needs of a forward-thinking café client who aimed to enhance its operational efficiency and customer satisfaction. The client, a vibrant café, recognized the growing demand for a more streamlined and modern approach to order planning and management. The main issue was the traditional order processes' manual and time-consuming nature, which necessitated automation and a more user-friendly online ordering system. The goal was to provide customers with a seamless experience while optimizing internal processes for café staff.

The primary challenge to be addressed is the inefficiency of the existing order management system. With the rise in demand for online services, the café aims to transition into a digital space where customers can easily place orders, customize their preferences, and enjoy a more efficient and modern café experience. Both the café's customers and internal staff will be the intended users. Customers will interact with the platform to place orders, while café staff will use the system to manage incoming orders, track inventory, and ensure timely and accurate order fulfillment.

The big picture involves transforming the café's operational landscape by implementing a comprehensive digital solution. Once the Café Cloud software is deployed, it will seamlessly integrate into the daily operations of the café, bridging the gap between customer expectations and traditional manual processes. This is set in a bustling café environment, where the software becomes an integral part of the workflow, meeting current demands for online ordering and future-proofing the café's operations in an increasingly digital-centric business landscape. Ultimately, Café Cloud aims to create a harmonious blend of technology and hospitality, enhancing the overall café experience for customers and staff.

## 1.2 Project Scope

The Café Cloud project encompasses a comprehensive set of features designed to streamline order planning and management processes within the café environment. The most critical features include:

**Online Ordering System:** The core functionality of Café Cloud, allowing customers to place orders seamlessly through a user-friendly interface. This feature includes the ability to customize orders, select preferences, and view a real-time menu.

**Development and Production Environments:** The project involves the establishment of two distinct environments – one for development and one for production. This ensures a systematic approach to testing, refining, and deploying new features without disrupting the live system.

**Secure Database Storage:** The system incorporates secure storage through MariaDB, hosting a MySQL database for order and menu data. This feature ensures data integrity, scalability, and efficient retrieval for order processing.

**Multi-Region Deployment:** Leveraging AWS, the project includes the deployment of the café's website in different AWS Regions. This not only enhances disaster recovery capabilities but also ensures high availability, minimizing downtime and providing a reliable service to customers.

**Parameter Management with Systems Manager:** The use of AWS Systems Manager Parameter Store facilitates secure and efficient management of application parameters. This feature enables easy configuration changes, allowing for flexibility in adapting to evolving requirements.

## **Inputs, Data Stores, and Outputs:**

**Inputs:** Customer orders, preferences, and modifications are key inputs to the system. Additionally, system configuration changes and updates from the development environment are considered inputs during the project lifecycle.

**Data Stores:** The MySQL database hosted on MariaDB serves as the primary data store for order and menu data. Amazon S3 is utilized for storing web application and database files, providing a backup and recovery mechanism.

**Outputs:** The primary output is the fulfillment of customer orders, with the system generating confirmation messages, receipts, and updating inventory levels. Additionally, during the development phase, outputs include feedback and updates to improve system functionality.

## **Major Constraints:**

**User Adoption:** The successful implementation of the Café Cloud system is contingent upon user adoption within the café and among its customer base. Strategies for promoting and facilitating this adoption will be critical for the project's success.

**Data Security:** Ensuring the security and privacy of user data is a paramount constraint. The system must adhere to industry standards and regulations to safeguard sensitive information.

**AWS Costs:** The project must manage AWS expenditures effectively to remain within budget constraints. Regular monitoring and optimization of cloud resources are essential to prevent unexpected financial burdens.

In summary, the Café Cloud project focuses on enhancing the café experience through the implementation of critical features such as online ordering, multi-region deployment, secure database storage, and efficient parameter management. Inputs, data stores, and outputs are carefully considered within the project scope, while constraints highlight the importance of user adoption, data security, and cost management for successful project execution.

# ARCHITECTURAL DESIGN

## 2.1. Section Overview:

The Café Cloud project aims to simplify and automate the various processes involved in planning and managing orders in a café. The platform will provide a user-friendly interface for customers to place online orders, improving their overall café experience. The project involves

- setting up the website on AWS,
- configuring two environments (development and production) in different AWS Regions and
- ensuring that the website is functional and accessible.

The café seeks to enhance the customer experience by offering online ordering capabilities. An improved café website with online ordering will be able to meet customer demands more effectively. This more robust web application will increase customer satisfaction by improving online ordering and allowing for controlled testing of new features. Additionally, it will be deployed across multiple regions, providing robust disaster recovery capabilities.

## 2.2 General Constraints:

Several global constraints shape the architectural design of Café Cloud. The choice of Amazon Web Services (AWS) imposes hardware and software environment constraints, influencing decisions on hosting and deployment strategies. Interface requirements for a user-friendly online ordering system migrate to a different cloud provider. Interface requirements for a user-friendly online ordering system guide the design of the system's front-end. External data representations, particularly in the context of menu and order data, play a crucial role in defining data structures. Performance requirements, such as response times for order processing, influence design decisions, as do network requirements for seamless communication between different components of the system.

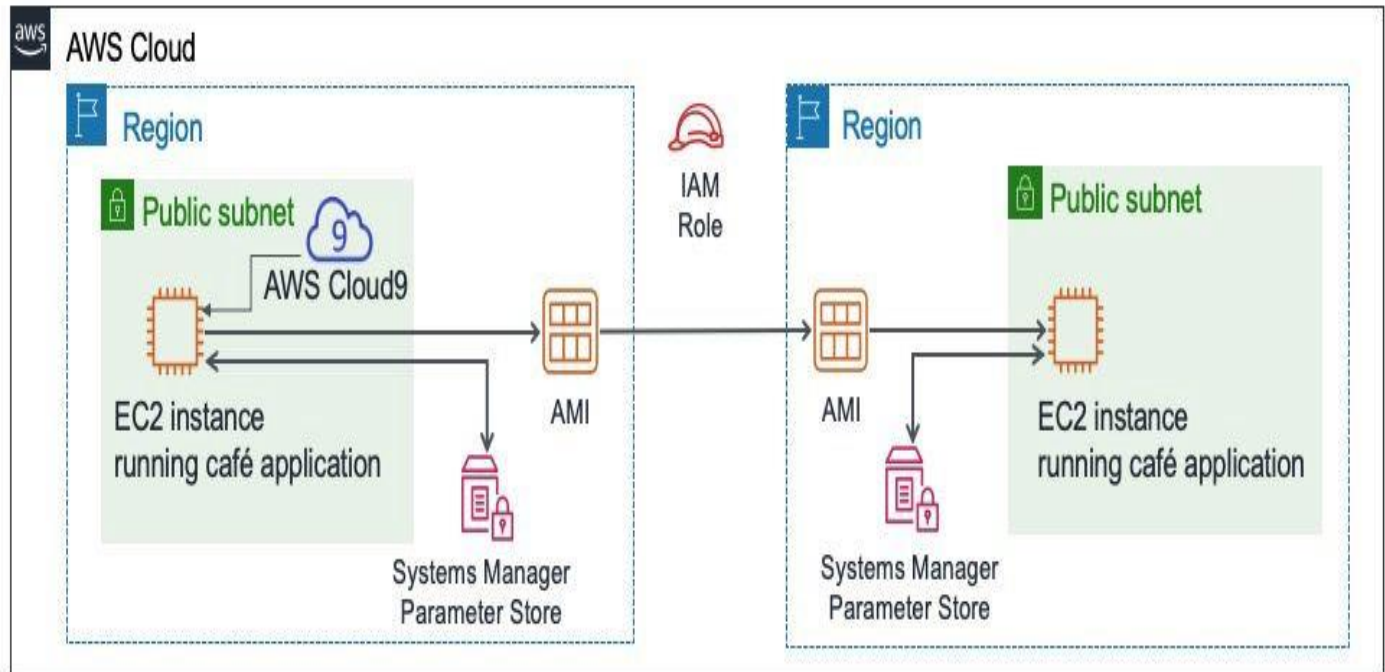
## 2.3. Data Design:

The data design of Café Cloud is centered around the utilization of MariaDB for hosting a MySQL database. This relational database structure ensures data integrity and scalability. The schema encompasses tables for orders, menu items, user information, and other relevant entities. Further details, including Entity-Relationship (ER) diagrams and data formats, are provided in appendices for reference. External files stored in Amazon S3 are used for data backup and recovery, contributing to a robust data management strategy.



## 2.4 Program Structure:

The following is the pictorial representation of the serverless web application using Amazon web services.



The architectural model chosen for Café Cloud is a multi-tiered structure, incorporating a presentation layer, application layer, and data layer. The presentation layer comprises the user interface for online ordering, developed using AWS Cloud9. The application layer houses the business logic and is deployed on Amazon EC2 instances. The data layer involves the MySQL database hosted on MariaDB. A pictorial representation, included as an appendix or referred to in block or class diagrams, illustrates the major components and their interactions.

The following are the brief description about AWS services that have used in the project:

**Amazon Simple Storage Service (Amazon S3)** is a widely used object storage service provided by Amazon Web Services (AWS). It allows users to store and retrieve any amount of data at any time from the internet. S3 is designed for object storage, which means it stores data as objects. Each object consists of data, a key (unique within a bucket), and metadata. Objects can range in size from a few bytes to terabytes. In S3, data is organized into containers called buckets. Each bucket has a globally unique name within AWS. Buckets are used to group and organize objects.

**Amazon Systems Manager (SSM)** is a comprehensive management service provided by Amazon Web Services (AWS) that allows users to automate and manage operational tasks on their AWS resources. It simplifies the complexity of system and application management, enabling administrators to efficiently operate at scale while maintaining security and compliance. SSM provides a centralized platform for tasks such as system configuration, patch management, and instance inventory tracking. Overall, Amazon Systems Manager is a powerful tool that enhances the operational efficiency of AWS environments by automating routine tasks, improving security, and ensuring that infrastructure remains in a desired and secure state.

**Amazon Identity and Access Management (IAM)** is a foundational service within Amazon Web Services (AWS) that enables users to securely control access to various AWS resources. With IAM, administrators can create and manage users, groups, and roles, defining fine-grained access policies to dictate what actions users can perform and what resources they can access. IAM integrates seamlessly with various AWS services, providing a secure and scalable approach to access management. Through IAM policies, organizations can implement the principle of least privilege, ensuring that users have only the necessary permissions required for their specific roles or tasks.

**Amazon Elastic Compute Cloud (Amazon EC2)** is a core and versatile compute service provided by Amazon Web Services (AWS). EC2 enables users to rent virtual servers, known as instances, in the cloud. EC2 instances offer a wide range of choices in terms of compute capacity, memory, storage, and networking capabilities, allowing users to tailor their virtual servers to meet specific performance needs. EC2 instances can run different operating systems, and users have control over instance security by configuring network access and using security groups.

**Amazon Virtual Private Cloud (Amazon VPC)** is a robust networking service provided by Amazon Web Services (AWS), offering a secure and isolated environment in the cloud. With AWS VPC, users can create their private, logically isolated network within the AWS infrastructure. This allows them to define their IP address range, configure route tables, and manage network gateways, providing a high level of control over their virtual network environment. VPC enables the creation of subnets, each residing in a specific Availability Zone, and users can customize network configurations.

**Amazon Cloud9** is a cloud-based integrated development environment (IDE) offered by Amazon Web Services (AWS) that enables developers to write, debug, and deploy code from any location with internet access. Developers can work on their projects directly from a web browser, eliminating the need for local installations and ensuring consistent environments. Additionally, Cloud9 integrates seamlessly with other AWS services, allowing for easy access to resources like databases, serverless functions, and storage solutions.

# DETAILED DESIGN

## 3.1. Section Overview:

The Detailed Design section delves into the intricate details of the Café Cloud system, providing a comprehensive understanding of each major component. This section aims to offer in-depth insights into the design decisions, data structures, methods, and interactions that collectively contribute to the functionality and efficiency of the system.

## 3.2. Component Detail:

### 3.2.1. User Interface Component:

**Description:** The User Interface (UI) component is responsible for presenting the online ordering system to customers. It includes web pages for menu browsing, order customization, and order confirmation.

#### Data Members:

**MenuData (type: JSON):** Stores the menu items and their details.

**OrderDetails (type: JSON):** Captures the customer's order details.

#### Methods:

**displayMenu():** Renders the menu on the UI.

**customizeOrder(itemID, customization):** Allows customers to customize their selected menu items.

**placeOrder():** Initiates the order placement process, capturing the finalized order details.

### 3.2.2. Order Management Component:

**Description:** The Order Management component handles the processing and fulfillment of customer orders.

#### **Data Members:**

**OrderQueue (type: Queue):** Stores incoming orders.

**Inventory (type: Database Table):** Tracks available stock of menu items.

#### **Methods:**

**processOrder(orderDetails):** Validates and processes incoming orders.

**updateInventory(itemID, quantity):** Adjusts inventory levels based on order fulfillment.

**notifyCustomer(orderStatus):** Sends notifications to customers regarding their order status.

### 3.2.3. Database Management Component:

**Description:** The Database Management component oversees the storage and retrieval of data in the MySQL database.

#### **Data Members:**

**ConnectionPool (type: Connection Pool):** Manages database connections.

**DataCache (type: Cache):** Temporarily stores frequently accessed data.

#### **Methods:**

**executeQuery(query):** Executes SQL queries on the database.

**fetchData(query):** Retrieves data from the database and updates the data cache.

**updateData(query, newData):** Modifies existing data in the database.

This structured approach provides a detailed breakdown of each major component, emphasizing their roles, data structures, and methods. The User Interface, Order Management, and Database Management components collectively form the backbone of the Café Cloud system, ensuring a seamless online ordering experience for customers while efficiently managing orders and data.

# User Interface Design

## 4.1. Section Overview:

The User Interface Design section offers a comprehensive overview of the design principles, rules, and components employed in the Café Cloud online ordering system. This section delves into the visual and interactive aspects of the system, providing insights into the user experience and the rationale behind design decisions.

## 4.2. Interface Design Rules:

The design of the Café Cloud user interface adheres to established conventions and standards to ensure a user-friendly and intuitive experience. The Material Design principles are adopted to provide a clean and visually appealing interface. Consistent color schemes, typography, and iconography are employed to enhance readability and navigation. Additionally, responsive design principles are implemented to ensure a seamless experience across various devices, catering to the diverse preferences of café customers.

## 4.3. GUI Components:

The Café Cloud user interface leverages various GUI components and APIs provided by the development environment, AWS Cloud9. These components include HTML for structuring the content, CSS for styling, and PHP for interactive elements.

### GUI Components Table:

Component	Description
HTML	Markup language for structuring web content.
CSS	Style sheet language for enhancing visual design.
PHP	Programming language widely used to build web applications.

#### **4.4. Detailed Description:**

The Café Cloud user interface is designed with a customer-centric approach, focusing on ease of use and visual appeal. Screenshots illustrating key interface elements, such as menu browsing, order customization, and confirmation, are provided in an appendix for reference. The home page welcomes users with a visually appealing layout, showcasing featured menu items and promotions.

The menu browsing page allows customers to explore available items, each accompanied by detailed descriptions and images. The order customization screen enables users to tailor their selections based on preferences, with real-time updates reflecting changes. The checkout and order confirmation pages provide clear summaries, ensuring a transparent and efficient ordering process.

Interactive elements, such as buttons and form fields, are strategically placed for intuitive navigation. Responsiveness is prioritized to ensure a consistent experience on desktops, tablets, and mobile devices. Visual cues, such as color changes and animations, are implemented to provide feedback on user interactions.

In conclusion, the Café Cloud user interface design combines aesthetic appeal with functionality, adhering to established design principles and leveraging GUI components to create a seamless and enjoyable online ordering experience for café customers.

## **Conclusion:**

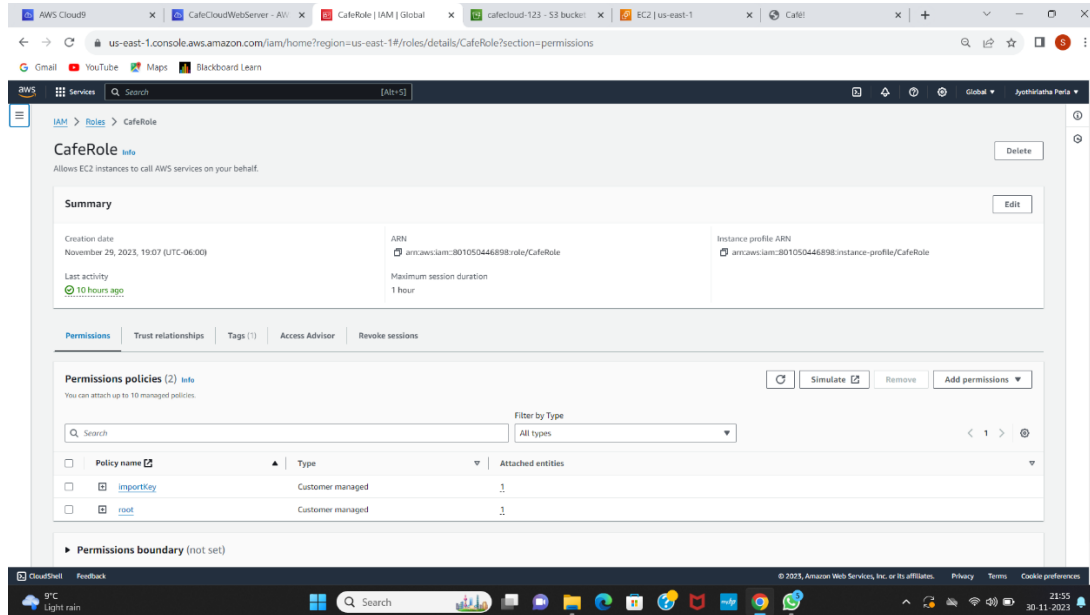
In conclusion, the detailed design documentation for Café Cloud has laid the groundwork for a robust and user-centric online ordering system. The architectural and user interface designs intricately address the needs of both the café customers and internal staff, ensuring a seamless experience. The multi-tiered architecture, microservices approach, and careful consideration of global constraints create a scalable and adaptable foundation for the implementation phase.

Moving forward, the implementation strategy will involve translating these detailed design specifications into code, leveraging the chosen technologies and cloud services. The development team will follow an agile methodology, allowing for iterative improvements and adaptations as the system evolves. Simultaneously, a rigorous testing strategy will be implemented, encompassing unit testing, integration testing, and user acceptance testing to validate the system's functionality, performance, and user experience.

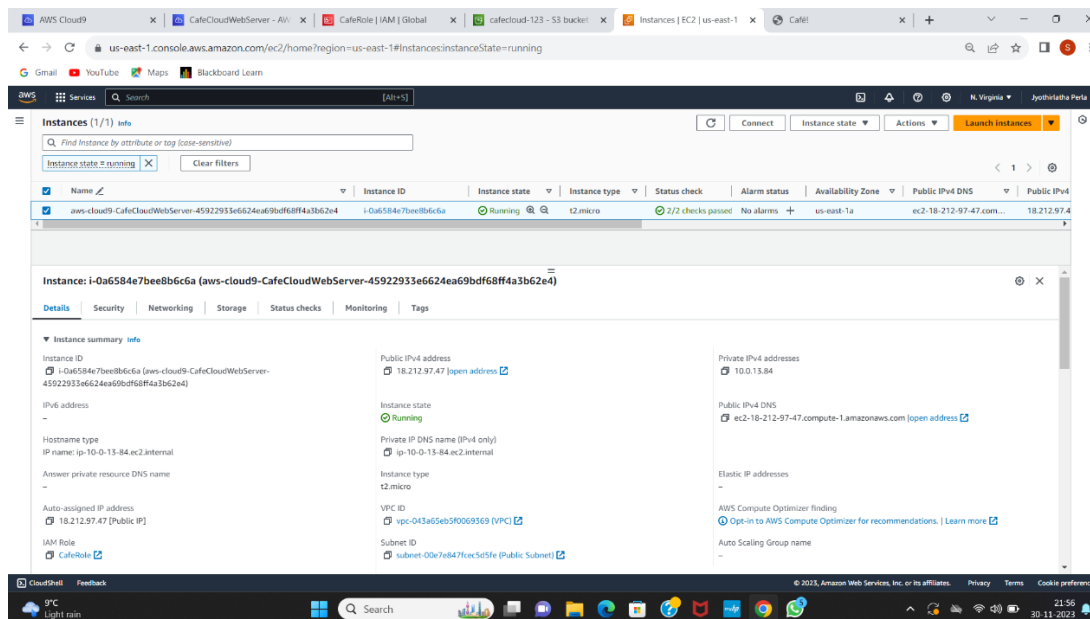
The Café Cloud project is poised to bring a transformative digital solution to the café environment, enhancing order management processes and providing customers with a modern and efficient online ordering experience. With a solid design foundation in place, the implementation and testing phases will play a crucial role in bringing this vision to fruition.

# Results:

## Identity Access Management(IAM)



## Amazon EC2





# Amazon Cloud9

The screenshot shows the AWS Cloud9 console with a green banner at the top stating "Successfully created CafeCloudWebServer. To get the most out of your environment, see [Best practices for using AWS Cloud9](#)". The left sidebar contains "My environments", "Shared with me", "All account environments", and "Documentation". The main content area displays the details for the "CafeCloudWebServer" environment.

**CafeCloudWebServer** [Delete] [Open in Cloud9] [Edit]

**Details**

Name CafeCloudWebServer	Owner ARN arn:aws:iam::801050446898:root	Status Ready
Description Hosting a Dynamic Webpage	Number of members 1	Lifecycle status Created
Environment type EC2 instance		

**EC2 instance** [Manage EC2 instance]

ARN arn:aws:cloud9:us-east-1:801050446898:environment:459229336624a6f8d6f68f4a3b62e4	Instance type t2.micro (1 GiB RAM • 1 vCPU)
Platform Amazon Linux 2	Storage EBS only

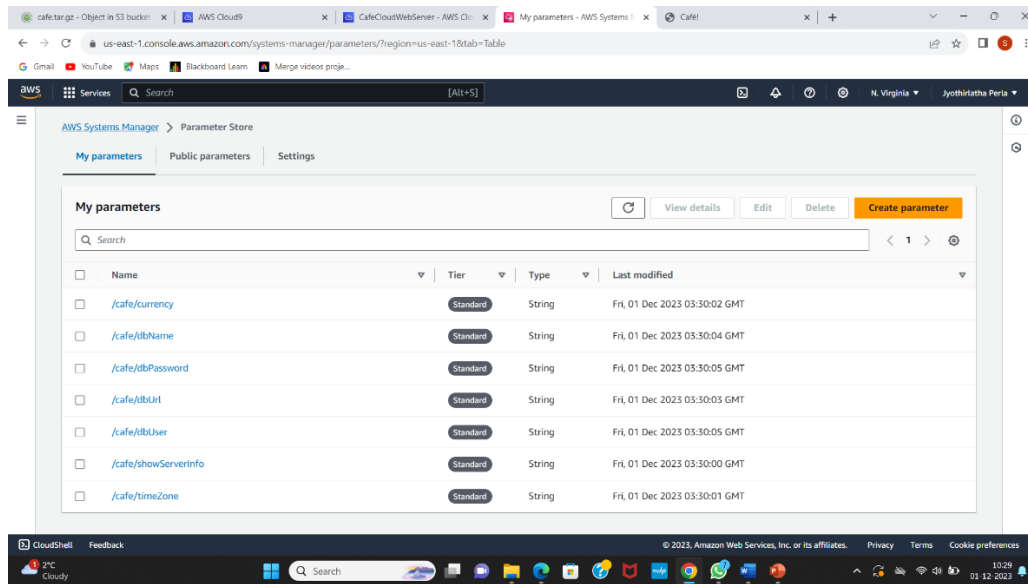
The screenshot shows the AWS Cloud9 console with the same green banner. The left sidebar is identical. The main content area displays the "Environments (1)" section.

**Environments (1)** [Delete] [View details] [Open in Cloud9] [Create environment]

My environments

Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
<a href="#">CafeCloudWebServer</a>	[Open]	EC2 instance	Secure Shell (SSH)	Owner	arn:aws:iam::801050446898:root

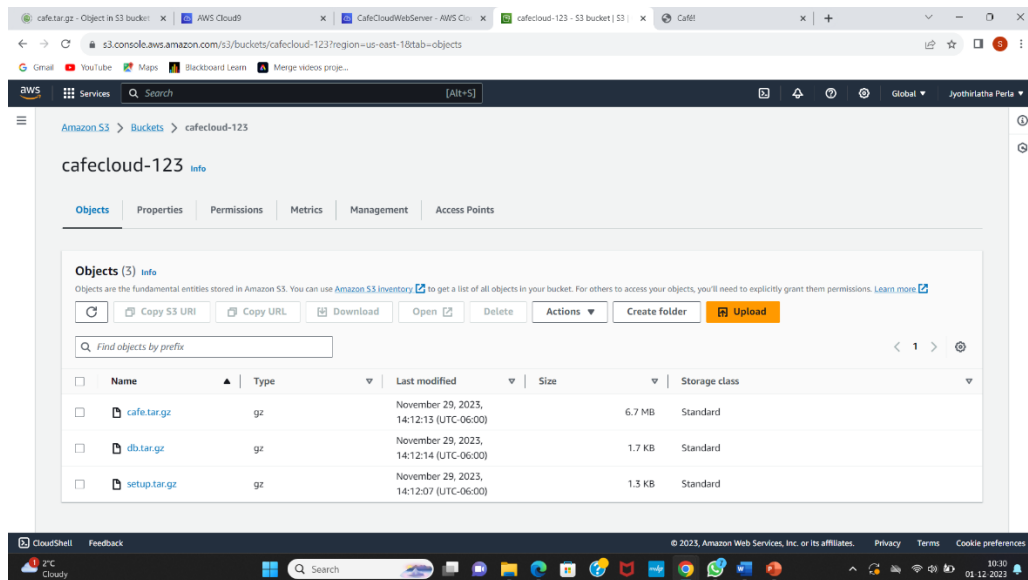
# Amazon Systems Manager



The screenshot shows the AWS Systems Manager Parameter Store console. The 'My parameters' tab is selected, displaying a list of parameters. The parameters are all of type 'String' and 'Standard' tier. The list includes parameters for currency, name, password, ID, user, server info, and time zone.

Name	Tier	Type	Last modified
/cafe/currency	Standard	String	Fri, 01 Dec 2023 03:30:02 GMT
/cafe/idName	Standard	String	Fri, 01 Dec 2023 03:30:04 GMT
/cafe/password	Standard	String	Fri, 01 Dec 2023 03:30:05 GMT
/cafe/idId	Standard	String	Fri, 01 Dec 2023 03:30:03 GMT
/cafe/user	Standard	String	Fri, 01 Dec 2023 03:30:05 GMT
/cafe/showServerInfo	Standard	String	Fri, 01 Dec 2023 03:30:00 GMT
/cafe/timeZone	Standard	String	Fri, 01 Dec 2023 03:30:01 GMT

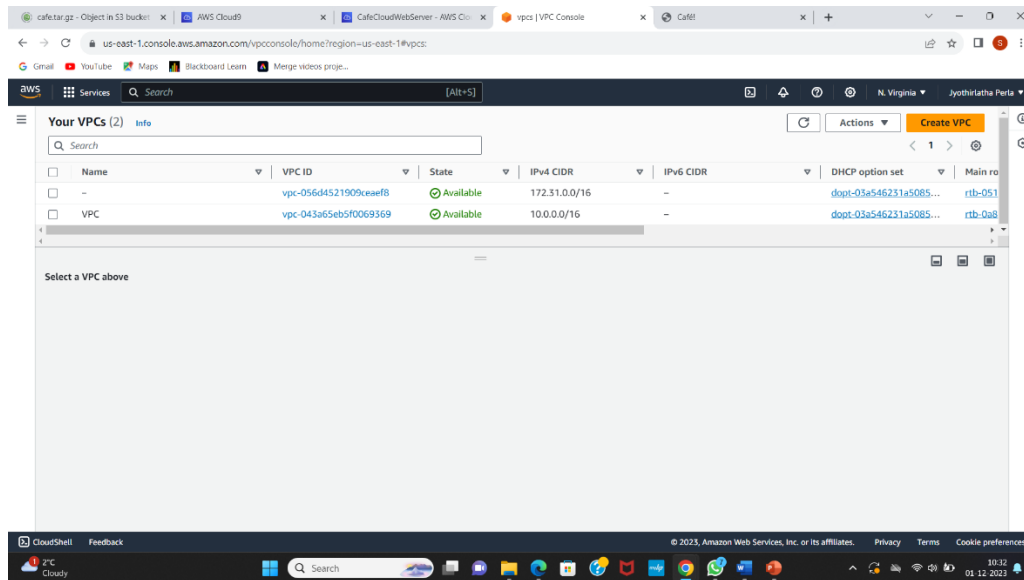
# Amazon S3



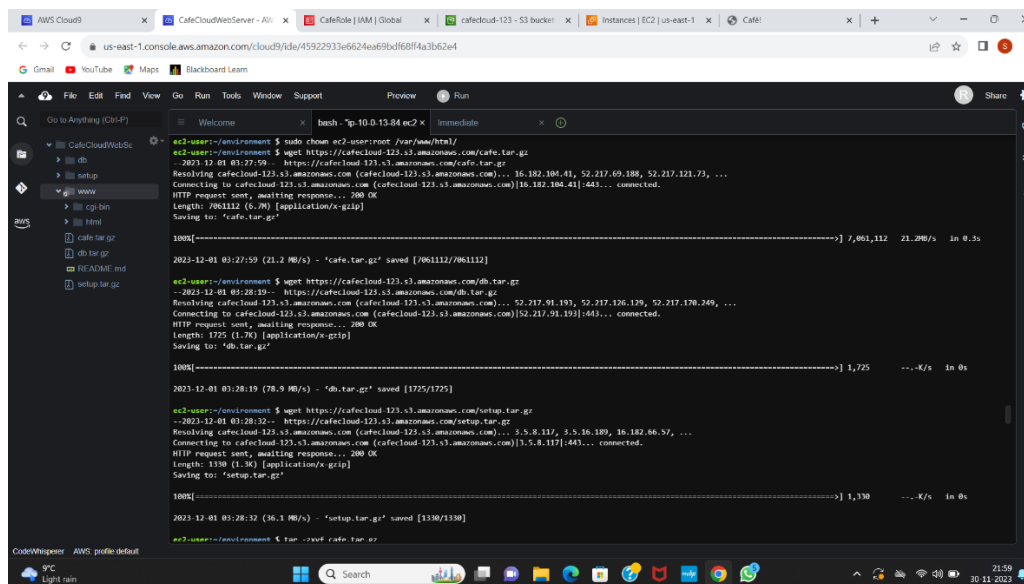
The screenshot shows the Amazon S3 console for the 'cafecloud-123' bucket. The 'Objects' tab is selected, displaying a list of objects. The objects are all of type 'gz' and 'Standard' storage class. The list includes objects for 'cafe.tar.gz', 'db.tar.gz', and 'setup.tar.gz'.

Name	Type	Last modified	Size	Storage class
cafe.tar.gz	gz	November 29, 2023, 14:12:13 (UTC-06:00)	6.7 MB	Standard
db.tar.gz	gz	November 29, 2023, 14:12:14 (UTC-06:00)	1.7 KB	Standard
setup.tar.gz	gz	November 29, 2023, 14:12:07 (UTC-06:00)	1.3 KB	Standard

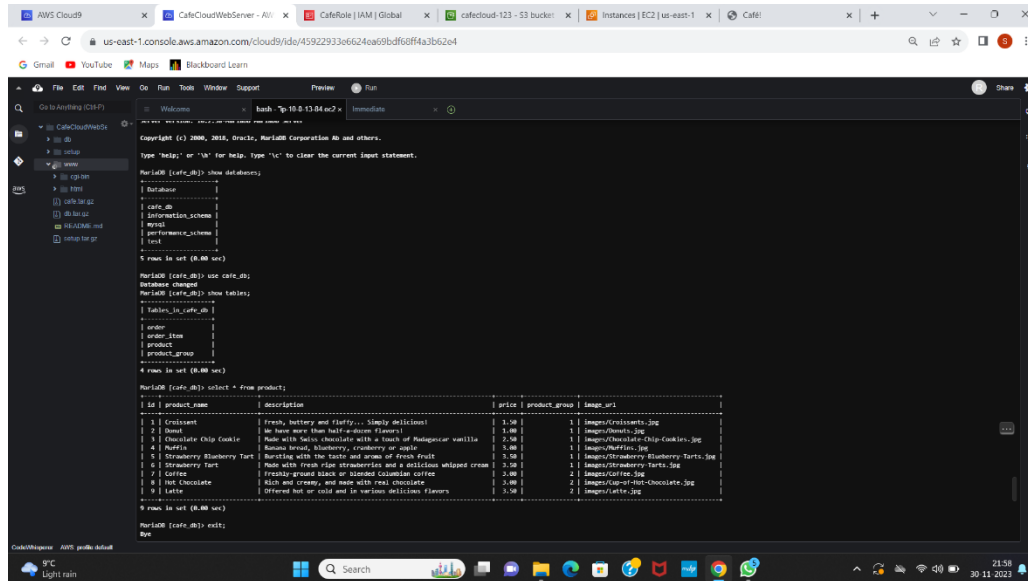
# Amazon VPC



## Retrieving and Unzipping our code, db and setup files



# Databases and Tables



```
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation AB and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

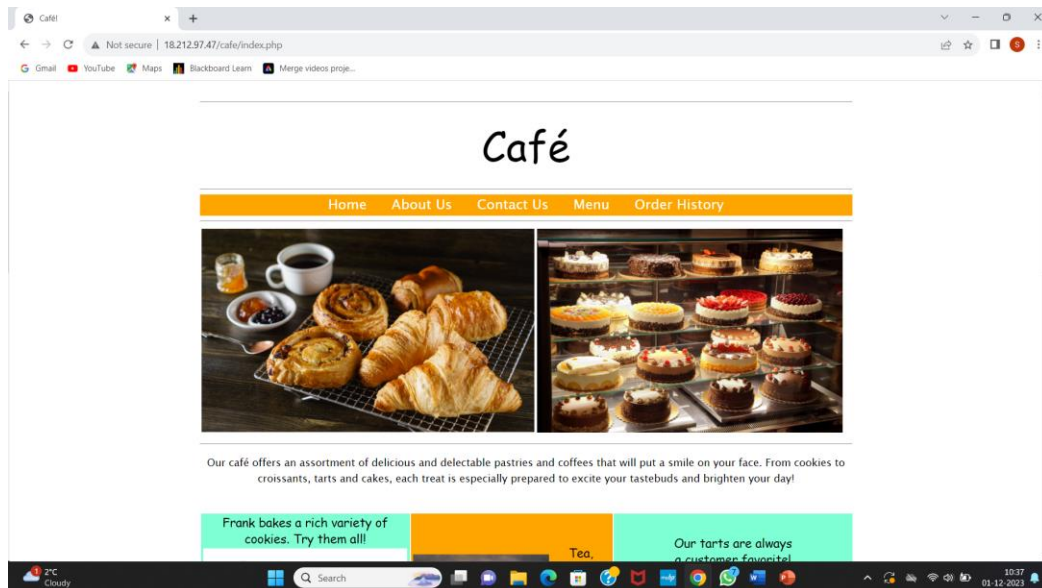
MariaDB [(cafe_db)] show databases;
+-----+
| Database |
+-----+
| cafe_db  |
+-----+

MariaDB [(cafe_db)] use cafe_db;
Database changed
MariaDB [(cafe_db)] show tables;
+-----+
| Tables_in_cafe_db |
+-----+
| order             |
| order_item        |
| product            |
| product_group     |
+-----+

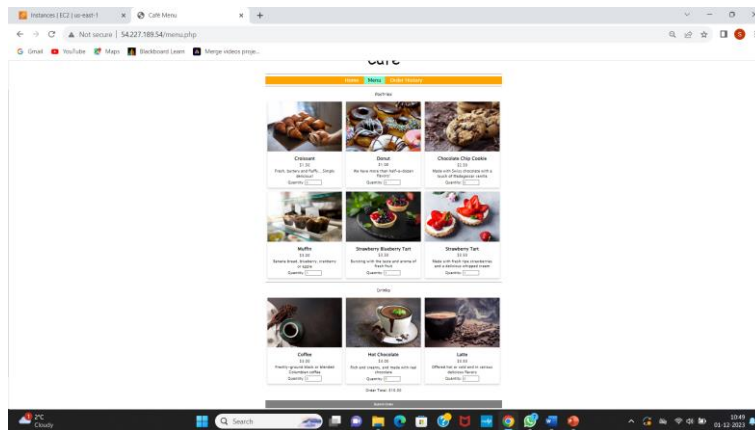
MariaDB [(cafe_db)] select * from product;
+----+ product_name | description | price | product_group | image_url |
+----+-----+-----+-----+-----+-----+
| 1 | Croissant | Fresh, buttery and flaky... Simply delicious! | 1.50 | 1 | images/Croissants.jpg |
| 2 | Tiramisu | We have many. Run, before we run! | 3.00 | 1 | images/Tiramisu.jpg |
| 3 | Chocolate Chip Cookie | Made with Swiss chocolate with a touch of Madagascar vanilla | 2.50 | 1 | images/Chocolate-Chip-Cookies.jpg |
| 4 | Tart | Seasonal fruit, blueberry, raspberry or apple | 3.00 | 1 | images/Tart.jpg |
| 5 | Strawberry Raspberry Tart | Bursting with the taste and aroma of fresh fruit | 3.50 | 1 | images/Strawberry-Raspberry-Tarts.jpg |
| 6 | Strawberry Tart | Made with fresh ripe strawberries and a delicious whipped cream | 3.50 | 1 | images/Strawberry-Tarts.jpg |
| 7 | Coffee | Freshly-ground black or blended Columbian coffee | 2.00 | 2 | images/Coffee.jpg |
| 8 | Hot Chocolate | Rich and creamy, and made with real chocolate | 2.00 | 2 | images/Hot-or-Cold-Chocolate.jpg |
| 9 | Latte | Offered hot or cold and in various delicious flavors | 3.50 | 2 | images/Latte.jpg |
+----+-----+-----+-----+-----+-----+

MariaDB [(cafe_db)] exit;
Bye
```

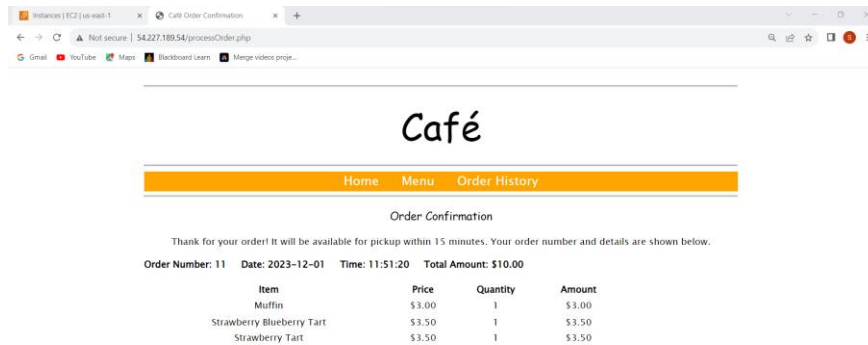
## Our Café Cloud Application



# Menu



## Clicking on Submit order would redirect to order confirmation



## Order History

