THOMPSON RIVERS UNIVERSITY

Department of Engineering

Faculty of Science

SENG 4620 - Practical Cloud Computing

Design Project: Building a Scalable and Global Cloud Infrastructure for a Local Business

This is a group project with a maximum of 3 students per team.

This semester, you'll go on a journey to build, secure, and scale the cloud infrastructure for a local business called *The Riverside Cafe*. Starting with no web presence, you'll guide The Riverside Cafe through a complete digital transformation by solving real-world business challenges at each step. The project is divided into eight distinct phases, each with its own deliverable.

Learning Objectives

- To architect a cloud infrastructure that meets evolving business requirements for scalability, security, cost-effectiveness, and reliability.
- To design and implement a defense-in-depth network security strategy by building a custom Virtual Private Cloud.
- To define and deploy an entire cloud infrastructure as code (IaC) using a version-controlled, automated CI/CD pipeline.
- To design a fault-tolerant architecture that uses load balancing and auto-scaling to handle dynamic traffic loads across multiple Availability Zones.
- To design and implement an event-driven, serverless workflow to solve a business problem in a cost-effective and operationally efficient manner.
- To synthesize and communicate complex technical architectures and design decisions through professional documentation, diagrams, and presentations.

The Project - Phase by Phase

Phase 1: The Launch and Global Presence (Due: End of Week 3)

- **Business Challenge:** The Riverside Cafe has no website and is invisible online.
- **Your Task:** Launch a simple, cost-effective, and reliable **static website** to put The Riverside Cafe on the map. You'll also establish data protection and disaster recovery plans from day one. (Note: The website files will be provided to you by a web developer.)

Phase 2: Dynamic Website with Online Ordering (Due: End of Week 4)

- **Business Challenge:** Customers want to place orders online. The static site can't handle this.
- **Your Task:** Evolve the architecture to host a **dynamic web application** on a virtual machine within a secure network that can process customer orders. (Note: The website files will be provided to you by a web developer.)

Phase 3: Building a Resilient Database (Due: End of Week 5)

- **Business Challenge:** The on-server database is a performance bottleneck and a management nightmare (backups, patching).
- **Your Task:** Decouple the database by migrating it from the web server to a **managed, relational database service** while improving security and resilience. (Note: The original database information and credentials will be provided by a database engineer.)

Phase 4: Fortifying the Network - Advanced Security (Due: End of Week 6)

- **Business Challenge:** The application server is still too exposed to the internet, and secure administrative access is needed.
- **Your Task:** Design and build a new, highly **secure network** from scratch, isolating servers in private subnets and implementing a bastion host for secure management access.

Phase 5: Architecting for Scale (Due: End of Week 8)

- **Business Challenge:** The Riverside Cafe is about to be featured on a TV show, anticipating a massive traffic spike that will crash the single server.
- **Your Task:** Re-architect the application tier for high availability and elasticity using a **load balancer and an auto-scaling group** across multiple Availability Zones.

Phase 6: Full Automation - Infrastructure as Code & CI/CD (Due: End of Week 9)

- **Business Challenge:** Manual deployments are slow, error-prone, and inconsistent, hindering expansion.
- **Your Task:** Define your entire infrastructure as code (**IaC**) and build an automated **CI/CD pipeline** to deploy changes reliably and quickly.

Phase 7: Optimizing with Serverless - Smart & Efficient (Due: End of Week 10)

- **Business Challenge:** A new daily sales report task is slowing down the main website and running a dedicated server for it is too expensive.
- **Your Task:** Solve the problem with a modern, event-driven approach. Build a cost-effective **serverless workflow** that runs the task on-demand without impacting performance.

Phase 8: Project Documentation and Presentation (Due: End of Week 12)

- **Business Challenge:** The project is a success, and now the work must be formally documented and presented for handover and knowledge sharing within the organization.
- **Your Task:** Create a comprehensive **final report**, a presentation, and a video demonstration that encapsulates the entire project journey and technical architecture.

Project Constraints

Throughout all phases of this project, your designs and implementations must consider the following cross-cutting constraints:

- **Cost-Effectiveness:** The Riverside Cafe is a small business with a limited budget. You must justify your selection of services and instance sizes based on a balance of performance and cost. Avoid over-provisioning.
- **Reliability:** The Riverside Cafe's website is a critical business tool. Your architecture must be resilient to failures, with clear plans for data protection and disaster recovery.
- **Security:** Your top priority is to protect The Riverside Cafe's data and infrastructure. You must adhere to the principle of least privilege, secure sensitive information, and build defense-in-depth.
- **Scalability:** While starting small, your designs should always consider future growth. The architecture must be able to scale efficiently to meet changing demands.

Teamwork

This project will be completed in groups of 2-3 students. You must inform the instructor of your team members via email by the **end of the first week**. For students who have not formed a team, the instructor will randomly create groups and post them on Moodle on the first day of the second week.

The most important part of a successful team is finding a time when everyone meets, as a group, each week, for at least two hours. Each student should expect to spend about four hours outside of class per week on the project. Compare your schedules to find a time block of at least two hours when you all can meet. Commit to meeting at this time each week and working together. If something comes up and someone can't make it, be sure to schedule another time for that week.

You will deliver your team charter during the first phase of the project. A team charter is a document of your own design. It should be "artfully" designed, expressing some interests and passions of your team. Things to discuss: What are your goals for the class? Who are your team members? What is your team mascot? How will the team celebrate triumphs? How will the team make important decisions? How will the team resolve conflicts and discuss problems? What does "leadership" mean to your team? Who is the person that hits "submit" on your reports and milestones? What are the skills of the team members? When will the team meet as a group each week? (Please be precise.) What will be the procedure for missing or being late to this meeting? How much advance notice must be given and using what method? A team charter template has been uploaded alongside the project files.

Assessment

This project is divided into eight phases. Each phase is detailed in a separate document. Phases 1 through 7 are each worth 100 marks. Phase 8 is worth 200 marks. The marking rubric for each phase is provided at the end of its respective document. Your total score will be out of 900 marks, which will then be scaled to a final project grade out of 100.

Project Phase 1: The Launch and Global Presence

Due Date: End of Week 3 **Points:** 100

Scenario

Your team has been hired as cloud engineers/architects for a small local business, "The Riverside Cafe", owned by David and Helen. The Riverside Cafe has a great local reputation but currently has no online presence. Their daughter, Emily, wants to create a simple website to showcase their desserts and coffees, and provide essential business information like location, hours, and contact details.

A web developer has designed a static website for Emily and will send you the files. However, Emily has a few concerns: Since David and Helen aren't very technical, Emily is worried about accidentally losing the website's content. As such, she wants to keep a history of all website versions and updated content. To avoid increasing storage costs, she wants to keep old versions of the files for only one month. In addition, the website is critical for the business, so it must be restorable if the primary cloud region experiences an outage.

Your first task is to establish this initial web presence. The goal is to launch a simple, reliable, and cost-effective static website while establishing best practices from the very beginning.

Project Goals

1. **Launch a Public Website.** The Riverside Cafe needs a publicly accessible website. The initial content will be static (HTML, CSS, images).
2. **Protect Website Data.** The website content will be updated frequently. A strategy must be in place to prevent accidental deletion or corruption of the website's files.
3. **Plan for the Future.** The solution must consider long-term cost management for storing multiple versions of files and have a basic disaster recovery plan.

Technical Requirements

- You are free to use any major cloud provider (AWS, Azure, Google Cloud).
- All resources you create must be documented.
- The provided source code for the website (`index.html` and an `images` folder) will be given to you. You are free to modify or enhance it.

Submission Instructions

You will submit a single PDF document to Moodle. This document must contain all parts of the deliverable outlined below. Please use clear headings for each section.

Deliverable Breakdown

- **Team Charter:** Include your completed team charter based on the Teamwork Guideline document provided in class.
- Provide the public, clickable URL to your live and functioning static website.
- Provide your Cloud Architecture Design.
- Provide a Data Protection Plan.
- Provide a Cost Optimization Plan.
- Provide a Disaster Recovery (DR) Plan.
- Provide a Justification. State which cloud provider and which specific object storage service (e.g., Amazon S3, Azure Blob Storage) you chose. Explain why this service is the ideal choice for hosting a static website compared to other options, such as using a virtual machine. Explain the steps you took to make the website's content publicly readable while keeping the storage administration secure. Include a code snippet of any access policies you created (e.g., an S3 bucket policy) and explain what the key statements in the policy do. Provide a high-level, step-by-step description of how you deployed the website files to your chosen storage service.

Grading Rubric

Criteria	**Description**	**Points**
Team Charter	The team charter is submitted and adheres to the provided guidelines.	10
Live Website	The provided URL leads to a functioning static website displaying The Riverside Cafe's content.	10
Architecture Justification	Clear justification for the choice of cloud provider and service. Explanation of security configuration is logical and demonstrates an understanding of public access vs. private administration.	20
Data Protection Plan	A suitable data protection feature (e.g., versioning) is identified, implemented, and clearly explained as a solution to the business problem.	15
Cost Optimization Plan	A logical lifecycle policy is designed and justified with respect to the business context.	20

Disaster Recovery Plan	A suitable cross-region replication strategy is implemented and the failover concept is explained.	15
Professionalism & Clarity	The submitted document is well-organized and clear. It also clearly discusses the team's working communication and individual contributions for the phase.	10
Total		**100**

Project Phase 2: Dynamic Website with Online Ordering

Due Date: End of Week 4 **Points:** 100

Scenario

The Riverside Cafe's new static website has been a hit! Customers love it, but they immediately started asking for more: the ability to place orders online. To meet this demand, The Riverside Cafe's development team has created a dynamic web application that can take orders and display an order history.

Your role as the cloud engineer-architect is to evolve the cloud infrastructure to support this new application. The simple object storage solution from Phase 1 is no longer sufficient. You need to provision a server environment that can run the application code and connect to a database.

Project Goals

1. The infrastructure must support a web application which has been built on a standard LAMP stack (Linux, Apache, MySQL/MariaDB, PHP).
2. The application server must be placed within a secure network, and sensitive information like database credentials must not be stored in the application's code.
3. To prepare for future growth and disaster recovery, you need to create a standardized, reusable image of your configured server.

Technical Requirements

- You will be provided with an application package (`Cafe-app.zip`) and a document with "Installation Notes" from the development team. These notes will contain the necessary commands to install the application and its dependencies on a standard Linux server. **You are not expected to write or debug the application code.**
- Your infrastructure must be defined within a custom virtual network (e.g., AWS VPC, Azure VNet). You should not use the default network provided by the cloud provider.
- All resources must be documented.

Submission Instructions

You will submit a single PDF document to Moodle. This document must contain all parts of the deliverable outlined below.

Deliverable Breakdown

- Provide a public URL to your live, functioning dynamic web application (e.g., `http://<your-server-ip>/Cafe`).
- Include a screenshot of the "Order History" page showing at least two test orders you have placed.

- **Virtual Network (VPC/VNet) Architecture Diagram:** Provide an architecture diagram of your virtual network. It must include the overall VPC/VNet CIDR block, at least one public and one private subnet, an Internet Gateway, and a NAT Gateway. Justify your choice of CIDR blocks and your subnetting strategy. Explain whether you placed the application's VM in the public or private subnet and justify your decision based on security.
- **Security Group Configuration:** Detail the inbound and outbound rules for the security group(s) associated with your VM. For each rule, explain its purpose, adhering to the principle of least privilege.
- **Secrets Management:** Describe the cloud service you used to manage this secret securely and how the VM was given permission to access it.
- **Reusable Server Image and Configuration:** Provide a high-level log of the key steps you took to configure the VM and deploy the application. What is the name of a reusable server image in your chosen cloud provider? Explain two major benefits of creating this image for scalability and disaster recovery. Provide a screenshot showing the created image.

Grading Rubric

Criteria	**Description**	**Points**
Live Application	URL and screenshot prove a functioning dynamic application.	10
Network Architecture	Diagram is clear, complete, and well-designed. Justifications for network and instance placement are sound.	30
Security Configuration	Security group rules are appropriate, well-explained, and follow the principle of least privilege. Secrets management strategy is sound and correctly implemented.	30
Golden Image	A server image is created and the purpose and benefits of it are clearly explained.	20
Professionalism & Clarity	The submitted document is well-organized and clear. It also clearly discusses the team's working communication and individual contributions for the phase.	10

Total		**100**

Project Phase 3: Building a Resilient Database

Due Date: End of Week 5 **Points:** 100

Scenario

The Riverside Cafe's online ordering system is a huge success, but the single server running both the web application and the database is struggling. The order history has become a valuable asset, but the database is difficult to manage (patching, backups) and poses a performance bottleneck.

Your task is to propose and implement a new architecture that addresses these critical business and operational challenges by decoupling the database.

Project Goals

1. Propose a new database solution that offloads the heavy lifting of database administration.
2. Separate the database from the web application server to improve security and allow each layer to scale independently.
3. Execute a seamless migration of all existing order data from the current database to the new solution with a clear plan.

Technical Requirements

- The new database solution must be a managed service (e.g., Amazon RDS, Azure SQL Database).
- The database must be placed in a **private subnet** to prevent direct access from the internet.
- You must migrate all existing data. When the migration is complete, the application should function exactly as before.
- The necessary information regarding the previous database setup (e.g., schema, data dump utilities) has been provided to you by The Riverside Cafe's development team.

Submission Instructions

You will submit a single PDF document to Moodle. This document must contain all parts of the deliverable outlined below.

Deliverable Breakdown

- **Problem Analysis:** In your own words, summarize the key business and technical challenges The Riverside Cafe is facing with its current database setup.
- **Proposed Solution and Architecture:** Identify the type of managed relational database service that solves these problems. Justify your choice, explaining how it addresses concerns about maintenance, backups, and specialized skills. Provide a **"Before" Architecture Diagram** of your architecture from Phase 2 and a new **"After" Architecture Diagram** that reflects your redesigned architecture. This diagram must show the web server, the new managed database instance in a private subnet, and the security group configurations allowing the connection.

- **Database Configuration Rationale:** What instance size did you choose for your database and why? Should you enable a high-availability feature (like Multi-AZ)? Justify your decision based on the trade-off between cost and reliability for The Riverside Cafe.
- **The Migration Plan:** Describe the step-by-step process you followed to migrate the data (mentioning tools like `mysqldump`). Explain how you reconfigured the web application to connect to the new database (e.g., updating the secret in Secrets Manager).
- **Verification:** How did you verify the migration was successful? Provide a screenshot of the "Order History" page *after* the migration, showing a new test order alongside all the previously migrated orders.

Grading Rubric

Criteria	**Description**	**Points**
Problem Analysis	Clear analysis of the existing problems and a well-justified proposal for a managed database solution.	10
Architecture Design	"Before" and "After" diagrams are clear and accurate. Rationale for instance size and high availability is well-reasoned.	30
Migration Plan	Migration process is clearly documented.	30
Verification	Reconfiguration and verification steps are explained, with screenshot proof of success.	20
Professionalism & Clarity	The submitted document is well-organized and clear. It also clearly discusses the team's working communication and individual contributions for the phase.	10
Total		**100**

Project Phase 4: Fortifying the Network - Advanced Security

Due Date: End of Week 6 **Points:** 100

Scenario

The Riverside Cafe's architecture is now more resilient, but a security audit reveals a major vulnerability: the application server is still in a public subnet, making it a direct target for attacks. A more secure, industry-standard architecture is required.

Your task is to design and implement a new, highly secure Virtual Private Cloud (VPC) environment from scratch.

Project Goals

1. **Design** a **secure** multi-tier VPC with appropriate public and private subnets to isolate resources.
2. **Implement** a **Secure Remote Access** solution, such as creating a bastion host (jump box) in the public subnet as the only entry point for SSH access to private instances.
3. **Enable Secure Outbound Traffic** by implementing a NAT Gateway to allow private instances to access the internet for patches without being exposed to inbound traffic.
4. **Apply Layered Security** using both security groups and network ACLs to create a defense-in-depth security posture.

Technical Requirements

- You must create a new VPC. Do not use the default VPC.
- Your application server (from Phase 2) must be launched in the private subnet.
- Your database (from Phase 3) should also be hypothetically placed in a private subnet in this new VPC.

Submission Instructions

You will submit a single PDF document to Moodle. This document must contain all parts of the deliverable outlined below.

Deliverable Breakdown

- **The Network Architecture:** Provide a detailed **Architecture Diagram** of your new VPC. It must clearly show your VPC/CIDR, public/private subnets, Internet Gateway, NAT Gateway, route tables, the Bastion Host, and the Application Server in their correct locations. For each of the following components, explain its function within your architecture: Internet Gateway, NAT Gateway, and Bastion Host.
- **The Security Groups:** Detail and justify the inbound/outbound rules for the security group attached to your **Bastion Host**. Detail and justify the inbound/outbound rules for the security

group attached to your **Application Server**, explaining how it only accepts SSH traffic from the bastion host.

- **Network ACL Design:** What is the key difference between a security group and a network ACL? Design and detail the rules for a custom network ACL for your private subnet. Explain how these rules provide an additional layer of security. Describe the exact, multi-step process an administrator would follow to connect from their computer to the private application server. Explain why this is more secure than direct access.

Grading Rubric

Criteria	**Description**	**Points**
Network Diagram	Diagram is detailed, accurate, and includes all required components in their correct places.	30
Component Justification	Explanations for IGW, NAT GW, and Bastion Host are clear and correct.	20
Security Group Design	Security group rules are well-designed for least-privilege access and are clearly justified.	20
Network ACL Design	Demonstrates understanding of NACLs vs. Security Groups. NACL rules are logical and add a meaningful security layer.	20
Professionalism & Clarity	The submitted document is well-organized and clear. It also clearly discusses the team's working communication and individual contributions for the phase.	10
Total		**100**

Project Phase 5: Architecting for Scale

Due Date: End of Week 8 **Points:** 100

Scenario

Incredible news! The Riverside Cafe is going to be featured on a famous TV food show. This is a massive opportunity, but it also presents a huge technical challenge. A massive, temporary spike in website traffic is expected, which will overwhelm the single application server.

Your task is to re-architect the application tier to be fully scalable and highly available, ensuring the website remains fast and responsive throughout the traffic surge.

Project Goals

1. **Ensure High Availability.** Deploy your architecture across at least two Availability Zones to withstand the failure of a single data center.
2. **Distribute the Load.** Implement a load balancer to act as the single entry point for all web traffic, distributing requests evenly.
3. **Implement Elasticity.** Create an auto-scaling mechanism that automatically launches and terminates application servers in response to demand.

Technical Requirements

- Your VPC must be configured to support a multi-AZ deployment (subnets in at least two AZs).
- Your application servers must be part of an Auto Scaling group managed by a launch template.
- The Application Load Balancer should be placed in the public subnets, forwarding traffic to the application servers in the private subnets.

Submission Instructions

You will submit a single PDF document to Moodle. This document must contain all parts of the deliverable outlined below.

Deliverable Breakdown

- **The Scalability & Availability Plan:** What are the two primary risks of The Riverside Cafe's current single-server architecture in this scenario? Describe the three core cloud services/features you will use to solve this problem (Load Balancing, Auto Scaling, Multi-AZ) and explain how each contributes to the solution.
- **The Resilient Architecture:** Provide a new, detailed architecture diagram showing your scalable design. It must illustrate the VPC spanning two AZs, the Application Load Balancer, the Auto Scaling group managing instances across both AZs, and the user traffic flow.
- **Launch Template Configuration:** Explain the purpose of a launch template and why it is critical for auto-scaling. Provide a screenshot of your launch template's configuration summary.

- **Auto Scaling Policy Design:** Detail and justify your choices for the Auto Scaling group's Desired, Minimum, and Maximum capacity. What metric did you choose to trigger scaling events (e.g., Average CPU Utilization)? Justify your choice of metric and target value.
- **Verification and Testing Plan:** Provide the public DNS name of your load balancer. Describe the process you would use to simulate load and test your auto-scaling setup. Provide screenshots showing the "scale-out" event: one showing the metric alarm was triggered, and another showing new instances launched by the Auto Scaling group.

Grading Rubric

Criteria	**Description**	**Points**
Analysis & Plan	Risks are correctly identified and the proposed solution using the three core services is well-explained.	20
Architecture Diagram	Diagram is detailed, accurate, and clearly shows the multi-AZ, load-balanced, auto-scaled architecture.	25
Launch Template	Purpose of the launch template is understood and configuration is shown.	10
Auto Scaling Policy	Scaling parameters and policy are well-justified for the scenario.	20
Testing & Verification	A clear testing plan is described, and screenshot evidence proves the system scaled as designed.	25
Total		**100**

Project Phase 6: Full Automation - Infrastructure as Code & CI/CD

Due Date: End of Week 9 **Points:** 100

Scenario

The Riverside Cafe's cloud architecture is now robust, but the deployment process is not. All infrastructure has been created manually, which is slow, error-prone, and makes it difficult to create consistent environments for testing or expansion into new regions.

Your task is to solve these operational problems by moving from manual "click-ops" to a fully automated, Infrastructure as Code (IaC) approach.

Project Goals

1. **Codify Your Infrastructure.** Define your entire networking and application architectures from the previous phases using an IaC tool (e.g., AWS CloudFormation, Terraform).
2. **Version Control Your Code.** Store your IaC templates in a source code repository (e.g., AWS CodeCommit, GitHub).
3. **Build an Automated Pipeline.** Create a CI/CD pipeline that automatically deploys or updates your infrastructure whenever a change is pushed to the repository.
4. **Demonstrate Reproducibility.** Prove the power of your IaC setup by rapidly deploying a complete copy of your environment into a new region.

Technical Requirements

- You must create at least two separate IaC templates: a **network template** and an **application template**.
- The application template must reference outputs from the network template.
- Your pipeline must be triggered by a commit to your source code repository.

Submission Instructions

You will submit a single PDF document to Moodle. This document must contain all parts of the deliverable outlined below.

Deliverable Breakdown

- **Infrastructure as Code:** State which IaC tool and version control system you chose and briefly explain why. Include the full code for your network infrastructure template. Include the full code for your application infrastructure template. Explain how your application template references your network template and why separating them is a good architectural practice.
- **The CI/CD Pipeline:** Provide a diagram illustrating your CI/CD pipeline, showing how a `git push` triggers the source, build, and deploy stages. Provide screenshots of your pipeline's configuration, showing the source and deployment stages. Show proof of a code change

triggering an automatic deployment. Include screenshots of a `git push` command and your pipeline showing the corresponding commit message in a "Succeeded" state.

- **Disaster Recovery & Replication:** Describe the process of using your IaC templates to deploy a complete copy of your infrastructure into a *new region*. Provide a screenshot showing the application stack successfully created in this second region.
- **IaC for Business Continuity:** In your own words, explain how having your infrastructure defined as code significantly improves The Riverside Cafe's disaster recovery strategy.

Grading Rubric

Criteria	**Description**	**Points**
IaC Templates	Templates are complete, well-structured, and correctly separated into modular network/application layers. The rationale for modularity is clear.	30
CI/CD Pipeline	The pipeline diagram is accurate. Configuration and successful execution triggered by a code change are proven with screenshots.	30
Cross-Region Deployment	The process for cross-region deployment is described correctly, and screenshot evidence is provided.	20
IaC Justification	The strategic benefit of IaC for disaster recovery is clearly and convincingly explained.	20
Total		**100**

Project Phase 7: Optimizing with Serverless - Smart & Efficient

Due Date: End of Week 10 **Points:** 100

Scenario

The Riverside Cafe owners now need a daily sales report emailed to them. The team's first attempt, running the report on the main web servers, slowed down the website. Their second idea, running a dedicated server just for the report, is too expensive for a task that only runs for a few minutes a day. They need a better way.

Your task is to design and implement a modern, cost-effective, serverless architecture to solve this reporting problem.

Project Goals

1. **Decouple the Reporting Task.** Isolate the report generation logic from the primary web application to ensure customer performance is never compromised.
2. Use a Function-as-a-Service (FaaS) offering to run the report-generation code on-demand, eliminating the need for a dedicated server.
3. **Automate with Event-Driven Triggers.** Implement a scheduler that automatically triggers the reporting process at a set time each day.
4. **Deliver the Report.** Use a notification service to email the final report to stakeholders.

Technical Requirements

- The compute function must run without a provisioned server (e.g., using AWS Lambda).
- The function must be placed within your VPC to securely access the production RDS database.
- The function must be triggered automatically on a daily schedule.
- You will be provided with the Python code for the function (`report-generator.zip`).

Submission Instructions

You will submit a single PDF document to Moodle. This document must contain all parts of the deliverable outlined below.

Deliverable Breakdown

- **Problem Analysis & Solution Design:** Explain the two main drawbacks of the initial ideas (running on existing web servers vs. a new dedicated server). Explain why a serverless FaaS approach is the ideal solution for this problem, focusing on **performance isolation** and **cost-effectiveness**.
- **The Serverless Architecture:** Provide a detailed diagram of your serverless workflow. It must show the scheduled trigger, the Lambda function, its integration into your VPC, the RDS Database, the Notification Service (e.g., SNS), and the final email delivery.

- **VPC Integration & Security:** Explain why the Lambda function needs to be associated with your VPC. Detail the security group configuration you created to allow the Lambda function to securely connect to the RDS database. Provide a screenshot of your Lambda function's final configuration (runtime, handler, VPC settings, etc.).
- **Automation and Verification:** What service did you use to schedule the daily execution? Provide the "cron expression" you used and explain what it means. Describe how you tested the complete workflow. Provide a screenshot of the final sales report email you received after a successful test run.

Grading Rubric

Criteria	**Description**	**Points**
Analysis & Solution	The analysis of the problem is accurate, and the justification for a serverless solution is strong and well-articulated.	25
Architecture Diagram	The diagram is clear, accurate, and includes all required components of the serverless workflow.	20
VPC & Security	The need for VPC integration is understood. The security group configuration is correct and well-explained.	25
Automation & Verification	The scheduling mechanism is correctly configured and explained. Screenshot evidence proves the entire workflow functions correctly.	30
Total		**100**

Project Phase 8: Documentation and Presentation

Due Date: End of Week 12 **Points:** 200

Scenario

The Riverside Cafe's digital transformation has been a remarkable success, thanks to your team's expertise. The owners are thrilled with the robust, scalable, and cost-effective infrastructure. As a final step, your consulting engagement is concluding. You must now create a complete documentation package and presentation to formally hand over the project to The Riverside Cafe's internal team and your own engineering managers, ensuring they understand the architecture and can maintain it going forward.

Phase Goals

1. Synthesize the knowledge gained throughout the semester into a comprehensive technical document.
2. Demonstrate proficiency in communicating complex cloud architectures and design decisions to a technical audience.
3. Create a professional-quality presentation and video demonstration that effectively showcases the project's journey and final state.

Technical Requirements

- Your final report must follow the provided **TRU Engineering Final Report Template**.
- Your presentation should adhere to the attached **Presentation Guideline** document.
- Your video demonstration should be a screen recording that is clear, concise, and walks through the live application, highlighting key architectural features.
- For all architecture diagrams, you can use online tools such as `app.diagrams.net`, which provides AWS service icons (in the “More Shapes” section, add the AWS libraries).

Submission Instructions

You will submit three separate files to Moodle:

- A final **Project Documentation** document (PDF).
- A **Presentation** file (PPTX or PDF).
- A **Demo Video** of your entire project (MP4 format or a link to an unlisted video on a streaming service).

Grading Rubric

Criteria	**Description**	**Points**

Final Project Documentation	The report is comprehensive, well-structured, technically detailed, and follows the provided template. It clearly documents the final architecture and the decisions made through all seven phases.	100
Presentation	The presentation is professional, clear, and effectively communicates the project's scope, challenges, solutions, and final architecture.	60
Demo Video	The video clearly and concisely demonstrates the functionality of the final, deployed application and infrastructure.	40
Total		**200**