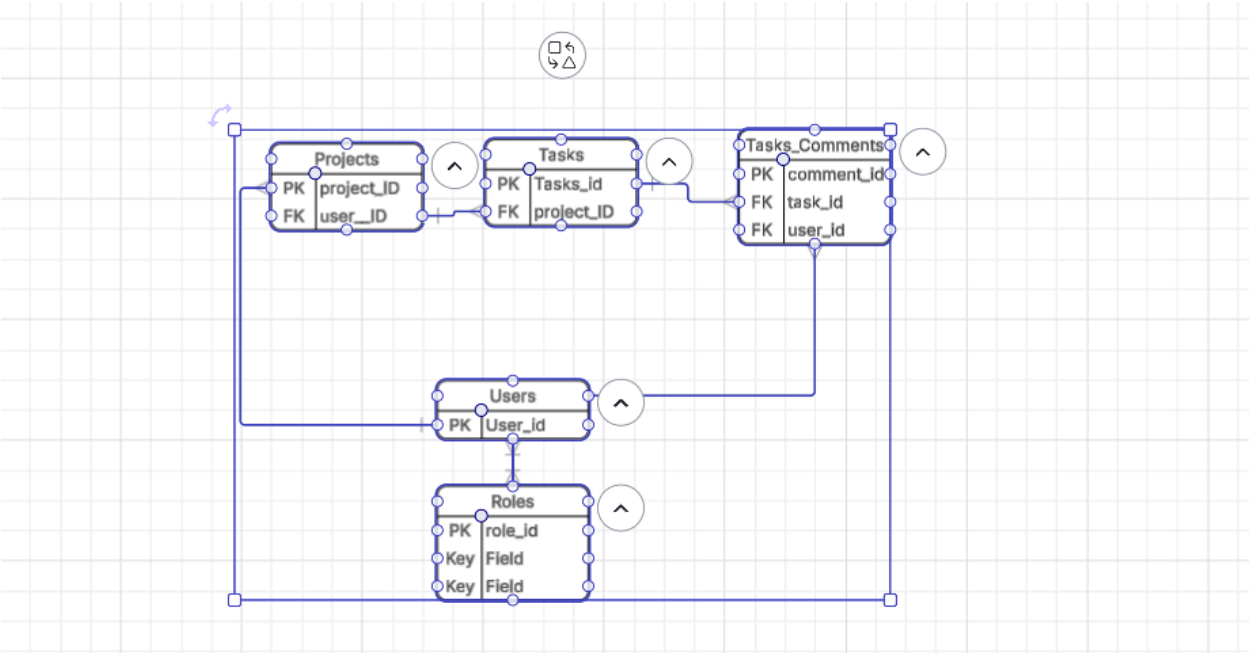
- Cloud Service Provider:
    - The Cloud Service provider that we will be utilizing for our project will be AWS. The reasoning behind this choice is because of the compact and simplistic nature that AWS provides for users.
    - Features utilized in this project will consist of mainly EC2 and RDS. S3 will

- Application Design:
    - Application's programming language
        - The programming language we will use for the project's application will mainly focus on JavaScript
    - Run-time environment
        - A Node.js environment will be used to run the application. Because Node is lightweight, scalable to manage numerous concurrent requests, and seamlessly integrates with popular CI/CD pipelines and AWS deployment patterns (EC2 + load balancer), it is a good choice for REST APIs.
    - Application API
        - The application will expose a RESTful API using standard HTTP methods . REST keeps the interface simple and consistent, supports stateless communication, and makes it easy for a future front-end (or tools like Postman) to interact with the backend.
    - Application framework
        - None, the application will not utilize a front-end framework such as angular or react.
    - Middleware
        - Express, to manage routing, handle HTTP requests, and process JSON payloads.

Operating System and Virtual Servers:

- The operating system utilized for the application will be **Windows.** This is for simplification in use for users and group members.
- server configuration (e.g., instance size, CPU, memory, storage).
    - Instance Type: t.3 small (2 vCPU & 2gb of Ram)
    - Storage**:** 20–30 GB EBS gp3 per instance

Database Design:



- The database management system selected for this application is **MySQL**, deployed using Amazon RDS for MySQL on AWS. If further needed, oracle can be used for SQL..

Network Architecture and Design:

- Illustrate the network architecture, including subnets (public and private), virtual private clouds (VPCs), and firewalls. The subnets need to be redundant for resiliency and high-availability purposes.
- Specify the configuration of security groups, including ingress and egress rules for the application and database. Be sure to include the ports necessary for basic trouble-shooting and administration, such as RDP for Windows servers and to support testing network connectivity through a ping command.
- **Port Configurations:**
    - 80 (HTTP) and/or 443 (HTTPS)

Data Visualization Tool:

- **Data Visualization tool: Tableau**
    - Tableau is a simplistic visualization tool. This will enable us to utilize a well known tool to assist in creating a visualization for our data.
    - Team familiarity and ease of use is important for this project.

Testing and Quality Assurance Process:

Define the testing and quality assurance processes that will be implemented during sprints to validate production code. Include the following:

Unit Testing:

- Utilize Jest for the API
- Validate request payloads, service methods, functions.

Integration Testing:

- Validate API + DB interactions using a test database
- Ensure migrations run and schema constraints behave as planned out.

End-to-End Testing: Strategy for ensuring the entire application behaves as expected from a user perspective.

- Test application to users outside of the project for assurance and feedback.

Authentication and Authorization Process:

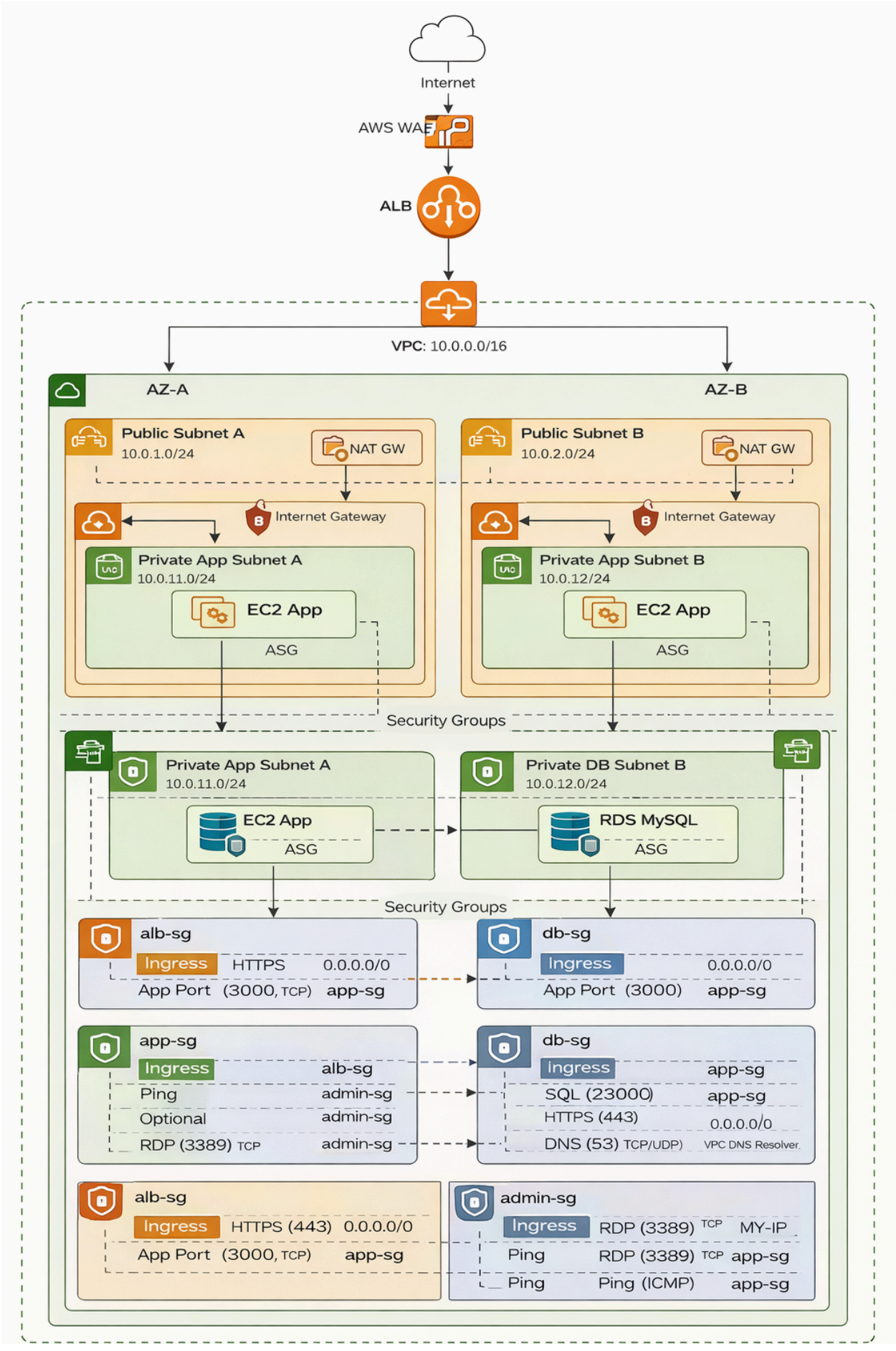
Specify whether anyone can access the system, or if the application needs to be restricted.

- This assignment is open to anyone required to view information
    - This includes the group members, students, administrators, etc.

Team Responsibilities:

- Each team member must contribute meaningfully to the project. Suggested roles include:
- **Cloud Architect:** Leads the selection and configuration of cloud resources.
- **Application Developer**: Designs the application's runtime environment, APIs, and port configurations.
- **Database Architect**: Develops the database schema and ensures compliance with third-normal form.
- **Network Engineer**: Designs the network architecture, including firewalls and subnets.
- **QA Analyst:** Develops the testing plan and tools for validating the system's functionality.
- **Project Manager**: Coordinates team activities and ensures timely submission of deliverables.

Kevin Coleto: **Project Manager and Network Engineer**
Andrew Park:Application Developer & Cloud Architect
Abdelhadi Maddi: Database Architect And QA Analyst


Internet
AWS WAF
ALB
VPC: 10.0.0.0/16
AZ-A
AZ-B
Public Subnet A
10.0.1.0/24
NAT GW
Internet Gateway
Private App Subnet A
10.0.11.0/24
EC2 App
ASG
Public Subnet B
10.0.2.0/24
NAT GW
Internet Gateway
Private App Subnet B
10.0.12/24
EC2 App
ASG
Security Groups
Private App Subnet A
10.0.11.0/24
EC2 App
ASG
Private DB Subnet B
10.0.12.0/24
RDS MySQL
ASG
Security Groups
alb-sg
Ingress HTTPS 0.0.0.0/0
App Port (3000, TCP) app-sg
db-sg
Ingress 0.0.0.0/0
App Port (3000) app-sg
app-sg
Ingress alb-sg
Ping admin-sg
Optional admin-sg
RDP (3389) TCP admin-sg
db-sg
Ingress app-sg
SQL (23000) app-sg
HTTPS (443) 0.0.0.0/0
DNS (53) TCP/UDP VPC DNS Resolver
alb-sg
Ingress HTTPS (443) 0.0.0.0/0
App Port (3000, TCP) app-sg
admin-sg
Ingress RDP (3389) TCP MY-IP
Ping RDP (3389) TCP app-sg
Ping Ping (ICMP) app-sg