**📄 Project Documentation: AWS + Pulumi Infrastructure Deployment & MySQL Setup**

**🔹 Objective**

The goal of this project was to design and deploy an AWS infrastructure using **Pulumi with TypeScript**, configure a private MySQL database server, and verify application connectivity using a non-root database user.

**✅ Task 1: AWS VPC & Networking Setup**

**What we did:**

* Created a **Virtual Private Cloud (VPC)** with a **10.0.0.0/16** CIDR block.
* Provisioned:
  + **1 public subnet** (for bastion/management access).
  + **1 private subnet** (for MySQL database server).
* Configured:
  + **Internet Gateway** for the public subnet.
  + **NAT Gateway** for private subnet outbound access.
  + Routing tables for proper traffic flow.

**Purpose:** Ensures secure networking where only the bastion host is public, and the MySQL server stays private.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**✅ Task 2: Bastion Host Deployment**

**What we did:**

* Deployed an **EC2 instance (Amazon Linux/Ubuntu)** in the **public subnet**.
* Configured **Security Groups**:
  + Allowed SSH access from our IP.
  + Allowed MySQL traffic **only from the bastion host**.
* Used this bastion as a **jump server** to access private resources.

**Purpose:** Acts as a secure entry point to reach the private MySQL server without exposing it publicly.

**✅ Task 3: Private MySQL Server Deployment**

**What we did:**

* Deployed an **Ubuntu 24.04 EC2 instance** in the **private subnet**.
* Installed and configured **MySQL Community Server**.
* Ensured:
  + Data directory initialized.
  + Service enabled and running.
  + Proper permissions for MySQL user mysql.

**Purpose:** Hosts the relational database securely inside the private network.

A screenshot of a computer

AI-generated content may be incorrect.

**✅ Task 4: MySQL User & Database Setup**

**What we did:**

* Logged in as root user (via bastion).
* Created a dedicated application user:

CREATE USER 'appuser'@'%' IDENTIFIED BY 'AppUserPassword123!';

CREATE DATABASE appdb;

GRANT ALL PRIVILEGES ON appdb.\* TO 'appuser'@'%';

FLUSH PRIVILEGES;

* Verified that appuser cannot perform root-level actions.

**Purpose:** Implements **least-privilege access** for applications.

**✅ Task 5: Connectivity Verification**

**What we did:**

* From bastion host, connected using:

mysql -u appuser -pAppUserPassword123! -h <PRIVATE\_DB\_IP> -e "SHOW DATABASES;"

* Verified that:
  + The connection works.
  + appuser can see appdb.
  + Unauthorized databases/actions are restricted.

**Purpose:** Confirms database access works only via authorized user & subnet.

A screenshot of a computer program

AI-generated content may be incorrect.

**✅ Task 6: Automation with Pulumi**

**What we did:**

* Wrote **Pulumi Infrastructure as Code (IaC)** in **TypeScript** to automate:
  + VPC, subnets, IGW, NAT.
  + Security Groups (for bastion and DB).
  + EC2 provisioning (bastion + private DB).
  + User data/bootstrap scripts for MySQL installation.
* Infrastructure is **repeatable, idempotent, and version-controlled**.

**Purpose:** Ensures **cloud resources are deployed consistently** and reduces manual configuration errors.

A screenshot of a computer

AI-generated content may be incorrect.