**Task 1 — Create a Secure VPC**

**Goal:** Provision a VPC with one public and one private subnet plus routing for egress.

**Requirements:**

1. VPC CIDR: 10.0.0.0/16
2. Public subnet: 10.0.1.0/24 in an AZ of your choice.
3. Private subnet: 10.0.2.0/24 in the same AZ.
4. Internet Gateway attached to the VPC.
5. Public Route Table with default route (0.0.0.0/0) to the IGW; associate with the public subnet.
6. NAT Gateway in the public subnet with its own Elastic IP.
7. Private Route Table with default route to the NAT Gateway; associate with the private subnet.

**Evidence to submission (Screenshots): A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screen shot of a computer code

AI-generated content may be incorrect.**

**Task 2 — Bastion Host in Public Subnet**

**Goal:** Launch a hardened bastion EC2 instance for SSH jump access.

**Requirements:**

1. EC2 in the public subnet with a public IPv4.
2. Security Group (bastion-sg): allow inbound TCP/22 only from your public IP.
3. System hardening basics via User Data:
   * Create a non‑root sudo user ops with SSH key injected from Pulumi config project:sshPublicKey.
   * Disable root SSH login and password authentication in sshd\_config.

**Evidence to submission (Screenshots):**

**A computer screen shot of a computer code

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.**

**Task 3 — Private EC2 Instance**

**Goal:** Launch an EC2 instance without public IP in the private subnet.

**Requirements:**

1. EC2 in private subnet, SG (app-sg) must allow inbound SSH (22) **only** from bastion-sg (security group reference), not from the internet.

**Evidence to submission (Screenshots):**

**A computer screen shot of a computer code

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.**

**Task 4 — Install & Manage MySQL with systemd**

**Goal:** Deploy MySQL on the private instance and manage it as a systemd service.

**Requirements:**

1. Use User Data (cloud-init) or a remote‑command provisioner to:
   * Install MySQL (community server) or MariaDB (acceptable).
   * Configure to listen on 127.0.0.1 and the instance’s private IP.
   * Create a database appdb and user appuser with a generated password.
2. Ensure service is **enabled** and **started** with systemd and survives reboot.

**Evidence to submission (Screenshots):**

**A screenshot of a computer program

AI-generated content may be incorrect.A white background with black text

AI-generated content may be incorrect.**

**Task 5 — End‑to‑End Connectivity Validation**

**Goal:** Prove secure connectivity and correct routing.

**Checks:**

From private instance → internet: curl https://aws.amazon.com works

**Evidence to submission (Screenshots):**

A close-up of a white screen

AI-generated content may be incorrect.

**Task 6 — Clean Infrastructure Teardown**

**Goal:** Demonstrate IaC lifecycle by destroying resources cleanly.

**Requirements:**

* Show pulumi destroy output completing without orphaned resources.

**Evidence to submission (Screenshots):**

**A screenshot of a computer

AI-generated content may be incorrect.**