Project ATTN - Phase 8: Deployment Architecture Report

Date: June 26, 2025
Subject: Completion of Final Architectural Configurations for the Production Environment
Status: READY FOR DEPLOYMENT

1. Overview

This report documents the successful completion of the final and most critical architectural configurations that make the project ready for deployment to a virtual machine (VM), following the completion of Phase 7 ("Production Hardening"). In this phase, two separate projects (the Main Application and the Face Recognition Microservice) have been integrated to operate as a single, unified system, with enhanced security and adherence to production environment standards.

The system is no longer just a development prototype but possesses a robust and secure architecture, ready to handle real user load.

2. Completed Architectural Enhancements

2.1. Unified Network Architecture (Shared Docker Network)

- **Implementation:** The main application and the microservice projects were configured to run on a single, shared, and named Docker network called attn_shared_network. This network was marked as external: true in the microservice project's docker-compose.yml file.
- **Result:** This allows the containers of both projects to communicate with each other directly and securely via container names, such as attn_redis or api-gateway, as if they were on the same machine. The need for temporary solutions like localhost or host.docker.internal has been completely eliminated.

2.2. Service Isolation and Security Hardening

- **Implementation:** The ports definitions for the api-gateway and microservice-nginx services in the microservice project were removed from the docker-compose.yml file.
- **Result:** These services are now completely isolated from the outside world. They can only be accessed by other authorized services within the same Docker network (e.g., the main application's Nginx or the application itself). This is a critical security measure that significantly reduces the attack surface.

2.3. Production-Grade Process Management (Gunicorn)

- **Implementation:** Both the backend-api service in the main application and the api-gateway service in the microservice were updated to run with the production-standard gunicorn -k uvicorn.workers.UvicornWorker ... command,

replacing the development-focused uvicorn --reload command.
- **Result:** This change places the applications behind a process manager. In the event of an unexpected crash of any worker process, Gunicorn automatically starts a new, healthy process, ensuring the application's **uninterrupted** operation. This exponentially increases the overall stability and reliability of the system.

2.4. Central Entry Point (Nginx Reverse Proxy)

- **Implementation:** An Nginx service named main-app-nginx was added to the main application's docker-compose.yml file. This Nginx listens on port 80, the sole entry point to the outside world, and forwards all incoming traffic to the attn_backend_api service on the internal network.
- **Result:** Our main application is no longer directly exposed to the internet. All requests are received and managed by Nginx, which is optimized for performance and security.

3. Pre-Deployment Checklist and Conclusion

The application is fully ready to be installed and launched on the VM. A final checklist before starting:

- **[ ] .env Files:** Ensure the .env files in both projects contain the correct and secure values for the production environment.
- **[ ] VM Firewall:** Verify that traffic to ports 80 (HTTP) and 443 (HTTPS) is allowed in the Google Cloud VM firewall rules. All other ports (5433, 6379, 8001, etc.) should be closed to the public.
- **[ ] DNS A Record:** Check that the A Record for the domain (e.g., api.your-domain.com) correctly points to the VM's static IP address.
- **[ ] Startup Procedure:**
  1. First, run docker-compose up --build -d for the main application to create the shared network.
  2. Then, run docker-compose up --build -d for the microservice project to connect to the existing network.

Upon completion of these steps, it is confirmed that the project has evolved from just a working application into a **secure, scalable, resilient, and easily maintainable professional system.**

Congratulations!