

List of the AWS services and resources needed for deployment.

To deploy the FastAPI Python app on AWS with a simple, cost-efficient, and secure design, the following services and resources will be required:

1. Amazon EC2 (Elastic Compute Cloud)

- An instance to host the Dockerized FastAPI application and SQLite database.
- The instance will be configured with a public IP to allow direct access to the application.

2. Amazon VPC (Virtual Private Cloud)

- To isolate resources in a secure default VPC network.

3. Security Groups:

- To define incoming HTTP (port 80) and SSH (port 22) traffic rules.

4. IAM (Identity and Access Management)

- To manage user and resource permissions securely.

5. Amazon S3 (Simple Storage Service)

- To store static files like architecture diagrams, logs, or backups.

6. Amazon CloudWatch

- To monitor application performance and log errors.
-

Architecture Diagram

Here's a simple explanation with architecture design of how the user request flows through the AWS infrastructure:

1. User Sends HTTP Request:

- A user accesses the application using the EC2 instance's public IP.
- Example: `http://<public-ip>/count`.

2. Request Reaches the FastAPI Application:

- The fast API app deployed in EC2 receives the request and identifies the `/count` endpoint.

3. SQLite Database Interaction:

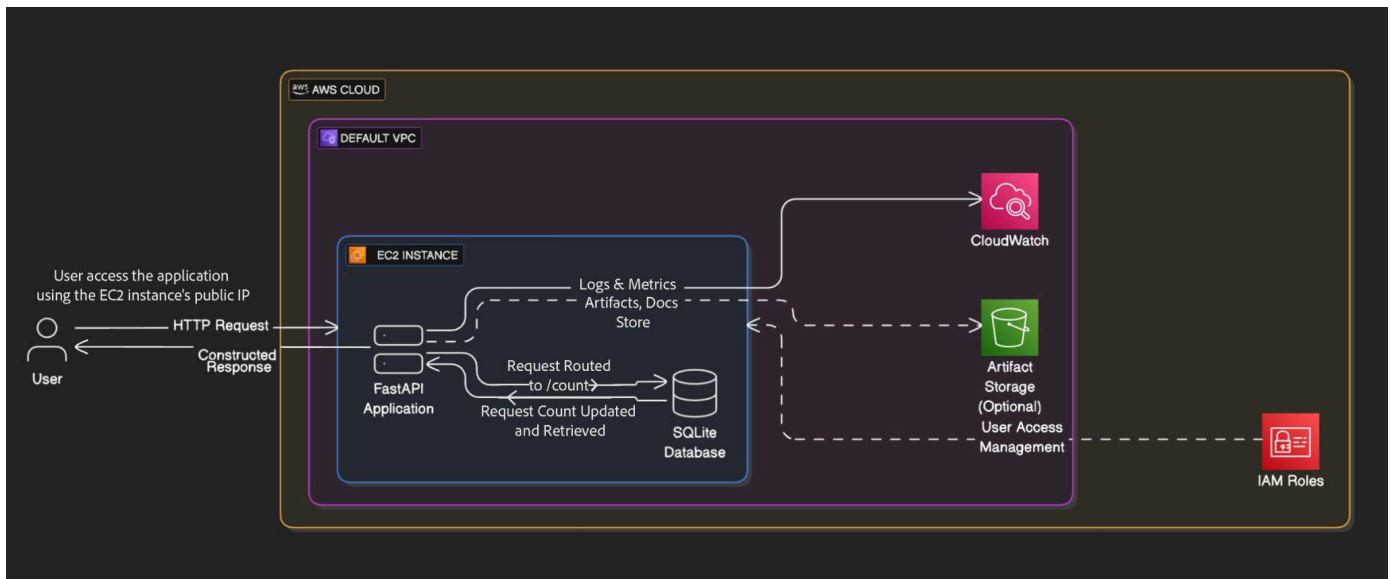
- The app increments the request count in the SQLite database and retrieves the updated count.

4. Response Construction:

- The app sends back a JSON response to the user.

5. Logs Sent to CloudWatch:

- The application and system logs are sent to CloudWatch for monitoring and troubleshooting.



Deployment Documentation

1. Launch the EC2 Instance

Navigate to the **EC2 Dashboard** and launch an instance:

- Use a Ubuntu 22.04
- Select an instance type t2.micro, cost-effective).
- Enable **public IP assignment** for the instance.

2. Configure the Security Group:

- Allow inbound traffic for:
 - **Port 80** (HTTP) for the application.
 - **Port 22** (SSH) for remote access for security.

3 . Attach an IAM Role:

- Create an IAM Role for users for security with CloudWatch access.
- Attach this role to the EC2 instance

4. Dockerize the Application

- Build the Docker image locally:
`docker-compose up --build`
- Push the image to Docker Hub.
`docker tag fast-api-app itsbijaya/fastapi-app:latest`
`docker push itsbijaya/fastapi-app:latest`

5. Pull the Dockerize Application

- SSH into the EC2 instance.
- Install the Docker and Docker Compose
- Push the image to Docker Hub
`docker pull itsbijaya/fastapi-app:latest`
- Run the Docker Image
`docker run -d -p 80:8000 --name fastapi-container itsbijaya/fast-api-app:latest`

6. Access the Application

- Access the app using the EC2 public IP:
`http://<EC2-public-ip>/count`

