Architecture Overview

This deployment strategy implements a **modern, serverless, cloud-native architecture** on AWS that uses AWS managed services to minimize infrastructure overhead while maximizing scalability, reliability, and security.

1. API Deployment:

- **Container-based**: Flask API packaged in Docker and stored in Amazon ECR
- **Serverless compute**: Runs on Amazon ECS with AWS Fargate (no server management)
- **Auto-scaling**: Application Load Balancer distributes traffic and scales containers based on demand

2. Database Management:

- **Fully managed**: Amazon RDS PostgreSQL with automated backups, patching, and maintenance
- **High availability**: Multi-AZ deployment for instant failover protection
- **Security-first**: Isolated in VPC with credentials managed by AWS Secrets Manager
- **Performance**: Built-in monitoring and optimization capabilities

3. Data Ingestion:

- **Scheduled processing**: Amazon EventBridge triggers data ingestion jobs
- **Serverless execution**: AWS Lambda functions process weather data from S3
- **Error resilience**: Dead Letter Queues ensure no data loss during failures
- **Cost-effective**: Pay-per-execution model with automatic scaling

Supporting Infrastructure

- **CI/CD Pipeline**: Automated deployment using AWS CodePipeline and CodeBuild
- **Monitoring**: CloudWatch and X-Ray provide comprehensive observability
- **Security**: AWS WAF protection with VPC network isolation
- **Cost Optimization**: Serverless and managed services reduce operational costs

Key Benefits

- **Scalability**: Automatically handles traffic spikes and data processing loads
- **Reliability**: Multi-AZ redundancy and automated failover mechanisms
- **Security**: Defense-in-depth approach with encryption and network isolation
- **Cost-Efficiency**: Pay-only-for-what-you-use serverless model
- **Operational Excellence**: Minimal infrastructure management with maximum automation