

F&B AI Platform — Deployment & Infrastructure Architecture

Cloud-Native, Scalable Deployment Strategy / AWS + GCP + 3rd-party Services

1. Cloud Infrastructure Overview



```

GCP --> DocumentAI

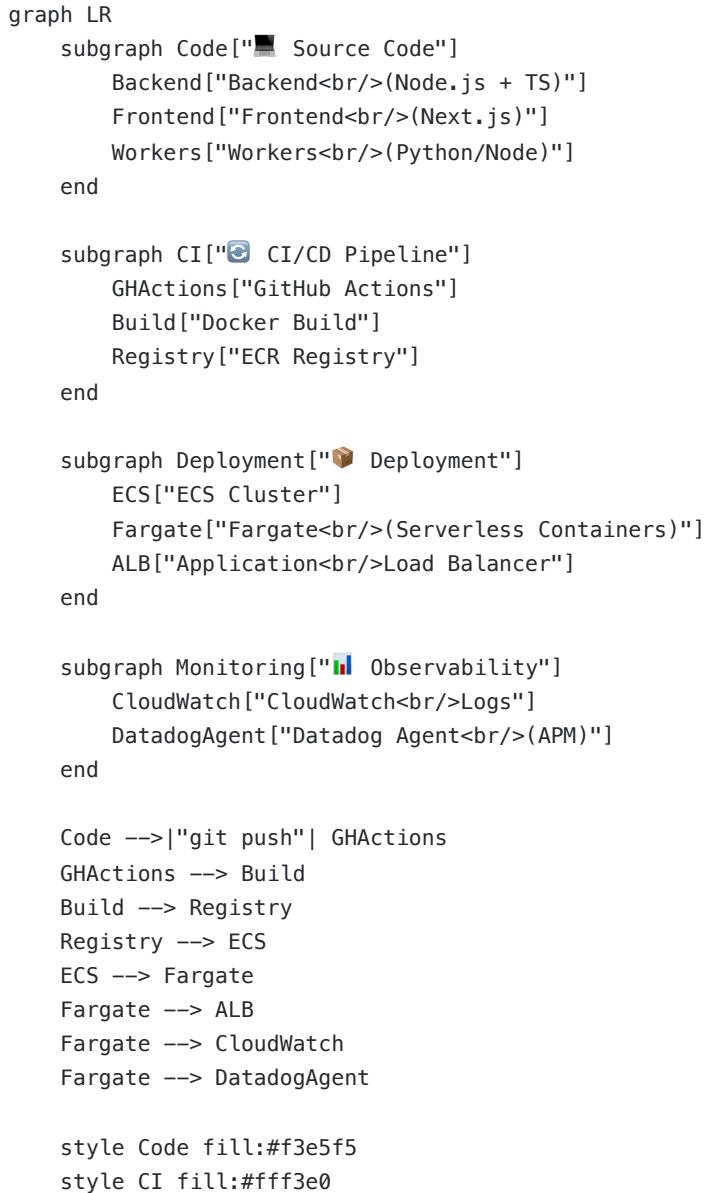
Managed -.->|Optional| Compute
Partners -.->|Integrations| Compute

style AWS fill:#ff9900,color:#000
style GCP fill:#4285f4,color:#fff
style Managed fill:#1565c0,color:#fff
style Partners fill:#f57c00,color:#fff

```

2. Application Deployment Architecture

2.1 Containerized Services



```

style Deployment fill:#e8f5e9
style Monitoring fill:#e0f2f1

```

2.2 Service Architecture (Microservices)

```

graph TD
    subgraph Services ["🔧 Microservices"]
        API["API Service<br/>(Express + GraphQL)<br/>Port: 3000"]
        Admin["Admin Service<br/>(Next.js Admin)<br/>Port: 3001"]
        Webhook["Webhook Service<br/>(POS/3rd-party)<br/>Port: 3002"]
        Worker["Background Worker<br/>(Bull Queues)<br/>Port: 3003"]
    end

    subgraph Networking ["🌐 Networking"]
        ALB["Application<br/>Load Balancer"]
        DNS ["Route 53<br/>(DNS)"]
        WAF["AWS WAF<br/>(Security)"]
    end

    subgraph Cache ["⚡ Caching Layer"]
        Redis["Redis Cloud"]
    end

    subgraph Data ["🗄 Data Layer"]
        RDS["PostgreSQL<br/>Primary"]
        RDSReplica["PostgreSQL<br/>Read Replica"]
    end

    DNS --> WAF
    WAF --> ALB
    ALB --> Services
    Services --> Redis
    Services --> RDS
    Services --> RDSReplica

    style Services fill:#f3e5f5
    style Networking fill:#e8f5e9
    style Cache fill:#fff3e0
    style Data fill:#e0f2f1

```

3. Database Architecture

3.1 PostgreSQL Clustering

```

graph TB
    subgraph Primary ["🗄 Primary DB"]
        PGPRI["PostgreSQL Primary<br/>eu-west-1"]
    end

```

```

subgraph Replicas ["📁 Read Replicas"]
    PGRR1 ["Read Replica 1<br/>eu-west-1"]
    PGRR2 ["Read Replica 2<br/>eu-central-1"]
    PGRR3 ["Read Replica 3<br/>us-east-1"]
end

subgraph Backup ["⌚ Backup"]
    PGBK ["Automated Backups<br/>(Daily)"]
    S3BAK ["Backup to S3<br/>(Long-term)"]
end

PGPRI -->|"Replication"| PGRR1
PGPRI -->|"Replication"| PGRR2
PGPRI -->|"Replication"| PGRR3

PGPRI -->|"Snapshot"| PGBK
PGBK --> S3BAK

style Primary fill:#e8f5e9
style Replicas fill:#c8e6c9
style Backup fill:#a5d6a7

```

3.2 Vector DB (Weaviate) Architecture

```

graph TD
    subgraph Weaviate ["🔎 Weaviate Cluster"]
        Node1 ["Node 1<br/>(Leader)"]
        Node2 ["Node 2<br/>(Shard 1)"]
        Node3 ["Node 3<br/>(Shard 2)"]
    end

    subgraph Classes ["📝 Data Classes"]
        SKUs ["NormalizedSKU<br/>Class"]
        Suppliers ["SupplierCatalog<br/>Class"]
        Policies ["Policies<br/>Class"]
    end

    subgraph Embeddings ["🧠 Vector Embeddings"]
        OpenAI ["OpenAI<br/>text-embedding-ada-002<br/>(1536-dim)"]
    end

    Node1 -->|"Coordinates"| Node2
    Node1 -->|"Coordinates"| Node3
    Node1 --> Classes
    Node2 --> Classes
    Node3 --> Classes
    Classes --> Embeddings

    style Weaviate fill:#f3e5f5

```

```

style Classes fill:#e0f2f1
style Embeddings fill:#fff3e0

```

4. LangGraph Agent Execution Environment

4.1 Agent Orchestration on Kubernetes

```

graph TD
    subgraph K8s ["🌐 Kubernetes Cluster"]
        subgraph Namespace ["Agents Namespace"]
            DeployAgent["Deployment:<br/>LangGraph Agent Pod<br/>(replicas: 3)"]
            StatefulSet["StatefulSet:<br/>Agent State Store<br/>(Persistent Vol)"]
            end

        subgraph Services ["K8s Services"]
            Service["Service:<br/>Agent Mesh<br/>(Load Balanced)"]
            end

        subgraph ConfigMaps ["ConfigMaps"]
            Models["LLM Model Names<br/>(GPT-4, Claude)"]
            Tools["Tool Definitions<br/>(Agent Capabilities)"]
            end
        end

        subgraph Queue ["📌 Job Queue"]
            BullQueue["Bull Queue<br/>(Redis-backed)"]
            end

        subgraph Database ["💾 State Storage"]
            PG["PostgreSQL<br/>Agent State Graphs"]
            end

        Queue --> DeployAgent
        DeployAgent --> Service
        DeployAgent --> Database
        Models --> DeployAgent
        Tools --> DeployAgent

        style K8s fill:#1565c0,color:#fff
        style Namespace fill:#90caf9
        style Queue fill:#fff3e0
        style Database fill:#e8f5e9
    
```

4.2 Agent Scaling Strategy

```

graph LR
    A["📥 Incoming AI Tasks<br/>from Event Bus"] --> B["🌐 Load Balancer"]
    B --> C["📊 Monitor Task Queue<br/>Current: 150 tasks<br/>Avg latency: 2.1s"]

```

```

C --> D{Scale<br/>Decision}

D -->|"Queue depth > 500"| E["⬆️ Scale UP<br/>Deploy +2 pod replicas<br/>Total: 5 pods"]
D -->|"Queue depth < 100"| F["⬇️ Scale DOWN<br/>Remove 1 pod replica<br/>Total: 2 pods"]
D -->|"OK (100–500)"| G["🔄 No change<br/>Maintain 3 pods"]

E --> H["✅ Load Balanced"]
F --> H
G --> H

H --> I["⌚ Agent Tasks<br/>Processing in parallel"]

style C fill:#fff3e0
style D fill:#ffd93d
style H fill:#c8e6c9

```

5. Security & Compliance Architecture

5.1 Network Security

```

graph TB
    subgraph Internet ["🌐 Internet"]
        Users["Users"]
        Suppliers["Suppliers"]
    end

    subgraph CloudFlare [".cloudflare CDN"]
        WAF["Web Application Firewall"]
        DDoS["DDoS Protection"]
    end

    subgraph AWSVPC ["AWS VPC"]
        IGW["Internet Gateway"]
        PublicSubnet["Public Subnet<br/>(ALB, NAT)"]
        PrivateSubnet["Private Subnet<br/>(ECS, RDS)"]
    end

    subgraph Security ["🔒 Security Groups"]
        ALBSG["ALB Security Group<br/>Port 443 only"]
        ECSSG["ECS Security Group<br/>Port 3000 (internal)"]
        RDSSG["RDS Security Group<br/>Port 5432 (ECS only)"]
    end

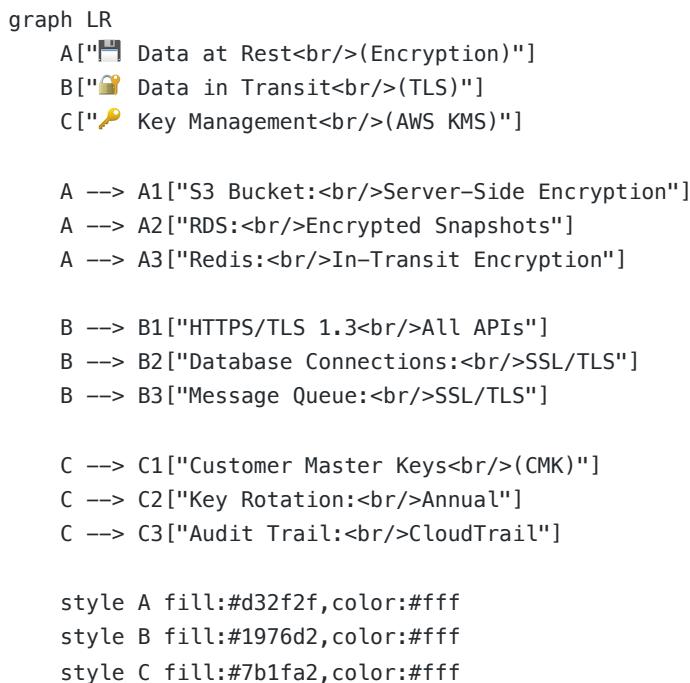
    Users --> CloudFlare
    Suppliers --> CloudFlare
    CloudFlare --> WAF
    WAF --> DDoS

```

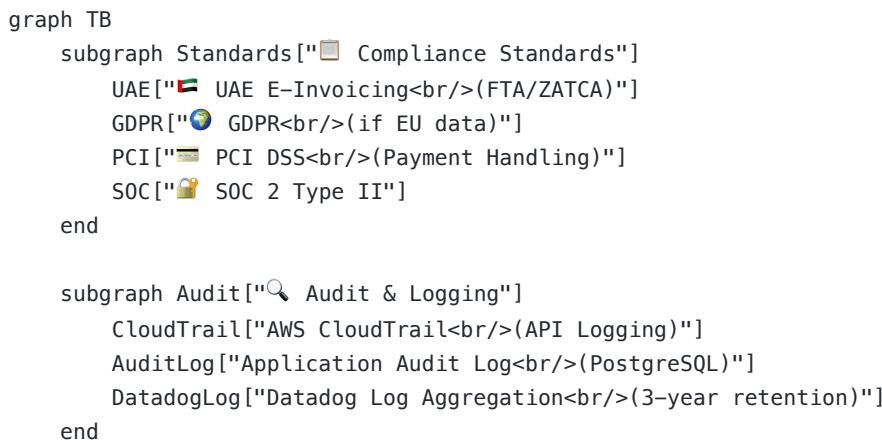
```
DDoS --> IGW
IGW --> PublicSubnet
PublicSubnet --> PrivateSubnet
PublicSubnet --> ALBSG
PrivateSubnet --> ECSSG
PrivateSubnet --> RDSSG

style CloudFlare fill:#f4c430
style AWSVPC fill:#ff9900,color:#000
style Security fill:#d32f2f,color:#fff
```

5.2 Data Encryption & Compliance



5.3 Compliance & Audit



```

subgraph Security [🛡️ Security Controls]
    RBAC ["Role-Based Access<br/>Control"]
    MFA ["Multi-Factor Auth<br/>(Okta/Auth0)"]
    WAF ["Web Application<br/>Firewall"]
end

Standards --> Audit
Audit --> Security

style Standards fill:#ffff3e0
style Audit fill:#f3e5f5
style Security fill:#e8f5e9

```

6. Monitoring & Observability Stack

6.1 Monitoring Architecture

```

graph TD
    subgraph Sources [📊 Data Sources]
        APP["Application Logs"]
        DB["Database Metrics"]
        INFRA["Infrastructure Metrics"]
        USER["User Analytics"]
    end

    subgraph Collection [📡 Collection]
        DatadogAgent["Datadog Agent<br/>(Container)"]
        APM["Datadog APM<br/>(Traces)"]
    end

    subgraph Platform [⌚ Datadog Platform]
        Dashboards["Dashboards"]
        Alerts["Alerting"]
        Analytics["Analytics"]
    end

    subgraph Actions [⚙️ Actions]
        Slack["Slack Notifications"]
        PagerDuty["PagerDuty<br/>(On-call)"]
        Logs["Log Retention<br/>(3 years)"]
    end

    Sources --> Collection
    Collection --> Platform
    Platform --> Actions

    style Platform fill:#1565c0,color:#fff

```

6.2 Key Metrics Tracked

```
graph LR
    subgraph Performance ["⚡ Performance"]
        APM["API Response Time"]
        DBLat["DB Query Latency"]
        QueueLag["Event Queue Lag"]
    end

    subgraph Availability ["🌐 Availability"]
        Uptime["Service Uptime %"]
        ErrorRate["Error Rate %"]
        HTTPStatus["HTTP Status Codes"]
    end

    subgraph Business ["💰 Business Metrics"]
        Orders["Orders Processed/day"]
        Revenue["Revenue Generated"]
        UserActiv["Active Users"]
    end

    subgraph Cost ["🔋 Cost Optimization"]
        CPUUse["CPU Utilization"]
        Memory["Memory Usage"]
        DataTrans["Data Transfer"]
    end

    subgraph AI ["🤖 Agent Metrics"]
        AgentExec["Agent Executions"]
        ToolCalls["Tool Calls/Agent"]
        DecisionTime["Decision Latency"]
    end

    style Performance fill:#fff3e0
    style Availability fill:#e8f5e9
    style Business fill:#f3e5f5
    style Cost fill:#e0f2f1
    style AI fill:#c8e6c9
```

7. Disaster Recovery & High Availability

7.1 DR Strategy

```
graph TB
    subgraph Primary ["🌐 Primary Region<br/>(eu-west-1)"]
        APPIR["Application"]
        DBIR["Database"]
    end
```

```

subgraph Secondary["🌐 Secondary Region<br/>(eu-central-1)<br/>Hot Standby"]
    APPSR["Application<br/>(Standby)"]
    DBSR["Database<br/>(Read Replica)"]
end

subgraph Failover["⚡ Failover Process"]
    Health["Health Check<br/>Every 30s"]
    Detect["Failure Detected<br/>(3 consecutive fails)"]
    Auto["Auto-Failover"]
    DNS["DNS Update<br/>(Route 53)"]
end

APPIR -.->|"Replication" | APPSR
DBIR -.->|"Streaming" | DBSR

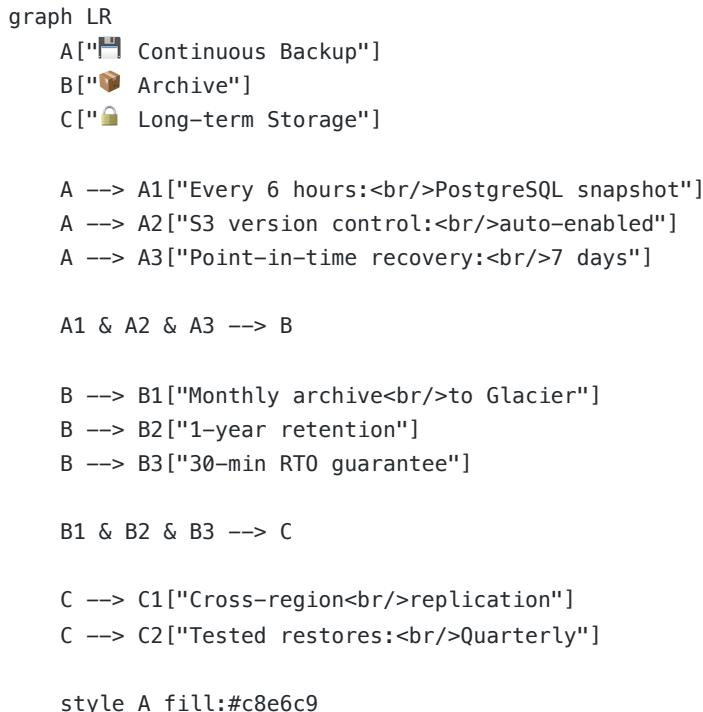
Health --> Detect
Detect --> Auto
Auto --> DNS

DNS -->|"2-3 min" | APPSR
DNS -->|"Complete Fail-over" | DBSR

style Primary fill:#a5d6a7
style Secondary fill:#ffff9c4
style Failover fill:#ffccbc

```

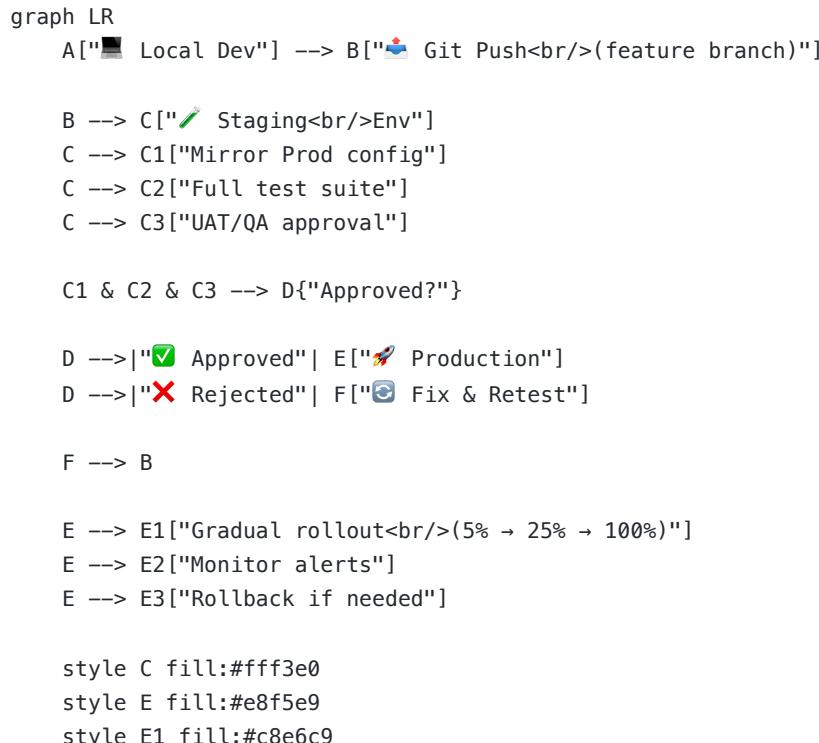
7.2 Backup Strategy



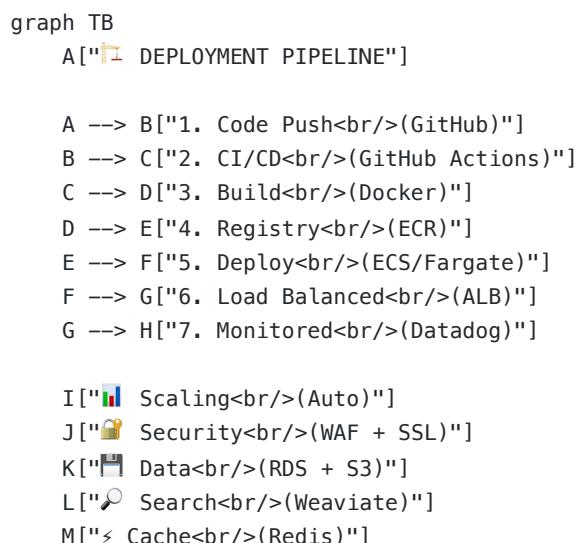
```
style B fill:#a5d6a7  
style C fill:#81c784
```

8. Development & Staging Environments

8.1 Environment Promotion



9. Complete Deployment Schematic



```
N["🔮 Agents<br/>(LangGraph)"]
```

```
H --> I  
H --> J  
H --> K  
H --> L  
H --> M  
H --> N
```

```
style A fill:#1565c0,color:#fff  
style B fill:#e3f2fd  
style C fill:#bbdefb  
style D fill:#90caf9  
style E fill:#64b5f6  
style F fill:#42a5f5  
style G fill:#2196f3  
style H fill:#1e88e5
```