Design a comprehensive cloud architecture for the “Agent Management Platform (AMP)”, a secure, scalable, multi-tenant AI agent orchestration solution. Deliver both a high-level diagram (e.g. in Mermaid or similar) and a detailed narrative. The architecture must follow a five-layer model and leverage a private GCP tenant:

1. **Pipeline Layer**

- Automated data/knowledge ingestion & transformation (e.g., Cloud Dataflow or Pub/Sub, Cloud Storage)

- Code/deployment pipelines (Terraform modules in Artifact Registry, Cloud Build triggered by Git branches: feature/*→dev, develop→test, main→preprod)

- Operations pipelines (Cloud Monitoring, Cloud Logging, automated rollbacks via Cloud Functions)

2. **Tools Layer**

- Internal tools hosted in private service catalog (Cloud Run/Cloud Functions)

- External third-party connectors (Azure OpenAI Service, Vertex AI, REST APIs)

- Secret management (Secret Manager, automatic rotation, replication)

3. **Models Layer**

- Containerized AI/ML models stored in Artifact Registry or Container Registry

- Model versioning & packaging workflows (Cloud Build, CI quality gates with tflint, Conftest)

- Vector DB for RAG (e.g., Vertex AI Matching Engine or managed vector store)

4. **Agents Layer**

- Microservices for agent runtimes on Cloud Run & Cloud Functions

- IAM roles & service accounts scoped per agent, least-privilege access

- Canary & blue-green deployments, horizontal autoscaling

5. **Workflows Layer**

- Orchestration via Workflows, Cloud Composer or Cloud Tasks
- YAML-driven definitions stored alongside code
- Error handling, retries, conditional branches

Cross-cutting requirements:

- Networking: VPC with subnets, private Google access, firewall rules, Cloud NAT
- Security & Compliance: Org Policy Service for region restrictions, Audit Logging, SOC2/GDPR controls
- CI/CD: GitOps with Cloud Build (terraform init/plan/apply), manual approvals for prod
- Environments: Parameterized tfvars for dev/test/preprod, service account per env
- Cost & Budget: Billing Budget API alerts, quota caps, automated cost reports
- Monitoring & Alerting: SLO/SLI dashboards in Cloud Monitoring, Pub/Sub-driven alerts
- Scalability: Auto-scaling Cloud Run, GKE or GKE Autopilot with HPA; multiregional storage with lifecycle rules

Produce:

1. A clear, layered diagram (Mermaid syntax or similar) showing components and data flows.
2. A narrative explaining each layer, key services, and how they interconnect to meet security, compliance, CI/CD, and autoscaling goals.

title Agent Management Platform (AMP) - Five Layer Cloud Architecture

// Cross-Cutting Requirements (vertical overlays)

"Cross-Cutting" [icon: shield] {

  Networking [icon: gcp-vpc, label: "VPC/Subnets/Firewall"]

```
  Security Compliance [icon: gcp-organization, label: "Org Policy/Audit Logging"]
  CI CD [icon: gcp-cloud-build, label: "CI/CD (Cloud Build, GitOps)"]
  Budget Alerts [icon: gcp-billing, label: "Budget/Quota Alerts"]
  Monitoring [icon: gcp-cloud-monitoring, label: "SLO/SLI Dashboards"]
}

// Pipeline Layer
Pipeline Layer [icon: pipeline, color: blue] {
  Data Pipeline [icon: gcp-dataflow, label: "Data Ingestion/Transformation"] {
    PubSub [icon: gcp-pubsub]
    Dataflow [icon: gcp-dataflow]
    Cloud Storage [icon: gcp-storage]
  }
  Code Pipeline [icon: gcp-cloud-build, label: "Code/Deployment Pipeline"] {
    Cloud Build [icon: gcp-cloud-build]
    Artifact Registry [icon: gcp-artifact-registry]
    Terraform Modules [icon: terraform]
    Git [icon: github]
  }
  Operations Pipeline [icon: gcp-cloud-functions, label: "Operations Pipeline"] {
    Cloud Monitoring [icon: gcp-cloud-monitoring]
    Cloud Logging [icon: gcp-logging]
    Cloud Functions [icon: gcp-cloud-functions, label: "Automated Rollbacks"]
  }
}
```

```
// Tools Layer
Tools Layer [icon: toolbox, color: purple] {
  Internal Tools [icon: gcp-cloud-run, label: "Internal Tools"] {
    Cloud Run Tools [icon: gcp-cloud-run]
    Cloud Functions Tools [icon: gcp-cloud-functions]
  }
  External Connectors [icon: plug, label: "External Connectors"] {
    Azure OpenAI [icon: azure-openai]
    Vertex AI [icon: gcp-vertex-ai]
    REST APIs [icon: api]
  }
  Secret Management [icon: gcp-secret-manager, label: "Secret Manager"] {
    Secret Manager [icon: gcp-secret-manager]
    Rotation [icon: refresh-ccw, label: "Auto Rotation"]
    Replication [icon: copy, label: "Replication"]
  }
}

// Models Layer
Models Layer [icon: brain, color: orange] {
  Model Registry [icon: gcp-artifact-registry, label: "Model Registry"] {
    Artifact Registry [icon: gcp-artifact-registry]
    Container Registry [icon: gcp-container-registry]
  }
  Model Versioning [icon: git-branch, label: "Versioning/Packaging"] {
    Cloud Build Models [icon: gcp-cloud-build]
```

```
    Quality Gates [icon: check-circle, label: "tflint/Conftest"]
  }
  Vector DB [icon: database, label: "Vector DB"] {
    Vertex Matching Engine [icon: gcp-vertex-ai]
    Managed Vector Store [icon: database]
  }
}

// Agents Layer
Agents Layer [icon: users, color: green] {
  Agent Runtimes [icon: gcp-cloud-run, label: "Agent Runtimes"] {
    Cloud Run Agents [icon: gcp-cloud-run]
    Cloud Functions Agents [icon: gcp-cloud-functions]
  }
  IAM Service Accounts [icon: user-check, label: "IAM/Service Accounts"]
  Deployment Strategies [icon: shuffle, label: "Deployment Strategies"] {
    Canary Deploy [icon: flag]
    Blue Green Deploy [icon: flag]
    Autoscaling [icon: trending-up]
  }
}

// Workflows Layer
Workflows Layer [icon: flowchart, color: yellow] {
  Orchestration Engine [icon: gcp-workflows, label: "Orchestration"] {
    Workflows [icon: gcp-workflows]
```

```
    Cloud Composer [icon: gcp-composer]
    Cloud Tasks [icon: gcp-tasks]
  }
  YAML Definitions [icon: file-text, label: "YAML Definitions"]
  Error Handling [icon: alert-triangle, label: "Error Handling/Retries"]
}

// Connections: Data and control flow
PubSub > Dataflow
Dataflow > Cloud Storage
Azure OpenAI > Artifact Registry
Vertex AI > Artifact Registry
REST APIs > Artifact Registry
Artifact Registry > Container Registry
Container Registry > Cloud Build Models
Cloud Build Models > Quality Gates
Quality Gates > Vertex Matching Engine
Vertex Matching Engine > Managed Vector Store
Workflows > YAML Definitions
Workflows > Error Handling

// Code pipeline feeds all layers
Cloud Build --> Artifact Registry: push models
Cloud Build --> Workflows: deploy

// Operations pipeline feeds monitoring and error handling
```

Cloud Monitoring --> Error Handling

Cloud Logging --> Error Handling

Cloud Functions --> Error Handling

// Secret management overlays

// Cross-cutting overlays (dashed lines)

Networking --> Workflows

Security Compliance --> Workflows

CI CD --> Cloud Build

CI CD --> Cloud Functions

CI CD --> Artifact Registry

CI CD --> Workflows

Budget Alerts --> Cloud Monitoring

Monitoring --> Cloud Monitoring

Monitoring --> Error Handling

Monitoring --> Workflows

Internal Tools > Artifact Registry

Internal Tools > Artifact Registry

Managed Vector Store > Agent Runtimes

Agent Runtimes > Workflows

Agent Runtimes > Workflows

Cloud Build --> Agent Runtimes: deploy

Secret Manager --> Internal Tools

Secret Manager --> Agent Runtimes

Secret Manager --> Model Registry

Networking --> Data Pipeline

Networking --> Internal Tools

Networking --> Model Registry

Networking --> Agent Runtimes

Security Compliance --> Data Pipeline

Security Compliance --> Internal Tools

Security Compliance --> Model Registry

Security Compliance --> Agent Runtimes

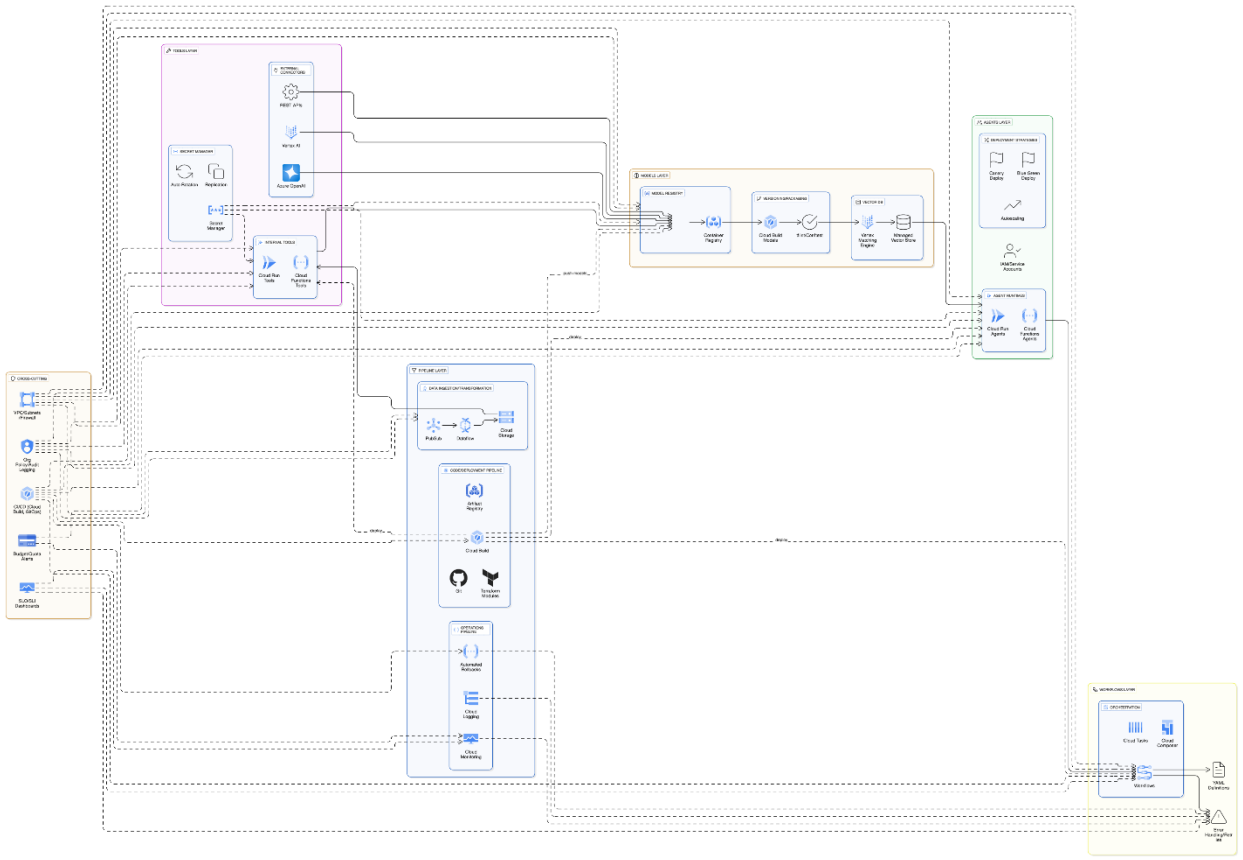
CI CD --> Internal Tools

CI CD --> Agent Runtimes

Budget Alerts --> Agent Runtimes

Internal Tools < Cloud Storage

Internal Tools <-- Cloud Build: deploy




Workflow Builder Layer
Non-Technical Users
Technical Users
Models Layer
AI Assistance
AI Agents
implementation Frameworks
Tools Layer
Internal Tools: Custom-made, GCP tenant-scoped service catalog entries.
External Tools: Third-party APIs/MCPs, managed connectors.
Data / Knowledge Base
Data Layer
Prompts
Pipeline Layer
Data / Knowledge Base Pipeline: Automated ingestion, transformation, and versioning of data.
Code / Deployment Pipeline: IaC templates (Terraform, Deployment Manager) stored in Git, triggered by commits.
Cloud Operations Pipeline: Monitoring, logging, alerting and automated rollbacks via Cloud Build and Cloud Functions.