Tutorial 26: Gemini Enterprise (formerly AgentSpace) - Enterprise Agent Platform

Difficulty: advanced
Reading Time: 2 hours
Tags: advanced, gemini-enterprise, enterprise, platform, governance
Description: Deploy and manage agents on Gemini Enterprise for enterprise-grade agent orchestration, collaboration, and governance.

:::info Product Rebranding

Note: Google AgentSpace was rebranded as **Gemini Enterprise** in late 2024. This tutorial uses the current product name and pricing (verified October 2025).

:::

:::info Verified Against Official Sources

This tutorial has been verified against official Google Cloud documentation.

Verification Date: October 12, 2025
Sources Checked:
- Official Gemini Enterprise website (cloud.google.com/gemini-enterprise)
- Pricing page (verified October 2025)
- Product documentation and FAQs

:::

Tutorial 26: Gemini Enterprise - Enterprise Agent Management

Goal: Deploy and manage AI agents at enterprise scale using Google Cloud's **Gemini Enterprise** platform (formerly AgentSpace)

Prerequisites:

- Tutorial 01 (Hello World Agent)
- Tutorial 02 (Function Tools)
- Tutorial 06 (Agents & Orchestration)
- Google Cloud account with billing enabled

What You'll Learn:

- Understanding Gemini Enterprise architecture
- Deploying ADK agents to Gemini Enterprise via Vertex AI Agent Builder
- Using pre-built Google agents (Idea Generation, Deep Research, NotebookLM)
- Building custom agents with Agent Designer (no-code builder)
- Managing agents at scale with governance and orchestration
- Integrating enterprise data sources (SharePoint, Drive, OneDrive, Salesforce)
- Gemini Enterprise pricing and licensing (Business $21, Enterprise Standard $30, Plus custom)
- Best practices for enterprise agent management

What is Gemini Enterprise?

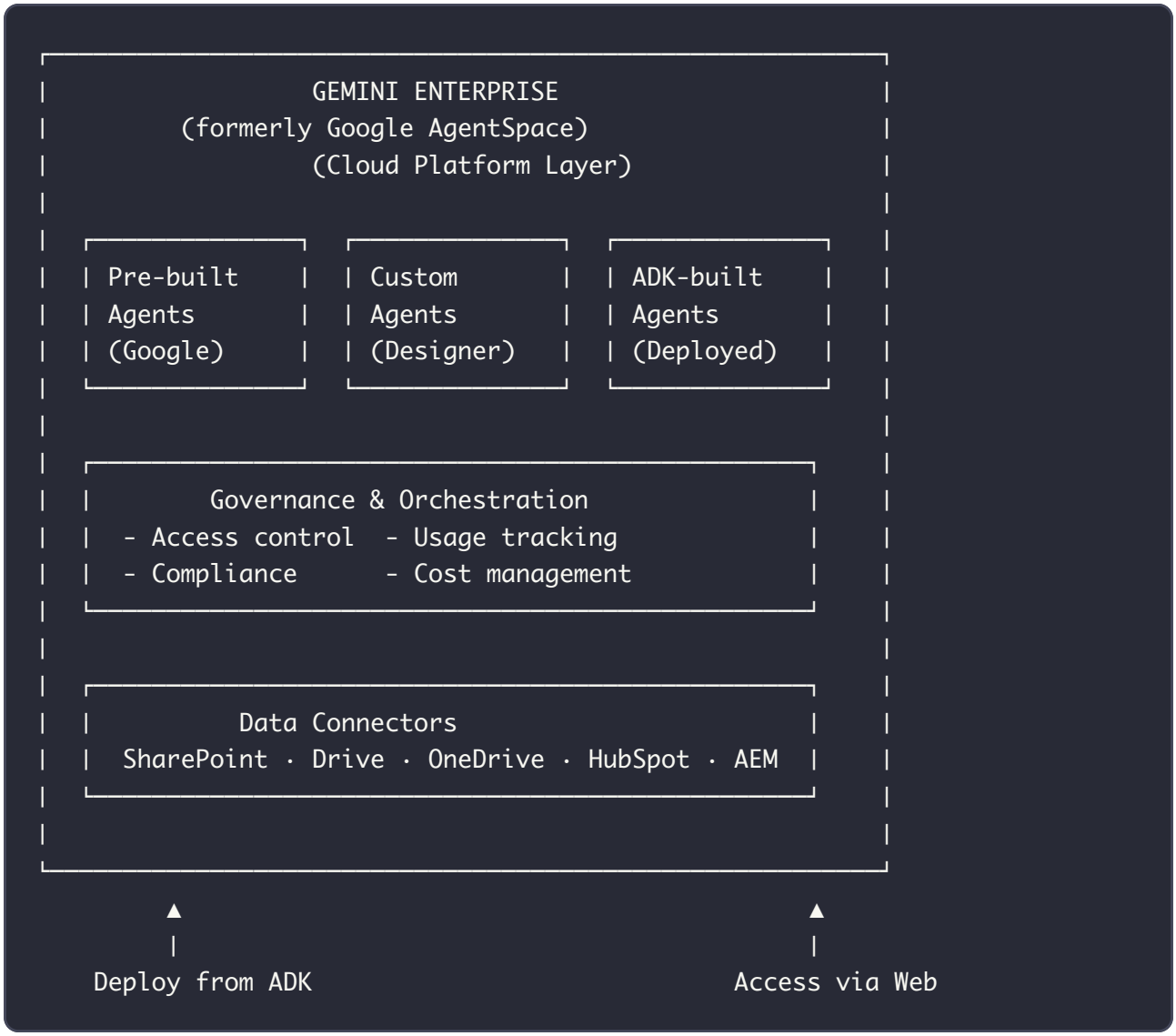
Gemini Enterprise (formerly Google AgentSpace) is Google Cloud's **enterprise platform for managing AI agents at scale**.

Official Site: cloud.google.com/gemini-enterprise (https://cloud.google.com/gemini-enterprise)

Historical Note: This product was originally launched as "Google AgentSpace" and was rebranded to "Gemini Enterprise" in late 2024 to align with Google's unified Gemini AI brand.

Relationship to ADK:

- **ADK (Agent Development Kit)**: Framework for *building* agents locally
- **Gemini Enterprise**: Platform for *deploying and managing* agents at scale
- Think: **ADK = Development** | **Gemini Enterprise = Operations**



Why Use Gemini Enterprise?

Need	Gemini Enterprise Solution
Deploy ADK agents to production	Managed hosting with auto-scaling
Manage multiple agents	Agent Gallery with discovery and sharing
Control agent access	Role-based access control (RBAC)
Monitor agent usage	Built-in observability and analytics
Connect to enterprise data	Pre-built connectors (SharePoint, Drive, etc.)
Ensure compliance	Governance policies and audit logs
Low-code agent creation	Agent Designer for non-developers
Quick start	Pre-built Google agents (Idea Generation, etc.)

1. Pre-built Google Agents

Gemini Enterprise includes **production-ready agents** built by Google:

Idea Generation Agent

What it does: Generates creative ideas based on prompts and context.

Use cases:

- Marketing campaigns
- Product brainstorming
- Content creation
- Strategic planning

Example:

```
User: "Generate 5 marketing campaign ideas for our new sustainable product lin

Agent:
1. "Green Future Challenge" - Social media campaign encouraging users to share
2. "Carbon Countdown" - Interactive calculator showing environmental impact of
3. "Eco-Warriors Program" - Loyalty program with sustainability incentives
4. "Nature's Return" - Augmented reality experience showing environmental rest
5. "Sustainable Stories" - Video series featuring customers' sustainability jo
```

Deep Research Agent

What it does: Conducts comprehensive research by searching, analyzing, and synthesizing information.

Use cases:

- Market research
- Competitive analysis
- Due diligence
- Literature reviews
- Technical investigations

Data sources:

- Google Search
- Connected enterprise documents
- Public datasets
- News articles
- Research papers

Example:

```
User: "Research emerging trends in electric vehicle battery technology"

Agent:
[REPORT] RESEARCH REPORT: EV Battery Technology Trends (2025)

Key Findings:
1. Solid-state batteries: 50% range improvement, commercial by 2026
2. Lithium-iron-phosphate (LFP): Cost reduction 30% since 2023
3. Silicon anodes: Energy density increase 20-40%
4. Dry electrode coating: Manufacturing cost down 15%
5. Battery-as-a-service models emerging

Market Leaders:
- QuantumScape (solid-state)
- CATL (LFP innovation)
- Panasonic (silicon anode)
- Tesla (4680 cells)

Sources: [15 citations from research papers, industry reports, news]
```

NotebookLM Enterprise

What it does: AI-powered notebook for research, note-taking, and knowledge synthesis.

Features:

- Document upload and analysis
- Interactive Q&A with documents
- Automated summarization
- Citation tracking
- Collaborative workspace

Use cases:

- Legal document review
- Research paper analysis
- Meeting notes and action items
- Knowledge base creation

Example workflow:

1. Upload 10 technical whitepapers
2. Ask: "What are the common security vulnerabilities across these papers?"
3. NotebookLM analyzes all documents
4. Provides synthesized answer with citations
5. Generate summary report

---

2. Why Use Gemini Enterprise?

Feature	Reason
No Infrastructure	Zero Kubernetes/scaling concerns - just deploy
Enterprise-Ready	Built-in auth, audit logs, SOC2/HIPAA compliance
Pre-Built Agents	Library of tested Google agents (e.g., Deep Research, NotebookLM, Idea Generation) ready to use immediately
Agent Designer	No-code builder for users to create agents with GUI
Data Connectors	One-click integration with Drive, Gmail, Salesforce, SharePoint, Adobe Experience Manager, ServiceNow, and SAP
Unified Governance	Centralized control over all agents (custom + Google's), permissions, secrets, data access
Pay-As-You-Go Agents	Inference costs only for what you use; no VM costs to serve models

---

3. Agent Gallery

Agent Gallery is Gemini Enterprise's **marketplace for discovering and sharing agents**.

Features

For Users:

- Browse available agents
- Search by category (Marketing, Engineering, Sales, HR)
- View agent ratings and reviews
- Try agents before deploying
- One-click installation

For Creators:

- Publish agents to company gallery
- Track usage metrics
- Receive feedback
- Update agents without breaking deployments
- Monetization (enterprise tier)

Example Categories

Marketing:

- Content Generator
- SEO Optimizer
- Campaign Planner
- Social Media Scheduler
- Brand Voice Analyzer

Sales:

- Lead Qualifier
- Proposal Writer
- Competitive Intel
- CRM Assistant
- Email Drafter

Engineering:

- Code Reviewer

- Documentation Generator
- Bug Analyzer
- Test Case Creator
- Architecture Advisor

HR:

- Resume Screener
- Interview Scheduler
- Onboarding Assistant
- Policy Explainer
- Performance Review Helper

Using Gallery Agents

```
# Conceptual example - actual API uses Vertex AI Agent Builder
from google.cloud import aiplatform
from google.cloud.aiplatform import AgentBuilderClient

# Initialize Vertex AI
aiplatform.init(project='your-project', location='us-central1')

# List available agents from gallery
client = AgentBuilderClient()
agents = client.list_agents(parent='projects/your-project/locations/us-central
for agent in agents:
    print(f"{agent.display_name}: {agent.description}")

# Deploy a custom ADK agent (use adk deploy command, or programmatically)
# adk deploy agent_engine --agent-path ./my_agent --project your-project

# Query deployed agent via Agent Builder API
agent_name = 'projects/your-project/locations/us-central1/agents/agent-abc123'
response = client.query_agent(
    agent=agent_name,
    query_input="Generate blog post outline about AI in healthcare"
)
print(response.response_text)
```

4. Deploying ADK Agents to Gemini Enterprise

Build locally with ADK → Deploy to Gemini Enterprise for production

Deployment Process

Step 1: Build Agent with ADK (local development)

```
# agent.py
from google.adk.agents import Agent
from google.adk.tools import FunctionTool

def analyze_sales_data(quarter: str, region: str) -> dict:
    """Analyze sales performance for specific quarter and region."""
    # Your business logic
    return {
        'revenue': 1250000,
        'growth': '+15%',
        'top_products': ['Product A', 'Product B']
    }

sales_agent = Agent(
    model='gemini-2.5-flash',
    name='sales_analyst',
    description='Analyzes sales data and provides insights',
    instruction="""
You are a sales data analyst.
Provide clear, actionable insights.
Highlight trends and opportunities.
    """.strip(),
    tools=[FunctionTool(analyze_sales_data)]
)
```

Step 2: Test Locally

```
from google.adk.agents import Runner

runner = Runner()
result = await runner.run_async(
    "What were our Q4 sales in the North region?",
    agent=sales_agent
)
print(result.content.parts[0].text)
```

Step 3: Package for Deployment

```
# Create deployment package
adk package \
  --agent agent.py:sales_agent \
  --requirements requirements.txt \
  --output sales-agent-v1.zip
```

Step 4: Deploy to Gemini Enterprise

```
# Deploy via ADK CLI (Vertex AI Agent Engine)
adk deploy agent_engine \
  --agent-path ./my_agent \
  --project your-project \
  --region us-central1 \
  --display-name "Sales Analyst Agent"

# Or package and deploy manually
gcloud ai agent-builder agents create \
  --project=your-project \
  --region=us-central1 \
  --display-name="Sales Analyst Agent" \
  --description="Q4 sales analysis"

# Output:
# Deployed: sales-analyst-prod (agent-abc123)
# URL: https://console.cloud.google.com/gen-app-builder/agents/agent-abc123
```

Step 5: Configure Production Settings

```yaml
# agentspace.yaml
name: sales-analyst-prod
version: 1.0.0
scaling:
  min_instances: 1
  max_instances: 10
  target_concurrency: 5
monitoring:
  alerts:
    - metric: error_rate
      threshold: 5%
      notification: ops-team@company.com
    - metric: latency_p95
      threshold: 2s
      notification: ops-team@company.com
governance:
  data_residency: us
  compliance: [SOC2, GDPR]
  audit_logging: true
connectors:
  - type: bigquery
    dataset: sales_data
    permissions: read
```

Step 6: Monitor in Gemini Enterprise Console

- Real-time usage metrics (Cloud Console → Gen App Builder → Agents)
- Error rates and logs (Cloud Logging integration)
- Cost tracking (BigQuery billing export)
- User feedback (built-in rating system)
- Performance trends (Cloud Monitoring dashboards)

---

5. Data Connectors

Gemini Enterprise provides **pre-built connectors** for enterprise data sources.

Available Connectors

Connector	Description	Use Cases
Google Drive	Access Drive files and folders	Document search, content analysis
SharePoint	Connect to SharePoint sites	Knowledge base, policy documents
OneDrive	Access OneDrive for Business	Personal files, team documents
HubSpot	CRM and marketing data	Lead management, customer insights
Salesforce	Sales and CRM data	Opportunity analysis, forecasting
Adobe AEM	Digital asset management	Content discovery, asset metadata
BigQuery	Data warehouse queries	Analytics, reporting, insights
Looker	Business intelligence	Dashboard data, metrics

Configuring Connectors

Example: SharePoint Connector

```yaml
# connector-config.yaml
connectors:
  - name: company-sharepoint
    type: sharepoint
    config:
      site_url: https://company.sharepoint.com/sites/knowledge-base
      authentication:
        type: oauth2
        client_id: ${SHAREPOINT_CLIENT_ID}
        client_secret: ${SHAREPOINT_CLIENT_SECRET}
      permissions:
        - read:documents
        - read:lists
    filters:
      - include: /Documents/**
      - exclude: /Documents/Archive/**
      - file_types: [.docx, .pdf, .xlsx]
    indexing:
      enabled: true
      schedule: daily
      incremental: true
```

Using Connectors in Agents

```python
from google.adk.agents import Agent
from google.adk.tools import url_context

policy_agent = Agent(
    model='gemini-2.5-flash',
    name='policy_assistant',
    instruction="""
You help employees understand company policies.
Always cite the specific policy document when answering.
If policy doesn't cover the question, say so clearly.
    """.strip(),
    tools=[
        url_context(
            name='company_policies',
            connector='company-sharepoint',
            path='/Documents/Policies/'
        )
    ]
)

# AgentSpace automatically handles:
# - Authentication to SharePoint
# - Document indexing
# - Search and retrieval
# - Permissions enforcement
```

6. Governance & Orchestration

Enterprise-grade controls for managing agents at scale.

Access Control

Role-Based Access Control (RBAC):

```yaml
# access-control.yaml
agents:
  - id: sales-analyst
    permissions:
      - role: sales-team
        access: [use, view_metrics]
      - role: sales-managers
        access: [use, view_metrics, edit_config]
      - role: admins
        access: [all]

  - id: hr-assistant
    permissions:
      - role: hr-team
        access: [use, view_metrics, edit_config]
      - role: employees
        access: [use]
      - role: contractors
        access: [] # No access
```

Data Access Controls:

```yaml
data_governance:
  pii_handling:
    mode: strict
    allowed_fields: [name, email, department]
    redacted_fields: [ssn, salary, medical_info]

  data_residency:
    primary: us-central1
    replicas: [europe-west1]
    prohibited_regions: [asia-pacific]

  retention:
    conversations: 90_days
    logs: 1_year
    audit_trail: 7_years
```

Usage Monitoring

Built-in Metrics:

- Queries per day/hour

- Average response time
- Error rates
- Token usage
- Cost per query
- User satisfaction scores

Custom Dashboards:

```
# Conceptual example
from google.cloud.agentspace import monitoring

# Create custom dashboard
dashboard = monitoring.Dashboard('Sales Agent Analytics')

dashboard.add_widget(
    monitoring.TimeSeriesChart(
        metric='agent_queries',
        agent_id='sales-analyst',
        aggregation='sum',
        group_by='user_department'
    )
)

dashboard.add_widget(
    monitoring.ScoreCard(
        metric='average_satisfaction',
        agent_id='sales-analyst',
        threshold_good=4.0,
        threshold_warning=3.0
    )
)
```

Cost Management

Budget Controls:

```
budgets:
  - agent: sales-analyst
    monthly_limit: $500
    alerts:
      - threshold: 80%
        action: notify_owner
      - threshold: 100%
        action: pause_agent

  - team: marketing-team
    monthly_limit: $2000
    alerts:
      - threshold: 90%
        action: notify_manager
```

Cost Optimization:

- Model selection (2.5-flash for routine, 2.5-pro for complex)
- Caching frequently accessed data
- Batching queries when possible
- Setting token limits per query
- Auto-scaling based on demand

---

7. Pricing & Plans

Gemini Enterprise Pricing (verified October 2025):

Gemini Business

Price: **$21 USD per seat per month**

Ideal for: Small businesses and teams within organizations

What's Included:

- Access to pre-built Google agents (Idea Generation, Deep Research, NotebookLM)
- Agent Designer (no-code agent builder)
- Agent Gallery access

- Gemini chat with higher quota
- Data connectors (Google Workspace, Microsoft 365)
- 25 GiB storage and data indexing per seat (pooled)
- Up to 300 seats
- Community support

Gemini Enterprise Standard

Price: **$30 USD per seat per month**

Ideal for: Large organizations needing enterprise-grade IT controls

Everything in Business, plus:

- Gemini Code Assist Standard (AI coding agent)
- Bring your own ADK-built agents or 3rd party agents
- Advanced security features (VPC-Service Controls, CMEK)
- Compliance support (SOC2, GDPR, HIPAA, FedRAMP High)
- Sovereign data boundaries for data residency
- Up to 75 GiB storage per seat (pooled)
- Unlimited seats
- Advanced governance and audit logs

Gemini Enterprise Plus

Price: **Contact sales for custom pricing**

Ideal for: Enterprises with complex requirements

Everything in Standard, plus:

- Premium support SLAs
- Custom data residency options
- Dedicated account team
- Custom integrations
- Volume discounts available

Usage-Based Costs (all editions, on top of license):

- **Model inference**: Same as Vertex AI pricing

- gemini-2.5-flash: ~$0.075/1M input tokens
- gemini-2.5-pro: ~$1.25/1M input tokens
- **Storage**: $0.023/GB/month (above included quota)
- **Data egress**: Standard Cloud pricing

Example Calculation:

Scenario: 50-person marketing team using Gemini Business

```
Base licenses:        50 seats × $21    = $1,050/month
                                        ————————————
Monthly fixed cost:                     $1,050

Estimated usage:
- 10,000 queries/month
- Avg 500 tokens/query (input + output)
- Using gemini-2.5-flash

Model cost: 10,000 × 500 × $0.075/1M  =    $0.38/month

Total monthly cost: ~$1,050
Per-seat cost: $1,050 / 50 = $21/seat/month (base license only)
```

Comparison to Previous Pricing: This replaces the legacy AgentSpace pricing announced earlier in 2024, which started at $25/seat. Current verified pricing (October 2025) starts at $21/seat for Business edition.

---

8. Real-World Example: Multi-Team Agent System

Scenario: Deploy agent ecosystem for entire company.

Architecture

```
Company Agent Ecosystem (AgentSpace)
├── Marketing Team (10 seats)
|   ├── Content Generator Agent
|   ├── SEO Optimizer Agent
|   └── Campaign Planner Agent
├── Sales Team (25 seats)
|   ├── Lead Qualifier Agent (ADK-built)
|   ├── Proposal Writer Agent
|   └── Competitive Intel Agent (ADK-built)
├── Engineering Team (40 seats)
|   ├── Code Reviewer Agent (ADK-built)
|   ├── Documentation Generator
|   └── Bug Analyzer Agent
└── HR Team (5 seats)
    ├── Resume Screener Agent
    ├── Interview Scheduler
    └── Onboarding Assistant (ADK-built)

Data Connectors:
- SharePoint (company policies, knowledge base)
- Google Drive (team documents)
- Salesforce (CRM data)
- BigQuery (analytics data)
- GitHub (code repositories)

Governance:
- Role-based access control
- Data residency: US only
- Compliance: SOC2, GDPR
- Audit logging: All agent interactions
```

Implementation

1. Deploy Custom ADK Agents

```
# lead_qualifier.py (built with ADK)
from google.adk.agents import Agent
from google.adk.tools import FunctionTool
from google.adk.models import GoogleGenAI

def check_company_size(company_name: str) -> dict:
    """Look up company size from database."""
    # Integration with company database
    return {'employees': 250, 'revenue': '50M'}

def score_lead(company_size: int, industry: str, budget: str) -> int:
    """Score lead from 0-100."""
    # Lead scoring logic
    score = 0
    if company_size > 100: score += 30
    if industry in ['technology', 'finance']: score += 30
    if budget == 'enterprise': score += 40
    return score

lead_qualifier = Agent(
    model='gemini-2.5-flash',
    name='lead_qualifier',
    description='Qualifies sales leads automatically',
    instruction="""
You qualify sales leads based on company profile.

Qualification criteria:
- Company size > 100 employees
- Industries: Technology, Finance, Healthcare
- Budget: Enterprise tier

Provide:
1. Lead score (0-100)
2. Key qualification factors
3. Recommended next steps
4. Potential objections
    """.strip(),
    tools=[
        FunctionTool(check_company_size),
        FunctionTool(score_lead)
    ]
)

# Deploy to AgentSpace
if __name__ == '__main__':
    from google.adk.deployment import deploy_to_agentspace
```

```
deploy_to_agentspace(
    agent=lead_qualifier,
    project='company-agentspace',
    region='us-central1',
    permissions=['sales-team@company.com'],
    connectors=['salesforce-crm']
)
```

2. Configure Data Connectors

```yaml
# agentspace-connectors.yaml
connectors:
  - name: salesforce-crm
    type: salesforce
    config:
      instance_url: https://company.my.salesforce.com
      authentication:
        type: oauth2
        client_id: ${SALESFORCE_CLIENT_ID}
        client_secret: ${SALESFORCE_CLIENT_SECRET}
      objects:
        - Lead
        - Opportunity
        - Account
        - Contact

  - name: company-sharepoint
    type: sharepoint
    config:
      site_url: https://company.sharepoint.com
      authentication:
        type: oauth2
      paths:
        - /Policies/**
        - /ProductDocs/**

  - name: engineering-github
    type: github
    config:
      organization: company-org
      authentication:
        type: personal_access_token
        token: ${GITHUB_TOKEN}
      repositories:
        - main-product
        - api-backend
        - mobile-app
```

3. Set Governance Policies

```yaml
# governance.yaml
global_policies:
  data_residency: us-central1
  compliance: [SOC2, GDPR, HIPAA]
  audit_logging: all_interactions
  pii_protection: enabled

team_permissions:
  - team: marketing-team@company.com
    agents: [content-generator, seo-optimizer, campaign-planner]
    data_access: [sharepoint:marketing/**, drive:marketing/**]

  - team: sales-team@company.com
    agents: [lead-qualifier, proposal-writer, competitive-intel]
    data_access: [salesforce:*, sharepoint:sales/**]

  - team: engineering-team@company.com
    agents: [code-reviewer, doc-generator, bug-analyzer]
    data_access: [github:*, bigquery:analytics_db]

  - team: hr-team@company.com
    agents: [resume-screener, interview-scheduler, onboarding-assistant]
    data_access: [sharepoint:hr/**, drive:hr/**]
    pii_access: [name, email, phone, resume]

budgets:
  - team: marketing-team
    monthly_limit: $500
  - team: sales-team
    monthly_limit: $1500
  - team: engineering-team
    monthly_limit: $2000
  - team: hr-team
    monthly_limit: $300
```

4. Monitor System-Wide

```python
# monitoring_dashboard.py
from google.cloud.agentspace import monitoring

# Create executive dashboard
exec_dashboard = monitoring.Dashboard('Company Agent Metrics')

# Add widgets
exec_dashboard.add_widget(
    monitoring.MetricCard('Total Queries Today', metric='total_queries')
)

exec_dashboard.add_widget(
    monitoring.BarChart(
        title='Queries by Team',
        metric='queries',
        group_by='team',
        time_range='today'
    )
)

exec_dashboard.add_widget(
    monitoring.LineChart(
        title='Cost Trend',
        metric='total_cost',
        time_range='30_days',
        group_by='team'
    )
)

exec_dashboard.add_widget(
    monitoring.Table(
        title='Most Used Agents',
        columns=['agent_name', 'queries', 'avg_satisfaction', 'cost'],
        sort_by='queries',
        limit=10
    )
)

# Publish dashboard
exec_dashboard.publish(viewers=['executives@company.com'])
```

9. Best Practices

Development Workflow

✔️ **DO**:

1. **Build locally with ADK** → Test thoroughly → Deploy to AgentSpace
2. **Version your agents** (v1.0, v1.1, etc.) for rollback capability
3. **Use staging environment** before production deployment
4. **Monitor metrics** after each deployment
5. **Collect user feedback** continuously
6. **Document agent capabilities** in Agent Gallery

❌ **DON'T**:

1. Deploy untested agents directly to production
2. Give all agents access to all data
3. Ignore cost monitoring
4. Skip governance configuration
5. Hard-code credentials
6. Deploy without rollback plan

Security

Agent Access:

- Use least-privilege principle for data connectors
- Regularly audit agent permissions
- Rotate API keys and credentials
- Enable MFA for AgentSpace access
- Monitor for unusual query patterns

Data Protection:

- Enable PII redaction for sensitive fields
- Configure data residency requirements
- Implement data retention policies

- Enable audit logging for compliance
- Regular security reviews

Cost Optimization

Model Selection:

```
# Use 2.5-flash for routine queries
routine_agent = Agent(model='gemini-2.5-flash')  # Cheaper

# Use 2.5-pro only for complex reasoning
complex_agent = Agent(model='gemini-2.5-pro')    # More expensive
```

Caching:

- Cache frequently accessed documents
- Use connector indexing for faster search
- Implement response caching for common queries

Query Optimization:

- Set max token limits
- Use concise instructions
- Batch similar queries when possible
- Disable streaming when not needed

Monitoring & Alerts

Key Metrics to Track:

1. **Usage**: Queries per day, peak hours
2. **Performance**: Average response time, error rate
3. **Cost**: Daily/monthly spend by team
4. **Quality**: User satisfaction, task completion rate
5. **Errors**: Failed queries, timeout rate

Alert Configuration:

```yaml
alerts:
  - name: High Error Rate
    condition: error_rate > 5%
    notification: ops-team@company.com
    severity: warning

  - name: Budget Exceeded
    condition: monthly_cost > budget_limit
    notification: finance-team@company.com
    severity: critical

  - name: Slow Response
    condition: p95_latency > 3s
    notification: eng-team@company.com
    severity: warning
```

Summary

You've learned how to deploy and manage agents at enterprise scale with Google AgentSpace:

Key Takeaways:

- ✔️ **AgentSpace** is Google Cloud's enterprise platform for agent operations
- ✔️ **ADK builds agents** locally → **AgentSpace deploys** to production
- ✔️ **Pre-built agents** available (Idea Generation, Deep Research, NotebookLM)
- ✔️ **Agent Designer** for low-code agent creation
- ✔️ **Agent Gallery** for discovering and sharing agents
- ✔️ **Data connectors** for SharePoint, Drive, Salesforce, etc.
- ✔️ **Governance & orchestration** for enterprise controls
- ✔️ **Pricing**: $25/seat/month + usage-based model costs
- ✔️ Deploy ADK agents with `adk package` and `gcloud agentspace deploy`
- ✔️ Monitor with built-in dashboards and custom metrics

When to Use Gemini Enterprise:

Use Case	Gemini Enterprise?
Prototyping new agent	❌ Use ADK locally
Production deployment	✔️ Deploy to Gemini Enterprise
Personal project	❌ Run locally or Cloud Run
Enterprise with 50+ users	✔️ Gemini Enterprise with governance
Need pre-built agents	✔️ Use Gallery agents (Deep Research, NotebookLM)
Custom agent with complex logic	[FLOW] Build with ADK → Deploy to Gemini Enterprise
Manage multiple teams	✔️ Gemini Enterprise with RBAC
Need enterprise data connectors	✔️ SharePoint, Drive, Salesforce connectors

Production Deployment Checklist:

- [ ] Agent tested thoroughly in local ADK environment
- [ ] Agent versioned (v1.0.0, v1.1.0, etc.)
- [ ] Data connectors configured with proper permissions
- [ ] Governance policies defined (RBAC, data residency)
- [ ] Budget limits set per team/agent
- [ ] Monitoring and alerts configured
- [ ] PII protection enabled for sensitive data
- [ ] Audit logging enabled for compliance
- [ ] Rollback plan documented
- [ ] User documentation published to Agent Gallery
- [ ] Staging environment tested before production
- [ ] Cost estimates reviewed and approved

Next Steps:

- **Tutorial 27**: Integrate third-party framework tools (LangChain, CrewAI)
- **Tutorial 28**: Use other LLMs with LiteLLM (OpenAI, Claude, Ollama)

- **Tutorial 19**: Implement Artifacts & File Management
- **Tutorial 18**: Master Events & Observability

Resources:

- Google AgentSpace (https://cloud.google.com/products/agentspace?hl=en)
- AgentSpace Documentation (https://cloud.google.com/agentspace/docs)
- Pricing Calculator (https://cloud.google.com/products/calculator)
- ADK Deployment Guide (https://google.github.io/adk-docs/deployment/)
- Data Connectors (https://cloud.google.com/agentspace/docs/connectors)

Congratulations! You now understand how to scale ADK agents to enterprise production with Gemini Enterprise. You can deploy custom agents, use pre-built agents (Deep Research, NotebookLM, Idea Generation), manage governance with RBAC and compliance features, and monitor operations at scale through the Cloud Console.

Generated on 2025-10-19 17:57:09 from 26_google_agentspace.md

Source: Google ADK Training Hub