Production & Deployment

> **Description:** Deploying ADK agents to production with observability, scaling, and service management

Purpose: Deploy ADK agents to production with proper observability, scaling, and service management.

Source of Truth:
google/adk-python/src/google/adk/cli/ (https://github.com/google/adk-python/tree/main/src/google/adk/cli/)
(ADK 1.15) + Official deployment guides

Table of Contents

Deployment Environments

Local Development

```
# Quick development testing
adk web agent_name

# Run with custom config
adk run agent_name --config config.yaml
```

Cloud Run (Serverless)

```
# Deploy to Cloud Run
adk deploy cloud_run agent_name

# Auto-scaling, pay-per-use
# Services: Cloud SQL, GCS, Vertex AI
```

Vertex AI Agent Engine (Managed)

```
# Enterprise deployment
adk deploy agent_engine agent_name

# Fully managed by Google
# High availability, monitoring
```

GKE (Kubernetes)

```
# Custom infrastructure
adk deploy gke agent_name

# Full control, custom scaling
```

Observability & Monitoring

Events (What Happened)

```
# Enable event logging
runner = Runner(
    event_service=LoggingEventService(level="DEBUG")
)

# Events captured:
# - AGENT_START/COMPLETE
# - TOOL_CALL_START/RESULT
# - LLM_REQUEST/RESPONSE
# - STATE_CHANGE
```

Tracing (Why It Happened)

```
# Detailed execution traces
runner = Runner(
    trace_service=CloudTraceService(project="my-project")
)

# View in Cloud Trace console
# Performance bottlenecks
# Error root causes
```

Callbacks (Custom Monitoring)

```
def monitor_agent(context, result):
    # Custom metrics
    log_performance(result.execution_time)
    alert_on_errors(result.errors)

agent = Agent(
    name="monitored_agent",
    callbacks=[monitor_agent]
)
```

Evaluation (Quality Metrics)

```
# Automated testing
adk eval agent_name --test-set my_tests.evalset.json

# Metrics:
# - tool_trajectory_avg_score (0-1)
# - response_match_score (0-1)
# - Custom LLM-as-judge metrics
```

💾 Service Configuration

Development (InMemory)

```
runner = Runner()  # All services default to InMemory
```

Production (Persistent)

```
runner = Runner(
    session_service=PostgresSessionService(uri="..."),
    artifact_service=GcsArtifactService(bucket="..."),
    memory_service=VertexAiMemoryBankService(project="...")
)
```

🔒 Security & Best Practices

- **Environment Variables**: Never commit secrets
- **Service Accounts**: Least privilege access
- **Input Validation**: Sanitize all inputs
- **Rate Limiting**: Protect against abuse

- **Error Handling**: Graceful failure modes

📊 Performance Optimization

- **Model Selection**: Right model for cost/performance
- **Caching**: Reuse expensive computations
- **Parallel Execution**: Independent tasks simultaneously
- **Batch Processing**: Group similar requests

🎯 Key Takeaways

1. **Multiple deployment options**: Local, Cloud Run, Vertex AI, GKE
2. **Observability layers**: Events, traces, callbacks, evaluation
3. **Service configuration**: InMemory for dev, persistent for prod
4. **Security first**: Environment variables, validation, rate limiting
5. **Performance**: Optimize models, caching, parallel execution

🔗 **Next**: Explore Advanced Patterns (advanced-patterns.md) for cutting-edge capabilities.

Generated on 2025-10-21 09:03:26 from production-deployment.md

Source: Google ADK Training Hub