Tutorial 00: Setup & Authentication - Getting Started with Google ADK

Difficulty: beginner
Reading Time: 30 minutes
Tags: beginner, setup, authentication, vertexai, gemini-api, gcp, api-keys
Description: Essential setup guide for Google ADK - learn how to obtain API keys, create GCP projects, configure authentication, and choose between VertexAI and Gemini API platforms.

:::info Verified Against Official Sources

This tutorial has been verified against official Google documentation and ADK source code.

Verification Date: October 15, 2025
ADK Version: 1.16.0+
Sources Checked:

- VertexAI Documentation (https://cloud.google.com/vertex-ai/generative-ai/docs)
- Gemini API Documentation (https://ai.google.dev/gemini-api/docs)
- ADK Python source code integration patterns

:::

Tutorial 00: Setup & Authentication - Getting Started with Google ADK

Goal: Set up authentication and choose the right Google AI platform for ADK development.

Prerequisites: None - This is the foundation for all other tutorials

Time Estimate: 30 minutes

Overview

Before building your first ADK agent, you need to set up authentication and choose your Google AI platform. Google provides two primary platforms for accessing Gemini models: **VertexAI** (part of Google Cloud Platform) and **Gemini API** (standalone Google AI service).

This foundational tutorial covers:

- Getting API keys and setting up authentication
- Understanding platform differences and choosing the right one
- Basic ADK setup and configuration
- Environment preparation for all subsequent tutorials

Important: Complete this tutorial first - all other tutorials depend on having proper authentication configured.

Platform Comparison

Quick Decision Guide

Use Case	Platform	Why
Learning ADK	Gemini API	Free, simple setup
Prototyping	Gemini API	1500 requests/day free
Production	VertexAI	Enterprise features, security
High Traffic	VertexAI	Provisioned throughput

Key Differences

Gemini API (Beginners):

- ✔️ API key authentication
- ✔️ 1500 requests/day free
- ✔️ No GCP account needed
- ❌ Basic features only

VertexAI (Production):

- ✔️ Enterprise security
- ✔️ GCP integration
- ✔️ Advanced monitoring
- ❌ Complex setup

Pricing: Identical - $0.30/1M input tokens, $2.50/1M output tokens.

Authentication Setup

Gemini API (Simple)

```
# 1. Get API key from https://aistudio.google.com/apikey
# 2. Set environment variable

# 3. Test connection
python -c "
from google.genai import Client
Client().models.generate_content(model='gemini-2.5-flash', contents='test')
"
```

VertexAI (Enterprise)

```
# 1. Set project

# 2. Authenticate
gcloud auth application-default login

# 3. Enable API
gcloud services enable aiplatform.googleapis.com

# 4. Test connection
python -c "
from google.genai import Client
Client(vertexai=True).models.generate_content(model='gemini-2.5-flash', conten
"
```

Cost Management

Free Tiers

- **Gemini API**: 1500 requests/day, 1M tokens/minute
- **VertexAI**: $300-500 initial credits (90 days)

Paid Usage

- **Input tokens**: $0.30 per 1M tokens
- **Output tokens**: $2.50 per 1M tokens
- **Same pricing** on both platforms

Cost Control

```
# Set budget alerts
gcloud billing budgets create adk-budget \
    --billing-account=YOUR_BILLING_ACCOUNT \
    --display-name="ADK Budget" \
    --budget-amount=50.00 \
    --threshold-rule=percent=50,percent=90
```

Setup Workflow

```
ADK Setup Flow - Choose Your Path
==================================

Path A: Gemini API (Recommended for beginners)
├── 1. Visit https://aistudio.google.com/apikey
├── 2. Create API key (free, instant)
├── 3. Set environment: export GEMINI_API_KEY=your-key
├── 4. Install ADK: pip install google-genai
├── 5. Create agent and run: adk web my_agent
└── ✔ Ready in 5 minutes!

Path B: VertexAI (For enterprise/production)
├── 1. Create GCP project at console.cloud.google.com
├── 2. Enable VertexAI API in project
├── 3. Install gcloud CLI
├── 4. Authenticate: gcloud auth application-default login
├── 5. Set project: gcloud config set project your-project
├── 6. Install ADK: pip install google-genai
├── 7. Create agent with vertexai=True
└── ✔ Enterprise-ready (15-30 minutes)

Common Issues & Solutions:
├── "API key invalid" → Check key in Google AI Studio
├── "ADC not found" → Run gcloud auth application-default login
├── "Quota exceeded" → Wait 1 minute or upgrade plan
└── "Permission denied" → Enable APIs in GCP console
```

Platform-Specific Features

VertexAI Exclusive:

```
# Provisioned throughput for guaranteed performance
# Advanced MLOps features
# VPC Service Controls for security
# Model monitoring and explainability
# Integration with BigQuery, Cloud Storage, etc.
```

Gemini API Exclusive:

```
# Google AI Studio interface
# Simple API key authentication
# Built-in playground for testing
# Ephemeral tokens for client-side apps
```

Integration Patterns

ADK Agent Implementation

VertexAI Pattern:

```
from adk import Agent
from google.genai import Client

# VertexAI agent (enterprise-ready)
vertex_agent = Agent(
    name="enterprise_agent",
    model="gemini-2.5-flash",
    instruction="You are an enterprise AI assistant",
    tools=[tool1, tool2],
    # Uses ADC automatically - no API key needed
)

# Deploy to VertexAI endpoints
# Integrated monitoring and logging
# VPC security controls
```

Gemini API Pattern:

```
from adk import Agent
from google.genai import Client

# Gemini API agent (developer-friendly)
gemini_agent = Agent(
    name="dev_agent",
    model="gemini-2.5-flash",
    instruction="You are a development AI assistant",
    tools=[tool1, tool2],
    # Uses GEMINI_API_KEY environment variable
)

# Quick deployment
# Simple authentication
# Cost-effective for development
```

Deployment Scenarios

Development Environment:

```
# Gemini API - Quick setup

adk web dev_agent  # Start development server
```

Production Environment:

```
# VertexAI - Enterprise deployment

gcloud auth application-default login
adk deploy vertexai prod_agent  # Deploy to VertexAI
```

Minimum Requirements for ADK

API Enablement Requirements

Gemini API (No GCP Required)

Minimum Requirements:

- ✔️ Google AI Studio account
- ✔️ API key from https://aistudio.google.com/apikey (https://aistudio.google.com/apikey)
- ❌ No GCP project required
- ❌ No APIs to enable

Verified Setup:

```
# Only requirement: API key

# Test ADK functionality
python -c "
from google.genai import Client
client = Client()
response = client.models.generate_content(
    model='gemini-2.5-flash',
    contents='Hello ADK'
)
print('✔️ Gemini API ready for ADK')
"
```

VertexAI (GCP Required)

Minimum APIs to Enable:

- ✔️ `aiplatform.googleapis.com` (VertexAI API)
- ✔️ `iam.googleapis.com` (Identity and Access Management)

Optional APIs for Advanced Features:

- `bigquery.googleapis.com` (BigQuery integration)
- `storage.googleapis.com` (Cloud Storage integration)
- `secretmanager.googleapis.com` (Secret Manager for keys)

Verified API Enablement:

```
# Enable minimum required APIs
gcloud services enable aiplatform.googleapis.com
gcloud services enable iam.googleapis.com

# Verify APIs are enabled
gcloud services list --enabled | grep -E "(aiplatform|iam)"

# Expected output:
# aiplatform.googleapis.com    Vertex AI API
# iam.googleapis.com           Identity and Access Management (IAM) API
```

User Rights and Permissions

Gemini API User Rights

Minimum Permissions:

- ✔️ Google account with access to Google AI Studio
- ✔️ Ability to create API keys
- ❌ No GCP IAM roles required

VertexAI User Rights

Minimum IAM Roles:

- ✔️ `roles/aiplatform.user` - Basic VertexAI access
- ✔️ `roles/iam.serviceAccountUser` - Service account usage (optional)

Verified Permission Setup:

```
# Check current user permissions
gcloud auth list

# Grant minimum required role (run as project admin)
gcloud projects add-iam-policy-binding your-project-id \
    --member="user:your-email@gmail.com" \
    --role="roles/aiplatform.user"

# Verify permissions
gcloud projects get-iam-policy your-project-id \
    --filter="bindings.members:user:your-email@gmail.com" \
    --format="table(bindings.role)"
```

Complete Minimal ADK Setup Verification

Gemini API Verification

```
#!/bin/bash
# Minimal ADK setup verification for Gemini API

# 1. Check API key exists
if [ -z "$GEMINI_API_KEY" ]; then
    echo "❌ GEMINI_API_KEY not set"
    exit 1
fi

# 2. Test API connectivity
python3 -c "
import os
from google.genai import Client

try:
    client = Client()
    response = client.models.generate_content(
        model='gemini-2.5-flash',
        contents='ADK setup test'
    )
    print('✔️ Gemini API ready for ADK')
    print(f'Response: {response.text[:50]}...')
except Exception as e:
    print(f'❌ Gemini API test failed: {e}')
    exit(1)
"

echo "🎉 ADK with Gemini API is fully operational!"
```

VertexAI Verification

```
#!/bin/bash
# Minimal ADK setup verification for VertexAI

PROJECT_ID=${GOOGLE_CLOUD_PROJECT:-"your-project-id"}

# 1. Check project exists
if ! gcloud projects describe $PROJECT_ID >/dev/null 2>&1; then
    echo "❌ Project $PROJECT_ID not found"
    exit 1
fi

# 2. Check required APIs
REQUIRED_APIS=("aiplatform.googleapis.com" "iam.googleapis.com")
for api in "${REQUIRED_APIS[@]}"; do
    if ! gcloud services list --enabled | grep -q $api; then
        echo "❌ API $api not enabled"
        exit 1
    fi
done

# 3. Check user permissions
USER_EMAIL=$(gcloud auth list --filter=status:ACTIVE --format="value(account)"
if ! gcloud projects get-iam-policy $PROJECT_ID \
    --filter="bindings.members:user:$USER_EMAIL" \
    --format="table(bindings.role)" | grep -q "aiplatform.user"; then
    echo "❌ User lacks aiplatform.user role"
    exit 1
fi

# 4. Test VertexAI connectivity
python3 -c "
import os
from google.genai import Client

try:
    client = Client(vertexai=True)
    response = client.models.generate_content(
        model='gemini-2.5-flash',
        contents='ADK setup test'
    )
    print('✔️ VertexAI ready for ADK')
    print(f'Response: {response.text[:50]}...')
except Exception as e:
    print(f'❌ VertexAI test failed: {e}')
    exit(1)
"
```

```
echo "🎉 ADK with VertexAI is fully operational!"
```

Service Account Setup (Optional but Recommended)

For production deployments, use service accounts instead of user accounts:

```
# Create service account
gcloud iam service-accounts create adk-service \
    --description="ADK service account" \
    --display-name="ADK Service"

# Grant minimal permissions
gcloud projects add-iam-policy-binding $PROJECT_ID \
    --member="serviceAccount:adk-service@$PROJECT_ID.iam.gserviceaccount.com" \
    --role="roles/aiplatform.user"

# Create key for ADK usage
gcloud iam service-accounts keys create adk-key.json \
    --iam-account=adk-service@$PROJECT_ID.iam.gserviceaccount.com

# Set environment for ADK
```

ADK-Specific Requirements

Python Dependencies:

```
# requirements.txt for minimal ADK setup
google-genai>=1.16.0
# ADK framework (when available)
# adk>=1.0.0
```

Python Version:

- Minimum: Python 3.8
- Recommended: Python 3.10+
- Verified: Python 3.11

Network Requirements:

- ✔️ HTTPS access to `*.googleapis.com`
- ✔️ DNS resolution working
- ❌ No proxy requirements (direct internet access)

Troubleshooting Minimum Setup

"API has not been used" error:

```
# Enable the API explicitly
gcloud services enable aiplatform.googleapis.com

# Wait 2-3 minutes for propagation
sleep 180

# Retry your ADK setup
```

"Permission denied" despite correct role:

```
# Check if organization policies block access
gcloud resource-manager org-policies list \
    --project=$PROJECT_ID

# Common issue: VertexAI disabled at org level
# Contact your GCP administrator
```

Service account key issues:

```
# Verify key format
cat adk-key.json | jq '.type'  # Should show "service_account"

# Check key expiration
cat adk-key.json | jq '.private_key_id'

# Regenerate if expired
gcloud iam service-accounts keys create new-adk-key.json \
    --iam-account=adk-service@$PROJECT_ID.iam.gserviceaccount.com
```

Best Practices

Security Essentials

API Keys:

- Never commit keys to code
- Use environment variables
- Rotate keys every 90 days

VertexAI:

- Use service accounts, not user accounts
- Grant minimal IAM permissions
- Enable VPC Service Controls for production

Environment Separation

```
# Development

# Production

# Use ADC with production service account
```

Troubleshooting Common Issues

Authentication Problems

"gcloud command not found"

```
# Install Google Cloud CLI
curl https://sdk.cloud.google.com | bash
exec -l $SHELL

# Verify installation
gcloud version
```

"ADC not found" error

```
# Run authentication
gcloud auth application-default login

# Set project
gcloud config set project your-project-id

# Verify
gcloud auth list
```

"API key invalid" error

```
# Check key format (should start with "AIza")
echo $GEMINI_API_KEY | head -c 10  # Should show "AIza..."

# Regenerate key at https://aistudio.google.com/apikey
# Update environment variable
```

Permission Issues

"Permission denied" in VertexAI

```
# Enable VertexAI API
gcloud services enable aiplatform.googleapis.com

# Grant necessary IAM roles
gcloud projects add-iam-policy-binding your-project \
    --member="user:your-email@gmail.com" \
    --role="roles/aiplatform.user"
```

"Quota exceeded" errors

```
# Check current usage in Google AI Studio
# Free tier: 15 RPM, 1500 RPD

# Wait and retry
sleep 60  # Wait 1 minute

# Or upgrade to paid tier in Google AI Studio
```

Network/Connectivity Issues

"Connection timeout" errors

```
# Check network connectivity
ping googleapis.com

# Verify API endpoints are accessible
curl -I https://generativelanguage.googleapis.com
```

DNS resolution issues

```
# Flush DNS cache (macOS)
sudo dscacheutil -flushcache
sudo killall -HUP mDNSResponder
```

Model-Specific Issues

"Model not found" errors

```
# Use correct model names
VALID_MODELS=(
    "gemini-2.5-pro"
    "gemini-2.5-flash"
    "gemini-2.5-flash-lite"
    "gemini-2.0-flash"
)

# Check model availability in your region
gcloud ai models list --region=us-central1
```

Slow response times

```
# Use faster models for development
FAST_MODELS=(
    "gemini-2.5-flash-lite"    # Fastest
    "gemini-2.5-flash"         # Balanced
)

# For production, use provisioned throughput
gcloud ai endpoints create provisioned-endpoint \
    --project=your-project \
    --region=us-central1 \
    --model=gemini-2.5-flash \
    --traffic-split=100
```

Environment Issues

Python import errors

```
# Install/update google-genai
pip install --upgrade google-genai

# Check Python version (3.8+ required)
python --version

# Verify package installation
python -c "import google.genai; print('OK')"
```

Environment variable not set

```
# Check if variable is set
echo $GEMINI_API_KEY  # Should show your key
echo $GOOGLE_CLOUD_PROJECT  # Should show project ID

# Set in current session

# Make permanent (add to ~/.bashrc or ~/.zshrc)
echo 'export GEMINI_API_KEY=your-key' >> ~/.zshrc
source ~/.zshrc
```

Frequently Asked Questions (FAQ)

Authentication & Setup

Q: Which platform should I choose for learning ADK?
A: Start with **Gemini API** - it has a generous free tier (1500 requests/day), simple API key setup, and is perfect for learning without GCP complexity.

Q: I'm getting "ADC not found" error. What do I do?
A: Run `gcloud auth application-default login` and ensure you've set your project with `gcloud config set project your-project-id`.

Q: My API key isn't working. What's wrong?
A: Check that your API key is correctly copied from Google AI Studio and set as `GEMINI_API_KEY` environment variable. Keys starting with "AIza" are correct.

Q: Can I use both platforms in the same project?
A: Yes! You can develop with Gemini API and deploy to production using VertexAI. Just configure different authentication methods.

Cost & Billing

Q: How do I avoid unexpected charges?

A:

- Use Gemini API free tier for development (1500 requests/day limit)
- Set up billing alerts in GCP console
- Monitor usage in Google AI Studio dashboard
- Use cost-effective models like `gemini-2.5-flash-lite` for simple tasks

Q: What's the actual cost difference between platforms?
A: For the same Gemini models, pricing is identical. VertexAI costs more due to GCP infrastructure, but offers enterprise features and potential discounts.

Q: How do I set up cost alerts?

```
# Create budget alert in GCP
gcloud billing budgets create my-adk-budget \
    --billing-account=YOUR_BILLING_ACCOUNT \
    --display-name="ADK Development" \
    --budget-amount=50.00 \
    --threshold-rule=percent=50 \
    --threshold-rule=percent=90
```

Security & Best Practices

Q: How do I secure my API keys?
A: Never commit keys to code. Use environment variables or GCP Secret Manager. Rotate keys regularly and restrict API key usage in Google AI Studio.

Q: Should I use VertexAI for production?
A: Yes, for enterprise applications. It provides VPC Service Controls, audit logging, and compliance certifications (SOC 2, HIPAA).

Q: How do I handle rate limits?
A: Implement exponential backoff retry logic. For VertexAI, consider provisioned throughput for guaranteed performance.

Troubleshooting

Q: "Quota exceeded" errors?
A: Free tier limits: Gemini API (15 RPM, 1500 RPD). Wait 1 minute or upgrade to paid tier.

Q: Model not found errors?
A: Ensure you're using correct model names: `gemini-2.5-flash`, `gemini-2.5-pro`, etc. Check platform availability.

Q: Permission denied in VertexAI?
A: Enable VertexAI API in GCP console and ensure your account has necessary IAM roles (VertexAI User).

Q: Slow response times?
A: Use `gemini-2.5-flash-lite` for speed, or VertexAI provisioned throughput for consistent performance.

Migration & Advanced

Q: How do I migrate from Gemini API to VertexAI?
A: Set up GCP project, enable APIs, run `gcloud auth application-default login`, update your ADK code to use `vertexai=True`.

Q: Can I use ADK with other Google services?
A: Yes! VertexAI integrates with BigQuery, Cloud Storage, Cloud Functions, and more for comprehensive AI solutions.

Q: What's the difference between model versions?
A: Use stable versions for production (`gemini-2.5-flash`). Preview versions (`gemini-2.5-flash-preview-09-2025`) may change.

Quick Start Commands

Gemini API (Recommended for beginners)

```
# 1. Get API key from https://aistudio.google.com/apikey
# 2. Set environment variable

# 3. Test setup
python -c "from google.genai import Client; print('Setup successful!')"
```

VertexAI (For production)

```
# 1. Set up GCP project

# 2. Authenticate
gcloud auth application-default login
gcloud config set project $GOOGLE_CLOUD_PROJECT

# 3. Enable VertexAI API
gcloud services enable aiplatform.googleapis.com

# 4. Test setup
python -c "from google.genai import Client; print('Setup successful!')"
```

Resources

- VertexAI Documentation (https://cloud.google.com/vertex-ai/generative-ai/docs)
- Gemini API Documentation (https://ai.google.dev/gemini-api/docs)
- ADK Platform Integration Guide (https://github.com/google/adk-python)
- Google AI Studio (https://aistudio.google.com)
- Google Cloud Console (https://console.cloud.google.com)

Generated on 2025-10-21 09:01:47 from 00_setup_authentication.md

Source: Google ADK Training Hub