

Agentic Automation in Google Workspace: Bridging Antigravity and Gemini 3.0

Abstract

This article explores automating Google Workspace by integrating Google Antigravity and Gemini 3.0 with Model Context Protocol (MCP) servers. We demonstrate how to overcome tool limits and utilize custom extensions to enable AI agents to securely execute scripts, manage files, and perform RAG-based tasks using private data.

Introduction

Google Antigravity and Gemini 3.0 are ushering in a new era of “Agent-First” development, transforming how we interact with cloud

environments. [Ref](#) A key component of this evolution is the integration of Model Context Protocol (MCP) servers. When connected to Antigravity, these servers empower the architecture to resolve complex, multi-step tasks by granting the AI direct, standardized access to external tools and proprietary data.

This article introduces a practical approach to **Google Workspace Automation** using Antigravity and Gemini 3.0. To achieve this, we utilize two specific MCP server extensions:

- [ToolsForMCPServer-extension](#): Utilizes Google Apps Script to enable the agent to securely read and write data across Google Workspace.
- [FileSearchStore-extension](#): Provides a managed Retrieval-Augmented Generation (RAG) system to ground automation in your private local documents.

Prerequisites

Before proceeding, ensure your environment meets the following requirements:

Gemini CLI: Must be installed and authenticated.

Extensions: The following Gemini CLI Extensions must be installed and verified as working:

- [ToolsForMCPServer-extension](#)
- [FileSearchStore-extension](#)

This guide assumes that the Gemini CLI allows the use of both extensions simultaneously.

Environment Setup

To implement this workflow, we must establish a communication bridge between the Google Antigravity and the MCP servers.

1. Install Google Antigravity

First, set up the IDE environment. Please refer to the official release and installation guide at <https://antigravity.google/>.

2. Automatic Installation of MCP Servers

Since `ToolsForMCPServer-extension` and `FileSearchStore-extension` function as MCP servers, they can be automatically detected and configured within Antigravity using a natural language prompt.

Launch Google Antigravity and input the following command:

```
Search the MCP servers from the Gemini CLI Extensions in my drive and install them.
```

This prompt triggers an update to the `mcp_config.json` file, registering the extensions as follows:

```
{
```

```
"mcpServers": {  
  
    "tools-for-mcp-server-extension": {  
  
        "command": "node",  
  
        "args": [  
  
            "/home/tanaike/.gemini/extensions/tools-for-mcp-server-extension/mcp-server/src/tools-for-mcp-server-extension.js"  
  
        ]  
  
    },  
  
    "file-search-store-extension": {  
  
        "command": "node",  
  
        "args": [  
  
            "/home/tanaike/.gemini/extensions/file-search-store-extension/mcp-server/src/file-search-store-extension.js"  
  
        ]  
  
    }  
}
```

```
    ]
```

```
    }
```

```
}
```

3. Configuration and Tool Limitations

After installation, refreshing the MCP servers panel should display the connected tools.

Important Note on Tool Limits:

During initial setup, you may encounter an error similar to: Error:

```
adding this instance with 160 enabled tools would exceed max limit of  
100. Also, the buttons for enabling or disabling are locked.
```

Press enter or click to view image in full size

Manage MCP servers 11 / 100 tools View raw config Refresh

file-search-store-extension 11 / 11

tools-for-mcp-server-extension [Configure](#)

Error: adding this instance with 160 enabled tools would exceed max limit of 100.

1. `get_exchange_rate`
Use this to get the current exchange rate. Using this, for example, it can exchange yen for dollars.

2. `get_current_weather`
Use this to get the current weather using the latitude and the longitude. At that time, convert the location to the latitude and the longitude and provide them to the function. If you cannot know the location, decide the location using the timezone.

3. `get_specific_date_weather`
Use this to get the weather for the specific date using the latitude and the longitude. At that time, convert the location to the latitude and the longitude and provide them to the function. The date is required to be included. The date format is "yyyy-MM-dd HH:mm". If you cannot know the location, decide the location using the timezone.

4. `get_current_date_time`
The current date and time are returned. Please provide your timezone. If you don't know the timezone, it is automatically detected with the script.

5. `analytics_admin_accountSummaries_list`
Use to retrieve a list of all Google Analytics accounts accessible by the current user. Each entry provides key details for the account and a summary of its properties, making it useful for discovering available data streams and managing permissions.

6. `analytics_admin_properties_get`
Use to get detailed information about a single Google Analytics property, providing essential details for management and analysis. Use this to confirm property settings or to retrieve its metadata.

7. `analytics_data_properties_runReport`
Use to fetch a custom report from a Google Analytics property. Specify the metrics (e.g., active users, event count) and dimensions (e.g., country, event name) to retrieve specific user activity data. This tool is best for answering questions about user behavior, such as "How many active users did we have in Japan last month?" or "What are the top 5 most popular events?"

8. `analytics_data_properties_runRealtimeReport`
Use to generate a customized report of real-time event data from a Google Analytics property, showing events and user activity that occurred within the last 30 minutes. Useful

Currently, Google Antigravity restricts the total number of active tools across all MCP servers. While reducing the count to 80 may clear the

error, it might still trigger a performance warning: `Warning: To optimize Agent, we recommend that up to 50 tools are enabled.`

Recommended Solution:

To ensure stability, I recommend reducing the number of enabled tools to approximately **25**. As shown below, this configuration removes both errors and warnings. I modified `export const tools = [,,,]` at the bottom of the script of `tools-for-mcp-server-extension/mcp-server/src/tools.js`. I believe that this limitation is expected to be resolved in future updates.

Press enter or click to view image in full size

Manage MCP servers 25 / 100 tools View raw config Refresh

file-search-store-extension	tools-for-mcp-server-extension	Configure	Enabled
11 / 11	14 / 58		<input checked="" type="checkbox"/>
tools-for-mcp-server-extension	Configure		<input checked="" type="checkbox"/>
	1. get_exchange_rate		<input checked="" type="checkbox"/>
	Use this to get the current exchange rate. Using this, for example, it can exchange yen for dollars.		
	2. get_current_weather		<input checked="" type="checkbox"/>
	Use this to get the current weather using the latitude and the longitude. At that time, convert the location to the latitude and the longitude and provide them to the function. If you cannot know the location, decide the location using the timezone.		
	3. get_specific_date_weather		<input checked="" type="checkbox"/>
	Use this to get the weather for the specific date using the latitude and the longitude. At that time, convert the location to the latitude and the longitude and provide them to the function. The date is required to be included. The date format is "yyyy-MM-dd HH:mm" If you cannot know the location, decide the location using the timezone.		
	4. get_current_date_time		<input checked="" type="checkbox"/>
	The current date and time are returned. Please provide your timezone. If you don't know the timezone, it is automatically detected with the script.		
	5. search_schedule_on_Google_Calendar		<input checked="" type="checkbox"/>
	Use to search the schedules (events) on Google Calendar by providing the date range.		
	6. create_schedule_on_Google_Calendar		<input checked="" type="checkbox"/>
	Use to create a new schedule (event) on Google Calendar.		
	7. delete_schedules_on_Google_Calendar		<input checked="" type="checkbox"/>
	Use to delete schedules (events) from Google Calendar.		
	8. update_schedule_on_Google_Calendar		<input checked="" type="checkbox"/>
	Use to update the schedule (event) on Google Calendar.		
	9. get_values_from_google_docs		<input checked="" type="checkbox"/>
	Use this to get text from Google Docs in a text format or a markdown format. The document ID is used for retrieving the values from the Google Docs. If you use the document URL, get the document ID from the URL and use the ID.		
	10. put_values_into_google_docs		<input checked="" type="checkbox"/>
	Use this to append or insert text to Google Docs. The document ID is used for putting the values to the Google Docs. If you use the document URL, get the document ID from the URL and use the ID.		

Note: If the prompt-based installation does not work in your specific environment, you may need to manually edit the `mcp_config.json` file with the paths shown above.

Use Cases & Walkthroughs

The following samples demonstrate the capabilities of the integrated environment.

Sample 1: Basic Query

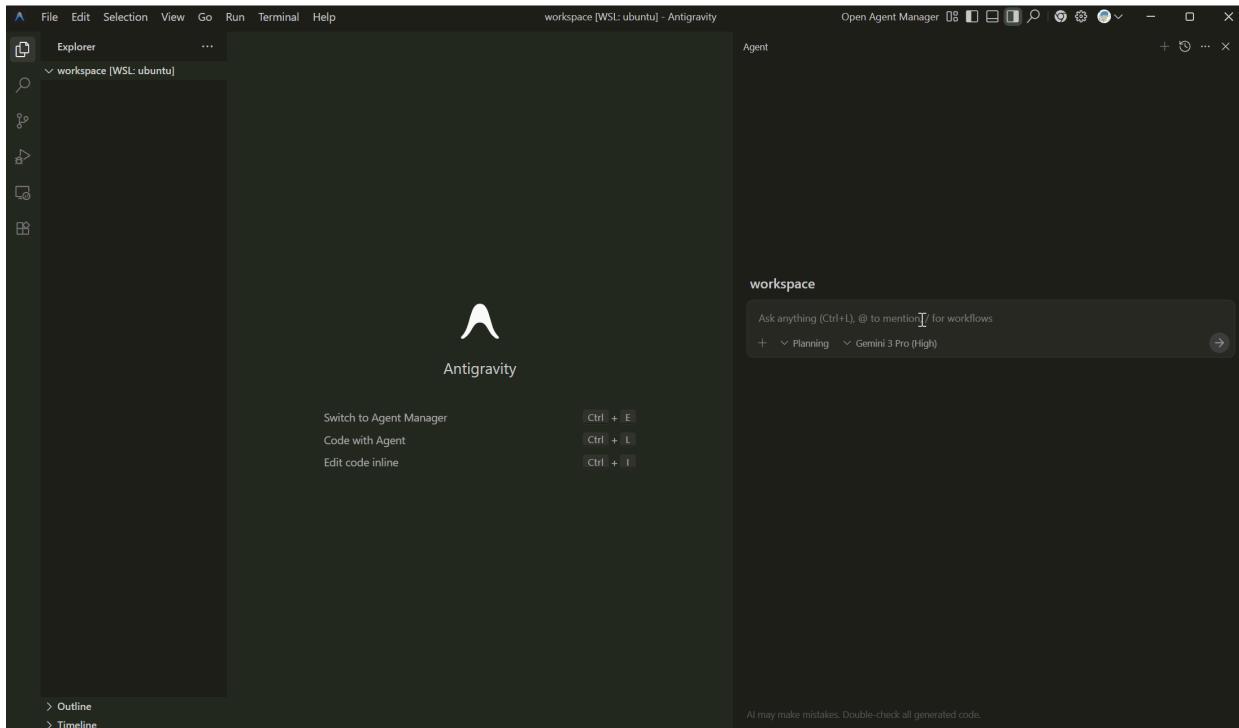
Objective: Retrieve real-time data (Weather).

Prompt:

What is the current weather in Tokyo?

Result:

Press enter or click to view image in full size



Observation: The MCP server of ToolsForMCPServer-extension successfully accessed external data to provide the answer.

Sample 2: Workflow Automation (Quiz Generation)

Objective: Create a quiz in Markdown and automatically generate a Google Form.

Prompt:

Your mission is as follows.

1. Generate a quiz. The quiz will be used to cultivate professional expertise in Google Apps Script. The quiz has 5 questions, and each question has five multiple-choice questions.
2. Create the questions and answers in a Markdown file as a Markdown format.
3. Generate a quiz using Google Forms.

Result:

Press enter or click to view image in full size



Implementation Plan:

```
# Google Apps Script Quiz Creation Plan
```

```
## Goal Description
```

Create a 5-question multiple-choice quiz about Google Apps Script, save it as Markdown, and generate a Google Form.

```
## Proposed Changes
```

```
### Documentation
```

```
#### [NEW] [quiz.md] (file:///home/tanaike/temp/workspace/quiz.md)
```

- Contains 5 multiple-choice questions with 5 options each.

```
### Automation
```

- Use `mcp1_generate_quiz_with_google_forms` to create the form.

```
## Verification Plan
```

```
### Manual Verification
```

- Review `quiz.md` for correctness.
- Check the generated Google Form URL to ensure questions are correct.

Walkthrough:

```
# Google Apps Script Quiz Walkthrough
```

I have successfully created the Google Apps Script Professional Expertise Quiz.

```
## Generated Assets
```

- ****Questions File**:**

```
[quiz.md] (file:///home/tanaike/temp/workspace/quiz.md)
```

- ****Google Form**:**

- [Edit URL] (<https://docs.google.com/forms/d/###/edit>)

- [Published URL] (<https://docs.google.com/forms/d/e/###/viewform>)

```
## Verification
```

- The form contains 5 multiple-choice questions.
- Each question has 5 options.
- The correct answer is set for each question (index 0 in the provided list, but randomized by Forms if configured, though the tool sets a specific correct index).

quiz.md:

```
# Google Apps Script Professional Expertise Quiz
```

```
## Question 1
```

Which method is primarily used to retrieve the currently active sheet within a Google Spreadsheet script?

- [x] SpreadsheetApp.getActiveSheet()
- [] SpreadsheetApp.getActiveSpreadsheet()

- [] SpreadsheetApp.getSheets()[0]
- [] SpreadsheetApp.openActive()
- [] SpreadsheetApp.setActiveSheet()

Question 2

Which method of the GmailApp service is used to send a basic email?

- [x] GmailApp.sendEmail(recipient, subject, body)
- [] GmailApp.createDraft(recipient, subject, body)
- [] GmailApp.send(recipient, subject, body)
- [] GmailApp.compose(recipient, subject, body)
- [] GmailApp.mail(recipient, subject, body)

Question 3

What is the correct method to create a text file in the root of Google Drive using DriveApp?

- [x] DriveApp.createFile(name, content)
- [] DriveApp.newFile(name, content)
- [] DriveApp.addFile(name, content)
- [] DriveApp.upload(name, content)
- [] DriveApp.insert(name, content)

Question 4

Which simple trigger function automatically runs when a user modifies a cell value in a spreadsheet?

- [x] onEdit(e)

- [] onChange(e)

- [] onOpen(e)

- [] onUpdate(e)

- [] onModify(e)

Question 5

Which method is used to perform an HTTP GET request to an external API?

- [x] UrlFetchApp.fetch(url)

- [] UrlFetchApp.get(url)

- [] UrlFetchApp.request(url)

- [] UrlFetchApp.send(url)

- [] UrlFetchApp.call(url)

Also, the following Google Form was created.

Press enter or click to view image in full size

Google Apps Script Professional Expertise Quiz

* 必須の質問です

Which method is primarily used to retrieve the currently active sheet within a Google Spreadsheet script? *

Select the correct method.

- SpreadsheetApp.getActiveSheet()
- SpreadsheetApp.getActiveSpreadsheet()
- SpreadsheetApp.getSheets()[0]
- SpreadsheetApp.openActive()
- SpreadsheetApp.setActiveSheet()

Which method of the GmailApp service is used to send a basic email? *

Select the correct method.

- GmailApp.sendEmail(recipient, subject, body)
- GmailApp.createDraft(recipient, subject, body)
- GmailApp.send(recipient, subject, body)
- GmailApp.compose(recipient, subject, body)
- GmailApp.mail(recipient, subject, body)

What is the correct method to create a text file in the root of Google Drive using DriveApp? *

Select the correct method.

- DriveApp.createFile(name, content)
- DriveApp.newFile(name, content)
- DriveApp.addFile(name, content)
- DriveApp.upload(name, content)
- DriveApp.insert(name, content)

Which simple trigger function automatically runs when a user modifies a cell value in a spreadsheet? *

Select the correct trigger function.

- onEdit(e)
- onChange(e)
- onOpen(e)
- onUpdate(e)
- onModify(e)

Which method is used to perform an HTTP GET request to an external API? *

Select the correct method.

- UrlFetchApp.fetch(url)
- UrlFetchApp.get(url)
- UrlFetchApp.request(url)
- UrlFetchApp.send(url)
- UrlFetchApp.call(url)

送信

フォームをクリア

Sample 3: Cross-Application Logic

Objective: Convert Markdown to Google Docs and email the result.

Prompt:

Your new mission is as follows.

1. Convert a file `quiz.md` to Google Docs.
2. Create an email including the converted Google Docs. The title is "sample email". Think of the email body by understanding the file.
3. Send an email to tanaike@hotmail.com

Result:

Press enter or click to view image in full size

The screenshot shows the Antigravity IDE interface. The top menu includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar indicates "workspace [WSL: ubuntu] - Antigravity - quiz.md". The left sidebar has an "Explorer" tab selected, showing a workspace folder with "quiz.md". The main editor area displays the following Google Apps Script code:

```
1 # Google Apps Script Professional  
2 # Expertise Quiz  
3 ## Question 1  
4 Which method is primarily used to  
5 retrieve the currently active  
6 sheet within a Google Spreadsheet  
7 script?  
8 - [x] SpreadsheetApp.  
9 getActiveSheet()  
10 - [ ] SpreadsheetApp.  
11 getActiveSpreadsheet()  
12 - [ ] SpreadsheetApp.getSheets()  
13 - [ ] SpreadsheetApp.openActive()  
14 - [ ] SpreadsheetApp.  
15 setActiveSheet()  
16 ## Question 2  
17 Which method of the GmailApp  
18 service is used to send a basic  
19 email?  
20 - [x] GmailApp.sendEmail  
21 (recipient, subject, body)  
22 - [ ] GmailApp.createDraft  
23 (recipient, subject, body)  
24 - [ ] GmailApp.send(recipient,  
25 subject, body)  
26 - [ ] GmailApp.compose(recipient,
```

The code is color-coded for syntax. A tooltip "Creating Google Apps Script Quiz" is visible at the bottom right. The status bar at the bottom says "AI may make mistakes. Double-check all generated code." and "15m".

The agent successfully bridged DriveApp (file conversion) and GmailApp (sending email). The recipient received a context-aware email containing the link to the newly created Google Doc.

Also, I got an email as follows.

Here is the converted Google Doc for the Google Apps Script

Professional Expertise Quiz.

<https://docs.google.com/document/d/###/edit>

Sample 4: RAG with File Search Store

Objective: Ingest external web content into a local vector store (RAG) and generate a summary.

Prompt:

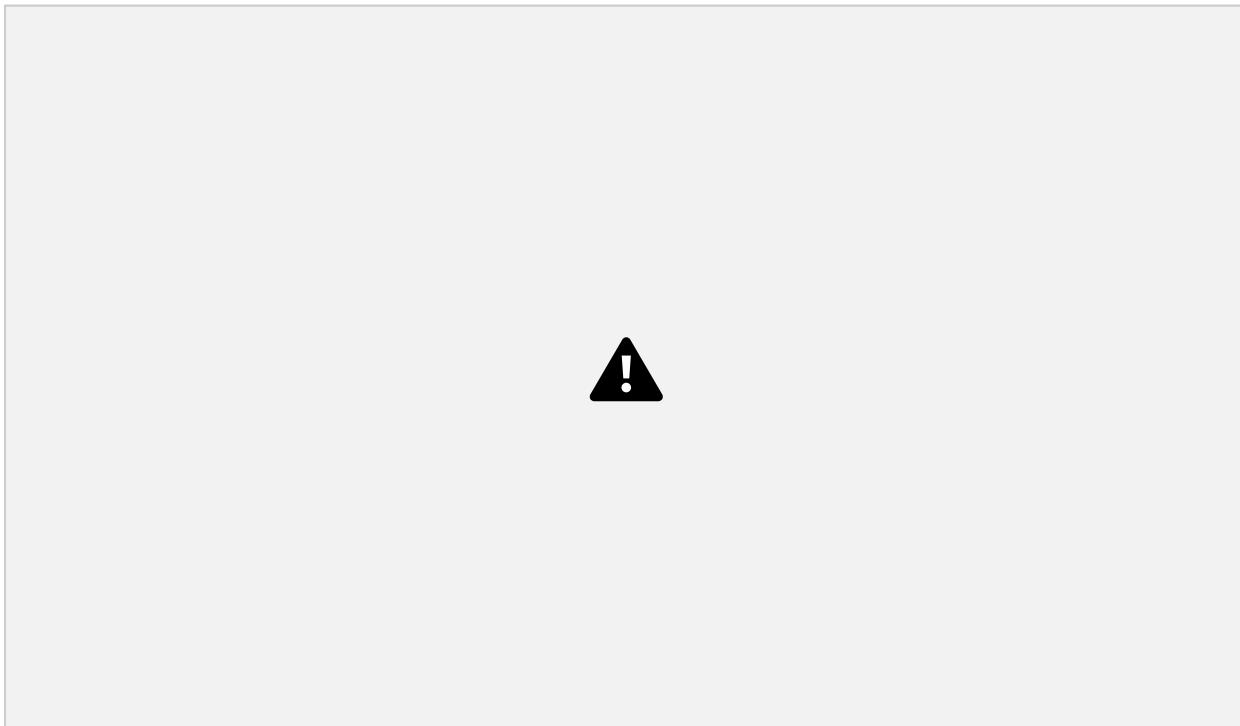
Your mission is as follows.

1. Create a new File Search Store.
2. Get the data from <https://tanaikech.github.io/about/>
3. Upload the data to the File Search Store.
4. Generate content with a prompt `Who is Kanshi Tanaike?` using the File Search Store.

5. Summarize the result.

Result:

Press enter or click to view image in full size



Implementation Plan:

```
# Implementation Plan - File Search Store Operations
```

Goal

Create a File Search Store, populate it with data from a specific URL, and use it to answer a question.

Proposed Changes

No code changes to the codebase. This is an operational task using MCP tools.

Verification Plan

- Verify File Search Store creation via tool output.
- Verify content retrieval via tool output.
- Verify upload success via tool output.
- Verify generation result matches the expected context (about Kanshi Tanaike).

Walkthrough:

```
# Walkthrough - File Search Store Operations
```

```
## Overview
```

This walkthrough documents the creation of a File Search Store, populating it with data from a URL, and generating content based on that data.

```
## Steps Taken
```

1. **Created File Search Store:**

- Name: `Tanaike Profile Store`
- Resource Name:
`fileSearchStores/tanaike-profile-store-v5u0bnofjgi9`

2. ****Retrieved Content**:**

- Source: `https://tanaikech.github.io/about/`
- Content fetched and combined from "About Me", "History", and "Activities" sections.

3. ****Uploaded Content**:**

- Uploaded as text media to the store.

4. ****Generated Content**:**

- Prompt: "Who is Kanshi Tanaike?"
- Result:

> Kanshi Tanaike is a Japanese scientist who holds a Ph.D. in Physics. He is recognized as a Google Developer Expert (GDE) in Google Workspace and achieved first place in the Outstanding GDE Award on September 26, 2025.

>

> His background includes:

> * Becoming a Google Developer Expert (Google Workspace) on May 4, 2020.

> * Being named a Google Cloud Champion Innovator (Google Workspace) on October 5, 2021.

> * The Google Cloud Champion Innovators program merging with the Google Developer Experts program on March 22, 2025.

>

> Kanshi Tanaike is driven by a strong curiosity to explore physics and create new solutions, some of which have found practical uses in fields like electronics, industrial machinery, architecture, and aerospace. He is active online, with presences on platforms such as GitHub, Gists, his personal homepage, Qiita (for tips in Japanese), Stackoverflow, and X (formerly Twitter).

```
## Verification
```

- The File Search Store was successfully created.
- Content was successfully retrieved and uploaded.
- The generated response accurately reflects the information from the source URL.

Summary

- **Agent-First Architecture:** Google Antigravity combined with Gemini 3.0 allows for complex, autonomous task execution rather than simple command processing.
- **MCP Integration:** The Model Context Protocol enables seamless connectivity between the AI agent and local tools or custom scripts.

- **Performance Optimization:** Currently, limiting the number of active tools (approx. 25) is crucial for maintaining stability and avoiding error limits in Antigravity.
- **Workspace Automation:** Through `ToolsForMCPServer-extension`, agents can control Google Apps Script to automate Forms, Docs, and Gmail workflows.
- **Private RAG Capability:** The `FileSearchStore-extension` allows agents to build on-the-fly knowledge bases from private files or URLs for grounded content generation.