Architecting Autonomous Intelligence: The Integration of Google Antigravity and Obsidian for Persistent Agentic Knowledge Systems

The evolution of artificial intelligence from conversational chatbots to autonomous agents represents a fundamental shift in the computational landscape. As of late 2025 and early 2026, the emergence of platforms like Google Antigravity has redefined the interface through which large language models, specifically frontier models such as Gemini 3.1 Pro and Claude 3.5 Sonnet, interact with the physical and digital world.[1] A critical challenge in this agentic era is the management of persistent knowledge. While traditional models suffer from transient context windows and "context rot," the integration of local-first knowledge management systems like Obsidian provides a sovereign, durable, and interlinked memory bank.[3] This report explores the technical architecture of Google Antigravity, the command-line capabilities of the Obsidian CLI, the specialized agent skills that empower models to master Obsidian's unique formats, and the practical workflows through which agents can preserve and link their own learning to achieve long-horizon task autonomy.

The Architectural Paradigm of Google Antigravity

Google Antigravity is not merely a development environment; it is an agent-first platform designed to facilitate the orchestration of multiple AI agents working across different workspaces.[1] Unlike standard IDEs that embed AI as a sidebar assistant, Antigravity flips the paradigm by embedding the editor, terminal, and browser surfaces into the agent's workflow.[5] This allows agents to operate at a task-oriented level, where the human developer supervises high-level plans rather than micro-managing code completions.[1]

Multi-Surface Orchestration and the Agent Manager

The Antigravity platform introduces the Agent Manager, a dedicated "mission control" interface.[5] Within this surface, developers can spawn and observe multiple agents working asynchronously.[1] For instance, one agent may be assigned to refactor a backend service while another simultaneously updates the frontend UI and a third conducts browser-based testing to verify integration.[2] This asynchronous pattern is essential for long-running maintenance tasks or complex bug fixes that would otherwise require constant context switching from the human user.[1]

The intelligence powering these agents is derived from the Gemini 3 series, with Gemini 3.1 Pro serving as the flagship reasoning model.[9] The platform also supports model optionality, including Anthropic's Claude Sonnet 4.5 and OpenAI's GPT-OSS-120b, allowing developers to choose the model best suited for specific task complexities.[1]

Model Tier	Primary Function within Antigravity	Key Capability
Gemini 3.1 Pro	Core Reasoning & Orchestration	Advanced logic, long-horizon planning, and tool use.[9]
Gemini 3 Flash	Background Tasks & Summarization	Fast context processing and checkpointing.[9]
Gemini 2.5 Computer Use	Browser Actuation	Direct control of Chrome for UI testing and research.[8]
Gemini 2.5 Flash Lite	Semantic Search	Powers the codebase semantic search tool.[9]
Claude Sonnet 4.6	Alternative Reasoning	Specialized in complex code refactors and predictable terminal ops.[2]

Trust through Artifacts and Verification

A core tenet of the Antigravity platform is the establishment of trust through transparency.[5] Instead of forcing users to audit thousands of raw tool calls, Antigravity agents generate "Artifacts"—tangible, human-readable deliverables that summarize the agent’s logic and progress.[1] These include task lists, implementation plans, screenshots of browser-based tests, and recordings of agent actions.[5] Users can provide feedback directly on these artifacts, using a Google Doc-style commenting system, which the agent incorporates in real-time without stopping its execution flow.[1]

This approach shifts the developer's role from writing code to reviewing "proof of work".[6] The agent is expected to think through the verification of its work, not just the work itself, ensuring that every code change is backed by a successful test or a visual walkthrough.[5]

Learning as a Core Primitive: Antigravity Knowledge Items

Antigravity distinguishes itself from predecessors by treating learning as a core primitive.[1] Agents do not start every session with a blank slate; instead, they both contribute to and retrieve from a persistent knowledge base.[5] This system captures what the platform refers to as "Knowledge Items" (KIs)—collections of related information on specific topics derived from past coding sessions.[5]

The Mechanics of Knowledge Extraction

As an agent interacts with a codebase and the user, it automatically analyzes the conversation to extract significant insights, recurring patterns, and derived architecture.[5] If an agent solves a particularly challenging configuration issue or develops a novel sync strategy for a local-first application, it preserves that solution as a Knowledge Item.[8] Each KI contains a title, a summary, and a collection of artifacts, such as code snippets or memories of specific user instructions.[13]

The persistence of this knowledge base across sessions allows the agent to build a long-term memory of the codebase and the developer's decisions.[8] When a new task is initiated, the agent scans the summaries of existing KIs; if a relevant item is identified, the agent "studies" the associated artifacts to ensure that it does not repeat past mistakes or ignore previously established constraints.[5]

Scoping Knowledge: Global vs. Workspace

Knowledge in Antigravity is organized into two primary scopes to manage relevance and privacy.[14] This dual-scope architecture ensures that project-specific secrets or proprietary boilerplate remain contained while general productivity improvements are shared across the developer's entire machine.[14]

Scope	Storage Location	Intended Usage
Workspace Scope	<project-root>/.agent/skills/	Project-specific deployment scripts, database schemas, and boilerplate generation.[12]
Global Scope	~/.gemini/antigravity/skills/	General utilities like JSON formatting, UUID generation, and code style reviewers.[12]

Obsidian as the Sovereign Memory for AI Agents

While Antigravity provides internal Knowledge Items, the developer community has increasingly leveraged Obsidian as a more comprehensive, human-auditable "second brain" for AI agents.[3] Obsidian's philosophy of "file over app"—storing all data in a local folder of plain-text Markdown files—makes it uniquely suited for consumption by agents that have direct filesystem access.[4]

The Multi-Root Workspace Integration

The most direct method for integrating Gemini 3.1 Pro with an Obsidian vault is through Antigravity's support for multi-root workspaces.[18] Because Antigravity operates like a high-powered IDE, adding the Obsidian vault folder as a root allows the agent full read/write access to the notes without requiring additional plugins.[18] This setup enables the agent to act as a librarian and editor for the user's knowledge base, refining messy data dumps, organizing project notes, and linking disparate ideas while the user works on code in a separate root.[18]

This integration transforms Obsidian from a passive note-taking tool into a dynamic, queryable knowledge base.[3] By using Antigravity's terminal and filesystem tools, the agent can treat the vault as a structured database, leveraging standard Unix utilities to search, filter, and transform the knowledge stored within.[4]

The Agent Client Protocol (ACP) and Vault Awareness

Beyond simple file access, the "Agent Client" plugin for Obsidian provides a bridge for agents using Zed's Agent Client Protocol (ACP).[20] This plugin allows agents like Claude Code or the Gemini CLI to run directly within the Obsidian environment, providing a "vault-aware" conversational interface.[16]

Key features of the Agent Client integration include:

- **Native Contextual Awareness:** Agents can reference vault notes using @notename or @[[note name]] during conversations.[20]
- **Session Persistence:** Every AI conversation can be automatically exported as a Markdown note, ensuring that the reasoning behind a project's evolution is preserved within the vault.[20]
- **Auto-Mention Mode:** The agent can be configured to automatically ingest the context of the current active note, facilitating seamless iterative editing and research synthesis.[20]

Command Line Capabilities of the Obsidian CLI

The release of the official Obsidian CLI (v1.12) marks a significant milestone for agentic automation.[19] Command-line interfaces are the "natural language" of AI agents, enabling them to chain together complex operations that would be cumbersome in a GUI.[19] The Obsidian CLI provides functionality equivalent to the GUI, allowing agents to manipulate the vault through a set of structured commands.[19]

CLI Operation and IPC Singleton Lock

The Obsidian CLI operates by communicating with a running instance of the Obsidian application via an Inter-Process Communication (IPC) singleton lock.[22] This means that for CLI commands to execute, Obsidian must be open on the host machine.[22] For non-interactive uses like cron jobs or background scripts, agents must first ensure the application is active.[22] The CLI supports a "silent" mode (now default) which allows it to perform operations without stealing focus from the user's active window.[19]

Core Command Reference for AI Agents

Agents utilize a wide array of CLI commands to manage the knowledge lifecycle, from ingestion and search to linking and task management.[21]

Category	Command	Parameters & Utility
Vault Info	obsidian vault	Displays vault name, path, file count, and total size.[22]
Navigation	obsidian daily	Opens or reads the current day's daily note; essential for chronological logging.[19]
Reading	obsidian read	Reads a note by name (with Wikilink resolution) or exact path.[21]
Searching	obsidian search	Full-text search with options for context matching (--context), limit, and JSON output.[22]
Tasks	obsidian tasks	Batch operations for checkboxes; can filter for todo or done states in specific files.[21]
Metadata	obsidian properties	Reads or sets YAML frontmatter; allows agents to manage notes like a database.[21]

Structure	obsidian backlinks	Identifies files linking to a target note, enabling graph-based reasoning.[22]
Automation	obsidian create	Generates a new note from a specified template or with initial content.[19]

Advanced Scripting and TUI Integration

The CLI also includes a Text User Interface (TUI) mode, which allows for fast keyboard-driven navigation, and a developer mode (dev:eval) for executing JavaScript directly within the Obsidian context.[21] Agents can use these developer features to trigger complex plugin behaviors or retrieve DOM elements from the Obsidian interface.[19] By combining the CLI with standard shell commands like grep, sed, and awk, agents can perform sophisticated data aggregation across the entire vault in seconds.[19]

Specialized Agent Skills for Obsidian Mastery

In the Antigravity and Claude Code ecosystems, "Skills" are modular capability extensions that move beyond general text generation to deep vertical domain integration.[26] The "Obsidian Skills" package, officially maintained and distributed via repositories like kepano/obsidian-skills, empowers agents to understand and generate content using Obsidian's unique syntax and file formats.[16]

The SKILL.md Architecture

A Skill is a directory-based package containing a SKILL.md file, which serves as the "brain" of the capability.[14] This file defines the trigger phrases, instructions, examples, and constraints that govern the agent's behavior.[14] When an agent encounters a task that matches the description in the SKILL.md frontmatter, it "activates" the skill, loading the full set of instructions into its context.[26]

Example SKILL.md Frontmatter for Obsidian Mastery:

YAML

```yaml
name: obsidian-helper
description: Use this skill to manage an Obsidian vault, including creating notes with Wikilinks, Callouts,
```

and YAML properties.

Mastery of Obsidian-Flavored Markdown

Without specialized skills, agents often default to standard Markdown, which lacks the advanced linking and semantic features that make Obsidian powerful.[26] The Obsidian Markdown skill ensures that agents correctly implement:

- **Wikilinks:** Using [[Note Name]] for bidirectional linking and [[Note#Heading]] or [[Note#^block-id]] for granular references.[26]
- **Callouts:** Employing semantic information boxes like > [!tip], > [!warning], or > [!bug] to highlight critical insights.[26]
- **Embeds:** Implementing the DRY (Don't Repeat Yourself) principle by embedding note fragments using !].[26]
- **Properties:** Managing structured YAML metadata to facilitate queries via community plugins like Dataview.[26]

Architecting Data with Obsidian Bases

One of the more advanced skills is the "Obsidian Bases Manager," which allows agents to interact with the .base file format.[29] Obsidian Bases are YAML-based files that define dynamic, database-like views of notes within a vault.[29]

Feature of Obsidian Bases	Agentic Capability	Use Case
Multi-View Layouts	Create tables, cards, lists, and maps.[30]	Project dashboards and visual media galleries.[30]
Global Filters	Apply recursive logical operators (and, or, not).[30]	Automated triage of tasks or high-priority research notes.[26]
Custom Formulas	Define calculated properties and logic.[30]	Statistical reporting on data-heavy note collections.[30]
Bi-directional Sync	Editing a cell in the Base mirrors the change in the underlying file.[31]	Bulk updates of note metadata during project migrations.[26]

Through this skill, an agent like Gemini 3.1 Pro can architect complex information structures, transforming a collection of static notes into a dynamic management system for tasks, contacts, or inventory.[26]

Visual Thinking with JSON Canvas

The JSON Canvas skill allows agents to create and edit .canvas files, which follow the open JSON Canvas Spec 1.0.[32] This empowers agents to engage in visual planning, mind mapping, and flowcharting.[32]

Agents can programmatically:

- **Create Nodes:** Define text nodes (Markdown), file nodes (attachments), link nodes (external URLs), and group nodes (visual containers).[32]
- **Connect Edges:** Draw relationships between nodes with custom labels, colors, and end shapes (e.g., arrows).[32]
- **Manage Layouts:** Specify pixel-perfect coordinates (x, y) and dimensions (width, height) to ensure readable, aligned diagrams.[32]

This skill is particularly valuable for "Agentic Planning," where an agent visualizes its task plan on a canvas for human review, improving transparency and auditability.[34]

The Three-Layer Stack: Automated Research and Learning

A practical application of this ecosystem is the "Three-Layer Stack," which connects NotebookLM, Antigravity, and Obsidian into a single, automated research pipeline.[3] This workflow allows users to process unlimited sources—PDFs, articles, YouTube videos—without manual copying or lost context.[3]

1. **Layer 1: NotebookLM (Research Engine):** Ingests and summarizes massive amounts of source material, providing a 200K+ token context window for initial synthesis.[3]
2. **Layer 2: Antigravity (Automation Bridge):** An agent in Antigravity uses an MCP (Model Context Protocol) server to programmatically query the notebooks in NotebookLM. It executes custom AI skills to extract key findings and define research workflows.[3]
3. **Layer 3: Obsidian (Knowledge Canvas):** The extracted findings flow directly into Obsidian. The agent uses its specialized skills to transform these results into interconnected permanent notes, complete with Wikilinks, tags, and YAML metadata.[3]

This compounding research archive ensures that knowledge doesn't disappear into transient chat logs but instead contributes to a living, queryable knowledge base that becomes more valuable with every project.[3]

Gemini 3.1 Pro: Reasoning over the Knowledge Graph

The capabilities of Gemini 3.1 Pro are central to the effectiveness of the Antigravity-Obsidian integration.[10] Built for tasks requiring advanced reasoning and long-horizon planning, Gemini 3.1 Pro is designed to synthesize dense research into functional output.[10]

Long-Horizon Task Management

In Antigravity, Gemini 3.1 Pro acts as an "autonomous actor" capable of navigating complex engineering tasks with minimal intervention.[10] For example, when asked to perform a database migration, the model does not simply write a script; it generates a structured implementation plan, assesses risks, architects a local-first sync engine, and generates unit tests for the matching logic.[10]

The model's ability to "think first" is facilitated by the Planning Mode in Antigravity.[12] Before touching any files, the agent reasons about the project structure and determines which stack-specific skills (e.g., PostgreSQL, Tailwind, Drizzle) should be activated.[35] This planning-first approach addresses the "runaway changes" problem common in earlier AI coding assistants, where models would proceed with edits without a cohesive architectural strategy.[2]

Semantic Search and Knowledge Retrieval

While the Obsidian CLI provides keyword-based search, agents often require semantic retrieval to find relevant context in large vaults.[38] Plugins like "Vector Search" and "EzRAG" provide agents with semantic search capabilities powered by embedding models (e.g., nomic-embed-text) or the Gemini File Search API.[38]

Plugin	Mechanism	Backend	Advantage for Agents
Vector Search	Semantic Indexing	Ollama (Local)	Fast, private, finds similar meanings without keywords.[38]
EzRAG	Chat-based Retrieval	Gemini API	Easy integration for Claude/Gemini agents to "chat with vault".[40]
MCP Advanced	Graph Analysis	Local REST API	Maps vault

			structure and connections for deep context.[41]
Vault Chat	Contextual RAG	OpenAI/Gemini API	High-level Q&A over the entire knowledge base.[38]

These tools enable the agent to find hidden connections in the user's ideas, suggesting related notes or previous solutions that go beyond simple text matching.[8]

Workflow Strategies: How Agents Preserve and Link Learning

For a Gemini or Claude agent to effectively preserve its own learning, it must follow a methodical workflow that integrates with Obsidian's structural conventions.[16] This "self-improvement" cycle allows the agent to build a persistent memory of the decisions, constraints, and patterns established during a project.[5]

The Documentation-First Workflow

A highly effective strategy for "good" vibe coding is the documentation-first approach.[2] Instead of generating code immediately, the agent is tasked with writing or extending separate documentation files within the vault.[2] These files describe:

- **System Architecture:** How the various components of the application interact.[2]
- **Data Models:** The structure of the information stored in the system.[2]
- **Operational Details:** How things are logged, failure modes, and security policies.[2]

By editing these documents before code generation, the agent and the human developer establish a "ground truth" that reduces context chaos and prevents duplicate code or surprise side effects.[2]

Automatic Backlinking and Synthesis

Agents are trained to use the Link System to build knowledge networks.[26] When an agent completes a subtask, it should not only update the source code but also create or update a "Knowledge Note" in the Obsidian vault.[13] This note should include:

1. **A Summary of the Task:** What was accomplished and why.[13]
2. **Key Implementation Details:** Important code snippets or configuration settings.[1]
3. **Wikilinks to Related Topics:** Connections to previous research notes or architectural decisions.[26]

4. **YAML Metadata:** Tags and properties that allow the note to be easily discovered by future agentic searches.[26]

This process of "active search and connection" ensures that every new task benefits from the accumulated wisdom of the vault.[44] The agent acts as its own archivist, ensuring that attribution and source tracking are maintained across long-term projects.[44]

Conclusion: The Sovereign Agentic Workspace

The integration of Google Antigravity and Obsidian represents a significant step toward a sovereign, autonomous development environment. By providing agents with a dedicated workspace, multi-surface control, and a persistent, interlinked knowledge base, the platform enables a new level of productivity and task complexity.[1] The Obsidian CLI provides the programmatic interface required for agents to master the vault, while specialized skills ensure that agents can leverage the full semantic power of Obsidian-flavored Markdown, Bases, and Canvases.[19]

As agents like Gemini 3.1 Pro continue to evolve, the ability to maintain a local-first, future-proof memory will be the differentiator between transient assistants and true autonomous partners.[3] The Antigravity-Obsidian ecosystem ensures that the intelligence generated by these agents remains in the hands of the developer, compounding over time to form a personalized, digital "second brain" that drives innovation and efficiency in the agentic era.

Works cited

1. Build with Google Antigravity, our new agentic development platform, accessed February 26, 2026, https://developers.googleblog.com/build-with-google-antigravity-our-new-agentic-development-platform/
2. Antigravity: Free Gemini 3 Pro, Claude 3.5 Sonnet, and My Vibe-Coding Workflow - AI Mind, accessed February 26, 2026, https://pub.aimind.so/antigravity-free-gemini-3-pro-claude-3-5-sonnet-and-my-vibe-coding-workflow-6ea5a1305623
3. I Connected NotebookLM + AntiGravity + Obsidian Into One AI Research Agent - Reddit, accessed February 26, 2026, https://www.reddit.com/r/startups_promotion/comments/1qon7sj/i_connected_notebooklm_antigravity_obsidian_into/
4. How I Use AI With My Obsidian Vault Every Day: 16 Practical Use Cases, accessed February 26, 2026, https://www.dsebastien.net/how-i-use-ai-with-my-obsidian-vault-every-day-16-practical-use-cases/
5. Introducing Google Antigravity, a New Era in AI-Assisted Software Development, accessed February 26, 2026,

https://antigravity.google/blog/introducing-google-antigravity

6. Google Antigravity Explained: From Beginner to Expert Guide - Helply, accessed February 26, 2026, https://helply.com/blog/google-antigravity-explained
7. Google Antigravity Features - AI Agents, Multi-Model Support & More, accessed February 26, 2026, https://antigravity.im/features
8. Google Antigravity and Gemini 3: A New Era of Agentic Development - Medium, accessed February 26, 2026, https://medium.com/@vfcarida/google-antigravity-and-gemini-3-a-new-era-of-agentic-development-f952ffe93b19
9. Models - Google Antigravity Documentation, accessed February 26, 2026, https://antigravity.google/docs/models
10. Gemini 3.1 Pro, Building with Advanced Intelligence in Google Antigravity, accessed February 26, 2026, https://antigravity.google/blog/gemini-3-1-pro-in-google-antigravity
11. Gemini 3.1 Pro: A smarter model for your most complex tasks - Google Blog, accessed February 26, 2026, https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/
12. Getting Started with Google Antigravity, accessed February 26, 2026, https://codelabs.developers.google.com/getting-started-google-antigravity
13. Knowledge - Google Antigravity Documentation, accessed February 26, 2026, https://antigravity.google/docs/knowledge
14. Authoring Google Antigravity Skills, accessed February 26, 2026, https://codelabs.developers.google.com/getting-started-with-antigravity-skills
15. Tutorial : Getting Started with Google Antigravity Skills - Medium, accessed February 26, 2026, https://medium.com/google-cloud/tutorial-getting-started-with-antigravity-skills-864041811e0d
16. I put Claude Code inside Obsidian, and it was awesome - XDA, accessed February 26, 2026, https://www.xda-developers.com/claude-code-inside-obsidian-and-it-was-eye-opening/
17. Obsidian vs Notion (2026): Features, Graph View, Pricing & Which Is Best for You - Pixno, accessed February 26, 2026, https://photes.io/blog/posts/obsidian-vs-notion
18. Using antigravity (Gemini 3) to read/write/manage my project vault (no plug-ins) - Reddit, accessed February 26, 2026, https://www.reddit.com/r/ObsidianMD/comments/1pijwcj/using_antigravity_gemini_3_to_readwritemanage_my/
19. CLI is ALL you need - DEV Community, accessed February 26, 2026, https://dev.to/lucifer1004/cli-is-all-you-need-4n2o
20. New Plugin: Agent Client - "Bring Claude Code, Codex & Gemini ..., accessed February 26, 2026, https://forum.obsidian.md/t/new-plugin-agent-client-bring-claude-code-codex-gemini-cli-inside-obsidian/108448

21. The Complete Obsidian CLI Setup Guide: A Record of Overcoming Windows Pitfalls - Zenn, accessed February 26, 2026, https://zenn.dev/sora_biz/articles/obsidian-cli-setup-guide?locale=en
22. skills/skills/adolago/obsidian-cli/SKILL.md at main · openclaw/skills ..., accessed February 26, 2026, https://github.com/openclaw/skills/blob/main/skills/adolago/obsidian-cli/SKILL.md
23. Obsidian 1.12.2 Desktop (Early access), accessed February 26, 2026, https://obsidian.md/changelog/2026-02-18-desktop-v1.12.2/
24. Changelog - Obsidian, accessed February 26, 2026, https://obsidian.md/changelog/
25. How are you using CLI besides AI? : r/ObsidianMD - Reddit, accessed February 26, 2026, https://www.reddit.com/r/ObsidianMD/comments/1r9ezpw/how_are_you_using_cli_besides_ai/
26. Obsidian Skills — Empowering AI Agents to Master Obsidian Knowledge Management | by Addo Zhang | Feb, 2026, accessed February 26, 2026, https://addozhang.medium.com/obsidian-skills-empowering-ai-agents-to-master-obsidian-knowledge-management-8b4f6d844b34
27. Master Google Antigravity Skills: Build Autonomous AI Agents | VERTU, accessed February 26, 2026, https://vertu.com/lifestyle/mastering-google-antigravity-skills-a-comprehensive-guide-to-agentic-extensions-in-2026/
28. Agent Skills Deep Dive: Building a Reusable Skills Ecosystem for AI Agents | by Addo Zhang, accessed February 26, 2026, https://addozhang.medium.com/agent-skills-deep-dive-building-a-reusable-skills-ecosystem-for-ai-agents-ccb1507b2c0f
29. accessed February 26, 2026, https://lobehub.com/ru/skills/davisbuilds-dojo-obsidian-bases#:~:text=Obsidian%20Bases%20are%20YAML%2Dbased,property%20configurations%2C%20and%20custom%20summaries.
30. Obsidian Bases Manager - Claude Code Skill - MCP Market, accessed February 26, 2026, https://mcpmarket.com/tools/skills/obsidian-bases-manager-7
31. Obsidian Bases — What are they good for (And what are they not?) | by Nick Felker, accessed February 26, 2026, https://fleker.medium.com/obsidian-bases-what-are-they-good-for-and-what-are-they-not-da620006cb34
32. json-canvas | Skills Marketplace - LobeHub, accessed February 26, 2026, https://lobehub.com/skills/einverne-agent-skills-json-canvas
33. json-canvas | Skills Marketplace - LobeHub, accessed February 26, 2026, https://lobehub.com/skills/kepano-obsidian-skills-json-canvas
34. Unveiling the JSON Canvas MCP Server: A Guide for AI Engineers - Skywork.ai, accessed February 26, 2026, https://skywork.ai/skypage/en/json-canvas-mcp-server-ai-engineers/1978652666416635904
35. Building with Gemini 3.1 Pro: The Ultimate Coding Agent Tutorial - DataCamp,

accessed February 26, 2026, https://www.datacamp.com/tutorial/building-with-gemini-3-1-pro-coding-agent-tutorial

36. Anti Gravity Explained: Google's Agent-First Development Platform - Zenn, accessed February 26, 2026, https://zenn.dev/neotechpark/articles/578723a5457e76
37. Google Antigravity vs Claude Code: Agent-First Development vs Terminal-First Control, accessed February 26, 2026, https://www.augmentcode.com/tools/google-antigravity-vs-claude-code
38. Obsidian plugin for Vector Search, accessed February 26, 2026, https://www.obsidianstats.com/plugins/vector-search
39. ashwin271/obsidian-vector-search: Obsidian plugin for Vector Search - GitHub, accessed February 26, 2026, https://github.com/ashwin271/obsidian-vector-search
40. EzRAG - Simple Semantic Search for Obsidian using Google Gemini : r/ObsidianMD - Reddit, accessed February 26, 2026, https://www.reddit.com/r/ObsidianMD/comments/1ozohwo/ezrag_simple_semantic_search_for_obsidian_using/
41. ToKiDoO/mcp-obsidian-advanced: Advanced MCP server ... - GitHub, accessed February 26, 2026, https://github.com/ToKiDoO/mcp-obsidian-advanced
42. All semantic-search Obsidian Plugins., accessed February 26, 2026, https://www.obsidianstats.com/tags/semantic-search
43. From Requirements to Release: Automated Development of Nexus MCP Server Using OpenClaw + Ralph Loop | by Addo Zhang | Feb, 2026, accessed February 26, 2026, https://addozhang.medium.com/from-requirements-to-release-automated-development-of-nexus-mcp-server-using-openclaw-ralph-loop-d6f9577d7997
44. interactive-writing-assistant | Skil... - LobeHub, accessed February 26, 2026, https://lobehub.com/ar/skills/jykim-claude-obsidian-skills-interactive-writing-assistant