

Level up your Gemini CLI - D&D Style

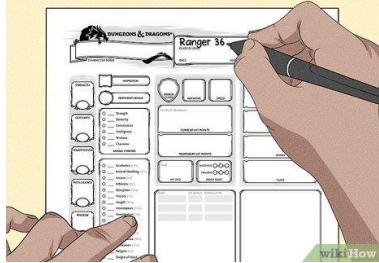
Abstract

You've heard about it, people talk about it, you might have tried it a few times too. But how many XP did you do on CLI? How many *hobgoblins* do you need to kill, MCP / Extensions do you need to write before getting to L36?

Riccardo is going to walk you through **L1** to **L36** in pure 1982's D&D style, from zero to hero and maybe even to Immortal!

Don't sweat on the slides: there will be **DEMOS** to show you how to progress and refactor your logic from simple to increasingly complex scenarios.

Note: this abstract was (astonishingly) *not* written by AI, but by a Googler's brain trained on multiple AI abstracts.



Level up your Gemini CLI!



Zurich, 13 Oct 2025

go/ricc-gemini-cli-slides-dnd



Riccardo Carlesso
Developer Advocate

About me



@SREccardo

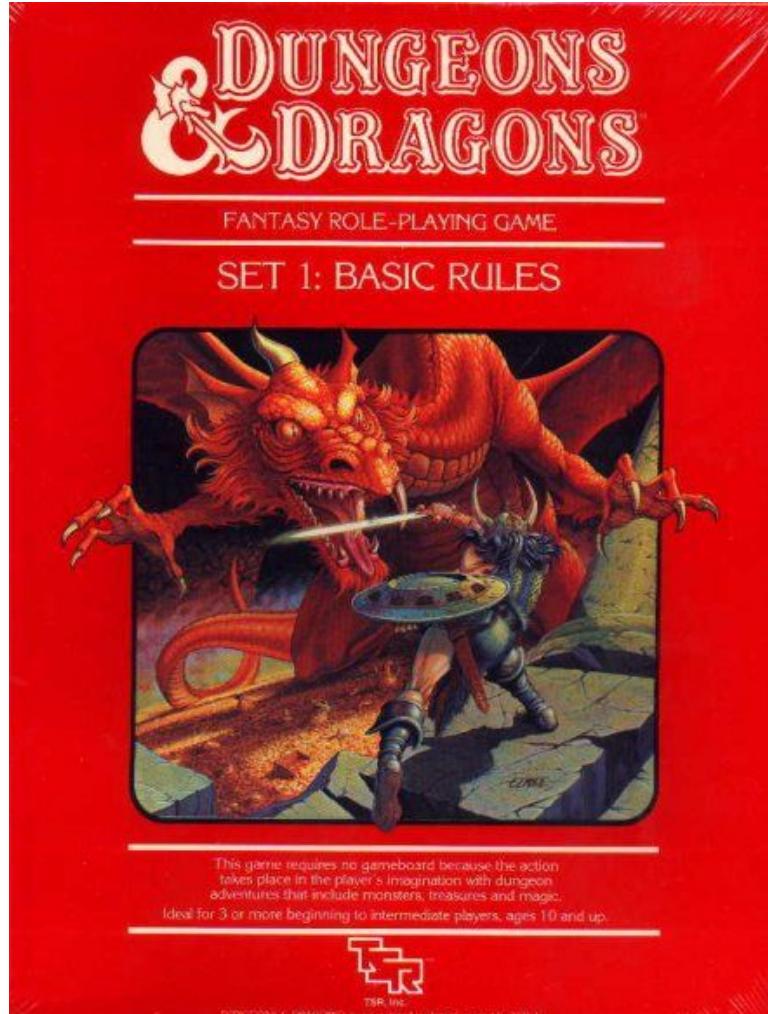
www.linkedin.com/in/riccardocarlesso/

Former sysadmin, Ruby [on Rails] developer, network admin.
13-years Googler, passionate about Reliability, SRE,
DevOps, **Maths**, yellow, piano, triathlons ,
caipirinhas and Amarone. **Bash** aficionado.
As a Developer Advocate for Google Cloud, I preach SRE,
DevOps, the Google Way of doing things to various
audiences.

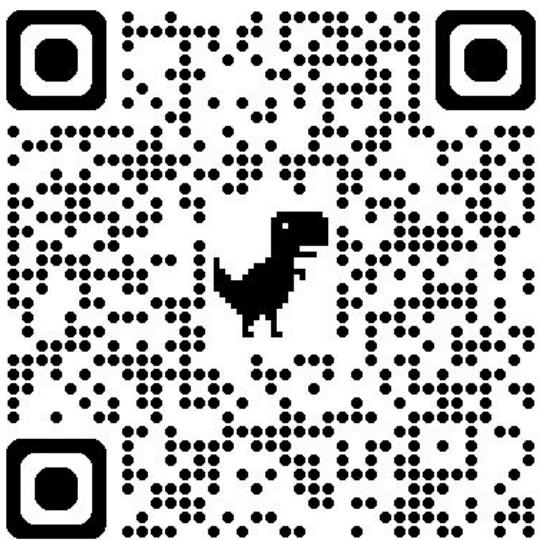


Gemini CLI: Basic

- *Installation*
- *First usage*



Open Source!



gemini-cli Private

Watch 26 Fork 21 Star 21

main 240 Branches 1 Tag Go to file Add file Code About

No description, website, or topics provided.

Readme Apache-2.0 license Security policy Activity Custom properties 21 stars 26 watching 21 forks

Releases 1 tags Create a new release

Packages No packages published Publish your first package

Contributors 40 + 26 contributors

abhipatel12 Show session summary on exit for ctrl+c x 2. Fix exit UI (#963) dd53e5c · 1 hour ago 744 Commits

.gcp refactor: rename gemini-code to gemini-cli (#B22) 4 days ago

.gemini toggle off (#809) 5 days ago

.github Update package-lock.json (#738) last week

.vscode refactor: rename gemini-code to gemini-cli (#B22) 4 days ago

docs add excludeTools flag to settings.json config (#957) 3 hours ago

eslint-rules feat: add custom eslint rule for cross-package imports (#... 2 months ago

packages Show session summary on exit for ctrl+c x 2. Fix exit UI (#... 1 hour ago

scripts fixes to proxy on macos; prevent curl from hanging during... 6 hours ago

.gitignore refactor: rename gemini-code to gemini-cli (#B22) 4 days ago

.npmrc refactor: rename gemini-code to gemini-cli (#B22) 4 days ago

.prettierrc.json Run npm run format 2 months ago

CONTRIBUTING.md fixes to proxy on macos; prevent curl from hanging during... 6 hours ago

Dockerfile restricted networking for all sandboxing methods, new se... yesterday

GEMINI.md Docs: Enhance "Writing Tests" section in GEMINI.md (#6... last week

LICENSE add LICENSE (#25) 2 months ago

Makefile Makefile for convenience (#833) 4 days ago

<https://github.com/google-gemini/gemini-cli>



ential

```
# Or: npx https://github.com/google-gemini/gemini-cli  
npm install -g @google/gemini-cli
```



Gemini CLI is a locally-run agentic application that orchestrates specialized agents



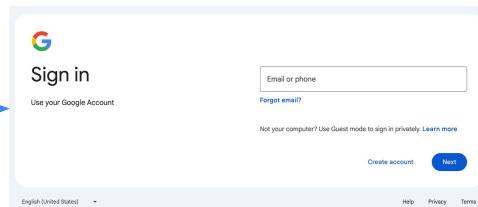
Authentication: /auth

```
> /auth

Get started

How would you like to authenticate for this project?

1. Login with Google
● 2. Use Gemini API Key
3. Vertex AI
```



gemini

```
# Get your key from https://aistudio.google.com/apikey
export GEMINI_API_KEY="YOUR_API_KEY"
gemini
```

```
# Set your Google Cloud Project
export GOOGLE_CLOUD_PROJECT="YOUR_PROJECT_NAME"
gemini
```

Five ways to authenticate them all..



Interactive and File/Shell Integration

/ commands

@ context

! shell

```
Error connecting to MCP server 'CloudRunBackgroundAgent': Connection failed for 'CloudRunBackgroundAgent': MCP error -32000: Connection closed

GEMINI

Tips for getting started:
1. Ask questions, edit files, or run commands.
2. Be specific for the best results.
3. /help for more information.

Using: 2 GEMINI.md files | 10 MCP servers (ctrl+t to view)
> Type your message or @path/to/file
```



Powerful Built-In Tools

File system tools (*read_file*, *write_file*,
list_directory, *search_file_content*),

Shell tool (*run_shell_command*)

Web tools (*web_fetch*,
google_web_search)

```
gent': MCP error -32000: Connection closed
Loaded cached credentials.

! Waiting for auth... (Press ESC to cancel)
```

A decorative graphic consisting of a grid of colored pixels (blue, purple, pink) arranged in a pattern that forms the word "GEMINI" when viewed from a distance.



Hierarchical Context & Memory

Project-specific and global instructions to the AI using **GEMINI.md** files

gent': MCP error -32000: Connection closed

GEMINI

Tips for getting started:

1. Ask questions, edit files, or run commands.
2. Be specific for the best results.
3. `/help` for more information.

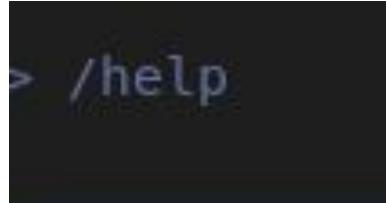
Using: 2 GEMINI.md files | 10 MCP servers (ctrl+t to view)

> Type your message or @path/to/file

~/java/bnb (dev*) no sandbox (see /docs) gemini-2.5-pro (100% context left)

Tip: /help

/help is your friend:



```
> ./help

Basics:
Add context: Use @ to specify files for context (e.g., @src/myFile.ts) to target specific files or folders.
Shell mode: Execute shell commands via ! (e.g., !npm run start) or use natural language (e.g. start server).

Commands:
/help - for help on gemini-cli
/clear - clear the screen and conversation history
/theme - change the theme
/auth - change the auth method
/editor - set external editor preference
/stats - check session stats
/mcp - list configured MCP servers and tools
/memory - manage memory. Usage: /memory <show|refresh|add> [text for add]
/tools - list available Gemini CLI tools
/about - show version info
/bug - submit a bug report
/save - save conversation checkpoint. Usage: /save [tag]
/resume - resume from conversation checkpoint. Usage: /resume [tag]
/quit - exit the cli
/compress - Compresses the context by replacing it with a summary.
! - shell command

Keyboard Shortcuts:
Enter - Send message
Shift+Enter - New line
Up/Down - Cycle through your prompt history
Alt+Left/Right - Jump through words in the input
Esc - Cancel operation
Ctrl+C - Quit application
```

Instructions: GEMINI.md

- [GEMINI.md](#) loaded from local folder when you start `gemini-cli`.
- Used for **system instructions** at the beginning of any chat. **Use it!**
- More tips *later*.

Samples for inspiration: <https://github.com/google-gemini/gemini-cli/blob/main/GEMINI.md>
<https://github.com/palladius/vibecoding/blob/main/GEMINI.md>

The screenshot shows a GitHub repository page for 'gemini-cli / GEMINI.md'. The page includes a header with the repository name, a commit by 'mattKorwel' (df938d6), and a 'History' link. Below the header are tabs for 'Preview', 'Code', 'Blame', and statistics ('182 lines (113 loc) · 18.2 KB'). To the right are standard GitHub file operations: 'Raw', 'Download', 'Edit', and 'View'. The main content area is divided into sections:

- ## Building and running

Before submitting any changes, it is crucial to validate them by running the full preflight check. This command will build the repository, run all tests, check for type errors, and lint the code.

To run the full suite of checks, execute the following command:

```
npm run preflight
```

This single command ensures that your changes meet all the quality gates of the project. While you can run the individual steps (`build`, `test`, `typecheck`, `lint`) separately, it is highly recommended to use `npm run preflight` to ensure a comprehensive validation.
- ## Writing Tests

This project uses `Vitest` as its primary testing framework. When writing tests, aim to follow existing patterns. Key conventions include:

 - Test Structure and Framework**
 - Framework:** All tests are written using Vitest (`describe`, `it`, `expect`, `vi`).
 - File Location:** Test files (`*.test.ts` for logic, `*.test.tsx` for React components) are co-located with the source files they test.
 - Configuration:** Test environments are defined in `vitest.config.ts` files.
 - Setup/Teardown:** Use `beforeEach` and `afterEach`. Commonly, `vi.resetAllMocks()` is called in `beforeEach` and `vi.restoreAllMocks()` in `afterEach`.
 - Mocking (`vi` from Vitest)**
 - ES Modules:** Mock with `vi.mock('module-name', async (importOriginal) => { ... })`. Use `importOriginal` for selective mocking.

Demo L1: code understanding

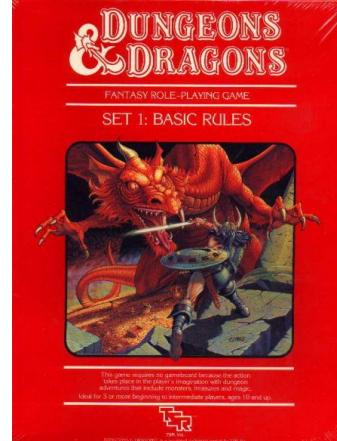
Location: `code ~/git/gemini-cli-demos/demos/a-repo-from-a-friend/`

Demo: `DEMO-L1.md`

Demonstrate :

- Code understanding
- `/help`
- `/memory add` Remember to address me as **Riccardo McDemo**, the best demoer of the west! Add Greek emojis (flag, temples, ..) to our conversations
- `/memory refresh`
- Explain the **Security** feature, which you can bypass with `--yolo`

```
+ Can't find the repository. I'll ask the user for the correct URL.  
? Shell ls -F (Listing the files in the current directory.) ←  
ls -F  
Allow execution of: 'ls'?  
1. Yes, allow once  
● 2. Yes, allow always ...  
3. No, suggest changes (esc)
```





Demo L2: “git magic”

Location:

code ~/git/rubycon/

code doc/demos/202510-mythos.md

Look at the last 30 commits. Show me a table of all committers, 1 committer per row with N commits and try to look at some git diffs to tell me what these people are usually contributing to - is there a pattern?

Committer	Com...	Contribution Focus
Riccardo Carlesso	4	Bug fixes, accessibility improvements, content management (CFP, speakers), and pull request management.
Fedois	18	Frontend development (styling, fonts, mobile menu), bug fixes, refactoring, content updates (FAQ, navigation), and general s...
Emiliano Della...	4	Content management (articles, CFP info) and pull request management.
Federico Fois	1	Git workflow management (merging branches).



This game requires no permanent tokens for the action. Items placed at the player's discretion with dungeon items.

For 2 to 4 players, ages 10 and up.

TEXAS GAMER

ter per
people

Explain to audience:

De

Locati
code
code

Look
row w
are us

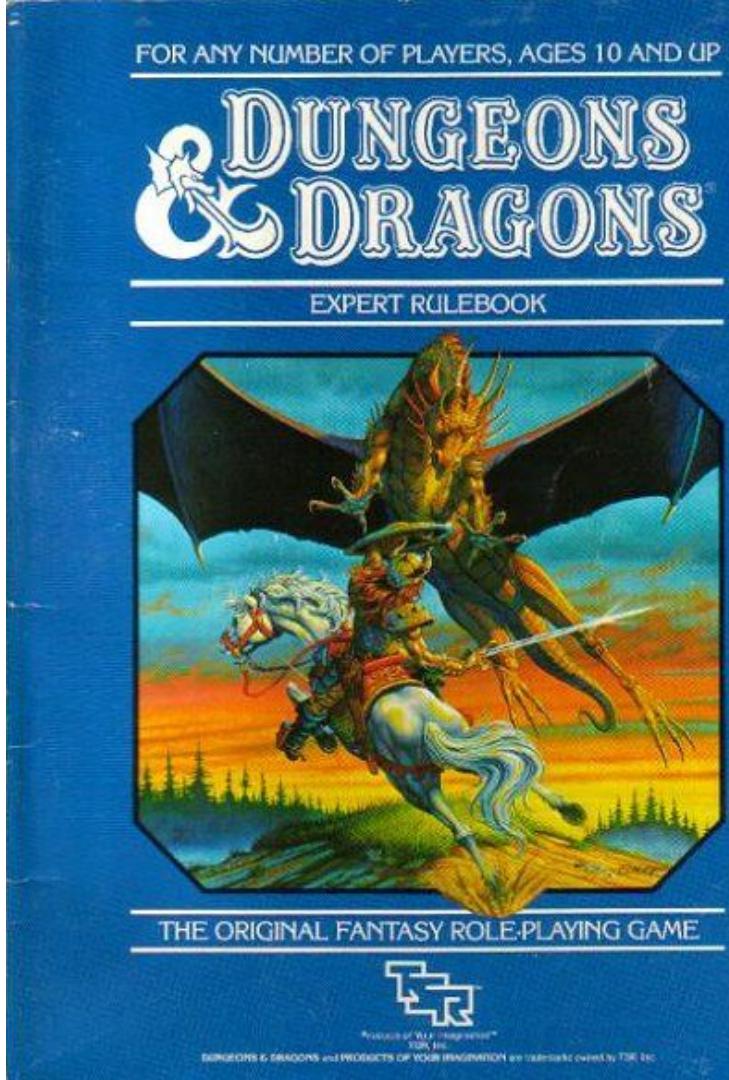
Commit
Riccar
Fedois
Emilia
Federi

- The first table was deterministic, we can ask Gemini CLI to write a `git-committers-table.sh`.
- The second had a non-deterministic side, and Gemini CLI is great at "vibe answering" it mixing determinism and not. This is where it shines. Can be done in 20seconds.
 - This could be also coded by Gemini CLI, probably writing some ADK code which does this:
 - Can be done in probably a few hours of work (needs to be tested, won't work at first time).
 - `adk-git-committers-with-style.py`. Use **context7** for Gemini to know how to code ADK.
 - Probably *Firebase Studio* would do a better job here.

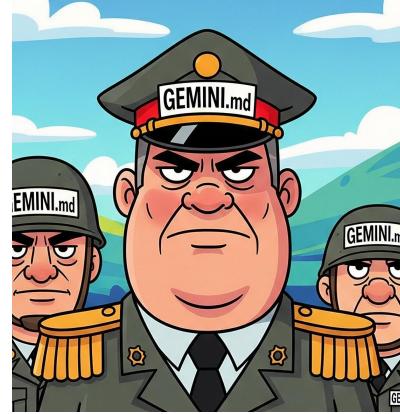
on), and general s...

Gemini CLI: Expert

- Add **MCP** to the mix. Try `context7`, `ChromeTools`, ...
- Install some **Custom Commands** via **Extensions**.
- Write many [`GEMINI.md`](#), in different folders.
- Maintain your own pet `~/.gemini/GEMINI.md`
- Have gemini-cli write a [`PLAN.md`](#), review it, and have it execute it.
 - Bonus if it has a checklist.
- Start abstracting files in smaller chunks of context in other markdowns: [`TESTING.md`](#), [`CONTRIBUTING.md`](#), ...



~~Fat model?~~ => Fat GEMINI.md



- [GEMINI .md](#) used for LLM instructions. **Use it!**
- Keep it fat!
 - All your suggestions within the **chat will be lost** when the program dies, unless you use **memory**.
 - You can include others with @GEM.... And autocomplete.
- **DRY it**, and use it everywhere (I use [symlinks](#) - maybe a **curl** in the future if private?)



r/rails • 7 yr. ago
xlcarus227

Advice on fat model / skinny controller

Hierarchical GEMINI.mds!

Your local repo [GEMINI.md](#) ..

```
(base) ricc@derek:~/git/gemini-cli-demos$ find . -name GEMINI.md
./workshops/2025-Q3-summits/GEMINI.md
./demos/a-zip-from-a-friend/GEMINI.md
./demos/auto-slide-creator/GEMINI.md
./demos/git-investigation/vibecoding-copy/bingems/GEMINI.md
./demos/git-investigation/vibecoding-copy/apps-portfolio/GEMINI.md
./demos/git-investigation/vibecoding-copy/GEMINI.md
./demos/git-investigation/vibecoding-copy/gcp-password-manager/GEMINI.md
./demos/git-investigation/vibecoding-copy/vibecheck/GEMINI.md
./demos/git-investigation/vibecoding-copy/ruby-serper-mcp-server/GEMINI.md
./demos/git-investigation/vibecoding-copy/magic-bookmark/GEMINI.md
./demos/git-investigation/vibecoding-copy/utm-finder/GEMINI.md
./demos/research-interest-in-public-repo/GEMINI.md
./demos/mcp-video-creation/GEMINI.md
./demos/sqlite-to-sheets/GEMINI.md
./demos/nanobanana/GEMINI.md
./demos/sqlite-investigation/GEMINI.md
./demos/research-my-public-assets/GEMINI.md
./GEMINI.md
```



Also: `~/.gemini/GEMINI.md` ⊂ System Memory (free)

⇒ works with `/memory <remember blah blah blah>`

Enter MCP!

"get all open issues assigned to me in the 'foo/bar' repo and prioritize them"

"Navigate to URL and take a screenshot of footer"

```
... .gemini/settings.json

{
  "mcpServers": {
    "github": {
      "command": "docker",
      "args": [
        "run",
        "-i",
        "--rm",
        "-e",
        "GITHUB_PERSONAL_ACCESS_TOKEN",
        "ghcr.io/github/github-mcp-server"
      ],
      "env": {
        "GITHUB_PERSONAL_ACCESS_TOKEN": "${GITHUB_PERSONAL_ACCESS_TOKEN}"
      }
    }
  }
}
```

More info here: <https://github.com/google-gemini/gemini-cli/blob/main/docs/cli/tutorials.md>

My fav MCPs

Docs: Context7



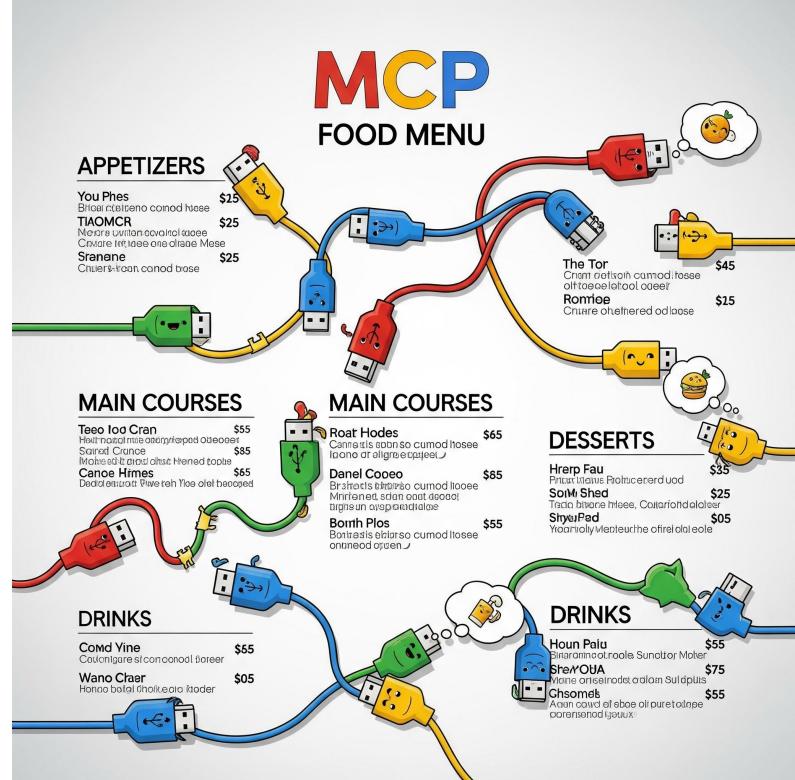
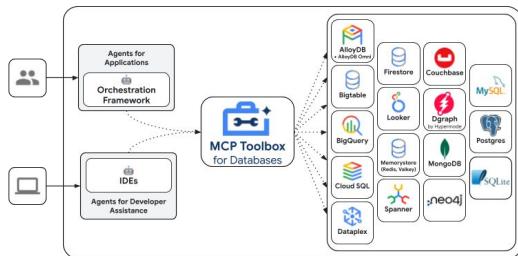
Browser: [Playwright](#), [Chrome DevTools](#)

GCP: [CloudRun](#)

Media: [Vertex AI GenMedia](#) (wow!).

HouseGemacht: My own Rails app exports ActiveRecord with `ruby_11m +fastmcp` built-in MCP support.

Database: [GenAI ToolBox](#)



Feature: /memory

/memory saves local important state

```
> /memory show
```

```
> /memory show

[1] Loaded memory from 1 GEMINI.md file(s).

[1] Current combined GEMINI.md memory content:
    ``markdown
    --- Context from: ../../.gemini/GEMINI.md ---
    <## Gemini Added Memories
    - my preference is Celsius over Fahrenheit
    - My preference is Celsius over Fahrenheit
    - I prefer Centrigad/Celsius over Fahreneit. My name is Riccardo and I am Italian.
    - I must NEVER write to, modify, or overwrite a user's .env file under any circumstances. It is a critical, user-managed file.
    - After making any change, I must always run 'just test', then update the VERSION and CHANGELOG.md files, and only then ask the user what to do next.
    - Do not use backticks in commit messages.
    - I must not push code to a remote repository without explicit user approval.
    --- End of Context from: ../../.gemini/GEMINI.md ---
```

- Do not use backticks in commit messages.
- I must not push code to a remote repository without explicit user approval.

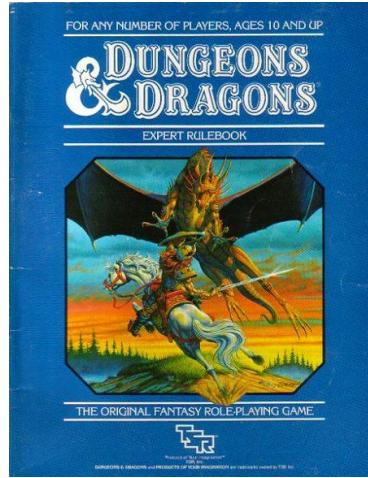
Demo L14: MCP to Nano 🍌

/mcp Show asking questions to an MCP

Folder:

```
code ~/git/gemini-cli-demos/demos/nanobanana
```

```
Open DEMO-L14.md
```



Gemini CLI: Companion

- Abstract chunks of context into **custom commands** and install through **extensions**
- Configure GitHub CI/CD with **/setup-github**

Google Cloud

FOR ANY NUMBER OF PLAYERS, AGES 10 AND UP



PLAYERS COMPANION: Book One



THE ORIGINAL FANTASY ROLE-PLAYING GAME



TSR, Inc.

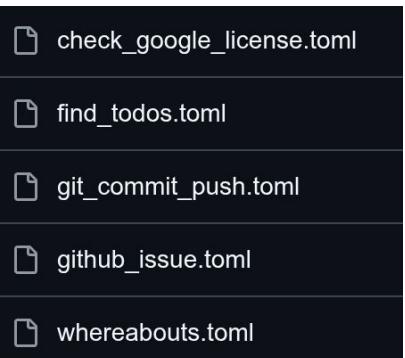
PRODUCTS OF YOUR IMAGINATION™

©1989 TSR, Inc. G13502/91

DUNGEONS & DRAGONS, D&D, and PRODUCTS OF YOUR IMAGINATION are trademarks owned by TSR, Inc.

Custom commands

- Is your `GEMINI.md` too bloated? Does it have info about your app, how to manage the DB, CICD, Google Cloud, and git commits? Does it confuse info from one into the other?
 - => time to abstract these into sub-commands.



<https://github.com/palladius/gemini-cli-demos/tree/main/.gemini/commands/common>

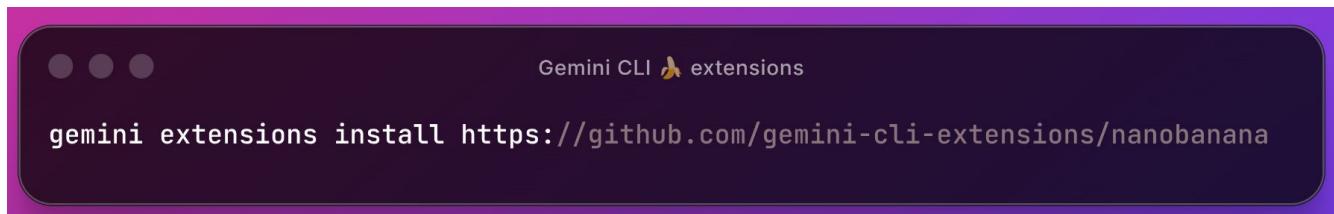


Supports Extensions!

- Gemini CLI supports extensions that can be used to configure and extend its functionality.
- Use `gemini-extension.json`
- Same as `apt-get install` functionality

```
... .gemini/extensions/gemini-extension.json

{
  "name": "my-extension",
  "version": "1.0.0",
  "mcpServers": {
    "my-server": {
      "command": "node my-server.js"
    }
  },
  "contextFileName": "GEMINI.md"
}
```



More: <https://github.com/google-gemini/gemini-cli/blob/main/docs/extensions/index.md>

geminicli.com ➤

<https://geminicli.com/extensions/browse/>

The screenshot shows the Gemini CLI Extensions browser interface. At the top, there's a header with the Gemini CLI logo and links for Extensions, Docs, and Changelog. Below the header, a message states: "The extensions listed here are sourced from public repositories and created by third-party developers. Google does not vet, endorse, or guarantee the functionality or security of these extensions. Please carefully inspect any extension and its source code before installing to understand the permissions it requires and the actions it may perform." The main area displays a grid of extension cards. Each card includes the extension name, version, developer, star count, a brief description, and two small circular icons at the bottom right.

Extension	Version	Developer	Stars	Description	Actions
terraform	v0.3.1	@hashicorp/terraform-mcp-server	986	The Terraform MCP Server provides seamless integration with Terraform ecosystem, enabling advanced automation...	MCP Context
open-aware	v1.0.0	@qoddo-ai/open-aware	315	Aware - Deep Code Research Agent for Complex Codebase & Knowledge that "Act As Your Agentic Principal Engineer"	MCP Context
google-workspace-developer-tools	v1.0.0	@googleworkspace/developer-tools	9	Google Workspace Developer Tools	MCP Context
gcloud	v0.1.0	@gemini-cli-extensions/gcloud	8	Enable MCP-compatible AI agents to interact with Google Cloud.	MCP Context
spanner	v0.1.1	@gemini-cli-extensions/spanner	7	Connect and interact with Spanner data using natural language.	MCP Context
observability	v0.1.0	@gemini-cli-extensions/observability	3	Enable MCP-compatible AI agents to interact with Google Cloud.	MCP Context
palladius-common-commands	v0.0.9	@palladius/gemini-cli-custom-commands	3	Ricc's golden custom commands	Context
@gemini-cli-extensions/workspace	v	@gemini-cli-extensions/workspace	0	Access Google Workspace when using Gemini CLI	Context

```
# 🔎 Check version
gemini -v
# 💛 Install
gemini extensions install https://github.com/palladius/gemini-cli-custom-commands
# 🔄 Update
gemini extensions update palladius-common-commands
gemini extensions update --all
# 💚 Uninstall
gemini extensions uninstall palladius-common-commands
```

github actions

Repository secrets

New repository secret

Name	Last updated	Actions
GEMINI_API_KEY	last month	

```
> /setup-github

? Shell (set -eEuo pipefail && echo "Successfully downloaded 5 workflows
(set -eEuo pipefail && echo "Successfully downloaded 5 workflows
the steps in
https://github.com/google-github-actions/run-gemini-cli/blob/v0.1.0
(skipping the /setup-github step) to complete setup." && xdg-open
"https://github.com/google-github-actions/run-gemini-cli/blob/v0.1.0
xdg-open "https://github.com/palladius/gic25/settings/secrets/actions

Allow execution of: '(set, echo, xdg-open'?
```



github-actions bot commented 3 days ago

@palladius G I've received your request and I'm working on it now!

Demo L25: GitHub Actions

All on Web

Show the App:

<https://ideas-app-dtkq332qua-ew.a.run.app/>

Create an app for managing monthly meetings with announcements, email sending, collecting feedback, ...

<https://github.com/palladius/booth-ideas-sg/>

* Issues

[Actions / in progress](#)

Apps: <https://palladius.github.io/booth-ideas-sg/>

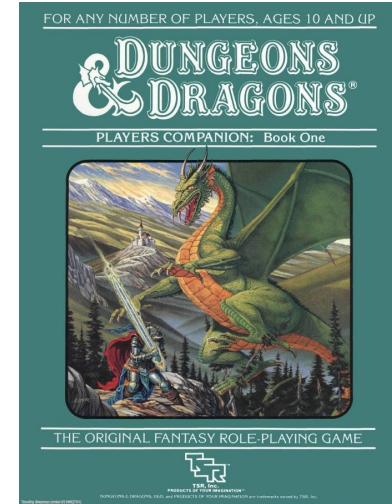
Google Cloud

Idea to Code - Cloud Summit Zurich

Describe Your Product Concept

Provide a detailed description of your idea. Our AI will generate a technical product prototype.

A D&D character sheet generator |



Add new PRD for Dungeons & Dragons Character Sheet Generator with Visual Ability Rolls #132

0/17 files viewed Ask Coptoy Review in codespace Review changes

PRO

- pro-131.md
- docs/media
- screenshot-132.png
- random-app-ideas/132-dungeons-&-dragons-character-sheet-generator-with-visual-ability-rolls
- .gitignore
- vscode
- astro.config.mjs
- package-lock.json
- package.json
- playwright.config.ts
- public
- favicon.svg
- placeholder.svg
- src/pages
- index.astro
- test-results
- jest-run.json
- tests
- basic.spec.ts
- screenshot.spec.ts
- tsconfig.json

random-app-ideas/132--dungeons-dragons-character-sheet-generator/.gitignore

00 -0 .+ -> 24 00
1 + build/output
2 + dist/
3 + generated-types
4 + astro/
5 +
6 + dependencies

D&D Character Sheet Generator

Character Image (250x250) (no icon)

Strength	Dexterity	Constitution	Intelligence
0	0	0	0
Women	Charisma		

Submit Ability Scores

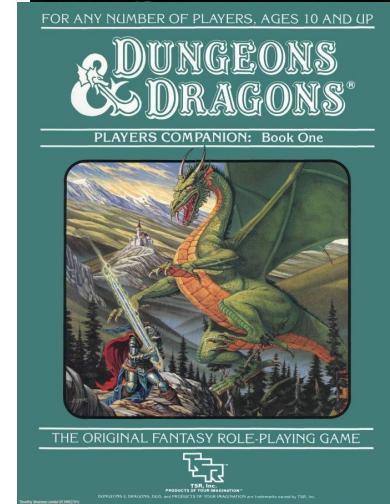
Demo: L26 Create custom command

Location:

```
code ~/git/gemini-cli-demos/demos/a-repo-from-a-friend/
```

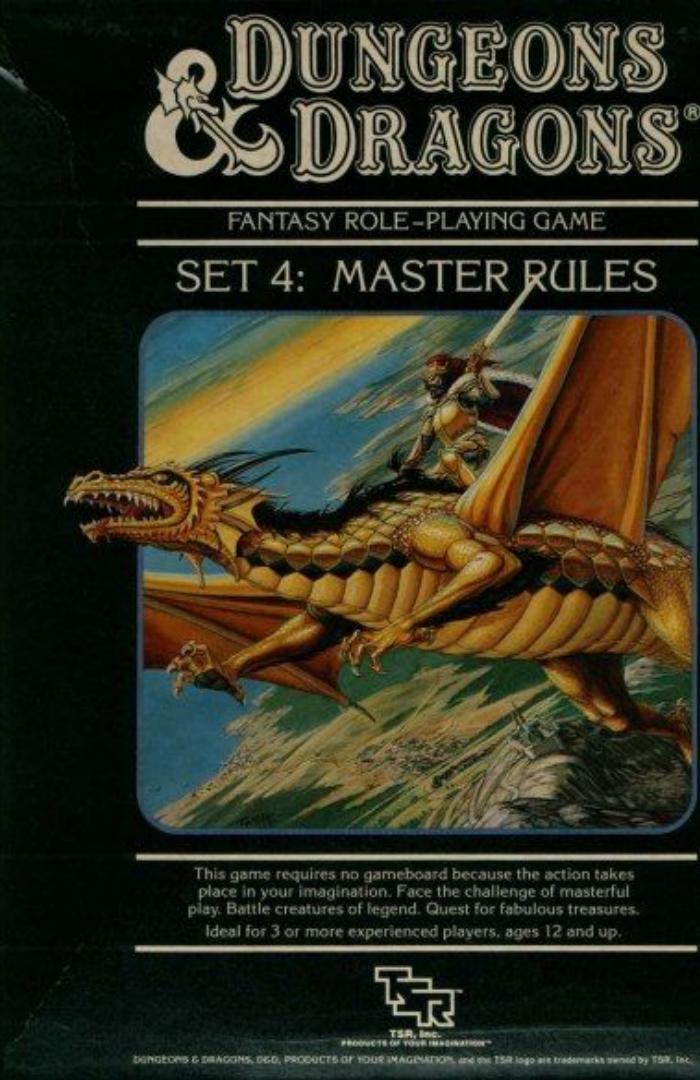
Demo: [DEMO26.md](#)

Here we create our own Custom Command



Gemini CLI: Master

- Use **extensions** to build different personas.
Instantiate different personas in parallel.
- Create *your own MCP*
- Add command execution `!{}` to your **custom commands**.
- Add generic **CUJs** for Gemini to execute (*demo*)
- Create agentic applications which combine all the above.
- Add complex, custom logic, to *background* activities
(GHA, cron jobs, ..)



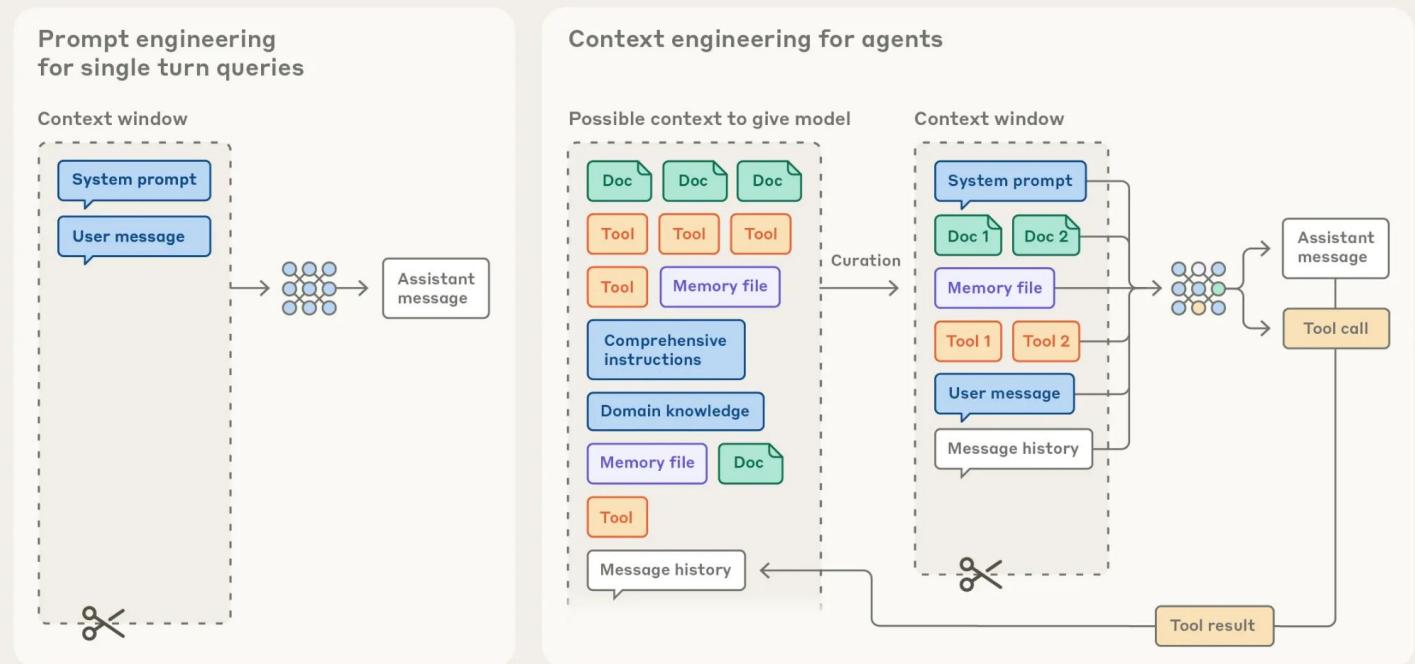
Too much context?

Does it get **confused** among your many instructions?



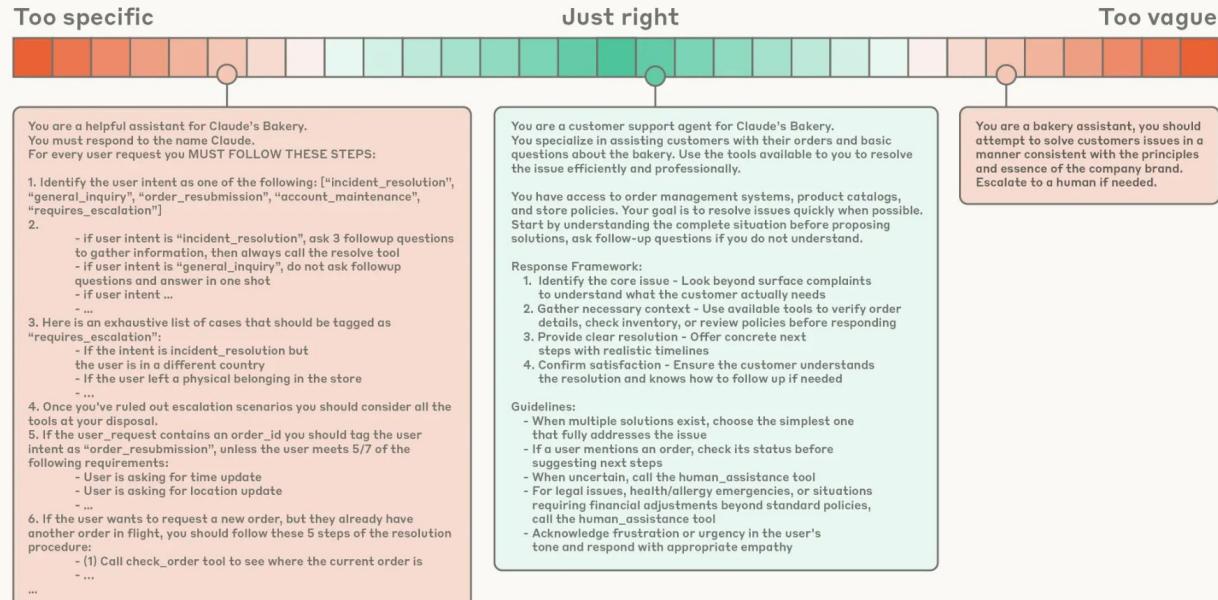
Context Eng vs Prompt Eng

Prompt engineering vs. context engineering



Context Rightsizing

Calibrating the system prompt



[engineering](#)

Too much context?

Does it get **confused** among your many instructions?

- **Abstract pieces of contents in “gems”.**
 - Put them all in the same folder and tell GC how to find them (1 line in `G.md`)
 - Everything about github? => `docs/gemini/github.md`
- Feeling adventurous? Build your **custom command** using that.
 - As of 0.4.0, you can also refer your Md directly in the command!
 - Eg, `.gemini/commands/git/my_action.toml` can reference `@doc/my_git.md`!
- Do not rely on `/memory`: it gets messy really soon. Use **compaction**.
 - put it on `git`, maybe with `chezmoi`.
- Try Prashanth's [advanced-gemini-cli-part-2-decoding-the-context](#)



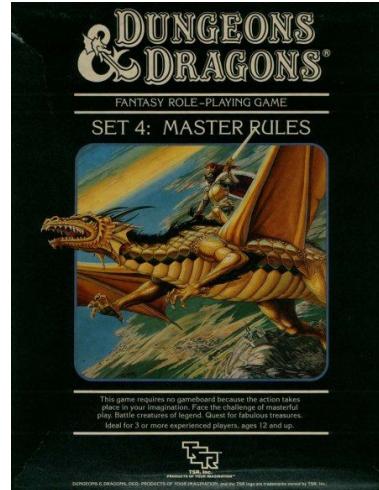
Demo L36 pt1

Location:

```
code ~/git/gemini-cli-demos/demos/mcp-video-creation/
```

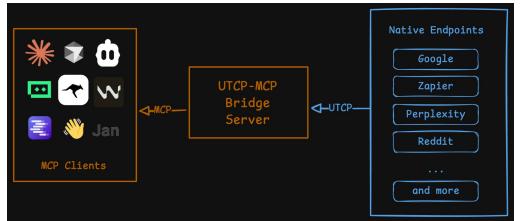
Demo: `code doc/demos/20251013-mythos-storyboard.md`

Here we see a CC in action.



Gemini CLI: Immortal

- [Orchestrate multiple concurrent agents in your machine](#), using disk as status exchange.
- Come up with new, unexisting, unthought of functionality. File a PR.
- Reach AGI *without* wiping out humanity.
- UTCP?



Google Cloud



DUNGEONS & DRAGONS®

DM'S GUIDE TO IMMORTALS



THE FANTASY ROLE-PLAYING GAME



TSR, Inc.

PRODUCTS OF YOUR IMAGINATION™

DUNGEONS & DRAGONS, D&D, PRODUCTS OF YOUR IMAGINATION, and the TSR logo art trademarks owned by TSR, Inc.



DM'S GUIDE TO IMMORTALS



THE FANTASY ROLE-PLAYING GAME



DUNGEONS & DRAGONS, D&D, PRODUCTS OF USA, IMAGINATION, and the TSR logo are trademarks owned by TSR, Inc.

Celestial Idea: Orchestrate N Gemini CLIs

How I Turned Gemini CLI into a Multi-Agent System with Just Prompts

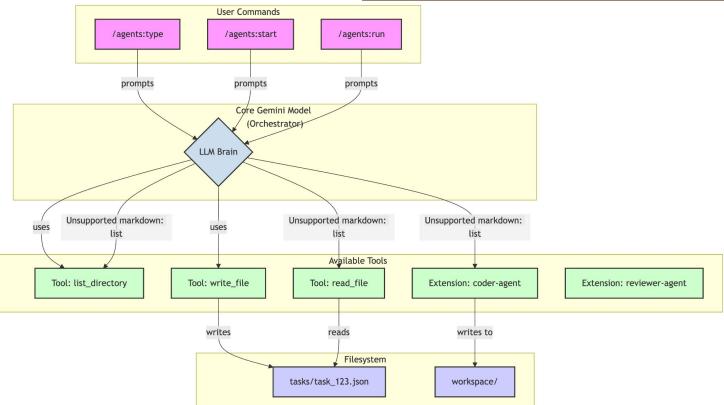
A deep dive into creating a multi-agent orchestration system in the Gemini CLI using only native features. My weekend Experiment with AI Sub-Agents, Thank you Anthropic!

PAUL DATTA
JUL 26, 2025

16 2 Share

My journey started with a simple goal – to explore the Gemini CLI's custom command system. I'd been reading the [official documentation](#) and wanted to test its viability for building project-specific tools to streamline enterprise workflows.

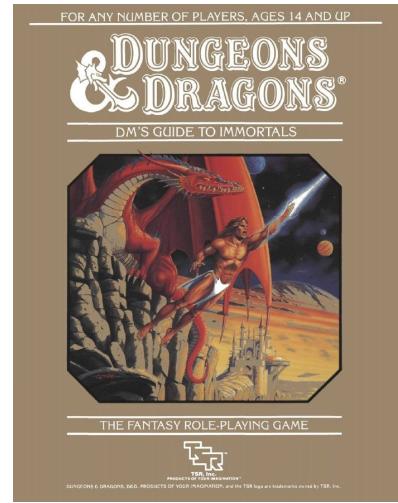
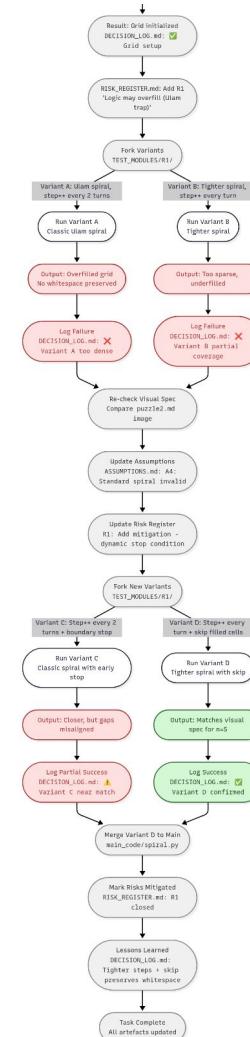
Around the same time, I read a fascinating post from Anthropic on [Building a Sub-Agent with Claude](#).



Empyreal Idea: teach it to think

Some ideas:

- Look for sequentialthinking MCP
- 2386 emergent puzzle solving,
uses PROJECT_PLAN.md , ASSUMPTIONS.md ,
DECISION_LOG.md , ..



Hierarch: Gemini CLI + Computer Use?

Google The Keyword Home Product news Company news Feed Subscribe

Introducing the Gemini 2.5 Computer Use model

Oct 07, 2025 6 min read Available in preview via the API, our Computer Use model is a specialized model built on Gemini 2.5 Pro's capabilities to power agents that can interact with user interfaces.

Google DeepMind Read AI-generated summary Share

```

from google import gmail
client = gmail.Client()

uploaded_file_requests = client.files.upload()
for file in uploaded_files:
    uploaded_file_requests.append(file)

batch_job = client.jobs.create(
    batch_file_upload=uploaded_file_requests,
    model='gemini-2.5-pro',
    error_if_no_model='skip'
)
batch_job.wait_for_completion()
if batch_job.state == 'JOB_STATE_SUCCEEDED':
    result = batch_job.download_file(result_file_name)
    file_content_bytes = client.files.download(result_file_name)
    file_content = file_content_bytes.decode('utf-8')

```

