

Chương 4: Hướng dẫn sử dụng công cụ dòng lệnh (CLI Reference)

- [Trang 4.1: Lệnh khởi tạo và quản lý dự án \(ag-kit init, ag-kit status\).](#)
- [Trang 4.2: Cập nhật phiên bản và quản lý dữ liệu \(ag-kit update\).](#)
- [Trang 4.3: Các tham số nâng cao \(Sử dụng --force, --path, --quiet\).](#)

Trang 4.1: Lệnh khởi tạo và quản lý dự án (ag-kit init, ag-kit status).

1. Tổng quan về công cụ CLI `ag-kit`

Công cụ dòng lệnh `ag-kit` là giao diện chính để quản lý vòng đời của Antigravity Kit trong các dự án. Nó cung cấp 3 lệnh chính:

Lệnh	Chức năng
<code>ag-kit init</code>	Khởi tạo (cài đặt) bộ kit vào dự án
<code>ag-kit update</code>	Cập nhật bộ kit lên phiên bản mới nhất
<code>ag-kit status</code>	Kiểm tra trạng thái cài đặt hiện tại

Trang này tập trung vào `ag-kit init` và `ag-kit status`.

2. Lệnh `ag-kit init` — Khởi tạo bộ kit

Mục đích

Lệnh này cài đặt toàn bộ thư mục `.agent/` vào dự án của bạn, bao gồm tất cả Agents, Skills, Workflows, quy tắc và scripts.

Cú pháp

```
bash
ag-kit init [options]
```

Hành vi mặc định (không có options)

Khi chạy `ag-kit init` mà không có tham số bổ sung:

1. Kết nối với GitHub để tải phiên bản mới nhất từ nhánh main.
2. Tạo thư mục `.agent/` trong thư mục **hiện tại** (nơi bạn đang đứng trong terminal).
3. Sao chép toàn bộ 20 Agents, 36 Skills, 11 Workflows cùng các file cấu hình.
4. Thông báo kết quả: thành công hay thất bại.

Ví dụ sử dụng cơ bản

```
bash
# Di chuyển vào thư mục dự án
cd /đường/dẫn/đến/dự-án

# Khởi tạo
ag-kit init
```

Điều kiện thành công

- Kết nối internet hoạt động.
- Quyền ghi vào thư mục hiện tại.
- Thư mục `.agent/` chưa tồn tại (nếu đã tồn tại cần thêm `--force`).

3. Kịch bản khởi tạo phổ biến

Kịch bản 1: Dự án mới hoàn toàn

```
bash
# Tạo thư mục dự án
mkdir my-project
cd my-project
```

```
# Khởi tạo dự án (ví dụ Next.js)  
npx create-next-app@latest .
```

```
# Cài đặt Antigravity Kit  
ag-kit init
```

Kịch bản 2: Thêm vào dự án đang có sẵn

```
bash  
# Di chuyển vào dự án hiện tại  
cd /projects/existing-project  
  
# Cài đặt Kit (sẽ không ảnh hưởng code hiện tại)  
ag-kit init
```

Kịch bản 3: Cài đặt vào thư mục con cụ thể

```
bash  
ag-kit init --path ./packages/my-app
```

```
bash  
ag-kit init --path ./packages/my-app
```

Kịch bản 4: Khởi tạo lại sau khi thử nghiệm

```
bash  
# Ghi đè lên .agent/ cũ  
ag-kit init --force
```

4. Cấu trúc kết quả sau `ag-kit init`

```
dự-án-của-bạn/  
├─ src/ (code dự án của bạn - không bị ảnh hưởng)  
├─ package.json (không bị ảnh hưởng)  
├─ .agent/ ← CHỈ THỂ MỤC NÀY ĐƯỢC TẠO MỚI  
├─ ARCHITECTURE.md  
├─ AGENT_FLOW.md  
├─ mcp_config.json  
├─ agents/ (20 Agent files)  
├─ skills/ (36 Skill directories)  
├─ workflows/ (11 Workflow files)  
├─ rules/  
│ └─ GEMINI.md  
├─ scripts/  
├─ checklist.py  
├─ verify_all.py  
├─ auto_preview.py  
└─ session_manager.py
```

“**Đảm bảo an toàn:** Lệnh `init` chỉ tạo thư mục `.agent/`. Nó **không** sửa đổi, không xóa, và không ảnh hưởng đến bất kỳ file code nào trong dự án của bạn.

5. Lệnh `ag-kit status` — Kiểm tra trạng thái cài đặt

Mục đích

Kiểm tra xem bộ kit đã được cài đặt đúng cách chưa và hiển thị thông tin về phiên bản đang dùng.

Cú pháp

```
bash
ag-kit status
```

Thông tin hiển thị

Lệnh này sẽ kiểm tra và báo cáo:

Mục kiểm tra	Trạng thái tốt	Trạng thái lỗi
Thư mục <code>.agent/</code>	<input type="checkbox"/> Tồn tại	<input type="checkbox"/> Không tìm thấy
File ARCHITECTURE.md	<input type="checkbox"/> Tìm thấy	<input type="checkbox"/> Thiếu
Thư mục <code>agents/</code>	<input type="checkbox"/> Đủ 20 Agent	<input type="triangle-up"/> Thiếu file
Thư mục <code>skills/</code>	<input type="checkbox"/> Đủ 36 Skill	<input type="triangle-up"/> Thiếu thư mục
Thư mục <code>workflows/</code>	<input type="checkbox"/> Đủ 11 Workflow	<input type="triangle-up"/> Thiếu file
Phiên bản cài đặt	<input type="checkbox"/> v2.0.x	<input type="info"/> Có bản mới hơn

Ví dụ kết quả

```
Antigravity Kit Status
=====
 Installation: Found at ./agent/
 Version: 2.0.2
 Agents: 20 found
 Skills: 36 found
 Workflows: 11 found
 Latest version: 2.0.2 (up to date)
```

6. Quy trình xác minh sau khi cài đặt

Sau khi chạy `ag-kit init`, thực hiện các bước sau để xác nhận mọi thứ hoạt động:

Bước 1: Kiểm tra thư mục đã được tạo

```
bash
ls .agent/ # macOS / Linux
dir .agent\ # Windows
```

Bước 2: Kiểm tra trạng thái qua CLI

```
bash
ag-kit status
```

Bước 3: Thử mở AI Chat trong IDE và gõ thử một lệnh

```
/status
```

Nếu AI nhận ra lệnh và hiển thị thông tin dự án, cài đặt đã thành công hoàn toàn.

7. Sự cố thường gặp với lệnh `init`

Lỗi	Nguyên nhân	Giải pháp
<code>ag-kit: command not found</code>	Chưa cài toàn cục hoặc PATH chưa đúng	Chạy <code>npm install -g @vudovn/ag-kit</code>
<code>.agent/ already exists</code>	Đã cài đặt trước đó	Thêm <code>--force</code> để ghi đè
<code>Permission denied</code>	Không có quyền ghi	Chạy terminal với quyền admin hoặc đổi owner thư mục
<code>Network error</code>	Không có kết nối internet	Kiểm tra mạng và thử lại
<code>Repository not found</code>	Tên package sai	Đảm bảo dùng đúng <code>@vudovn/ag-kit</code>

Trang 4.2: Cập nhật phiên bản và quản lý dữ liệu (ag-kit update).

1. Lệnh `ag-kit update` — Cập nhật bộ kit

Mục đích

Cập nhật toàn bộ nội dung thư mục `.agent/` lên phiên bản mới nhất từ repository chính thức.

Cú pháp

```
bash
ag-kit update [options]
```

2. Điều gì xảy ra khi chạy `ag-kit update`?

Lệnh này thực hiện các bước theo thứ tự:

1. Kết nối đến GitHub để lấy phiên bản mới nhất

|



2. So sánh phiên bản hiện tại với phiên bản mới nhất

|

▼

3. ⚠ XÓA toàn bộ thư mục `.agent/` hiện tại

|

▼

4. Tải về và giải nén phiên bản mới

|

▼

5. Tạo lại `.agent/` với nội dung mới

|

▼

6. Báo cáo kết quả cập nhật

⚠ **Cảnh báo quan trọng:** Bước 3 là **xóa hoàn toàn** thư mục `.agent/`. Nếu bạn đã tùy chỉnh Agent, Skill hoặc Workflow trong thư mục này, **tất cả sẽ bị mất** nếu không sao lưu trước.

3. Chiến lược sao lưu trước khi cập nhật

Phương án 1: Sao lưu thủ công (Đơn giản nhất)

```
bash
# Tạo bản sao thư mục .agent/
cp -r .agent/ .agent-backup/ # macOS / Linux
xcopy .agent .agent-backup /E /I # Windows

# Thực hiện cập nhật
ag-kit update
```

Sau khi cập nhật, merge lại các tùy chỉnh của bạn thủ công

Phương án 2: Dùng Git (Khuyến nghị cho dự án nhóm)

Nếu thư mục `.agent/` đã được commit vào Git:

```
bash
# Commit trạng thái hiện tại để lưu lịch sử
git add .agent/
git commit -m "backup: save custom .agent configs before update"

# Cập nhật Kit
ag-kit update

# Nếu cần khôi phục tùy chỉnh, dùng git diff để xem sự khác biệt
git diff HEAD~1 .agent/
```

Phương án 3: Tách biệt tùy chỉnh (Bền vững nhất)

Tạo các file tùy chỉnh riêng biệt thay vì sửa file gốc:

- Thêm Agent tùy chỉnh: `.agent/agents/my-team-agent.md` (file mới, không ghi đè file gốc)
- Thêm Workflow tùy chỉnh: `.agent/workflows/my-deploy.md` (file mới)
- **Không sửa** các file Agent/Skill gốc (ví dụ không sửa `frontend-specialist.md`)

Với phương án này, sau khi `update`, các file gốc được cập nhật, còn các file tùy chỉnh của bạn vẫn giữ nguyên.

4. Khi nào nên cập nhật?

Tình huống	Hành động
------------	-----------

Phiên bản mới có tính năng bạn cần	☐ Nên cập nhật
Có bản vá lỗi (bugfix) quan trọng	☐ Nên cập nhật
Cần kiểm tra tương thích trước	☐ Đọc CHANGELOG trước, rồi cập nhật
Đang giữa dự án phức tạp	⚠️ Cân nhắc — cập nhật sau khi hoàn thành sprint
Đã tùy chỉnh nhiều file gốc	⚠️ Sao lưu kỹ trước khi cập nhật

5. Kiểm tra CHANGELOG trước khi cập nhật

Trước khi chạy lệnh update, hãy đọc file CHANGELOG.md để biết những thay đổi trong phiên bản mới:

```
bash
# Xem CHANGELOG trên GitHub
# https://github.com/vudovn/antigravity-kit/blob/main/CHANGELOG.md
```

Mỗi phiên bản mới thường ghi rõ:

- **Added:** Thứ gì mới được thêm vào.
- **Changed:** Thứ gì thay đổi (có thể ảnh hưởng đến tùy chỉnh của bạn).
- **Removed:** Thứ gì bị xóa (cẩn thận nếu bạn đang phụ thuộc vào nó).
- **Fixed:** Lỗi nào đã được vá.

6. Cập nhật từ một nhánh cụ thể

Nếu bạn muốn thử nghiệm phiên bản chưa phát hành chính thức:

```
bash
# Cập nhật từ nhánh 'dev' (thử nghiệm)
ag-kit update --branch dev

# Quay về nhánh chính thức
ag-kit update --branch main
```

Lưu ý: Chỉ dùng nhánh `dev` cho mục đích thử nghiệm, không dùng trên môi trường production.

7. Rollback (Hoàn tác cập nhật)

Antigravity Kit không có lệnh rollback tích hợp. Để hoàn tác:

Nếu đã sao lưu thủ công:

```
bash
# Xóa .agent/ mới
rm -rf .agent/

# Khôi phục bản cũ
cp -r .agent-backup/ .agent/
```

Nếu đang dùng Git:

```
bash
# Khôi phục về commit trước khi update
git checkout HEAD~1 -- .agent/
```

Nếu muốn cài lại một phiên bản cụ thể: Hiện tại chưa có cú pháp chọn phiên bản cụ thể trong CLI. Cách làm thủ công là:

- Clone repository về: `git clone https://github.com/vudovn/antigravity-kit.git`
- Checkout tag phiên bản mong muốn: `git checkout v2.0.1`
- Sao chép thư mục `.agent/` vào dự án của bạn.

8. Cập nhật trong môi trường CI/CD

Nếu bộ kit là một phần trong quy trình CI/CD của đội:

```
yaml
# Ví dụ trong GitHub Actions
```

```
- name: Install/Update Antigravity Kit
```

```
run: ag-kit init --quiet --force
```

```
# (hoặc ag-kit update --quiet)
```

Tham số `--quiet` giúp ẩn bớt output không cần thiết trong log CI/CD.

Trang 4.3: Các tham số nâng cao (Sử dụng `--force`, `--path`, `--quiet`).

1. Tổng quan các tham số

Công cụ `ag-kit` hỗ trợ nhiều tham số (options/flags) để tinh chỉnh hành vi cho từng tình huống cụ thể.

Tham số	Lệnh áp dụng	Chức năng tóm tắt
<code>--force</code>	<code>init</code>	Ghi đè thư mục <code>.agent/</code> đã tồn tại
<code>--path</code>	<code>init</code>	Chỉ định thư mục cài đặt khác thay vì thư mục hiện tại
<code>--branch</code>	<code>init</code> , <code>update</code>	Chọn nhánh Git cụ thể để tải về
<code>--quiet</code>	<code>init</code> , <code>update</code>	Chạy im lặng, ẩn bớt output
<code>--dry-run</code>	<code>init</code> , <code>update</code>	Xem trước các thao tác sẽ thực hiện mà không thực sự làm gì

2. Tham số `--force`

Chức năng

Cho phép ghi đè lên thư mục `.agent/` đã tồn tại. Mặc định, `ag-kit init` sẽ báo lỗi nếu thư mục này đã có.

Cú pháp

```
bash
ag-kit init --force
```

Khi nào dùng

Tình huống	Dùng <code>--force</code> không?
Cài đặt lần đầu	<input type="checkbox"/> Không cần
Thư mục <code>.agent/</code> đã tồn tại, muốn cài lại	<input type="checkbox"/> Cần thiết
Muốn xóa mọi tùy chỉnh và trở về mặc định	<input type="checkbox"/> Cần thiết
Cần reset Kit sau khi thử nghiệm hỏng	<input type="checkbox"/> Cần thiết

Ví dụ

```
bash
# Thử cài đặt thông thường (sẽ báo lỗi nếu đã có .agent/)
ag-kit init
# Error: .agent/ already exists. Use --force to overwrite.

# Cài đặt lại với --force
ag-kit init --force
#  Overwriting .agent/... Done!
```

⚠ **Cảnh báo:** `--force` sẽ **xóa toàn bộ** thư mục `.agent/` hiện tại trước khi cài lại. Mọi tùy chỉnh trong đó sẽ bị mất. Hãy sao lưu trước nếu cần.

3. Tham số `--path`

Chức năng

Chỉ định thư mục đích để cài đặt Kit thay vì thư mục hiện tại trong terminal.

Cú pháp

```
bash
ag-kit init --path <đường-dẫn-đến-thư-mục>
```

Khi nào dùng

Tình huống	Ví dụ
Cài vào thư mục con cụ thể	<code>ag-kit init --path ./apps/frontend</code>
Cài vào monorepo package	<code>ag-kit init --path ./packages/web</code>
Cài vào đường dẫn tuyệt đối	<code>ag-kit init --path /home/user/projects/myapp</code>
Không muốn phải <code>cd</code> vào thư mục	<code>ag-kit init --path ./client</code>

Ví dụ thực tế

```
bash
# Cấu trúc monorepo:
my-monorepo/
├─ apps/
│  └─ web/ ← muốn cài vào đây
│  └─ mobile/ ← và cả đây
└─ packages/

# Cài vào web app mà không cần cd
ag-kit init --path ./apps/web

# Cài vào mobile app
ag-kit init --path ./apps/mobile

# Kết hợp với --force nếu đã có .agent/ cũ
ag-kit init --path ./apps/web --force
```

4. Tham số `--branch`

Chức năng

Chỉ định nhánh Git cụ thể trên repository để tải bộ kit. Mặc định sử dụng nhánh main (phiên bản ổn định nhất).

Cú pháp

```
bash
ag-kit init --branch <tên-nhánh>
ag-kit update --branch <tên-nhánh>
```

Các nhánh phổ biến

Nhánh	Mô tả	Nên dùng
main	Phiên bản ổn định chính thức	<input type="checkbox"/> Production, dự án thực tế
<code>dev</code>	Phiên bản phát triển, có thể không ổn định	<input checked="" type="checkbox"/> Thử nghiệm tính năng mới

Ví dụ

```
bash
# Cài từ nhánh chính thức (mặc định)
ag-kit init
# Tương đương với:
ag-kit init --branch main

# Thử nghiệm tính năng mới nhất (chưa phát hành)
ag-kit init --branch dev

# Cập nhật từ nhánh dev
```

```
ag-kit update --branch dev
```

Lưu ý: Nhánh `dev` có thể chứa các tính năng chưa hoàn thiện. Chỉ dùng cho môi trường phát triển và thử nghiệm cá nhân.

5. Tham số `--quiet`

Chức năng

Chạy lệnh ở chế độ im lặng — ẩn bớt các thông báo thông thường, chỉ hiển thị lỗi nghiêm trọng (nếu có).

Cú pháp

```
bash
ag-kit init --quiet
ag-kit update --quiet
```

Khi nào dùng

- **CI/CD pipelines:** Khi bạn không muốn log quá dài làm rối output của build pipeline.
- **Script tự động:** Khi chạy trong script shell không cần đọc log chi tiết.
- **Chạy hàng loạt:** Khi cần cài/cập nhật Kit cho nhiều dự án cùng lúc.

Ví dụ trong GitHub Actions

```
yaml
name: Setup Antigravity Kit
on: [push]

jobs:
  setup:
    runs-on: ubuntu-latest
```

steps:

```
- uses: actions/checkout@v3
```

```
- name: Install Node.js
```

```
uses: actions/setup-node@v3
```

with:

```
node-version: '20'
```

```
- name: Install Antigravity Kit
```

```
run: |
```

```
npm install -g @vudovn/ag-kit
```

```
ag-kit init --quiet --force
```

6. Tham số `--dry-run`

Chức năng

Thực hiện một lần chạy "giả định" — hiển thị tất cả những gì sẽ xảy ra nhưng **không thực sự làm gì** trên hệ thống. Không tạo, không xóa, không thay đổi file nào.

Cú pháp

```
bash
ag-kit init --dry-run
ag-kit update --dry-run
```

Khi nào dùng

- Trước khi cài đặt trên dự án quan trọng để xem trước thao tác.
- Kiểm tra xem lệnh có hoạt động đúng không mà không có rủi ro.
- Xem danh sách file sẽ được tạo ra.

Ví dụ kết quả của `--dry-run`

```
bash
ag-kit init --dry-run
```

[DRY RUN] The following actions would be performed:

- Create directory: ./agent/
- Create file: ./agent/ARCHITECTURE.md
- Create file: ./agent/AGENT_FLOW.md
- Create directory: ./agent/agents/ (20 files)
- Create directory: ./agent/skills/ (36 directories)
- Create directory: ./agent/workflows/ (11 files)
- Create directory: ./agent/rules/
- Create directory: ./agent/scripts/

[DRY RUN] No changes were made.

7. Kết hợp nhiều tham số

Các tham số có thể kết hợp với nhau tùy ý:

```
bash
# Cài vào thư mục cụ thể, ghi đè nếu tồn tại, chạy im lặng
ag-kit init --path ./apps/web --force --quiet

# Xem trước thao tác cập nhật từ nhánh dev
ag-kit update --branch dev --dry-run

# Cài đặt vào thư mục cụ thể, xem trước (không thực sự làm)
ag-kit init --path ./packages/shared --dry-run
```

8. Bảng tham chiếu nhanh (Quick Reference)

```
bash
```

```
# Cài đặt cơ bản
```

```
ag-kit init
```

```
# Cài đặt (ghi đè nếu đã có)
```

```
ag-kit init --force
```

```
# Cài vào thư mục khác
```

```
ag-kit init --path ./my-app
```

```
# Cài từ nhánh cụ thể
```

```
ag-kit init --branch dev
```

```
# Chạy im lặng (cho CI/CD)
```

```
ag-kit init --quiet --force
```

```
# Xem trước không làm gì
```

```
ag-kit init --dry-run
```

```
# Kết hợp: cài vào thư mục khác, ghi đè, im lặng
```

```
ag-kit init --path ./apps/web --force --quiet
```

```
# Cập nhật và xem trước
```

```
ag-kit update --dry-run
```

```
# Cập nhật im lặng (cho automation)
```

```
ag-kit update --quiet
```

```
# Kiểm tra trạng thái
```

```
ag-kit status
```