

Phát Triển Với AI

TỪ Ý TƯỞNG ĐẾN SẢN PHẨM

Hướng dẫn toàn diện tất-cả-trong-một để xây dựng website, ứng dụng hoặc startup của riêng bạn từ đầu. Làm chủ các AI agent và công cụ, tích hợp thanh toán, triển khai máy chủ, và học các chiến thuật marketing để tăng trưởng tự nhiên.

apifreellm.com

Phiên bản 1.0 — Tháng 2, 2026

Mục Lục

1. Giới Thiệu

Thế Giới Đã Thay Đổi

Tại Sao Khóa Học Này Tồn Tại

2. Bộ Công Cụ Phát Triển Ưu Tiên AI

Bức Tranh Toàn Cảnh Về AI Agent

Chọn & Cấu Hình Agent Của Bạn

Công Cụ Thiết Yếu: Git & GitHub CLI

3. Dự Án Đầu Tiên: Từ Số Không Đến Lên Mạng

Đừng Bao Giờ Bắt Đầu Từ Đầu

Chọn Stack Phù Hợp

4. Từ Code Đến Sản Phẩm Thực Tế

Thanh Toán Với Stripe

Hosting: AWS, Hetzner & Các Lựa Chọn Khác

Cloudflare: Hiệu Suất & Bảo Vệ

CI/CD Với GitHub Actions

5. Chiến Thuật Marketing & SEO

Chiến Lược Tăng Trưởng Tự Nhiên

SEO, Nội Dung & Phân Phối

Bonus: Mã Nguồn Sẵn Sàng Sử Dụng

1. Giới Thiệu

Bạn đang đọc dòng này ngay lúc này vì ở đâu đó, bằng cách nào đó, sự kết hợp của các chiến lược marketing, kỹ thuật SEO, và định vị thông minh đã đưa khóa học này đến màn hình của bạn. Chỉ riêng điều đó đã nói lên một điều: các chiến thuật trong hướng dẫn này thực sự hiệu quả. Và vâng, bạn sẽ được học từng chiến thuật một.

Thế Giới Đã Thay Đổi

Phát triển phần mềm không còn như trước nữa. Vài năm trước, xây dựng một ứng dụng web đòi hỏi hàng tháng làm việc, một đội ngũ lập trình viên, và ngân sách đáng kể. Ngày nay, một người duy nhất với đúng công cụ và đúng kiến thức có thể xây dựng và ra mắt ứng dụng sẵn sàng cho production chỉ trong vài ngày, không phải vài tháng.

Khóa học này được viết bởi những chuyên gia làm việc trong ngành và sử dụng các công cụ AI agentic hàng ngày để xây dựng sản phẩm thực. Chúng tôi không dạy lý thuyết từ sách giáo khoa. Chúng tôi dạy những gì thực sự hoạt động trong thế giới thực, ngay bây giờ.

Tại Sao Khóa Học Này Tồn Tại

AI và LLM giờ đây là những công cụ thực thi tốt hơn và nhanh hơn con người. Chúng có thể viết code, debug, tái cấu trúc, và triển khai với tốc độ mà không con người nào có thể sánh được. Nhưng có một thứ chúng thiếu về cơ bản: **ý tưởng**.

Các mô hình AI là những công cụ thực thi phi thường, nhưng chúng không phải là người sáng tạo. Chúng không nhìn thấy khoảng trống trên thị trường và nghĩ "Mình có thể xây dựng thứ gì đó để giải quyết vấn đề này." Đó là công việc của bạn. Vai trò của bạn là kiến trúc sư của ý tưởng. AI là người xây dựng của bạn.

Khi bạn đưa ra hướng dẫn kém cho một LLM, nó vẫn sẽ tạo ra thứ gì đó. Nó sẽ không dừng lại để giải thích điều gì thực sự tốt hơn. Chất lượng của những gì bạn xây dựng tỷ lệ thuận trực tiếp với chất lượng kiến thức của bạn. Khóa học này sẽ lấp đầy khoảng cách đó.

Đây không chỉ là một khóa học về phát triển phần mềm. Đây là bản thiết kế hoàn chỉnh để đi từ ý tưởng đến sản phẩm thực tế, tạo ra doanh thu. Bao gồm cả các chiến thuật marketing và SEO cực kỳ hiệu quả — chính những kỹ thuật đã đưa bạn đến trang này.

2. Bộ Công Cụ Phát Triển Ưu Tiên AI

Công cụ bạn chọn sẽ quyết định tốc độ bạn di chuyển. Trong chương này, chúng tôi phân tích các AI coding agent hiện có, giúp bạn chọn đúng công cụ, và dạy bạn các chiến lược prompting phân biệt người mới với chuyên gia.

Lưu ý: Hướng dẫn này được viết trên Windows, nhưng tất cả công cụ và các bước đều hoạt động giống hệt trên macOS và Linux. Google Antigravity, Claude Code, và mọi ứng dụng khác được thảo luận trong khóa học này đều hoàn toàn đa nền tảng. Nếu bạn đang dùng Mac hoặc Linux, chỉ cần làm theo các bước tương tự — giao diện và quy trình làm việc đều giống nhau.

Bức Tranh Toàn Cảnh Về AI Agent

Không gian công cụ AI coding đã bùng nổ. Nhưng không phải tất cả công cụ đều được tạo ra như nhau. Một số chỉ là công cụ gợi ý tự động được nâng cấp. Số khác là các agent tự chủ hoàn toàn, có thể đọc toàn bộ codebase của bạn, lên kế hoạch, thực hiện thay đổi trên nhiều file, chạy test, và tự sửa lỗi. Hiểu sự khác biệt này là rất quan trọng.

Có hai loại công cụ AI coding cơ bản:

- **Trợ lý inline** — Chúng nằm trong trình soạn thảo và gợi ý code khi bạn gõ. Hãy nghĩ đến GitHub Copilot ban đầu. Chúng mang tính phản ứng: đợi bạn viết, rồi cố đoán phần tiếp theo. Hữu ích, nhưng có giới hạn.
- **Công cụ agentic** — Đây là một thứ hoàn toàn khác. Bạn giao cho chúng một nhiệm vụ bằng ngôn ngữ tự nhiên, và chúng tự động lên kế hoạch, viết code, chỉnh sửa, debug, và lặp lại. Chúng không chỉ gợi ý một dòng code. Chúng xây dựng tính năng. Đây mới là sức mạnh thực sự.

Đây là những công cụ quan trọng ngay lúc này:

Claude Code (bởi Anthropic)

Claude Code, theo kinh nghiệm của chúng tôi, là AI coding agent mạnh nhất hiện có. Đây là một agent hoạt động trong terminal, trực tiếp trong thư mục dự án của bạn. Bạn đưa ra hướng dẫn bằng ngôn ngữ tự nhiên, và nó đọc file, viết code, chạy lệnh, tạo commit, và tự sửa lỗi. Nó có toàn quyền truy cập vào hệ thống file và shell, điều này làm nó cực kỳ hiệu quả cho công việc phát triển thực tế. Bạn có thể cài đặt qua npm (`npm install -g @anthropic-ai/claude-code`) hoặc sử dụng như extension trên VS Code, điều mà chúng tôi sẽ thảo luận ngay sau đây.

Google Antigravity

Antigravity là môi trường phát triển của Google, mang khả năng chat và agent AI trực tiếp vào quy trình làm việc của bạn. Hãy coi nó như một không gian làm việc thông minh, nơi bạn có thể tương tác với các AI agent qua giao diện chat, và từ mỗi cuộc hội thoại agent bạn có thể mở trình soạn thảo VS Code tích hợp. Đây là điểm mấu chốt: Antigravity cung cấp lớp AI hội thoại, trong khi VS Code cung cấp sức mạnh chỉnh sửa code. Sự kết hợp này rất liền mạch — bạn thảo luận về những gì cần xây dựng với agent, rồi nhảy thẳng vào code mà không cần chuyển cửa sổ.

Cursor

Cursor là một bản fork của VS Code với AI được tích hợp sâu. Nó có cả gợi ý inline và chế độ agentic "Composer" cho phép bạn mô tả thay đổi trên nhiều file. Giao diện rất quen thuộc nếu bạn đã dùng VS Code, và nó hỗ trợ nhiều mô hình (Claude, GPT, v.v.). Đây là một công cụ tốt, đặc biệt nếu bạn thích quy trình làm việc hoàn toàn trực quan. Tuy nhiên, khả năng agentic của nó, dù tốt, không tự chủ bằng Claude Code. Bạn thường sẽ cần xem xét và phê duyệt từng thay đổi từng bước một.

Cấu hình chúng tôi khuyến nghị: **Google Antigravity + Claude Code như extension VS Code**. Sử dụng chat agent của Antigravity để lên kế hoạch và thảo luận công việc, rồi mở VS Code tích hợp và để Claude Code xử lý phần thực thi nặng. Điều này cho bạn tốt nhất từ cả hai thế giới: hội thoại thông minh để lên kế hoạch, và thực thi code tự chủ để xây dựng.

Chọn & Cấu Hình Agent Của Bạn

Cài đặt và chạy một công cụ agentic là phần dễ. Khai thác tối đa nó đòi hỏi hiểu cách nó hoạt động, chọn gói đăng ký phù hợp, và cấu hình đúng cách.

Gói đăng ký Claude: bắt đầu với Pro

Claude Code yêu cầu tài khoản Anthropic. Chúng tôi khuyên bạn bắt đầu với **gói đăng ký Claude Pro**, đây là gói trả phí cơ bản. Giá cả phải chăng và cho bạn quyền truy cập Claude Code với đầy đủ tính năng. Đây là điểm khởi đầu hoàn hảo để thử nghiệm, học quy trình làm việc, và xây dựng dự án đơn giản đầu tiên.

Tuy nhiên, hãy lưu ý rằng gói Pro có **giới hạn sử dụng**. Nếu bạn dùng Claude Code chuyên sâu, bạn sẽ chạm giới hạn khá nhanh. Để học và làm dự án nhỏ, nó quá đủ. Nhưng nếu bạn đã biết mình muốn dùng Claude Code làm công cụ phát triển chính và dự định làm việc trên đó nhiều giờ mỗi ngày, hãy cân nhắc nâng cấp lên gói cao hơn (như Max) ngay từ đầu. Các gói cao hơn có mức sử dụng nhiều hơn đáng kể, nghĩa là ít bị gián đoạn hơn và quy trình làm việc mượt mà hơn khi xây dựng dự án lớn.

Mô hình: Claude Opus 4.6

Chúng tôi hiện khuyến nghị sử dụng **Claude Opus 4.6** làm mô hình phát triển chính. Đây là mô hình tốt nhất hiện tại để xây dựng website và ứng dụng. Nó hiểu các kiến trúc phức tạp, viết code sạch và sẵn sàng cho production, xử lý thay đổi trên nhiều file với độ chính xác đáng kinh ngạc, và hiếm khi cần sửa lại ngay lần đầu. Khi làm việc với Claude Code, hãy đặt Opus 4.6 làm mô hình mặc định. Sự khác biệt về chất lượng đầu ra so với các mô hình nhỏ hơn là ngay lập tức có thể nhận thấy.

Thiết lập Antigravity

Từ thời điểm này, hướng dẫn này sử dụng **Google Antigravity** làm môi trường phát triển chính. Đây là cách chuẩn bị mọi thứ.

Bước 1: Tạo thư mục dự án. Trước khi mở Antigravity, tạo một thư mục trên máy tính nơi dự án đầu tiên sẽ nằm. Ví dụ, trên Windows bạn có thể tạo `C:\Projects\my-first-app`, hoặc trên Mac/Linux `~/Projects/my-first-app`. Đây là thư mục mà Antigravity sẽ mở như workspace. Nó có thể trống lúc này — chúng ta sẽ điền code vào sau trong khóa học.

Bước 2: Cài đặt và khởi chạy Antigravity. Tải Google Antigravity, cài đặt, và mở nó. Đăng nhập bằng tài khoản Google khi được yêu cầu.

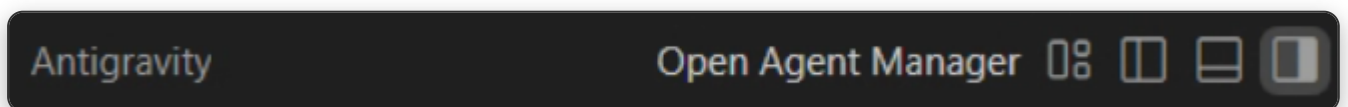
Trong quá trình cài đặt, Antigravity sẽ yêu cầu bạn cấu hình một vài chính sách. Để có quy trình phát triển tự chủ và nhanh chóng, chúng tôi khuyến nghị chọn **cấu hình tùy chỉnh** và đặt các tùy chọn sau:

- **Terminal Execution Policy** → Always Proceed
- **Review Policy** → Always Proceed
- **JavaScript Execution** → Always Proceed

Với các cài đặt này, các agent của Antigravity sẽ có thể thực thi lệnh, ghi file, và chạy code tự chủ mà không cần xác nhận ở mỗi bước. Đây là điều cho phép trải nghiệm phát triển nhanh chóng và mượt mà mà chúng tôi hướng tới trong khóa học này.

Cảnh báo bảo mật: Khi bạn cho phép các agent tự động tiến hành, cả Antigravity và Claude Code đều có thể thực thi hành động trên hệ thống của bạn mà không cần phê duyệt thủ công. Điều này có nghĩa là bạn cần cẩn thận với những gì bạn cho chúng tiếp cận. Đừng trỏ agent vào codebase có thể chứa malware, và hãy thận trọng với link hoặc tài nguyên từ nguồn không đáng tin. Trong kỷ nguyên AI, có những vector tấn công mới — kẻ xấu có thể tạo nội dung được thiết kế đặc biệt để lừa LLM thực thi các lệnh có hại. Luôn theo dõi những gì agent của bạn đang làm. Chúng tôi khuyến nghị cấu hình "Always Proceed" để nhanh, nhưng hãy cảnh giác và xem xét kết quả thường xuyên.

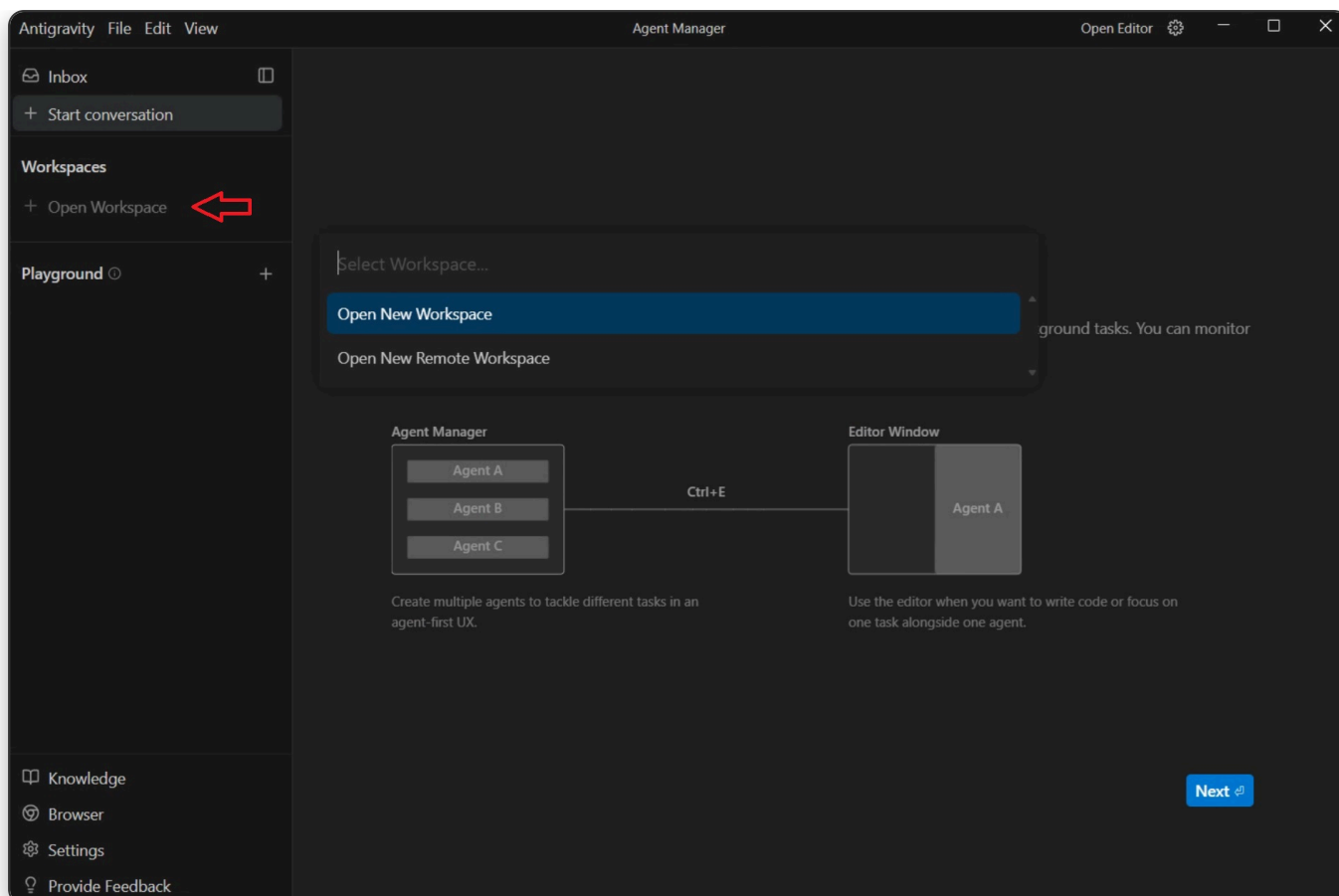
Bước 3: Mở Agent Manager. Khi Antigravity đang chạy, nhấp vào **"Open Agent Manager"** trên thanh trên cùng của cửa sổ.



Nút "Open Agent Manager" trên thanh trên cùng của Antigravity.

Agent Manager là trung tâm điều khiển của Antigravity. Đây là nơi bạn quản lý dự án, bắt đầu các cuộc hội thoại AI, và mở không gian làm việc phát triển.

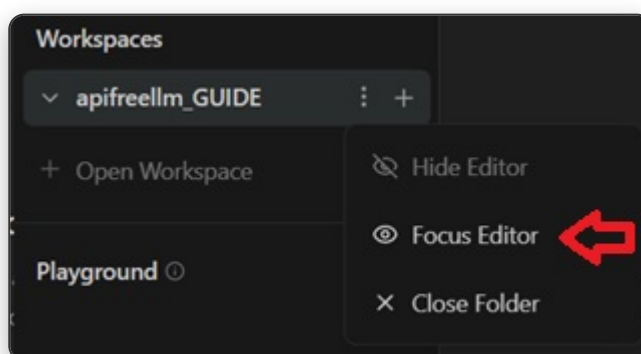
Bước 4: Tạo workspace đầu tiên. Trong Agent Manager, nhìn vào thanh bên trái. Dưới mục **"Workspaces"**, nhấp **"+ Open Workspace"**. Một menu thả xuống sẽ xuất hiện — chọn **"Open New Workspace"**.



Nhấp "+ Open Workspace" trong thanh bên trái, rồi chọn "Open New Workspace".

Antigravity sẽ yêu cầu bạn chọn một thư mục. Điều hướng đến thư mục dự án bạn đã tạo ở Bước 1 và chọn nó. Workspace mới sẽ xuất hiện trong thanh bên trái dưới "Workspaces", với tên của thư mục bạn đã chọn. Hãy coi mỗi workspace như một phiên phát triển riêng biệt — một workspace cho mỗi dự án. Bạn có thể tạo bao nhiêu workspace tùy ý, và chuyển đổi giữa chúng từ Agent Manager bất cứ lúc nào.

Bước 5: Mở trình soạn thảo. Khi workspace đã được tạo, bạn sẽ thấy tên của nó trong thanh bên. Nhấp vào **ba dấu chấm dọc (:)** bên cạnh tên workspace, rồi chọn "**Focus Editor**". Thao tác này mở môi trường VS Code đầy đủ cho workspace đó, nơi bạn sẽ viết và chỉnh sửa code.

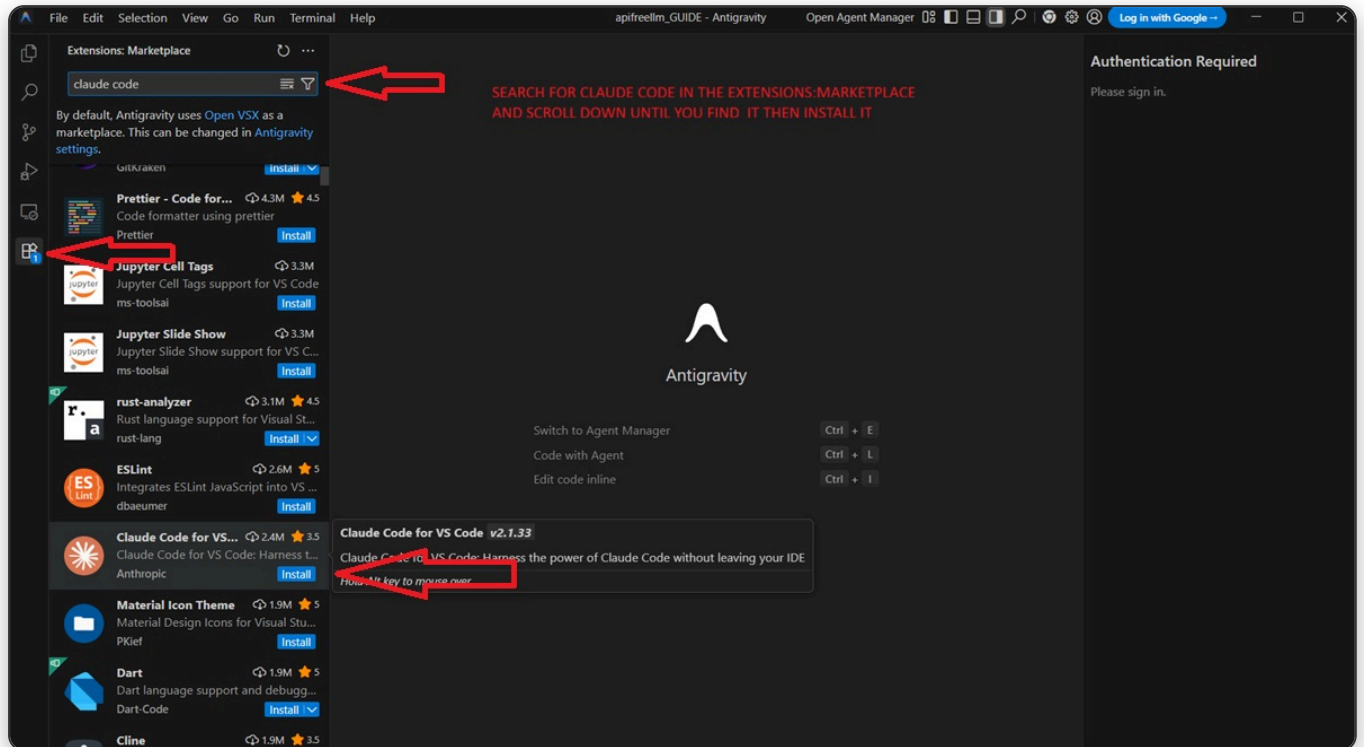


Nhấp ba dấu chấm bên cạnh tên workspace và chọn "Focus Editor".

Cài đặt Claude Code như extension VS Code

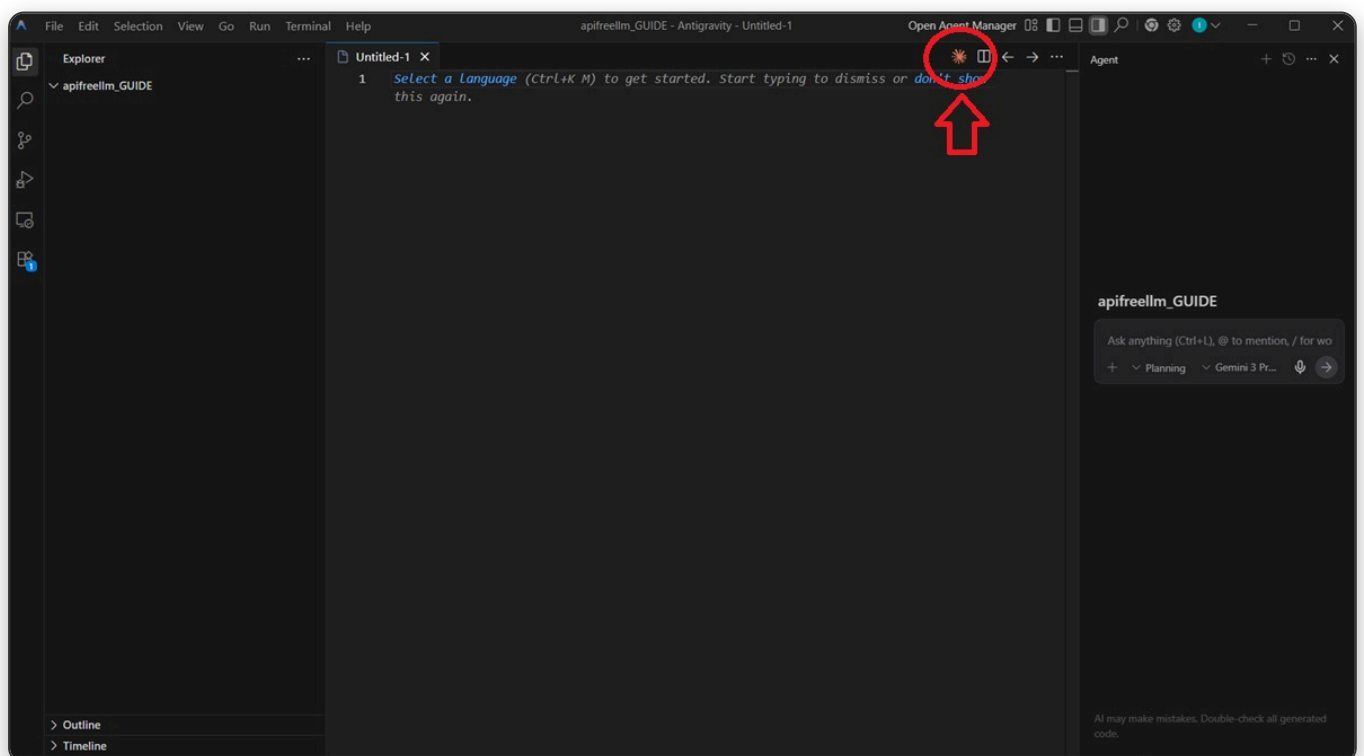
Giờ khi bạn đã mở trình soạn thảo, đã đến lúc cài Claude Code. Vì trình soạn thảo dựa trên VS Code, bạn có quyền truy cập marketplace extension. Nhấp vào biểu tượng **Extensions** trong thanh bên

trái (hoặc nhấn `Ctrl+Shift+X`). Trong thanh tìm kiếm, gõ **"claude code"**. Bạn có thể cuộn xuống qua các kết quả — tìm **"Claude Code for VS Code"** của Anthropic. Khi tìm thấy, nhấp nút **Install**.



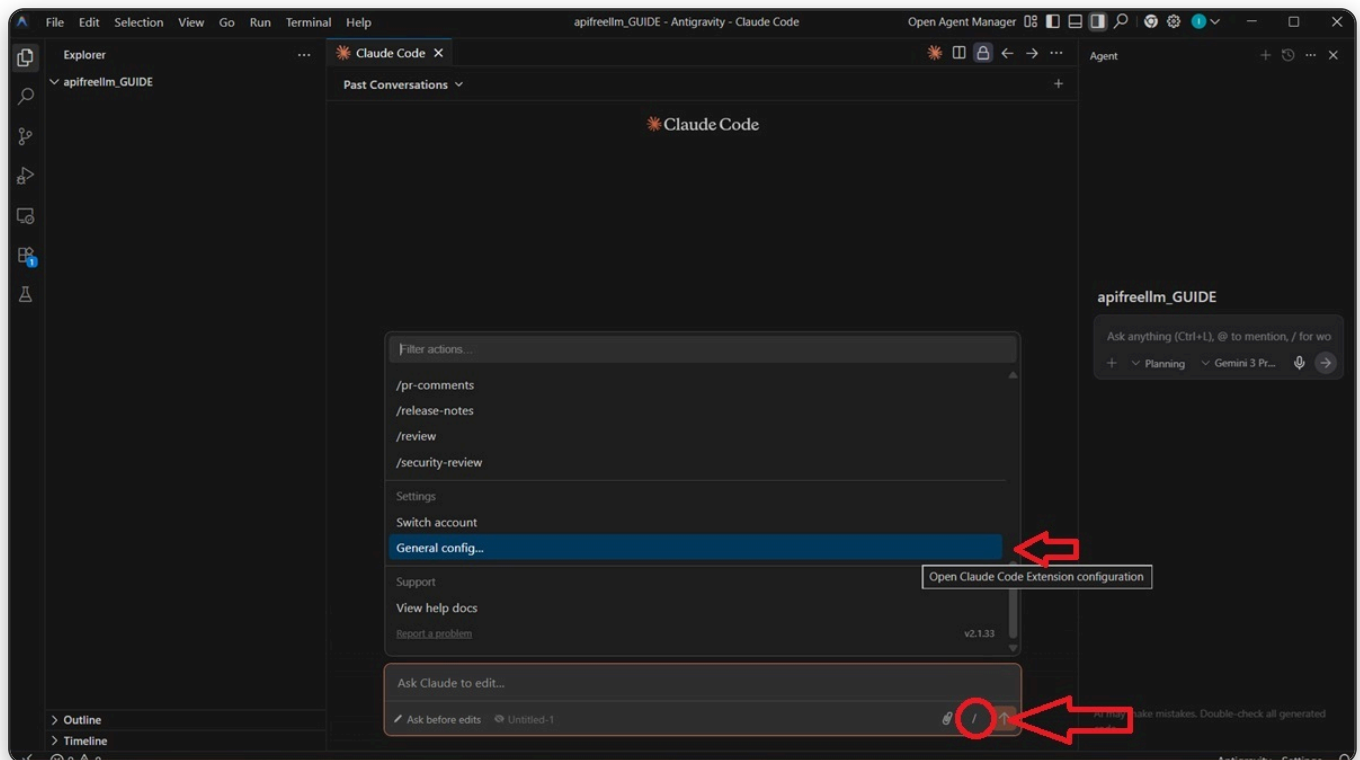
Tìm "claude code" trong marketplace Extension, cuộn xuống để tìm, và nhấp Install.

Sau khi cài đặt, đóng bảng Extensions và mở một file mới (hoặc bất kỳ file nào trong dự án). Bạn sẽ thấy một biểu tượng **Claude Code** nhỏ xuất hiện ở khu vực phía trên bên phải của trình soạn thảo — nó trông như một biểu tượng nhỏ màu cam. Nhấp vào đó để mở bảng chat Claude Code.



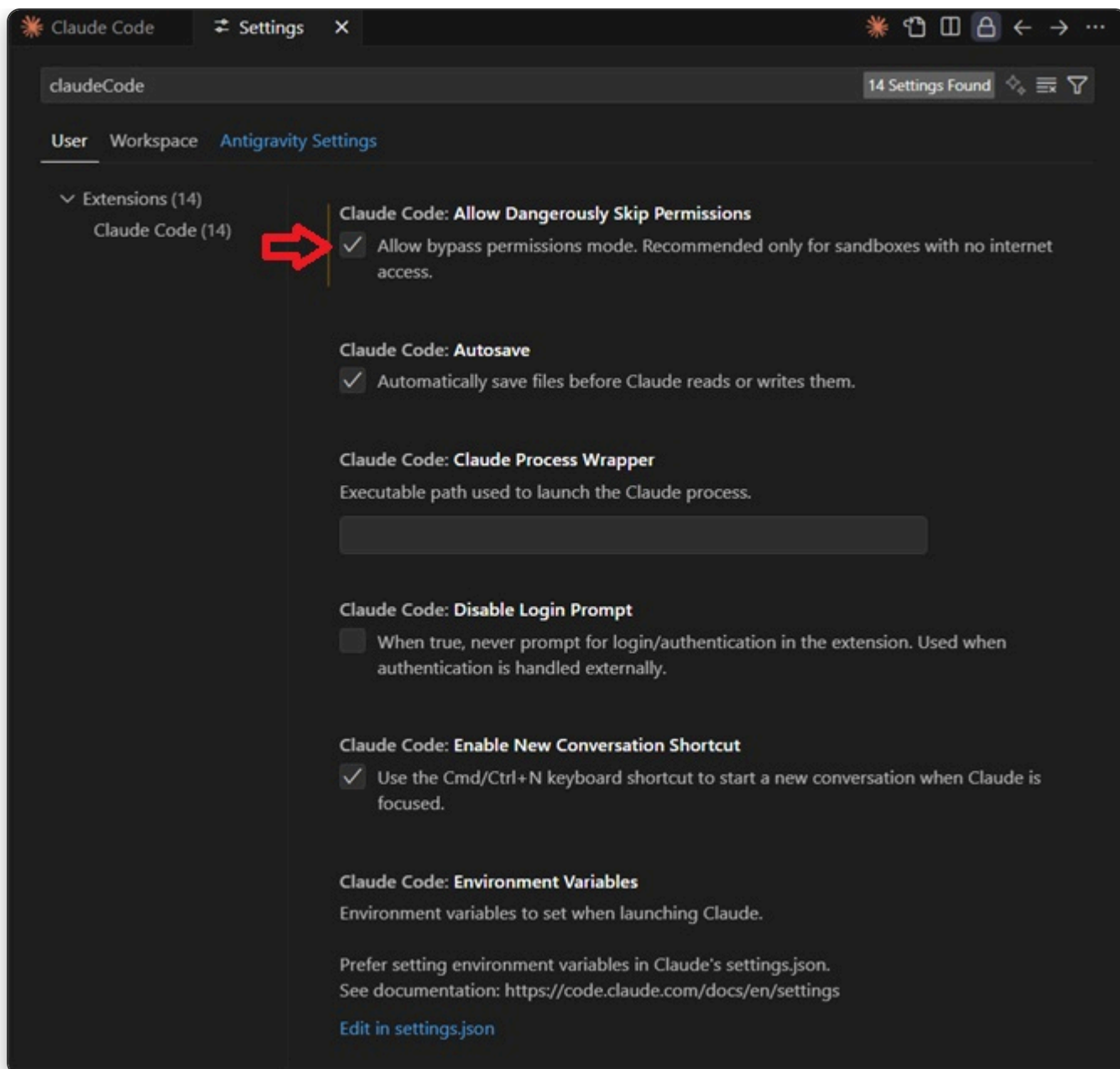
Nút Claude Code (được khoanh tròn) xuất hiện ở phía trên bên phải trình soạn thảo. Nhấp vào để mở chat Claude Code.

Claude Code sẽ yêu cầu bạn đăng nhập bằng tài khoản Anthropic (tài khoản có gói Pro). Sau khi đăng nhập, bạn cần cấu hình nó. Trong bảng chat Claude Code, nhấn nút **/** ở cuối bảng. Một menu sẽ xuất hiện với nhiều lệnh khác nhau. Cuộn xuống mục **Settings** và nhấn **"General config..."** để mở cấu hình extension Claude Code.



Nhấn nút **/** ở cuối, rồi cuộn đến **Settings** và chọn **"General config..."** để mở cấu hình.

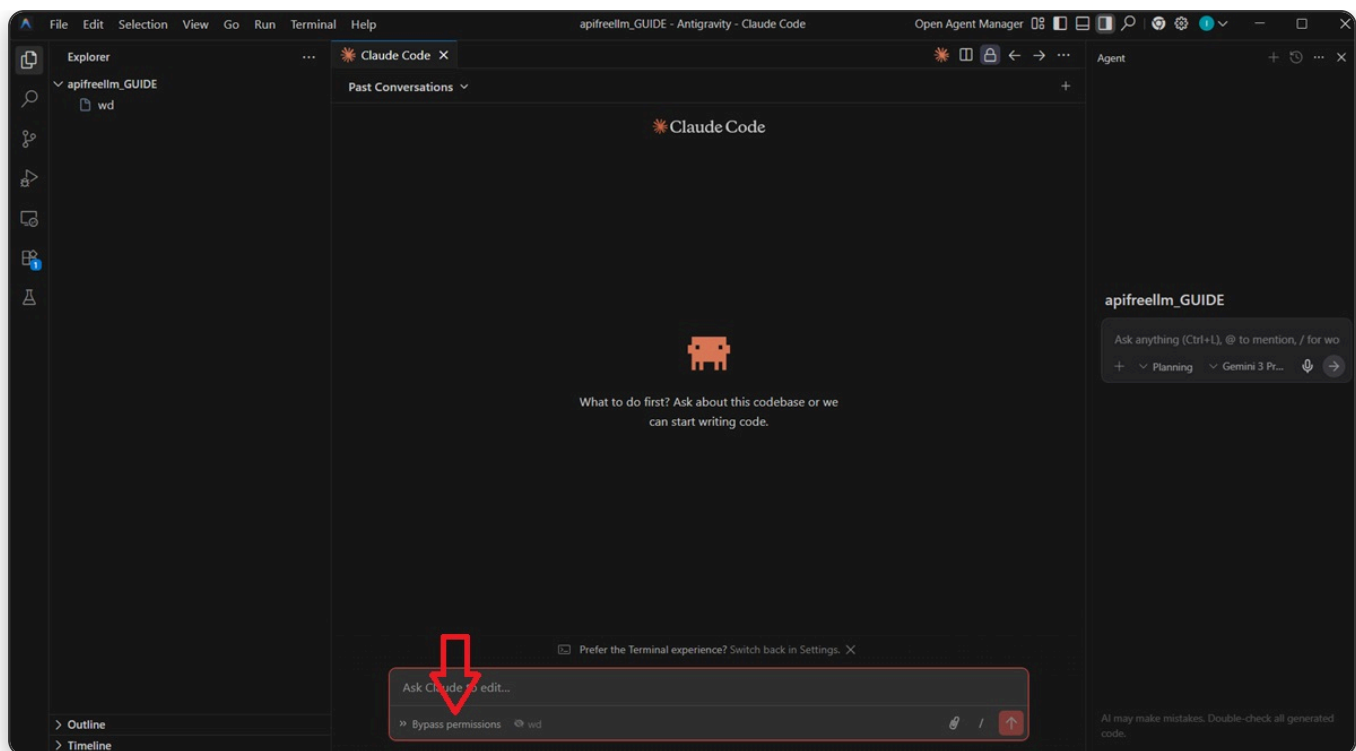
Trong bảng cấu hình mở ra, tìm tùy chọn có tên **"Allow Dangerously Skip Permissions"**. Bật toggle này. Khi được kích hoạt, bạn sẽ có thể chọn **"Bypass permissions"** làm chế độ quyền, cho phép Claude Code hoạt động hoàn toàn tự chủ — đọc file, viết code, chạy lệnh terminal, và thực hiện thay đổi mà không cần xác nhận ở mỗi bước.



Bật toggle "Allow Dangerously Skip Permissions" trong cài đặt Claude Code, rồi chọn "Bypass permissions".

Quan trọng: Với bypass permissions được bật, Claude Code sẽ thực thi hành động trên hệ thống mà không cần phê duyệt thủ công. Điều này rất cần thiết cho quy trình phát triển mượt mà, nhưng đi kèm với trách nhiệm. Hãy thận trọng với code bạn yêu cầu nó chạy, và không bao giờ trở nó vào repository không đáng tin hoặc URL đáng ngờ. Các AI agent có thể bị khai thác thông qua prompt injection — nội dung độc hại ẩn trong file hoặc website nhằm lừa agent thực thi lệnh có hại. Luôn xem xét những gì Claude Code đang làm, đặc biệt khi làm việc với tài nguyên bên ngoài.

Chúng tôi cũng khuyến nghị bật cài đặt đặt "**Bypass permissions**" làm **lựa chọn mặc định** cho các cuộc chat mới, để bạn không phải chọn thủ công mỗi khi bắt đầu cuộc hội thoại mới. Bây giờ đóng Claude Code và mở lại (bằng cách nhấp nút biểu tượng Claude Code lần nữa). Từ lúc này, bạn sẽ thấy tùy chọn "**Bypass permissions**" có sẵn trong nút chọn quyền ở cuối bảng chat Claude Code. Nhấp vào để kích hoạt nó.



Chọn "Bypass permissions" từ nút được chỉ trong ảnh chụp màn hình.

Với Bypass permissions được kích hoạt, Claude Code sẽ thực thi lệnh, tạo file, chỉnh sửa code, và chạy các thao tác terminal hoàn toàn tự động — mà không dừng lại để xin phê duyệt ở mỗi bước. Đây là điều cho phép bạn **tự động hóa 100% quy trình phát triển**: bạn mô tả những gì muốn xây dựng, và Claude Code sẽ tự chủ xây dựng từ đầu đến cuối. Không cần nhấp "Accept" trên từng thay đổi file hoặc thực thi lệnh nữa. Bạn đưa ra hướng dẫn, và agent xử lý phần còn lại.

Quản lý mức sử dụng một cách thông minh

Một điều bạn cần biết từ đầu: mỗi tương tác với Claude Code tiêu tốn **token**, và gói đăng ký của bạn có giới hạn sử dụng. Tin tốt là bạn có thể theo dõi chính xác mức sử dụng. Trong bảng chat Claude Code, nhấp nút **/** — trong số các lệnh có sẵn, bạn sẽ tìm thấy **mức sử dụng hiện tại**, hiển thị số token đã tiêu thụ và dung lượng còn lại.

Không phải tất cả mô hình đều tiêu thụ token với cùng tốc độ. **Claude Opus 4.6** là mô hình thông minh và có khả năng nhất, nhưng cũng tiêu thụ nhiều token nhất mỗi tương tác. Các mô hình nhỏ hơn như **Sonnet** hoặc **Haiku** kém mạnh hơn nhưng có giới hạn sử dụng cao hơn và tiêu thụ ít token hơn đáng kể. Khuyến nghị của chúng tôi: dùng **Opus cho các tác vụ phức tạp** đòi hỏi suy luận sâu, thay đổi trên nhiều file, hoặc quyết định về kiến trúc — đây là nơi trí tuệ của nó tạo ra sự khác biệt thực sự. Cho các tác vụ đơn giản hơn như sửa nhanh, chỉnh sửa nhỏ, hoặc câu hỏi đơn giản, hãy chuyển sang mô hình nhẹ hơn để tiết kiệm token Opus cho lúc thực sự cần.

Còn một chiến lược khác để tiết kiệm token Claude: **sử dụng agent tích hợp của Antigravity** cho các câu hỏi không đòi hỏi mức trí tuệ của Claude. Antigravity hỗ trợ nhiều mô hình, nhưng chúng tôi khuyến nghị dùng **Gemini** cho các câu hỏi nhanh — vì Antigravity là sản phẩm của Google, Gemini có giới hạn sử dụng cao nhất và cũng là mô hình nhanh nhất trên nền tảng. Cần nhớ nhanh về CSS? Muốn biết cú pháp lệnh Git? Tò mò về cách một thư viện hoạt động? Hỏi Antigravity thay vì Claude Code. Như vậy, bạn giữ token Claude cho công việc phát triển nặng, nơi chúng tạo ra tác động lớn nhất.

Như bạn có thể thấy trong ảnh chụp màn hình trước đó, chúng tôi khuyến nghị giữ **chat Claude Code ở bên trái** và **chat agent Antigravity ở bên phải**. Bố cục song song này cho bạn truy cập tức thì đến cả hai agent mọi lúc.

Quy trình thông minh: **Opus để xây dựng, mô hình nhẹ hơn cho tác vụ nhanh, Antigravity cho câu hỏi chung**. Như vậy bạn tối đa hóa giá trị gói đăng ký Claude. Nếu bạn sắp hết giới hạn sử dụng Claude, hãy nhớ rằng bạn luôn có agent Gemini của Antigravity ngay bên cạnh cho các câu hỏi đơn giản hơn — hãy dùng nó để tiết kiệm token Claude cho các tác vụ phát triển thực sự cần chúng.

Cấu hình thiết yếu

Bất kể bạn cài Claude Code bằng cách nào, các bước cấu hình sau sẽ cải thiện đáng kể kết quả:

- **Sử dụng file CLAUDE.md** — Đặt file `CLAUDE.md` trong thư mục gốc dự án. File này được Claude Code tự động đọc ở đầu mỗi phiên. Dùng nó để mô tả cấu trúc dự án, quy ước coding, tech stack, và bất kỳ quy tắc nào agent cần tuân theo. Hãy coi nó như tài liệu hướng dẫn cho lập trình viên AI của bạn.
- **Giữ dự án gọn gàng** — AI agent hoạt động tốt hơn đáng kể với codebase sạch, có cấu trúc tốt. Nếu code của bạn lộn xộn, agent sẽ tạo ra kết quả lộn xộn. Cấu trúc thư mục tốt, quy ước đặt tên rõ ràng, và pattern nhất quán tạo ra sự khác biệt lớn.
- **Sử dụng quản lý phiên bản** — Luôn làm việc với Git đã được khởi tạo. Điều này cho bạn lưới an toàn. Nếu agent mắc lỗi, bạn có thể hoàn nguyên ngay lập tức. Nó cũng cho phép agent tạo commit cho bạn, rất hữu ích để theo dõi những gì đã thay đổi và tại sao.

Công Cụ Thiết Yếu: Git & GitHub CLI

Trước khi bắt đầu xây dựng bất cứ thứ gì, có hai công cụ cần được cài trên hệ thống: **Git** và **GitHub CLI (gh)**. Đây là nền tảng cho bất kỳ quy trình phát triển hiện đại nào, và chúng cho phép AI agent quản lý repository code của bạn một cách tự chủ.

Tại sao Git và GitHub CLI quan trọng

Git là hệ thống quản lý phiên bản theo dõi mọi thay đổi trong dự án. Đây là lưới an toàn của bạn: nếu Claude Code mắc lỗi, bạn có thể hoàn nguyên ngay lập tức. Nó cũng cho phép agent tạo commit, quản lý branch, và giữ lịch sử sạch sẽ về sự phát triển của dự án — tất cả tự động.

GitHub CLI (gh) là công cụ dòng lệnh cho phép truy cập trực tiếp GitHub từ terminal. Đây là chìa khóa cho tự động hóa hoàn toàn: một khi `gh` được cài đặt và xác thực, Claude Code có thể tạo repository, đẩy code, quản lý pull request, cấu hình cài đặt repository, thiết lập GitHub Actions cho triển khai, thêm secret, và nhiều hơn nữa — tất cả từ terminal, mà bạn không cần mở GitHub trong trình duyệt.

Cài đặt Git và GitHub CLI

Cách đơn giản nhất để cài các công cụ này là nhờ **Claude Code** hoặc **Antigravity** làm cho bạn. Chỉ cần nói với agent: *"Cài đặt Git và GitHub CLI trên hệ thống của tôi."* Agent sẽ phát hiện hệ điều hành và chạy các lệnh cài đặt phù hợp. Trên Windows, nó thường dùng `winget` hoặc tải trình cài đặt; trên macOS, dùng `brew`; trên Linux, `apt` hoặc trình quản lý gói của hệ thống.

Nếu bạn muốn cài thủ công, bạn có thể tải Git từ trang web chính thức và GitHub CLI từ trang phát hành GitHub. Nhưng để AI xử lý thì nhanh hơn và tránh các lỗi cài đặt phổ biến.

Xác thực GitHub CLI

Sau khi cài đặt, bạn cần đăng nhập để `gh` có thể truy cập tài khoản GitHub. Một lần nữa, bạn có thể đơn giản nhờ agent: *"Đăng nhập GitHub CLI cho tôi."* Agent sẽ chạy `gh auth login` và hướng dẫn bạn qua quy trình xác thực, thường bao gồm mở link trình duyệt và nhập mã. Sau khi xác thực, agent có toàn quyền truy cập repository GitHub của bạn.

Đây là bước ngoặt. Với `gh` đã xác thực, bạn có thể nói với Claude Code những thứ như: *"Tạo repository private mới tên my-app, khởi tạo dự án, và đẩy code lên."* Hoặc sau này: *"Thiết lập GitHub Action tự động triển khai lên máy chủ mỗi khi push lên main."* Agent xử lý mọi thứ — tạo file, cấu hình secret, thiết lập workflow — mà bạn không cần rời trình soạn thảo. Đây mới là tự động hóa phát triển thực sự.

3. Dự Án Đầu Tiên: Từ Số Không Đến Lên Mạng

Môi trường của bạn đã sẵn sàng. Claude Code đã mở, Antigravity đang chạy, Git và GitHub CLI đã được cấu hình. Đã đến lúc xây dựng thứ gì đó thực sự. Từ đây trở đi, khóa học chuyển từ thiết lập sang chiến lược — các kỹ thuật thực tế và insight sẽ cho bạn lợi thế thực sự khi xây dựng với AI agent.

Đừng Bao Giờ Bắt Đầu Từ Đầu

Đây là lời khuyên quan trọng nhất trong toàn bộ khóa học: **đừng bao giờ xây dựng từ đầu.**

Mặc dù Claude Opus là mô hình cực kỳ tiên tiến và thông minh, nó vẫn sẽ mắc lỗi. Mọi mô hình AI đều vậy. Nó có thể hiểu sai cấu trúc dự án, sử dụng thư viện đã lỗi thời, tạo bố cục file không nhất quán, hoặc sinh code không khớp nhau khi dự án phát triển. Bắt đầu từ thư mục trống nghĩa là AI phải đưa ra hàng trăm quyết định mà không có điểm tham chiếu — và một số quyết định đó chắc chắn sẽ sai.

Đây là thực tế: **99.9% những gì bạn muốn xây dựng đã tồn tại** ở dạng nào đó. Dù là cửa hàng thương mại điện tử, bảng điều khiển SaaS, trang portfolio, ứng dụng mạng xã hội, hay nền tảng đặt lịch — ai đó đã xây dựng thứ tương tự. Và nhiều dự án trong số này là mã nguồn mở, có sẵn dưới dạng template trên GitHub, sẵn sàng để clone và tùy chỉnh.

Quy trình của bạn nên luôn bắt đầu như nhau: **tìm template trước**. Lên GitHub và tìm các dự án mã nguồn mở phù hợp với những gì bạn muốn xây dựng. Tìm một cái gần với tầm nhìn của bạn, clone vào thư mục dự án, rồi mở Claude Code vào đó. Giờ thay vì xây dựng từ số không, agent đang sửa đổi, cải thiện, và tùy chỉnh một codebase hiện có, đang hoạt động. Sự khác biệt về chất lượng và tốc độ là rất lớn.

Template không phải là gian lận — đó là chiến lược. Lập trình viên chuyên nghiệp sử dụng boilerplate và starter kit mọi lúc. Bạn không sao chép sản phẩm của người khác. Bạn đang sử dụng nền tảng cấu trúc và xây dựng sản phẩm độc đáo của riêng mình trên đó. AI hoạt động tốt hơn đáng kể khi có các pattern sẵn để theo thay vì phải phát minh mọi thứ từ hư không.

Chọn Stack Phù Hợp

Nếu tìm kiếm template không có kết quả và bạn thực sự cần bắt đầu dự án mới, công nghệ bạn chọn có thể tạo ra sự khác biệt — đặc biệt khi làm việc với AI agent. Tuy nhiên, không có lựa chọn bắt buộc duy nhất nào. Stack tốt nhất là stack đưa bạn đến sản phẩm hoạt động nhanh nhất.

Claude Code, giống như tất cả LLM, về cơ bản viết **code web** tốt hơn: HTML, CSS, JavaScript, và TypeScript. Đây là ngôn ngữ của internet, và là thứ các mô hình này được huấn luyện nhiều nhất. Dự án của bạn càng gần với công nghệ web, AI sẽ hoạt động càng tốt.

Cho **ứng dụng web**, **Next.js** là lựa chọn tuyệt vời nếu bạn tìm được. Đây là framework React hiện đại với server-side rendering tích hợp, rất quan trọng cho SEO. Claude Code hoạt động đặc biệt tốt với dự án Next.js: nó hiểu file-based routing, API route, server component, và toàn bộ hệ sinh thái. Bạn sẽ tìm thấy rất nhiều template Next.js trên GitHub cho hầu như mọi loại ứng dụng.

Cho **ứng dụng desktop** (file thực thi cho Windows, macOS, hoặc Linux), **Electron** là lựa chọn tuyệt vời. Electron cho phép bạn xây dựng ứng dụng desktop bằng HTML, CSS, và JavaScript — cùng các công nghệ web mà Claude giỏi. Vì giao diện về cơ bản là trang web được render trong cửa sổ native, AI có thể xây dựng ứng dụng desktop đẹp và chức năng với cùng sự dễ dàng như xây website.

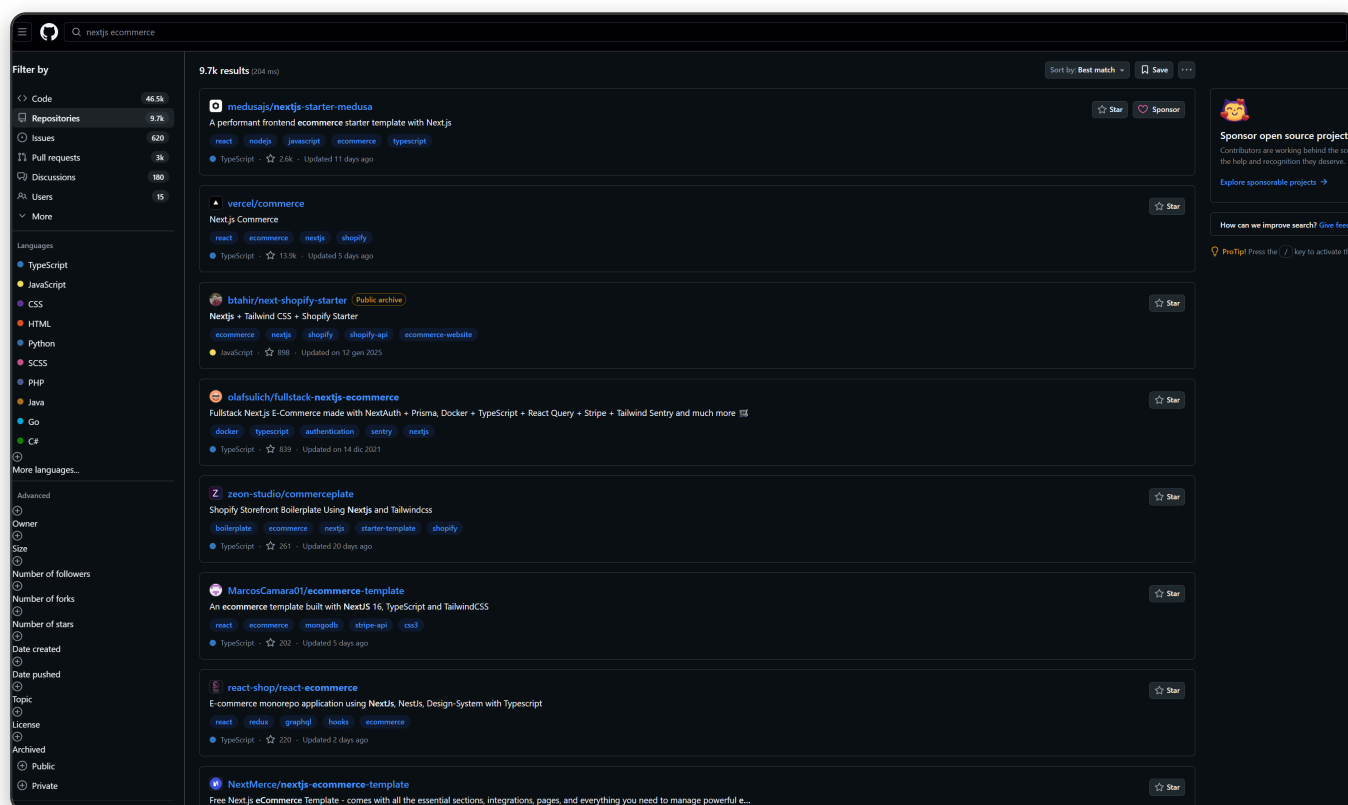
Nhưng đây là điểm quan trọng cần lưu ý: **một template tốt bằng stack cũ hầu như luôn tốt hơn bắt đầu từ đầu với stack mới**. Bạn có thể tìm thấy dự án PHP, jQuery, hoặc Laravel đúng là thứ bạn cần — hoàn chỉnh, cấu trúc tốt, đã được thử thách, với tất cả tính năng đã được triển khai. Trong trường hợp đó, hãy dùng nó. AI agent có thể làm việc với bất kỳ công nghệ nào, và thời gian bạn tiết kiệm từ nền tảng vững chắc, sẵn có vượt xa các lợi thế lý thuyết của framework mới hơn. Claude Code có thể hiểu và chỉnh sửa codebase PHP, Python, Ruby, hoặc bất kỳ ngôn ngữ nào khác.

Ưu tiên rất đơn giản: **tìm template tốt nhất có sẵn**. Nếu bạn tìm thấy hai template chất lượng tương đương, một dùng Next.js và cái kia dùng PHP, hãy chọn Next.js. Nhưng nếu template PHP hoàn chỉnh hơn, nhiều tính năng hơn, và được bảo trì tốt hơn — hãy chọn nó mà không ngần ngại. Chất lượng và tính hoàn chỉnh của điểm xuất phát quan trọng hơn sự hiện đại của stack.

Nguyên tắc chung: dùng bất cứ thứ gì bạn tìm được đưa bạn gần nhất đến mục tiêu. Ưu tiên công nghệ web hiện đại (Next.js, Electron, React Native) khi có sẵn, nhưng đừng bao giờ từ chối template tốt chỉ vì nó dùng stack cũ. Thời gian tiết kiệm từ nền tảng vững chắc luôn giá trị hơn sự thuần túy về stack.

Ví dụ thực tế: tìm template

Giả sử bạn muốn xây dựng cửa hàng thương mại điện tử. Thay vì nói với Claude Code *"Xây cho tôi trang thương mại điện tử từ đầu"*, hãy mở GitHub và tìm thứ gì đó như **"nextjs ecommerce"**. Bạn sẽ tìm thấy hàng chục dự án sẵn có với danh sách sản phẩm, giỏ hàng, quy trình thanh toán, và tích hợp thanh toán đã được xây sẵn.



Tìm nhanh trên GitHub cho "nextjs ecommerce" đã hiển thị nhiều template hứa hẹn.

Một điều quan trọng cần nhớ: nhiều template trên GitHub là **dự án freemium** — phần cốt lõi là mã nguồn mở và miễn phí, nhưng một số tính năng hoặc phiên bản cao cấp yêu cầu trả phí. Luôn kiểm tra README và giấy phép của repository trước khi chọn template. Tránh bất cứ thứ gì yêu cầu đăng ký trả phí hoặc chi phí ẩn mà bạn không cần.

Nhìn vào kết quả tìm kiếm, chúng ta có thể thấy **fullstack-nextjs-ecommerce** — và đây là điểm khởi đầu tuyệt vời. Hãy phân tích tại sao. Dự án sử dụng Next.js với TypeScript, đúng là thứ chúng ta muốn cho phát triển hỗ trợ AI. Nhưng điều thực sự nổi bật là những gì đã được tích hợp: **Stripe cho thanh toán, PostgreSQL với Prisma** cho cơ sở dữ liệu, và **NextAuth cho xác thực**. Đây là ba phần quan trọng nhất — và dễ gây lỗi nhất — của bất kỳ ứng dụng web nào.

Có **Stripe đã được tích hợp** đặc biệt có giá trị. Xử lý thanh toán liên quan đến webhook, quản lý session, xử lý lỗi, và các trường hợp biên có thể phức tạp đáng ngạc nhiên. Khi thanh toán gặp lỗi, khách hàng là người chịu thiệt — thanh toán thất bại, thanh toán trùng, hoặc quy trình checkout bị

hông là loại lỗi phá hủy niềm tin và khiến bạn mất tiền thực. Bắt đầu với template đã có tích hợp Stripe hoạt động giúp bạn tránh được những vấn đề này.

Dự án cũng sử dụng **PostgreSQL** làm cơ sở dữ liệu, đây là lựa chọn vững chắc. Postgres đáng tin cậy, có tài liệu tốt, có mặc định bảo mật tốt hơn một chút so với các lựa chọn thay thế như MySQL, và có thể dễ dàng host trên Amazon AWS, Hetzner, hoặc các nhà cung cấp tương tự — chúng tôi sẽ nói về thiết lập máy chủ và triển khai trong chương tiếp theo. Việc **NextAuth** đã được kết nối sẵn có nghĩa là xác thực người dùng được xử lý ngay từ đầu, một phần phức tạp khác bạn không cần xây từ đầu.

Đây là sức mạnh của việc bắt đầu với template phù hợp: thay vì dành hàng ngày (hoặc hàng tuần) thiết lập thanh toán, cơ sở dữ liệu, và xác thực — tất cả đều dễ bị sai — bạn bắt đầu với dự án mà các hệ thống quan trọng này đã hoạt động. Bạn có thể tập trung hoàn toàn vào việc tùy chỉnh sản phẩm theo tầm nhìn của mình: thay đổi thiết kế, thêm sản phẩm, sửa đổi logic kinh doanh, và xây dựng các tính năng làm cửa hàng của bạn trở nên độc đáo.

Khi bạn đã tìm được template, bước tiếp theo rất đơn giản: **tải về và đặt vào thư mục workspace**. Mở thư mục đó với Claude Code (hoặc Antigravity) và bắt đầu làm việc. Bạn có thể bảo Claude Code sửa đổi template trực tiếp — thay đổi thương hiệu, thêm trang mới, làm lại bố cục, tích hợp tính năng mới — hoặc bạn có thể dùng template làm **tài liệu tham khảo** cho dự án riêng. Ngay cả khi bạn đang xây dựng thứ hơi khác, việc có template trong cùng workspace nghĩa là Claude Code có thể xem nó, nghiên cứu các pattern code, và dùng chúng làm nền tảng cho những gì nó viết.

Đây là điểm then chốt: **luôn cho Claude Code thứ gì đó để tham khảo**. Khi agent có codebase vững chắc, đang hoạt động để xem, nó viết code tốt hơn đáng kể. Nó tuân theo cùng pattern, dùng cùng quy ước, và tạo ra kết quả nhất quán và đáng tin cậy. Khi không có gì để tham khảo và phải sinh mọi thứ từ đầu, đó là lúc lỗi xảy ra — cấu trúc file không nhất quán, sai phiên bản thư viện, code không khớp nhau. Template đóng vai trò như mỏ neo giữ AI có nền tảng và tạo ra kết quả chất lượng cao, mạch lạc.

Nếu template không có thanh toán hoặc cơ sở dữ liệu thì sao? Cũng không sao. Khóa học này bao gồm các file tích hợp Stripe sẵn có mà bạn có thể đưa trực tiếp cho Claude Code để tích hợp thanh toán vào bất kỳ dự án nào. Cho cơ sở dữ liệu, chúng tôi khuyến nghị PostgreSQL hoặc bất kỳ cơ sở dữ liệu nào template đã sử dụng — đừng chống lại lựa chọn của template trừ khi có lý do mạnh mẽ. Tương tự cho xác thực: nếu template dùng NextAuth, Clerk, hoặc hệ thống auth nào khác, hãy làm việc với nó. Mục tiêu là tận dụng những gì đã được xây dựng, không phải thay thế mọi thứ.

4. Từ Code Đến Sản Phẩm Thực Tế

Sản phẩm của bạn đã được xây dựng. Giờ nó cần chấp nhận thanh toán, chạy trên máy chủ, và nhanh chóng và an toàn cho người dùng toàn cầu. Chương này bao gồm các dịch vụ thiết yếu

biến dự án từ code cục bộ thành doanh nghiệp trực tuyến, sẵn sàng cho production.

Lưu ý về chương này. Chúng tôi giữ mọi thứ ngắn gọn và đi thẳng vào trọng tâm. Mục tiêu là cho bạn biết bạn cần làm **gì** và **tại sao** — không phải viết hướng dẫn dài dòng lãng phí thời gian. Bất kỳ LLM nào (Claude, Gemini, ChatGPT) đều có thể giải thích chi tiết, hướng dẫn bạn từng bước, và trả lời câu hỏi tốt hơn nhiều so với trang tĩnh. Bất cứ khi nào điều gì chưa rõ, hãy hỏi agent: *"Tôi đang theo một khóa học và nó nói tôi cần [làm X]. Bạn có thể giải thích ý nghĩa và hướng dẫn tôi?"* Cách này nhanh hơn, cá nhân hóa hơn, và luôn cập nhật.

Thanh Toán Với Stripe

Nếu sản phẩm của bạn bán bất cứ thứ gì, bạn cần bộ xử lý thanh toán. **Stripe** là chuẩn ngành: đáng tin cậy, tài liệu đầy đủ, và được hỗ trợ bởi hầu như mọi AI agent nhờ mức độ phổ biến rộng rãi.

Đây là những gì bạn cần làm:

- **Tạo tài khoản Stripe** tại stripe.com. Bạn sẽ nhận được **API key** (publishable key cho frontend, secret key cho backend) và quyền truy cập cả **chế độ test** (thanh toán giả cho phát triển) và **chế độ live** (tiền thật).
- **Thiết lập webhook** trong bảng điều khiển Stripe. Webhook là các URL trên máy chủ mà Stripe gọi khi có sự kiện xảy ra — thanh toán thành công, đăng ký được gia hạn, thanh toán thất bại. Đây là cách ứng dụng biết khi nào kích hoạt đăng ký, xác nhận đơn hàng, hoặc xử lý lỗi. Bạn cần webhook cho cả **test cục bộ** (Stripe có công cụ CLI chuyển tiếp sự kiện đến localhost) và **production** (trở đến máy chủ live). Luôn kiểm tra mọi thứ hoạt động cục bộ trước khi đưa lên production.
- **Tích hợp Stripe vào dự án.** Nếu template đã có Stripe, chỉ cần điền API key. Nếu không, khóa học này bao gồm các file tích hợp Stripe sẵn có — đặt chúng vào workspace và bảo Claude Code tích hợp. Stripe hỗ trợ thanh toán một lần và đăng ký định kỳ, cả hai đều sử dụng trang checkout được host sẵn, xử lý xác thực thẻ, 3D Secure, và tuân thủ PCI cho bạn.

Một quy tắc quan trọng: **luôn xác minh thanh toán phía server thông qua webhook**, không bao giờ tin frontend. Webhook là Stripe thông báo trực tiếp cho server rằng tiền thực sự đã được chuyển — đây là nguồn sự thật đáng tin cậy duy nhất.

Hosting: AWS, Hetzner & Các Lựa Chọn Khác

Ứng dụng của bạn cần một nơi để chạy. Tin tốt: **AWS cho bạn Free Tier** khi đăng ký — trong **một năm đầy đủ**, bạn nhận được **server t2.micro** và **database micro** hoàn toàn miễn phí. Đủ để host dự án đầu tiên trong khi bạn xác thực ý tưởng và bắt đầu có người dùng.

Không biết cách thiết lập tài khoản AWS hoặc sử dụng Free Tier? Hãy hỏi Claude hoặc Antigravity — chúng sẽ hướng dẫn bạn từng bước.

Khi bạn vượt qua free tier, đây là những gì bạn cần biết về kích cỡ server:

- **t3.small** là lựa chọn tối ưu cho hầu hết ứng dụng — CPU tốt, RAM đủ, và giá hợp lý (~\$16/tháng trên AWS). Chữ "t" là thể hệ instance; các thể hệ cũ hơn (t2, v.v.) đôi khi có thể đắt hơn mà hiệu năng kém hơn, nên hãy dùng thể hệ mới nhất có sẵn.
- **Hetzner** là lựa chọn thay thế châu Âu *rẻ hơn đáng kể* — bạn có thể có hiệu năng tương đương t3.small với khoảng **\$4/tháng**. Tức là rẻ hơn khoảng 4 lần so với AWS cho cấu hình tương tự.
- **Không phải ứng dụng nào cũng cần server**. Nếu dự án là trang tĩnh hoặc ứng dụng JAMstack, bạn có thể không cần server riêng. Các nền tảng như Vercel, Netlify, hoặc Cloudflare Pages có thể host miễn phí hoặc gần như miễn phí. Luôn hỏi LLM: *"Hosting tốt nhất cho ứng dụng cụ thể của tôi là gì?"*

Khuyến nghị của chúng tôi: **bắt đầu với AWS Free Tier** cho năm đầu tiên, rồi đánh giá xem Hetzner hoặc nhà cung cấp khác có phù hợp hơn cho ngân sách và vị trí khán giả của bạn không. Nếu hầu hết người dùng ở châu Âu, server Đức của Hetzner sẽ cho độ trễ thấp hơn. Nếu khán giả toàn cầu hoặc ở Mỹ, các region AWS có thể phù hợp hơn.

SSH key & quản lý server bằng AI. Để Claude Code kết nối và quản lý server từ xa, bạn cần thiết lập **SSH key**. Nhờ Claude tạo và cấu hình nó trên server. Khi đã kết nối, Claude có thể triển khai code, quản lý dịch vụ, xử lý sự cố — tất cả từ terminal của bạn. Cho tác vụ quản lý server, chúng tôi khuyến nghị dùng **Opus** để có kết quả tốt nhất, dù Sonnet cũng hoạt động được nếu bạn muốn tiết kiệm token (với xác suất lỗi hơi cao hơn).

Cloudflare: Hiệu Suất & Bảo Vệ

Khi ứng dụng đã trên server, bạn cần thứ gì đó đứng phía trước để bảo vệ và tăng tốc nó. Đó là **Cloudflare**. Tin tốt: **gói miễn phí là quá đủ** cho đại đa số website. Bạn không cần gói trả phí.

Nhưng trước tiên, bạn cần một **tên miền**. Chúng tôi khuyến nghị mua trực tiếp từ **Cloudflare** hoặc **Namecheap** — cả hai đều đáng tin cậy và giá hợp lý. Khi có tên miền, bạn cần cấu hình **DNS** trỏ đến địa chỉ IP của server AWS hoặc Hetzner. Hãy hỏi LLM: *"Tôi mua tên miền trên [Cloudflare/Namecheap]. Làm thế nào để thiết lập DNS trỏ đến server tại [IP của bạn]?"* Nó sẽ hướng dẫn bạn.

Tại sao Cloudflare quan trọng đến vậy? Nó bảo vệ trang web khỏi **tấn công DDoS**, các **lỗ hổng bảo mật** khác nhau, và các lỗ hổng web phổ biến. Nó cũng cung cấp **bộ nhớ cache toàn cầu**, nghĩa là nội dung được phục vụ từ server gần người dùng, làm trang web nhanh hơn trên toàn thế giới. Không có dịch vụ như Cloudflare, bạn đang phơi bày trang web trước rủi ro nghiêm trọng — đặc biệt trong kỷ nguyên AI, khi bất kỳ ai cũng có thể dùng công cụ AI để tìm lỗ hổng, khai thác cấu hình sai, hoặc thậm chí đánh cắp dữ liệu từ website bảo vệ kém. Đừng bỏ qua bước này.

Mẹo nhanh: chế độ Flexible SSL. Khi thiết lập Cloudflare, bạn có thể chọn chế độ **Flexible** SSL. Điều này có nghĩa là kết nối giữa Cloudflare và server dùng HTTP thuần, nhưng kết nối giữa Cloudflare và người dùng cuối là HTTPS — nên khách truy cập thấy ổ khóa an toàn. Điều này giúp bạn không cần cấu hình chứng chỉ SSL trên server, rất tốt để bắt đầu nhanh. Cho ứng dụng production xử lý dữ liệu nhạy cảm, bạn nên chuyển sang chế độ **Full** (mã hóa đầu cuối). Nhưng cho các bài test và ra mắt đầu tiên, Flexible hoàn toàn ổn và tiết kiệm nhiều thời gian thiết lập. Hỏi LLM để giải thích sự khác biệt nếu bạn chưa chắc chế độ nào phù hợp cho dự án.

Cloudflare cung cấp nhiều hơn chỉ bảo vệ. Gói miễn phí bao gồm các tính năng mạnh mẽ mà nhiều lập trình viên thậm chí không biết:

- **Cloudflare Pages** — hosting tĩnh miễn phí cho ứng dụng frontend (React, Next.js static export, Vue, v.v.). Bạn push lên GitHub, Cloudflare tự động build và deploy. Không cấu hình, không chi phí. Hoàn hảo cho landing page, portfolio và ứng dụng JAMstack.
- **Edge Functions (Workers)** — chạy code serverless ở edge, gần người dùng của bạn, trên gói miễn phí. Tuyệt vời cho API routes, chuyển hướng, A/B testing và logic backend nhẹ mà không cần server riêng.
- **CDN & Caching** — các tài nguyên tĩnh của bạn (hình ảnh, CSS, JS) được cache toàn cầu, giúp trang web của bạn tải cực nhanh từ bất kỳ đâu trên thế giới.

Câu hỏi quan trọng là: **ứng dụng của bạn thực sự cần server riêng, hay Cloudflare có thể xử lý miễn phí?** Hỏi Claude: *"Tôi đang xây dựng [mô tả ứng dụng]. Tôi có cần server riêng trên AWS/Hetzner không, hay có thể dùng Cloudflare Pages và Workers miễn phí?"* Claude sẽ phân tích trường hợp cụ thể của bạn và cho bạn biết lựa chọn tốt nhất. Bạn có thể ngạc nhiên khi biết bao nhiêu dự án có thể chạy hoàn toàn trên gói miễn phí của Cloudflare.

API Keys: Tự Động Hóa Mọi Thứ

Đây là mẹo thay đổi cuộc chơi mà hầu hết tutorial bỏ qua: **tạo API key cho các nhà cung cấp hosting** và đưa cho Claude. Điều này cho phép Claude quản lý hạ tầng trực tiếp từ terminal — không cần click trên dashboard, không copy-paste, không làm thủ công.

- **Cloudflare API Key** — vào dashboard Cloudflare → My Profile → API Tokens → tạo token. Với key này, Claude có thể tự động cấu hình DNS records, tạo dự án Pages, quản lý Workers, cập nhật cài đặt SSL và nhiều hơn nữa. Nói với Claude: *"Đây là Cloudflare API token của tôi. Cấu hình DNS cho domain trỏ đến IP server của tôi."* Xong trong vài giây.
- **AWS Access Keys** — vào AWS IAM → tạo access key. Với key này, Claude có thể quản lý EC2 instances, cấu hình security groups, tạo RDS databases, quản lý S3 buckets và xử lý toàn bộ hạ tầng AWS bằng code. Nói với Claude: *"Đây là AWS credentials của tôi. Khởi chạy EC2 instance t3.small ở us-east-1 và cấu hình security groups cho web app."*
- **Hetzner API Token** — vào Hetzner Cloud Console → project → Security → API Tokens → tạo một token. Claude sau đó có thể tạo servers, cấu hình firewalls, quản lý snapshots và xử lý toàn

bộ hạ tầng Hetzner. Nói với Claude: *"Đây là Hetzner API token của tôi. Tạo server CX22 ở Nuremberg với Ubuntu."*

Lưu ý bảo mật về API keys. Các API key này rất mạnh — hãy coi chúng như mật khẩu. **Đừng bao giờ commit vào Git**, đừng chia sẻ công khai, và đừng paste vào giao diện chat mà bạn không tin tưởng. Lưu chúng trong file `.env` hoặc truyền trực tiếp cho Claude trong phiên terminal. Nếu key bị lộ, thu hồi ngay từ dashboard nhà cung cấp và tạo key mới. Ngoài ra, khi tạo API tokens, **luôn áp dụng nguyên tắc đặc quyền tối thiểu**: chỉ cấp quyền thực sự cần thiết, không phải quyền admin đầy đủ.

Sự kết hợp các API key này với Claude cực kỳ mạnh mẽ. Bạn có thể nói: *"Tôi có ứng dụng Next.js. Deploy lên Cloudflare Pages, cấu hình custom domain và DNS — đây là API keys của tôi."* Và Claude sẽ làm mọi thứ tự động. Hoặc: *"Tạo EC2 instance trên AWS, cài Node.js và PM2, cấu hình nginx, cấu hình Cloudflare DNS và deploy ứng dụng."* Thiết lập hạ tầng hoàn chỉnh trong vài phút, không phải hàng giờ.

CI/CD Với GitHub Actions

Nhớ khi chúng ta thiết lập Git và GitHub CLI ở Chương 2 không? Đây là lúc mọi thứ phát huy tác dụng. Với `gh` đã xác thực, bạn có thể bảo Claude Code **đẩy dự án lên repository private trên GitHub**, thiết lập **GitHub Actions**, cấu hình tất cả **secret** cần thiết (SSH key của server, biến môi trường, v.v.), và tạo pipeline triển khai tự động — tất cả từ terminal, mà bạn không cần chạm vào giao diện GitHub.

Ý tưởng rất đơn giản: khi GitHub Actions được cấu hình, **mỗi lần Claude Code đẩy code lên repository, nó tự động triển khai lên server**. Không cần SSH thủ công, không sao chép file, không chạy lệnh trên server. Claude đẩy, GitHub Actions nhận, kết nối đến server AWS hoặc Hetzner qua SSH, và triển khai mọi thứ. Hoàn toàn tự động.

Chỉ cần nói với Claude: *"Đẩy dự án này lên repository private mới trên GitHub, thiết lập GitHub Action tự động triển khai lên server mỗi khi push lên main. Đây là IP server và SSH key."* Claude đã biết cách làm — nó sẽ tạo file workflow, thêm SSH key và IP server dưới dạng GitHub secret, và cấu hình toàn bộ pipeline. Đây chính xác là lý do chúng ta cài GitHub CLI trước đó.

Lưu ý về IP động. IP tĩnh thường tốn thêm phí trên cả AWS và Hetzner, nên bạn có thể sẽ dùng **IP động** để tiết kiệm. Điều này có nghĩa mỗi khi bạn dừng và khởi động lại server, IP thay đổi. Khi đó, bạn cần cập nhật ở **hai nơi**: bản ghi DNS Cloudflare và secret `SERVER_IP` trên GitHub. Bảo Claude lưu IP server dưới dạng secret `SERVER_IP` trong GitHub Actions, để khi nó thay đổi bạn chỉ cần cập nhật một biến. Cũng bảo Claude nhắc bạn kiểm tra và cập nhật IP nếu triển khai thất bại — IP thay đổi hầu như luôn là nguyên nhân.

5. Chiến Thuật Marketing & SEO

Sản phẩm của bạn đã lên mạng. Giờ thế giới cần biết nó tồn tại. Marketing hiệu quả nhất cho sản phẩm indie là tự nhiên — miễn phí, sáng tạo, và mạnh mẽ đáng ngạc nhiên khi làm đúng. Chương này bao gồm chính xác các chiến thuật chúng tôi sử dụng.

Chiến Lược Tăng Trưởng Tự Nhiên

Hãy thành thật: **marketing tốt nhất không giống marketing**. Internet tràn ngập quảng cáo, và mọi người đã phát triển phản xạ bỏ qua bất cứ thứ gì có cảm giác quảng bá. Cái thực sự hiệu quả là **marketing ngầm** — nội dung trông như thông tin chân thực, câu hỏi, hoặc gợi ý thay vì quảng cáo. Mọi người tự nhiên tò mò và thích giúp đỡ bằng các gợi ý. Hãy tận dụng điều đó.

Reddit là một trong những nền tảng mạnh nhất cho việc này. Nó rất lớn, được Google index cao, và đầy các cộng đồng ngách nơi đối tượng mục tiêu của bạn đã có mặt. Nhưng đây là điểm mấu chốt: **đừng bao giờ đăng marketing hung hăng**. Đừng viết *"Hãy xem trang web tuyệt vời mới của tôi!"* — bài đó sẽ bị downvote, xóa, hoặc bị cấm. Thay vào đó, viết thứ gì đó như:

- *"Ai có thể gợi ý trang tương tự [đối thủ nổi tiếng] hoặc [trang của bạn] không? Đang tìm các lựa chọn thay thế."*
- *"Có ai đã thử [danh mục sản phẩm của bạn] chưa? Tôi tìm thấy [trang của bạn] nhưng tò mò về các lựa chọn khác."*
- Trả lời câu hỏi trong các subreddit liên quan và đề cập sản phẩm một cách tự nhiên khi nó thực sự hữu ích.

Đây là cách phần lớn marketing indie thành công hoạt động trên Reddit. Bạn không nói dối — bạn đang đóng khung sản phẩm như một phát hiện trong cuộc trò chuyện chân thực. Mọi người nhấp vì tò mò, không phải vì cảm giác bị bán hàng. Và đây là bonus: **bài đăng Reddit được Google index**, nên một bài viết đặt đúng chỗ có thể mang lại lưu lượng tìm kiếm tự nhiên trong nhiều tháng hoặc thậm chí nhiều năm.

Bạn cũng có thể **tài trợ bài đăng Reddit** với ngân sách nhỏ để tăng tạm thời. Điều này đẩy bài lên cao hơn trong kết quả tìm kiếm, giúp Google index nhanh hơn, và tạo khả năng hiển thị ban đầu. Hỏi LLM cách quảng bá bài đăng Reddit hoạt động và ngân sách hợp lý.

Chiến thuật phễu. Trên trang chủ hoặc landing page, thêm thứ gì đó **thu hút mọi người độc lập với sản phẩm chính** — công cụ miễn phí, chủ đề đang hot, thông tin hữu ích, hoặc thứ gì đó đang phổ biến. Nó đóng vai trò **phễu**: mọi người đến vì nội dung miễn phí/thú vị, khám phá sản phẩm của bạn, và một phần trăm trong số họ chuyển đổi. Hãy coi nó như mỗi câu cung cấp giá trị thực sự đồng thời dẫn đến thứ bạn đang bán.

SEO, Nội Dung & Phân Phối

Trước khi bắt đầu bất kỳ hoạt động marketing nào, hãy thiết lập **phân tích và theo dõi**. Bạn cần hai thứ:

- **Google Analytics** — thêm mã theo dõi vào trang web (nhờ Claude tích hợp). Cũng có **ứng dụng di động** để bạn kiểm tra lưu lượng từ điện thoại bất cứ lúc nào. Công cụ này hiển thị tổng lượt truy cập, hành vi người dùng, nguồn lưu lượng, và dữ liệu chuyển đổi.
- **Google Search Console** — thiết lập và kết nối với Google Analytics. Search Console theo dõi cụ thể **lưu lượng tự nhiên từ Google**: truy vấn tìm kiếm nào đưa người đến trang web, tần suất bạn xuất hiện trong kết quả tìm kiếm, và tỷ lệ nhấp. Hỏi LLM cách thiết lập và kết nối cả hai công cụ.

Nếu có ngân sách, **Google Ads** có thể cho bạn cú hích ban đầu. Chạy quảng cáo ngắn hạn giúp xây dựng niềm tin với thuật toán Google và thu hút lưu lượng sớm trong khi SEO tự nhiên phát triển. Nhưng chi phí tích lũy nhanh, nên hãy coi nó như bộ tăng tốc ngắn hạn, không phải chiến lược dài hạn. LLM có thể giải thích chi tiết về thiết lập và ngân sách Google Ads.

Về SEO, hãy bắt đầu với những điều cơ bản. Mở **DevTools** trên trình duyệt (F12), vào **Lighthouse**, và tạo báo cáo. Công cụ này cho bạn điểm về Performance, Accessibility, Best Practices, và **SEO**. Sửa những gì nó yêu cầu — và vâng, bạn có thể nhờ Claude Code xử lý.

Các hành động SEO cần thực hiện:

- **Thêm bản dịch** cho các trang. Hỗ trợ đa ngôn ngữ tăng phạm vi tiếp cận đáng kể. Nhờ Claude Code thiết lập internationalization cho dự án.
- **Thêm meta tag đúng cách** — title, description, Open Graph tag cho chia sẻ mạng xã hội, structured data. Chúng ảnh hưởng trực tiếp đến cách trang web xuất hiện trong kết quả tìm kiếm Google.
- **Tạo và gửi sitemap**. Tạo sitemap cập nhật và upload lên Google Search Console. Điều này cho Google biết chính xác trang nào tồn tại trên trang web và giúp chúng được index nhanh hơn.

SEO cần kiên nhẫn. Đừng mong đợi lưu lượng tự nhiên qua đêm — cần hàng tuần hoặc hàng tháng để Google index và xếp hạng trang đúng cách. Nhưng khi nó khởi động, đó là **lưu lượng miễn phí liên tục** mà bạn không tốn một đồng. Những lượt nhấp mà bạn phải trả hàng trăm euro qua Google Ads sẽ đến miễn phí thông qua tìm kiếm tự nhiên. Trong thời gian chờ đợi, hãy nỗ lực trên các nền tảng xã hội như Reddit để xây dựng khả năng hiển thị ban đầu và backlink. SEO là cuộc chơi dài hạn, nhưng đây là khoản đầu tư marketing giá trị nhất bạn có thể thực hiện.

Xu hướng mới nổi: lưu lượng từ AI

Có một nguồn lưu lượng mới và đang phát triển nhanh mà hầu hết mọi người chưa chú ý: **LLM giới thiệu trang web của bạn**. ChatGPT, Gemini, Claude, và các trợ lý AI khác ngày càng được sử dụng như công cụ tìm kiếm. Khi ai đó hỏi "*Trang web nào tốt cho [danh mục sản phẩm của bạn]?*", các mô

hình này có thể và thực sự giới thiệu các website cụ thể — và điều đó tạo ra lưu lượng thực. Một phần đáng kể lượt truy cập của chúng tôi đến từ ChatGPT và các LLM khác.

Để tận dụng điều này, vào **bảng điều khiển Cloudflare** và đảm bảo bạn **tắt tùy chọn chặn AI crawler**. Nhiều trang web chặn bot AI theo mặc định, ngăn LLM tìm hiểu về nội dung của bạn. Bằng cách cho phép AI crawler truy cập trang web, bạn để các mô hình này index các trang, hiểu những gì bạn cung cấp, và có thể giới thiệu bạn cho người dùng trong tương lai. Đây về cơ bản là **SEO miễn phí cho kỷ nguyên AI** — và hiệu quả có thể rất lớn.

Hãy nghĩ xa hơn Google. SEO truyền thống nhắm vào tìm kiếm Google. Nhưng giới thiệu từ AI đang trở thành nguồn lưu lượng lớn — và xu hướng này chỉ đang tăng tốc. Đảm bảo trang web có thể truy cập được bởi AI crawler, có nội dung rõ ràng và mô tả, và có cấu trúc tốt để LLM dễ dàng hiểu những gì bạn cung cấp. Các trang web định vị mình ngay bây giờ cho khám phá bởi AI sẽ có lợi thế lớn khi kênh này phát triển.

Cảm Ơn Bạn!

Cảm ơn bạn đã mua khóa học này và tin tưởng giao thời gian và tiền bạc cho chúng tôi. Chúng tôi thực sự hy vọng bạn thấy nó có giá trị và nó giúp bạn xây dựng thứ gì đó thực sự.

Chúng tôi rất muốn nghe từ bạn. **Tham gia cộng đồng Discord** tại apifreellm.com — đó là nơi chúng tôi giao lưu, chia sẻ cập nhật, và giúp đỡ nhau.

Nếu bạn có chút thời gian, **hãy viết đánh giá** và cho chúng tôi biết bạn nghĩ gì. Bạn thấy phần nào hữu ích nhất? Còn thiếu gì? Có thể cải thiện gì? Chúng tôi thực sự quan tâm đến phản hồi của bạn — khóa học này là dự án đang phát triển và chúng tôi muốn tiếp tục cải thiện dựa trên những gì **bạn** thực sự cần.

[Hẹn gặp trên Discord. Giờ hãy đi xây dựng thứ gì đó tuyệt vời.](#)