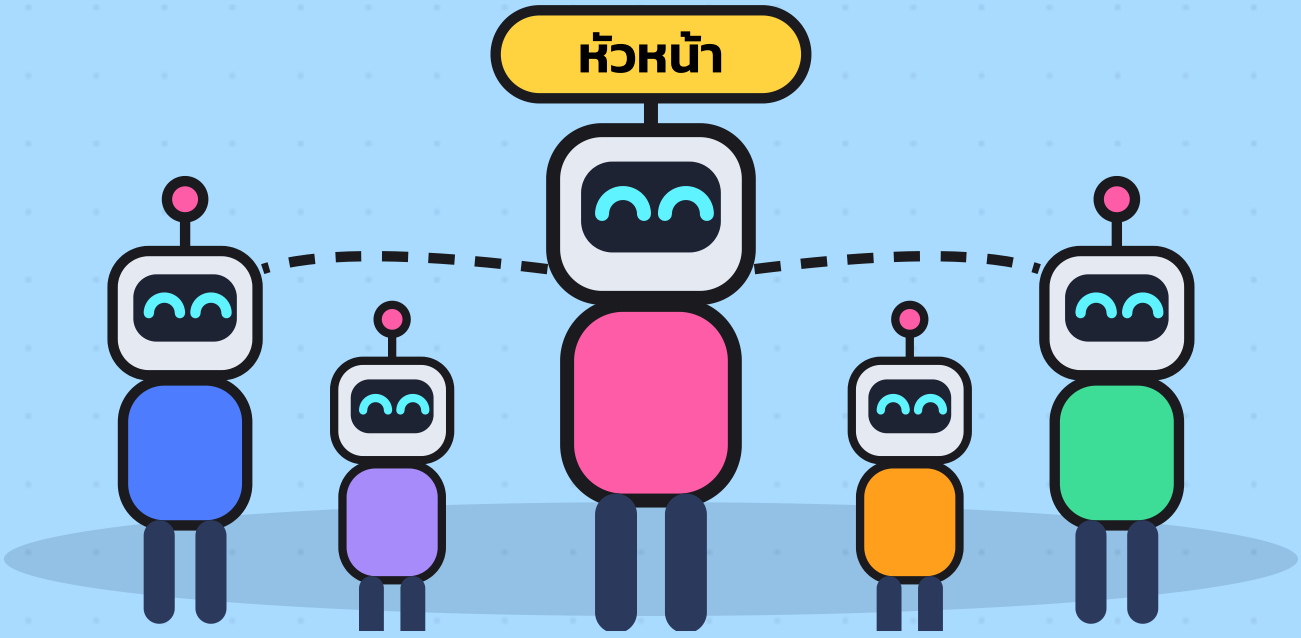
OpenCode

# สร้างทีม AI ให้ช่วยเขียนโค้ด

หัวหน้า 1 ตัว แจกงานให้ลูกทีมเฉพาะทาง: frontend · reviewer · tester
พร้อมไฟล์ตัวอย่าง ก๊อปไปใช้ได้ทันที



# ทีม AI ใน OpenCode คืออะไร?

OpenCode แบ่งเอเจนต์เป็น 2 แบบ: **Primary** คือตัวที่เราคุยด้วยตรงๆ และ **Subagent** คือผู้ช่วยเฉพาะทาง ที่ตัวหลักเรียกใช้ให้เอง

**สูตรทีม:** Primary 1 ตัวเป็น "หัวหน้า" + Subagent หลายตัวแบ่งหน้าที่กัน หัวหน้าจะดูจาก `description` ของลูกทีม แล้วเลือกแจกงานเอง

## เอเจนต์ที่มีมาให้อยู่แล้ว

| ชื่อ | ประเภท | ทำอะไร |
|---|---|---|
| Build | Primary | ตัวหลัก (ค่าเริ่มต้น) ใช้เครื่องมือได้ครบ แก้โค้ดได้ |
| Plan | Primary | วิเคราะห์/วางแผน ถูกจำกัดสิทธิ์ (ค่าเริ่มต้นไม่แก้ไฟล์) |
| General | Subagent | งานค้นหาซับซ้อน งานหลายขั้นตอน |
| Explore | Subagent | สำรวจโค้ดแบบเร็ว อ่านอย่างเดียว |
| Scout | Subagent | ค้นเอกสาร/ไลบรารีภายนอก อ่านอย่างเดียว |

**สลับตัวหลัก**
กดปุ่ม `Tab` เพื่อสลับระหว่าง Build ↔ Plan (หรือ primary ตัวอื่นที่สร้างเอง)

**เรียกลูกทีมเอง**
พิมพ์ `@ชื่อ` ในข้อความ เช่น `@reviewer ตรวจไฟล์ที่เพิ่งแก้`

# ขั้นที่ 1–2: เตรียมเครื่อง + สร้างลูกทีม

### 1 ติดตั้งและเปิดใช้

- ติดตั้งตามหน้า opencode.ai (หน้าเว็บตอนนี้แนะนำ `curl -fsSL https://opencode.ai/v2/install | bash`)
- เข้าโฟลเดอร์โปรเจกต์ แล้วพิมพ์ `opencode`
- เชื่อมโมเดลด้วย `/connect` แล้วพิมพ์ `/init` ให้มันอ่านโปรเจกต์และสร้างไฟล์ `AGENTS.md`

**ระวัง** ตอนนี้มีทั้ง OpenCode 1 และ 2 คำสั่งติดตั้งบางแบบใน docs เป็นของเวอร์ชัน 1 ถ้าเจอปัญหาให้เช็กเวอร์ชันกับหน้า opencode.ai/docs

### 2 สร้างลูกทีม: เลือก 1 จาก 3 วิธี

| วิธี | ทำยังไง |
|---|---|
| ก. ง่ายสุด | พิมพ์ `opencode agent create` ตอบคำถาม (เก็บที่ไหน / ทำอะไร / ให้สิทธิ์อะไร) แล้วมันเขียนไฟล์ให้เอง |
| ข. ไฟล์ Markdown | สร้างไฟล์ `.md` ในโฟลเดอร์ agents ชื่อไฟล์ = ชื่อเอเจนต์ (แนะนำ ดูตัวอย่างหน้าถัดไป) |
| ค. opencode.json | เขียนใต้หัวข้อ `"agent"` ในไฟล์ตั้งค่า |

## เก็บไฟล์ไว้ที่ไหน

โครงสร้างโฟลเดอร์

```
# เฉพาะโปรเจกต์นี้ (แนะนำ แชร์ให้ทีมผ่าน Git ได้)
my-project/
└ .opencode/
   └ agents/
      ├ frontend.md    → เอเจนต์ชื่อ frontend
      ├ reviewer.md    → เอเจนต์ชื่อ reviewer
      └ tester.md      → เอเจนต์ชื่อ tester

# ใช้ได้ทุกโปรเจกต์ในเครื่อง
~/.config/opencode/agents/
```

# ตั้งค่าลูกทีมได้อะไรบ้าง

## ค่าในหัวไฟล์ (frontmatter)

| ค่า | ความหมาย |
|---|---|
| `description` | **ต้องมี** บอกว่าทำอะไร + ใช้เมื่อไหร่ (หัวหน้าใช้ตัดสินใจแจกงาน) |
| `mode` | `primary` / `subagent` / `all` (ค่าเริ่มต้น `all`) |
| `model` | ใช้โมเดลแยกเฉพาะตัวนี้ รูปแบบ `provider/model-id` |
| `temperature` | ยิ่งต่ำยิ่งนิ่ง เหมาะกับงานตรวจ/เทสต์ |
| `steps` | จำกัดจำนวนรอบที่ทำงานได้ |
| `permission` | สิทธิ์ เช่น `edit`, `shell`, `webfetch`, `task` ตั้งเป็น `allow` / `ask` / `deny` |

## สิทธิ์ (permission) ที่ตั้งได้

ค่าที่ใส่ได้: `allow` ทำเลย · `ask` ถามเราก่อน · `deny` ห้าม

| สิทธิ์ | คุมอะไร |
|---|---|
| `edit` | เขียน/แก้ไฟล์ (รวมการ patch) |
| `shell` | รันคำสั่งใน terminal ตั้งรายคำสั่งได้ เช่น `"git status *": "allow"` |
| `read` `glob` `grep` `list` | อ่าน/ค้นไฟล์ในโปรเจกต์ |
| `task` | เรียก subagent ตัวอื่น (ตั้งตามชื่อได้) |
| `webfetch` `websearch` | ดึงเว็บ / ค้นเว็บ |
| `external_directory` | แตะไฟล์นอกโฟลเดอร์โปรเจกต์ |

**ตัวอย่าง: ให้รันได้แค่ git status ที่เหลือถามก่อน**

```yaml
permission:
  shell:
    "*": ask
    "git status *": allow
```

ตั้งกฎกว้าง (`*`) ไว้ก่อน แล้วค่อยตามด้วยกฎเฉพาะ เพราะกฎที่ตรงตัวหลังสุดจะชนะ

# ขั้นที่ 3: ไฟล์ตัวอย่าง ก๊อปไปใช้ได้เลย

วางทั้ง 3 ไฟล์ไว้ใน `.opencode/agents/` ของโปรเจกต์ แก้ข้อความให้ตรงกับงานของคุณ

.opencode/agents/frontend.md

```
---
description: ทำงานฝั่งหน้าเว็บ (React, CSS, UI) ใช้เมื่องานเกี่ยวกับหน้าจอหรือคอมโพเนนต์
mode: subagent
---
คุณคือ frontend developer ดูแลเฉพาะไฟล์ฝั่ง UI
- ทำตามสไตล์และคอมโพเนนต์ที่โปรเจกต์ใช้อยู่แล้ว
- อธิบายสั้นๆ ว่าแก้ไฟล์ไหนบ้าง
```

.opencode/agents/reviewer.md

```
---
description: รีวิวโค้ดหาบั๊กและจุดเสี่ยง ใช้หลังแก้โค้ดเสร็จทุกครั้ง
mode: subagent
temperature: 0.1
permission:
  edit: deny
  shell: deny
---
คุณคือ code reviewer ห้ามแก้ไฟล์เอง
- รายงานปัญหาพร้อมชื่อไฟล์และบรรทัด
- แยกระดับ: ต้องแก้ / ควรแก้ / แนะนำ
```

.opencode/agents/tester.md

```
---
description: เขียนและรันเทสต์ ใช้เมื่อต้องยืนยันว่าโค้ดทำงานถูก หรือหลังเพิ่มฟีเจอร์
mode: subagent
permission:
  shell: ask
---
คุณคือ tester เขียนเทสต์ให้ครอบคลุมกรณีปกติและกรณีผิดพลาด
แล้วรันเทสต์จนผ่าน ถ้าไม่ผ่านให้บอกสาเหตุ
```

**อยากให้แต่ละตัวใช้โมเดลต่างกัน?** เพิ่มบรรทัด `model: provider/model-id` เช่น ให้ reviewer ใช้โมเดลที่ถูกกว่าเพื่อประหยัดค่าใช้จ่าย (เปลี่ยนเป็นชื่อโมเดลที่คุณเชื่อมไว้จริง)

# ขั้นที่ 4: สั่งงานทีม

### A ปล่อยให้หัวหน้าแจกงานเอง

อยู่ที่ Build แล้วสั่งเป็นภาษาคน เช่น:

```
เพิ่มหน้า login พร้อมฟอร์มอีเมล/รหัสผ่าน
เสร็จแล้วให้ reviewer ตรวจ และให้ tester เขียนเทสต์ด้วย
```

หัวหน้าจะเลือกเรียกลูกทีมจาก `description` ยิ่งเขียนว่า "ใช้เมื่อ..." ชัด ยิ่งแจกงานแม่น

### B เรียกลูกทีมเองตรงๆ

```
@reviewer ตรวจไฟล์ที่เพิ่งแก้ในโฟลเดอร์ src/auth
@tester เขียนเทสต์ให้ฟังก์ชัน login
```

### C วางแผนก่อนลงมือ

กด `Tab` ไปที่ Plan ให้มันวางแผนก่อน (ไม่แตะไฟล์) พอแผนโอเคค่อยกด `Tab` กลับไป Build ให้ทีมลงมือ

## คุมว่าหัวหน้าเรียกใครได้บ้าง (ขั้นสูง)

ใน `opencode.json` ตั้งสิทธิ์ `task` ด้วยรูปแบบชื่อ (glob) กฎที่ตรงตัว**หลังสุด**ชนะ

**ตัวอย่าง permission.task**

```jsonc
"task": {
  "*": "deny",            // ปิดทุกตัวก่อน
  "frontend": "allow",
  "tester": "allow",
  "reviewer": "ask"       // ถามเราก่อนเรียก
}
```

# เคล็ดลับ & ข้อควรระวัง

**เขียน description ให้คม**
บอกทั้ง "ทำอะไร" และ "ใช้เมื่อไหร่" เช่น *"รีวิวโค้ด ใช้หลังแก้โค้ดเสร็จทุกครั้ง"*

**ให้สิทธิ์เท่าที่จำเป็น**
ตัวที่แค่ตรวจหรือค้น ให้ `edit: deny` ตัวที่รันคำสั่ง ให้ `shell: ask` ไว้ก่อน

**แบ่งโมเดลประหยัดเงิน**
งานคิดหนักใช้โมเดลเก่ง งานตรวจ/สรุปใช้โมเดลถูก ตั้ง `model` แยกรายตัวได้

**แชร์ทีมให้เพื่อน**
เก็บใน `.opencode/agents/` แล้ว commit ขึ้น Git ทุกคนในโปรเจกต์ได้ทีมเดียวกัน

**! ข้อควรรู้**

- **Desktop app:** หน้า docs เรื่องเอเจนต์อธิบายผ่านไฟล์ตั้งค่าและ terminal ไม่ได้ระบุเรื่องแอปเดสก์ท็อปตรงๆ ถ้าใช้แอป ลองสร้างไฟล์ใน `.opencode/agents/` แล้วเปิดโปรเจกต์ในแอป จากนั้นพิมพ์ `@` ดูว่าลูกทีมขึ้นไหม
- **ชื่อโฟลเดอร์:** docs ปัจจุบันใช้ `agents/` (มี s) ถ้าเวอร์ชันของคุณไม่เจอ ให้เช็ก docs ของเวอร์ชันที่ใช้
- **ค่าใช้จ่าย:** ตัวโปรแกรมฟรี (MIT) แต่ทุกเอเจนต์เรียกโมเดลจริง ทีมใหญ่ = ใช้โทเคนมากขึ้น
- **ความปลอดภัย:** อย่าให้เอเจนต์ที่ไม่จำเป็นได้สิทธิ์ `shell` แบบ `allow`

**ทำตามแล้วติดตรงไหน คอมเมนต์ถามได้เลย**
ติดตาม **keaykub.dev** · เทคยากๆ เล่าให้ง่ายใน 1 นาที

**อ้างอิง:** opencode.ai/docs/agents · opencode.ai/docs · github.com/sst/opencode (ข้อมูล ณ ต.ค. 2026 ฟีเจอร์อาจเปลี่ยนตามเวอร์ชัน)