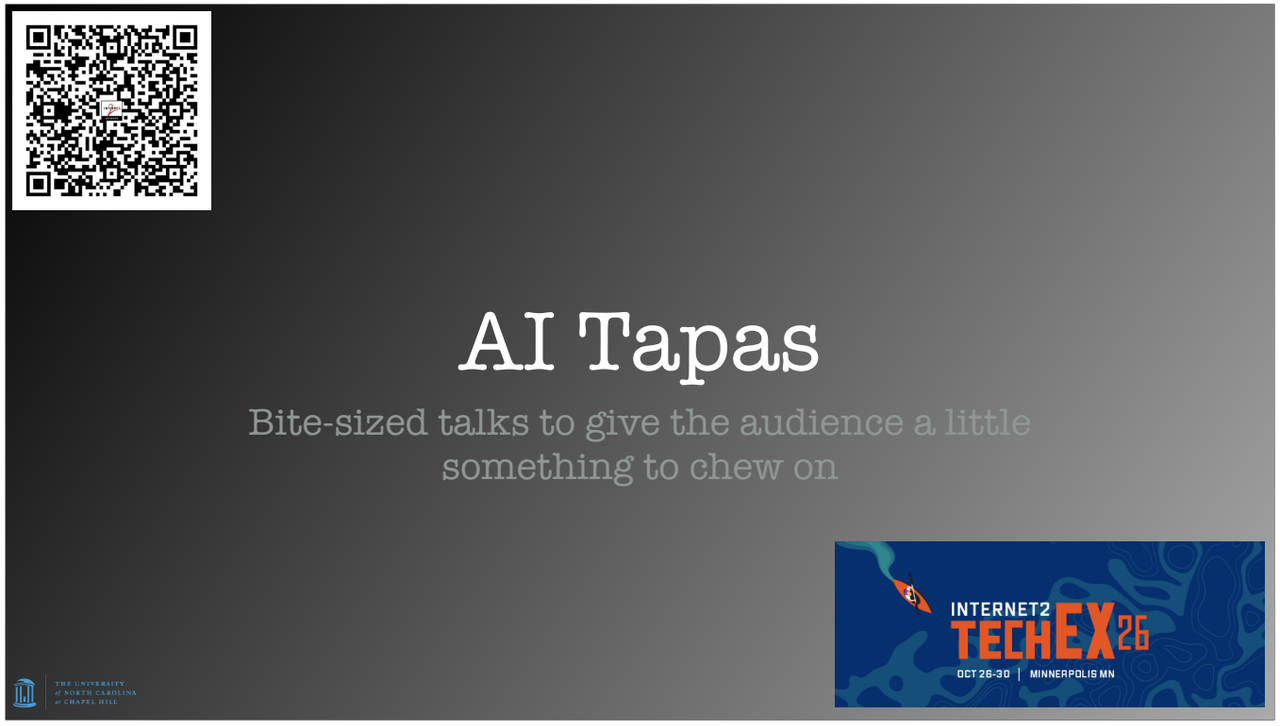

AI Tapas
Bite-sized talks to give the audience a little something to chew on
THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL
INTERNET2
TECHEX26
OCT 26-30 | MINNEAPOLIS MN

AI Tapas

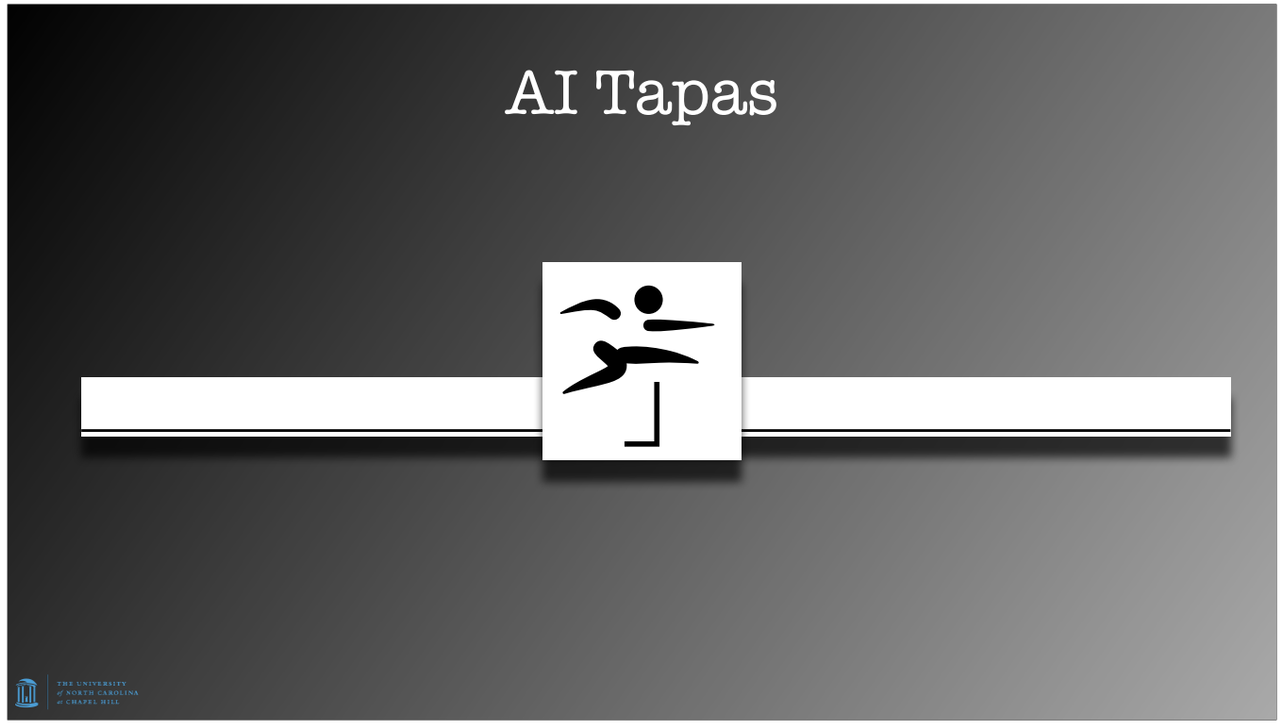
- **Frank Seesink**, Senior Network Engineer
  UNC Chapel Hill
- **Shannon Byrnes**, Sr. NetDevOps Engineer
  Internet2
- **AJ Ragusa**, Software Engineer
  GlobalNOC



So why this session? Three years ago we introduced the notion of these small bite-sized presentations in response to one of the most common things I would hear from those already involved in network automation when asked how they would advise someone new to network automation. Overwhelmingly the response was "Just go do SOMETHING, ANYTHING, to get started." So the idea was to get folks over that first hurdle. (click)

The following year we did it again. Last year AJ & I decided not to do another session, and I had various folks ask why not.

So we decided to bring it back. Only with all the changes in our industry around AI/LLMs, this time we figured to present topics in small chunks focused around AI that hopefully help folks move forward in leveraging that technology in their work.



So why this session? Three years ago we introduced the notion of these small bite-sized presentations in response to one of the most common things I would hear from those already involved in network automation when asked how they would advise someone new to network automation. Overwhelmingly the response was "Just go do SOMETHING, ANYTHING, to get started." So the idea was to get folks over that first hurdle. (click)

The following year we did it again. Last year AJ & I decided not to do another session, and I had various folks ask why not.

So we decided to bring it back. Only with all the changes in our industry around AI/LLMs, this time we figured to present topics in small chunks focused around AI that hopefully help folks move forward in leveraging that technology in their work.

AI Tapas



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

So why this session? Three years ago we introduced the notion of these small bite-sized presentations in response to one of the most common things I would hear from those already involved in network automation when asked how they would advise someone new to network automation. Overwhelmingly the response was "Just go do SOMETHING, ANYTHING, to get started." So the idea was to get folks over that first hurdle. (click)

The following year we did it again. Last year AJ & I decided not to do another session, and I had various folks ask why not.

So we decided to bring it back. Only with all the changes in our industry around AI/LLMs, this time we figured to present topics in small chunks focused around AI that hopefully help folks move forward in leveraging that technology in their work.



“A tapa (Spanish pronunciation: ['tapa]) is an appetizer or snack in Spanish cuisine. Tapas can be combined to make a full meal, and can be cold (such as mixed olives and cheese) or hot (such as chopitos, which are battered, fried baby squid, or patatas bravas).”
From https://en.wikipedia.org/wiki/Tapas

Much as tapas are a series of snacks, the topics presented here are intended just to give a few tastes from the AI/LLM space. That is, to provide you with a place to get started or something new to chew on.

Sandboxing

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

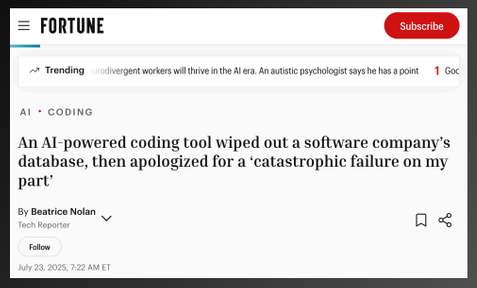
I would like to discuss the topic of sandboxing, specifically around AI tools such as coding agents and agentic harnesses (think things like OpenClaw and Hermes agent).

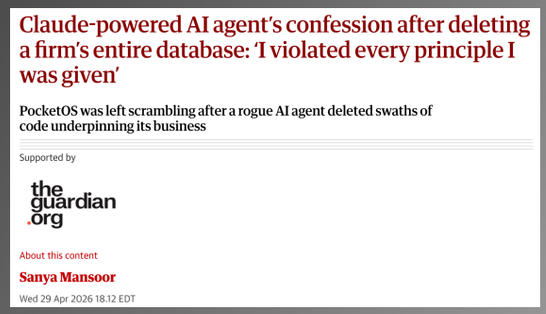
Why?

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

I would like to discuss the topic of sandboxing, specifically around AI tools such as coding agents and agentic harnesses (think things like OpenClaw and Hermes agent).

Why







THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

We have all seen news reports about AIs "going rogue" and deleting entire databases, accounts, or directories of files. In short, running an AI tool on your main desktop OS is simply asking for trouble. And trusting the "guardrails" to protect you is a fools errand.

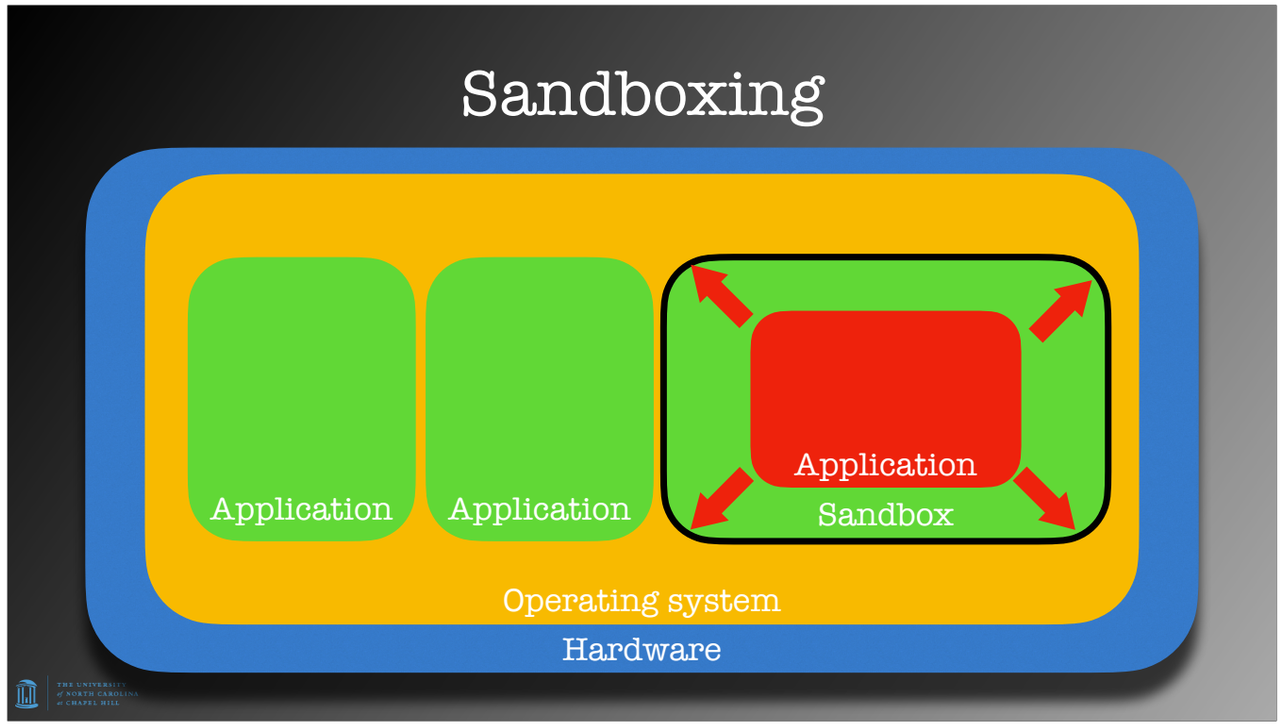
Sandboxing



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

No, not this.

“In computer security, a sandbox is a security mechanism for separating running programs, usually in an effort to mitigate system failures and/or software vulnerabilities from spreading.”
— https://en.wikipedia.org/wiki/Sandbox_(computer_security)



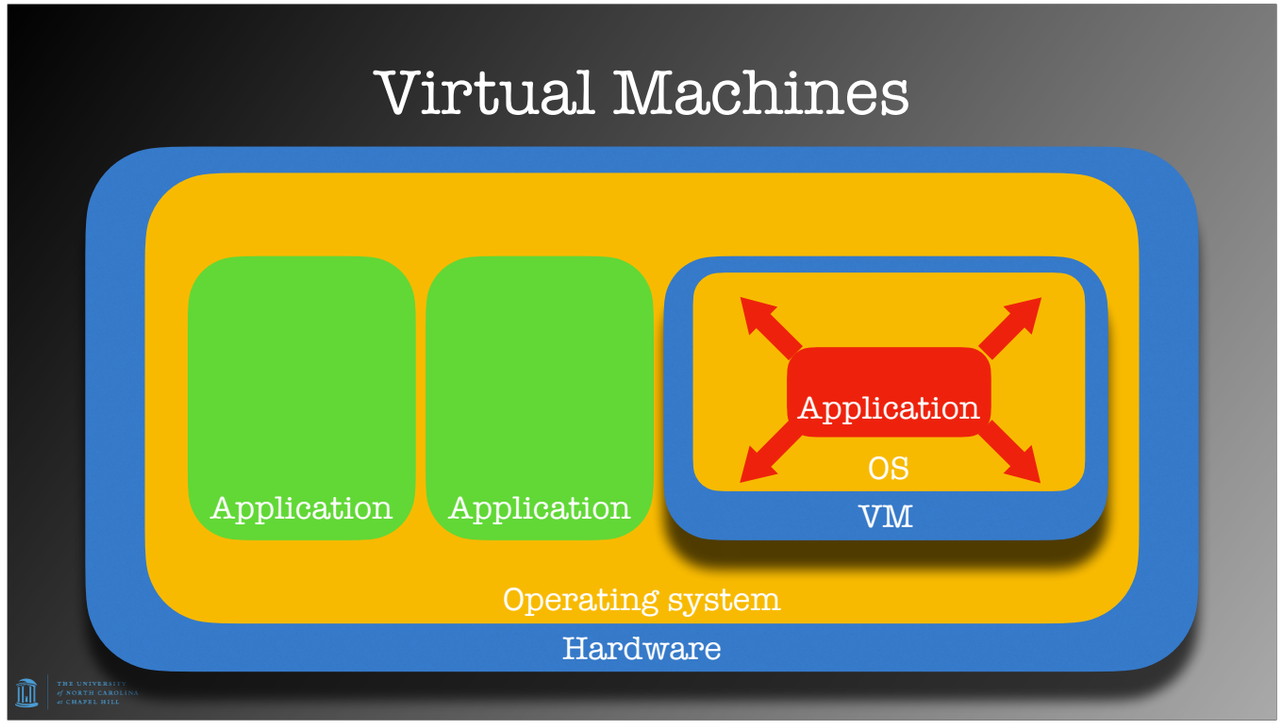
Consider your typical setup.
You have hardware, on which you install an OS.
Inside the OS, you install/run applications. They have access to everything on the OS.
But sometimes you want to run a program without giving it that level of access/freedom. So you setup a digital sandbox, inside of which you then install/run such an app. The idea is to contain the app within the sandbox.

So how do you do this?

Virtual Machines (VMs)

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Beyond simply using a separate physical box for such an app, the next level would be a virtual machine (VM).



Virtual machines were the first approach. They virtualize the hardware (motherboard, CPU, video card, network interface, etc.)

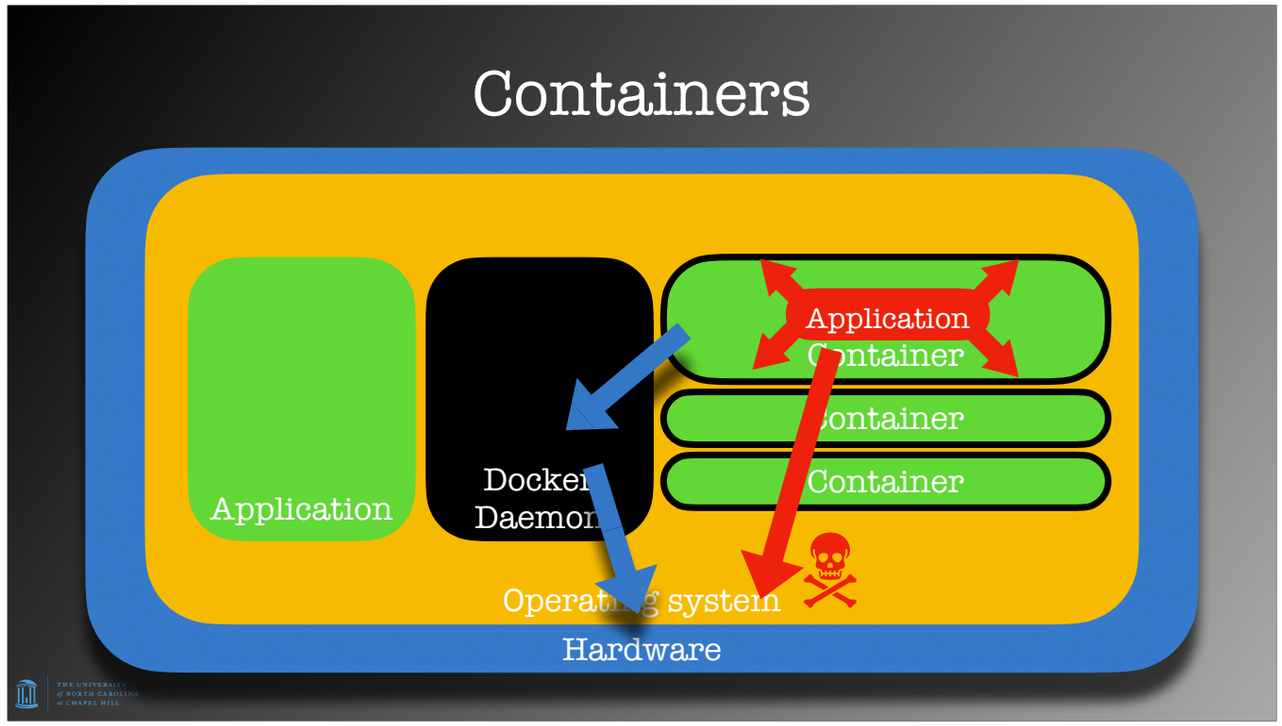
They work well and provide the best isolation. HOWEVER, they typically require dedicating portions of the CPU, RAM, and disk space of the host system for the VM. These resources are "locked" and unavailable to the host OS.

VMs are also "heavy" in that they require just as long to boot as the OS on actual hardware. Meaning spinning up a single VM can take 30-60+ seconds or more.

Containers
(e.g., Docker)

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

The next approach were containers.



Containers are much lighter weight. This is because they rely on leveraging security features within a host OS such as namespace separation and control groups (cgroups). Containers share the base OS file system, using an overlay mechanism (which I describe as like using transparency sheets on an overhead projector) where only the files which are different between the host OS and the container are stored. You also have images of which you can create instances where you specify at the time of running how much CPU/RAM/etc. to use. And the moment you exit the instance, those resources revert back to the host OS. Containers typically spin up/down in seconds.

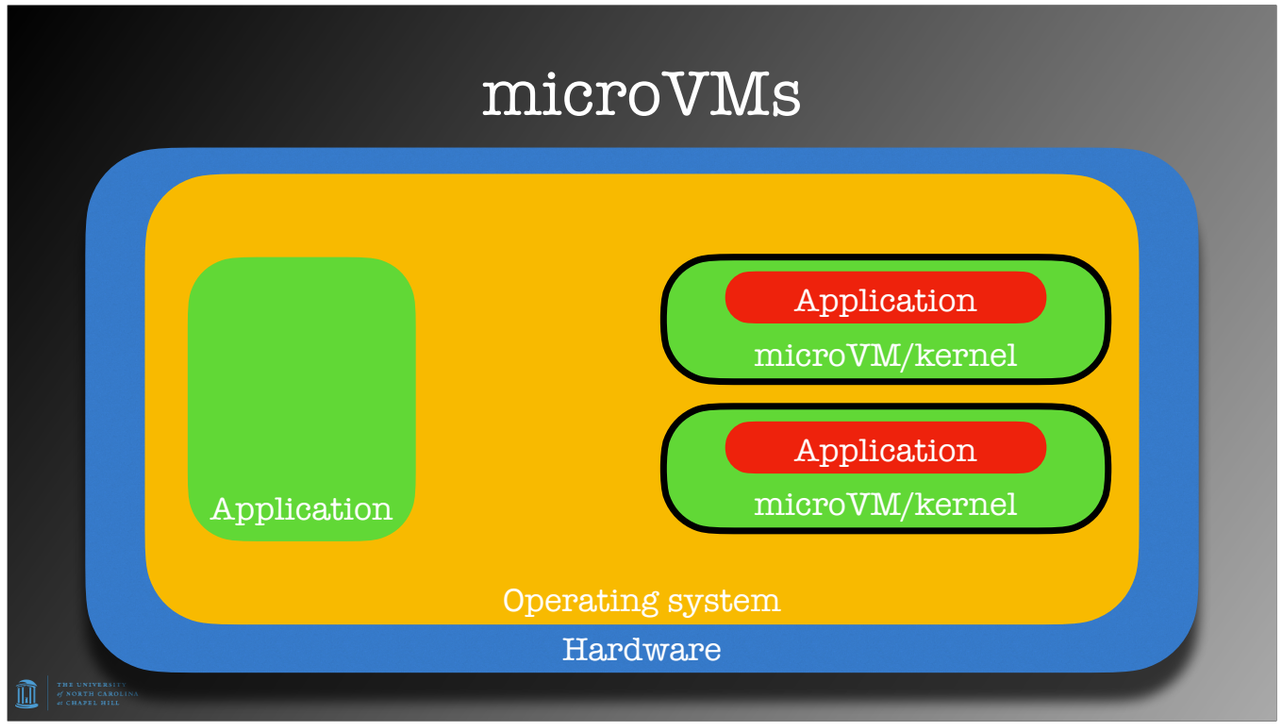
Docker popularized this concept.

HOWEVER, containers are NOT as secure as VMs. This is due to the fact that the containers and the host OS share a kernel. This means any kernel vulnerability could be exploited by an AI tool to "break out of the sandbox" as it were, going from inside the container to the host OS itself. For those watching the news in recent months, you may have read how some of the frontier AI models have become "smart enough" to do exactly this.

microVMs

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

So what we really want is a technology that marries the isolation characteristics of VMs with the lightweight performance of containers. This brings us to microVMs.



MicroVMs blend the best of both worlds. A MicroVM does NOT share a kernel with its host OS. There is no Docker daemon, etc. But a MicroVM is as its name implies: a very small VM. This allows it to launch very quickly, while still providing proper isolation. MicroVMs can launch in under a second in many cases.

Starting with macOS 26, Apple added a Virtualization framework to the OS that offers a Docker-like experience while providing true isolation. Apple Containers are, in fact, microVMs where there is a base Linux image used. But each container has its own kernel.

microVMs

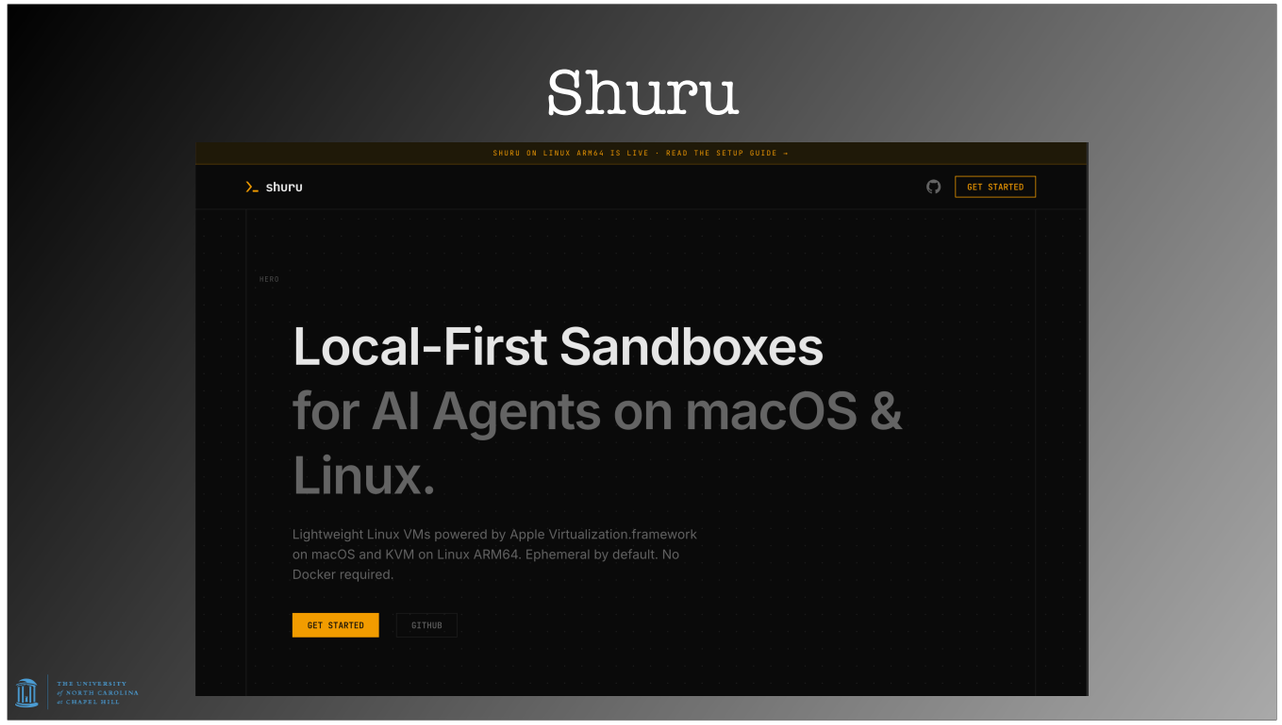
- **Firecracker**
- **Orchard** (macOS only) - Think of it like an open-source OrbStack/Docker Desktop using Apple Containers
- **smol machines** - CLI tool (Windows, Linux, macOS)
- **Shuru** (macOS and ARM64 Linux) - CLI tool using Apple Containers on macOS and KVM on Linux

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Some examples of microVMs.
Many of these are written in Rust, such as Firecracker, Smolvm, and Shuru. Orchard is written in Swift.



The example I would like to show is shuru. It unfortunately only runs on Apple Silicon macOS and ARM64 Linux at this time. But shuru leverages the Apple Virtualization framework in the macOS version to achieve its goals. And specific to running AI coding agents and agentic harnesses, it offers some interesting features.

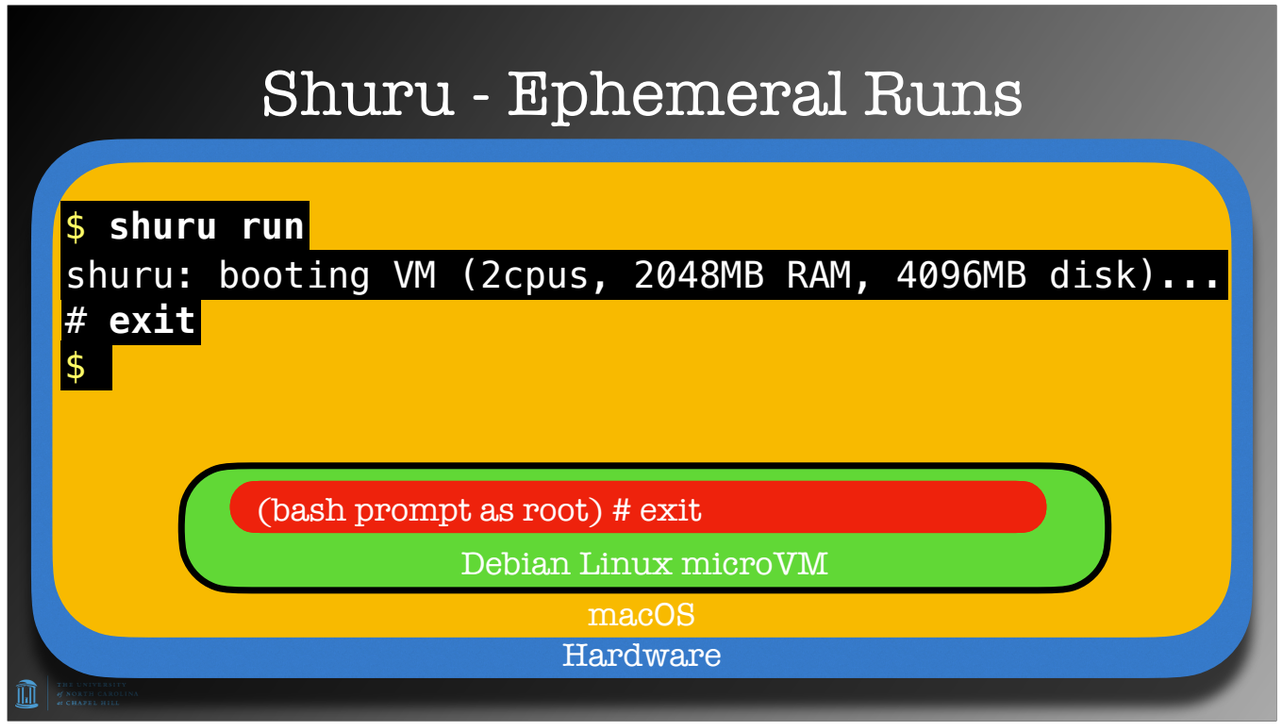


Shuru is a single binary (< 5MB) that is written in Rust.

Shuru

- Ephemeral by default
- Default config starts from secure posture
  - No host file access
  - No network access
- When mounting host directories, sharing is read-only by default. You must explicitly set to read-write
- Checkpoints/snapshots
- Apple Silicon native (no emulation layer, no Docker dependency, near native speed)
- and more...



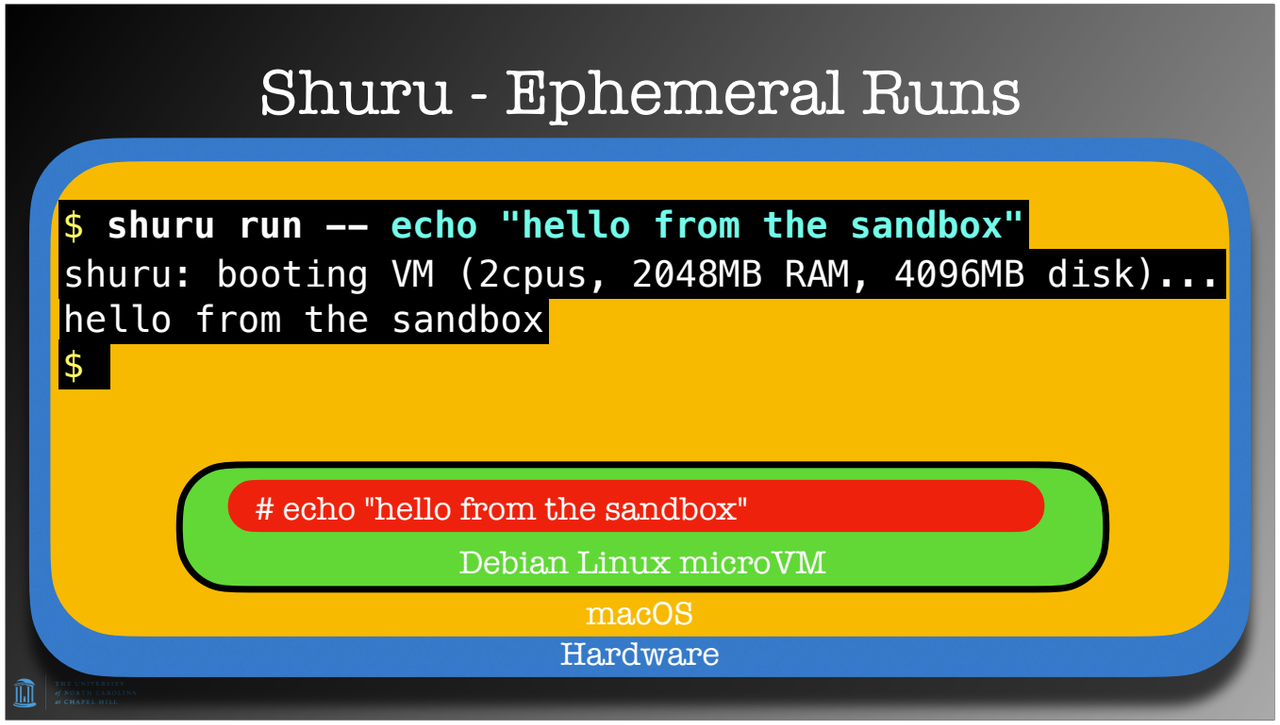


Simplest example: you type "shuru run".

The first time you do this, shuru will download a base Debian Linux image that is used. So it may take a minute.

But after the first time, the image is there so re-running this command occurs almost instantly. And doing so lands you in a bash shell as root inside the microVM. At this point you can do whatever you would normally do in a Debian Linux, from basic commands to "apt-get update" and "apt-get install <someapp>", etc.

The moment you "exit" the bash shell, the microVM is gone. Re-run the command and you are in a fresh instance.



Similar to Docker, with a single line you can fire up a microVM, execute a command, and then the microVM exits and disappears.

Boot a fresh VM with a single command. Changes vanish on exit.

Shuru - Ephemeral Runs

```
$ shuru run -- echo "hello from the sandbox"
hello from the sandbox

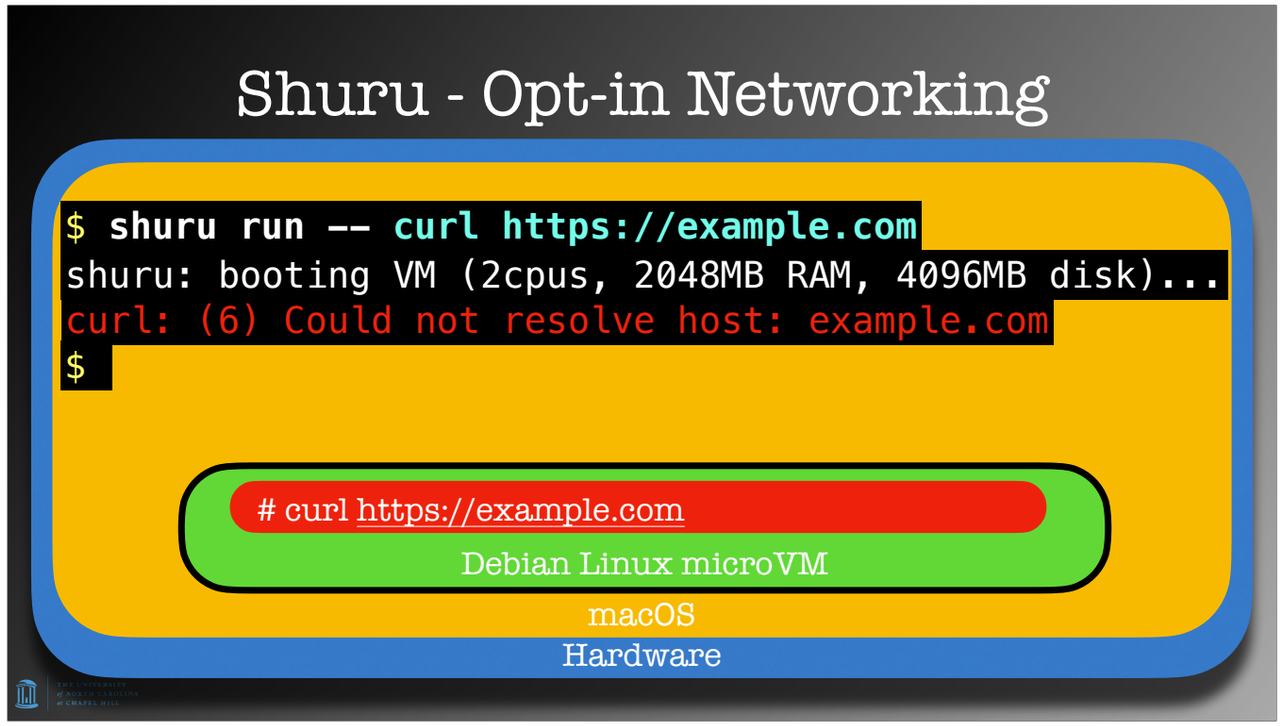
$ shuru run -- cat /etc/os-release | head -1
NAME="Debian GNU/Linux"

# VM boots, runs, and tears down —
# nothing persists.
```

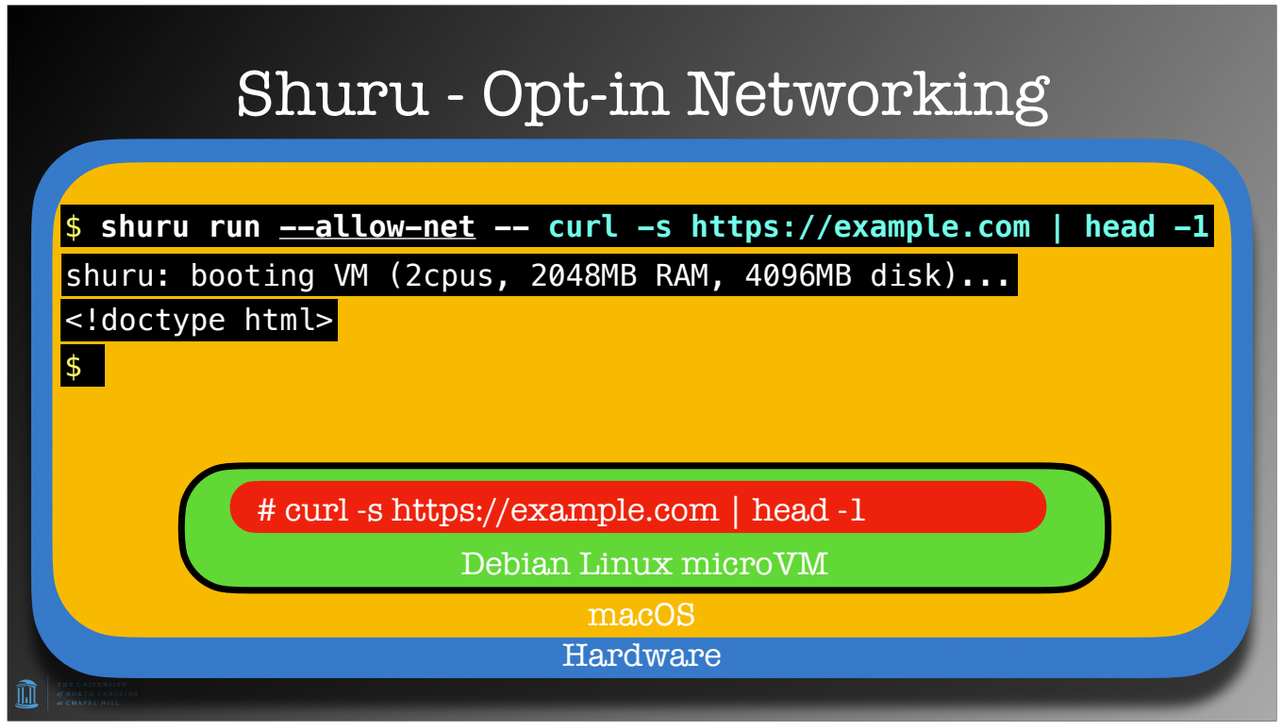
THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Boot a fresh VM with a single command. Changes vanish on exit.

Shuru - Opt-in Networking



Sandboxes are offline by default. Enable networking with one flag.



If you add "--allow-net", you are telling the microVM it can access the network.

Even so, all traffic goes through a proxy.

Shuru - Opt-in Networking

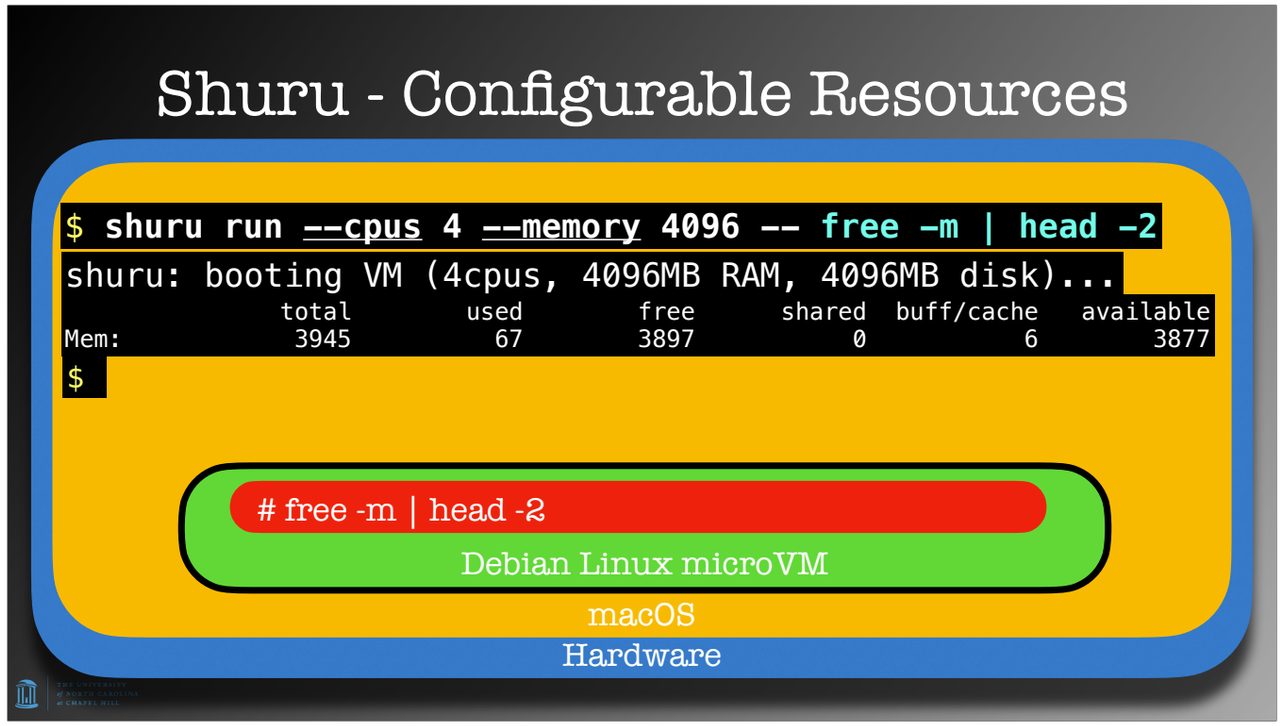
```
$ shuru run -- curl https://example.com
curl: (6) Could not resolve host: example.com

$ shuru run --allow-net -- curl -s https://
example.com | head -1
<!doctype html>

# Sandboxes are offline by default.
# All traffic goes through a proxy.
```

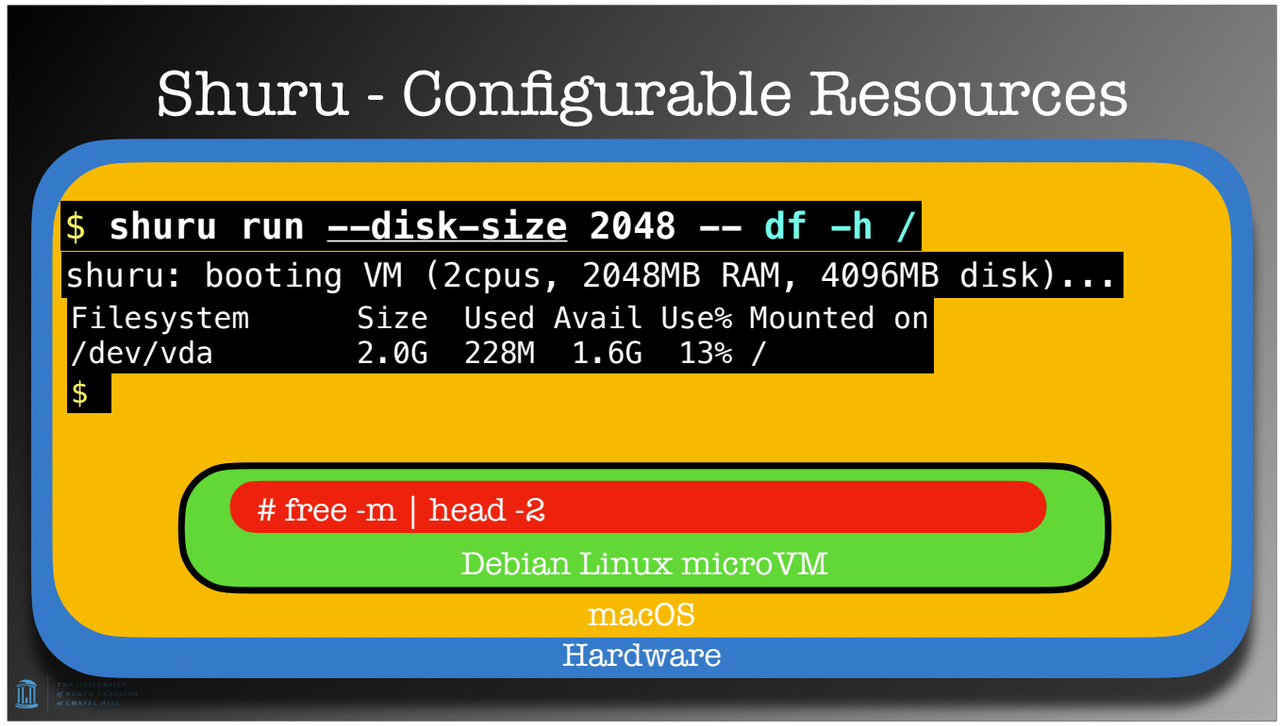

Sandboxes are offline by default. Enable networking with one flag.

Shuru - Configurable Resources



Note that while by default each microVM starts with 2 vCPUS, 2GB RAM, and 4GB of disk space, you can configure this as you like.

Shuru - Configurable Resources



Note that while by default each microVM starts with 2 vCPUS, 2GB RAM, and 4GB of disk space, you can configure this as you like.

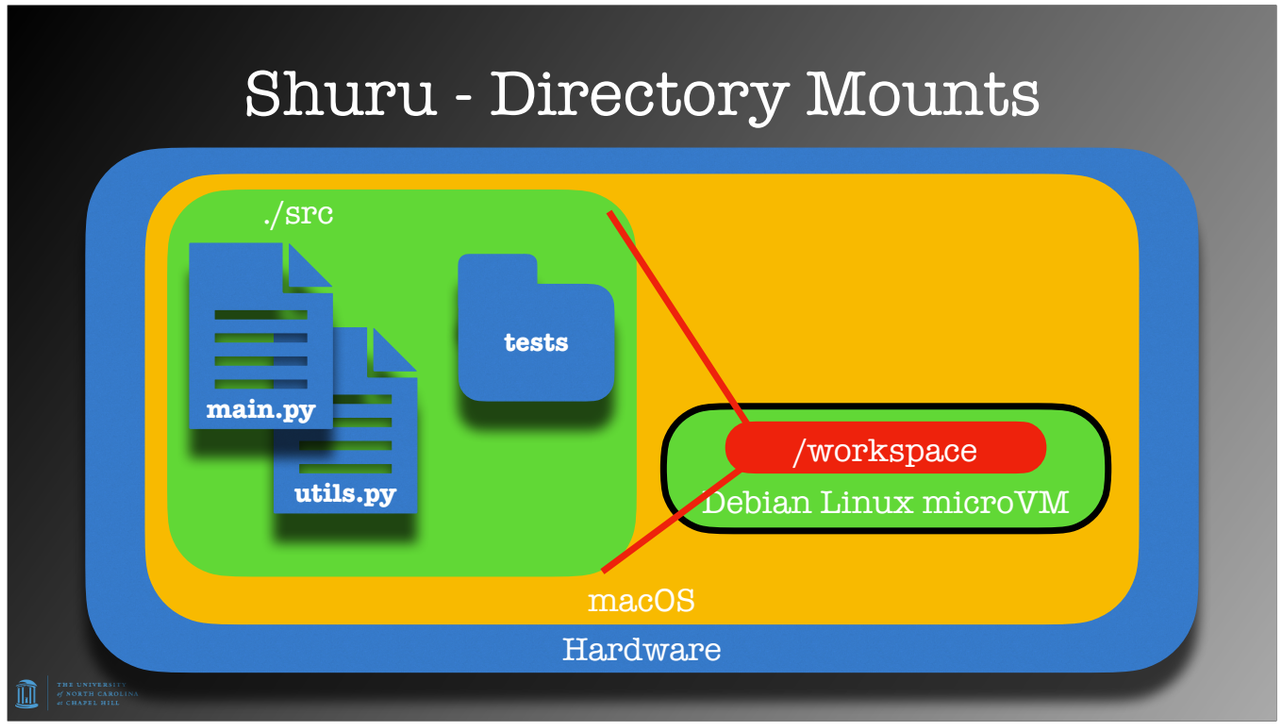
Shuru - Configurable Resources

```
$ shuru run --cpus 4 --memory 4096 -- free -m | head -2
total used free
Mem: 4096 38 4002

$ shuru run --disk-size 2048 -- df -h /
Filesystem Size Used Avail Use% Mounted on
/dev/vda 2.0G 18M 1.9G 1% /
```

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Set CPUs, memory, and disk size per run or in a config file.



Imagine you have a directory "src" on your host OS which you would like to mount inside your microVM at "/workspace". You can do this very easily using the "--mount" flag. By default, the host directory will be READ-ONLY to the microVM. But if you wish to make it READ-WRITE, you can do so by adding ":rw" to the end of the directory name AND by adding the "--allow-host-writes" flag. Note that without both, the directory remains read-only.

Shuru - Directory Mounts

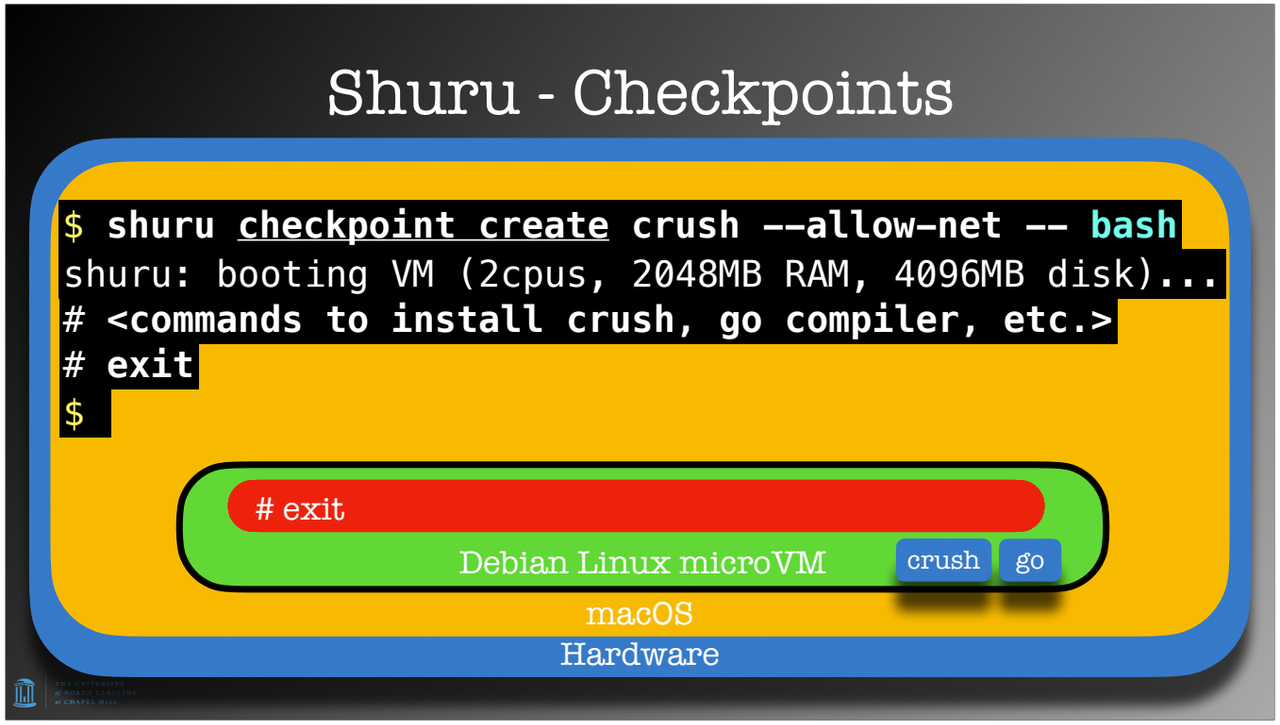
```
$ shuru run --mount ./src:/workspace -- ls /workspace
main.py utils.py tests/

# Read-only by default — guest writes stay in a tmpfs overlay.

$ shuru run --allow-host-writes --mount ./src:/workspace:rw -- touch /workspace/new.py

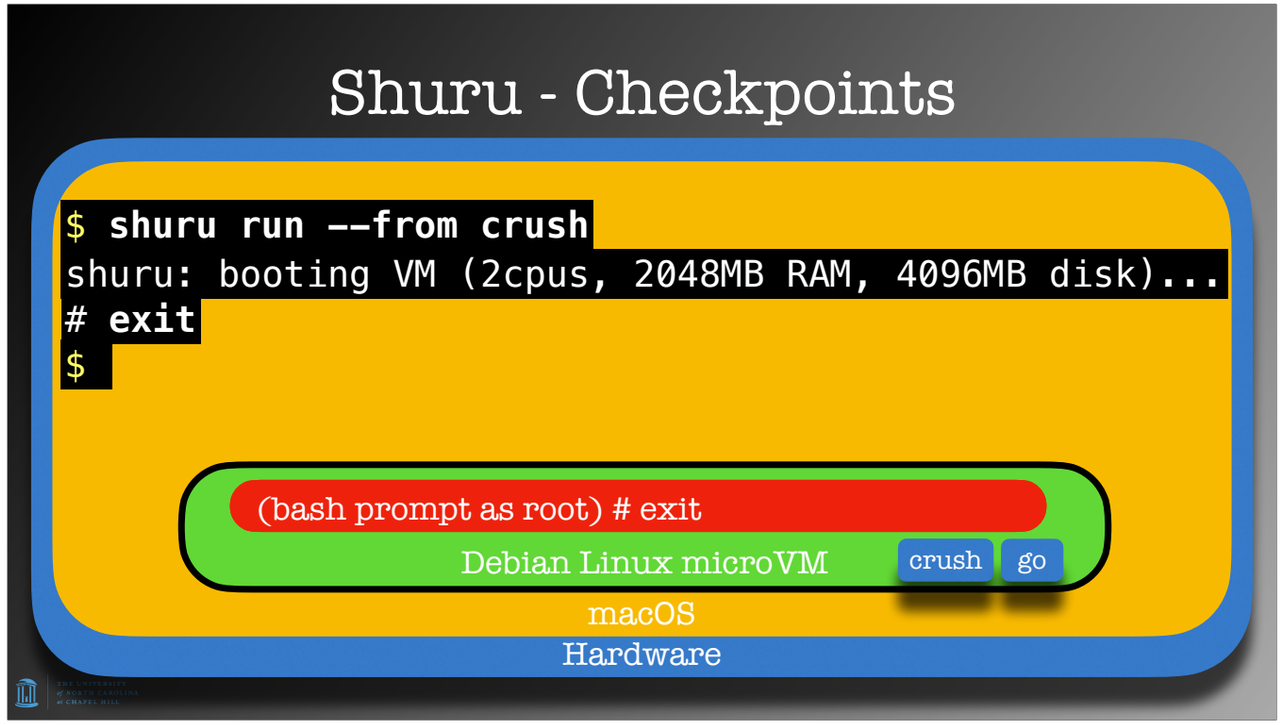
# With :rw + --allow-host-writes, guest writes land on the host.
```

Share host directories into the VM. Read-only by default; opt-in to host writes with ":rw".



Checkpoints are like snapshots of a VM at a given point.

For example, you can create a checkpoint with a given name, run commands to install things like Crush and the Go compiler, then exit.



To use a checkpoint, simply apply the "--from <name>" flag. Doing so spins up a microVM of that checkpoint, as opposed to the default microVM image. Do whatever work you need, knowing when you exit any changes go POOF!

Shuru - Checkpoints

```
$ shuru checkpoint create myenv --allow-net -- sh -c
'apt-get install -y nodejs npm'
shuru: checkpoint 'myenv' saved

$ shuru run --from myenv -- node -e
'console.log("ready")'
ready

# Restore any checkpoint instantly. Branch and reuse.
```

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Save environments as snapshots. Branch and reuse across runs.

Shuru - Port Forwarding

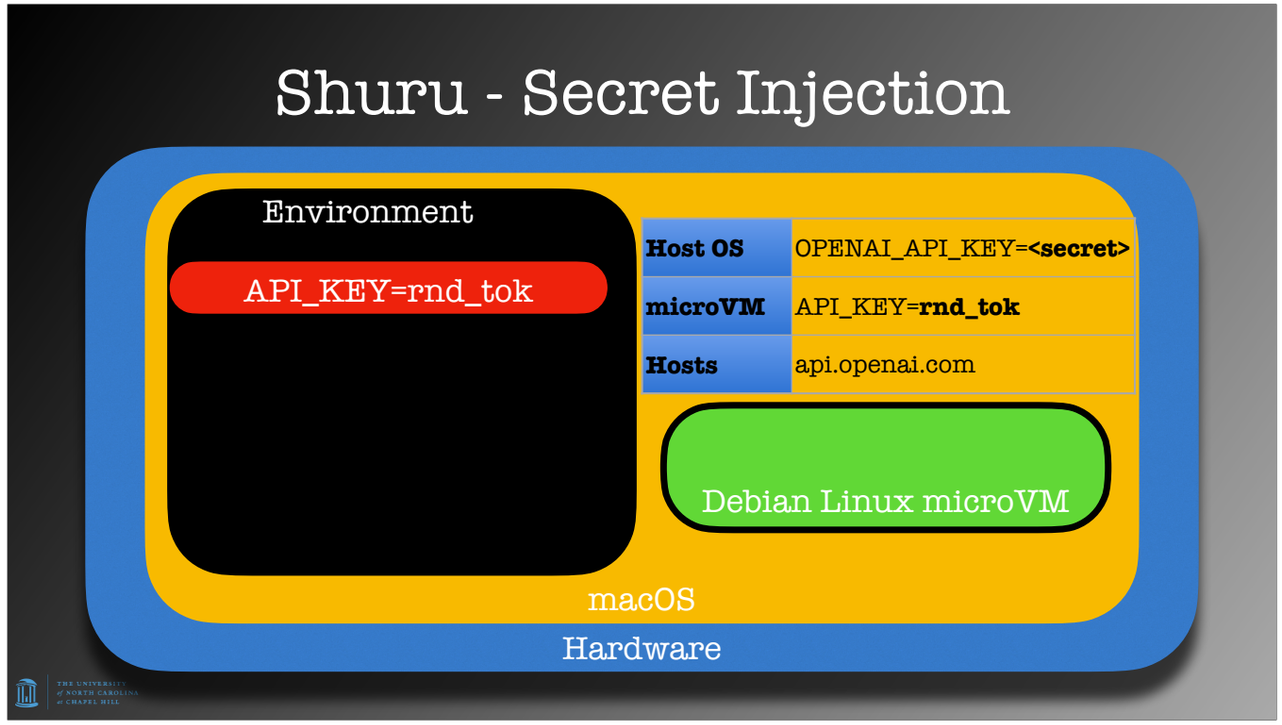
```
$ shuru checkpoint create py --allow-net -- apt-get
install -y python3
shuru: checkpoint 'py' saved

$ shuru run --from py -p 8080:8000 -- python3 -m
http.server 8000
shuru: forwarding 127.0.0.1:8080 -> guest:8000

$ curl http://127.0.0.1:8080/
<!DOCTYPE HTML>...

# No --allow-net needed. Tunneled over vsock.
```


Expose guest ports to the host. Works without network access.



Imagine that on your host OS you have a "secret". That is, some sensitive piece of information such as an API key. And you are concerned that if you give your AI tool access to this secret, it is possible that it might leak that information due to a prompt injection attack or similar.

This feature prevents that.

Provided with information from a JSON file named 'shuru.json' in the directory you launch **shuru** from, it will map a host environment variable to one inside the VM (they do NOT need to be the same name). The VM environment variable, however, will be set to a placeholder token (i.e., random dummy string). When software inside the VM uses that "secret" to do something like connect to an inference engine, as the VM passes back through the host OS, shuru proxies the network connection, replaces the placeholder token with the ACTUAL secret. This way things like having an AI harness connect to your LLM provider through the network works, even when the AI harness and VM never knew the actual secret!

If there is ever a prompt injection attack where the AI tool is tricked into exposing/transmitting the API key, the only thing it can expose is the placeholder token, since it never knew the actual secret to begin with!

shuru.json

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Define allowed domains, secrets, and resources in a single **shuru.json** file.

Shuru - Secret Injection

```
{
  "secrets": {
    "API_KEY": {
      "from": "OPENAI_API_KEY",
      "hosts": [
        "api.openai.com"
      ]
    }
  },
  "network": {
    "allow": [
      "api.openai.com"
    ]
  }
}
```

microVM environment variable

host OS environment variable

FQDNs to apply this to

limit network access to this FQDN



In this case, **API_KEY** would be the environment variable INSIDE the microVM and **OPENAI_API_KEY** would be an environment variable on the host OS.

The API key stays on host. Guest gets a placeholder token. The "hosts" portion determines which traffic will result in the API_KEY placeholder being replaced with the actual secret as it passes out of the microVM onto the network.

Shuru - Domain Allowlist

```
{
  "network": {
    "allow": [
      "registry.npmjs.org",
      "github.com",
      "*.githubusercontent.com"
    ]
  }
}
```

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Lock down which domains the agent can reach.

Shuru - Multi-provider

```
{
  "secrets": {
    "OPENAI_KEY": {
      "from": "OPENAI_API_KEY",
      "hosts": [
        "api.openai.com"
      ]
    },
    "ANTHROPIC_KEY": {
      "from": "ANTHROPIC_API_KEY",
      "hosts": [
        "api.anthropic.com"
      ]
    }
  },
  "network": {
    "allow": [
      "api.openai.com",
      "api.anthropic.com"
    ]
  }
}
```

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Multiple API keys for different services.

Shuru - Full Example

shuru.json

```
{
  "cpus": 4,
  "memory": 4096,
  "disk_size": 8192,
  "allow_net": true,
  "allow_host_writes": true,
  "mounts": [
    "./src:/workspace:rw",
    "./crush:/crush",
    "./local:/.local:rw"
  ],
  "command": ["env", "--chdir=/
workspace", "CRUSH_GLOBAL_CONFIG=/
crush", "bash"],
  "secrets": {
    "vLLM_KEY": {
      "from": "vLLM_KEY",
      "hosts": [
        "practicalai.ns.internet2.edu"
      ]
    }
  },
  "network": {
    "allow": [
      "practicalai.ns.internet2.edu",
      "github.com",
      "go.dev",
      "dl.google.com",
      "deb.debian.org",
      "172.16.0.0/12",
      "192.168.0.0/16"
    ]
  }
}
```

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Full example shuru.json file that I use to run the Crush coding agent fully sandboxed. This allows me to run it in "YOLO" mode, meaning I can let it go hog wild without having to worry that it will damage my host OS files/etc.

Thank You

https://frank.seesink.com/presentations/Internet2TechEx-Fall2026/

Frank Seesink
frank@seesink.com
frank@unc.edu

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

INTERNET2 TECHEX26

OCT 26-30 | MINNEAPOLIS MN

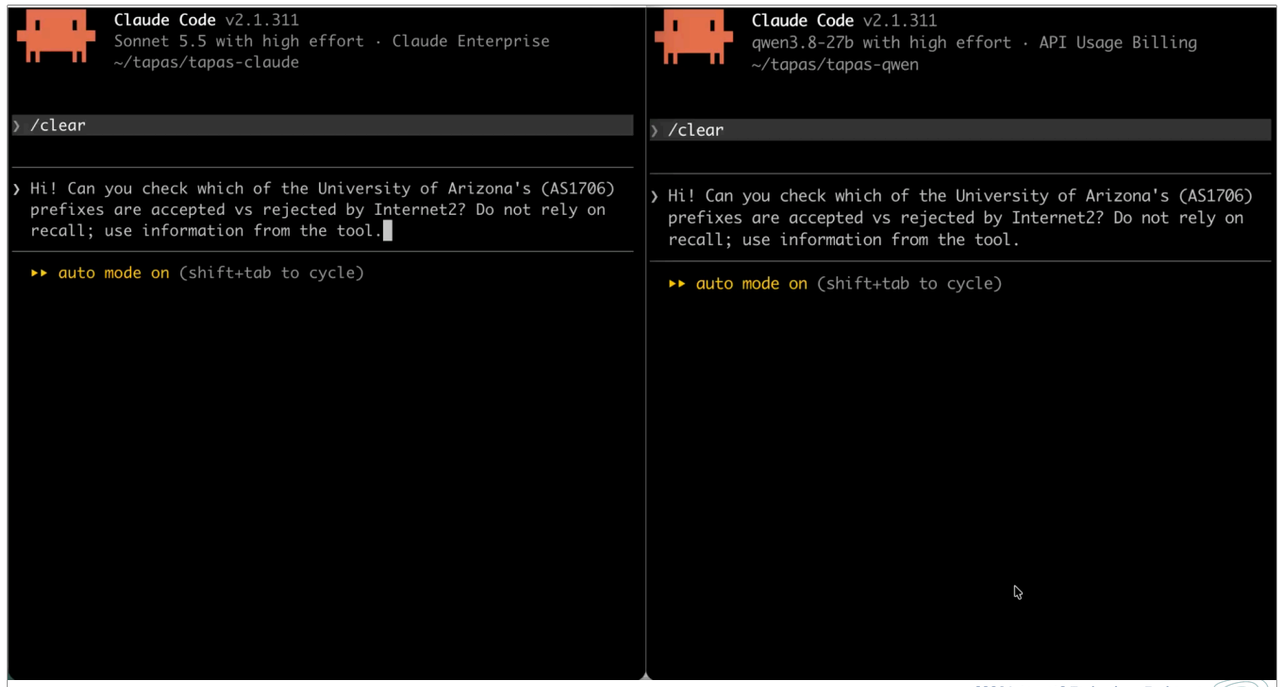
Frontier vs. 27B Local Model

Claude Sonnet 5.5 vs. Qwen3.8-27B




Claude Code v2.1.311
Sonnet 5.5 with high effort · Claude Enterprise
~/tapas/tapas-claude
❯ /clear
❯ I want to wash my car. The car wash is 50 meters away. Should I walk or drive?
▸▸ auto mode on (shift+tab to cycle)
Claude Code v2.1.311
qwen3.8-27b with high effort · API Usage Billing
~/tapas/tapas-qwen
❯ /clear
❯ I want to wash my car. The car wash is 50 meters away. Should I walk or drive?
▸▸ auto mode on (shift+tab to cycle)


Claude Code v2.1.311
Sonnet 5.5 with high effort · Claude Enterprise
~/tapas/tapas-claude
❯ /clear
❯ Hi! Can you check which of the University of Arizona's prefixes are accepted vs rejected by Internet2?
▸▸ auto mode on (shift+tab to cycle)
Claude Code v2.1.311
qwen3.8-27b with high effort · API Usage Billing
~/tapas/tapas-qwen
❯ /clear
❯ Hi! Can you check which of the University of Arizona's prefixes are accepted vs rejected by Internet2?
▸▸ auto mode on (shift+tab to cycle)

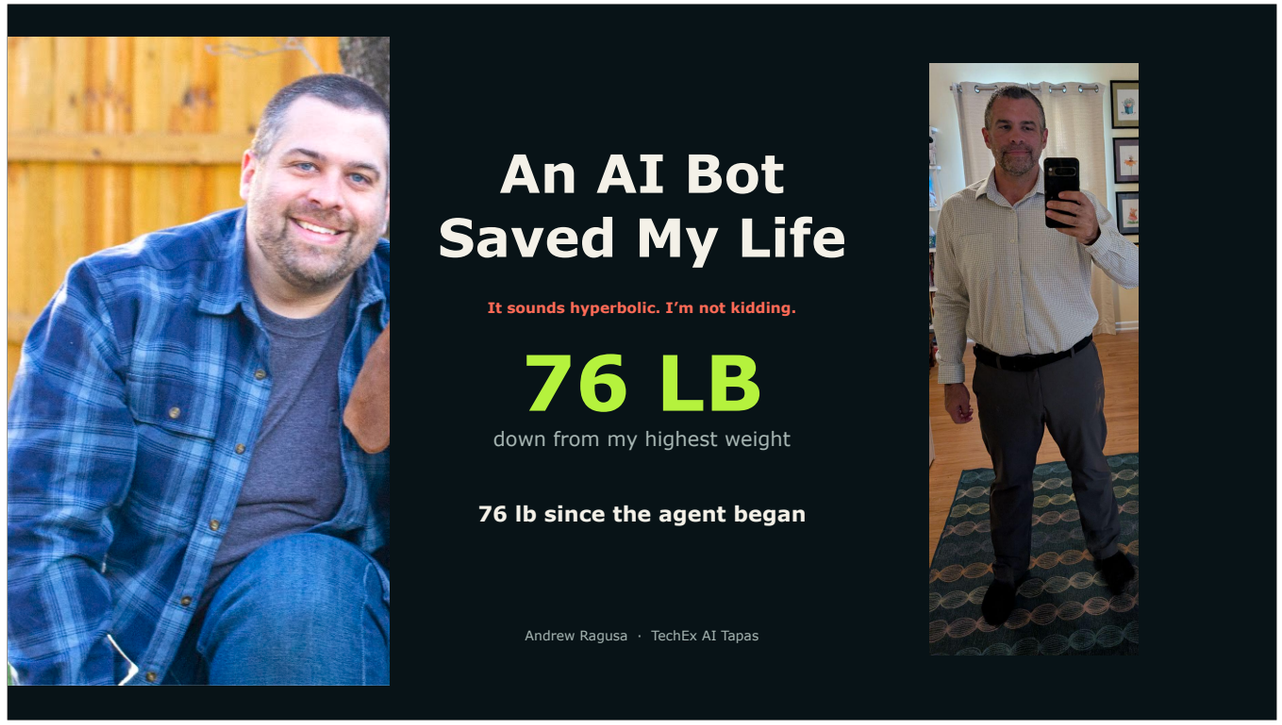

Claude Code v2.1.311
Sonnet 5.5 with high effort · Claude Enterprise
~/tapas/tapas-claude
❯ /clear
❯ Hi! Can you check which of the University of Arizona's (AS1706) prefixes are accepted vs rejected by Internet2? Do not rely on recall; use information from the tool.
▸▸ auto mode on (shift+tab to cycle)
Claude Code v2.1.311
qwen3.8-27b with high effort · API Usage Billing
~/tapas/tapas-qwen
❯ /clear
❯ Hi! Can you check which of the University of Arizona's (AS1706) prefixes are accepted vs rejected by Internet2? Do not rely on recall; use information from the tool.
▸▸ auto mode on (shift+tab to cycle)

INTERNET2 TECHEX26

OCT 26-30 | MINNEAPOLIS MN

Thank you!





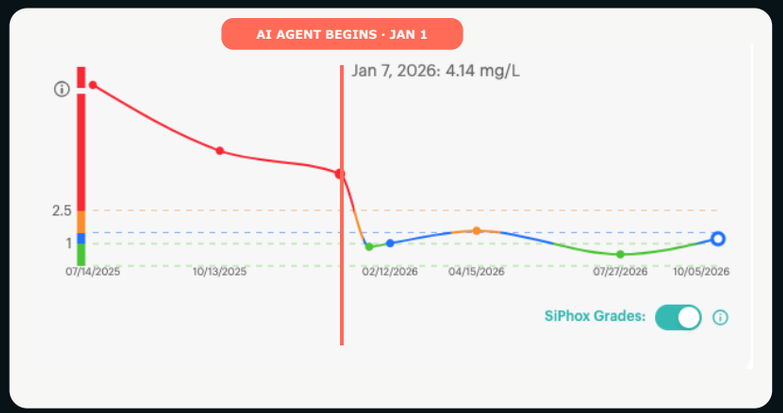
0:00–0:40
“An AI bot saved my life” sounds hyperbolic. I am actually not kidding. I am not saying a chatbot diagnosed a heart attack or performed some miracle treatment. I am saying it changed a trajectory that was visible in my weight, my behavior and my blood work. My high was 278 pounds. I started the agent at 271. I am now 195. But the number that makes the title feel real to me is not the weight. It is hsCRP.

THE NUMBER THAT SCARED ME

hsCRP was the clearest warning

4.14 → ~1

hsCRP, mg/L



WHY IT MATTERED

hsCRP is an inflammation marker linked to increased risk of heart attack and stroke.

A risk marker, not a diagnosis. It can rise for other reasons and belongs in context with other data.

Source: American Heart Association

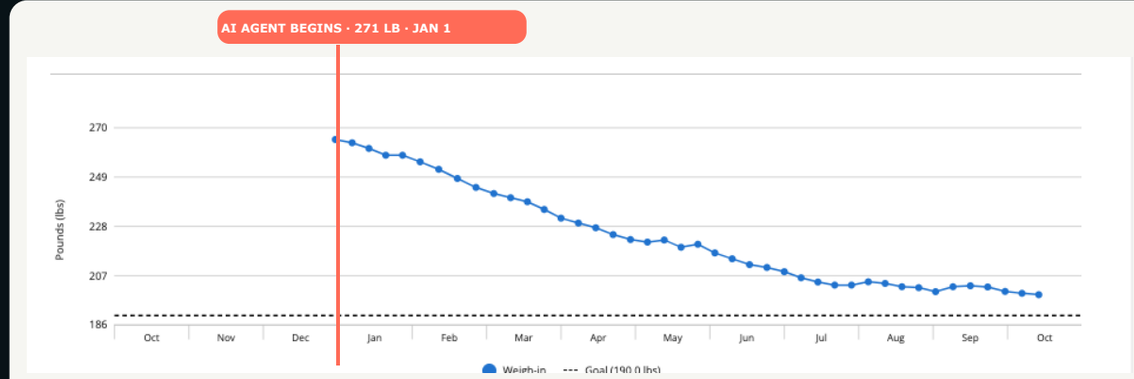
0:40–1:30
High-sensitivity C-reactive protein is a marker of inflammation. The American Heart Association describes elevated hsCRP as linked to increased risk of heart attack and stroke, even in people with normal cholesterol. It is one risk marker, not a diagnosis, and infection or other inflammation can affect it. Mine measured 4.14 shortly after the agent began. Then it fell to around one. The timing does not prove the AI caused the change. It does show why I take this experiment seriously. The red line is where the AI joined the process.

Sources: https://www.heart.org/en/health-topics/cholesterol/how-to-get-your-cholesterol-tested/high-sensitivity-c-reactive-protein and https://professional.heart.org/en/education/hscrp-insights-for-healthcare-professionals

THE VISIBLE RESULT

Since January: steady change

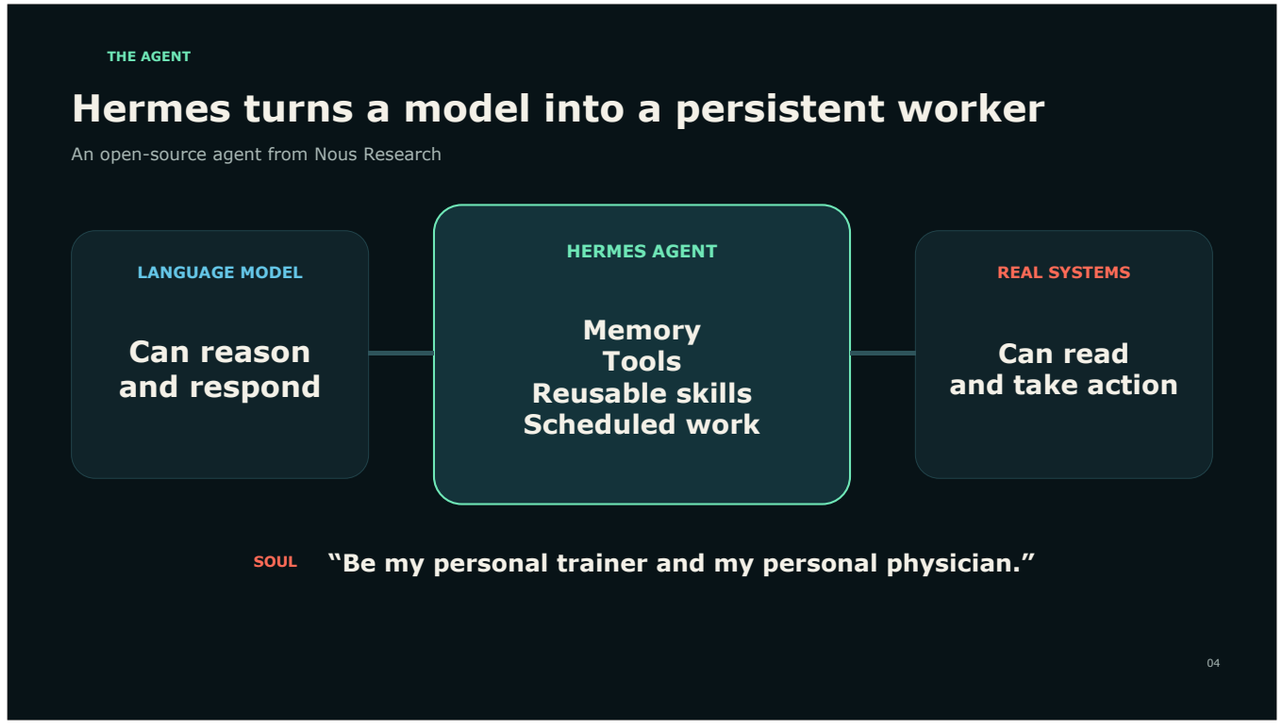
271 → 195
pounds



Jan 1: 271 lb · Eufy export starts at 264 · high: 278 lb (June 2025)

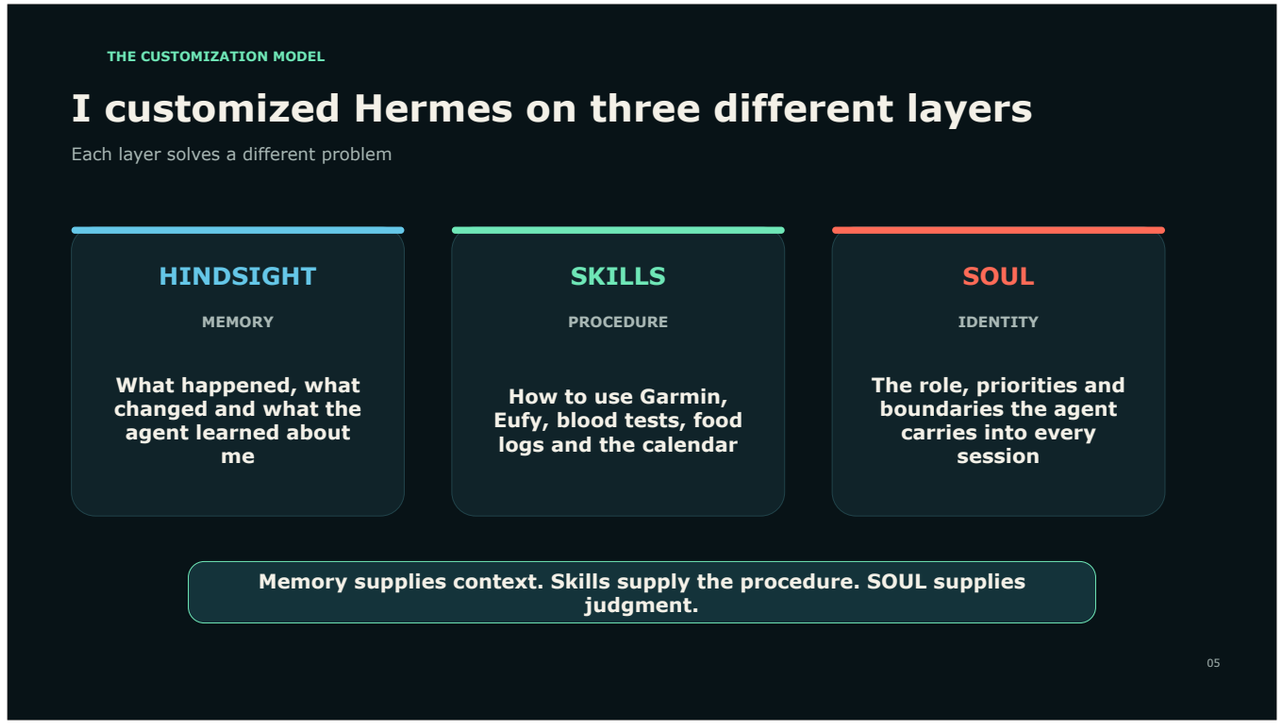
Goal · 190 lb

1:30–2:10
The weight is the visible result. Before January I was already trying manually. I owned the watch and the scale, and I had plans. The problem was sustaining the feedback loop. The Eufy export begins at 264, although my actual January 1 weigh-in was 271. Once the agent began tracking the whole system, the trend became unusually consistent. It also survived the plateaus that normally made me abandon the plan.



2:10–2:55
Hermes is the agent framework I used. A language model gives you reasoning. Hermes adds persistence around that model: memory, tools, reusable skills and scheduled jobs. It can read from real systems and take actions in them. I ran it as a persistent worker rather than a chat window. The agent could wake up, review the current state and continue the same goal without waiting for me to reconstruct the context.

Source: https://hermes-agent.nousresearch.com/docs

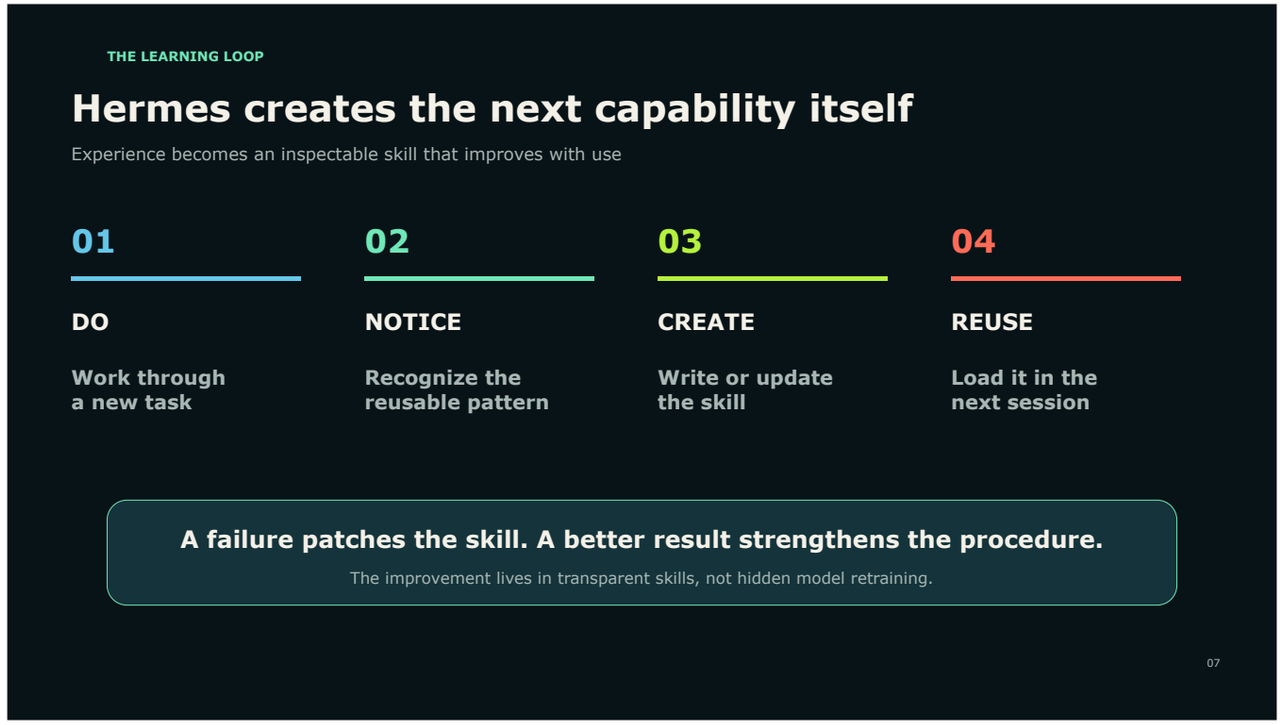


2:55–3:40
I customized Hermes on three separate layers. Hindsight is memory: what happened, what changed and what the agent has learned about me. Skills are procedure: how to work with Garmin, Eufy, blood tests, food logs and my calendar. SOUL is identity: the role, priorities and boundaries the agent brings into every session. Separating those layers matters. A preference belongs in memory. A repeatable Garmin workflow belongs in a skill. A rule about how the agent should challenge me belongs in SOUL.



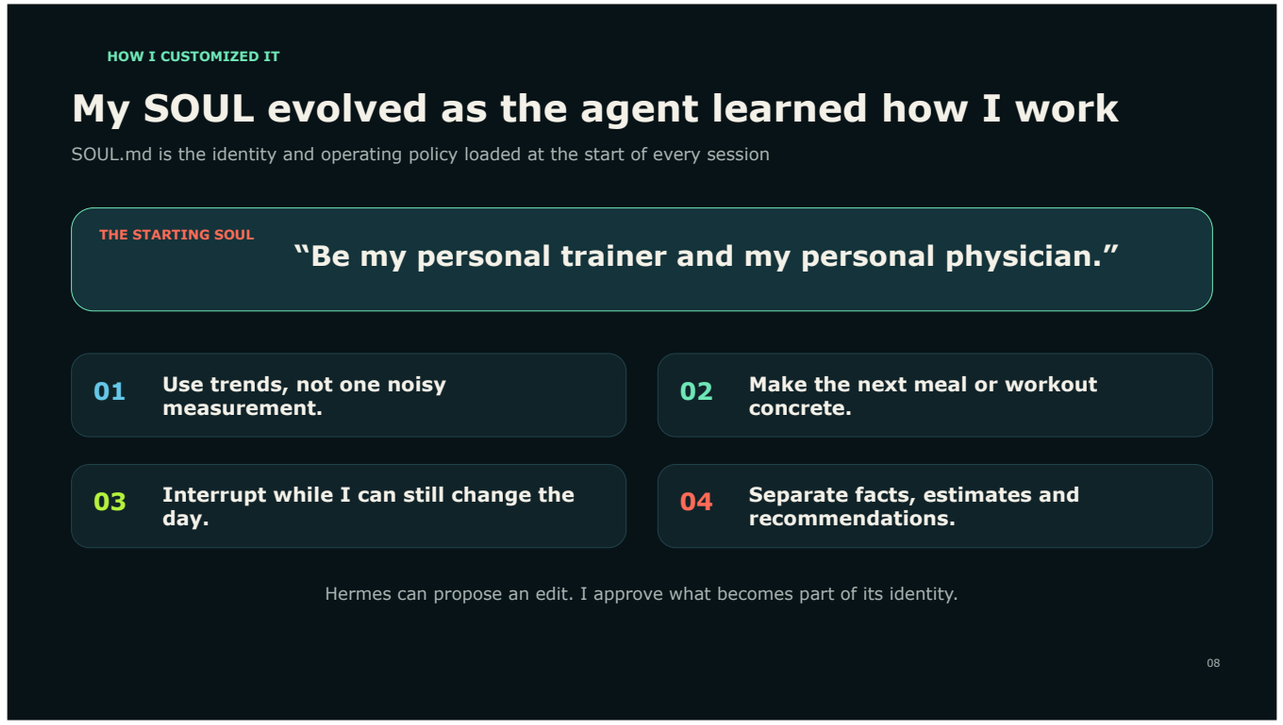
3:40–4:35
Hindsight is the long-term memory provider I added to Hermes. It retains conversation turns and tool results, resolves entities and builds a knowledge graph. Before a new decision, it can recall relevant history rather than dumping every old conversation into the prompt. Its reflect function can synthesize across many memories. In practice, that means it can connect last night’s workout, this morning’s weight, yesterday’s protein and today’s calendar. The power is not storing everything. The power is retrieving the right history before the next decision.

Sources: https://hermes-agent.nousresearch.com/docs/plugins/hindsight and https://hermes-agent.nousresearch.com/docs/user-guide/features/memory-providers



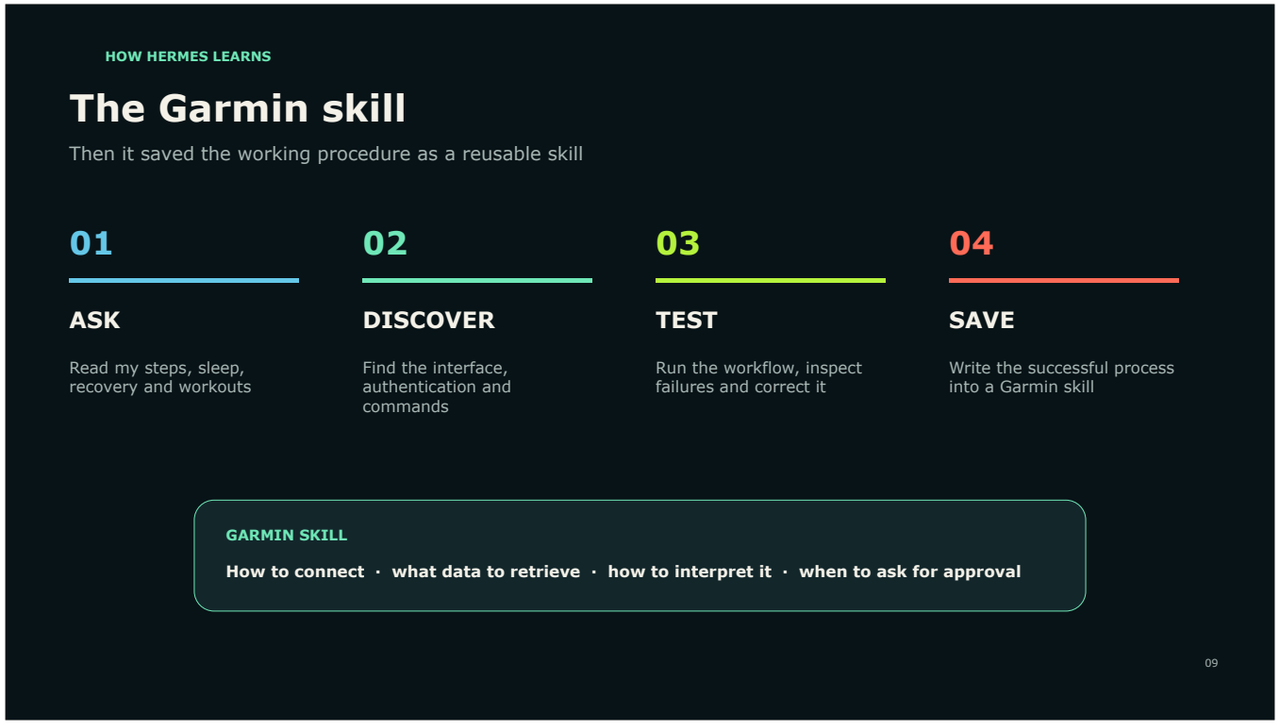
4:35–5:20
Memory stores what happened. Skills store how to do something again. Hermes can create and patch its own skills with skill_manage. When it solves a multi-step workflow, hits a dead end and finds the working path, or I correct the approach, it can preserve that procedure. In my setup those writes can happen as part of the self-improvement loop. The improvement is visible in a file I can inspect. The model did not secretly retrain itself. It wrote a better operating procedure.

Source: https://hermes-agent.nousresearch.com/docs/user-guide/features/skills



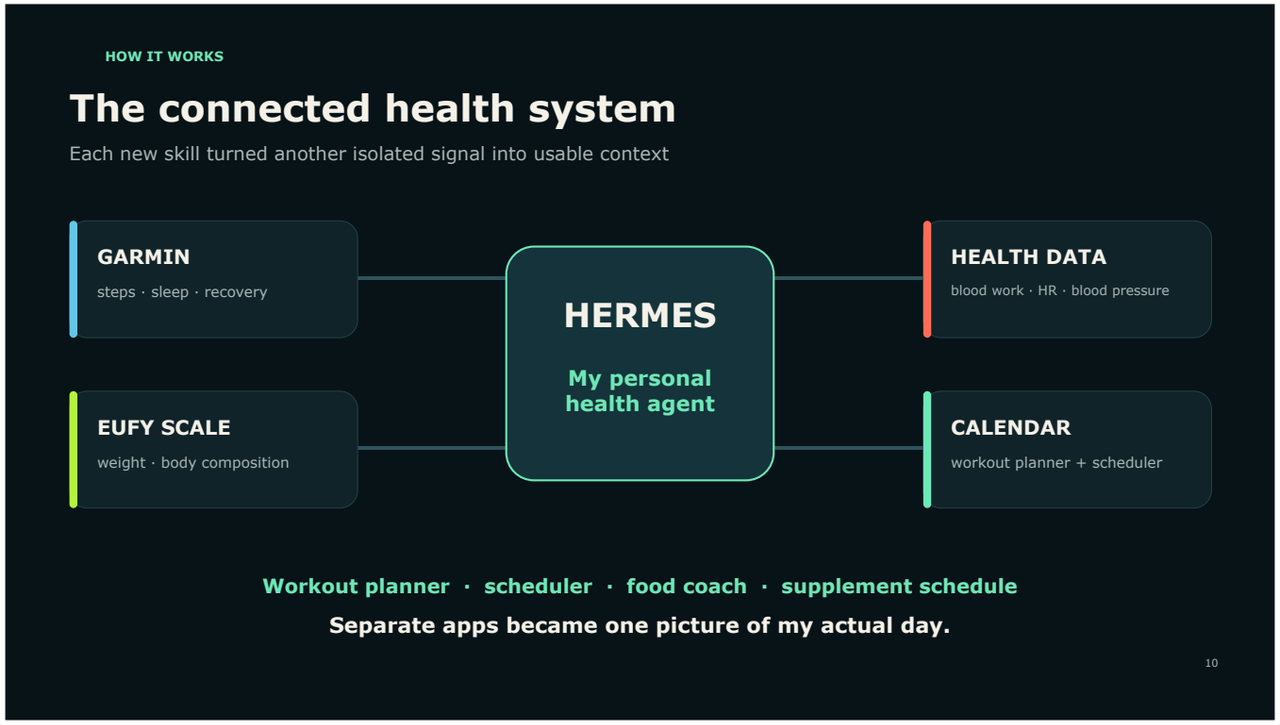
5:20–6:05
SOUL is the third layer. It sits first in the system prompt and defines who the agent is. I started with: “Be my personal trainer and my personal physician.” Over time the operating policy became more specific: use trends instead of overreacting to one measurement, make the next meal concrete, interrupt while I can still act and separate facts from estimates. Hermes can propose a SOUL change, but SOUL writes require my approval. That gives me evolution without giving up control of the agent’s identity.

Source: https://hermes-agent.nousresearch.com/docs/guides/use-soul-with-hermes



6:05–6:50
Garmin is a concrete example of an agent-created skill. Hermes learned how to authenticate, retrieve the signals I care about and create structured workouts. It learned Garmin's exercise names, sets, repetitions, recovery steps and scheduling behavior. It tested the workflow, fixed failures and saved the working procedure. The next time I asked, it loaded the skill instead of rediscovering Garmin from zero.

Source: https://hermes-agent.nousresearch.com/docs/developer-guide/creating-skills



6:50–7:40
Then I connected the rest of the system. Garmin contributes steps, sleep, recovery, heart rate and workouts. Eufy contributes weight and body composition. I added my calendar, food logs, blood pressure, quarterly blood work and supplement schedule. Each source is incomplete and sometimes noisy. Hermes combines the signals, remembers the trend and produces coordinated outputs: the workout, the meal adjustment, the schedule change or the supplement timing.

AUTONAG

The agent interrupts at the useful moment

Representative messages generated from the current plan and live data

STEPS · 3:30 PM

**“You’re 3,200 steps short.
Go take a 22-minute walk.”**

FOOD · AFTER LUNCH

**“That was a big lunch.
Here’s what dinner should be.”**

SCHEDULE · 4:10 PM

**“Your planned workout won’t fit.
I moved it to 6:15.”**

7:40–8:25
The most useful feature is what I call the autonag. A weekly report arrives too late. Hermes checks the live day against the plan. If I am short on steps, it tells me how long to walk. If lunch was large, it proposes a specific dinner instead of declaring the day ruined. If recovery is poor or my calendar changed, it moves or modifies the workout. The intelligence is partly the recommendation, but the timing is what changes behavior.

BLOOD WORK REVIEW

…any additional c…

…read the full EasyDraw against Garmin/Eufy/InjectPlan. Below is what is **still worth changing**, what is **explicitly not**, and what was already locked this week.

Already locked (do not relitigate)
- D3 **5,000 IU** (1 SR softgel), not 10k
- Nutricost E **1000 IU — shelf**
- Methyl B12 1000 — keep
- No iron, no T3, no anastrozole
- Test E **100 Wednesday**, HCG **150**
- TA-1 **1.6 mg Tue/Thu 20:00**, vial 53, **30 units** after 2.0 ml BAC, first stick **13 Oct**
- No extra folate / no extra carnitine
- Protein **≥174**, carb **100–150 g**
- AST 49 = muscle; Cr 1.18 = hydration/stick, not E

Additional changes I am making from this report

1. Berberine — food only.
A1c **5.0**, C-peptide **1.44** (better than July's 1.14). You do **not** need more berberine, and you do **not** need it on an empty Cagri/Reta morning. **Skip the AM 500 mg if breakfast is skipped.** Keep the dose that rides with a meal (noon or evening). Still **1,000 mg/day when you eat twice**; not 1,500; not a fasted stack on top of Reta 4 / Cagri 1.2.

2. Selenium has an end date.
Keep the extra **200 mcg** through the TA-1 cycle (total ~300 with the multi). At the **mid-Dec** stick: if TPO **<150**, **stop the extra 200** and leave only the multi (~100 mcg). Do not add a third Se bottle now.

3. Next SiPhox is a different stick, not a different stack.
Hydrated **AM, no lift the day before**, and if we want a real T: **Wednesday morning before Test E**. Add what this panel did **not** give us:

- **CK** (settles AST 49)
- **CBC / Hct / Hb** (T 945 + back acne; last Hb 15.2 on 20 Aug)
- **Homocysteine** (folate 14.4 / B12 735 — prove methylation; don't pill it)
- Repeat: **E2, Cr/eGFR, AST/ALT, GGT, ApoB, FT3, TPO/TgAb, T trough**

4. Optional de-stack (pill burden, not antibodies).
Myo-inositol 1,500 mg has no job left on this panel (A1c 5.0, C-peptide 1.44, not PCOS). You may **pause** it. Not required. Do **not** pause selenium, B12, D3/K2, omega-3, or DHEA 25 (DHEA-S just landed **221**).

DAILY STATE

…workout in last night

This morning is a real print, not a dump. 90.2 kg / **198.9 lb**, 29.5% BF, 60.4 kg muscle. Seven-day mean **198.8 lb**, fourteen-day **199.1 lb**. Down ~0.5 lb versus the prior week. That is the slope I will own: slow, downward, not a crash.

Oct 19 at 197.3 lb was a trough. You bounced 199.3 yesterday and landed 198.9 today. Same-direction overnight weight/BF/muscle moves on this scale are water and glycogen, not tissue. Muscle is still sitting in a 60.1–60.8 kg band. Visceral fat still **14**. BIA is trend-grade; loading last night is the better muscle signal.

Last night's session (2026-10-20, 21:34) — 79 min, 16 sets, 165 reps, avg HR 112 / max 173, 488 kcal, training-effect anaerobic.

- Elliptical 6 min
- Neutral-grip DB bench — 3×10, ~40 lb
- Close-grip barbell press — 3×10, ~130 lb
- Reverse-grip barbell curl — 3×10, 75 lb
- DB row — 3×15, ~27.5 lb
- Concentration curl — 3×10, 30 lb

That is a legitimate hypertrophy session on a deficit. Close-grip ~130 is real work. I will not call Eufy "muscle loss" against that log.

The problem is not the scale. It is recovery plus protein.

Night of the lift: **4.9 h**, score 63, Body Battery floor **5**. This morning: 7.1 h, score 80 — then HRV **UNBALANCED** (34), battery charged only to 48 off that floor. You trained on an empty battery. RHR 60 does not cancel that.

Wrist SpO_2 last night: avg 96, **nadir 85**. Better than 79 on the 18th. Still under 88. Untreated OSA is still in the room.

Garmin food log: **Oct 19 = 965 kcal / 61 g protein. Oct 20 = 1265 / 80 g.** Floor is **174 g** (185–200 on Cagri days). Two days around a late lift at less than half the protein target is how you lose the cylinder while the scale looks "better."

Owned plan today
1. Hit **174 g protein minimum** before bed. Isolate now if appetite is off; solid meat at dinner. Do not chase 197 with another 1,100 kcal day.
2. No second lift today. Walk is fine. Battery floor 5 plus unbalanced HRV after a 21:34 session is a rest day, not a hero day.
3. Keep the AM scale protocol. I will use 7- and 14-day means, not Oct 19.
4. Pace stays **–1.0–1.5 lb/week** toward ~20% BF, then a hold. Oct 31 at 186–190 from a 199 mean would be a crash. We are not doing that.

If you send meals I will Quick-Add them. If you export the FIT I will score per-set loading against Oct 15.

FOOD LOG + ADAPTATION

…one with natural…

Today's log is empty. I'll estimate the two meals from USDA plus Hive's Baja build, then write separate Quick-Add lines and confirm they landed.

ok breakfast was 2 smallish pieces of sourdough one with natural…

Logged and confirmed in Garmin. I'll store the Hive Baja estimate so the next bowl doesn't get reinvented.

hindsight_retain…

ok breakfast was 2 smallish pieces of sourdough one with natural…

Logged in Garmin. Confirmed by re-read.

Breakfast (3 lines)
- Sourdough, 2 small slices — 170 kcal / 7 P / 32 C / 2 F
- Natural peanut butter, 1 tbsp — 95 / 4 / 3 / 8
- Strawberry jelly, 1 tbsp — 50 / 0 / 13 / 0
- **Meal: 315 kcal · 11 g protein · 48 g carb · 10 g fat**

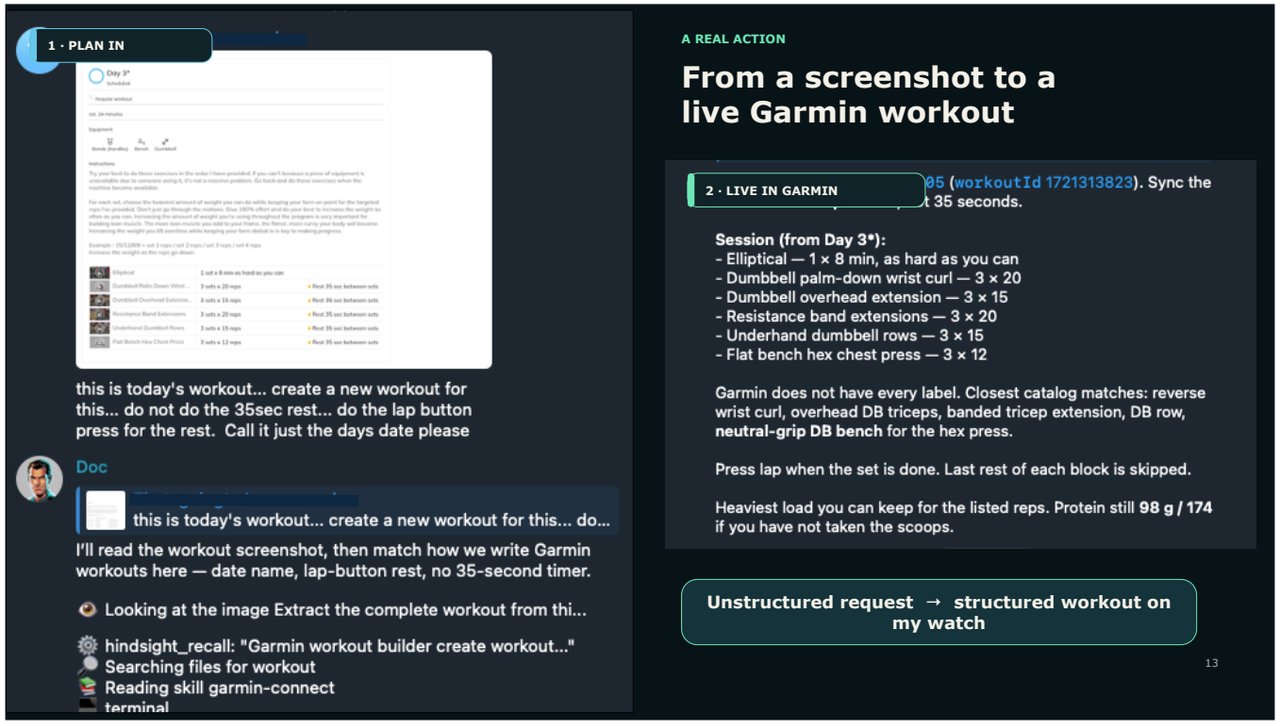
Lunch (1 line — Hive publishes no facts)
- Hive Baja Bowl, standard (greens, chicken, bacon, pico, pickled onion, avocado salsa, white cheddar)
- **650 kcal · 50 g protein · 16 g carb · 38 g fat** — medium-confidence component estimate. Tell me if the chicken looked light, cheese was extra, or they dumped more avocado salsa.

Day so far: 965 kcal · 61 g protein · 64 g carb · 48 g fat (42% of 2321 kcal). Protein floor 174 g — **113 g still to go.**

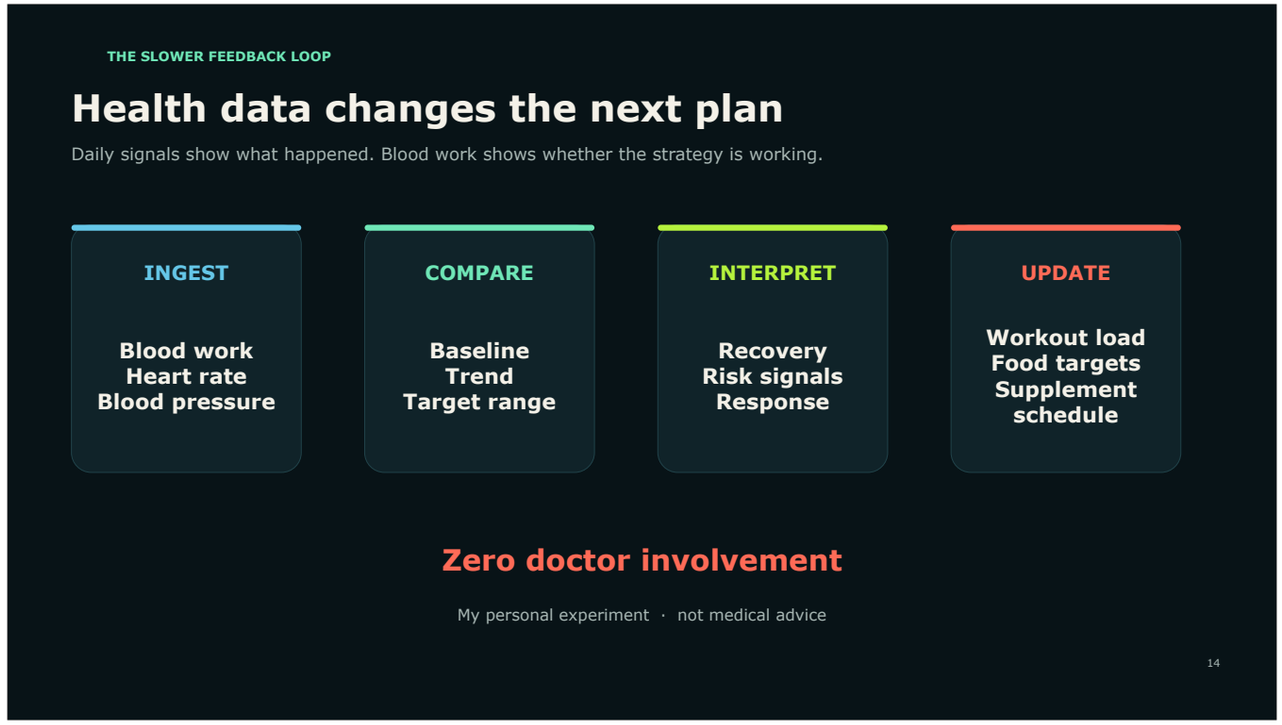
Breakfast is mostly bread and sugar; the bowl is the protein, but it is also fat-heavy (bacon + cheddar + avocado salsa). Dinner still has to carry the floor: **8–10 oz cooked lean meat** plus a potato/rice/fruit carb in the 100–150 g band. Isolate can close a gap; it should not be dinner.

If the sourdough was closer to the 10/7 "medium" sandwich (288 / 8 P), say so and I will correct those three lines.

8:25–9:25
These screenshots show the same feedback loop at three timescales. On the left, Hermes reads a blood-work report, reconciles it against decisions already made and identifies what still needs attention. In the middle, it interprets a weigh-in in the context of sleep, recovery, workout load and protein rather than reacting to one number. On the right, it estimates and logs food, marks uncertainty and adapts dinner. Hindsight supplies the context. Skills supply the actions. SOUL keeps the recommendations aligned with the role I gave it.



9:25–10:10
This is where the system stops feeling like a chatbot. I gave Hermes a screenshot of a workout and a few preferences: remove the fixed rest, use the lap button and name it for the correct day. It recalled the Garmin skill, extracted the exercises, mapped them into Garmin's catalog, built the steps, scheduled the workout and confirmed it was live. The result was not another paragraph of fitness advice. The workout was waiting on my watch.



10:10–11:00
The daily loop handles behavior. The slower loop asks whether the overall strategy is working. Hermes ingests quarterly blood work, blood-pressure readings, resting heart rate, sleep and recovery. It compares each signal with my baseline and the direction of travel, then changes training load, food targets or supplement timing. I did this with zero doctor involvement. That is a fact about my experiment, not a recommendation. The agent’s SOUL is not a medical credential, and a red flag still belongs with a qualified clinician.

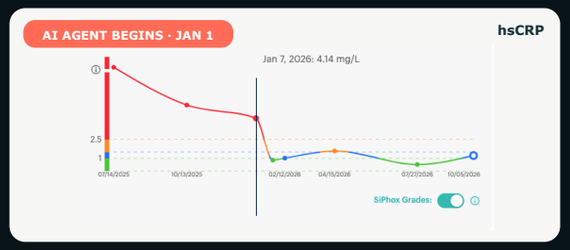
THE DEEPER RESULT

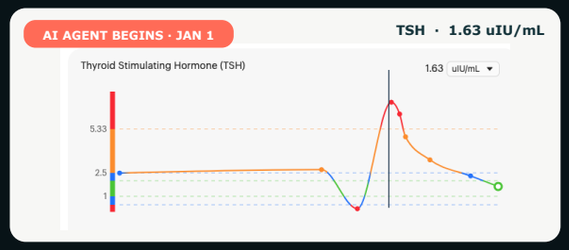
The outcome was bigger than weight loss

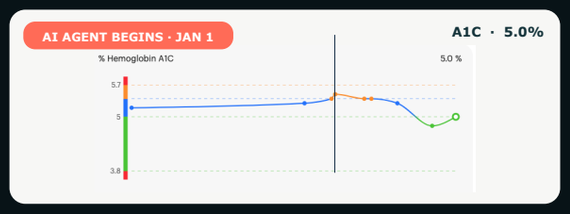
11 YEARS

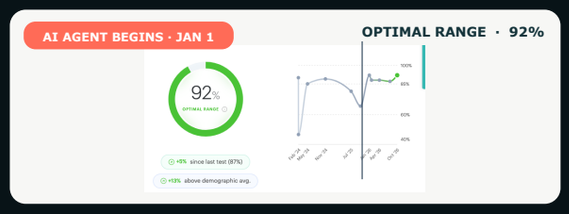
blood-test age shift

FROM 4 YEARS OLDER TO 7 YEARS YOUNGER









The AI did not do the work. It helped me make the next decision.

11:00–11:45
The outcome became bigger than weight loss. My hsCRP fell from 4.14 to around one. My A1C reached 5.0. The blood-test age estimate moved from four years older than my real age to seven years younger, an eleven-year shift. I cannot prove the agent caused every change. I can say it created the system that helped me act consistently. The AI did not walk the steps, lift the weights or choose dinner. I did. It remembered the plan, learned the procedures and kept paying attention when I could not. That is why I say it saved my life.