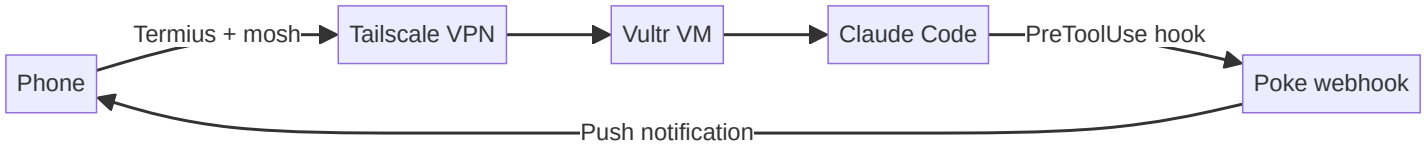
CLAUDE CODE ON-THE-GO

Author: mgranda
Date: January 2026

I run six Claude Code agents in parallel from my phone. No laptop, no desktop—just Termius on iOS and a cloud VM.

The Setup



The loop is: kick off a task, pocket the phone, get notified when Claude needs input. Async development from anywhere.

Infrastructure

A Vultr VM in Silicon Valley:

Spec	Value
Instance	vhf-8c-32gb
Cost	$0.29/hr (~$7/day when running)
Access	Tailscale-only (no public SSH)

I pay only when working. Two scripts handle lifecycle:

```
vm-start   # Start VM, wait for Tailscale, connect via mosh
vm-stop    # Halt VM
```

I also have an iOS Shortcut that calls the Vultr API directly—start the VM from my phone before I even open Termius.

The VM's public IP has no SSH listener. All access goes through Tailscale's private network. Defense in depth: cloud firewall blocks everything except Tailscale coordination, local nftables as backup, fail2ban for good measure.

Mobile Terminal

Termius handles SSH and mosh on iOS/Android. Mosh is the key—it survives network transitions. Switch from WiFi to cellular, walk through a dead zone, put the phone to sleep. The connection persists.

```
mosh --ssh="ssh -p 47892" mgranda@100.107.156.125
```

One gotcha: mosh doesn't forward SSH agent. For git operations that need GitHub auth, I use regular SSH inside tmux.

Session Persistence

The shell auto-attaches to tmux on login. Close Termius, reopen hours later, everything's still there.

```
# In .zshrc
if [[ -z "$TMUX" ]]; then
    tmux attach -t main 2>/dev/null || tmux new -s main
fi
```

Multiple Claude agents run in parallel windows. C-a c for new window, C-a n to cycle. Works well on a phone keyboard.

Push Notifications

This is what makes mobile development practical. Without notifications, you'd constantly check the terminal. With them, you can walk away.

The hook in ~/.claude/settings.json:

```
{
  "hooks": {
    "PreToolUse": [{
      "matcher": "AskUserQuestion",
      "hooks": [{
        "type": "command",
        "command": "~/.claude/hooks/poke-notify.sh question"
      }]
    }]
  }
}
```

When Claude calls AskUserQuestion, the hook fires. A simple script extracts the question and POSTs to Poke's webhook:

```
QUESTION=$(echo "$EVENT_DATA" | jq -r '.tool_input.questions[0].question')
MESSAGE="$PROJECT_NAME: Claude needs input - $QUESTION"
curl -X POST "$API_URL" -d "{\"message\": \"$MESSAGE\"}"
```

Phone buzzes. Notification shows the question. Tap, respond, continue.

Trust Model

I run Claude Code in permissive mode. The VM is isolated—no access to production systems, no secrets beyond what's needed for development. Worst case: Claude does something unexpected on a disposable VM.

Cost control adds another layer. The VM costs $0.29/hr. Even if something runs away, the daily cap is bounded.

Parallel Development

Git worktrees let me run multiple features simultaneously:

```
~/Code/myproject/              # main
~/Code/myproject-sidebar/      # feature branch
~/Code/myproject-dark-mode/    # another feature
```

Each worktree gets its own tmux window with a Claude agent. Port allocation is hash-based—deterministic from branch name, no conflicts:

```
hash_val = sum(ord(c) for c in branch_name)
django_port = 8001 + (hash_val % 99)
```

Six agents, six features, one phone.

What This Enables

Review PRs while waiting for coffee. Kick off a refactor on the train. Fix a bug from the couch while watching TV.

The pattern: start a task that will take Claude 10-20 minutes, do something else, get notified, respond, repeat. Development fits into the gaps of the day instead of requiring dedicated desk time.

The Components

Tool	Purpose
Vultr	Cloud VM ($0.29/hr, pay-per-use)
Tailscale	Private network, secure access
Termius	iOS/Android SSH client
mosh	Network-resilient shell
tmux	Session persistence
Poke	Push notifications via webhook
Claude Code	The actual work

The setup took one Claude Code session to build—gave it my Vultr API key and access to gh, asked for a secure dev VM. Now I code from my phone.