



---

# Claude Code Memory Issues

*The Real Causes and Solutions*

JANUARY 20, 2026

[THEDECIPHERIST.COM](https://thedeCipherist.com)

# TABLE OF CONTENTS

---

The Problem	3
Root Cause #1: VS Code Extension Host Memory Limit	3
Root Cause #2: Session File Size Limits	4
Root Cause #3: WSL2 + Docker Memory Leak (The Big One)	4
Root Cause #4: No Persistent Memory Between Sessions	5
The Real Solution: Move Docker Off WSL2 Entirely	6
Quick Reference	7
Context Window Facts	8
Commands to Know	8

**TL;DR:** Claude Code's memory issues aren't random - they're caused by VS Code extension limits, session file caps, and WSL2/Docker memory leaks. Here's what's actually happening and how to fix it.

---

## THE PROBLEM

---

You're deep in a session with Claude Code, making great progress. You step away for coffee. Come back and... Claude acts like you're strangers.

### Sound familiar?

- Context wasn't cleared
- VS Code stayed open
- You just took a short break

Yet Claude has amnesia. What gives?

---

## ROOT CAUSE #1: VS CODE EXTENSION HOST MEMORY LIMIT

---

The VS Code extension architecture has a hard memory limit (~2-3GB). It wasn't designed for persistent AI operations.

Every interaction stores context in memory:

- Files you're working on
- Conversation history
- Code changes being tracked

When you're rapidly prototyping, this builds up fast. Hit the limit → extension host terminates → Claude Code dies with it.

### The fix:

Restart extension host before it crashes:

`Ctrl+Shift+P` → "Developer: Restart Extension Host"

Do this when you notice "Code Helper (Plugin)" exceeding 2GB in Task Manager.

---

## ROOT CAUSE #2: SESSION FILE SIZE LIMITS

---

Claude Code stores sessions as `.jsonl` files in `~/.claude/projects/.../`

When a session's context grows too large, resuming becomes impossible. You get "Prompt is too long" before you can even type.

### From GitHub Issue #14472:

*"Sessions that grow large become permanently inaccessible. Users lose all conversation history and must start fresh."*

### The fix:

Checkpoint regularly. After completing significant work:

*"Let's checkpoint. Summarize what we've built before we continue."*

Use built-in commands:

- `/compact` - Reduces context size strategically
- `/clear` - Fresh start when needed
- `/context` - Debug context issues (v1.0.86+)

---

## ROOT CAUSE #3: WSL2 + DOCKER MEMORY LEAK (THE BIG ONE)

---

If you're running Claude Code with MCP servers in Docker on WSL2, here's what's happening:

```
Claude Code → MCP servers (Docker) → Docker daemon (WSL2)
→ Linux kernel caches aggressively → vmem process (Windows)
→ Memory balloons to 10GB+ → System instability
→ Docker gets weird → Claude loses MCP connections → Session corrupts
```

**The dirty secret:** WSL2 allocates memory but doesn't release it back to Windows. The Linux kernel caches aggressively by design.

Check Task Manager → `vmem` = WSL2's memory consumption.

## Quick Fixes:

### 1. Cap WSL2 memory

Create `%UserProfile%\.wslconfig`:

```
[wsl2]
memory=8GB
swap=2GB
localhostForwarding=true
```

Then: `wsl --shutdown`

### 2. Docker cleanup

```
# See what's eating memory
docker system df

# Nuclear cleanup
docker system prune -a --volumes
```

### 3. Container memory limits

```
services:
  your-mcp:
    mem_limit: 256m
    mem_reservation: 128m
```

Container hits limit → only that container restarts, not your whole system.

## ROOT CAUSE #4: NO PERSISTENT MEMORY BETWEEN SESSIONS

---

### From GitHub Issue #14227:

```
"Claude Code starts every session with zero context. There is no memory of previous sessions."  
> "It's a goldfish."
```

This is by design currently. Each session starts fresh.

### The workaround:

Keep a `CLAUDE.md` in your project root with:

- Project context
- Architecture decisions
- What you've built so far

Claude reads this automatically at session start.

---

## THE REAL SOLUTION: MOVE DOCKER OFF WSL2 ENTIRELY

---

Here's my actual setup right now: Claude Desktop + 6 VS Code instances running Claude Code. vmmem sits at 11.6GB and climbs to 15GB+ over a session.

The quick fixes help, but the **real fix** is getting Docker completely out of WSL2.

### The architecture that actually works:

Windows (VS Code UI)

↓

WSL2 (LEAN - dev only)

├─ Project files

├─ Node.js / dev server

├─ Claude Code

├─ NO Docker ❌

└─ vmmem: 2-4GB ✓

↓ (socket/TCP)

Separate Linux box (Ubuntu PC, old laptop, mini PC, even Raspberry Pi 5)

├─ Native Docker / Docker Swarm

├─ All your containers

├─ Memory-capped per container

├─ No vmmem overhead

└─ Can reboot without touching Windows

### Why this works:

1. **WSL2 stays lean** - Only runs dev tools, not containers
2. **vmmem drops from 15GB to 2-4GB** - Massive difference
3. **HMR stays fast** - Project files still native in WSL
4. **Docker runs native** - No Windows/Linux memory translation overhead
5. **Blast radius contained** - Linux box acting up? Restart it. Windows keeps working.

### Hardware options for the Linux box:

- Old laptop you have lying around
- Mini PC (Intel NUC, Beelink ~\$150-200)
- Raspberry Pi 5 8GB (~\$80) for lighter workloads
- Or a proper server if you're scaling

The containers connect back via socket/TCP. Claude Code in WSL talks to your MCP servers on the Linux box. Same functionality, fraction of the memory.

### My results:

METRIC	BEFORE	AFTER
vmmem	11-15GB (climbing)	2-4GB (stable)
System stability	Degrades over hours	Stays clean all day
Container crashes	Takes down everything	Only that container restarts

This isn't a workaround - it's the proper architecture. WSL2 was never meant to run production container fleets.

---

## QUICK REFERENCE

---

PROBLEM	SOLUTION
Claude forgets mid-session	Restart Extension Host
"Prompt too long" on resume	Use <code>/compact</code> , checkpoint often
vmmem eating 10GB+	Cap WSL2 memory, clean Docker
Containers crashing randomly	Add <code>mem_limit</code> to each container
Always starting from zero	Use <code>CLAUDE.md</code> for project context

---

## CONTEXT WINDOW FACTS

---

PLAN	CONTEXT WINDOW
Standard	200K tokens
Enterprise (Sonnet 4)	500K tokens

- 1 token  $\approx$  0.75 words
  - Each message = Claude re-reads ENTIRE conversation
  - Usage resets every 5 hours
- 

## COMMANDS TO KNOW

---

```
# Check WSL memory usage
wsl -- free -h

# Check Docker memory
docker stats

# Clean Docker
docker system prune -a --volumes

# Restart WSL
wsl --shutdown
```

VS Code:

- `Ctrl+Shift+P` → "Developer: Restart Extension Host"
  - `/compact` - Reduce context
  - `/clear` - Fresh session
  - `/context` - Debug context
- 

This isn't a Claude problem - it's an architecture problem. The extension host, session files, and WSL2 memory management all have limits that weren't designed for persistent AI operations.

Work with those limits, not against them.

---

*Anyone else dealing with this? What workarounds have you found?*