

VS Code Remote-SSH 접속 시 원격 서버에서 전체 개발환경이 실행되는 문제

개요

VS Code Remote-SSH로 서버에 접속하면, 단순 파일 탐색과 터미널 사용 목적이더라도 서버에 VS Code Server가 설치·실행되며, 언어 서버(tsserver, PHP Tools, HTML/JSON Language Server 등), 파일 감시자(fileWatcher), 확장 호스트(extensionHost) 등 전체 개발환경 프로세스가 원격 서버의 RAM과 CPU를 점유한다.

이것은 버그가 아니라 **Microsoft의 의도된 설계(by design)**이다. 그러나 운영 서버에서는 심각한 리소스 경쟁과 OOM 위험을 초래한다.

본 이슈에서는 아키텍처의 근거 자료, 실제 피해 사례, 수행한 대응 조치, 그리고 대안을 정리한다.

1. 문제 상세

1.1 발견 경위

2026-04-22, 서버 MemAvailable이 813MB(20.8%)로 저하된 원인을 조사한 결과, **VS Code Server 관련 프로세스 13개가 RAM 2,362MB(서버 전체의 60.3%)를 점유**하고 있었다.

접속자는 1명이었고, 파일 편집과 터미널만 사용하는 상태였다.

PID	RSS	프로세스	비고
184586	617MB	extensionHost	DevSense 확장 4개 로드
184958	605MB	tsserver (semantic)	--max-old-space-size=3072
185085	341MB	devsense.phptools	PHP 전체 인덱싱
185184	186MB	devsense.intelli-php	AI 자동완성 모델
184957	163MB	tsserver (partial)	--max-old-space-size=3072
184213	103MB	server-main.js	VS Code Server 핵심
185204	89MB	html language server	내장 확장
184974	77MB	typingsInstaller	TS 타입 설치
184228	60MB	fileWatcher	파일 변경 감시
184258	60MB	ptyHost	통합 터미널
184994	49MB	json language server	내장 확장
184180	11MB	code command-shell	SSH 터널 진입점
184209	0MB	sh wrapper	프로세스 래퍼
합계	2362MB	13 프로세스	

1.2 접속 해제 후에도 프로세스가 잔류

VS Code를 종료(접속 해제)해도 **13개 프로세스 전량이 서버에 잔류**했다. `autoShutdown` 기본 설정이 미작동하여 24시간 이상 점유가 지속되었다.

2. 원인: Microsoft의 Remote-SSH 아키텍처

2.1 공식 문서 근거

VS Code Remote-SSH 공식 문서

“ URL: <https://code.visualstudio.com/docs/remote/ssh>

핵심 설명: Remote-SSH 확장은 원격 머신에 VS Code Server를 설치하고, 명령어와 확장을 원격 머신에서 직접 실행한다. 소스 코드가 로컬에 없어도 IntelliSense, 코드 네비게이션, 디버깅 등 로컬 수준의 개발 경험을 제공하는 것이 목적이다.

원문: "the extension runs commands and other extensions directly on the remote machine. The extension will install VS Code Server on the remote OS"

이것은 SSH를 단순 파일 전송/터미널 용도로 사용하는 일반적인 기대와 근본적으로 다르다.

VS Code Remote Development Overview

“ URL: <https://code.visualstudio.com/docs/remote/remote-overview>

핵심 설명: Remote Development 확장 팩의 각 확장은 컨테이너, WSL, 또는 원격 머신에서 직접 명령어와 확장을 실행한다. 이를 통해 로컬에서 실행하는 것과 동일한 경험을 제공한다.

원문: "Each extension in the Remote Development extension pack can run commands and other extensions directly inside a container, in WSL, or on a remote machine so that everything feels as it does when you run locally."

Extension Host 아키텍처 문서

“ URL: <https://code.visualstudio.com/api/advanced-topics/extension-host>

VS Code는 확장을 두 가지로 분류한다:

- **UI Extension** (`extensionKind: ["ui"]`): 테마, 스니펫, 키맵 등. **로컬에서 실행**.
- **Workspace Extension** (`extensionKind: ["workspace"]`): 디버거, 린터, 언어 서버 등. **원격 서버에서 실행**.

대부분의 개발 확장은 Workspace Extension으로 분류되어 **원격 서버에서 실행되는 것이 기본 동작**이다.

원문: "`extensionKind: ["workspace"]` — Indicates the extension requires access to workspace contents and therefore needs to run where the workspace is located."

Supporting Remote Development (확장 개발자용 가이드)

“ URL: <https://code.visualstudio.com/api/advanced-topics/remote-extensions>

확장이 양쪽 모두에서 실행 가능한 경우, UI Extension은 로컬 Extension Host에서, Workspace Extension은 Remote Extension Host(VS Code Server 안의 작은 서버)에서 실행된다.

`remote.extensionKind` 설정으로 특정 확장의 실행 위치를 강제 변경할 수 있으나, 이는 테스트 용도로만 권장되며 확장이 정상 동작하지 않을 수 있다.

원문: "Using `remote.extensionKind` allows you to quickly test published versions of extensions without having to modify their `package.json` and rebuild them."

Remote Development FAQ

“ URL: <https://code.visualstudio.com/docs/remote/faq>

VS Code Server는 Remote Development 확장의 구성요소이며, VS Code 클라이언트가 관리한다. 사용자가 접속할 때 자동으로 설치/업데이트되며, 별도 사용이나 다른 클라이언트의 사용은 의도되지 않았다.

최소 요구사항: **1GB RAM** (권장 2GB RAM, 2코어 CPU).

원문: "1 GB RAM is required for remote hosts, but at least 2 GB RAM and a 2-core CPU is recommended."

2.3 tsserver 기본 힙 크기 문제

VS Code에 내장된 TypeScript 언어 서버(tsserver)의 기본 `--max-old-space-size`는 **3072MB**이다. 이것은 대규모 TypeScript 프로젝트(수십만 줄)를 위한 설정이다.

본 서버의 BookStack 프로젝트는 TS 파일 215개, 63,551줄이며, 실사용 메모리는 50~200MB 수준이다. 그러나 기본 설정이 적용되면 V8 힙 상한이 서버 RAM(3,919MB)의 **78%**가 된다.

tsserver는 semantic 인스턴스와 partial 인스턴스 2개가 동시 실행되므로, 이론상 힙 상한의 합계는 6,144MB로 서버 RAM의 157%에 달한다.

3. 관련 GitHub 이슈 (동일 문제 보고)

아래 이슈들은 동일한 아키텍처적 문제로 인한 피해를 보고한 것이다.

3.1 메모리 관련

이슈	제목/내용	핵심
microsoft/vscode#151205	Remote SSH RAM 과다 점유	모든 원격 확장을 비활성화해도 10GB 서버에서 RAM 3.7GB 점유. 다른 앱이 메모리 할당 실패로 종료.
microsoft/vscode-remote-release#9778	vscode-server 메모리 누수	폴더를 열 때마다 별도 vscode-server 인스턴스 생성. 창을 닫아도 메모리 미회수. IDE 미사용 상태에서 8GB 잔류.
microsoft/vscode-remote-release#7825	Remote SSH 메모리 고갈	모든 확장 제거 후에도 fileWatcher 프로세스가 메모리를 제한 없이 소비.
microsoft/vscode-remote-release#3195	VSCode Server Node 리소스 과다 소비	WSL 2에서 폴더 열기만으로 Node 프로세스가 RAM과 CPU를 전부 소비. 메모리 누수로 BSOD 유발.
microsoft/vscode-remote-release#10567	비정상 메모리/CPU 사용으로 인한 크래시	Ubuntu 24 서버에서 접속 시 확장 자동 설치와 함께 리소스 소비 급증.

3.2 CPU 관련

이슈	제목/내용	핵심
microsoft/vscode-remote-release#3319	tsserver 고 CPU 사용	AWS t2.micro에서 tsserver와 typingsInstaller가 CPU 80% 이상 점유. JS만 작업하는데 TS 언어 서버가 실행. cgroup CPU 제한 기능 요청.
microsoft/vscode-remote-release#2716	고 CPU 사용	extensionHost 프로세스가 CPU 99% 점유.
microsoft/vscode-remote-release#1656	고 CPU 사용	다수 NodeJS 인스턴스로 인한 CPU 과부하.

3.3 아키텍처 개선 요청

이슈	제목/내용	핵심
microsoft/vscode#194583	확장 실행 위치 변경 UI 제공 요청	<code>remote.extensionKind</code> 설정의 문서가 부실하고, 원격 서버에서 불필요한 리소스 사용 문제. "모든 확장을 ui로 실행" 옵션 요청.
microsoft/vscode-remote-release#9454	사전 설치된 서버/확장 환경 지원 요청	접속할 때마다 새 버전 설치. 기존 설치를 재사용하는 옵션 요청.

4. 수행한 대응 조치

4.1 tsserver 힙 크기 제한

서버의 Machine settings(`/root/.vscode-server/data/Machine/settings.json`)에서 tsserver 힙 상한을 3,072MB → **256MB**로 변경.

```
{
  "typescript.tsserver.maxTsServerMemory": 256
}
```

BookStack TS 215파일/63K줄 규모에 256MB는 충분하다.

4.2 서버 실행 불필요 확장 제거

서버에 자동 설치된 DevSense 확장 4개(phptools, intelli-php, composer, profiler)를 삭제. 합계 527MB + 디스크 145MB 회수.

```
rm -rf /root/.vscode-server/extensions/devsense.phptools-vscode-*
rm -rf /root/.vscode-server/extensions/devsense.intelli-php-vscode-*
rm -rf /root/.vscode-server/extensions/devsense.composer-php-vscode-*
rm -rf /root/.vscode-server/extensions/devsense.profiler-php-vscode-*
```

로컬 VS Code에서 재설치를 방지하기 위해 `remote.extensionKind`로 로컬 전용 실행 강제:

```
"remote.extensionKind": {
  "devsense.phptools-vscode": ["ui"],
  "devsense.intelli-php-vscode": ["ui"]
}
```

4.3 VS Code Server 설정 강화

```
{
  "remote.autoForwardPorts": false,
  "remote.autoShutdown": true,
  "remote.autoShutdownDelay": 10,
  "extensions.autoUpdate": false,
  "search.followSymlinks": false,
  "files.watcherExclude": {
    "**/vendor/**": true,
    "**/node_modules/**": true,
    "**/storage/**": true,
    "**/public/dist/**": true
  }
}
```

```
}
}
```

설정	변경 전	변경 후	효과
tsserver.maxTsServerMemory	3072	256	인스턴스당 V8 힙 -92%
autoShutdownDelay	30분	10분	접속 해제 후 잔류 시간 -67%
extensions.autoUpdate	true	false	백그라운드 확장 업데이트 방지
autoForwardPorts	true	false	불필요 포트 포워딩 차단
watcherExclude	미설정	4경로	fileWatcher 부하 축소

4.4 구버전 서버 디렉토리 정리

VS Code Server 업데이트마다 새 디렉토리가 누적됨. 5개 버전(1.9GB) → 2개 버전(869MB)으로 정리. **디스크 855MB 회수.**

4.5 재발 방지 — 3중 방어 체계

단일 방어 수단은 실패할 수 있으므로 계층적 방어를 구축했다.

계층 1: autoShutdown (10분) 접속 해제 후 10분 대기 → VS Code Server 자동 종료.

계층 2: cron (매시간, 5h 임계) `/etc/cron.d/bookstack-vscode-cleanup`: autoShutdown 실패 시 백업. 5시간 이상 실행된 vscode-server 프로세스를 SIGTERM.

```
15 * * * * root ps -eo pid,etimes,cmd --no-headers | \
  awk '/vscode-server/ && $2 > 18000 {system("kill -TERM " $1)}' 2>/dev/null
```

계층 3: daily-cleanup.sh (05:45, 3개 서브섹션) `/var/www/bookstack/batch/daily-cleanup.sh` 섹션 6에 추가:

- 6a. orphan 프로세스 정리 (5시간 임계)
- 6b. 금지 확장(DevSense) 자동 삭제 — 로컬에서 재설치되더라도 일 1회 자동 제거
- 6c. Machine settings 무결성 검증 — 설정이 변경되었을 경우 원래 값으로 복원

방어 계층 흐름:

접속 해제 → autoShutdown(10분) → cron(매시간, 5h) → daily-cleanup(매일, 5h) → 월간 재부팅

실패 시나리오	최대 잔류 시간
-----	-----
autoShutdown 정상 작동	10분

autoShutdown 실패 → cron 보정	6시간	
autoShutdown + cron 모두 실패 → daily	24시간	

4.6 결과

	Before	After	절감
VS Code Server			
extensionHost	617MB	101MB	-84%
tsserver (x2)	768MB	0MB	(256MB 상한)
PHPTools	341MB	0MB	제거
IntelliPHP	186MB	0MB	제거
기타 node (7→4개)	347MB	330MB	
소계	2362MB	431MB	-82%
서버 전체			
used	2773MB	991MB	-64%
available	813MB	2603MB	+3.2배
earlyoom 마진	225MB	2015MB	+9.0배

접속 해제 후 10분 이내에 VS Code Server 전체가 종료되어 **2,362MB 회수**(RAM의 60%) 확인.

5. 대안 검토

5.1 SSH FS 확장

“ URL: <https://marketplace.visualstudio.com/items?itemName=Kelvin.vscodesshfs> GitHub: <https://github.com/SchoofsKelvin/vscode-sshfs>

서버에 아무것도 설치하지 않고 순수 SSH/SFTP 프로토콜로 동작하는 경량 확장이다.

- 서버 프로세스: 0개
- 서버 RAM 사용: 0MB
- 파일 탐색기: O (VS Code 내 Explorer 통합)
- 터미널: O (원격 터미널 지원)
- IntelliSense/디버깅: X (미지원)

Microsoft Q&A에서도 Remote-SSH의 서버 부담 문제에 대한 대안으로 SSH FS가 권장되고 있다:

“ URL: <https://learn.microsoft.com/en-us/answers/questions/5565268/visual-studio-code-extension-of-ssh-connection-oth>

UNSW CSE 학과에서는 학생 서버 보호를 위해 Remote-SSH 대신 SSH FS 사용을 공식 권장한다:

“ URL: <https://cgi.cse.unsw.edu.au/~learn/homecomputing/sshfs-remote/> 인용: "Remote-SSH 플러그인은 계정을 가득 채우고 서버를 느리게 만들 수 있으므로, 대신 SSH FS를 사용하라."

5.2 비교

방식	서버 프로세스	서버 RAM	IntelliSense	파일탐색	터미널	서버 설치물
Remote-SSH (기본)	6~13개	431MB~2.3GB	O	O	O	VS Code Server (~1GB 디스크)
Remote-SSH (경량화 완료)	4~6개	257~431MB	△	O	O	VS Code Server

방식	서버 프로세스	서버 RAM	IntelliSense	파일탐색	터미널	서버 설치물
						(~869MB 디스크)
SSH FS	0개	0MB	X	O	O	없음
순수 SSH 터미널	0개	0MB	X	X	O	없음

6. 결론 및 권장사항

6.1 운영서버에 Remote-SSH를 사용하는 것은 위험하다

Remote-SSH는 설계 자체가 "원격 서버를 로컬 개발 환경과 동일하게 사용"하는 것을 전제한다. 이 아키텍처는 개발 전용 VM이나 컨테이너에서는 합리적이지만, **운영 서버에서는 다음의 위험을 초래한다**:

1. **리소스 경합**: 운영 서비스(Apache, MySQL, PDF 엔진 등)와 VS Code Server가 동일한 RAM/CPU를 공유
2. **프로세스 잔류**: autoShutdown 실패 시 접속 해제 후에도 수백 MB~수 GB의 프로세스가 무기한 잔류
3. **확장 자동 설치**: 로컬 PC에서 설치한 확장이 서버에 자동으로 설치·실행됨. 개발자가 의도하지 않아도 발생
4. **OOM kill 경합**: earlyoom이나 커널 OOM killer가 운영 서비스 대신 VS Code를 kill하거나, 그 역으로 VS Code 때문에 운영 서비스가 kill될 수 있음

6.2 적용된 대응 (현재 상태)

현재 서버에는 Section 4의 모든 조치가 적용되어 있다:

- tsserver 힙 256MB 제한
- 불필요 확장 제거 + 재설치 방지
- 3중 방어 체계 (autoShutdown + cron + daily-cleanup)
- Machine settings 무결성 자동 복원

이로 인해 VS Code Server RSS가 2,362MB → 431MB(-82%)로 감소했으며, earlyoom 마진이 225MB → 2,015MB(+9.0배)로 확보되었다.

6.3 향후 검토

1. **SSH FS 전환 검토**: 파일 탐색 + 터미널만 필요한 경우 SSH FS로 전환하면 서버 부담이 완전히 제거됨
2. **운영서버 접속 가이드 문서화**: Remote-SSH 접속 시 주의사항, 금지 확장 목록, 설정 기준값을 팀 위키에 게시
3. **VS Code Server 완전 제거 명령**: 비상 시 `Remote-SSH: Uninstall VS Code Server from Host...` 명령 또는 수동 정리 절차 문서화

7. 참고 자료 모음

Microsoft 공식 문서

문서	URL
Remote-SSH 공식 가이드	https://code.visualstudio.com/docs/remote/ssh
Remote Development Overview	https://code.visualstudio.com/docs/remote/remote-overview
Extension Host 아키텍처	https://code.visualstudio.com/api/advanced-topics/extension-host
Supporting Remote Development (확장 개발자용)	https://code.visualstudio.com/api/advanced-topics/remote-extensions
Remote Development FAQ	https://code.visualstudio.com/docs/remote/faq
Remote Development Tips & Tricks	https://code.visualstudio.com/docs/remote/troubleshooting
VS Code Server 설명	https://code.visualstudio.com/docs/remote/vscode-server

GitHub 이슈

이슈 번호	제목	URL
vscode#151205	Remote SSH RAM 과다 점유	https://github.com/microsoft/vscode/issues/151205
vscode#194583	확장 실행 위치 변경 UI 요청	https://github.com/microsoft/vscode/issues/194583
vscode-remote-release#9778	vscode-server 메모리 누수	https://github.com/microsoft/vscode-remote-release/issues/9778
vscode-remote-release#7825	fileWatcher 메모리 고갈	https://github.com/microsoft/vscode-remote-release/issues/7825
vscode-remote-release#3319	tsserver 고 CPU (t2.micro)	https://github.com/microsoft/vscode-remote-release/issues/3319
vscode-remote-release#3195	Node 리소스 과다 소비 (BSOD)	https://github.com/microsoft/vscode-remote-release/issues/3195
vscode-remote-release#10567	비정상 메모리/CPU 크래시	https://github.com/microsoft/vscode-remote-release/issues/10567
vscode-remote-release#2716	extensionHost CPU 99%	https://github.com/microsoft/vscode-remote-release/issues/2716
vscode-remote-release#1656	다중 인스턴스 CPU 과부하	https://github.com/microsoft/vscode-remote-release/issues/1656
vscode-remote-release#9454	사전 설치 서버/확장 요청	https://github.com/microsoft/vscode-remote-release/issues/9454

대안 도구

도구	URL
SSH FS (VS Code 확장)	https://marketplace.visualstudio.com/items?itemName=Kelvin.vscode-sshfs
SSH FS GitHub	https://github.com/SchoofsKelvin/vscode-sshfs
Microsoft Q&A: Remote-SSH 대안	https://learn.microsoft.com/en-us/answers/questions/5565268/
UNSW CSE SSH FS 권장 가이드	https://cgi.cse.unsw.edu.au/~learn/homecomputing/sshfs-remote/