PR #23279 完整报告

sgl-project/sglang

[CI] Fix nightly docker builds failing on root-owned workspace leftovers

合并时间：2026-04-21 02:56

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23279

执行摘要

- 一句话：修复自托管 CI 节点因 root 权限残留文件导致的 Docker 构建工作流失败。
- 推荐动作：该 PR 是典型的 CI 基础设施修复，逻辑简单直接。对于负责 CI/CD 的工程师，值得快速浏览以了解自托管 runner 共享工作空间时的权限处理模式。关注点在于 sudo rm -rf 的使用场景和与 pr-test.yml 的现有方案的一致性。无需深入源码级分析。

功能与动机

PR body 明确指出，nightly 的 'Build and Push Development Docker Images' 工作流（以及相关兄弟工作流）在 actions/checkout@v4 步骤失败，错误为 'EACCES: permission denied, rmdir '.../sgl-kernel/build/.cmake''。根本原因是自托管 runner（如 labubu）同时携带 arm-docker-build-node 和 arm-kernel-build-node 标签（x64 节点同理），这些工作流共享同一个 _work 目录。sgl-kernel/build.sh 在容器内以 root 身份运行，导致生成的文件在宿主机上归 root 所有，后续工作流无法删除这些文件。pr-test.yml 中已经为 kernel-build 作业处理了此问题，本 PR 将相同的 sudo rm -rf 步骤镜像到所有 docker-build-node 工作流中，防止它们互相干扰。

实现拆解

1. 识别问题工作流：分析哪些工作流使用 x64-docker-build-node 或 arm-docker-build-node 标签，并可能因共享工作空间而失败。确定了 5 个目标文件：release-docker-dev.yml（nightly 构建）、release-docker.yml（标签发布）、release-docker-runtime.yml（runtime 镜像发布）、trivy-scan-dev.yml（漏洞扫描）、patch-docker-dev.yml（手动补丁工作流）。
2. 添加清理步骤：在每个目标工作流的 jobs 中，在 actions/checkout@v4 步骤之前，插入一个名为“Cleanup workspace (remove root-owned files from prior runs)”的步骤，执行 sudo rm -rf "$GITHUB_WORKSPACE"/* || true。这确保了工作空间目录在检出前被彻底清空，即使有 root 所有权的残留文件也能被删除。
3. 保持一致性：清理步骤的添加位置和命令与 pr-test.yml 中已有的处理方式（第 431 行和第 479 行）保持一致，确保解决方案的统一性。
4. 排除非目标工作流：create-manifests 作业运行在 ubuntu-22.04（GitHub 托管）上，不共享自托管 runner 的工作空间，因此不需要此步骤。

5. 无测试或配置配套改动：此变更仅涉及 CI 工作流配置，不涉及源码、测试或部署配置的修改。

关键文件：

- .github/workflows/release-docker-dev.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：这是 nightly Docker 镜像构建的主要工作流，直接修复了报告中提到的失败问题。
- .github/workflows/release-docker.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：处理标签发布的 Docker 镜像构建，同样使用 docker-build-node runner，需要相同的修复。
- .github/workflows/release-docker-runtime.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：涉及 runtime 镜像的发布工作流，也使用 docker-build-node runner，修复确保一致性。
- .github/workflows/trivy-scan-dev.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：每日漏洞扫描工作流，使用 docker-build-node runner 进行镜像扫描，需要清理以避免失败。
- .github/workflows/patch-docker-dev.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：手动补丁工作流，同样使用 docker-build-node runner，添加清理以确保稳定性。

关键符号：未识别

评论区精华

由于 review 评论为空，没有具体的讨论交锋。PR body 中已详细解释了问题根因和解决方案，并提供了测试计划。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 权限风险：sudo rm -rf 命令具有破坏性，如果路径变量错误或命令被误用，可能删除非预期文件。但此处使用 "$GITHUB_WORKSPACE"/* 并添加 || true，限制了删除范围并忽略错误，风险可控。
  2. 兼容性风险：变更仅影响特定标签的 CI runner，对源码逻辑、运行时行为或用户 API 无影响。
  3. 性能风险：每次运行都执行清理可能增加少量开销，但相对于构建时间可忽略，且避免了因失败导致的重试成本。
  4. 安全风险：无新增安全漏洞，清理的是临时工作空间文件。
- 影响：
  1. 对系统影响：修复了 CI 流水线的稳定性问题，确保 Docker 镜像构建工作流能可靠运行，减少因权限问题导致的构建失败。
  2. 对用户影响：最终用户无感知，但开发者能获得更稳定的 nightly 镜像和发布流程。

3. 对团队影响：减少了 CI 维护负担，避免了因残留文件导致的调试时间。
4. 影响范围：仅限于使用自托管 docker-build-node runner 的 CI 工作流，不影响其他测试或部署流程。 - 风险标记：权限操作风险，CI 稳定性依赖

关联脉络

- PR #23208 [CI] Partition stage-a-test-cpu into 4 matrix shards: 同为 CI 基础设施优化，涉及工作流配置调整，但解决的是测试超时问题而非权限问题。
- PR #23247 [AMD] Fix multimodal timeout issue：rocm7.2 PR Test: 涉及 CI 工作流修复（调整分区数），但针对 AMD 测试而非 Docker 构建。
- PR #23213 wait for reap in kill_process_tree: 同为 bugfix，但解决的是资源竞争条件，而非 CI 权限问题。