PR #7315 完整报告

PaddlePaddle/FastDeploy

[CI] Ensure container cleanup after job to avoid resource leakage

合并时间：2026-04-10 22:32

原文链接：http://prhub.com.cn/PaddlePaddle/FastDeploy/pull/7315

执行摘要

本 PR 通过在所有主要 CI workflow 中添加容器清理步骤、移除不必要特权参数，并优化依赖安装方式，旨在解决资源泄漏问题，提升 CI 稳定性和安全性。变更影响 13 个文件，重点关注清理逻辑设计和安全权衡，建议 CI 维护者精读以借鉴优化策略。

功能与动机

动机源于 CI 管道在作业取消或失败时可能留下运行容器或未清理工作空间，导致资源泄漏、工作空间冲突和不稳定性。PR body 明确指出："The CI pipeline may leave behind running containers or uncleaned workspaces when jobs are canceled or fail unexpectedly." 此外，移除 --privileged 参数以减少安全风险，因为当前工作流无需该特权。

实现拆解

实现按模块拆解如下：

- CI Workflow 清理步骤：在 12 个 .github/workflows/ 下的 YAML 文件（如 _accuracy_test.yml、_base_test.yml）中添加统一步骤，使用 if: always() 条件确保作业后执行 docker exec -t ${{ runner.name }} /bin/bash -c 'find /workspace -mindepth 1 -delete' 和 docker rm -f ${{ runner.name }}，清理工作空间和容器。
- 安全性改进：从所有 docker run 命令中移除 --privileged 参数，仅保留 --cap-add=SYS_PTRACE，示例变更： ```yaml docker run --rm --net=host \
- --cap-add=SYS_PTRACE --privileged --shm-size=64G \
- --cap-add=SYS_PTRACE --shm-size=64G \ ```
- 依赖管理优化：在 scripts/run_pre_ce.sh 中，将直接安装 xgrammar==0.1.19 和 torch==2.6.0 替换为使用预构建 wheel 文件，从特定 URL 下载以加速 CI 环境设置。

评论区精华

review 讨论中最有价值的交锋集中于 --rm 参数的设计权衡：

> fastdeploy-bot 指出："docker run 使用了 --rm 参数，导致容器在退出时自动删除。后续清理步骤中的 docker rm -f ${{ runner.name }} 会因容器已不存在而失败，且 docker exec 在容器退出后无法执行工作空间清理。建议移除 --rm 参数。"

结论是采纳该建议，移除 --rm 以确保显式清理逻辑生效，这体现了在自动化清理与容器生命周期管理之间的重要设计决策。

风险与影响

风险：

- 清理步骤依赖 Docker 命令，若执行失败（如权限不足或网络问题）可能导致残留。
- 预构建 wheel 文件从外部 URL 安装，增加了网络依赖和版本锁定风险，若资源不可用可能中断 CI。
- 移除 --rm 后，如果清理步骤未触发，容器可能永久残留，需监控清理成功率。

影响：

- 对系统：减少 CI 资源泄漏，提升作业隔离性和稳定性，预期降低因残留容器导致的故障率。
- 对安全：移除 --privileged 降低攻击面，符合安全最佳实践。
- 对团队：提高开发效率，减少 CI 失败后的手动清理工作，影响范围限于基础设施层。

关联脉络

从历史 PR 看，本 PR 与近期多个 CI 优化 PR 形成脉络：

- PR #7289 修复预构建 wheel 安装，与本 PR 的依赖管理优化相关联。
- PR #7283 添加 no_proxy 配置，同属 CI 网络和稳定性改进系列。 这些 PR 共同反映了团队持续投入 CI 基础设施的健壮性优化，以支持 FastDeploy 项目的高频集成和测试需求。