

PR #34521 完整报告

sgl-project/sglang

[AMD] Publish an undated tag for the miles nightly images

合并时间: 2026-08-12 14:27

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/34521>

执行摘要

本 PR 为 AMD miles 系列 ROCm nightly 镜像发布流程增加了无日期标签 `rocm/sgl-dev:miles-<variant>`, 让文档和用户无需拼接日期即可稳定引用最新构建。改动仅涉及两个 GitHub Actions workflow 各 3 行, 通过直接 `docker push` 本地已有标签实现, 不触发重建, 也不影响任何运行时代码。属于低风险、低复杂度的 CI 发布流程改进。

功能与动机

AMD miles 夜间镜像此前只发布带日期的 tag, 如 `rocm/sgl-dev:miles-<variant>-${DATE}`, 导致外部没有稳定名字可以引用。PR body 原话: The AMD nightly images are published only under a dated tag, so nothing can reference them stably. 提交信息也强调文档和用户不得不先查当前日期才能拉取最新构建。本 PR 的目标就是让两个 nightly workflow 在推送日期标签的同时, 把无日期标签指向同一个构建。

实现拆解

1. 变更入口: `.github/workflows/release-docker-amd-miles-rocm700-nightly.yml` 与 `.github/workflows/release-docker-amd-miles-rocm720-nightly.yml`, 两个文件结构对称, 各在 `Push dated tag` 步骤之后追加一个 `Push the undated tag` 步骤。
2. 最初方案 (第一个 commit): 用 `docker buildx imagetools create` 从刚推送的日期标签创建 `rocm/sgl-dev:miles-<variant>`。好处是只写 registry 元数据, 无重建、无 layer 上传, 且天然保证两种标签解析到同一 digest。
3. 最终方案 (第二个 commit): 改为 `docker push rocm/sgl-dev:miles-${{ matrix.variant }}`。因为构建阶段 `docker/build.py --image-tag custom --custom-tag miles` 已在本地给镜像打上无日期标签, 后续才用日期重新打标; 直接 push 更精准, 且避免了 `imagetools create` 默认 `--prefer-index=true` 在单平台镜像上产生“image index 包装 manifest”的层级问题。
4. 配套改动: 无。没有测试、schema、运行时配置或部署脚本联动, 这是一次纯 CI workflow 调整。

本 PR 的变更是两行 CI workflow YAML 步骤, 没有可展开的源码实现单元。其核心逻辑等价于: 在既有的 `docker push rocm/sgl-dev:miles-<variant>-${DATE}` 之后, 追加 `docker push rocm/sgl-dev:miles-<variant>`, 利用本地已有的镜像标签完成无日期标签发布。

评论区精华

PR 没有 review 评论，reviewer [bingxche](#) 直接 APPROVE。设计权衡体现在两次提交的演进中：

第一次提交：`docker buildx imagetools create` 只写 manifest 引用，不重建也不传 layer，从日期标签创建可保证两个名字解析到同一 digest。

第二次提交：`imagetools create` 默认 `--prefer-index=true`，而这些变体是单平台镜像，会把无日期标签发布成 image index 包装 manifest 的形式；改为 `docker push` 更准确，也更简单。

这一来一回展示了“最小 registry 操作”与“保持发布对象形态一致”之间的取舍，最终选择了语义更直接的方案。

风险与影响

- 无日期标签会被每次 nightly 构建覆盖，只能表达“最新”语义，无法作为历史快照引用；若某次构建在 push 前失败，标签可能停留在旧镜像上。
- 风险面仅限两个 CI workflow，不触及推理、调度、内核等任何运行时路径；单平台 image index 的隐患已通过改用 docker push 消除。
- 对用户和文档的收益是稳定引用名；对团队的维护成本是后续需保持 rocm700 与 rocm720 两个 workflow 对称演进。

关联脉络

本 PR 属于持续进行的 AMD/CI 基础设施维护。仓库近期有 [\[CI\] Align rerun-test environment with the test stages](#) (PR#34195) 对 GitHub Actions 环境变量做对齐，以及 [\[diffusion\] fix: nightly diffusion benchmark passes the retired --warmup flag](#) (PR#34423) 修复 nightly 脚本参数，说明 CI 与 nightly 发布链路正在被系统性梳理。本 PR 补上了镜像发布中“稳定引用名”这一环，与此方向一致。