

PR #40415 完整报告

vllm-project/vllm

[CI] Automate Docker Hub release image publishing

合并时间: 2026-05-06 08:15

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/40415>

执行摘要

- 一句话: 自动化 Docker Hub 发布镜像流程
- 推荐动作: 建议 DevOps 和 CI 工程师精读此 PR, 尤其是 review 中关于 shell 脚本错误处理、竞争条件和依赖管理的讨论。这些模式在自动化发布流程中非常典型, 值得借鉴。发布负责人应关注 PyPI 上传问题是否被后续修复。

功能与动机

以前发布 Docker 镜像需要人工从 Buildkite 注解中复制命令执行, 容易出错且遗漏了 Ubuntu 24.04 镜像。`docker manifest rm` 在首次发布时因缺少 `ll true` 导致失败。本 PR 旨在完全自动化发布流程, 减少人工操作失误, 并统一镜像发布方式。

实现拆解

1. 创建自动化发布脚本 (`.buildkite/scripts/publish-release-images.sh`): 180 行的 Bash 脚本, 负责从 ECR 拉取已构建好的各架构镜像, 用 `latest` 和 `v$VERSION` 标签重新标记, 推送到 Docker Hub, 然后创建多架构 manifest 并推送。脚本支持 CUDA 13.0 (默认)、CUDA 12.9、Ubuntu 24.04 (两种 CUDA 版本)、ROCm 和 CPU 变体。CPU 镜像处理使用 `docker manifest inspect` 检查是否存在, 若不存在则跳过并警告, 避免静默失败。
2. 调整 Buildkite Pipeline 配置 (`.buildkite/release-pipeline.yaml`): 在 `Publish release images` 分组之前新增一个 block 步骤 `block-publish-release-images`, 要求操作员手动确认后才能进行发布。该 block 依赖于所有多架构 manifest 创建步骤、ROCm 构建和发布版本输入, 同时通过 `allow_failure: true` 依赖两个 CPU 构建步骤, 等待它们完成但不阻塞整个 pipeline。实际发布步骤 `publish-release-images-dockerhub` 在 block 解封后执行, 调用上述脚本。此外还为两个 CPU 构建步骤添加了 `key:` 标识, 便于依赖引用。
3. 简化注解脚本 (`.buildkite/scripts/annotate-release.sh`): 删除了原先打印 Docker `pull/tag/push/manifest` 命令的 93 行内容, 仅保留 wheel 下载说明和一条提示 (说明镜像发布已自动化)。原脚本中 `ROCM_BASE_CACHE_KEY` 等不再需要, 一并移除。
4. 修复 PyPI 上传脚本依赖 (`.buildkite/scripts/upload-release-wheels-pypi.sh`): 在创建 `sdist` 前安装 `requirements/build/cuda.txt`, 确保 `setup.py` 的运行依赖 (如 torch) 可用。但这一改动在 review 中被指出可能引入 CUDA 工具链依赖冲突, 尚未完全解决。

关键文件:

- `.buildkite/scripts/publish-release-images.sh` (模块 发布脚本; 类别 `infra`; 类型 `core-logic`) : 新增的自动化发布脚本核心, 实现了从 ECR 拉取、打标签、推送、创建 manifest 的完整流程, 支持所有镜像变体。
- `.buildkite/release-pipeline.yaml` (模块 Pipeline 配置; 类别 `config`; 类型 `configuration`) : Pipeline 配置新增了 `block gate` 和自动化发布步骤, 调整了 CPU 构建步骤的依赖关系, 是自动化发布能在 CI 中正确运行的关键配置变更。
- `.buildkite/scripts/annotate-release.sh` (模块 注解脚本; 类别 `infra`; 类型 `core-logic`) : 移除原先打印 Docker 命令的 93 行内容, 仅保留 `wheel` 下载信息, 简化了发布注解的输出。
- `.buildkite/scripts/upload-release-wheels-pypi.sh` (模块 上传脚本; 类别 `infra`; 类型 `core-logic`) : 添加了 `sdist` 构建依赖安装, 但 review 中指出该改动可能引入 CUDA 工具链依赖问题, 后续需关注。

关键符号: 未识别

关键源码片段

`.buildkite/scripts/publish-release-images.sh`

新增的自动化发布脚本核心, 实现了从 ECR 拉取、打标签、推送、创建 manifest 的完整流程, 支持所有镜像变体。

```
#!/bin/bash
# SPDX-License-Identifier: Apache-2.0

set -euo pipefail

# 获取发布版本号, 并去掉可能的 v 前缀
RELEASE_VERSION=$(buildkite-agent meta-data get release-version --default "" | sed 's/^v//')
if [ -z "${RELEASE_VERSION}" ]; then
    echo "ERROR: release-version metadata not set"
    exit 1
fi

COMMIT="${BUILDKITE_COMMIT}"

# 登录 ECR
aws ecr-public get-login-password --region us-east-1 | docker login --username AWS --password-stdin public.ecr.aws/q9t5s3a7

# ---- CUDA 13.0 (default) ----
docker pull "public.ecr.aws/q9t5s3a7/vllm-release-repo:${COMMIT}-x86_64"
docker pull "public.ecr.aws/q9t5s3a7/vllm-release-repo:${COMMIT}-aarch64"

# 标记并推送 x86_64 镜像
docker tag "public.ecr.aws/q9t5s3a7/vllm-release-repo:${COMMIT}-x86_64" "vllm/vllm-openai:latest-x86_64"
docker tag "public.ecr.aws/q9t5s3a7/vllm-release-repo:${COMMIT}-x86_64" "vllm/vllm-openai:v${RELEASE_VERSION}-x86_64"
```

```

docker push "vllm/vllm-openai:latest-x86_64"
docker push "vllm/vllm-openai:v${RELEASE_VERSION}-x86_64"

# 标记并推送 aarch64 镜像
docker tag "public.ecr.aws/q9t5s3a7/vllm-release-repo:${COMMIT}-aarch64" "vllm/vllm-openai:latest-aarch64"
docker tag "public.ecr.aws/q9t5s3a7/vllm-release-repo:${COMMIT}-aarch64" "vllm/vllm-openai:v${RELEASE_VERSION}-aarch64"
docker push "vllm/vllm-openai:latest-aarch64"
docker push "vllm/vllm-openai:v${RELEASE_VERSION}-aarch64"

# 先清理旧 manifest（第一次失败时忽略）
docker manifest rm "vllm/vllm-openai:latest" || true
docker manifest rm "vllm/vllm-openai:v${RELEASE_VERSION}" || true

# 创建并推送多架构 manifest
docker manifest create "vllm/vllm-openai:latest" "vllm/vllm-openai:latest-x86_64" "vllm/vllm-openai:latest-aarch64"
docker manifest create "vllm/vllm-openai:v${RELEASE_VERSION}" "vllm/vllm-openai:v${RELEASE_VERSION}-x86_64" "vllm/vllm-openai:v${RELEASE_VERSION}-aarch64"
docker manifest push "vllm/vllm-openai:latest"
docker manifest push "vllm/vllm-openai:v${RELEASE_VERSION}"

```

评论区精华

- CPU 拉取错误处理：claude[bot] 指出原 `if docker pull ... 2>/dev/null; then` 会掩盖所有错误（标签不匹配、认证过期等），建议改用 `docker manifest inspect`。作者采纳，重写了 CPU 镜像检测逻辑。
- 版本号 v 前缀不一致：CPU 构建使用原始版本号打标签，而发布脚本会 `strip v` 再重新添加，导致不匹配。作者移除了多余的 `v strip`，现在统一使用带 `v` 的版本号。
- 发布与 CPU 构建的竞争条件：发布步骤一开始没有依赖 CPU 构建，若操作员同时放行，发布可能拉到未完成的 CPU 镜像。作者在 `block` 依赖中添加了 `build-cpu-release-image-x86/arm64` 并设置 `allow_failure: true`，确保等待构建完成。
- `set -euo pipefail` 元数据获取失效：`buildkite-agent meta-data get` 失败导致脚本无错误消息退出。作者通过添加 `--default ""` 参数解决。
 - PyPI 上传脚本依赖问题（未解决）：`claude[bot]` 指出安装 `requirements/build/cuda.txt` 会在无 CUDA 的环境中触发 CUDA 路径检测失败，建议安装最小依赖集或使用 `--no-deps` 等方式。该问题在 PR 合并时尚未修复。
 - CPU 镜像拉取的错误处理 (correctness): 作者已修改脚本，使用 `docker manifest inspect` 替代静默 `pull`，并明确区分镜像缺失与真实错误。
 - 版本号 v 前缀不一致 (correctness): 作者已删除多余的 `v strip`，保持版本号一致性。
 - 发布与 CPU 构建的竞争条件 (design): 作者添加了对 `build-cpu-release-image-x86/arm64` 的依赖（带 `allow_failure`），确保等待构建完成。
 - `set -euo pipefail` 导致 `metadata` 获取失败无诊断 (correctness): 作者使用了 `--default ""` 参数，避免 `get` 失败时退出。

- PyPI 上传脚本依赖问题 (other): PR 合并前未修复, 作者未回应, 该问题仍存在。

风险与影响

- 风险:
 - 版本号格式依赖: 虽然修复了 v 前缀问题, 但仍依赖于操作员在输入版本时的一致性。若输入不带 v 的版本, CPU 构建和发布脚本仍可能因标签不匹配失败 (但当前脚本已统一处理)。
 - CPU 镜像发布失败时静默跳过: 当 CPU 镜像不存在时, publish 脚本会跳过并发出 WARNING, 但不会导致 pipeline 失败。操作员可能忽略此警告, 导致版本发布缺失 CPU 镜像。
 - PyPI 上传依赖问题: upload-release-wheels-pypi.sh 中安装 cuda.txt 可能导致 sdist 构建失败, 从而影响 PyPI 发布流程。该问题在 PR 中未修复, 需后续关注。
 - 夜间构建过滤隐式依赖: publish 步骤过滤夜间构建依赖于 Buildkite 的 transitive filter 语义, 若 block 步骤被过滤掉则后续步骤自动消失, 但显式添加 if 条件会更安全。当前未显式添加, 存在未来因 pipeline 结构变化而意外被包含的风险。
- 影响:
 - 对发布流程: 从手动复制命令变为自动化执行, 大幅降低人为失误的概率。操作员只需在 block 步骤确认即可, 流程缩短且更可靠。
 - 对团队效率: 发布工程师无需在 Buildkite 和终端之间切换, 减少了约 5 分钟的机械操作时间。
 - 对系统影响: 无运行时影响, 仅影响 CI/CD 流水线。
 - 对镜像完整性: 增加了 Ubuntu 24.04 镜像的发布支持, 填补了之前的缺失。
 - 风险标记: CPU 镜像跳过风险, 版本号格式依赖, PyPI 依赖问题, 夜间构建过滤隐式依赖

关联脉络

- 暂无明显关联 PR