PR #34253 完整报告

sgl-project/sglang

[CI] Add output_tag input to the Patch Docker Image workflow

合并时间：2026-08-10 17:48

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34253

执行摘要

- 一句话：新增 output_tag，避免补丁镜像覆盖 nightly 发布
- 推荐动作：值得精读，尤其是 GitHub Actions 输入安全加固的写法：用 env 注入 inputs、case 做整串字符集校验、前缀白名单代替 denylist、持久 runner 上清理临时文件。若后续要扩展其他发布流程，这套模式可以直接复用。

功能与动机

PR body 指出：patch-docker-dev pushes the built image back to the tag it patched, so applying a PR patch onto dev overwrites the published nightly (and replaces its multi-arch manifest with an x64-only image). 因此需要独立的 output_tag 把补丁镜像发布到隔离名称空间，并顺带解决 tag 输入注入与持久 runner 残留 patch 文件的问题。

实现拆解

1. 输入与并发控制调整：在 .github/workflows/patch-docker-dev.yml 的 workflow_dispatch.inputs 中，将 image_tag 设为必填并修正描述（dev、dev-cu13、dev-cu12）；新增必填的 output_tag，要求以 patch- 开头（如 patch-myfeature）。concurrency.group 由 patch-docker-<image_tag> 改为 patch-docker-<image_tag>-<output_tag>，避免不同目标 tag 的构建互相取消。
2. 新增 tag 校验步骤：新增 Resolve and validate image tags step，通过 env 注入 IMAGE_TAG 和 OUTPUT_TAG，避免 ${{ }} 插值拼进脚本；validate_tag 用 case 模式做整串字符集校验（拒绝空串和非法字符）；再强制 OUTPUT_TAG 以 patch- 开头。这里用前缀白名单隔离 release-docker*.yml 的多架构发布 tag，而不是 denylist 现有名称，因为发布 tag 方案会演进。校验通过后把 BASE_IMAGE、OUT_IMAGE 写入 GITHUB_ENV。
3. 构建 / 推送步骤改造：pull base image 与生成 patch 改用 BASE_IMAGE、PR_NUMBERS 环境变量，推送目标改为 OUT_IMAGE；构建前清理 /tmp/patch-ctx，防止持久 runner 上残留 patch 被 COPY *.patch 拾取；所有 patch 为空时由 warning+SKIP_BUILD 改为 ::error 并 exit 1。
4. 步骤摘要与配套：push 步骤的 GITHUB_STEP_SUMMARY 输出 OUT_IMAGE，并明确提示 linux/amd64 only，不会保留 base 镜像的多架构 manifest；无测试配套，改动仅限 workflow YAML，需手动 dispatch 验证。

关键文件：

- .github/workflows/patch-docker-dev.yml（模块 发布流水线；类别 infra；类型 infrastructure；符号 validate_tag）：全部变更所在：新增 output_tag 输入、tag 校验与 env 注入、构建 / 推送目标从原 tag 切换到独立输出 tag，并清理残留 patch 与空 patch 失败语义。

关键符号：validate_tag

关键源码片段

.github/workflows/patch-docker-dev.yml

全部变更所在：新增 output_tag 输入、tag 校验与 env 注入、构建 / 推送目标从原 tag 切换到独立输出 tag，并清理残留 patch 与空 patch 失败语义。

```
on:
  workflow_dispatch:
    inputs:
      image_tag:
        # 基础镜像 tag，例如 dev、dev-cu13、dev-cu12
        description: "Base image tag to patch (e.g. dev, dev-cu13, dev-cu12)"
        required: true
      output_tag:
        # 发布目标 tag，必须以 patch- 开头；本 job 是 x64-only，
        # 不能覆盖 release-docker*.yml 生成的多架构发布 tag
        description: "Tag to publish as. Must start with 'patch-' (e.g. patch-myfeature)"
        required: true

jobs:
  patch:
    runs-on: x64-docker-build-node
    steps:
      - name: Resolve and validate image tags
        env:
          # inputs 通过 env 注入，避免把用户输入用 ${{ }} 拼接进 shell 脚本
          IMAGE_TAG: ${{ inputs.image_tag }}
          OUTPUT_TAG: ${{ inputs.output_tag }}
        run: |
          # case 模式逐字符校验字符集，空串和非法字符直接报错；
          # 正则 ^...$ 是逐行锚定，multi-line 输入会绕过，所以不能用它做整串校验
          validate_tag() {
            case "$2" in
              ""|[!a-zA-Z0-9_]*|*[!a-zA-Z0-9._-]*)
                echo "::error::${1} is not a valid Docker tag"
                exit 1
                ;;
            esac
          }
          validate_tag image_tag "${IMAGE_TAG}"
          validate_tag output_tag "${OUTPUT_TAG}"
```

```
# 用 patch- 前缀白名单与发布 tag 隔离，而不是 denylist 现有发布名，
# 因为发布 tag 方案会演进，denylist 会失效
case "${OUTPUT_TAG}" in
  patch-?*) ;;
  *)
    echo "::error::output_tag must start with 'patch-'"
    exit 1
    ;;
esac

echo "BASE_IMAGE=lmsysorg/sglang:${IMAGE_TAG}" >> "$GITHUB_ENV"
echo "OUT_IMAGE=lmsysorg/sglang:${OUTPUT_TAG}" >> "$GITHUB_ENV"
```

评论区精华

该 PR 无任何 review 评论，唯一审核人 Fridge003 直接 APPROVED。三笔提交体现了设计收敛：先引入 output_tag，再加固校验并强制 patch- 前缀，最后删除冗余检查和 SKIP_BUILD 分支、空 patch 时直接失败——最终实现比 PR body 描述更严格（body 称空 output_tag 保留原地刷新，实际已改为必填）。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 破坏性变更：output_tag 变为必填且必须 patch- 前缀，image_tag 也改为必填；任何依赖旧行为（把 patch 推到 dev 覆盖 nightly）的脚本或文档会立即失败。PR body 原本描述空 output_tag 保留原地刷新，但最终提交删除了该分支，存在文档与实现不一致的风险。
  2. x64-only 发布：构建在 x64-docker-build-node 上，推送到独立 tag 的镜像仍是单架构，若用户误用于 arm64 或多架构场景会失败；GITHUB_STEP_SUMMARY 有说明但无法强制。
  3. tag 校验盲区：validate_tag 只校验字符集和 patch- 前缀，未覆盖 Docker tag 的 128 长度上限以及开头不能是 . 或 - 等约束，极端输入会在 docker push 阶段才失败。
  4. 无自动化测试：workflow 逻辑无法在 PR CI 中验证，只能手动 dispatch 触发。 - 影响：影响面限于手动维护镜像发布的团队：触发 patch-docker-dev 时必须提供 image_tag 和符合 patch- 前缀的 output_tag。收益是保护 dev 等夜间发布镜像不被 x64-only 覆盖，同时提升工作流安全性；对其他运行时与推理路径无影响。 - 风险标记：必填 output_tag 破坏现有调用，x64-only 单架构镜像，原地刷新模式被移除，无自动化测试覆盖

关联脉络

- PR #34186 [CI] Key scheduled CUDA suites by runner_config instead of hand-written jobs: 同期 CI 基础设施层改造，将工作流调度改为注册式配置；与本 PR 同属 CI 工业化演进，但涉及不同 workflow 文件。

- PR #34231 [CI] Keep the torch compilation cache instead of wiping it on install: 同为 CI/Docker 安装流程加固，关注持久 runner 上缓存与残留文件的处理，思路与本 PR 清理 /tmp/patch-ctx 一致。