

# PR #47180 完整报告

vllm-project/vllm

[CI] Add TORCH\_NIGHTLY=1 build mode (run full suite on torch nightly)

合并时间: 2026-07-10 16:38

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/47180>

## 执行摘要

- 一句话: 新增 TORCH\_NIGHTLY=1 模式支持 PyTorch nightly 全量 CI
- 推荐动作: 本 PR 设计清晰, 采用图像交换避免管道复制, 值得 CI 维护者阅读。同时可作为一个如何实现 CI 变体配置的参考案例。

## 功能与动机

来自 PR body: 'Add an opt-in path to build and run the existing vLLM CI suite against the latest PyTorch nightly wheels, to produce a continuous "vLLM vs torch nightly" signal (e.g. a daily Buildkite scheduled build).'

## 实现拆解

1. 在 image\_build.sh 中新增 TORCH\_NIGHTLY=1 检测, 条件满足时委托给专用的 nightly 构建脚本 image\_build\_torch\_nightly.sh, 将镜像标记为常规标签, 从而复用整个 CI 管道。
2. 在 trigger-ci-build.sh 中添加 --torch-nightly 标志, 设置 TORCH\_NIGHTLY=1 并强制 RUN\_ALL=1, 便于手动触发 nightly 构建。
3. 在 check-ray-compatibility.sh 中根据 TORCH\_NIGHTLY 标志选择 nightly 或 stable 的 PyTorch 索引 URL, 避免兼容性检查失败。
4. 修改 CMakeLists.txt 在 TORCH\_NIGHTLY=1 时跳过 torch 版本不匹配警告, 因为 nightly 构建的版本并非预期的稳定版本。
5. 更新 docs/contributing/ci/update\_pytorch\_version.md 文档, 描述如何使用 TORCH\_NIGHTLY=1 标志进行 nightly 测试。

关键文件:

- .buildkite/scripts/trigger-ci-build.sh (模块 CI 脚本; 类别 other; 类型 core-logic): 核心入口脚本, 新增 --torch-nightly 标签和环境变量数组构建逻辑
- .buildkite/image\_build/image\_build.sh (模块 构建脚本; 类别 other; 类型 core-logic): 构建镜像的核心脚本, 新增 TORCH\_NIGHTLY 检测并委托给 nightly 构建脚本
- .buildkite/scripts/check-ray-compatibility.sh (模块 CI 脚本; 类别 other; 类型 core-logic): ray 兼容性检查脚本, 根据 TORCH\_NIGHTLY 使用不同 PyTorch 索引
- CMakeLists.txt (模块 构建系统; 类别 config; 类型 configuration): 构建系统配置, 新增 TORCH\_NIGHTLY\_BUILD 标志以跳过版本警告

- docs/contributing/ci/update\_pytorch\_version.md (模块 文档; 类别 docs; 类型 documentation) : 文档, 新增 TORCH\_NIGHTLY 用法的说明

关键符号: 未识别

## 关键源码片段

### [.buildkite/scripts/trigger-ci-build.sh](#)

核心入口脚本, 新增 --torch-nightly 标签和环境变量数组构建逻辑

```
# 默认不启用 torch nightly
TORCH_NIGHTLY=false

# 选项解析循环
while [[ $# -gt 0 ]]; do
    case $1 in
        --torch-nightly)
            # 启用 torch nightly 构建模式
            TORCH_NIGHTLY=true
            shift
            ;;
        # ... 其他选项 ...
    esac
done

# 构建环境变量数组, 用于传递给 Buildkite 构建
BUILD_ENV=("RUN_ALL=1" "NIGHTLY=1")
if [[ "$TORCH_NIGHTLY" == true ]]; then
    # 添加 TORCH_NIGHTLY=1 以触发夜间构建
    BUILD_ENV+=("TORCH_NIGHTLY=1")
fi

# 组装 Buildkite 命令
CMD=(bk build create --commit "${COMMIT}" --branch "${BRANCH}" --message "${MESSAGE}")
for env_var in "${BUILD_ENV[@]"; do
    CMD+=(--env "${env_var}")
done
```

### [.buildkite/image\\_build/image\\_build.sh](#)

构建镜像的核心脚本, 新增 TORCH\_NIGHTLY 检测并委托给 nightly 构建脚本

```
# 检测 TORCH_NIGHTLY=1 以构建基于 PyTorch nightly 的 CI 镜像
if [[ "${TORCH_NIGHTLY:-0}" == "1" ]]; then
    echo "--- :warning: TORCH_NIGHTLY=1 -- building base image on PyTorch nightly"
    SCRIPT_DIR="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
    # 委托给专门的 nightly 构建脚本, 并传递所有参数
    exec "${SCRIPT_DIR}/image_build_torch_nightly.sh" \
        "${REGISTRY}" "${REPO}" "${BUILDKITE_COMMIT}" "${BRANCH}" "${IMAGE_TAG}"
fi
```

## 评论区精华

两个主要讨论线程：

- prepare-torch-nightly.sh 未被使用：khluu 指出该文件未在任何地方被调用；atalman 回应称该文件在图像交换方法中已冗余，并已将其删除（在提交中移除）。
- trigger-ci-build.sh 的使用频率：khluu 对该脚本的实际用途提出疑问；atalman 承认该脚本主要用于 Simon 的手动触发，本次仅添加 --torch-nightly 标志作为便利，生产路径是标签和调度构建。
  - prepare-torch-nightly.sh 未被调用 (question): atalman 确认该文件因图像交换方法已冗余，已在提交中移除。
  - trigger-ci-build.sh 的使用频率 (question): atalman 同意该脚本主要用于手动触发，本次仅添加 --torch-nightly 标志作为便利，生产路径是准备标签和调度构建。

## 风险与影响

- 风险：风险较低。主要风险包括：nightly 镜像可能因依赖冲突或 torch bug 导致构建失败；check-ray-compatibility.sh 中索引 URL 选择逻辑需确保 TORCH\_NIGHTLY 未设置时行为正确；CMakeLists.txt 跳过警告可能掩盖非 nightly 构建的版本不匹配，但由于仅在明确设置 TORCH\_NIGHTLY=1 时跳过，风险可控。
- 影响：对用户无直接影响。对 CI 团队提供了新的测试维度，有助于提前发现 PyTorch 兼容性问题。对系统增加一个可选的构建模式，由环境变量或标签控制，不影响现有流程。
- 风险标记：新增可选 CI 变体，nightly 构建不稳定风险

## 关联脉络

- PR #382 Add TORCH\_NIGHTLY flag for full CI run against torch nightly: 关联 issue, 提出了 TORCH\_NIGHTLY 标志的原始需求。
- PR #46510 [Core] Update requirements to use PyTorch nightly wheels: 之前的草稿 PR, 本 PR 是其清理替代，修复了其中的 bug 并简化了实现。