

PR #2404 完整报告

radixark/miles

Install the kubernetes toolchain the charts and the k8s provider need

合并时间: 2026-08-12 11:26

原文链接: <http://prhub.com.cn/radixark/miles/pull/2404>

执行摘要

- 一句话: 镜像与 CI 预装 kubectl/helm/k8s 客户端, 为 kubernetes 功能铺路
- 推荐动作: 建议重点阅读 docker/install-kube-tools.sh 的下载校验与清理模式, 以及 pr-test.yml/docker-build.yml 中针对共享 runner 的锁等待与 --ignore-installed 处理; 这两处是后续 CI/ 镜像维护中容易复用的工程经验。整体不值得精读, 但作为 kubernetes 功能的前置基座值得留意后续消费它的 PR。

功能与动机

PR body 明确说明这是前置准备: **Nothing on main uses these yet; they land ahead of the kubernetes work so the image is ready when it does**。同时指出当前缺失会静默失败的三条路径: 缺少 `kubernetes_asyncio` 导致相关模块在收集阶段失败; 缺少 `helm` 时 chart 测试自我跳过并报告成功; 缺少 `kubectl` 时 launcher 无法驱动 pod 内命令。

实现拆解

1. 新增集中安装脚本 `docker/install-kube-tools.sh`: 固定 kubectl v1.36.3 与 helm v4.2.3, 从 dl.k8s.io 与 get.helm.sh 下载二进制 / 压缩包, 与官方 sha256 校验和比对后再通过 install 落到 /usr/local/bin, 并在同一 RUN 内清理临时文件。
2. 接入两个 Dockerfile: docker/Dockerfile 与 docker/Dockerfile.rocm 均新增 COPY `docker/install-kube-tools.sh /tmp/install-kube-tools.sh`, 再执行安装脚本并通过 `${TARGETARCH}` 传入架构, 随后 `rm -f` 删除脚本本体; ROCm 镜像额外补充 ARG `TARGETARCH` 声明。
3. 依赖清单 `requirements.txt`: 新增 `kubernetes_asyncio` (36.1.0)。作者在 PR body 中核对镜像内已安装的包集合, 确认 aiohttp、certifi、python-dateutil、pyyaml、six、urllib3 均已存在, 不会引入新的传递依赖。
4. 镜像重建触发链: docker-build.yml 与 pr-test.yml 的路径过滤加入 `docker/install-kube-tools.sh`, 保证将来升级工具版本时能触发镜像重建, 避免改脚本但镜像不重建的脱节。
5. CI runner 配套: `_run-cpu-ci.yml` 与 `pre-commit.yml` 通过 `azure/setup-helm@v4` 安装 helm v4.2.3; pr-test.yml 与 docker-build.yml 将 python3/pip3 安装改为缺失才执行, 并加上 `DPkg::Lock::Timeout=600` 等待 apt 锁、`pip3 install --ignore-installed typer` 规避 typer 0.27 与 rich 版本冲突。

配套验证：作者对 [radixark/miles:pr-2404](#) 镜像做 smoke test，确认三个工具版本正确、训练栈 (torch 2.11.0+cu130、sglang、ray、megatron-core) 未变、fast 测试 3322 passed、sglang 成功启动 Qwen2.5-0.5B-Instruct 完成推理。

关键文件：

- docker/install-kube-tools.sh (模块 部署脚本；类别 infra；类型 infrastructure)：新的安装脚本集中管理 kubectl 与 helm 的下载、sha256 校验、安装和清理，两个 Dockerfile 都调用它，是本 PR 的核心资产。
- .github/workflows/pr-test.yml (模块 CI 工作流；类别 infra；类型 infrastructure)：路径过滤加入 install-kube-tools.sh，并把 python3/pip3 安装改为按需 + 等待 apt 锁 + --ignore-installed，修复共享 runner 上的静默失败。
- .github/workflows/docker-build.yml (模块 CI 工作流；类别 infra；类型 infrastructure)：镜像构建工作流同样加入路径过滤与 python/typer 修复，是本 PR 触发链的关键一环。
- .github/workflows/_run-cpu-ci.yml (模块 CI 工作流；类别 infra；类型 infrastructure)：CPU CI 中添加 helm 安装，保证 chart 测试在 CI 中真实执行而不自我跳过。
- .github/workflows/pre-commit.yml (模块 CI 工作流；类别 infra；类型 infrastructure)：pre-commit 环境也安装 helm，保证本地校验与 CI 行为一致。
- docker/Dockerfile.rocm (模块 镜像构建；类别 infra；类型 infrastructure)：ROCm 镜像补上 ARG TARGETARCH 并调用安装脚本，保持与主镜像工具链一致。
- docker/Dockerfile (模块 镜像构建；类别 infra；类型 infrastructure)：主镜像调用安装脚本，把 kubectl/helm 打进镜像层。
- requirements.txt (模块 依赖清单；类别 config；类型 dependency)：新增 kubernetes_asyncio 依赖，供 k8s worker provider 使用；作者验证无新增传递依赖。

关键符号：未识别

关键源码片段

[docker/install-kube-tools.sh](#)

新的安装脚本集中管理 kubectl 与 helm 的下载、sha256 校验、安装和清理，两个 Dockerfile 都调用它，是本 PR 的核心资产。

```
#!/usr/bin/env bash
# 安装 kubectl 与 helm 到 /usr/local/bin 目录，固定版本并做 sha256 校验。
# PR#2404 为后续 kubernetes 工作提前准备镜像内工具链。
set -euxo pipefail

# 版本集中定义，升级时只改这里
KUBECTL_VERSION="v1.36.3"
HELM_VERSION="v4.2.3"

# 支持显式传入架构；Dockerfile 里通过 ARG TARGETARCH 传入
arch="${1:-}"
[ -n "$arch" ] || arch="$(dpkg --print-architecture)"

# kubectl 组件：下载二进制与校验和，校验通过后安装，失败立即中断
```

```
curl -fsSL -o /tmp/kubectI "https://dl.k8s.io/release/${KUBECTL_VERSION}/bin/linux/${arch}/kubectI"
curl -fsSL -o /tmp/kubectI.sha256 "https://dl.k8s.io/release/${KUBECTL_VERSION}/bin/linux/${arch}/kubectI.sha256"
echo "$(cat /tmp/kubectI.sha256) /tmp/kubectI" | sha256sum -c -
install -m 0755 /tmp/kubectI /usr/local/bin/kubectI

# helm 组件：下载 tar 包与校验和；awk 只取哈希字段，避免换行干扰校验
curl -fsSL -o /tmp/helm.tar.gz "https://get.helm.sh/helm-${HELM_VERSION}-linux-${arch}.tar.gz"
curl -fsSL -o /tmp/helm.tar.gz.sha256 "https://get.helm.sh/helm-${HELM_VERSION}-linux-${arch}.tar.gz.sha256sum"
echo "$(awk '{print $1}' /tmp/helm.tar.gz.sha256) /tmp/helm.tar.gz" | sha256sum -c -
tar -xzf /tmp/helm.tar.gz -C /tmp
install -m 0755 "/tmp/linux-${arch}/helm" /usr/local/bin/helm

# 清理临时文件，避免把校验和与压缩包带进镜像层
rm -rf /tmp/kubectI /tmp/kubectI.sha256 /tmp/helm.tar.gz /tmp/helm.tar.gz.sha256 "/tmp/linux-${arch}"

# 安装后立即验证，构建期就能发现装错
kubectI version --client
helm version --short
```

评论区精华

review 环节无实质讨论，[guapisolo](#) 直接 APPROVED，无评论内容。唯一有价值的补充来自作者在 issue 评论中的镜像 smoke test：验证 [kubectI](#) v1.36.3、[helm](#) v4.2.3、[kubernetes_asyncio](#) 36.1.0 均安装正确，且训练栈（torch、sglang、ray、megatron-core）未受影响，fast 测试通过、sglang 可正常服务模型。这直接回应了“镜像体积变大是否影响训练 / 推理路径”的潜在疑虑。

- 暂无高价值评论线程

风险与影响

- 风险：
 1. 新增两个构建期外部下载源（dl.k8s.io、get.helm.sh），虽有 pin 与 sha256 校验兜底，但上游发布策略变化可能拉长镜像构建时间或导致构建失败。
 2. requirements.txt 新增 kubernetes_asyncio 后，所有按该清单安装的环境都会引入它；作者确认无新增传递依赖，但安装期生成约 21 MiB __pycache__，会反映在镜像层尺寸上。
 3. 镜像体积增加约 38.7 MiB 拉取量（磁盘约 145-165 MiB），对 19.12 GB 整体镜像占比很小，但仍是长期膨胀的一部分。
 4. pr-test.yml 与 docker-build.yml 的 apt 锁处理只覆盖当前两个 job，其他 workflow 若复用同一共享 runner，仍可能遇到同类锁竞争。

5. kubectl/helm 版本固定后需要团队持续跟进安全更新；版本升级成本虽集中在一个脚本，但仍需维护。 - 影响：对用户：镜像体积略增但功能无感知，后续 kubernetes 相关功能可直接使用镜像内工具链。对系统：CI 构建链路新增两个下载与校验步骤，chart 测试不再自我跳过，CI 结果更可信，共享 runner 的锁竞争问题得到缓解。对团队：新增三个需要跟进版本的依赖与两个外部下载源；kubernetes 工作落地时无需再改镜像构建逻辑，路径过滤已保证版本变更会触发镜像重建。
- 风险标记：外部下载源新增，镜像体积增加，共享 runner 锁竞争，依赖版本维护，当前无运行时使用

关联脉络

- PR #2363 ci(docker): run image builds on docker-build runners: 本 PR 的第三个 commit 修复了 #2363 引入的 docker-build runner 上 python3/typer 安装与 apt 锁竞争问题，属于同一 CI 镜像构建基础设施演进线。