构建和发布版本

目录
概述
指令
步骤
1. 预构建检查
2. 构建发布包
3. 将发布包上传到 WordPress.org
4. 部署到测试环境
如果在监控期间检测到严重问题
5. 发表发布版本
6. 发布后的任务

:::important

虽然本指南可作为参考，但请记住，特定于版本的说明已在 版本跟踪问题中提供，这些问题是在发布周期之前创建的，并且应优先使用。

:::

本页面概述了从发布分支构建任何 WooCommerce
版本的必要步骤。请查看流程图以了解整体过程和决策表。以下提供了分步说明。

概述
发布流程图

指令
按照以下步骤的顺序执行。在运行任何 GitHub 工作流程时，请确保从 trunk 分支（默认分支）进行操作，并输入指定的版本或分支。

请随时准备好 发布故障排除与恢复 指南，以防遇到任何问题。

步骤

1. 预构建检查

- [ ] 确认 GitHub 服务 正在运行。
- [ ] 验证 发布里程碑 上是否存在任何未解决的问题或拉取请求。如有必要，请通知作者合并或关闭。
- [ ] 确保不存在任何 带有 "cherry pick failed" 标签 的拉取请求，这些拉取请求适用于此版本，并且尚未采取任何操作。
- [ ] 确认发布分支中 `readme.txt` 文件的 `Stable tag` 值与 WordPress.org 的 `trunk` 中的值相匹配。

2. 构建发布包

- [ ] 运行工作流 **Release: Bump version number**: 输入发布主版本 (`x.y`) 作为 *Release branch*，并从下拉菜单中选择发布类型。
- [ ] 审查并合并针对发布分支生成的 PR。
- [ ] 运行工作流 **Release: Compile changelog**: 输入发布主版本 (`x.y`) 作为 *Version*，并保持 *Release date* 为空，除非是提前构建发布包。
- [ ] 审查并合并生成的 PR: 一个针对 `trunk`，另一个针对发布分支。两者都应在发布里程碑下。
- [ ] 运行工作流 **Release: Build ZIP file** 以构建资源并创建 GitHub 发布: 输入发布主版本 (`x.y`) 作为 *Release branch*，并选中 *Create GitHub release*。
- [ ] 确认已在仓库中创建了草稿发布，并附带了 `woocommerce.zip` 资源 在仓库中。

3. 将发布包上传到 WordPress.org

- [ ] 运行工作流 **Release: Upload release to WordPress.org**: 输入发布版本 (`x.y.z`) 作为 *Release tag to upload*，并确保选中确认复选框。
- [ ] 确认 SVN 标签 存在于 WordPress.org SVN。
- [ ] 使用存储在密钥库中的 `WordPress.org "WooCommerce"` 用户帐户凭据，登录到 WordPress.org 并批准发布。
- [ ] 几分钟后，确认发布包 可供下载。

4. 部署到测试环境

:::caution 此步骤仅适用于 `rc` 或稳定版本 (`x.y.0` 及更高版本)。 :::

- [ ] 按照 部署到测试环境的指南 进行操作，并在部署后监控 4 小时（RC）或 2 小时（稳定版本）。
- [ ] 在发布 Slack 频道中创建一个线程，用于监控和讨论此发布。

如果在监控期间检测到严重问题

- [ ] 请求在测试环境中回滚。
- [ ] 暂停发布过程，并且 不要继续执行与此问题相关的任何步骤 。请遵循 故障排除指南_____ 中的步骤。

5. 发表发布版本

- [ ] （仅适用于稳定版本） 运行工作流程 **Release: Update stable tag**：输入发布版本 （`x.y.z`） 作为 *Version*，并确保勾选 确认 复选框。
- [ ] 发表之前创建的 发布草稿_____，以及可能存在的其他来自先前尝试的发布草稿。 确保 仅在稳定版本中勾选 "设置为最新发布 "。

6. 发布后的任务

:::caution 此步骤仅适用于稳定版本 （`x.y.0` 及更高版本 ）。 :::

- [ ] 至少等待 1 小时，确保所有自动流程完成，并合并所有与 发布里程碑_____ 相关的后续 PR。
- [ ] 继续监控与发布相关的错误，至少 3 天。请参阅 发布监控指南_____ 了解更多 详情。

版本号 #2
由 壳壳虫 创建于 2026-03-09 11:53:58 UTC
由 壳壳虫 更新于 2026-04-03 09:34:26 UTC