

## Challenge 2: Controlling deployments to production #4

[Edit](#) [New issue](#)[Open](#) 4 tasks jordan-sparks-po · bot · opened this issue 2 minutes ago · 1 comment

jordan-sparks-po · bot · commented 2 minutes ago

### Challenge 2: Controlling deployments to production

As a **Product Owner**, I want to establish a robust and secure workflow for our software development lifecycle, ensuring our release process is controlled and our main codebase protected. This setup will enable us to maintain high standards of quality and security —essential for staying ahead in today's fast-paced tech environment.

#### Why:

- **Controlled Releases:** By setting up a release pipeline with manual approvals, we ensure that every release is scrutinized and meets our quality benchmarks before going live. This need became evident after our public website experienced multiple downtimes due to direct pushes to production without sufficient oversight.
- **Protected Codebase:** Branch protection and restricted commit access safeguard our main branch from unreviewed changes, preventing potential disruptions in our production environment. This change is critical because in the past, direct changes led to significant downtime.
- **Focused Ownership:** Codeowners for specific sections of the project enhance accountability and ensure that only qualified team members make critical changes. The necessity of this became apparent after an incident where tickets were mistakenly sold on our test environment.
- **Clear history of changes:** Implementing a rebase strategy, we not only secure our codebase but also ensure that changes to the product are in an easy-to-read chronological order.

#### Acceptance Criteria:

- Release Pipeline Configuration:**
  - Implement a release pipeline using Deployment Environments in our project management tool.
  - Include a manual approval step before changes are deployed to production, ensuring every release is vetted for stability and compliance.
- Branch Protection Rules:**
  - Set up branch protection rules for the main branch to ensure it cannot be directly committed to. All changes must be made through pull requests reviewed according to the GitHub Flow.
  - This helps maintain code integrity and encourages peer review.
- Codeowners Setup:**
  - Establish Codeowners for the catalog section of our website. Only designated team members (the catalog team) should be able to update catalog files, ensuring expert oversight.
- Rebase Strategy Implementation:**
  - Adopt a rebase strategy for merging pull requests to keep our project history linear. This approach will simplify our commit history and make tracking changes more manageable.

#### Tasks:

- ☐ Configure the release pipeline with appropriate environments and manual approval steps.
- ☐ Implement branch protection rules to enforce pull request reviews and restrict direct commits to the main branch.
- ☐ Set up Codeowners for the catalog section, specifying team members in the repository settings.
- ☐ Enable a rebase strategy for merging pull requests to maintain a linear history in our repository.

"Efficiency is doing better what is already being done." — Peter Drucker



alex-fletcher-lead · bot · commented 2 minutes ago



Hey team, this challenge reminds me of a system I helped architect at a previous gig where we beefed up security while making our workflow more ninja-like—silent but deadly effective! 🥷

We're going to set up a solid, fail-safe release pipeline and tighten our codebase security. I've sketched out a plan on our Wiki inspired by best practices and past experiences with similar setups. We'll use Deployment Environments for a controlled release pipeline, and implement branch protection plus Codeowners to ensure only the right eyes and hands are on our precious code.

Check out the detailed strategy and initial setup on the [Wiki page](#). Let's lock this down:

- Need more help or have questions? `/help` is your go-to 🛠️.
- Want the nuts and bolts of each setup step? `/expert-tip` will break it down 📖.
- Ready to see if your implementation checks out? Punch `/verify` for my eagle eyes on it 🔍.
- If you want me to lead the charge on locking this down, just type `/fix` and consider it done 🛡️.

Oh! And don't forget, to proceed to a new challenge, type `/fixflag`. Emily will close the issue, so you don't need to worry about it. But before that, use `/fix` and apply the code provided to be ready for the next challenge.



Add a comment

Write Preview

H B I T `<>`

Add your comment here...

Markdown is supported

Paste, drop, or click to add files

Close issue

[Comment](#)

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).



© 2024 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact](#) [Manage cookies](#) [Do not share my personal information](#)