Migration runbook: Reporting Tools

💡 Feel free to add to this runbook to further clarify our process. If you uncover new information that may be relevant org-wide, please contribute back to the central migration notes

🚨 If you run into a blocking issue while migrating, reach out to me. If the fix is complicated enough to warrant another ticket, set the status of your repo in [the migration sheet](https://docs.google.com/spreadsheets/d/1-hIUCAMCpzbMfF5rA54-z9NDt3b2S8jXAPk_bENISG4/edit?usp=sharing) to "Blocked" and update the "Notes" section with context/new ticket slug.

Further reading:

Much of the runbook will be streamlined from the org-wide Migration Notes. This is done to limit the on-boarding necessary for Reporting Tools developers to participate in the migration project. However, if you find this document lacks sufficient information or justification, the above linked notes are helpful.

Runbook

Step 1: Prep work

Head over to the migration spreadsheet. Update "Jira Ticket" with a link to your migration ticket and the "Status" column to "In progress"

There are also a few tasks to complete to get ahead of a few speed bumps:

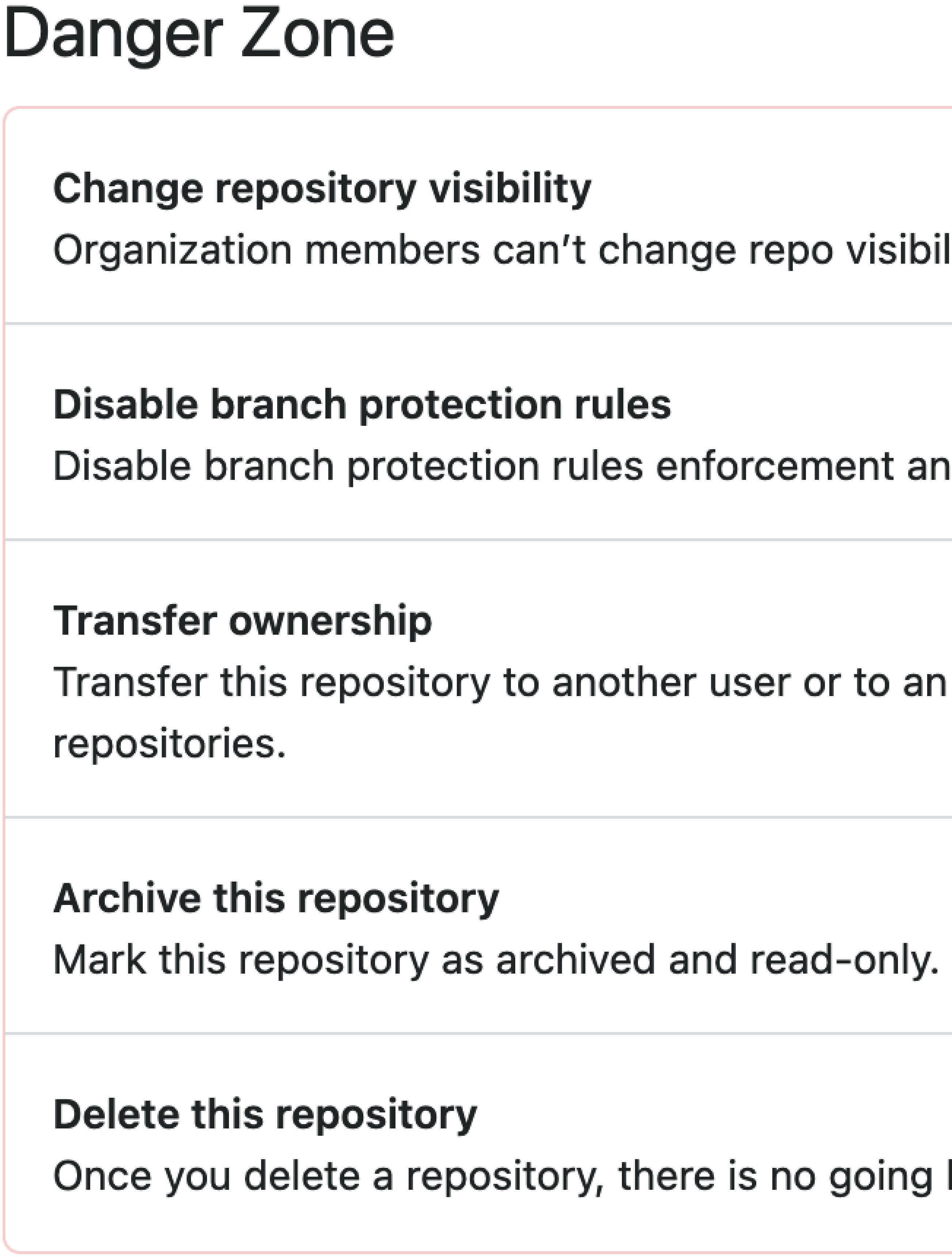
1. Add your personal account as an admin to the repo you're migrating ([Justification](https://arcpublishing.atlassian.net/wiki/spaces/WE/pages/3843457210/GitHub+Org+Migration+Notes#GitHub%E2%80%99s-Transfer-Utility)). **Being an admin by proxy of a** `**WPMedia**` **team is not enough.** **You will lose admin access to the repo unless you add your personal account.** While you're at it, take a screenshot of the access roster. In step 3 we will want to provision access for the `WPMedia` team's `WashPost`'s team analog. If you see infbot-news-engineering, [this doc](https://paper.dropbox.com/doc/Debate-Decide-Commit-aws-news-engineering-inf--CLbzJgiMXE~9ZQ~0ka0Hh9okAg-HvJWYG8bWoa3ogdwJX4IG) adds more context, but we can generally delete this safely for projects using the v2 pipeline template.

2. Identify whether your repo is referenced by an AWS CodePipeline configuration file in [the](https://github.com/WPMedia/aws-group-news-engineering)

`[AWS-group-news-engineering](https://github.com/WPMedia/aws-group-news-engineering)` [repo](https://github.com/WPMedia/aws-group-news-engineering). You can search for your repo's name within the `aws-group-news-engineering` repo. If your repo contains a `buildspec.yml` or `cfn/` folder, that also means it's most likely referenced by a CodePipeline somewhere. **Now, if the CodePipeline config uses a Stackjack template other than** `**pipeline-v2.template.yml**`**, you will have to migrate to the v2 template before you begin the migration process.** Otherwise, you will not be able to smoothly update the CodePipeline ([Justification](https://arcpublishing.atlassian.net/wiki/spaces/WE/pages/3843457210/GitHub+Org+Migration+Notes#Stackjack-CodePipeline)).

Step 2: Transfer

Use the GitHub Transfer utility located in your repo's "general" settings. This utility will do most of the heavy lifting of moving over branches, branch protections and settings. Full transfer utility documentation here.



Danger Zone

Change repository visibility

Organization members can't change repo visibil

Disable branch protection rules

Disable branch protection rules enforcement an

Transfer ownership

Transfer this repository to another user or to an
repositories.

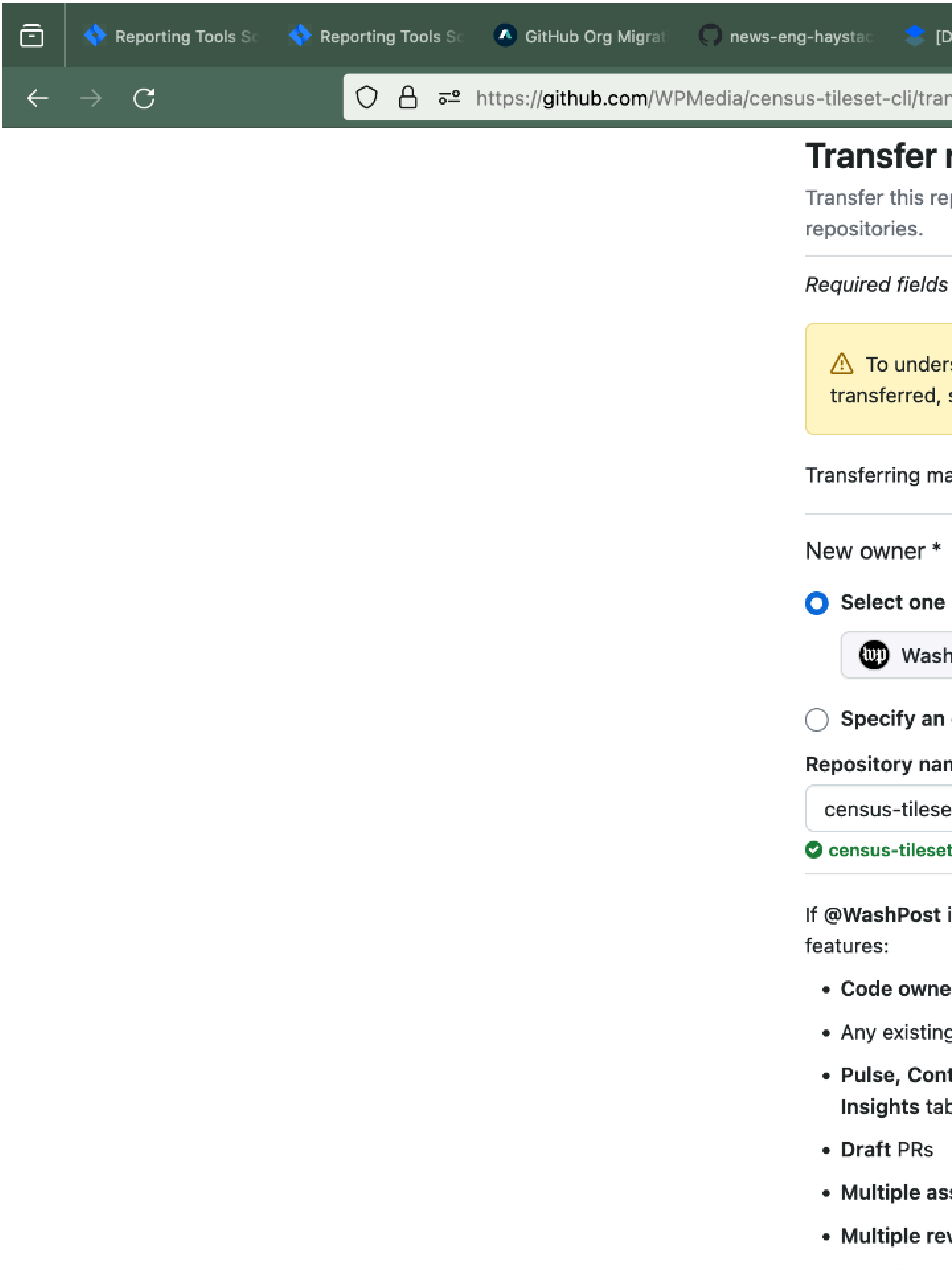
Archive this repository

Mark this repository as archived and read-only.

Delete this repository

Once you delete a repository, there is no going

Hitting “Transfer” will bring you to the following form page. **Do not hit the red button until you read the two warnings below.**



https://github.com/WPMedia/census-tileset-cli/tran

Transfer r

Transfer this rep
repositories.

Required fields

⚠ To unders
transferred, s

Transferring ma

New owner *

● **Select one**

Wash

○ **Specify an**

Repository nan

census-tilese

✓ census-tileset

If **@WashPost** i
features:

- **Code owne**
- Any existing
- **Pulse, Cont**
  Insights tab
- **Draft** PRs
- **Multiple as**
- **Multiple rev**

🚨 In the transfer form, you may elect to assign a new repository name. We've decided to strip `news-eng-*` in favor of shorter project names. rt-*` is also acceptable for projects with general-sounding names. Use your good discretion. If you are assigned repo that is named based on the news-eng org structure, please update the name when you transfer.

🚨 For "New owner", select "Select one of my organizations" and "WashPost" from the drop down menu. **Do not** transfer to the "washingtonpost" org. This is a different GitHub org. I repeat, "WashPost" not "washingtonpost".

> NOTE: If you don't see WashPost on your list of organizations to transfer to, you probably just need to re-authenticate. Browse to https://github.com/WashPost/ and you will be prompted to log in if your session has expired. 🤦‍♂️

Once you begin the transfer, it shouldn't take more than a few minutes for the repo to show up in the list of WashPost Repos.

Step 3: Navigate to the new `WashPost` repo and update the "Collaborators and teams" access settings.

As long as you added your personal account as an admin when the repo was still in `WPMedia`, you should have access to the `WashPost` version's settings.

Now, add the "Reporting Tools" team as an admin to the repo. **Then**, remove your personal account as an admin. (Justification).

In addition, refer back to the screenshot you took of the original repo's access setting. If there are any additional teams that need access to the new repo, add their WashPost org team analog.

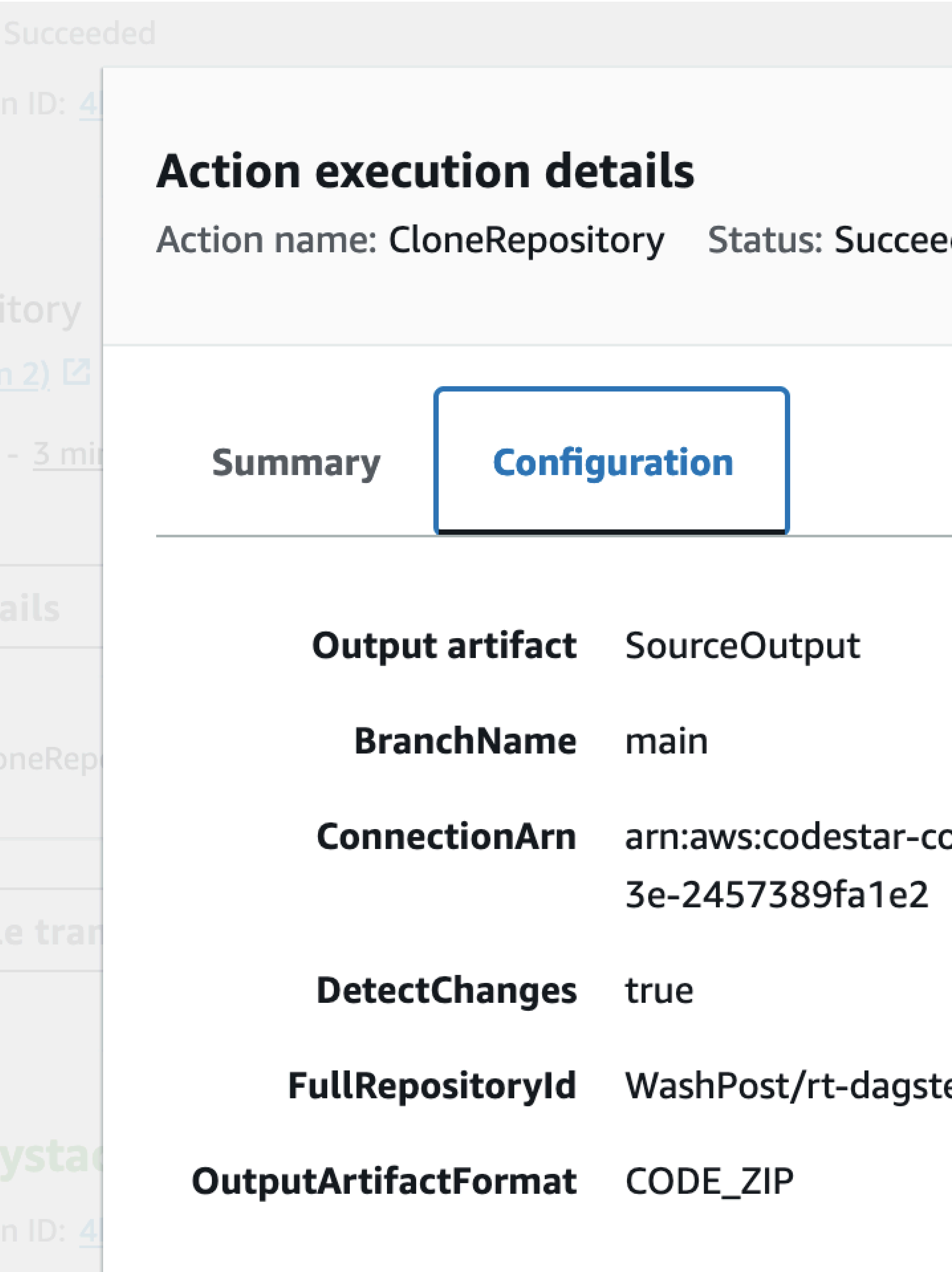
This work captures our generalized access management strategy for Reporting Tools repos. If you think the repo you're transferring should have a different access configuration, raise it with the team and implement something different.

Step 4: Update CodePipelines

If you identified your repo as being attached to a CodePipeline in Step 1, head back to `[aws-group-news-engineering](https://github.com/WPMedia/aws-group-news-engineering)` and submit/merge changes to update the CodePipeline config file to reference the new repo location.

💡 To reduce the chance of error, only update the git owner and git repo. An example PR.

🚨 Once the change is merged, watch the CodePipeline you just updated redeploy to confirm the connection to the new repo. An easy way to check is to hit the “View details” link and check out the configuration



Action execution details

Action name: CloneRepository Status: Succee

Output artifact SourceOutput

BranchName main

ConnectionArn arn:aws:codestar-co
3e-2457389fa1e2

DetectChanges true

FullRepositoryId WashPost/rt-dagste

OutputArtifactFormat CODE_ZIP

Step 5: Update CODEOWNERS, CODEOWNERS branch protections and references to WPMedia

Update CODEOWNERS file, CODEOWNERS branch protections and references to WPMedia

Our team uses a `[CODEOWNERS](https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/about-code-owners)` file to help provide granular restrictions on who can merge code. This is especially helpful in repos with cross-team activity. A `CODEOWNERS` file also enables our round-robin PR assignments.

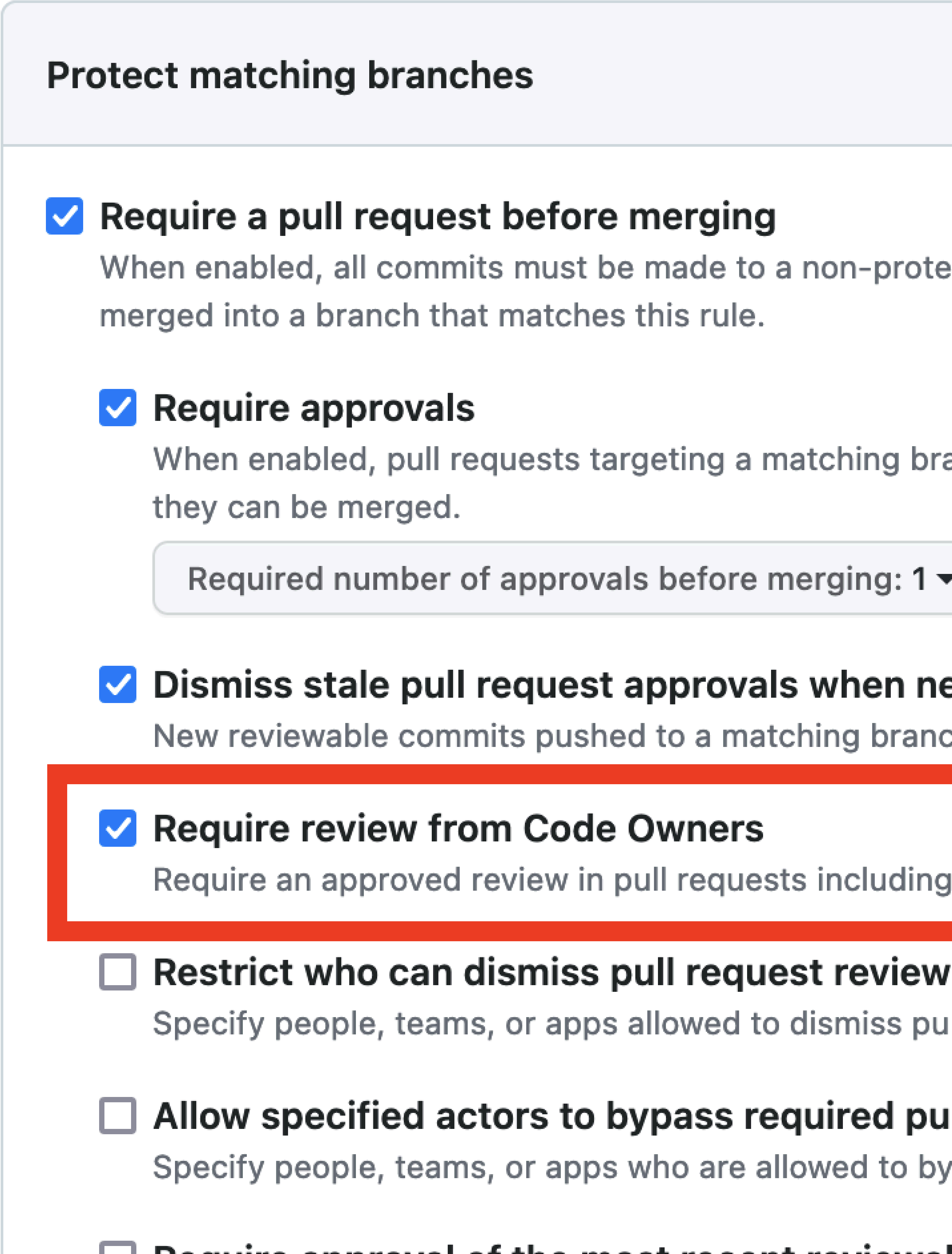
Update the `CODEOWNERS` file to ensure our `@WashPost/reporting-tools` team is listed as the primary code owner for the repo. You may elect to include additional restrictions on a directory or file-by-file basis.

💡 You should also replace references of `WPMedia` in favor of `WashPost` in the same PR.

This is a good opportunity to update your local origin for this repo. If you already have the original `WPMedia` version of this repo cloned locally, you will have to use a command to update the remote URL before you can push the change to the new repo: `git remote set-url origin <ssh address>`. If you have not already cloned the repo onto your local machine, you can simply `git clone <ssh address>` as normal.

🚨 A reminder: `git clone` or `git remote` with an HTTPS repo URL is no longer recommended. You now need to use `git clone git@github.com:WashPost/<repo name>.git`

We also recommend adding “Require review from Code Owners” to any protected branch in the repo.



Protect matching branches

☑ Require a pull request before merging

When enabled, all commits must be made to a non-prote
merged into a branch that matches this rule.

☑ Require approvals

When enabled, pull requests targeting a matching bra
they can be merged.

Required number of approvals before merging: 1 ▾

☑ Dismiss stale pull request approvals when ne

New reviewable commits pushed to a matching branc

☑ Require review from Code Owners

Require an approved review in pull requests including

□ Restrict who can dismiss pull request review

Specify people, teams, or apps allowed to dismiss pu

□ Allow specified actors to bypass required pu

Specify people, teams, or apps who are allowed to by

□ Require approval of the most recent reviewa

Step 6: Any additional repo configuration

Each repository has different development objectives. We can help facilitate them through intentional configuration of GitHub settings.

All existing settings will have been migrated over through GitHub's transfer utility. Still, take a few minutes to familiarize yourself with the objectives of the repo and consider whether or not to adjust the settings.

For example, you may want to add a setting that will delete a PR branch once it's merged, or implement a pull request template.

No need to go crazy here. These will be "while-we're-at-it" adjustments.

Step 7: Update GitHub migration doc

You're done! Wrap up the migration by updating our records.

In the GitHub migration doc we have four columns to collect relevant information about the migration:

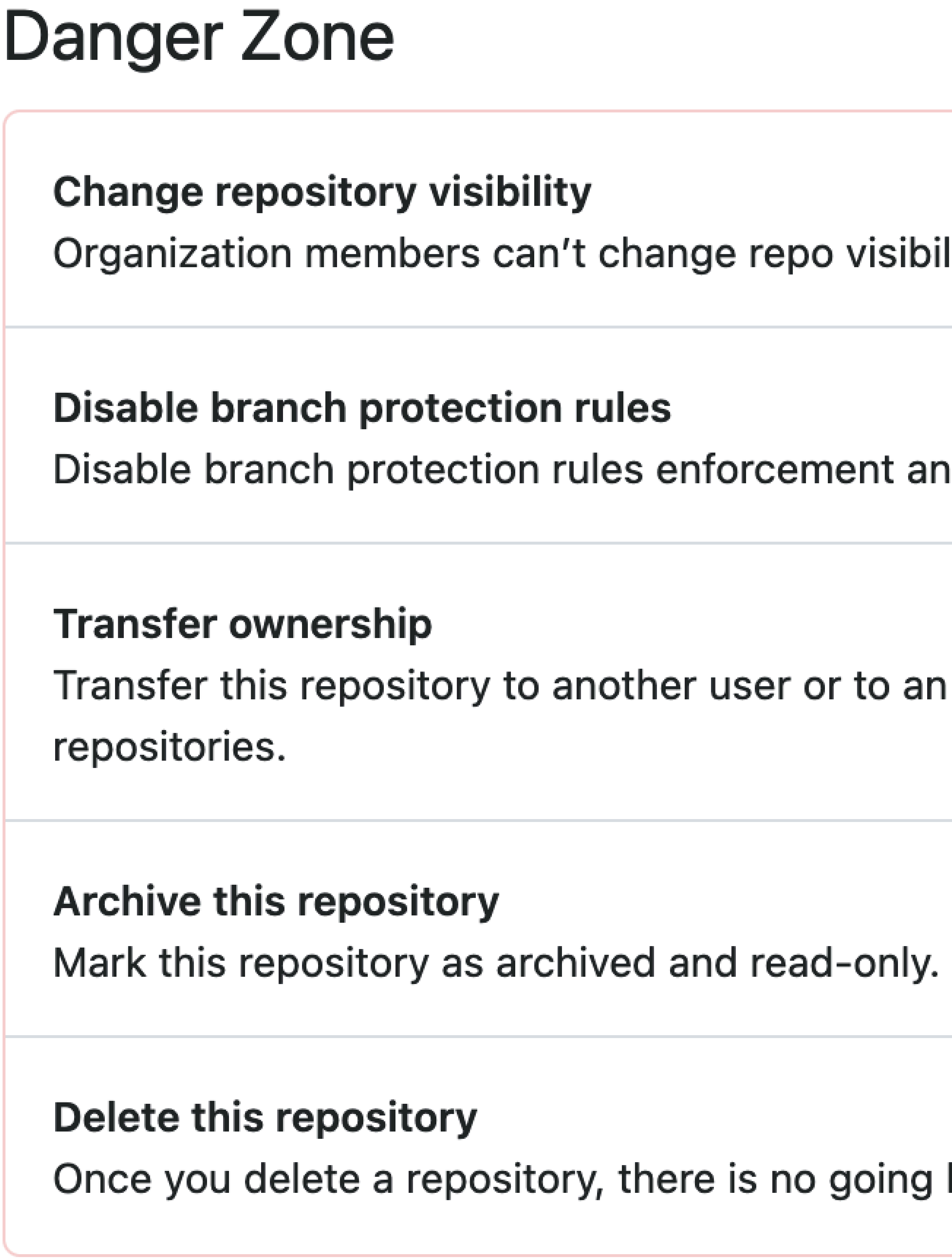
- "Name - WashPost org" — The repo name as it is in the `WashPost` GitHub org
- "Github Teams - WashPost org" — The `WashPost` GitHub org teams assigned to the repo (as admins or otherwise). Example: `[“reporting-tools”]`
- "Git remote origin - WashPost org" — command to update local repo remote origin
- "Notes" — Any lingering miscellaneous differences about the transfer worth recording such as access updates, setting changes, etc.

Finally, set "Status" to "Done" and "Archived" to "yes" or "no"

Step N. (Conditional) Archive

In most cases, we won't be archiving our migrated repos. However, if if a repo appears abandoned or otherwise unused, consider archiving it. Archived repos become read-only and any scheduled processes stop. However, the information within is completely preserved. The repo can be re-activated at any time

To Archive, back to the Danger Zone settings



Danger Zone

Change repository visibility

Organization members can't change repo visibil

Disable branch protection rules

Disable branch protection rules enforcement an

Transfer ownership

Transfer this repository to another user or to an
repositories.

Archive this repository

Mark this repository as archived and read-only.

Delete this repository

Once you delete a repository, there is no going