CircleCI - Automerge Github PR

# Script

#!/bin/bash -e

# CircleCI Automerge GitHub Pull Request Script

# Ensure in PR

if [ -z "$CIRCLE\_PULL\_REQUEST" ]; then

>&2 echo 'Not in pull request, skipping automerge'

exit 1

fi

# Ensure all required environment variables are present

if [ -z "$CIRCLE\_PROJECT\_REPONAME" ] || \

[ -z "$CIRCLE\_PROJECT\_USERNAME" ] || \

[ -z "$CIRCLE\_PULL\_REQUEST" ] || \

[ -z "$CIRCLE\_BRANCH" ] || \

[ -z "$CIRCLE\_SHA1" ] || \

[ -z "$GITHUB\_SECRET\_TOKEN" ]; then

>&2 echo 'Required variable unset, automerging failed'

exit 1

fi

# Extract GitHub PR number

github\_pr\_number="$(echo "$CIRCLE\_PULL\_REQUEST" | sed -n 's/^.\*\/\([0-9]\+\)$/\1/p')"

if [ -z "$github\_pr\_number" ]; then

>&2 echo 'GitHub PR number not found'

exit 1

fi

# Fetch target branch name

curl -L "https://github.com/stedolan/jq/releases/download/jq-1.6/jq-linux64" -o jq

chmod +x jq

url="https://api.github.com/repos/$CIRCLE\_PROJECT\_USERNAME/$CIRCLE\_PROJECT\_REPONAME/pulls/$github\_pr\_number"

target\_branch=$(curl -H "Authorization: $GITHUB\_SECRET\_TOKEN" "$url" | ./jq '.base.ref' | tr -d '"')

if [ -z "$target\_branch" ]; then

>&2 echo 'Failed to fetch GitHub PR target branch'

exit 1

fi

echo : "

CircleCI Automerge Pull Request

Repo: $CIRCLE\_PROJECT\_REPONAME

Pull Request: $github\_pr\_number

Merging: $CIRCLE\_BRANCH >> $target\_branch

"

# Merge PR via GitHub API

curl \

-X PUT \

-H "Accept: application/vnd.github.v3+json" \

-H "Authorization: token $GITHUB\_SECRET\_TOKEN" \

"https://api.github.com/repos/$CIRCLE\_PROJECT\_USERNAME/$CIRCLE\_PROJECT\_REPONAME/pulls/$github\_pr\_number/merge" \

-d '{"commit\_title":"CircleCI automerge '"$CIRCLE\_BRANCH >> $target\_branch"'", "sha": "'"$CIRCLE\_SHA1"'"}'

# Detail

#### Automatically Merge Successful GitHub Pull Requests on CircleCI

##### Add script to project

Add to project in suitable scripts directory:

```

./scripts/circle-ci/circle-ci-github-pr-automerge.sh

```

Make sure it is excutable:

```sh

chmod +x ./scripts/circle-ci/circle-ci-github-pr-automerge.sh

```

##### Configure CircleCI

Then add the run the script after tests have completed in the project ```.circle-ci/config.yml```. For example:

```yml

jobs:

test:

docker:

- image: circleci/ruby:2.7.2-node-browsers

steps:

- checkout

- ruby/install-deps

- ruby/rspec-test

- run:

name: Automerge PR

command: |

set +e

./scripts/circle-ci/circle-ci-github-pr-automerge.sh

if [ $? = 0 ]; then

echo 'Merge successful'

else

echo 'No merge'

fi

```

The 'Automerge PR' run command first sets the shell ```-e``` flag to prevent immediate exiting if the automerge script returns a non-zero exit code (which it will if there is no need for merge or fails). Next, it runs the script and echos the outcome.

##### Set environment in CircleCI

Generate a GitHub Personal Access Token in [github.com/settings/tokens](https://github.com/settings/tokens) with public or private repo access. This allows the script to fetch information about the pull request from GitHub, and process the merge.

Ensure the required secret environment variables are set in the build by adding them using the [CircleCI project settings]( <https://circleci.com/docs/2.0/env-vars/#setting-an-environment-variable-in-a-project>):

```sh

GITHUB\_SECRET\_TOKEN=<github-personal-access-token>

```

##### Create GitHub action

CircleCI does not automatically trigger a pipeline (build) on the creation of a github pull request, meaning if the pipeline for the commit has already been triggered in CircleCI before the pull request is created (which is common), the required pull request environment variables will not be present, and the automerge script will fail.

To get around this, create a basic [GitHub action]( https://docs.github.com/en/free-pro-team@latest/actions) for the project which triggers a new CircleCI pipeline.

Add a new GitHub action workflow in GitHub workflows ```.github/workflows/main.yml```:

```yaml

name: CI

on:

pull\_request:

types: [opened,reopened]

jobs:

build:

runs-on: ubuntu-latest

steps:

- name: 'Trigger CircleCI PR Build'

env:

CIRCLECI\_API\_TOKEN: ${{ secrets.CIRCLECI\_API\_TOKEN }}

run: |

curl -X POST \

-H "Circle-Token: $CIRCLECI\_API\_TOKEN" \

-H 'Content-Type: application/json' \

-H 'Accept: application/json' \

-d "{\"branch\":\"$GITHUB\_HEAD\_REF\"}" \

"https://circleci.com/api/v2/project/gh/$GITHUB\_REPOSITORY/pipeline"

```

Create a [CircleCI API token](https://circleci.com/docs/2.0/managing-api-tokens/) to allow GitHub to trigger the pipeline. Then save the token as an [encrypted secret]( <https://docs.github.com/en/free-pro-team@latest/actions/reference/encrypted-secrets>) called ```CIRCLECI\_API\_TOKEN``` in the GitHub project settings.