

git, github

Connect to Frontend

DevOps

DevOps = Development + Operations

- 개발과 운영의 합성어
- Dev: Plan - Code - Build - Test
- Ops: Release - Deploy - Operate - Monitor
- Cross Functional Team: 개발과 운영을 한 팀으로 묶어 프로세스의 자동, 단일화
- CI/CD Tool 이용하여 Build, Test, Deploy 자동화
- Pros
 - 커뮤니케이션 리소스 개선
 - 개발, 배포 속도가 빨라짐
 - 프로세스 간소화
 - 짧은 릴리즈 주기

CI/CD



- TravisCI, CircleCI, Jenkins

CI(Continuous Integration)

- 자동화된 프로세스
- 코드 변경사항의 정기적 빌드, 테스트 병합 자동화
- Pros
 - 빠른 디버깅
 - 코드 품질 개선
 - 검증 및 릴리즈 시간 단축

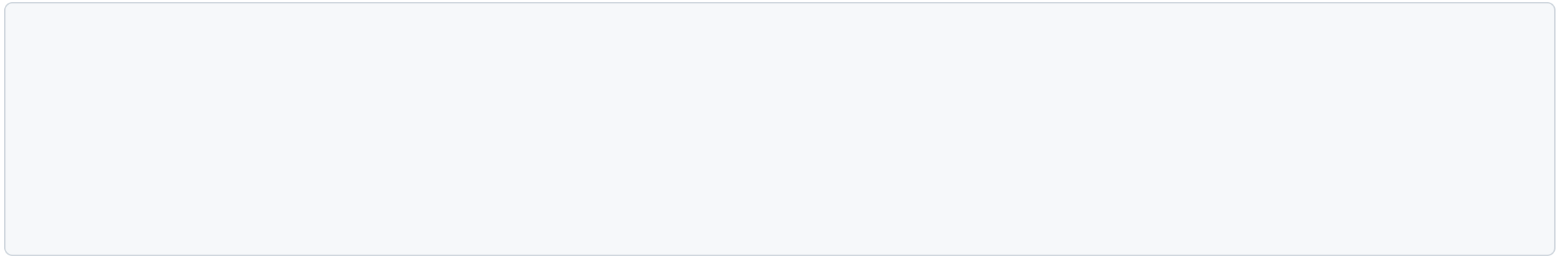
CD(Continuous Delivery(or Deployment))

- Continuous Delivery: 공유 저장소로 자동 Release(Test -> Staging)
- Continuous Deployment: Production Level까지 자동 Deploy(Test -> Staging -> Production)
- MSA(MicroService Architecture) + Agile 일 경우, 사용자에게 최대한 빠른 시간안에 Production 제공 필요

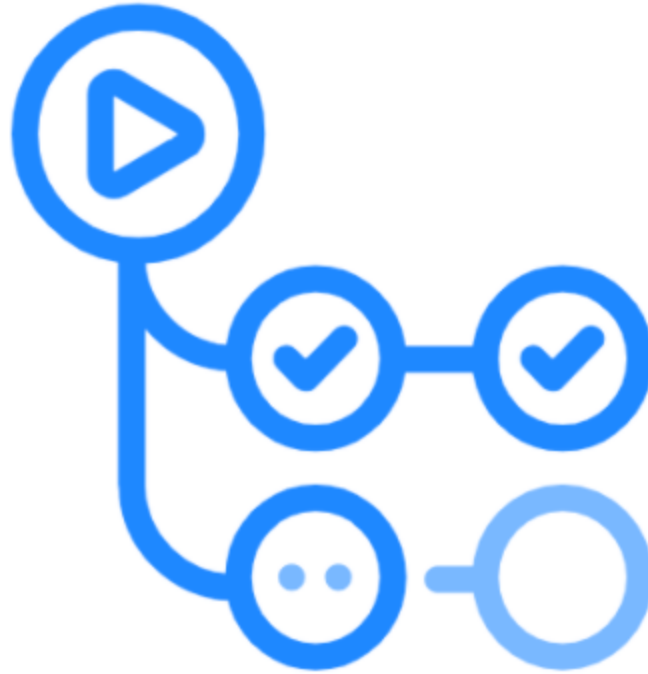
Issue & PR Templates

- configuration file을 만들어 issue와 PR 내용의 template 지정 가능
- `.github/ISSUE_TEMPLATE/*.md` 에 template 생성 가능
- `Settings` - Features_Set up templates 에서 손쉽게 생성 가능
- `.github/PULL_REQUEST_TEMPLATE/pull_request_template.md`

pull_request_template.md



CI/CD tools for github



GitHub Actions

github actions

- github에서 공식 제공하는 CI/CD Tools
- 개발 workflow 자동

github actions - core

- Workflow
- Event
- Job
- Step
- Action
- Runner

Workflow

- Job들로 구성. Event에 의해 트리거되는 자동화된 프로세스
- 최상위 개념
- YAML으로 작성되며, `.github/workflows`에 저장

Event

- Workflow를 실행하는 규칙
- Push, pull request, Cron, webhook으로 연결된 외부 이벤트에 의해 실행

Job

- Step들로 구성
- 가상환경의 인스턴스에서 실행
- 다른 Job에 의존관계를 가질 수 있고, 독립 병렬 실행도 가능

Step

- Task들의 집합으로 커맨드를 실행하거나 action을 실행

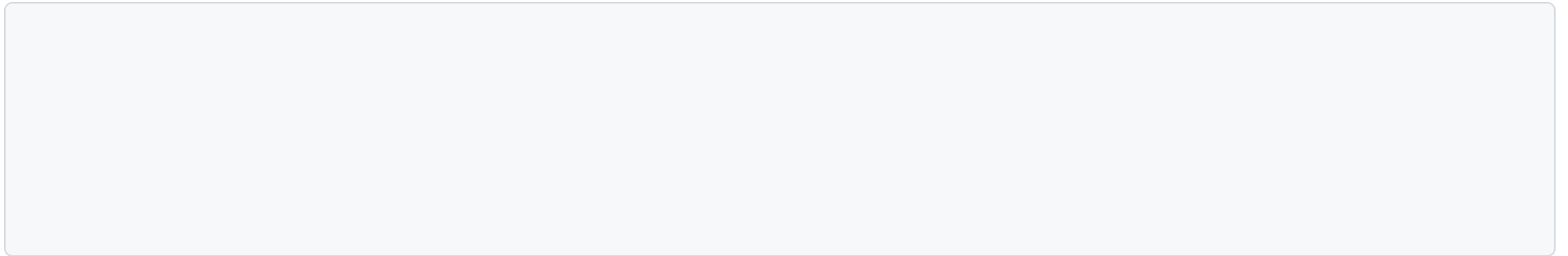
Action

- 가장 작은 단위
- Step을 연결해 Job을 구성
- 재사용 가능
- marketplace나 개인이 만든 action을 사용할 수 있음

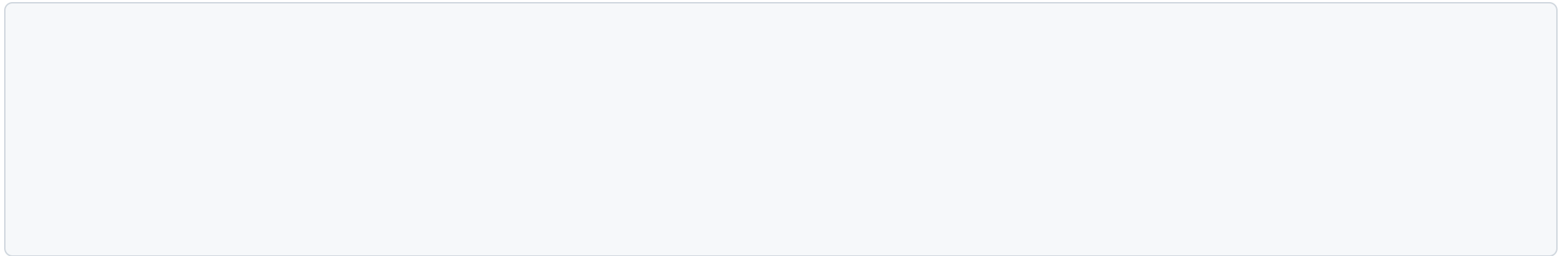
Runner

- workflow가 실행될 인스턴스
- github-hosted runner와 self-hosted runner로 나뉨

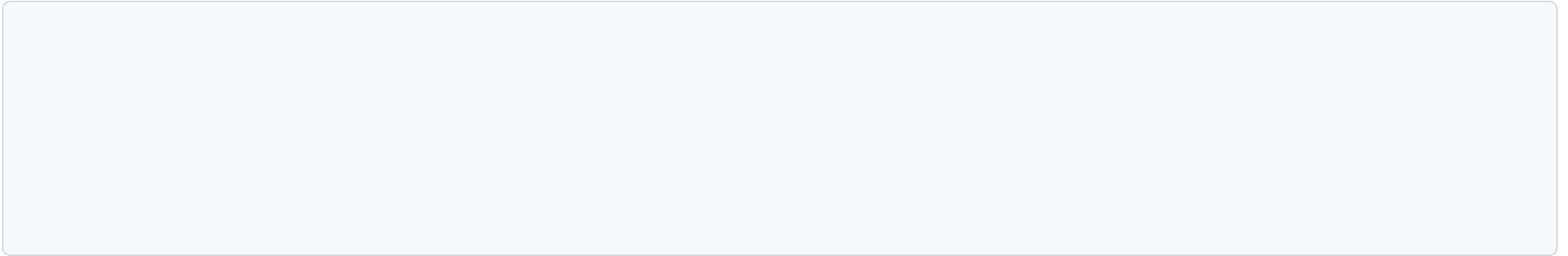
echo hello



echo something



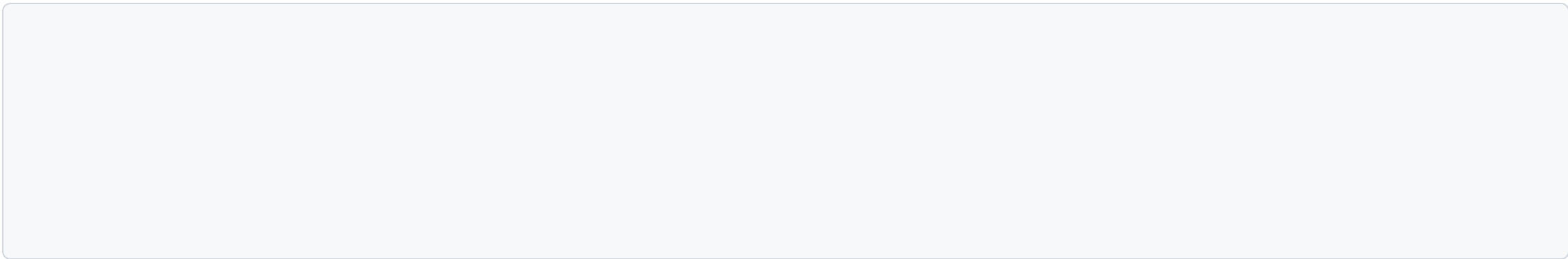
Adding a workflow status badge



Adding Another types of status badge

shields.io

규칙 정하기

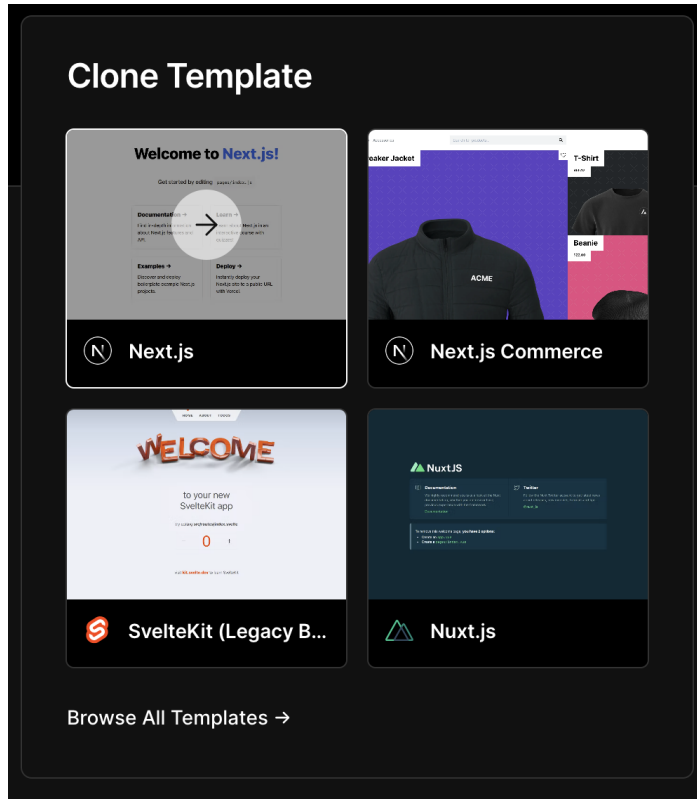


Don't reinvent the wheel

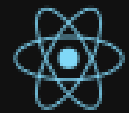
- CI/CD 파이프라인은 그 절차와 각 단계에서의 할 일이 정해져 있는 편
- 이미 만들어진 것에 커스터마이징 하는 것이 CI/CD 스크립트 짜는 시간을 개발에 투입 가능

github actions with vercel

Clone template



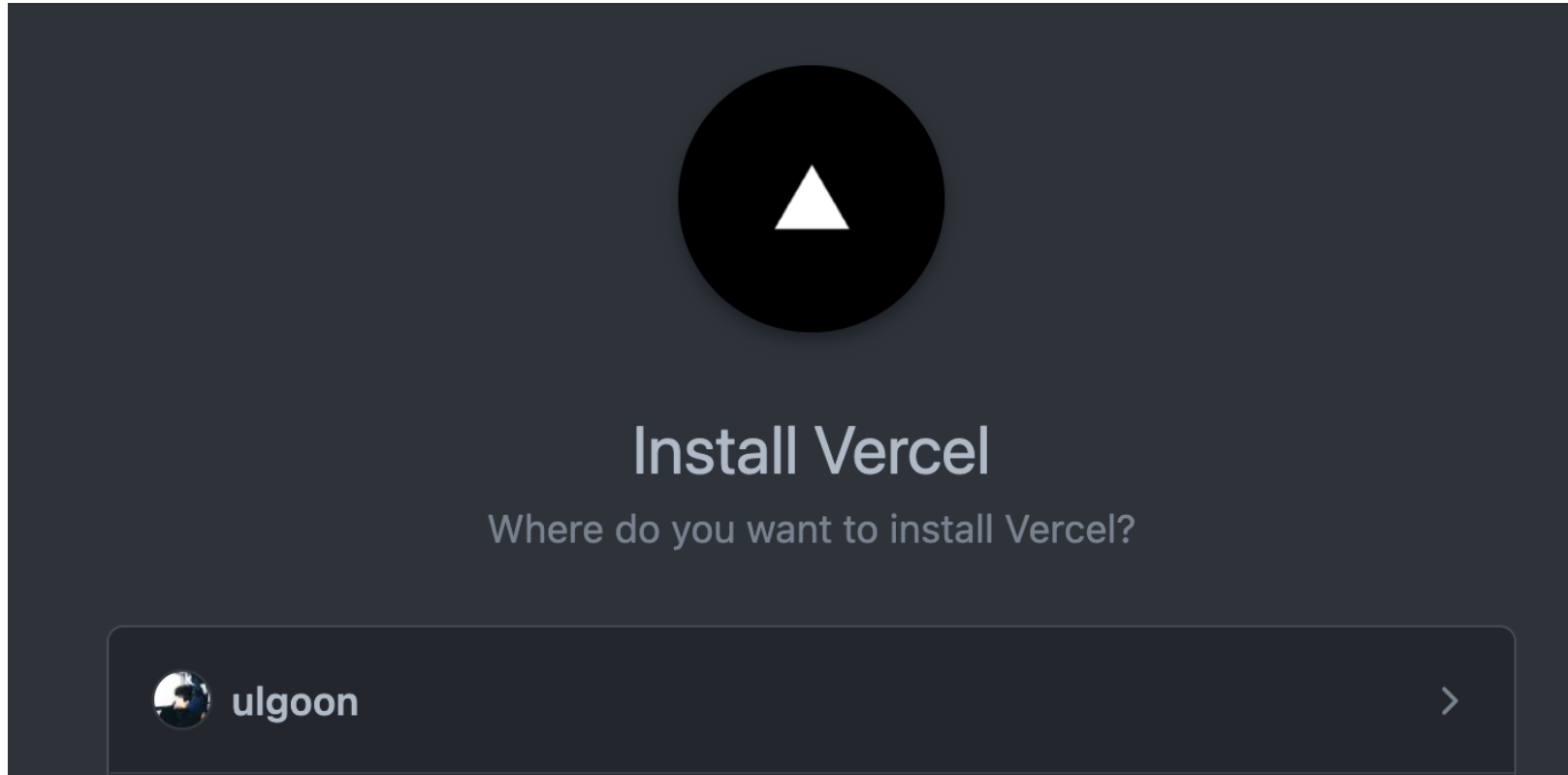
Let's use CRA



Create React App

A client-side React app created with create-react-app.

Install your vercel



Create git repository

You're almost done.

Please follow the steps to configure your Project and deploy it.



Edit src/App.js and save to reload.
[Learn React](#)


Create React App

A client-side React app created
with create-react-app.

Create Git Repository

To ensure you can easily update your project after deploying it, a Git repository must be created. Every push to that Git repository will be deployed automatically.

Git Scope

 ulgoon

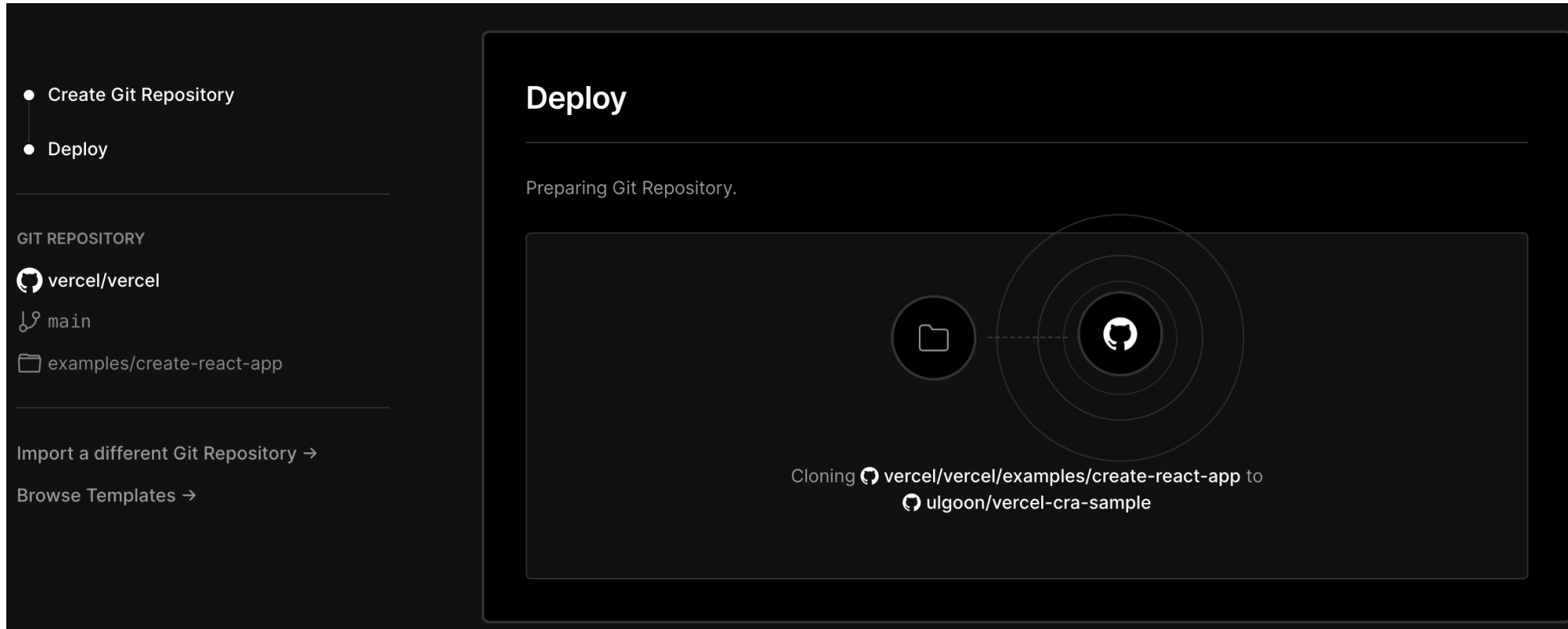
Repository Name

vercel-cra-sample

☒ Create private Git Repository

Create

Deploy done




Real done!

Congratulations!

You just deployed a new Project to Vercel.


[Continue to Dashboard](#)





Edit src/App.js and save to reload.

[Learn React](#)



Next Steps

 **Instant Previews**

Push a new branch to preview changes instantly

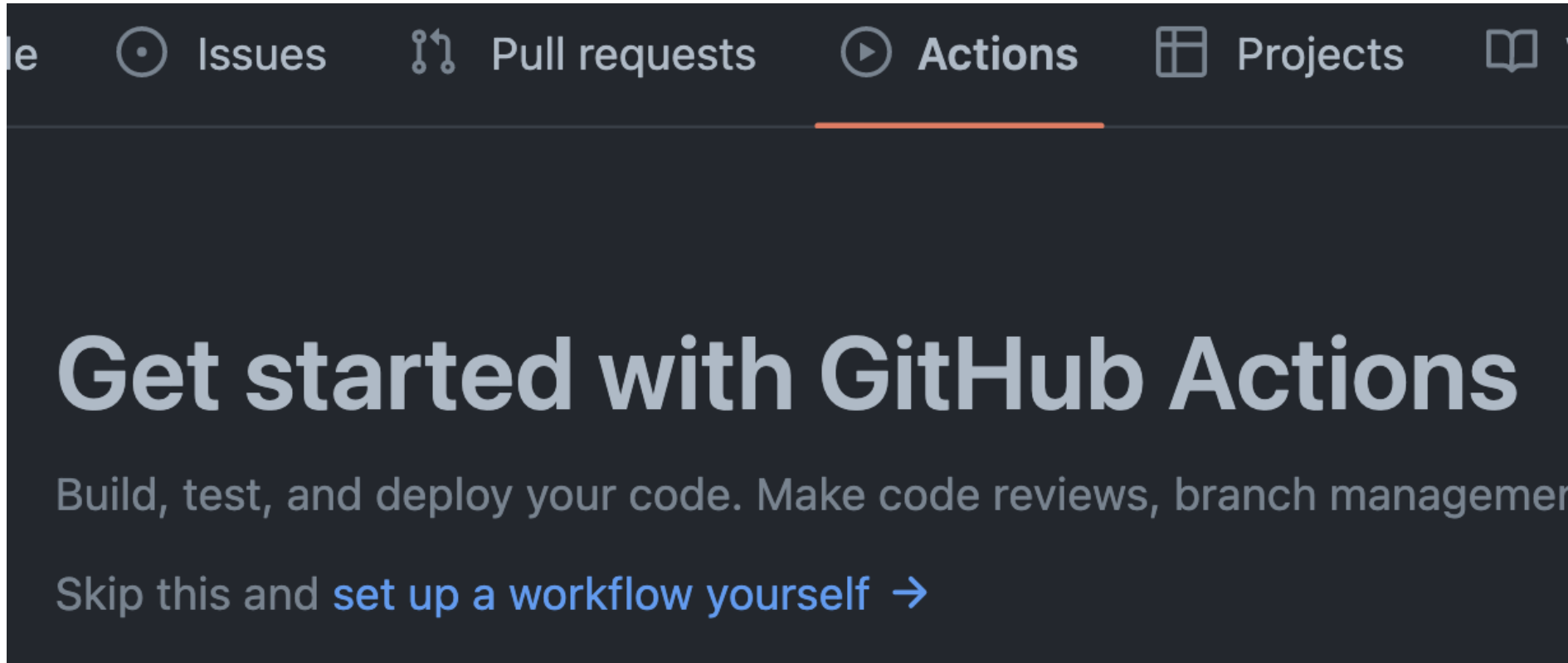
 **Add Domain** 

Add a custom domain to your project

 **Enable Analytics** 

Track how users experience your site over time

Go to actions tab



Settings - Secrets - Actions

Actions secrets / New secret

Name *




VERCEL_PROJECT_ID

Secret *

sample

Add secret

Requirements

Repository secrets	
	VERCEL_ORG_ID
	VERCEL_PROJECT_ID
	VERCEL_TOKEN

Let's Create token

<https://vercel.com/account/tokens>

Create Token

Enter a unique name for your token to differentiate it from other tokens. Then select the scope for the token. Learn more about [access tokens](#).

TOKEN NAME

SCOPE

Full Account

EXPIRATION

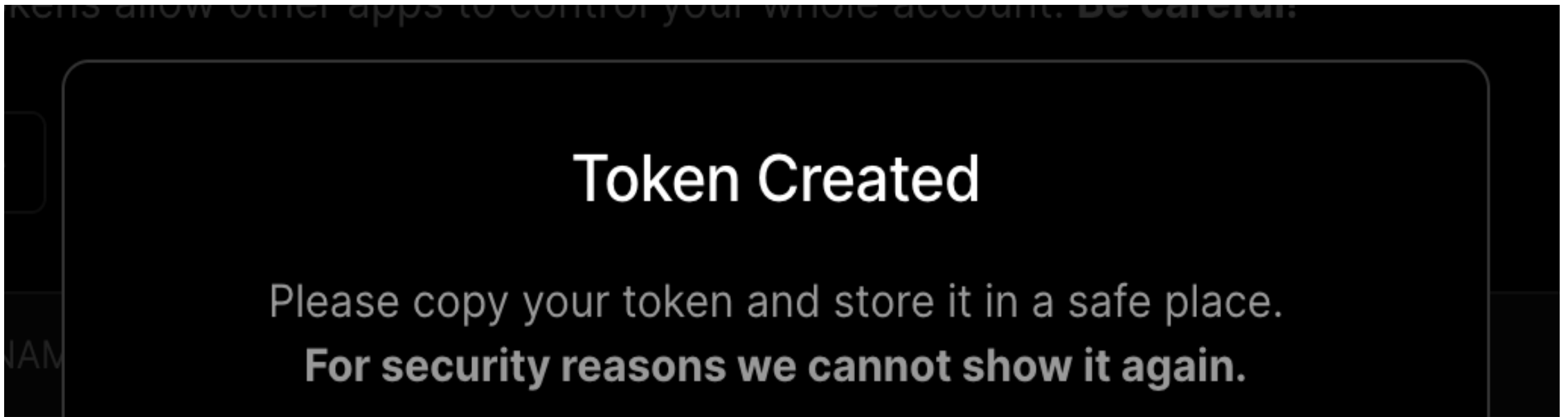
1 Year

This token will expire on 12/20/23

CANCEL



CREATE TOKEN

Token created



How to get VERCEL_ORG_ID, VERCEL_PROJECT_ID

- using vercel(`$ npm i -g vercel`)
- `$ vercel link` on your repo
- <https://vercel.com/docs/cli>

```
vercel-cra-sample on  main via  v14.17.6
→ vercel link
Vercel CLI 28.9.0
> > No existing credentials found. Please log in:
> Log in to Vercel (Use arrow keys)
> Continue with GitHub
Continue with GitLab
Continue with Bitbucket
Continue with Email
Continue with SAML Single Sign-On
Cancel
```

CLI Login on your browser




CLI Login Success

Vercel CLI was successfully authenticated as:

ulgoon89@gmail.com

You can now close this tab and return to Vercel CLI.

link done

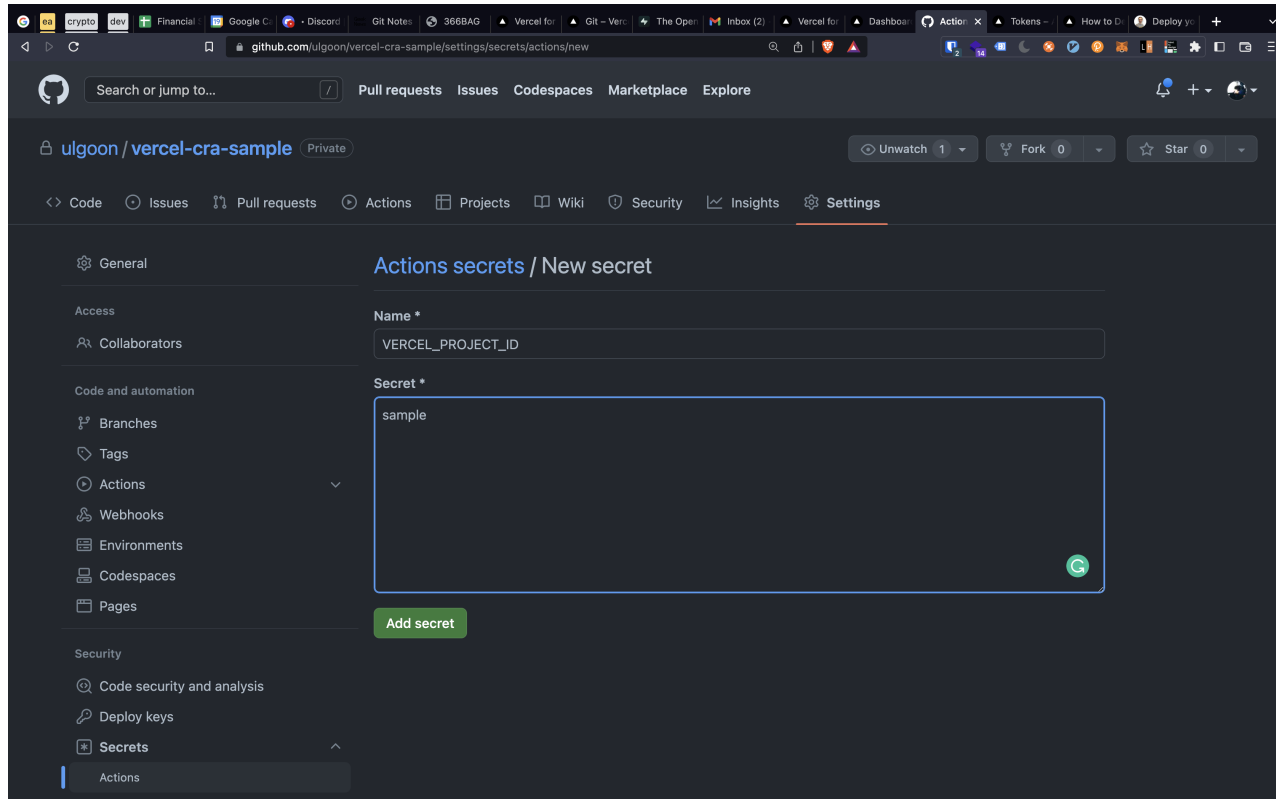
```
vercel-cra-sample on  main via  v14.17.6
→ vercel link
Vercel CLI 28.9.0
> > No existing credentials found. Please log in:
> Log in to Vercel github
> Success! GitHub authentication complete for ulgoon89@gmail.com
? Set up "~/Documents/dev/vercel-cra-sample"? [Y/n] y
? Which scope should contain your project? ulgoon
? Found project "ulgoon/vercel-cra-sample". Link to it? [Y/n] y
 Linked to ulgoon/vercel-cra-sample (created .vercel)
```

Get your org, prjt id from `.vercel/project.json`

```
.. (up a dir)
</dev/vercel-cra-sample/
▶ .git/
▶ .github/
▼ .vercel/
   project.json
```

```
{"projectId": "prj_ws[REDACTED]v", "orgId": "n3[REDACTED]"}
```

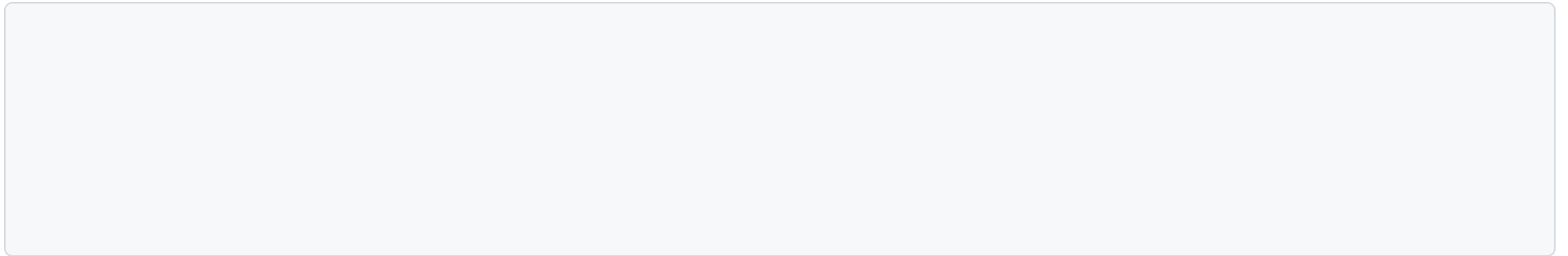
Paste it!



vercel preview and deploy script

<https://vercel.com/guides/how-can-i-use-github-actions-with-vercel>

Preview



Deploy

