

Experiment 9

Source Code:

A screenshot of a GitHub code viewer interface. At the top, there are tabs for 'Code' and 'Blame', followed by the text '29 lines (21 loc) · 730 Bytes'. The main area displays a YAML workflow file. The workflow is named 'Vercel Preview Deployment'. It includes environment variables for 'VERCEL_ORG_ID' and 'VERCEL_PROJECT_ID' using secrets. The trigger is set to 'push' on the 'main' branch. A single job named 'Deploy-Preview' is defined, running on 'ubuntu-latest'. The job contains three steps: 1) 'actions/checkout@v4', 2) 'Install Vercel CLI' using 'npm install --global vercel@latest', and 3) 'Pull Vercel Environment Information' using 'vercel pull --yes --environment=preview --token=\${{ secrets.VERCEL_TOKEN }}'.

```
1  name: Vercel Preview Deployment
2
3  env:
4    VERCEL_ORG_ID: ${ secrets.VERCEL_ORG_ID }
5    VERCEL_PROJECT_ID: ${ secrets.VERCEL_PROJECT_ID }
6
7  on:
8    push:
9      branches-ignore:
10       - main
11
12  jobs:
13    Deploy-Preview:
14      runs-on: ubuntu-latest
15
16      steps:
17        - uses: actions/checkout@v4
18
19        - name: Install Vercel CLI
20          run: npm install --global vercel@latest
21
22        - name: Pull Vercel Environment Information
23          run: vercel pull --yes --environment=preview --token=${{ secrets.VERCEL_TOKEN }}
```

Fig 9.1

CodeBlame28 lines (21 loc) · 745 Bytes

```
1  name: Vercel Production Deployment
2
3  env:
4    VERCEL_ORG_ID: ${ secrets.VERCEL_ORG_ID }
5    VERCEL_PROJECT_ID: ${ secrets.VERCEL_PROJECT_ID }
6
7  on:
8    push:
9      branches:
10       - main
11
12  jobs:
13    Deploy-Production:
14      runs-on: ubuntu-latest
15      steps:
16        - uses: actions/checkout@v4
17
18        - name: Install Vercel CLI
19          run: npm install --global vercel@latest
20
21        - name: Pull Vercel Environment Information
22          run: vercel pull --yes --environment=production --token=${ secrets.VERCEL_TOKEN }
23
24        - name: Build Project Artifacts
```

Fig 9.2

Output:

JayKerkar

HomeTech StackAboutProjectsFAQ

Precision in Web |

Specialize in building both web and mobile applications, covering the full spectrum of development. My true passion lies in coding and creating intuitive digital experiences for users. I'm eager to connect and discover fresh opportunities in the tech world.

View on GitHub

Follow on LinkedIn

Behind the Scenes: Unveiling the Technologies That Power Our Website

📱

💻

🚀

🔗

Created

bytwis04...

11/1/24

Ready

Duration

19s 346d ago

Environment

Production

Current

Domains

jaykerkar.vercel.app +2

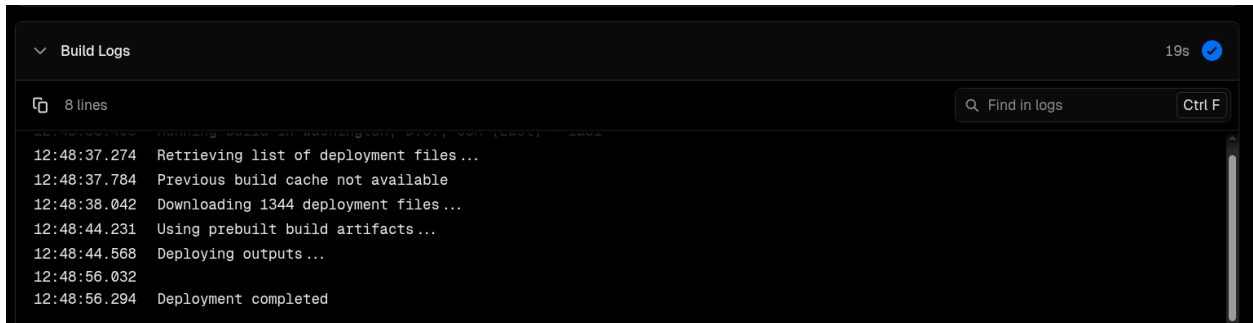
client-side-is7suan2k-bytwis-projects.vercel.app

Source

<> View code

>_ vercel deploy

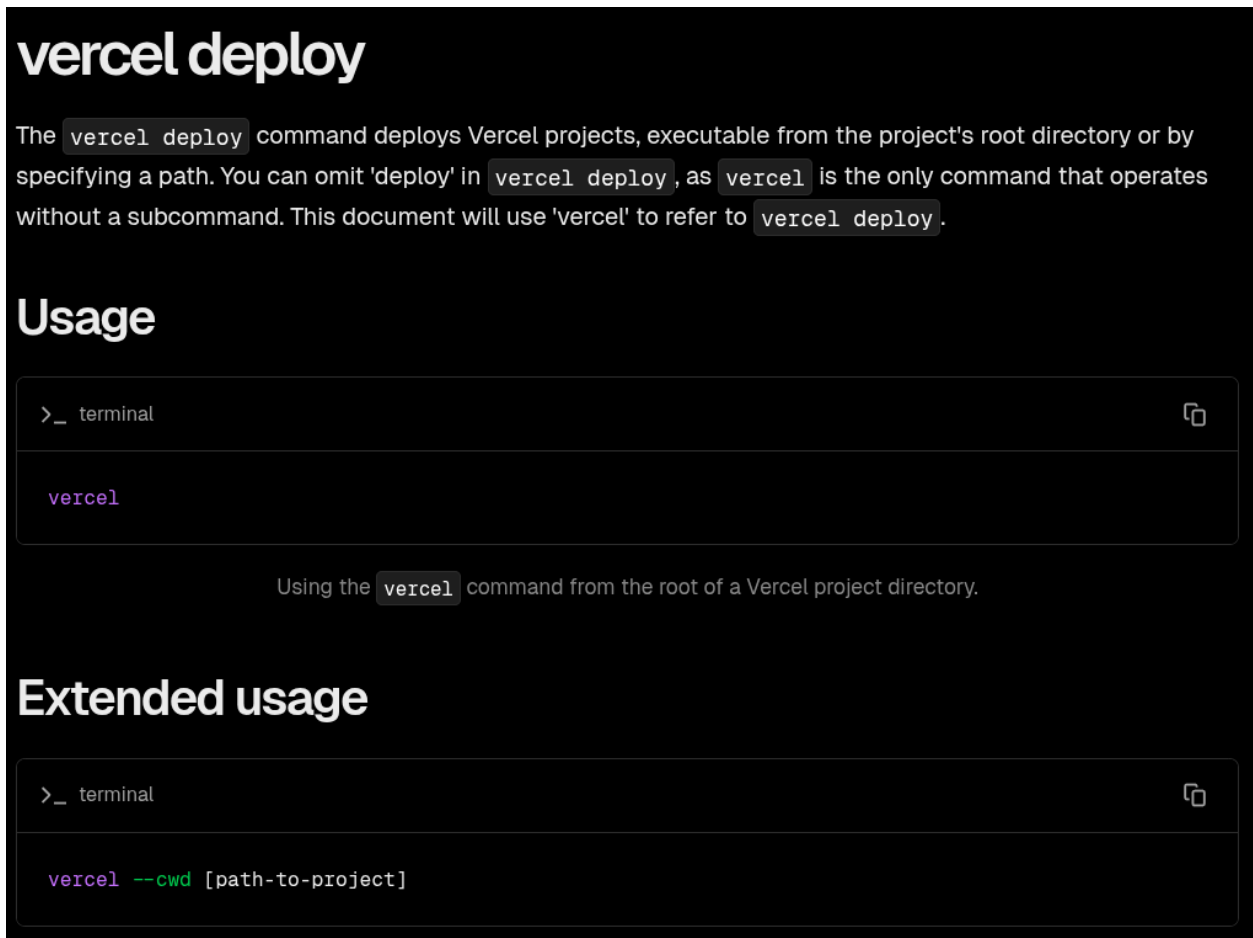
Fig 9.3



The screenshot shows a terminal window titled "Build Logs" with a search bar and a "Ctrl F" button. The logs display the following sequence of events:

```
12:48:37.274 Retrieving list of deployment files...
12:48:37.784 Previous build cache not available
12:48:38.042 Downloading 1344 deployment files...
12:48:44.231 Using prebuilt build artifacts...
12:48:44.568 Deploying outputs...
12:48:56.032
12:48:56.294 Deployment completed
```

Fig 9.4



The screenshot displays the Vercel CLI help page. It features the "vercel deploy" command description, a "Usage" section showing the command "vercel", and an "Extended usage" section showing the command "vercel --cwd [path-to-project]".

vercel deploy

The `vercel deploy` command deploys Vercel projects, executable from the project's root directory or by specifying a path. You can omit 'deploy' in `vercel deploy`, as `vercel` is the only command that operates without a subcommand. This document will use 'vercel' to refer to `vercel deploy`.

Usage

```
>_ terminal
vercel
```

Using the `vercel` command from the root of a Vercel project directory.

Extended usage

```
>_ terminal
vercel --cwd [path-to-project]
```

Fig 9.5