

DevOps bez bólu

historia o skutecznym wprowadzaniu CI/CD do
swoich projektów



Krystian Sobolewski

Software Engineer
Devops Engineer



[/krystian-sobolewski](#)



Znacie to?



Znacie to?

“Zapomniałem odpalić...”



Znacie to?

“Zapomniałem odpalić buildu”



Znacie to?

“Zapomniatem odpalić buildu”

“Zapomniatem odpalić testy”



Znacie to?

“Zapomniałem odpalić buildu”

**“Cache się nie przetadował po
wrzuceniu zmian”**

“Zapomniałem odpalić testy”



Znacie to?

“Zapomniałem odpalić buildu”

“Zapomniałem odpalić testy”

“Zrobię fix bezpośrednio na produkcji”

“Cache się nie przetadował po wrzuceniu zmian”



Znacie to?

“Zapomniałem odpalić buildu”

“Zapomniałem odpalić testy”

“Arek jest na urlopie, a tylko on wie jak się to uruchamia”

“Cache się nie przetadował po wrzuceniu zmian”

“Zrobię fix bezpośrednio na produkcji”



Znacie to?

“Zapomniałem odpalić buildu”

“Zapomniałem odpalić testy”

“Źle się skopiowało”

“Cache się nie przetadował po wrzuceniu zmian”

“Arek jest na urlopie, a tylko on wie jak się to uruchamia”

“Zrobię fix bezpośrednio na produkcji”





“Zapomniałem odprężyć kciuki”

“Zapomniałem odpalić testy”

“Cache się nie przetłumaczył po wrzuceniu zmian”

“Zrobię fix bezpośrednio na produkcji”



DevOps bez bólu

historia o skutecznym wprowadzaniu CI/CD do
swoich projektów



DevOps



Dev | Ops



Development | Ops



Developer



Developer



Software Developer



Software Development

Tworzenie oprogramowania



Software development

- Pisanie kodu
- Czytanie kodu



Software development

- Pisanie kodu
- Czytanie kodu
- Testowanie



Software development

- Pisanie kodu
- Czytanie kodu
- Testowanie
- Debugowanie



Software development

- Pisanie kodu
- Czytanie kodu
- Testowanie
- Debugowanie
- Rozwiązywanie problemów



Dev | Operations



Procesy



Procesy



Dev Operations:

- Budowanie



Dev Operations:

- Budowanie
- Wdrażanie



Dev Operations:

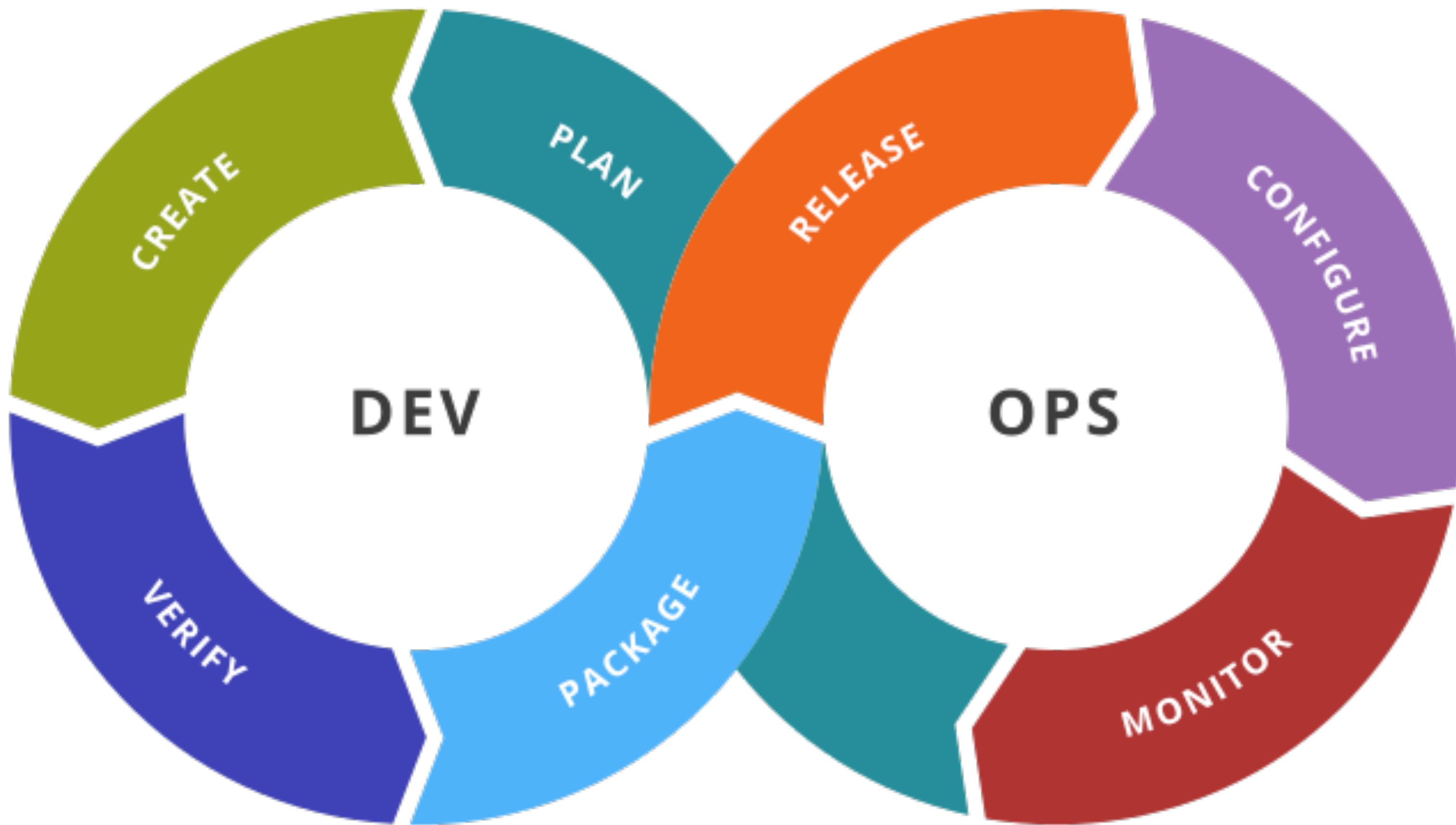
- Budowanie
- Wdrażanie
- Testowanie



Dev Operations:

- Budowanie
- Wdrażanie
- Testowanie
- Monitoring





Story time!

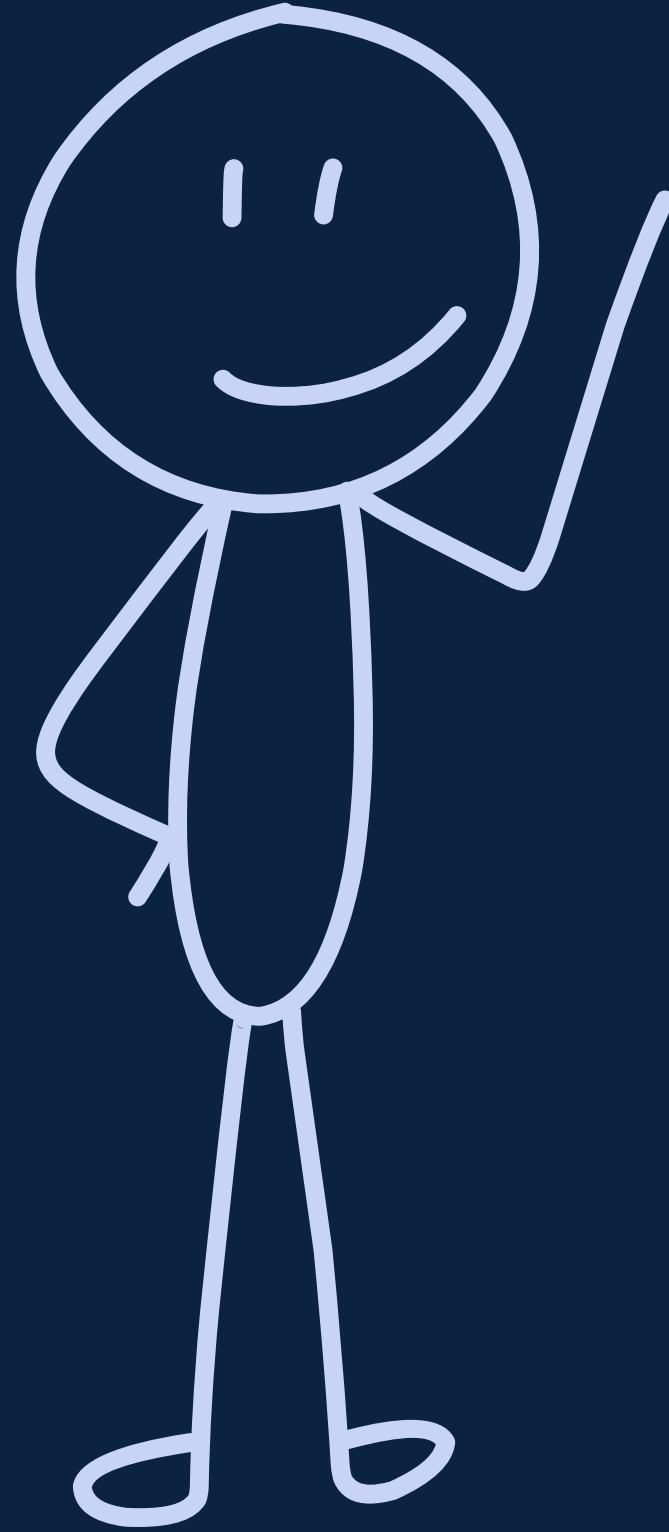


Rozdział 0

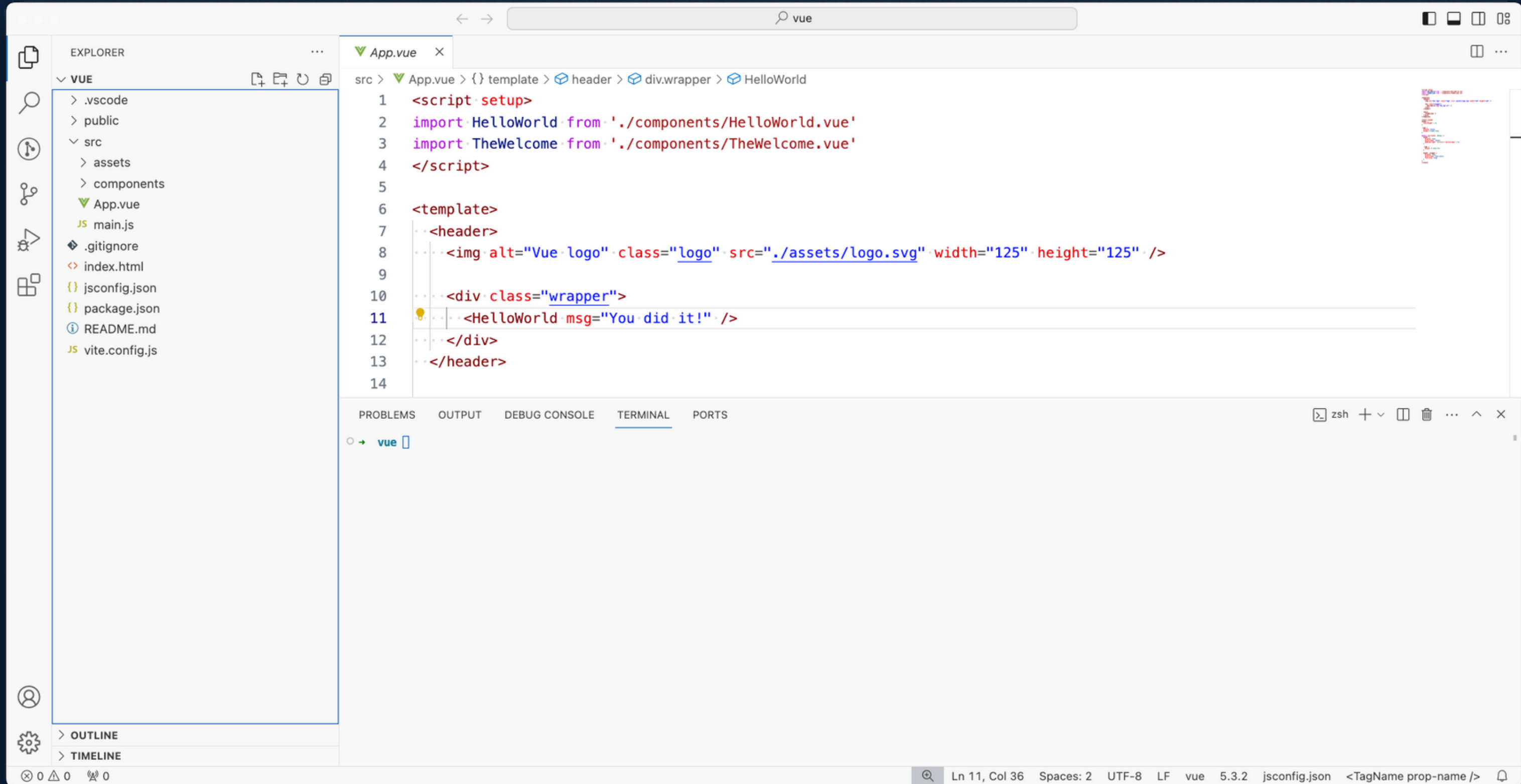
Intro



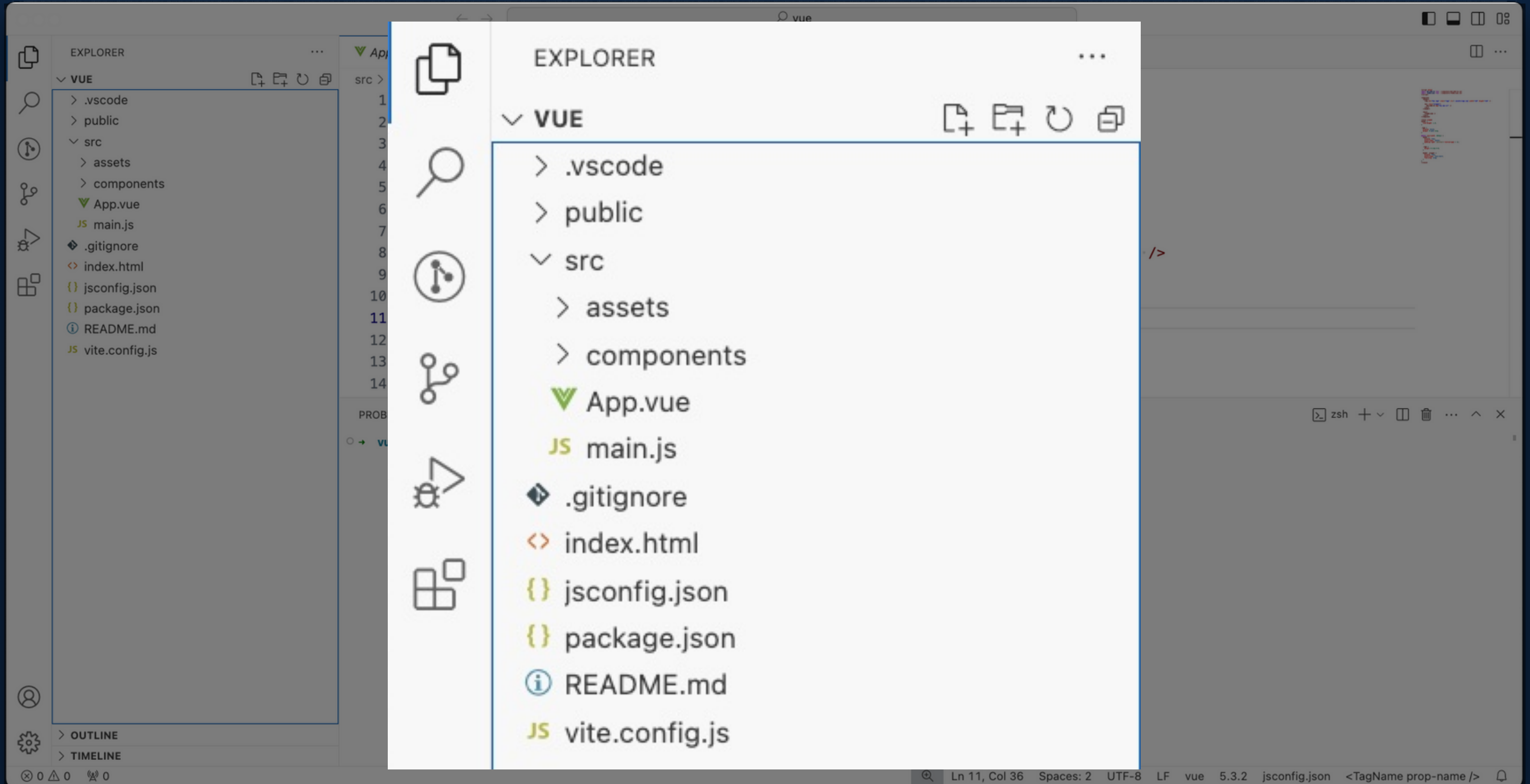
John



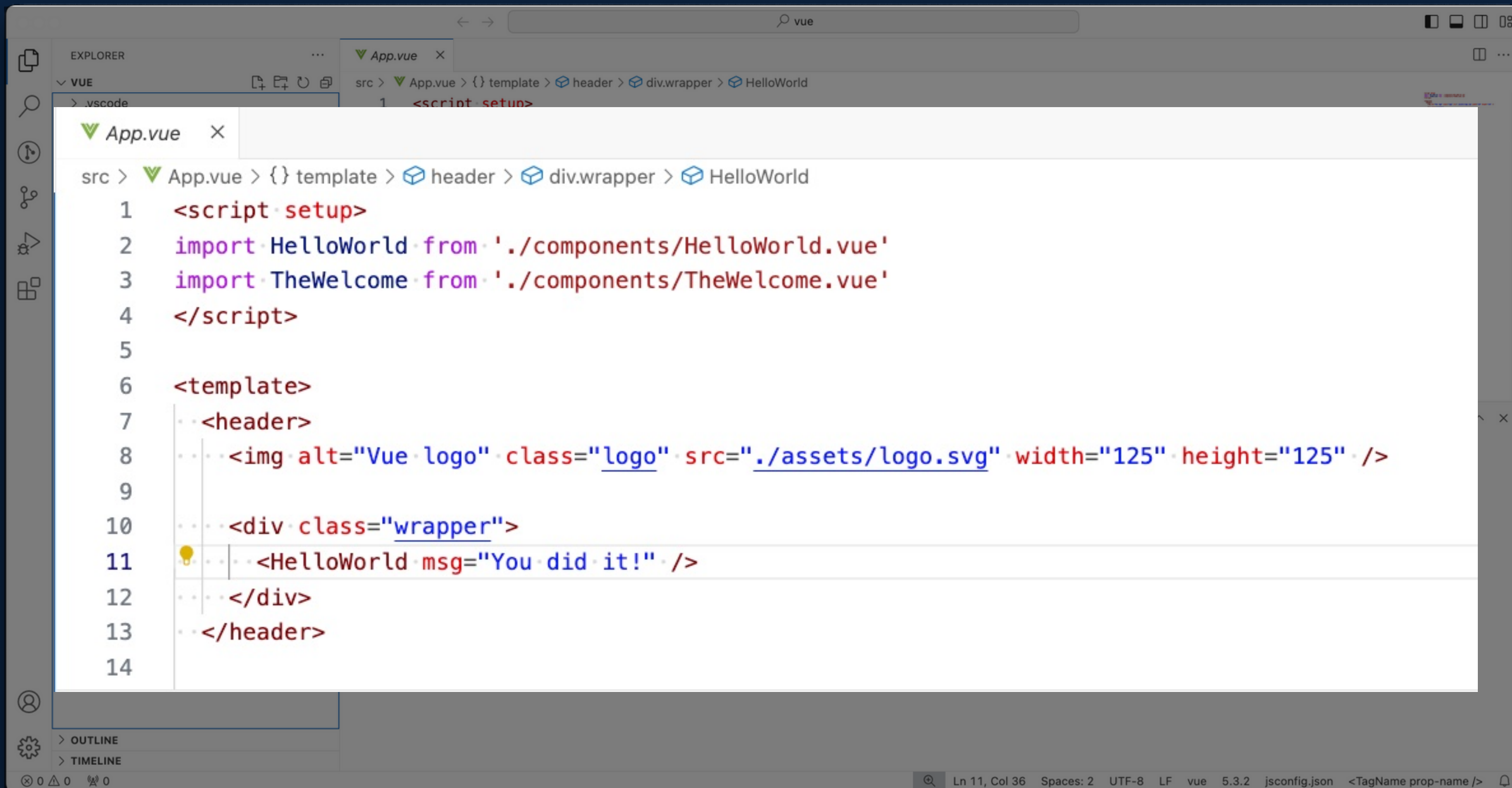
Aplikacja



Aplikacja



Aplikacja



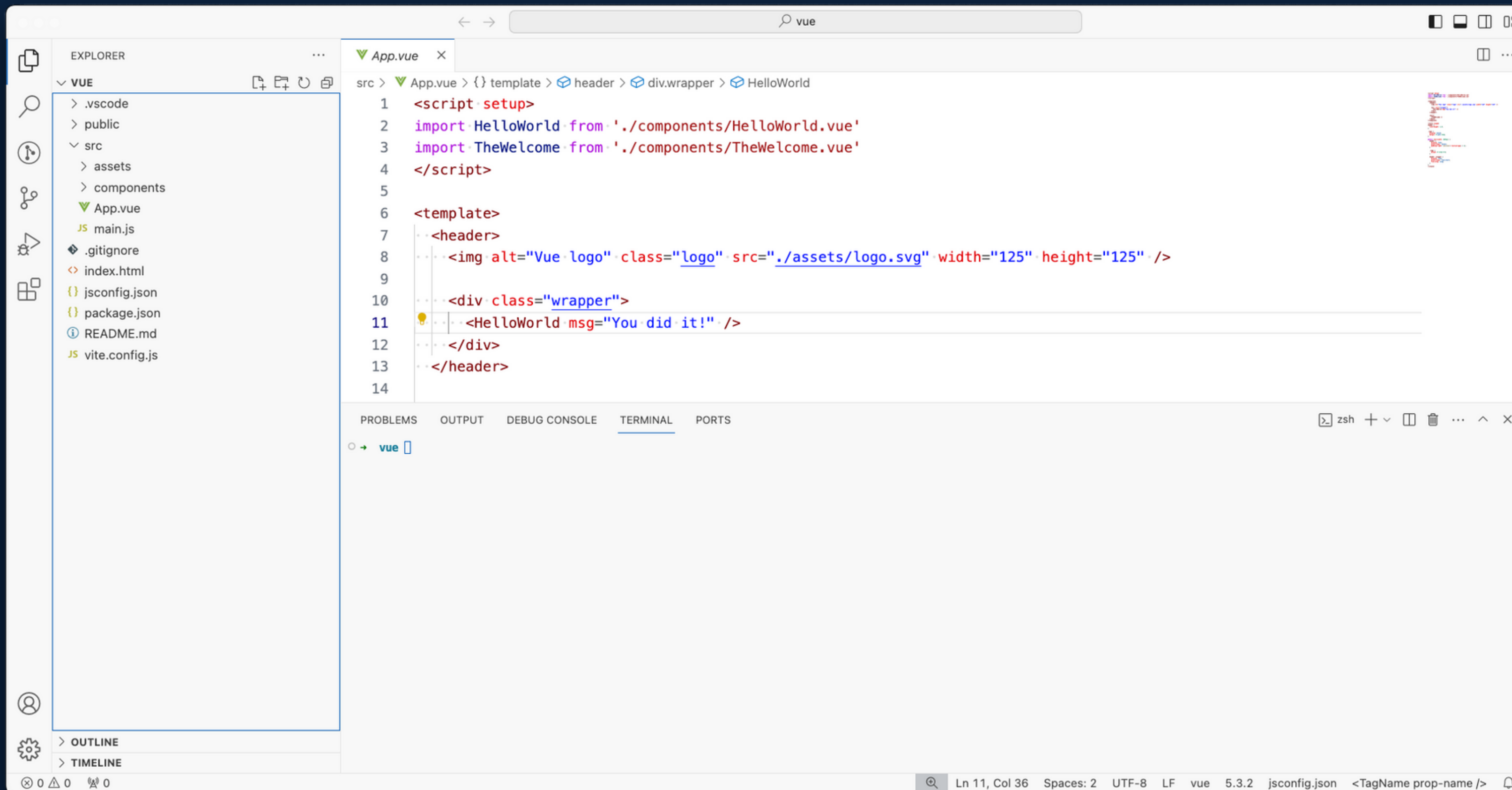
The screenshot shows a code editor window with the following content:

```
src > App.vue > {} template > header > div.wrapper > HelloWorld
1  <script setup>
2  import HelloWorld from './components/HelloWorld.vue'
3  import TheWelcome from './components/TheWelcome.vue'
4  </script>
5
6  <template>
7  .. <header>
8  ... 
9
10 ... <div class="wrapper">
11 .. <HelloWorld msg="You did it!" />
12 ... </div>
13 .. </header>
14
```

The status bar at the bottom indicates: Ln 11, Col 36 Spaces: 2 UTF-8 LF vue 5.3.2 jsconfig.json <TagName prop-name />



Uruchamianie - instalowanie zależności



The screenshot shows the Visual Studio Code editor interface. On the left, the Explorer sidebar displays the project structure for a Vue.js application:

- EXPLORER
 - VUE
 - .vscode
 - public
 - src
 - assets
 - components
 - App.vue
 - main.js
 - .gitignore
 - index.html
 - jsconfig.json
 - package.json
 - README.md
 - vite.config.js

The main editor area shows the code for `App.vue`. The breadcrumb navigation indicates the current file path: `src > App.vue > {} template > header > div.wrapper > HelloWorld`. The code is as follows:

```
1 <script setup>
2 import HelloWorld from './components/HelloWorld.vue'
3 import TheWelcome from './components/TheWelcome.vue'
4 </script>
5
6 <template>
7   <header>
8     
9
10    <div class="wrapper">
11      <HelloWorld msg="You did it!" />
12    </div>
13  </header>
14
```

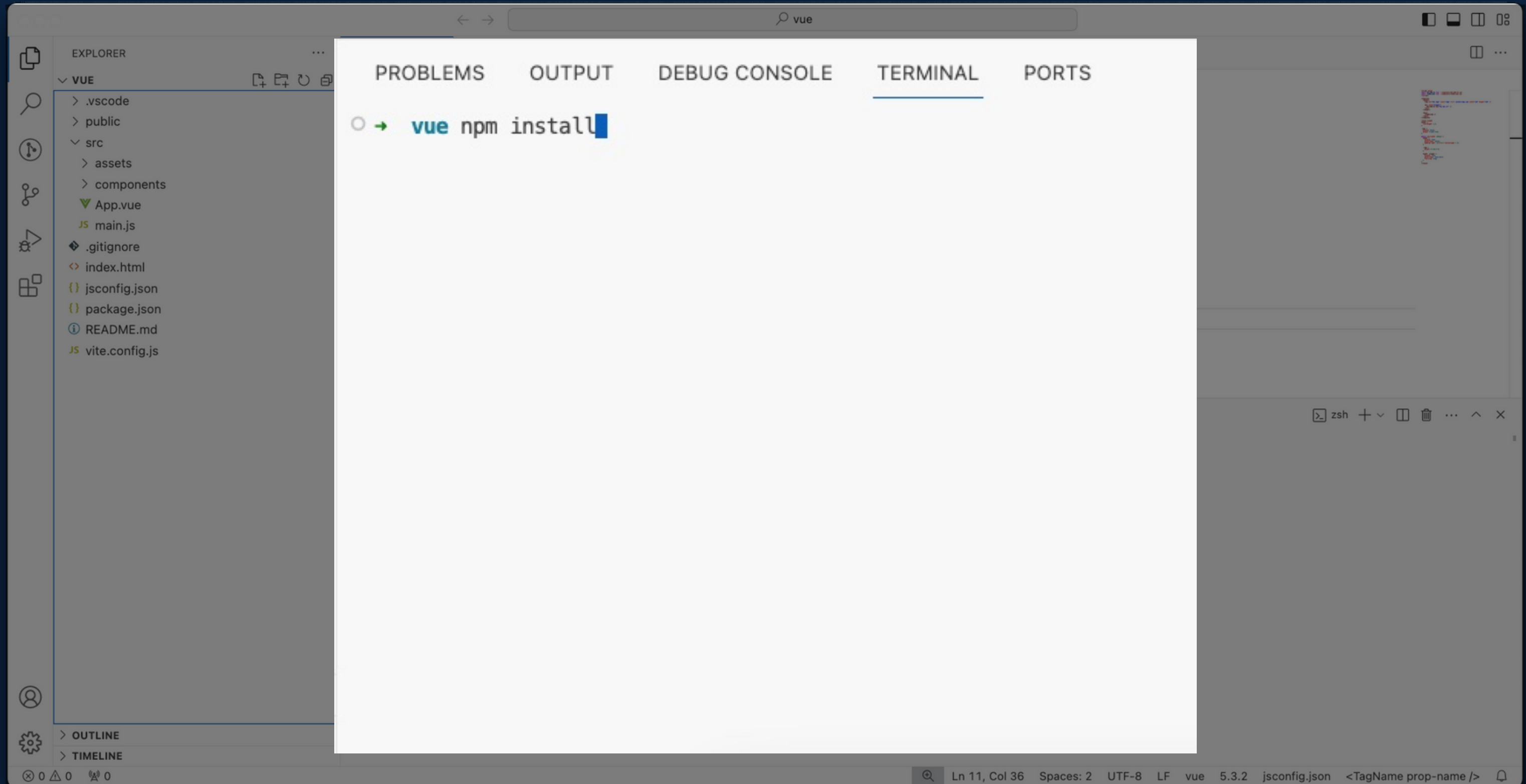
At the bottom of the editor, the TERMINAL panel is visible, showing a prompt for the `vue` command:

```
vue
```

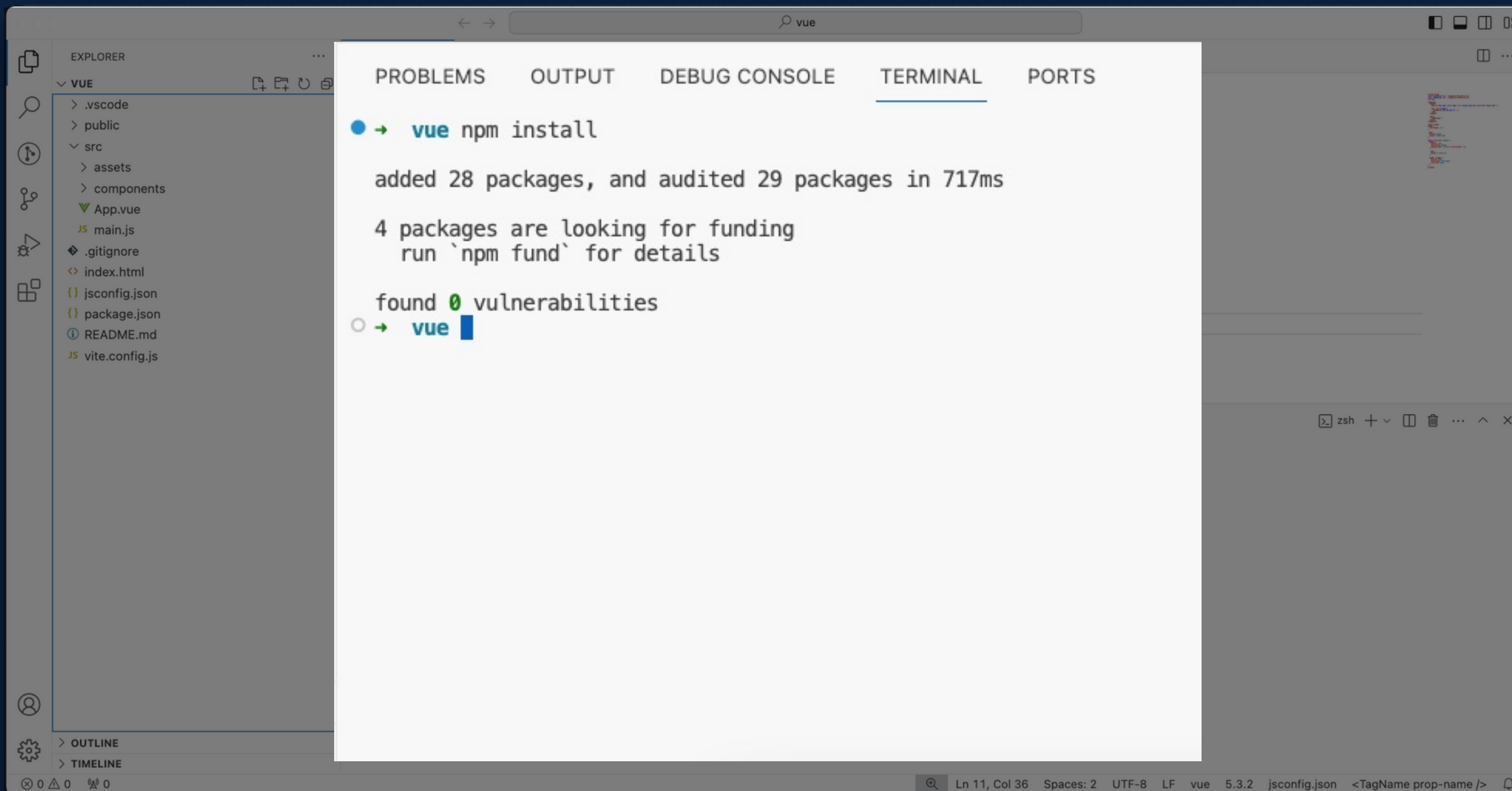
The status bar at the bottom indicates the current file is `vue` at line 11, column 36, with 2 spaces, UTF-8 encoding, LF line endings, and version 5.3.2.



Uruchamianie - instalowanie zależności



Uruchamianie - instalowanie zależności



The screenshot shows a VS Code editor window with a terminal pane open. The terminal displays the output of the command `vue npm install`. The output indicates that 28 packages were added and 29 packages were audited in 717ms. It also shows that 4 packages are looking for funding and that 0 vulnerabilities were found. The terminal prompt is `vue`.

```
vue npm install

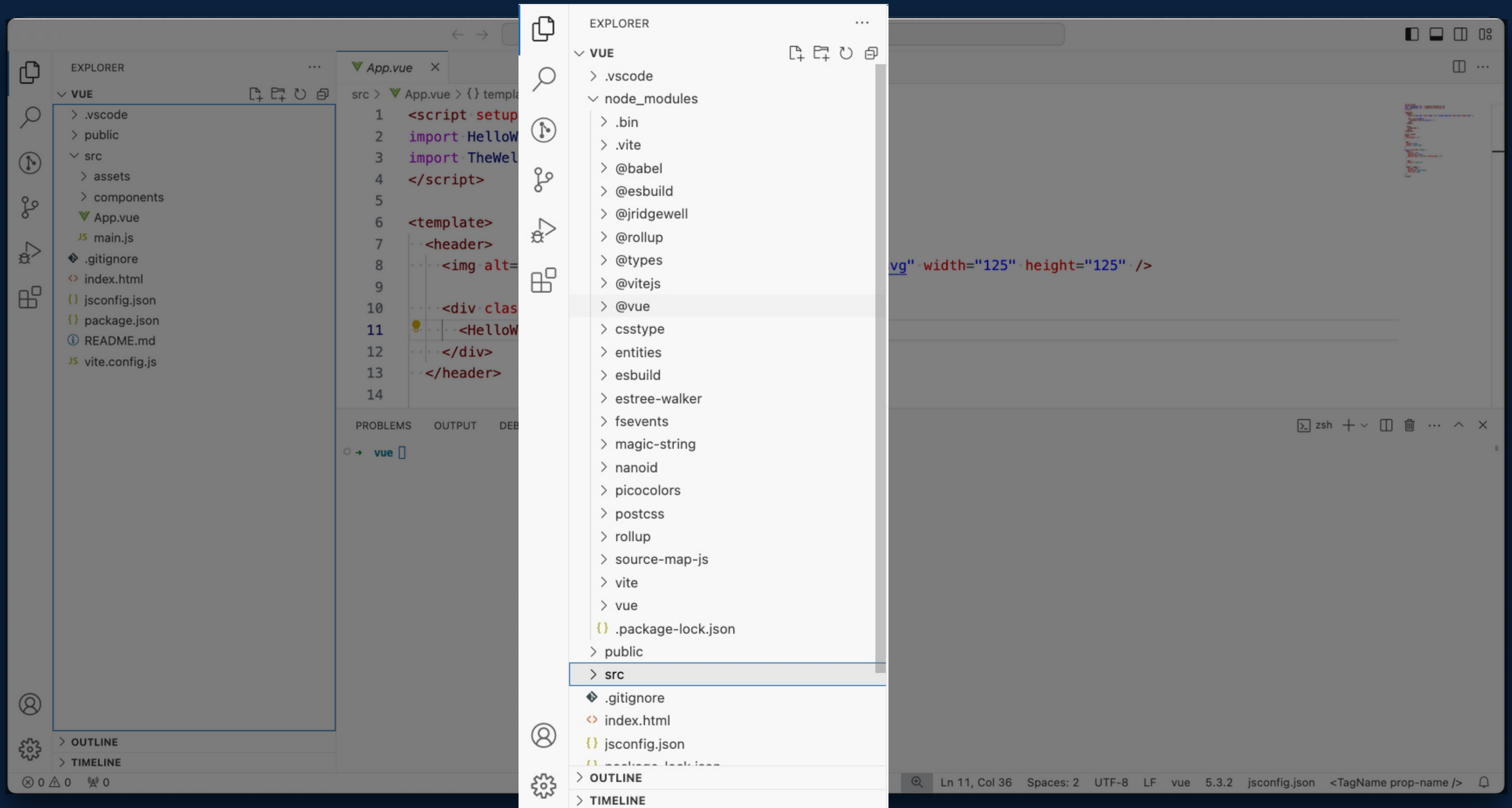
added 28 packages, and audited 29 packages in 717ms

4 packages are looking for funding
  run `npm fund` for details

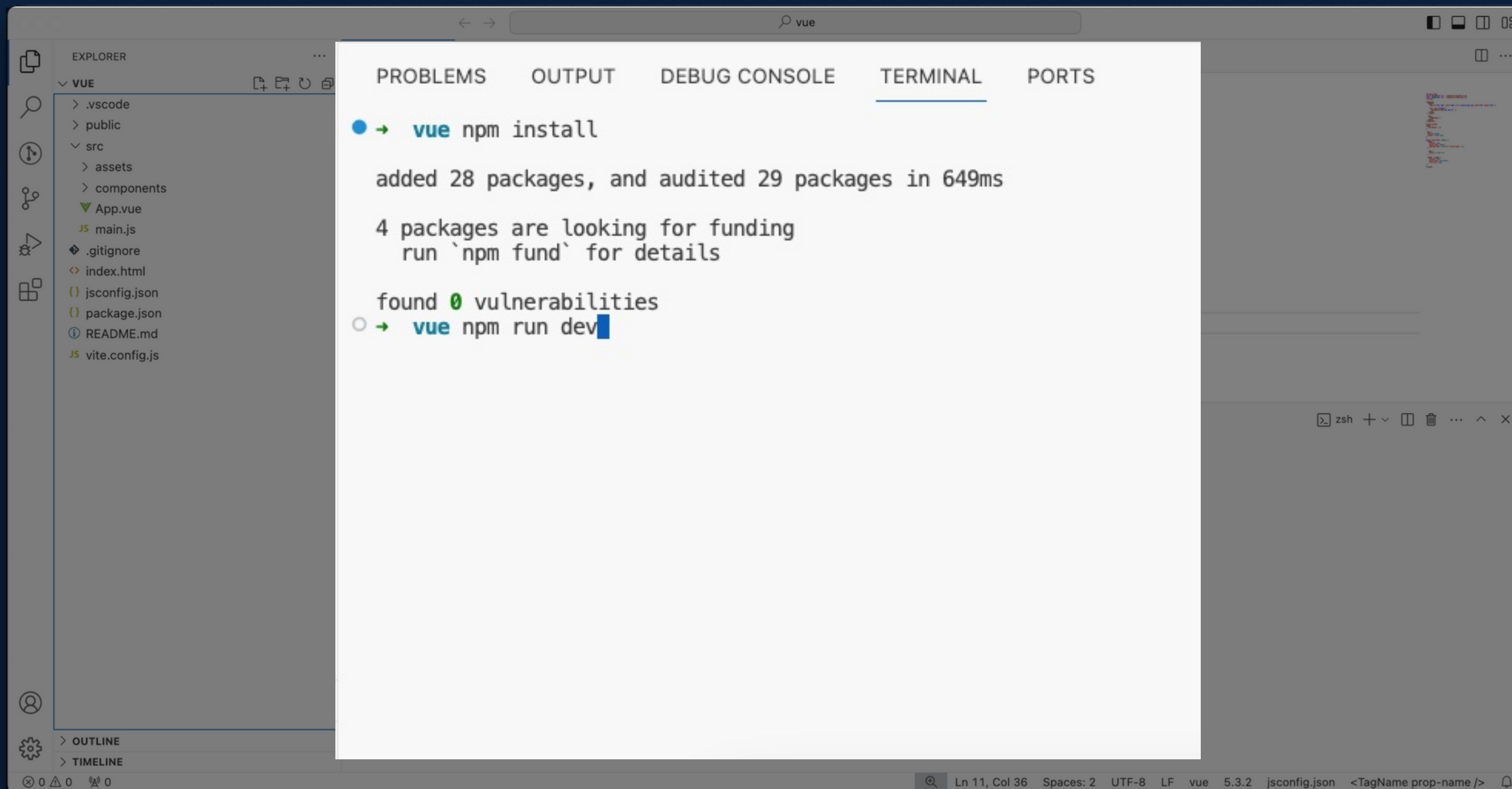
found 0 vulnerabilities
vue
```



Uruchamianie - instalowanie zależności



Uruchamianie - budowanie lokalne



Uruchamianie - budowanie lokalne

The screenshot shows a VS Code window with the terminal open. The terminal output is as follows:

```
vue
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
○ → vue npm run dev
> vue@0.0.0 dev
> vite

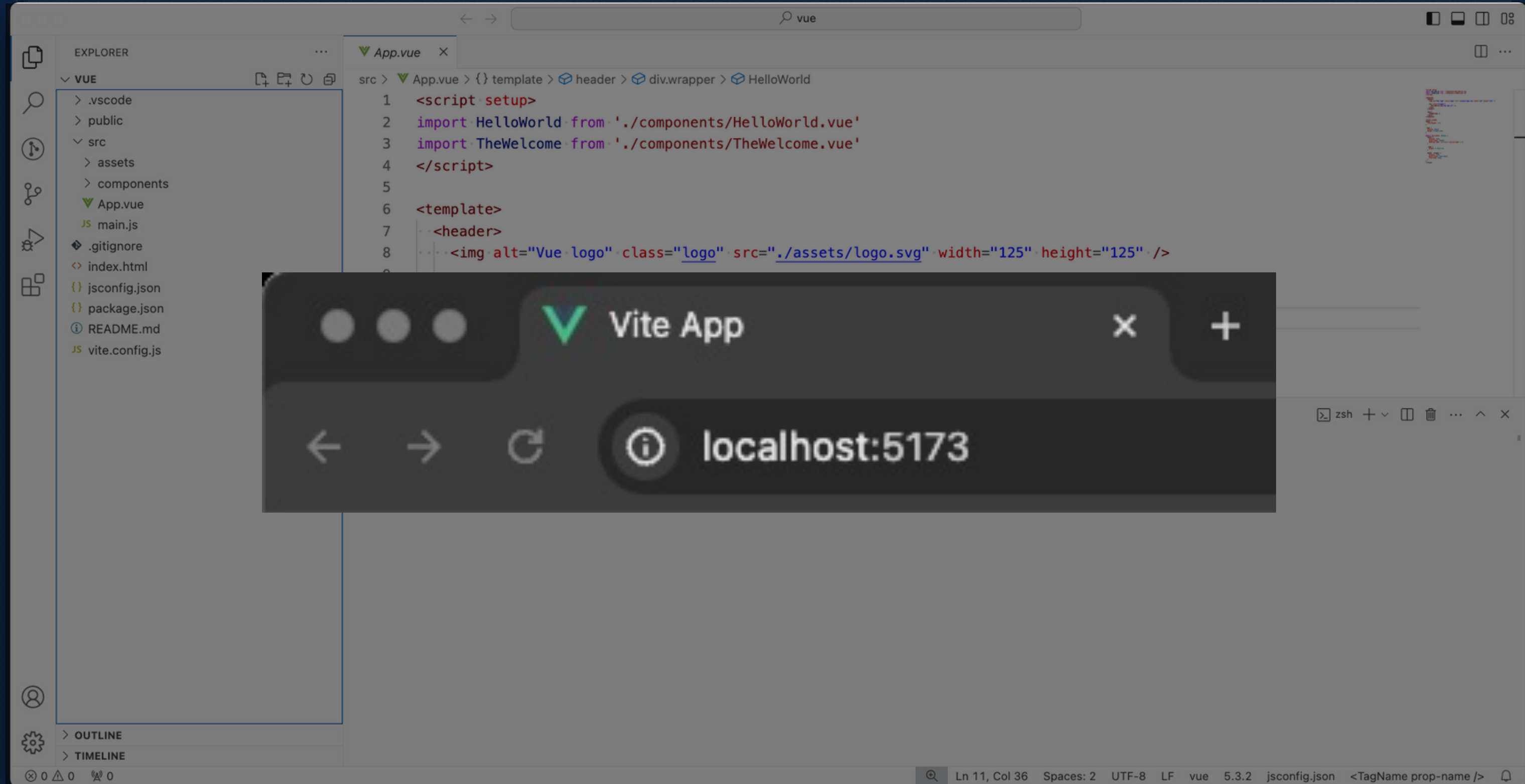
VITE v5.1.3 ready in 454 ms

→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

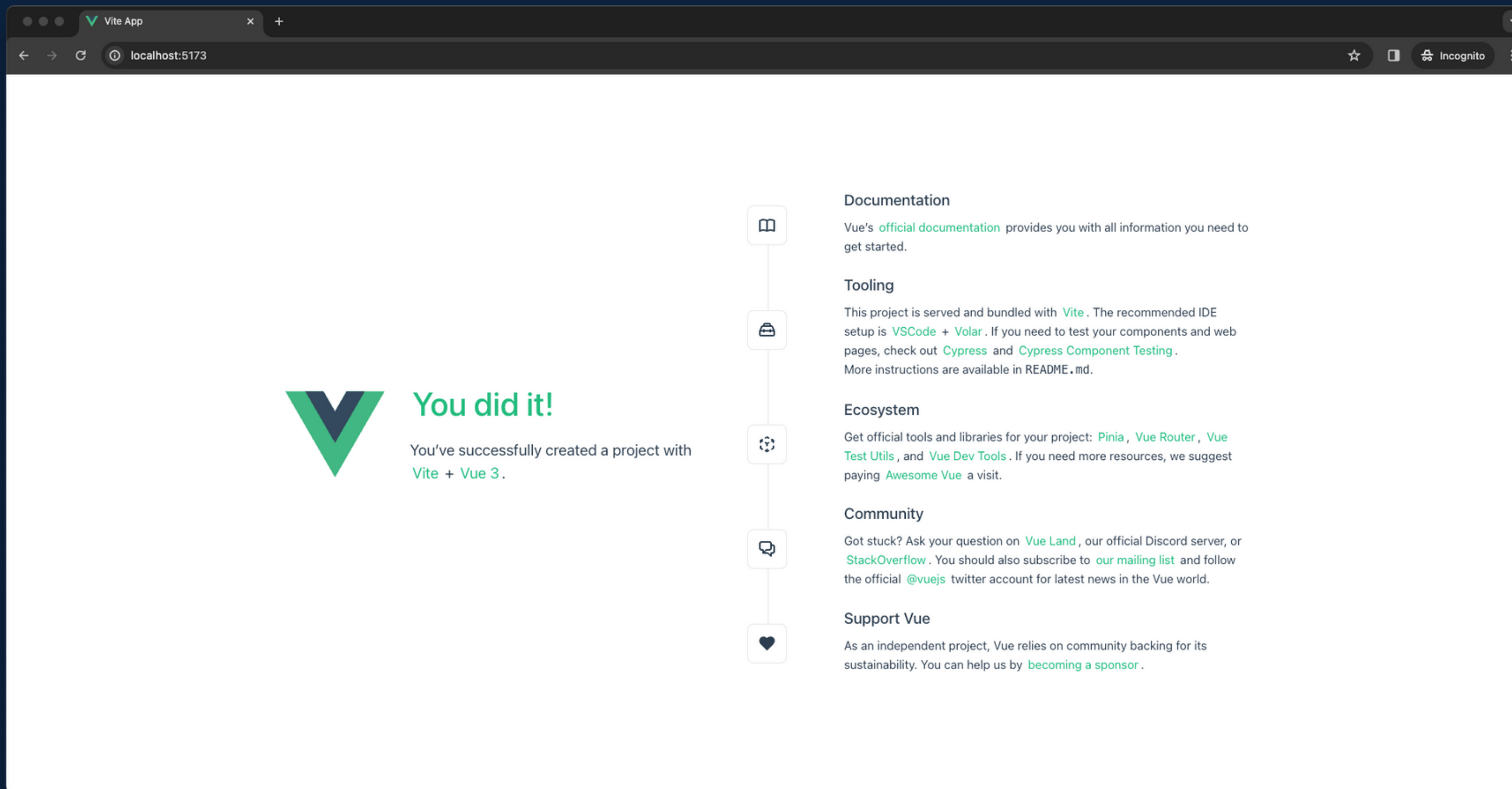
The Explorer sidebar on the left shows a project structure for 'VUE' with files like `App.vue`, `main.js`, `index.html`, `package.json`, and `vite.config.js`.



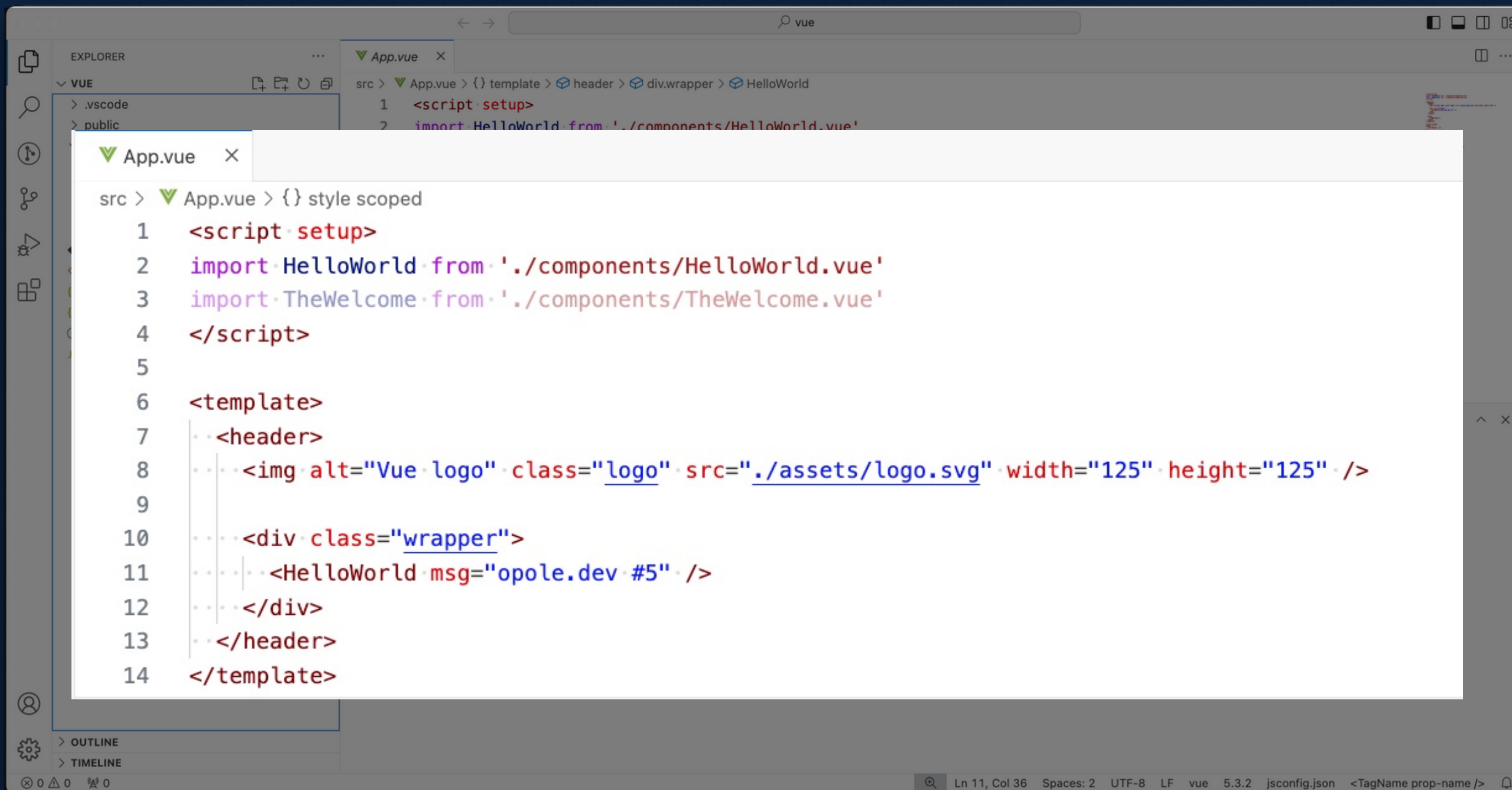
Uruchamianie - efekt



Uruchamianie - efekt



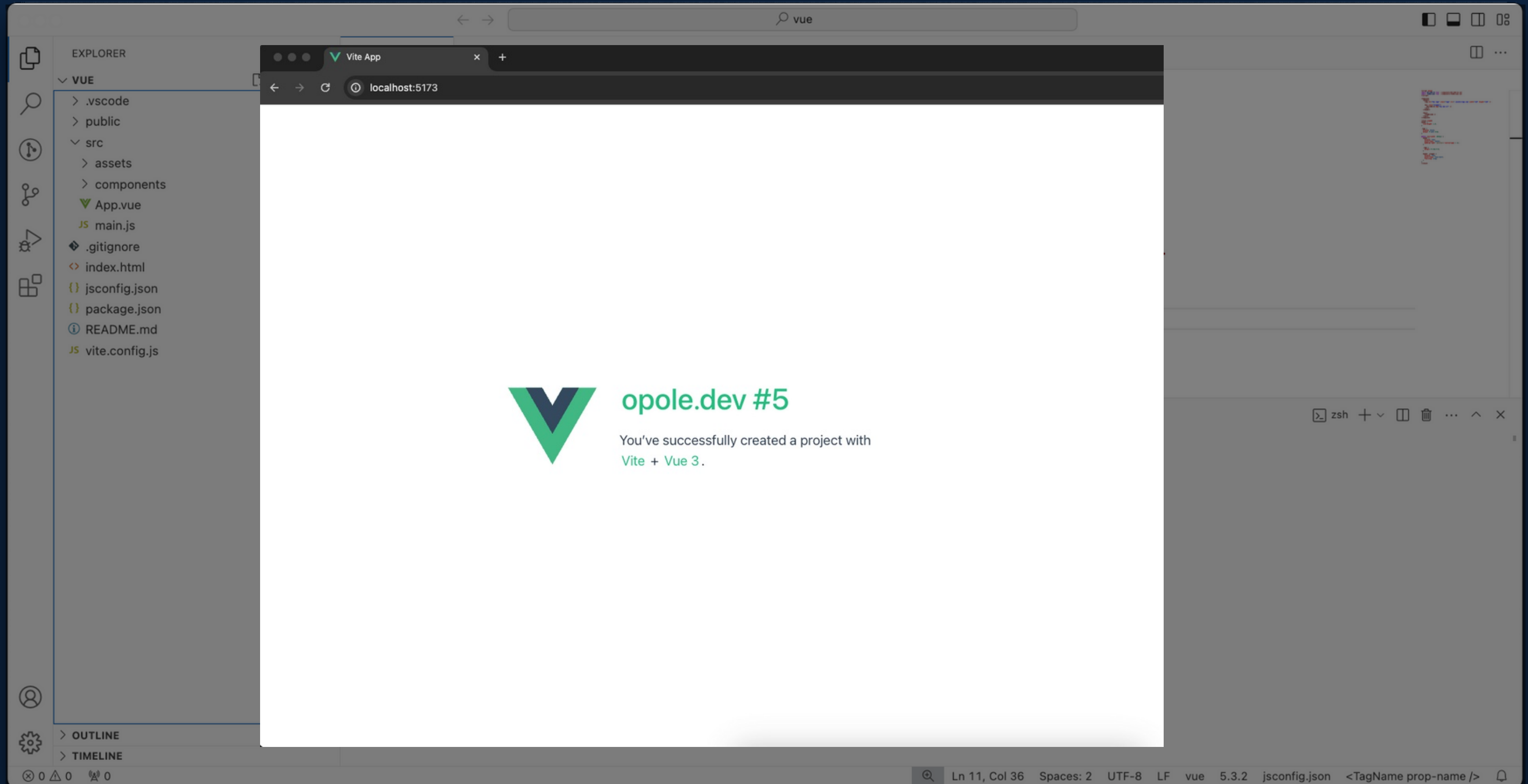
Rozwój



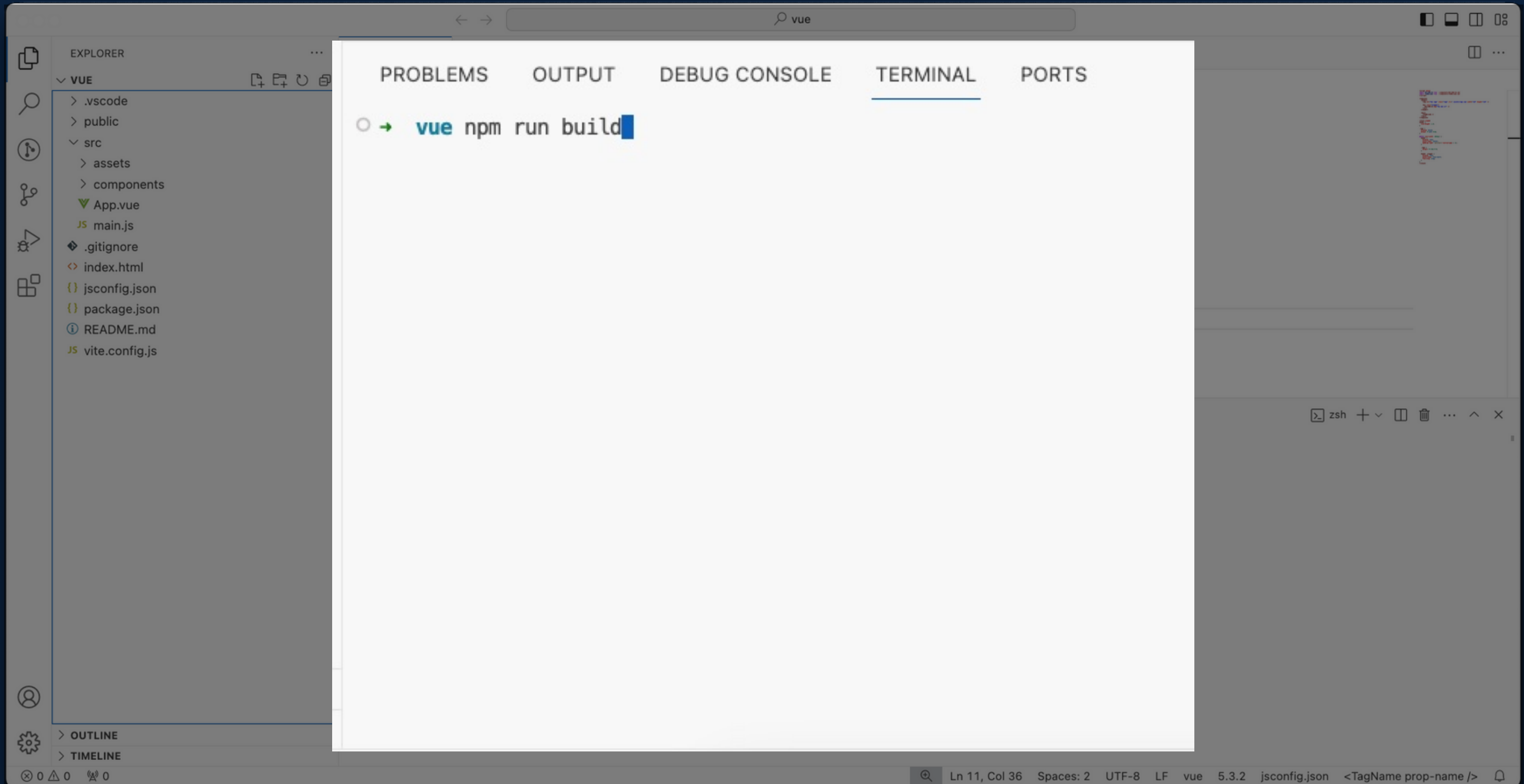
```
src > App.vue > {} style scoped
1  <script setup>
2  import HelloWorld from './components/HelloWorld.vue'
3  import TheWelcome from './components/TheWelcome.vue'
4  </script>
5
6  <template>
7  .. <header>
8  ..   
9
10  ..   <div class="wrapper">
11  ..     <HelloWorld msg="opole.dev #5" />
12  ..   </div>
13  .. </header>
14  </template>
```



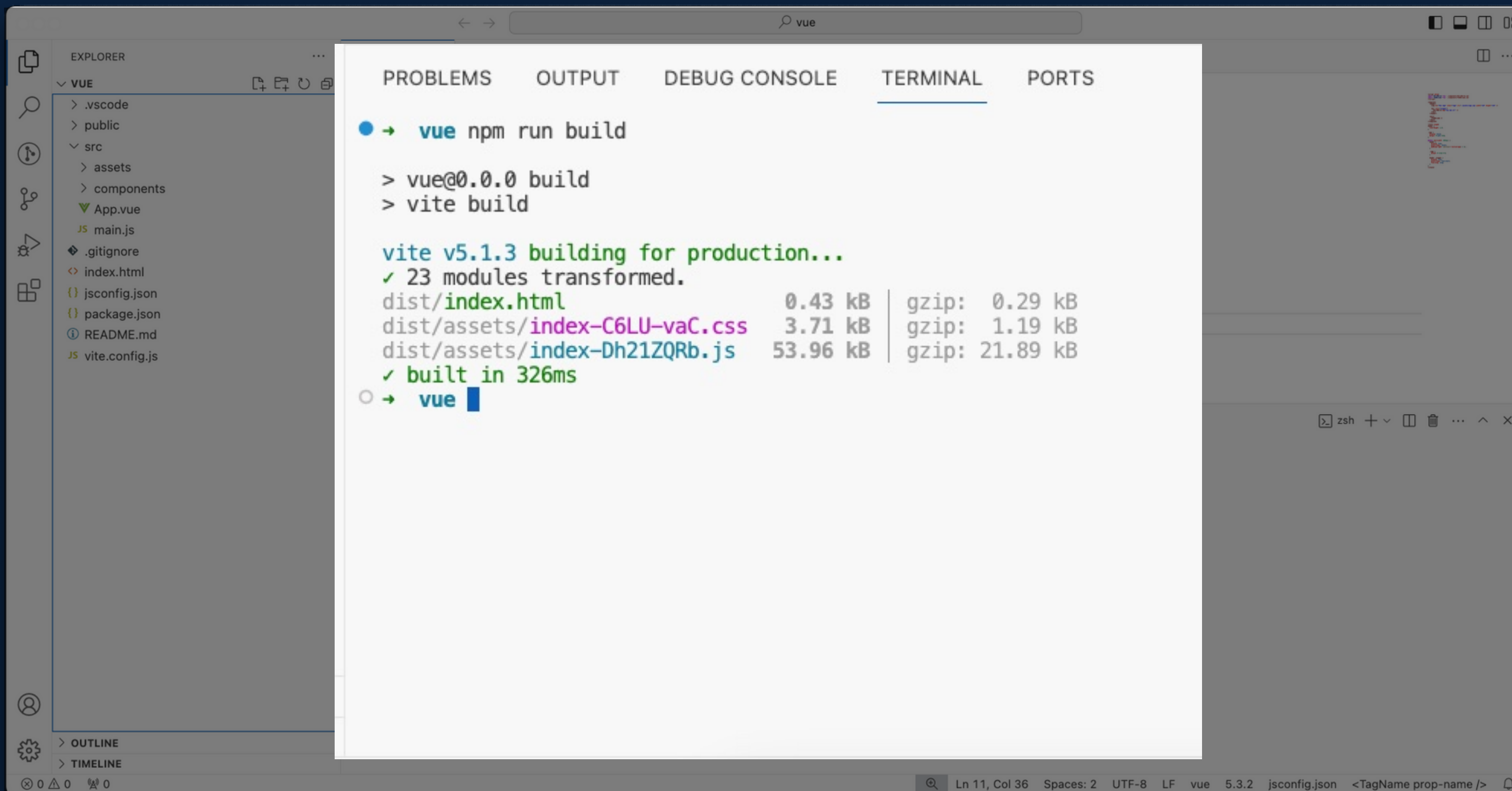
Rozwój



Budowanie wersji produkcyjnej



Budowanie wersji produkcyjnej



The screenshot shows a VS Code editor window with a terminal pane open. The terminal displays the output of the command `vue npm run build`. The output shows that Vite v5.1.3 is building for production, transforming 23 modules. The build output includes the following files and their sizes:

| File | Size | Gzip Size |
|--------------------------------|----------|-----------|
| dist/index.html | 0.43 kB | 0.29 kB |
| dist/assets/index-C6LU-vaC.css | 3.71 kB | 1.19 kB |
| dist/assets/index-Dh21ZQRb.js | 53.96 kB | 21.89 kB |

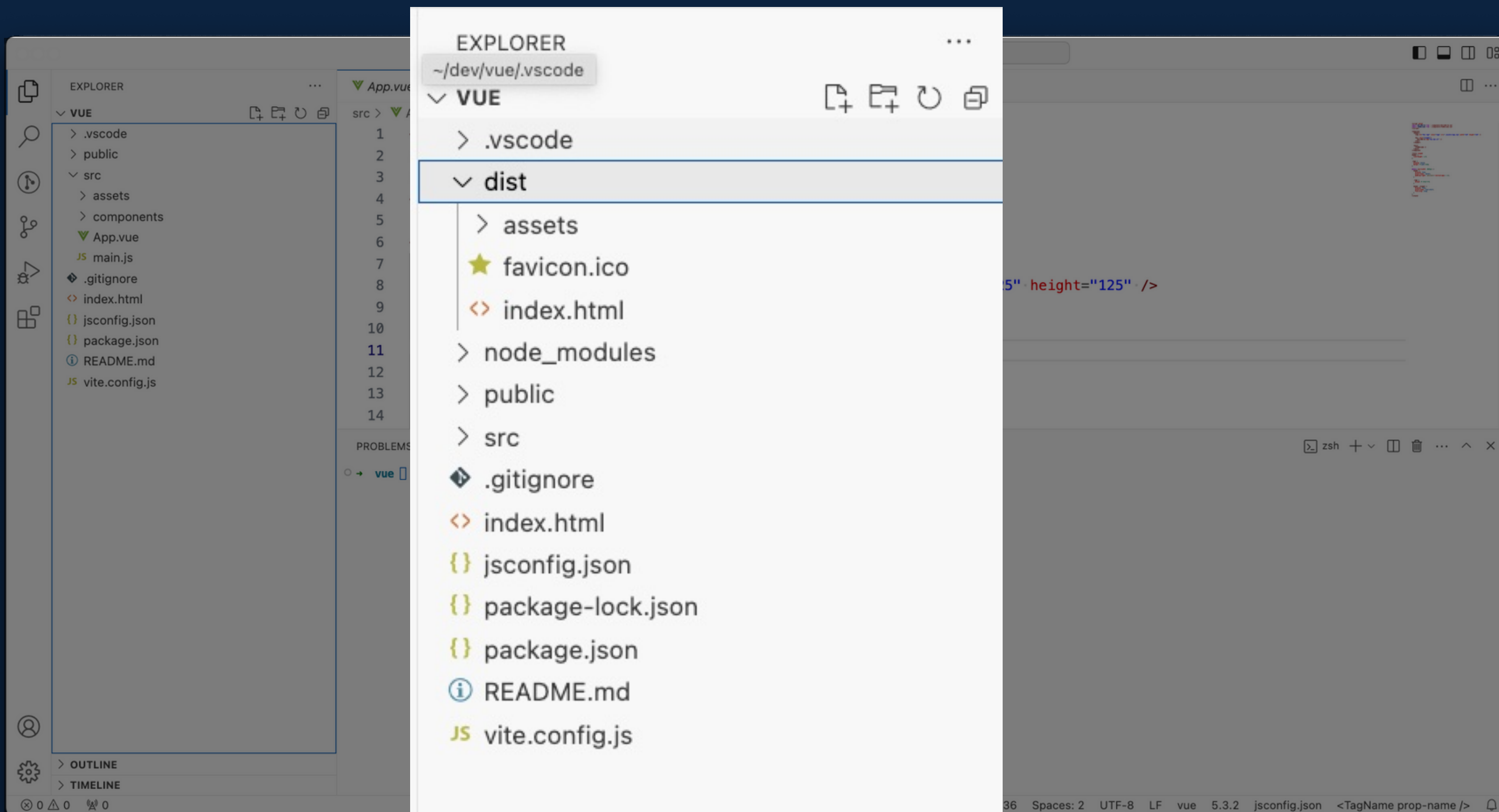
The terminal also shows that the build was completed in 326ms. The terminal prompt is `vue`.

```
vue npm run build
vue@0.0.0 build
vite build

vite v5.1.3 building for production...
✓ 23 modules transformed.
dist/index.html           0.43 kB | gzip: 0.29 kB
dist/assets/index-C6LU-vaC.css 3.71 kB | gzip: 1.19 kB
dist/assets/index-Dh21ZQRb.js 53.96 kB | gzip: 21.89 kB
✓ built in 326ms
vue
```



Budowanie wersji produkcyjnej



Oprogramowanie “napisane”



Oprogramowanie “napisane”

Zbudowane...



Oprogramowanie “napisane”

Zbudowane... Co dalej?



Udostępnianie



FTP



Jest prosto i działa...



Jest prosto i działa...

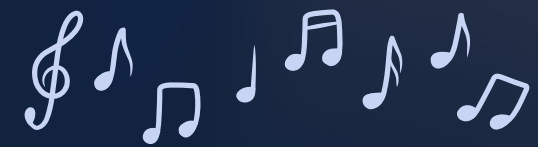
W czym problem? Po co zmieniać?





Zagrozenia

TUDU DUDU



Zmiany bezpośrednio na serwerze



Brak “oficjalnej” wersji
aplikacji



Błędy w transmisji danych



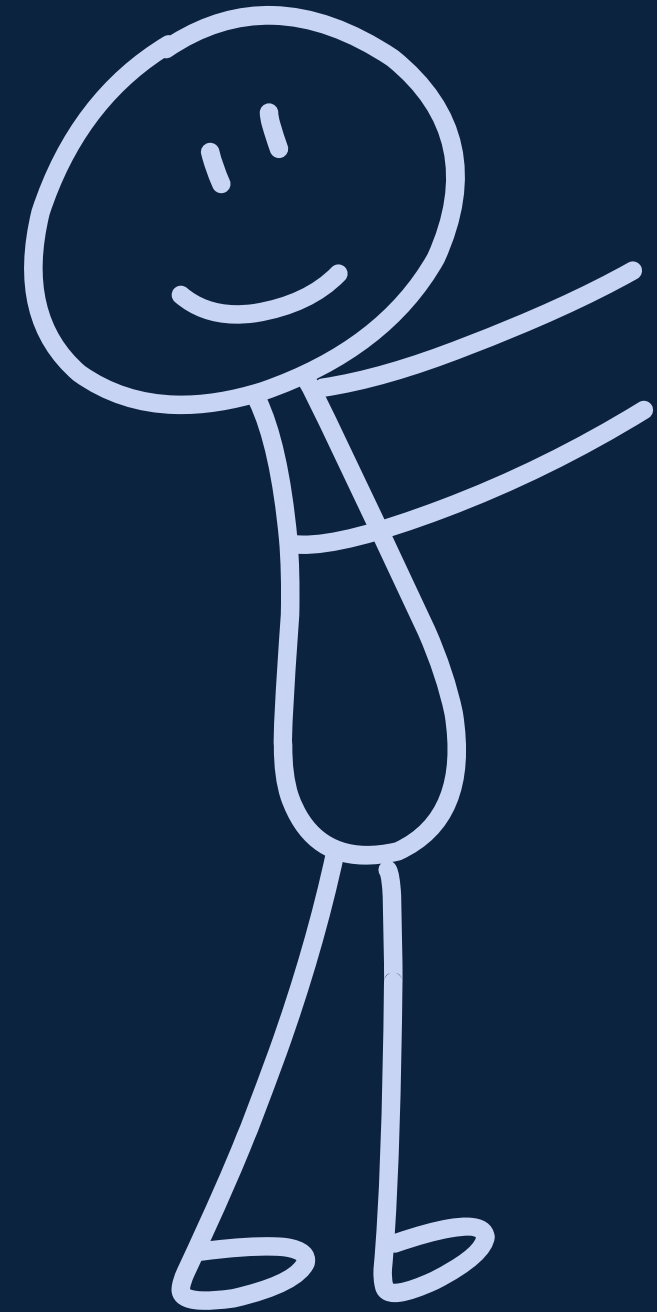
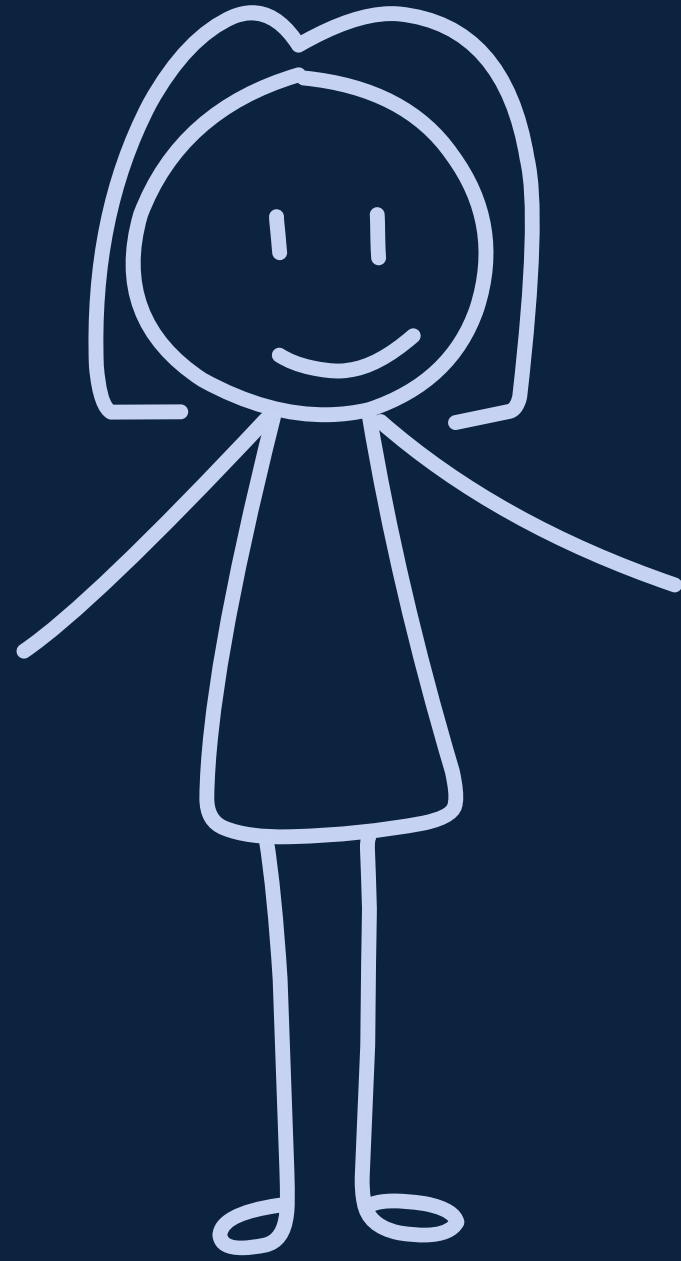
“Zapomniałem przebudować”



Trzeba to robić ręcznie



John z zespołem



Tak się nie da pracować!



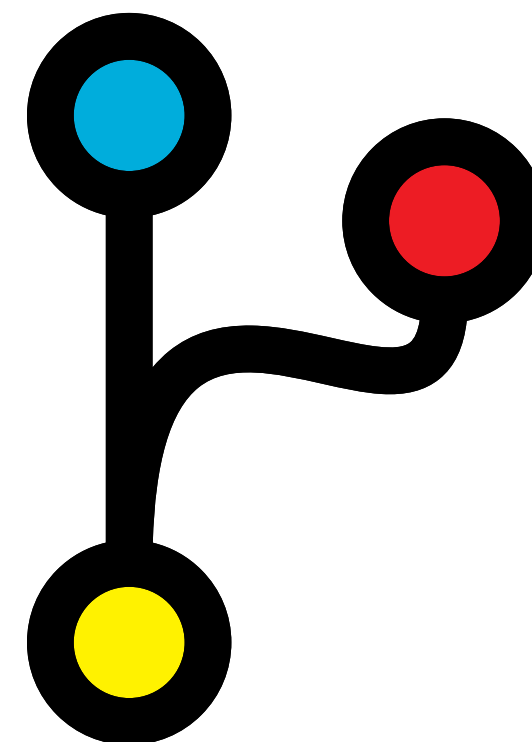
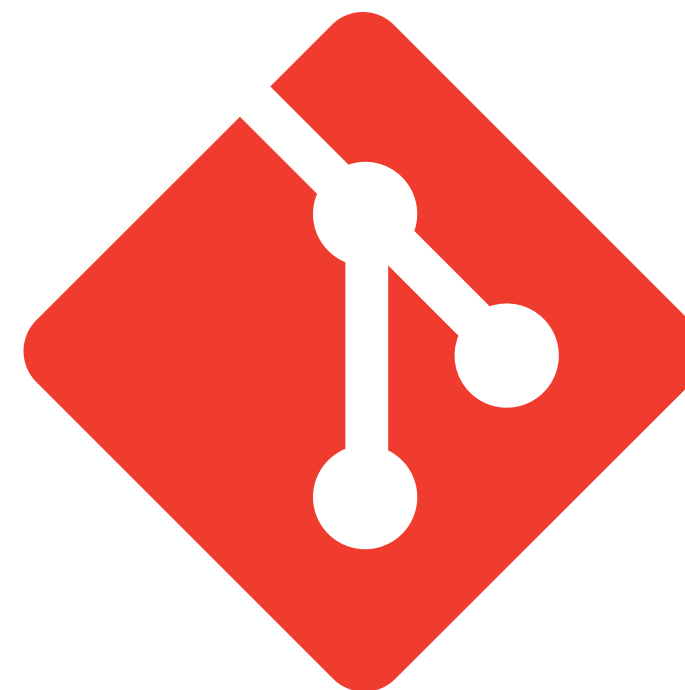
Rozdział 1

Git



Git

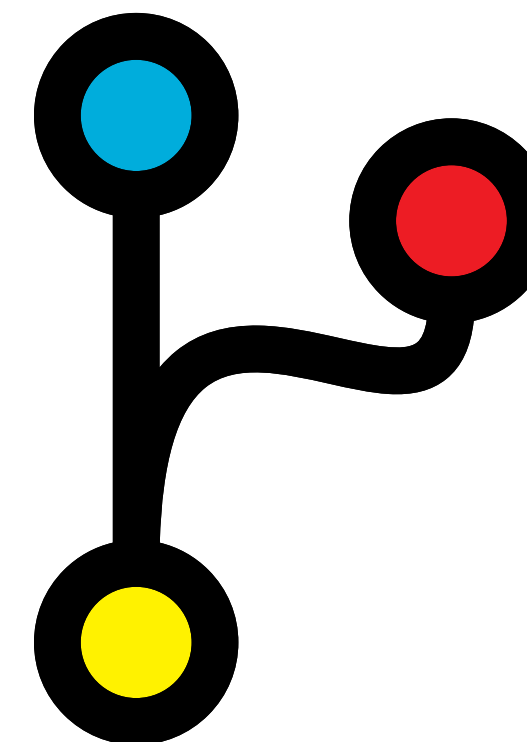
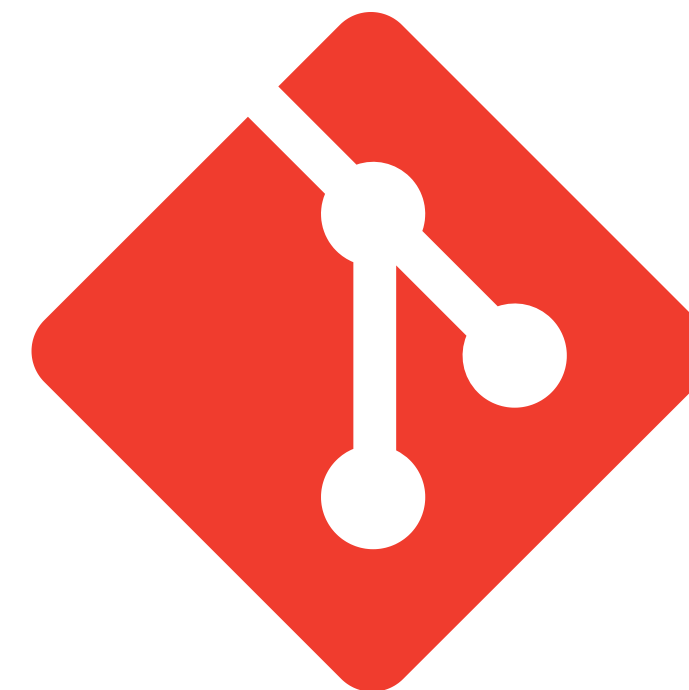
Jedno z najważniejszych narzędzi!



Git

Jedno z najważniejszych narzędzi!

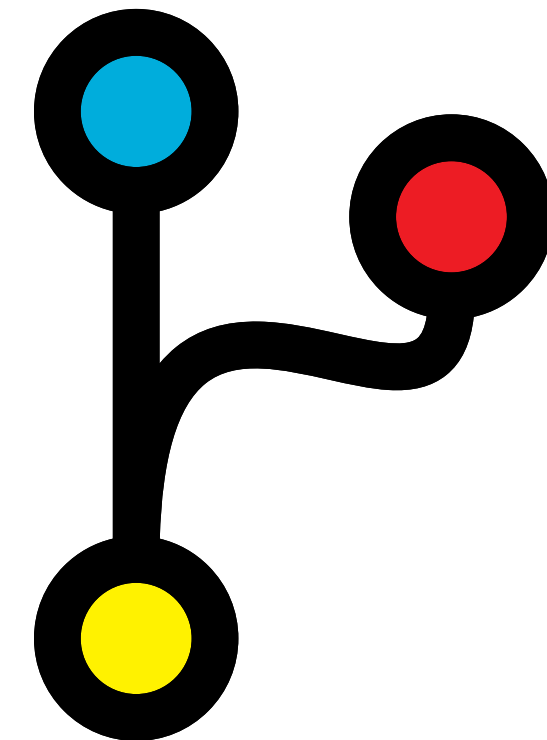
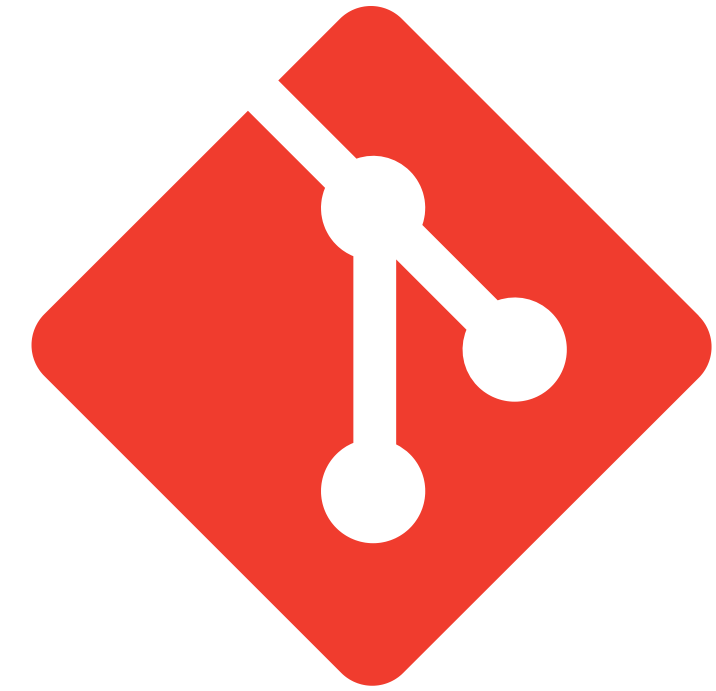
- Wersjonowanie kodu



Git

Jedno z najważniejszych narzędzi!

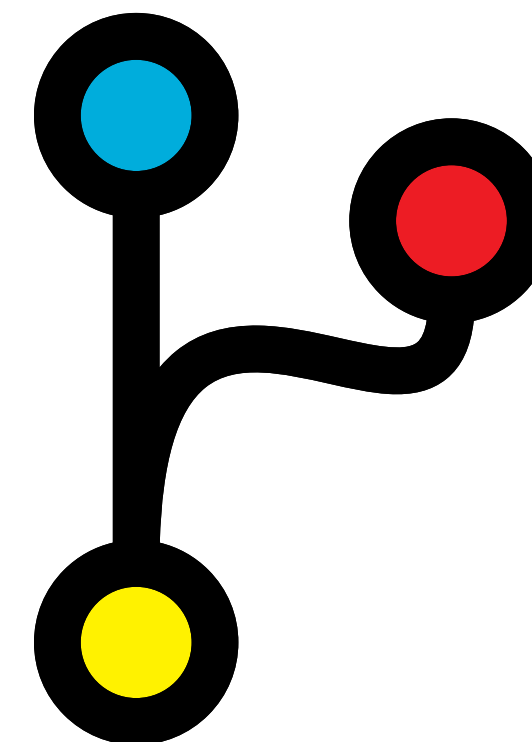
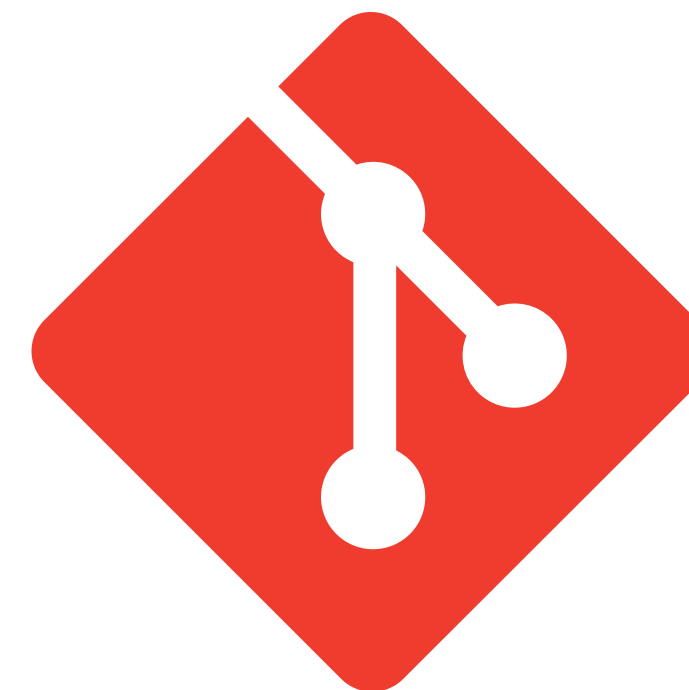
- Wersjonowanie kodu
- Równoległa praca nad wieloma zmianami



Git

Jedno z najważniejszych narzędzi!

- Wersjonowanie kodu
- Równoległa praca nad wieloma zmianami
- Łączenie zmian ze sobą

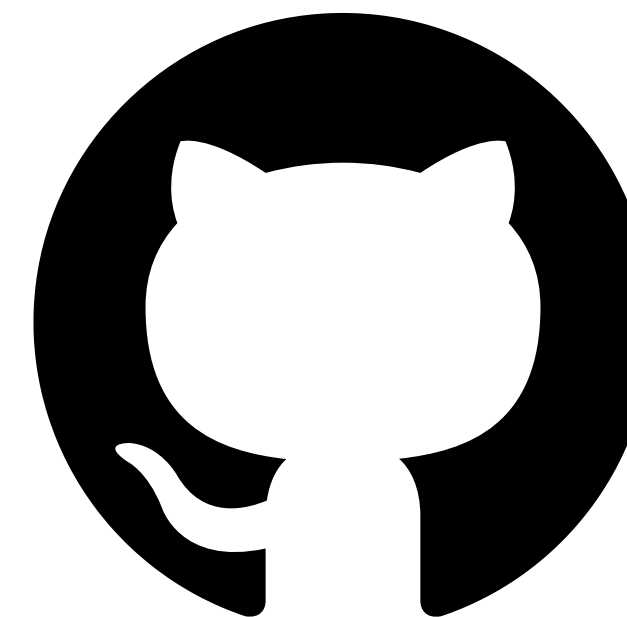




Platformy

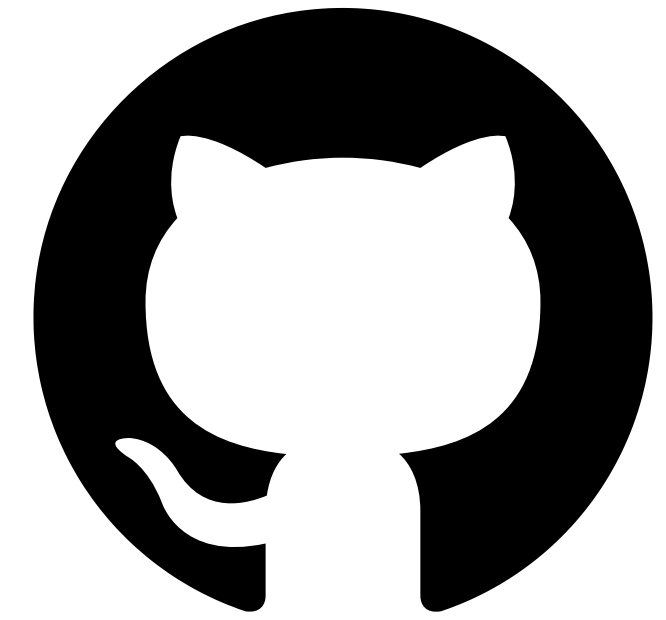
Platformy

- GitHub



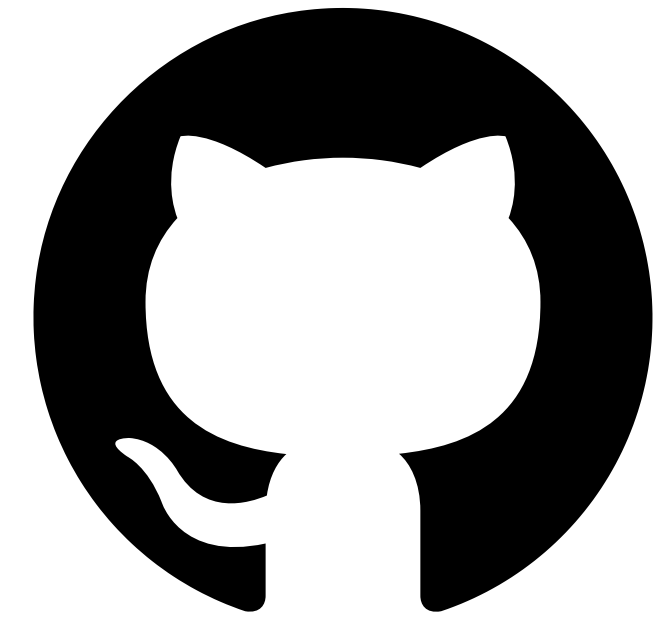
Platformy

- GitHub
- GitLab



Platformy

- GitHub
- GitLab
- Bitbucket



Co zyskał John dzięki narzędziu Git?

Jest git :)

- Łatwiejsza praca zespołowa
- Główne źródło wiedzy o kodzie aplikacji
- Możliwość równoległej pracy na wielu wersjach kodu
- Możliwość łączenia różnych wersji kodu



Git jako narzędzie wdrażania (git pull)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

- → **vue git:(master)** `x git commit -am "feature: add something new to opole.dev"`
[master 1750f39] feature: add something new to opole.dev
1 file changed, 1 insertion(+), 1 deletion(-)
- → **vue git:(master)** `git push`
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 463 bytes | 463.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
To ssh://[redacted]:[redacted]@k.sobolewski/vue.git
 cade327..1750f39 master -> master
- → **vue git:(master)** `ssh user@server`



Git jako narzędzie wdrażania (git pull)

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

GITLENS

- → **vue git:(master)** `git commit -am "feature: add something new to opole.dev"`
[master 1750f39] feature: add something new to opole.dev
1 file changed, 1 insertion(+), 1 deletion(-)
- → **vue git:(master)** `git push`
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 463 bytes | 463.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
To ssh://[redacted]:[redacted]@k.sobolewski/vue.git
 cade327..1750f39 master -> master
- → **vue git:(master)** `ssh user@server`



Git jako narzędzie wdrażania (git pull)

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

GITLENS

- → `vue git:(master) x git commit -am "feature: add something new to opole.dev"`
[master 1750f39] feature: add something new to opole.dev
1 file changed, 1 insertion(+), 1 deletion(-)
- → `vue git:(master) git push`
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 463 bytes | 463.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
To ssh://[redacted]:[redacted]@k.sobolewski/vue.git
cade327..1750f39 master -> master
- → `vue git:(master) ssh user@server`



Git jako narzędzie wdrażania (git pull)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

- → **vue git:(master)** x git commit -am "feature: add something new to opole.dev"
[master 1750f39] feature: add something new to opole.dev
1 file changed, 1 insertion(+), 1 deletion(-)
- → **vue git:(master)** git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 463 bytes | 463.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
To ssh://[redacted]:[redacted]@k.sobolewski/vue.git
 cade327..1750f39 master -> master
- → **vue git:(master)** ssh user@server



Git jako narzędzie wdrażania (git pull)

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

GITLENS

```
● → vue git:(master) git pull
Updating cade327..1750f39
Fast-forward
 src/App.vue | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
○ → vue git:(master) █
```



Git jako narzędzie wdrażania (git pull)

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

GITLENS

```
● → vue git:(master) git pull
Updating cade327..1750f39
Fast-forward
 src/App.vue | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
○ → vue git:(master) █
```



Git jako narzędzie wdrażania (git tag)

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS
```

- → **vue git:(master)** git commit -am "feature: add sth really new to opole.dev"
[master bb848d8] feature: add sth really new to opole.dev
1 file changed, 1 insertion(+), 1 deletion(-)
- → **vue git:(master)** git tag v1.2
- → **vue git:(master)** git push && git push --tags
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 897 bytes | 897.00 KiB/s, done.
Total 8 (delta 4), reused 0 (delta 0), pack-reused 0
To ssh://[redacted]/k.sobolewski/vue.git
1750f39..bb848d8 master -> master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To ssh://[redacted]/k.sobolewski/vue.git
* [new tag] v1.1 -> v1.1
* [new tag] v1.2 -> v1.2
- → **vue git:(master)** ssh user@server



Git jako narzędzie wdrażania (git tag)

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS
```

- → **vue git:(master)** git commit -am "feature: add sth really new to opole.dev"
[master bb848d8] feature: add sth really new to opole.dev
1 file changed, 1 insertion(+), 1 deletion(-)
- → **vue git:(master)** git tag v1.2
- → **vue git:(master)** git push && git push --tags
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 897 bytes | 897.00 KiB/s, done.
Total 8 (delta 4). reused 0 (delta 0), pack-reused 0
To ssh://_ /k.sobolewski/vue.git
1750f39..bb848d8 master -> master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To ssh://_ /k.sobolewski/vue.git
* [new tag] v1.1 -> v1.1
* [new tag] v1.2 -> v1.2
- → **vue git:(master)** ssh user@server



Git jako narzędzie wdrażania (git tag)

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS
```

- → `vue git:(master)` `git commit -am "feature: add sth really new to opole.dev"`
[master bb848d8] feature: add sth really new to opole.dev
1 file changed, 1 insertion(+), 1 deletion(-)
- → `vue git:(master)` `git tag v1.2`
- → `vue git:(master)` `git push && git push --tags`
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 897 bytes | 897.00 KiB/s, done.
Total 8 (delta 4), reused 0 (delta 0), pack-reused 0
To ssh://[redacted]@k.sobolewski/vue.git
1750f39..bb848d8 master -> master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To ssh://[redacted]@k.sobolewski/vue.git
* [new tag] v1.1 -> v1.1
* [new tag] v1.2 -> v1.2
- → `vue git:(master)` `ssh user@server`



Git jako narzędzie wdrażania (git tag)

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS
```

- → `vue git:(master) git commit -am "feature: add sth really new to opole.dev"`
[master bb848d8] feature: add sth really new to opole.dev
1 file changed, 1 insertion(+), 1 deletion(-)
- → `vue git:(master) git tag v1.2`
- → `vue git:(master) git push && git push --tags`
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 897 bytes | 897.00 KiB/s, done.
Total 8 (delta 4), reused 0 (delta 0), pack-reused 0
To ssh://[redacted]@k.sobolewski/vue.git
1750f39..bb848d8 master -> master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To ssh://[redacted]@k.sobolewski/vue.git
* [new tag] v1.1 -> v1.1
* [new tag] v1.2 -> v1.2

```
○ → vue git:(master) ssh user@server
```



Git jako narzędzie wdrażania (git tag)

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    GITLENS
```

- → **vue git:(v1.1)** git status
HEAD detached at v1.1
nothing to commit, working tree clean
- → **vue git:(v1.1)** git fetch
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 445 bytes | 111.00 KiB/s, done.
From ssh:// /k.sobolewski/vue
 bb848d8..467f593 master -> origin/master
 * [new tag] v1.2 -> v1.2
- → **vue git:(v1.1)** git checkout v1.2
Previous HEAD position was 1750f39 feature: add something new to opole.dev
HEAD is now at 467f593 feature: add sth really new to opole.dev
- → **vue git:(v1.2)** █



Git jako narzędzie wdrażania (git tag)

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    GITLENS
```

```
● → vue git:(v1.1) git status
HEAD detached at v1.1
nothing to commit, working tree clean
```

```
● → vue git:(v1.1) git fetch
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 445 bytes | 111.00 KiB/s, done.
From ssh://          /k.sobolewski/vue
   bb848d8..467f593  master    -> origin/master
  * [new tag]       v1.2      -> v1.2
```

```
● → vue git:(v1.1) git checkout v1.2
Previous HEAD position was 1750f39 feature: add something new to opole.dev
HEAD is now at 467f593 feature: add sth really new to opole.dev
```

```
○ → vue git:(v1.2) █
```



Git jako narzędzie wdrażania (git tag)

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS
```

- → **vue git:(v1.1)** git status
HEAD detached at v1.1
nothing to commit, working tree clean
- → **vue git:(v1.1)** git fetch
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 445 bytes | 111.00 KiB/s, done.
From ssh:// /k.sobolewski/vue
 bb848d8..467f593 master -> origin/master
* [new tag] v1.2 -> v1.2
- → **vue git:(v1.1)** git checkout v1.2
Previous HEAD position was 1750f39 feature: add something new to opole.dev
HEAD is now at 467f593 feature: add sth really new to opole.dev
- → **vue git:(v1.2)** █



Git jako narzędzie wdrażania (git tag)

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS
```

- → **vue git:(v1.1)** git status
HEAD detached at v1.1
nothing to commit, working tree clean
- → **vue git:(v1.1)** git fetch
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 445 bytes | 111.00 KiB/s, done.
From ssh:// /k.sobolewski/vue
 bb848d8..467f593 master -> origin/master
 * [new tag] v1.2 -> v1.2
- → **vue git:(v1.1)** git checkout v1.2
Previous HEAD position was 1750f39 feature: add something new to opole.dev
HEAD is now at 467f593 feature: add sth really new to opole.dev
- → **vue git:(v1.2)** █



Co jeszcze zyskał John?

Jest jeszcze bardziej git :)

- Metody wdrożenia odporne na błędy
- Częściowa standaryzacja wdrożeń
- Łatwiejsze przywracanie zmian w przypadku błędu



Jest lepiej ale jeszcze nie jest idealnie...

- Niektóre komendy nadal trzeba uruchamiać lokalnie lub na serwerze
- Korzystanie z wdrażania za pomocą git może wymagać przechowywania w repozytorium artefaktów budowania – potencjalne konflikty podczas pracy zespołowej





Zhuowei Zhang

@zhuowei · [Follow](#)



Never spend 6 minutes doing something by hand when you can spend 6 hours failing to automate it

6:28 AM · Apr 26, 2020



4.5K



Reply



Copy link

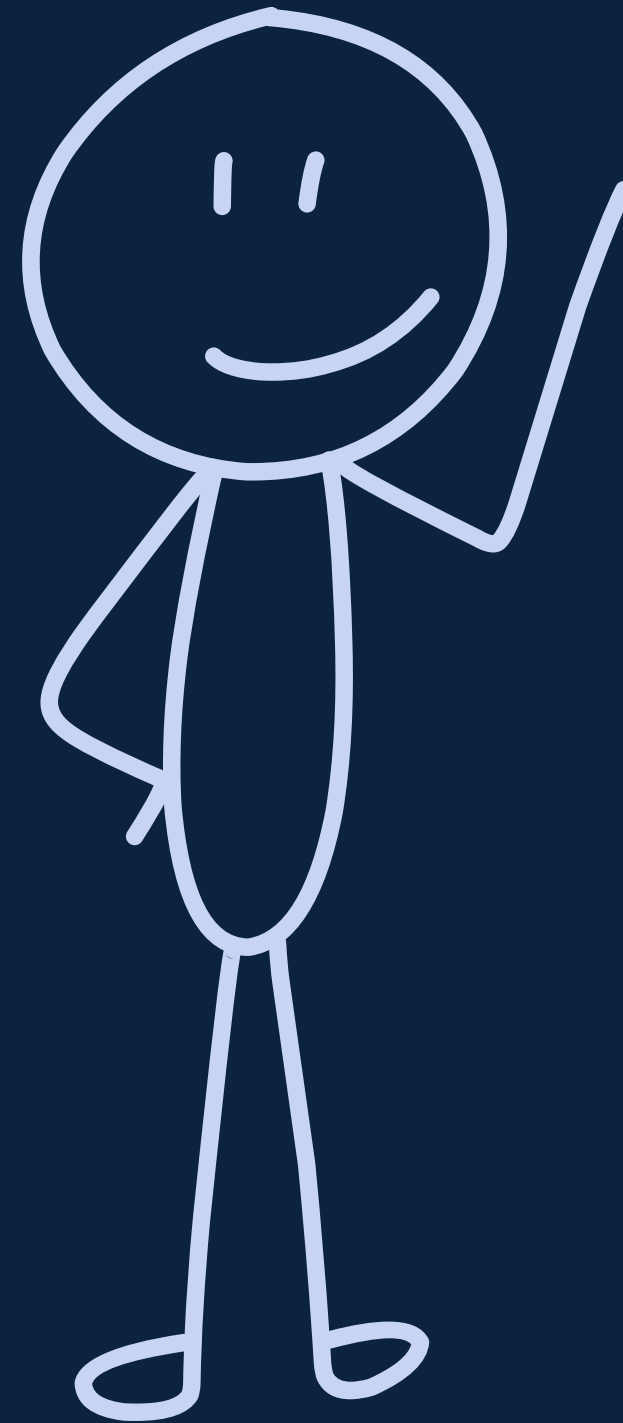


Rozdział 2

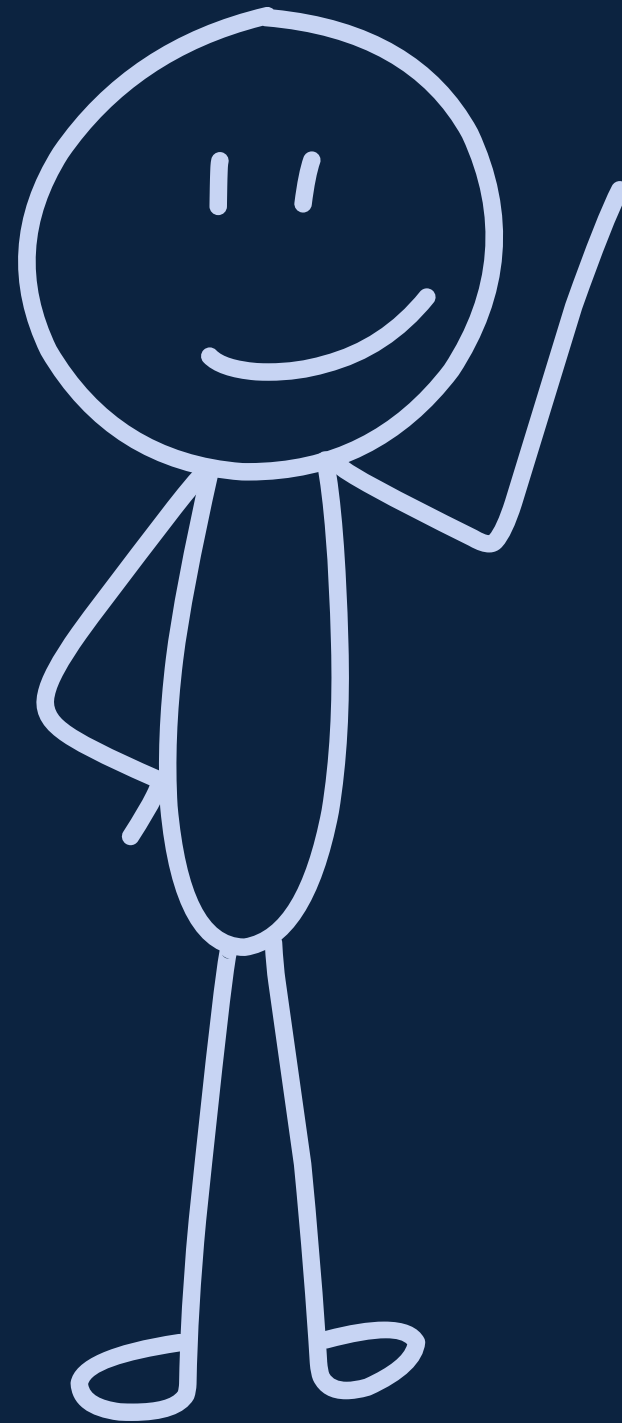
Automatyzacja



John potrafi programować



John potrafi programować



Wbudowane narzędzia automatyzacji

- GitHub Actions



Wbudowane narzędzia automatyzacji

- GitHub Actions
- Gitlab CI



Wbudowane narzędzia automatyzacji

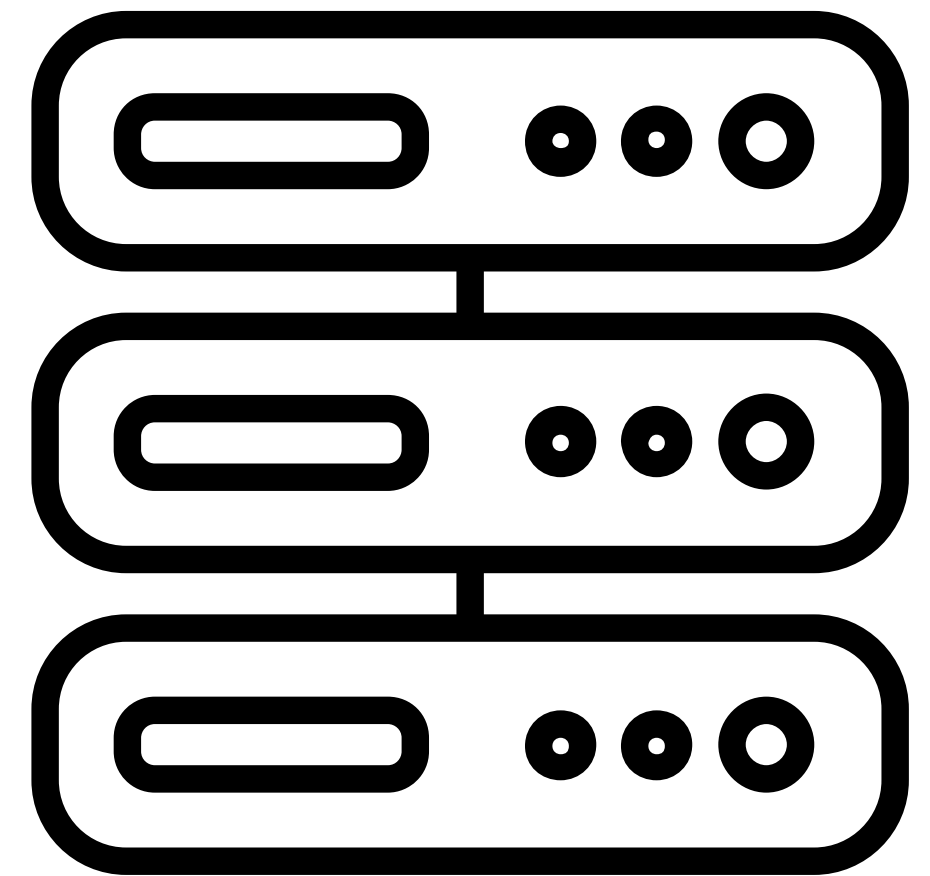
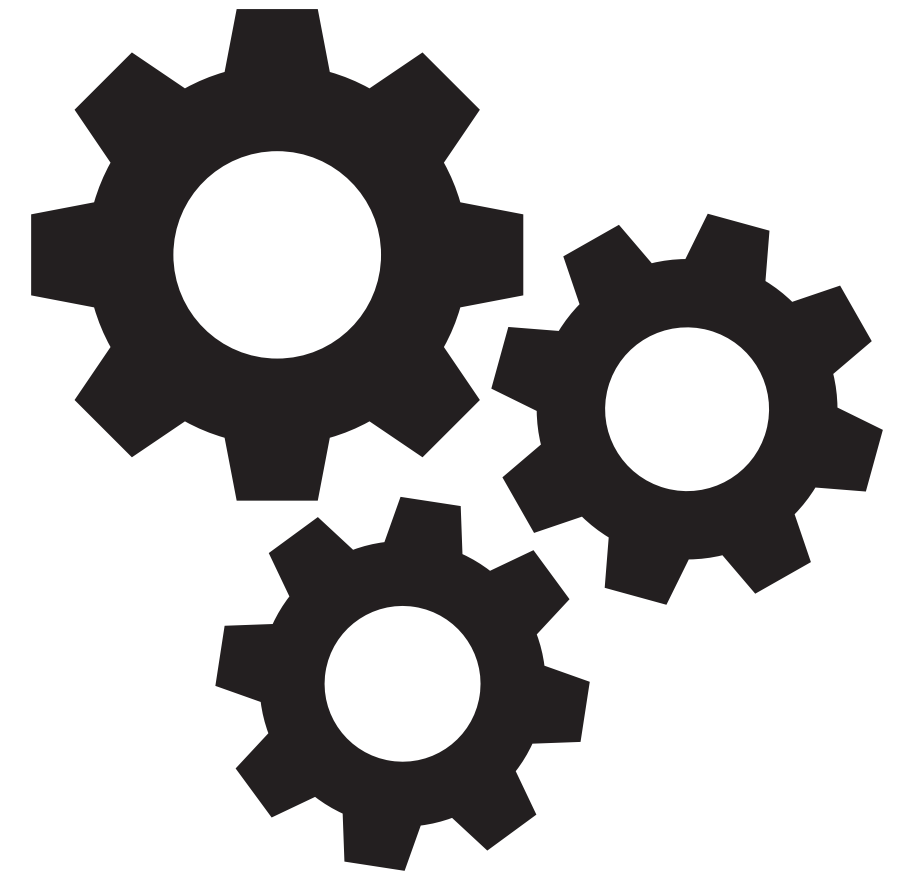
- GitHub Actions
- Gitlab CI
- Bitbucket Pipelines



Runner

“Inny komputer” (serwer), który ma zainstalowane specjalne oprogramowanie i uruchamia zadania.

Cyklicznie odpytuje platformę o to, czy pojawiły się do wykonania zadania – najczęściej gdy do repozytorium dodawane są nowe zmiany.



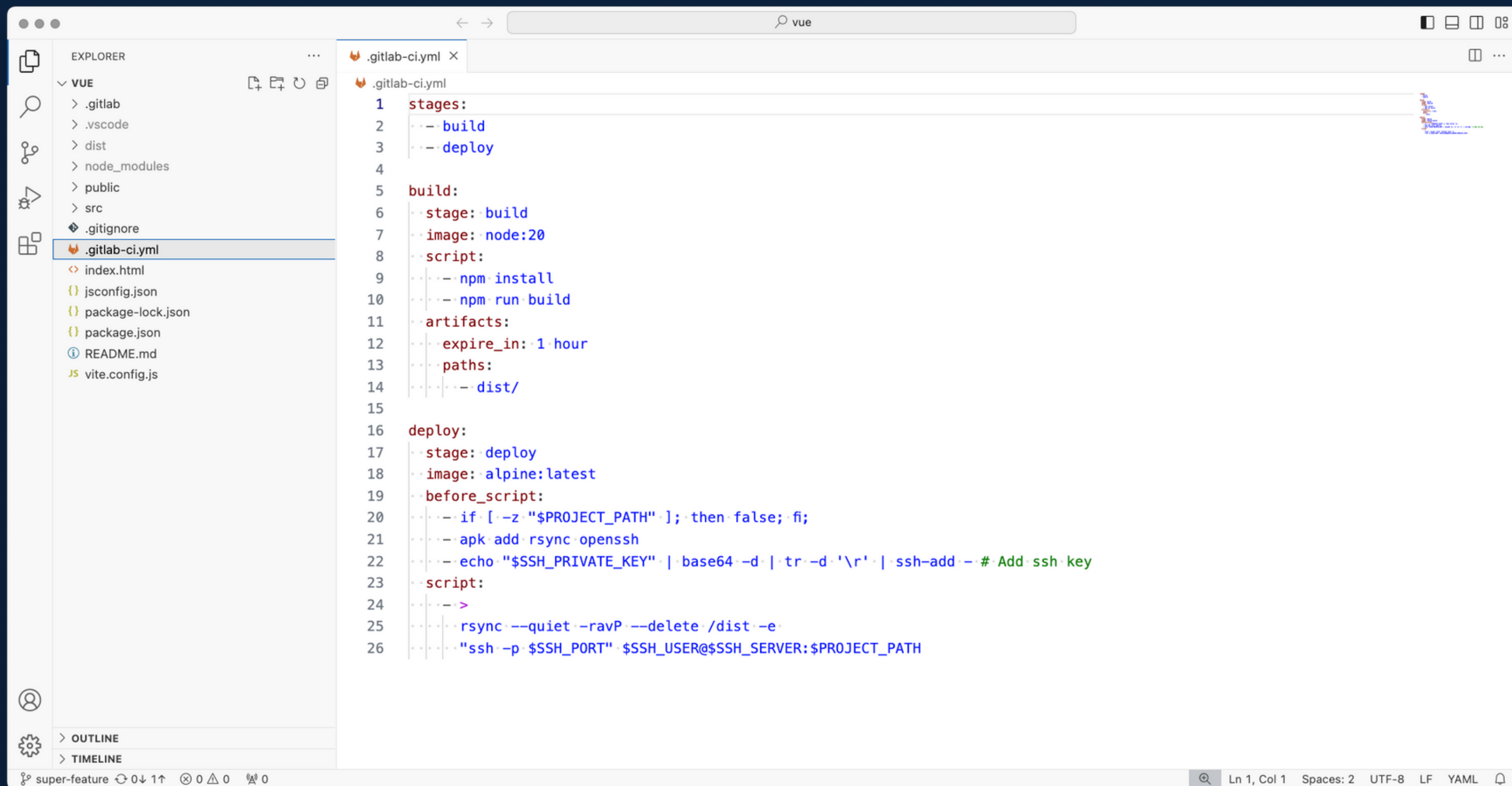
Definicja zadań (jobs)

Przepis dla runnera

Dowolne komendy które jesteśmy w stanie wykonać w środowisku uruchomieniowym.



Definicja zadań



The image shows a code editor window with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with files like .gitlab, .vscode, dist, node_modules, public, src, .gitignore, .gitlab-ci.yml, index.html, jsconfig.json, package-lock.json, package.json, README.md, and vite.config.js. The code editor displays the content of .gitlab-ci.yml, which defines two stages: build and deploy. The build stage uses a node:20 image and runs npm install and npm run build. The deploy stage uses an alpine:latest image and runs a script to add an SSH key and rsync files to a remote server.

```
1 stages:
2   -- build
3   -- deploy
4
5 build:
6   stage: build
7   image: node:20
8   script:
9     -- npm install
10    -- npm run build
11  artifacts:
12    expire_in: 1 hour
13    paths:
14      -- dist/
15
16 deploy:
17   stage: deploy
18   image: alpine:latest
19   before_script:
20     -- if [-z "$PROJECT_PATH"]; then false; fi;
21     -- apk add rsync openssh
22     -- echo "$SSH_PRIVATE_KEY" | base64 -d | tr -d '\r' | ssh-add - # Add ssh key
23   script:
24     -- >
25     -- rsync --quiet --ravP --delete /dist -e
26     -- "ssh -p $SSH_PORT $SSH_USER@$SSH_SERVER: $PROJECT_PATH"
```



Pipeline

Zestaw zadań podzielonych na etapy.

Wizualizacja etapów zdefiniowanych zadań.



Build

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

● → vue npm install

added 28 packages, and audited 29 packages in 717ms

4 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
○ → vue █
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

● → vue npm run build

> vue@0.0.0 build
> vite build

vite v5.1.3 building for production...
✓ 23 modules transformed.
dist/index.html           0.43 kB | gzip: 0.29 kB
dist/assets/index-C6LU-vaC.css  3.71 kB | gzip: 1.19 kB
dist/assets/index-Dh21ZQRb.js 53.96 kB | gzip: 21.89 kB
✓ built in 326ms
○ → vue █
```



& Deploy

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

● → vue git:(master) git pull
Updating cade327..1750f39
Fast-forward
 src/App.vue | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
○ → vue git:(master) █
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

● → vue git:(v1.1) git status
HEAD detached at v1.1
nothing to commit, working tree clean
● → vue git:(v1.1) git fetch
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 445 bytes | 111.00 KiB/s, done.
From ssh://_ /k.sobolewski/vue
 bb848d8..467f593  master    -> origin/master
 * [new tag]      v1.2      -> v1.2
● → vue git:(v1.1) git checkout v1.2
Previous HEAD position was 1750f39 feature: add something new to opole.dev
HEAD is now at 467f593 feature: add sth really new to opole.dev
○ → vue git:(v1.2) █
```



Build & deploy

   vue - .gitlab-ci.yml

```
1  stages:  
2    - build  
3    - deploy
```



Build & deploy

```
● ● ● vue - .gitlab-ci.yml  
  
1  build:  
2    stage: build  
3    image: node:20  
4    script:  
5      - npm install  
6      - npm run build
```



Build & deploy

```
● ● ● vue - .gitlab-ci.yml  
1  deploy:  
2    stage: deploy  
3    image: alpine:latest  
4    script:  
5      - >  
6        rsync --quiet -ravP --delete dist/ -e  
7        "ssh -p $SSH_PORT" $SSH_USER@$SSH_SERVER:$PROJECT_PATH
```



Build & deploy

```
vue - .gitlab-ci.yml

1  deploy:
2    stage: deploy
3    image: alpine:latest
4    script:
5      - ssh -tt $SSH_USER@$SSH_SERVER -p $SSH_PORT
6        "
7        cd $PROJECT_PATH &&
8        yarn encore dev
9        "
```



Build & deploy

```
vue - .gitlab-ci.yml  
  
1  deploy:  
2    stage: deploy  
3    image: alpine:latest  
4    script:  
5      - ssh -tt $SSH_USER@$SSH_SERVER -p $SSH_PORT  
6        "  
7        cd $PROJECT_PATH &&  
8        git pull -f --quiet  
9        "
```



Build & deploy

```
vue - .gitlab-ci.yml

1  stages:
2    - build
3    - deploy
4
5  build:
6    stage: build
7    image: node:20
8    script:
9      - npm install
10     - npm run build
11
12  deploy:
13    stage: deploy
14    image: alpine:latest
15    script:
16      - >
17        rsync --quiet -ravP --delete dist/ -e
18        "ssh -p $SSH_PORT" $SSH_USER@$SSH_SERVER:$PROJECT_PATH
```



Efekt




Efekt



Dlaczego?

deploy

✖ Failed Started 1 day ago by  Krystian Sobolewski



```
1 Running with gitlab-runner 16.8.0 (c72a09b6)
2   on codelabs-gitlab-runner ██████████
> 3 Preparing the "docker" executor 00:02
> 7 Preparing environment 00:01
> 9 Getting source from Git repository 00:01
> 16 Downloading artifacts 00:01
✓ 19 Executing "step_script" stage of the job script 00:01
20 Using docker image sha256:05455a08881ea9cf0e752bc48e61bbd71a34c029bb13df01e40e3e70e0d007bd for alpine:latest with digest alpine@sha256:c5b1261d6d3e4307162693
    1fc004f70149baeba2c8ec672bd4f27761f8e1ad6b ...
21 $ rsync --quiet -ravP -e "ssh -p $SSH_PORT" --delete dist/ $SSH_USER@$SSH_SERVER:$PROJECT_PATH
22 /bin/sh: eval: line 149: rsync: not found
✓ 23 Cleaning up project directory and file based variables 00:01
24 ERROR: Job failed: exit code 127
```



Dlaczego?

```
Executing "step_script" stage of the job script
Using docker image sha256:05455a08881ea9cf0e752bc48e61bb
1fc004f70149baeba2c8ec672bd4f27761f8e1ad6b ...
$ rsync --quiet -ravP -e "ssh -p $SSH_PORT" --delete dist
/bin/sh: eval: line 149: rsync: not found
Cleaning up project directory and file based variables
ERROR: Job failed: exit code 127
```



Poprawki - instalacja narzędzi

```
vue - .gitlab-ci.yml

1  deploy:
2    stage: deploy
3    image: alpine:latest
4    before_script:
5      - apk add rsync openssh
6    script:
7      - >
8        rsync --quiet -ravP --delete /dist -e
9        "ssh -p $SSH_PORT" $SSH_USER@$SSH_SERVER:$PROJECT_PATH
```



Poprawki - instalacja narzędzi

```
vue - .gitlab-ci.yml

1  deploy:
2    stage: deploy
3    image: alpine:latest
4    before_script:
5      - apk add rsync openssh
6    script:
7      - >
8        rsync --quiet -ravP --delete /dist -e
9        "ssh -p $SSH_PORT" $SSH_USER@$SSH_SERVER:$PROJECT_PATH
```



Efekt



Poprawki - zmienne środowiskowe

```
vue - .gitlab-ci.yml

1  deploy:
2    stage: deploy
3    image: alpine:latest
4    variables:
5      SSH_USER: user
6      SSH_SERVER: server
7      SSH_PORT: 22
8      PROJECT_PATH: /var/www/project
9    before_script:
10     - apk add rsync openssh
11    script:
12     - >
13       rsync --quiet -ravP --delete /dist -e
14       "ssh -p $SSH_PORT" $SSH_USER@$SSH_SERVER:$PROJECT_PATH
```



Poprawki - zmienne środowiskowe

```
vue - .gitlab-ci.yml


























1  deploy:
2    stage: deploy
3    image: alpine:latest
4  variables:
5    SSH_USER: user
6    SSH_SERVER: server
7    SSH_PORT: 22
8    PROJECT_PATH: /var/www/project
9  before_script:
10   - apk add rsync openssh
11  script:
12   - >
13     rsync --quiet -ravP --delete /dist -e
14     "ssh -p $SSH_PORT" $SSH_USER@$SSH_SERVER:$PROJECT_PATH
```



Efekt



Poprawki - zmienne środowiskowe

| CI/CD Variables </> 5 | | Reveal values | | Add variable |
|---|---|---|---|---|
| ↑ Key | Value | Environments | Actions | |
| PROJECT_PATH  Protected | *****  | All (default)  |  |  |
| SSH_PORT  Protected | *****  | All (default)  |  |  |
| SSH_PRIVATE_KEY  Protected Masked | *****  | All (default)  |  |  |
| SSH_SERVER  Protected Masked | *****  | All (default)  |  |  |
| SSH_USER  Protected Masked | *****  | All (default)  |  |  |



Poprawki - autoryzacja

```
vue - .gitlab-ci.yml

1  deploy:
2    stage: deploy
3    image: alpine:latest
4    before_script:
5      - if [ -z "$PROJECT_PATH" ]; then false; fi;
6      - apk add rsync openssh
7      - echo "$SSH_PRIVATE_KEY" | base64 -d | tr -d '\r' | ssh-add - # Add ssh key
8    script:
9      - >
10     rsync --quiet -ravP --delete /dist -e
11     "ssh -p $SSH_PORT" $SSH_USER@$SSH_SERVER:$PROJECT_PATH
```



Poprawki - autoryzacja i zabezpieczenia

```
vue - .gitlab-ci.yml

1  deploy:
2    stage: deploy
3    image: alpine:latest
4    before_script:
5      - if [ -z "$PROJECT_PATH" ]; then false; fi;
6      - apk add rsync openssh
7      - echo "$SSH_PRIVATE_KEY" | base64 -d | tr -d '\r' | ssh-add - # Add ssh key
8    script:
9      - >
10     rsync --quiet -ravP --delete /dist -e
11     "ssh -p $SSH_PORT" $SSH_USER@$SSH_SERVER:$PROJECT_PATH
```



Efekt



Poprawki - artefakty

```
● ● ● vue - .gitlab-ci.yml

1  build:
2    stage: build
3    image: node:20
4    script:
5      - npm install
6      - npm run build
7    artifacts:
8      expire_in: 1 hour
9      paths:
10     - dist/
```



Poprawki - artefakty

```
● ● ● vue - .gitlab-ci.yml

1  build:
2    stage: build
3    image: node:20
4    script:
5      - npm install
6      - npm run build
7  artifacts:
8    expire_in: 1 hour
9    paths:
10     - dist/
```



Efekt



CI / CD



CI / CD

Continuous Integration / Continuous Deployment



Jak nie zrobić sobie krzywdy

- Środowiska testowe
- Pamiętać o braku interaktywności
- Dbać o szybkość działania i cache



Środowiska testowe

```
vue - .gitlab-ci.yml

1  build:
2    stage: build
3    image: node:20
4    environment:
5      name: ${CI_COMMIT_REF_SLUG}
6    script:
7      - npm install
8      - npm run ${CI_ENVIRONMENT_NAME}
9    artifacts:
10     expire_in: 1 hour
11     paths:
12       - dist/
13   only:
14     - sandbox
```



Optymalizacja

```
vue - .gitlab-ci.yml

1  build:
2    stage: build
3    image: ${CI_DEPENDENCY_PROXY_GROUP_IMAGE_PREFIX}/node:20
4    environment:
5      name: ${CI_COMMIT_REF_SLUG}
6    cache:
7      key: ${CI_ENVIRONMENT_NAME}-node_modules
8      paths:
9        - node_modules/
10   script:
11     - npm install
12     - npm run ${CI_ENVIRONMENT_NAME}
13   artifacts:
14     expire_in: 1 hour
15     paths:
16       - dist/
17   only:
18     - sandbox
```



Co w tym momencie ma John?

CI / CD

- Standaryzacja
- Automatyzacja
- Powtarzalność

Ponadto

- Żywa dokumentacja!
- Możliwości!



Rozdział 3

Możliwości



Code review

Added new feature

[Open](#) Krystian Sobolewski requested to merge `super-feature` into `master` just now

Overview 0 **Commits** - **Pipelines** 0 **Changes** -

Szablon Merge Request

Checklista Code Review

- Zgodność z XXX
- Element do weryfikacji
- Kolejny punkt

Checklista testów

- Sprawdzone na środowisku XXX
- Sprawdzone coś jeszcze

0 0

Pipeline #33960 running

Pipeline running for `29c30d03` on `super-feature`

8 **Approve** Approval is optional

Ready to merge!

Delete source branch Squash commits Edit commit message

1 commit and 1 merge commit will be added to master.

Merge...



Testy automatyczne

```
vue - .gitlab-ci.yml

1  stages:
2    - build
3    - tests
4    - deploy
5    ...
6
7  test.unit:
8    stage: tests
9    image: php:8.3-cli
10   script:
11     - php vendor/bin/phpunit tests
```





Uruchamianie manualne

```
vue - .gitlab-ci.yml




1  deploy:
2    stage: deploy
3    image: alpine:latest
4    environment:
5      name: ${CI_COMMIT_REF_SLUG}
6    script:
7      - ...
8
9  deploy.production:
10   extends: [deploy]
11   when: manual
12   only:
13     - tags
```





Bardziej zaawansowany pipeline

✓ Passed Merge branch 'release/v1.2' into rc  

🕒 00:11:18
📅 1 year ago

#13371  rc  4b81ef47 

Pipeline Needs Jobs 6 Tests 0

| build | test | distribute | deploy | e2e | Downstream |
|--------------------------|--------------------------|-------------------------------|---------------------------|--|--|
| ✓ build.dev | ✓ test.unit | ✓ distribute.dev | ✓ deploy.dev | ✓ test.api Trigger job  | ✓ tests #13377 Multi-project  |



Rozdział 4

Perspektywy



Platform as a Service

Opisujemy środowisko uruchomieniowe naszej aplikacji

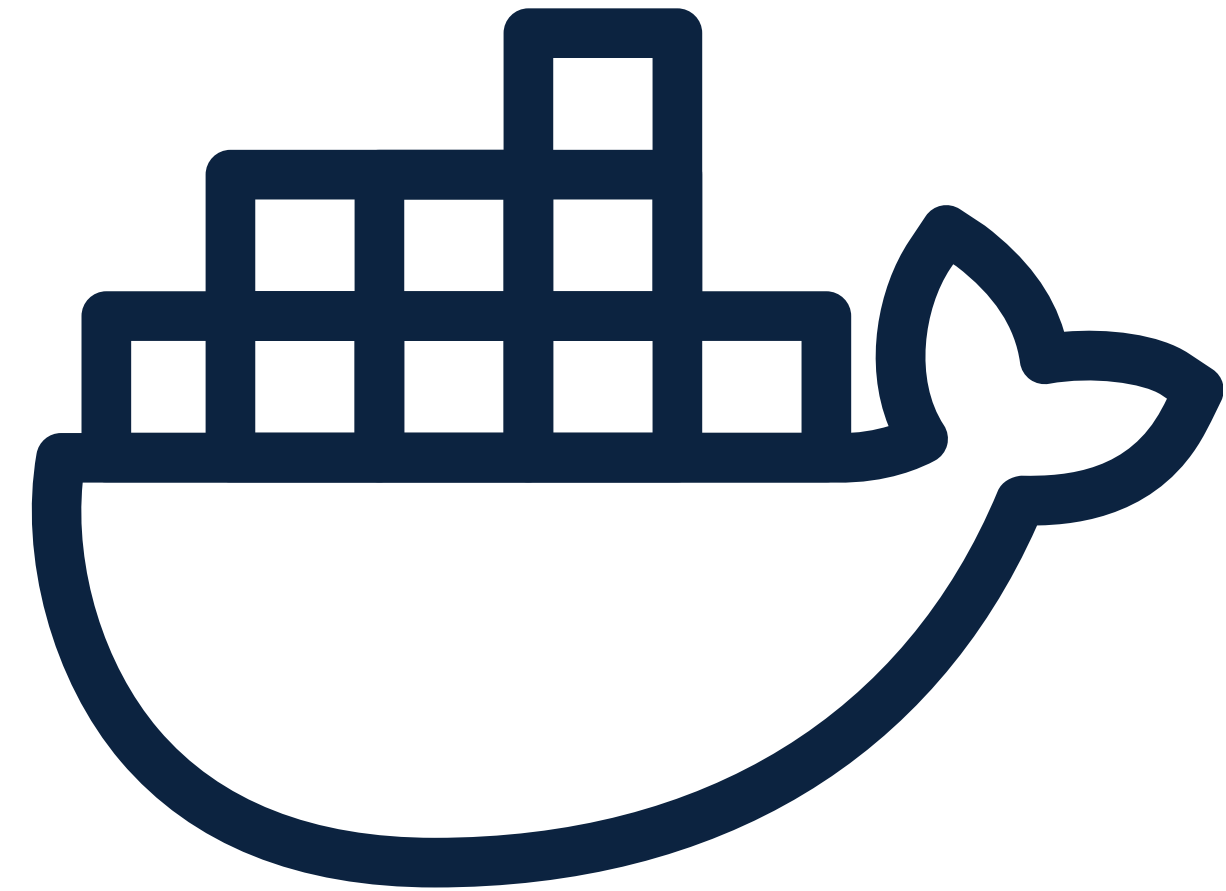
Na przykład

- DigitalOcean App Platform
- Heroku Platform



Konteneryzacja

- “U mnie działa”
- ”Ale Twojego komputera nie damy użytkownikowi”



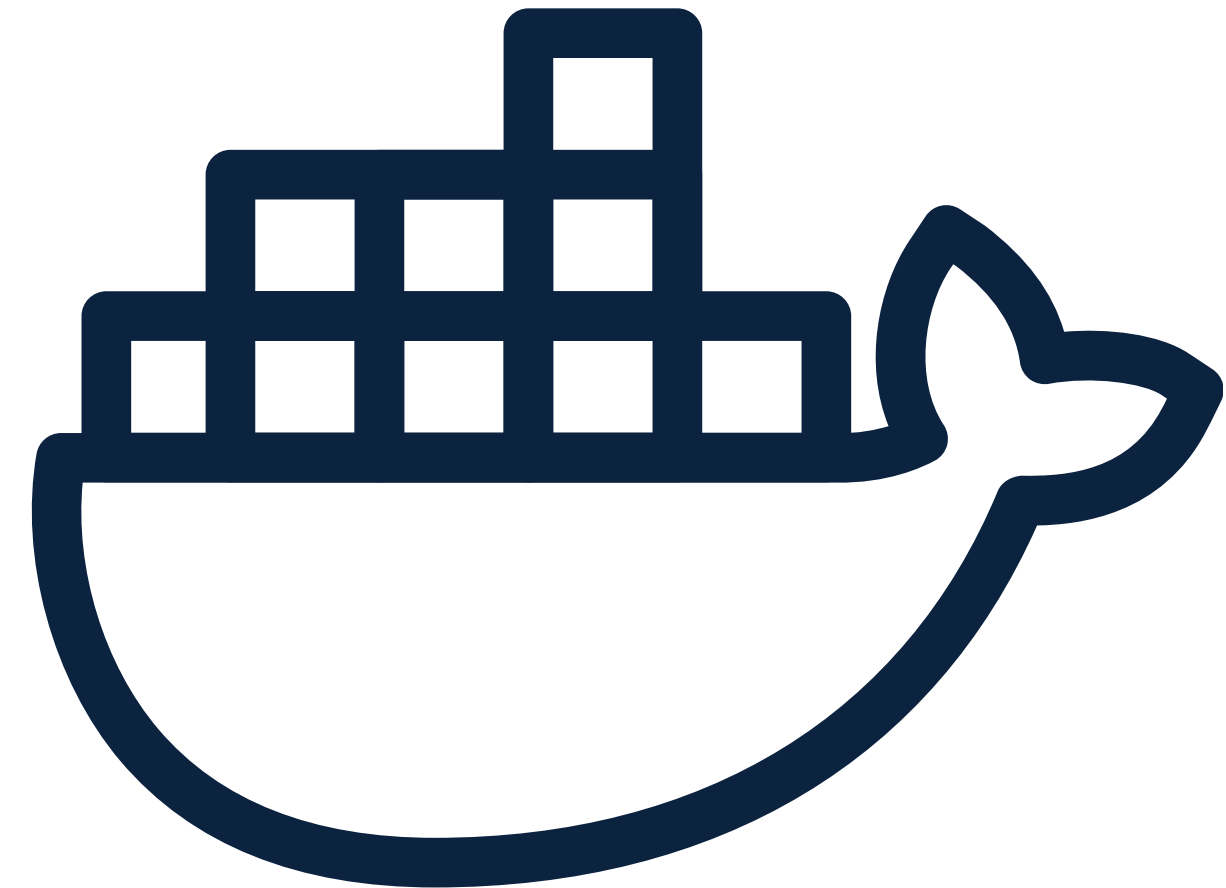
Konteneryzacja

- “U mnie działa”
- ”Ale Twojego komputera nie damy użytkownikowi”

Opisujemy środowisko uruchomieniowe naszej aplikacji

Na przykład

- Docker



Konteneryzacja

```
vue - .gitlab-ci.yml

1  distribute:
2    stage: build
3    image: docker
4    services:
5      - docker:dind
6    environment:
7      name: ${CI_COMMIT_REF_SLUG}
8    variables:
9      REGISTRY_IMAGE: ${CI_REGISTRY_IMAGE}/${CI_ENVIRONMENT_NAME}
10     REGISTRY_IMAGE_LATEST: ${CI_REGISTRY_IMAGE}/${CI_ENVIRONMENT_NAME}:latest
11    script:
12      - docker login -u ${CI_REGISTRY_USER} -p ${CI_REGISTRY_PASSWORD} ${CI_REGISTRY}
13      - docker build --pull -t ${REGISTRY_IMAGE} .
14      - docker tag ${REGISTRY_IMAGE} ${REGISTRY_IMAGE_LATEST}
15      - docker push ${REGISTRY_IMAGE}
16      - docker push ${REGISTRY_IMAGE_LATEST}
```



Konteneryzacja

```
vue - .gitlab-ci.yml

1  distribute:
2    stage: build
3    image: docker
4    services:
5      - docker:dind
6    environment:
7      name: ${CI_COMMIT_REF_SLUG}
8    variables:
9      REGISTRY_IMAGE: ${CI_REGISTRY_IMAGE}/${CI_ENVIRONMENT_NAME}
10     REGISTRY_IMAGE_LATEST: ${CI_REGISTRY_IMAGE}/${CI_ENVIRONMENT_NAME}:latest
11    script:
12      - docker login -u ${CI_REGISTRY_USER} -p ${CI_REGISTRY_PASSWORD} ${CI_REGISTRY}
13      - docker build --pull -t ${REGISTRY_IMAGE} .
14      - docker tag ${REGISTRY_IMAGE} ${REGISTRY_IMAGE_LATEST}
15      - docker push ${REGISTRY_IMAGE}
16      - docker push ${REGISTRY_IMAGE_LATEST}
```



Container as a Service

Uruchamiamy przygotowane kontenery

Na przykład

- AWS ECS
- Google Cloud Platform
- Azure Containers



Container as a Service

```
vue - .gitlab-ci.yml

1  deploy:
2    stage: deploy
3    image: aws-base:latest
4    environment:
5      name: ${CI_COMMIT_REF_SLUG}
6    variables:
7      REGISTRY_IMAGE: ${CI_REGISTRY_IMAGE}/${CI_ENVIRONMENT_NAME}
8    before_script:
9      - docker login -u ${CI_REGISTRY_USER} -p ${CI_REGISTRY_PASSWORD} ${CI_REGISTRY}
10   script:
11     - docker pull ${REGISTRY_IMAGE}
12     - aws ecr get-login-password --region ${AWS_DEFAULT_REGION} | docker login --username AWS --password-stdin ${AWS_ECR_REGISTRY}
13     - docker tag ${REGISTRY_IMAGE} ${AWS_ECR_REGISTRY}/${AWS_ECR_IMAGE}:${CI_COMMIT_SHORT_SHA}
14     - docker push ${AWS_ECR_REGISTRY}/${AWS_ECR_IMAGE}:${CI_COMMIT_SHORT_SHA}
15     - docker tag ${REGISTRY_IMAGE} ${AWS_ECR_REGISTRY}/${AWS_ECR_IMAGE}:latest
16     - docker push ${AWS_ECR_REGISTRY}/${AWS_ECR_IMAGE}:latest
17     - aws ecs update-service --cluster ${AWS_CLUSTER} --service ${AWS_SERVICE} --force-new-deployment
```



Container as a Service

```
vue - .gitlab-ci.yml

1  deploy:
2    stage: deploy
3    image: aws-base:latest
4    environment:
5      name: ${CI_COMMIT_REF_SLUG}
6    variables:
7      REGISTRY_IMAGE: ${CI_REGISTRY_IMAGE}/${CI_ENVIRONMENT_NAME}
8    before_script:
9      - docker login -u ${CI_REGISTRY_USER} -p ${CI_REGISTRY_PASSWORD} ${CI_REGISTRY}
10   script:
11     - docker pull ${REGISTRY_IMAGE}
12     - aws ecr get-login-password --region ${AWS_DEFAULT_REGION} | docker login --username AWS --password-stdin ${AWS_ECR_REGISTRY}
13     - docker tag ${REGISTRY_IMAGE} ${AWS_ECR_REGISTRY}/${AWS_ECR_IMAGE}:${CI_COMMIT_SHORT_SHA}
14     - docker push ${AWS_ECR_REGISTRY}/${AWS_ECR_IMAGE}:${CI_COMMIT_SHORT_SHA}
15     - docker tag ${REGISTRY_IMAGE} ${AWS_ECR_REGISTRY}/${AWS_ECR_IMAGE}:latest
16     - docker push ${AWS_ECR_REGISTRY}/${AWS_ECR_IMAGE}:latest
17     - aws ecs update-service --cluster ${AWS_CLUSTER} --service ${AWS_SERVICE} --force-new-deployment
```



Orkiestracja

Opisujemy relacje między serwisami oraz sposób ich wdrożenia

Na przykład

- Kubernetes
- Docker Swarm



Orkiestracja

```
vue - .gitlab-ci.yml

1  deploy:
2    stage: deploy
3    image: docker
4    environment:
5      name: ${CI_COMMIT_REF_SLUG}
6    variables:
7      REGISTRY_IMAGE: ${CI_REGISTRY_IMAGE}/${CI_ENVIRONMENT_NAME}
8      DOCKER_HOST: ${REMOTE_HOST}
9    before_script:
10     ...
11    script:
12     - docker pull ${REGISTRY_IMAGE}
13     - docker stack deploy --with-registry-auth -c docker-compose.yml -c docker-compose.prod.yml ${COMPOSE_PROJECT_NAME}
```



Orkiestracja

```
vue - .gitlab-ci.yml

1  deploy:
2    stage: deploy
3    image: docker
4    environment:
5      name: ${CI_COMMIT_REF_SLUG}
6    variables:
7      REGISTRY_IMAGE: ${CI_REGISTRY_IMAGE}/${CI_ENVIRONMENT_NAME}
8      DOCKER_HOST: ${REMOTE_HOST}
9    before_script:
10     ...
11   script:
12     - docker pull ${REGISTRY_IMAGE}
13     - docker stack deploy --with-registry-auth -c docker-compose.yml -c docker-compose.prod.yml ${COMPOSE_PROJECT_NAME}
```



Monitoring

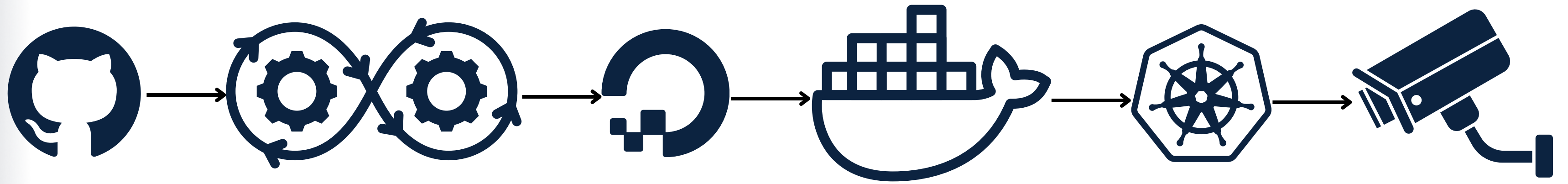
Obserwujemy naszą aplikację, jesteśmy informowani o incydentach, wyciągamy wnioski

Na przykład

- Sentry
- Prometheus
- Grafana
- ELK (Elasticsearch, Logstash, and Kibana)



Kroki



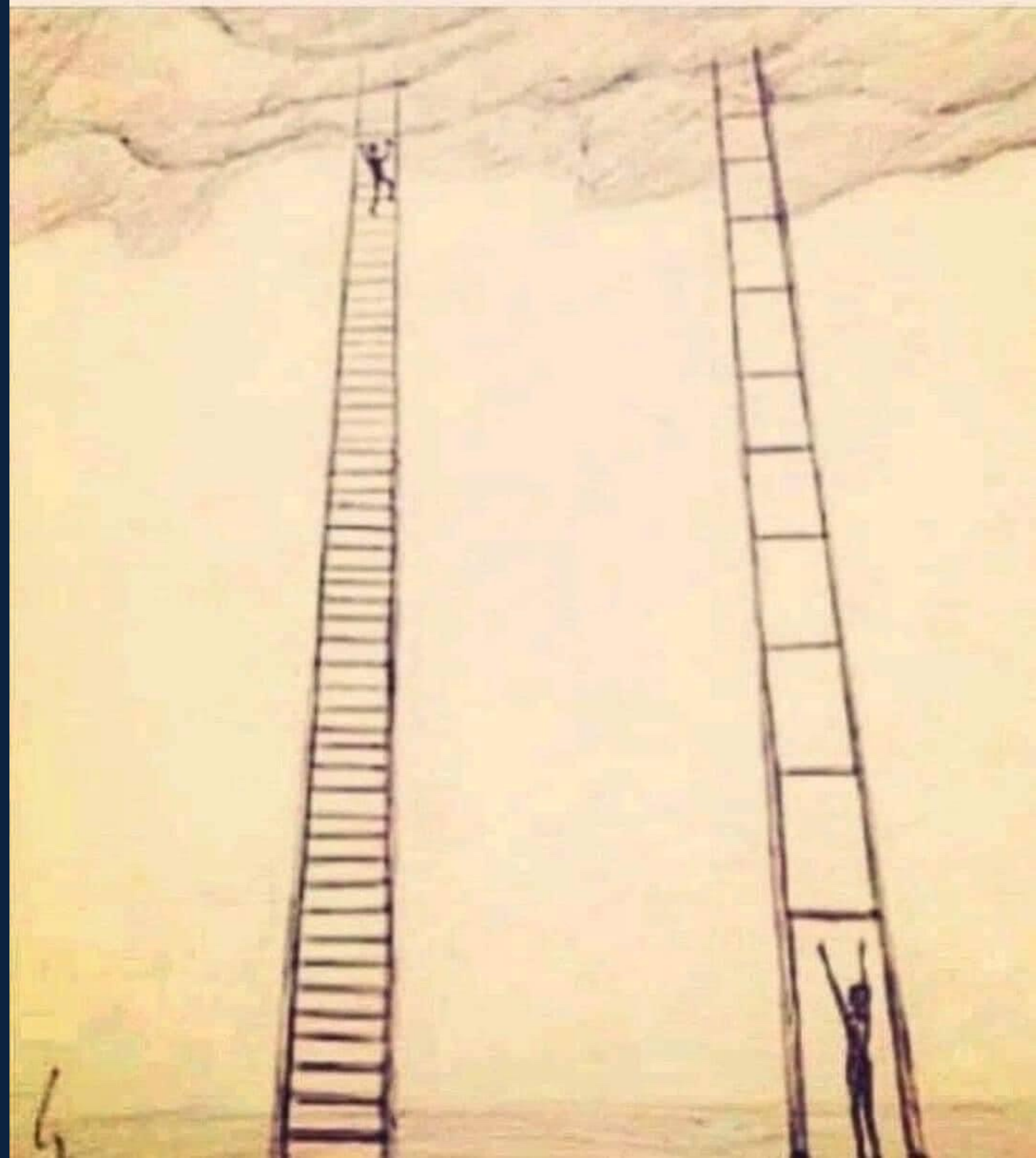
W jaki sposób w takim razie
robić to “bez bólu”?



W jaki sposób robić to “bez bólu”?

- Małymi krokami

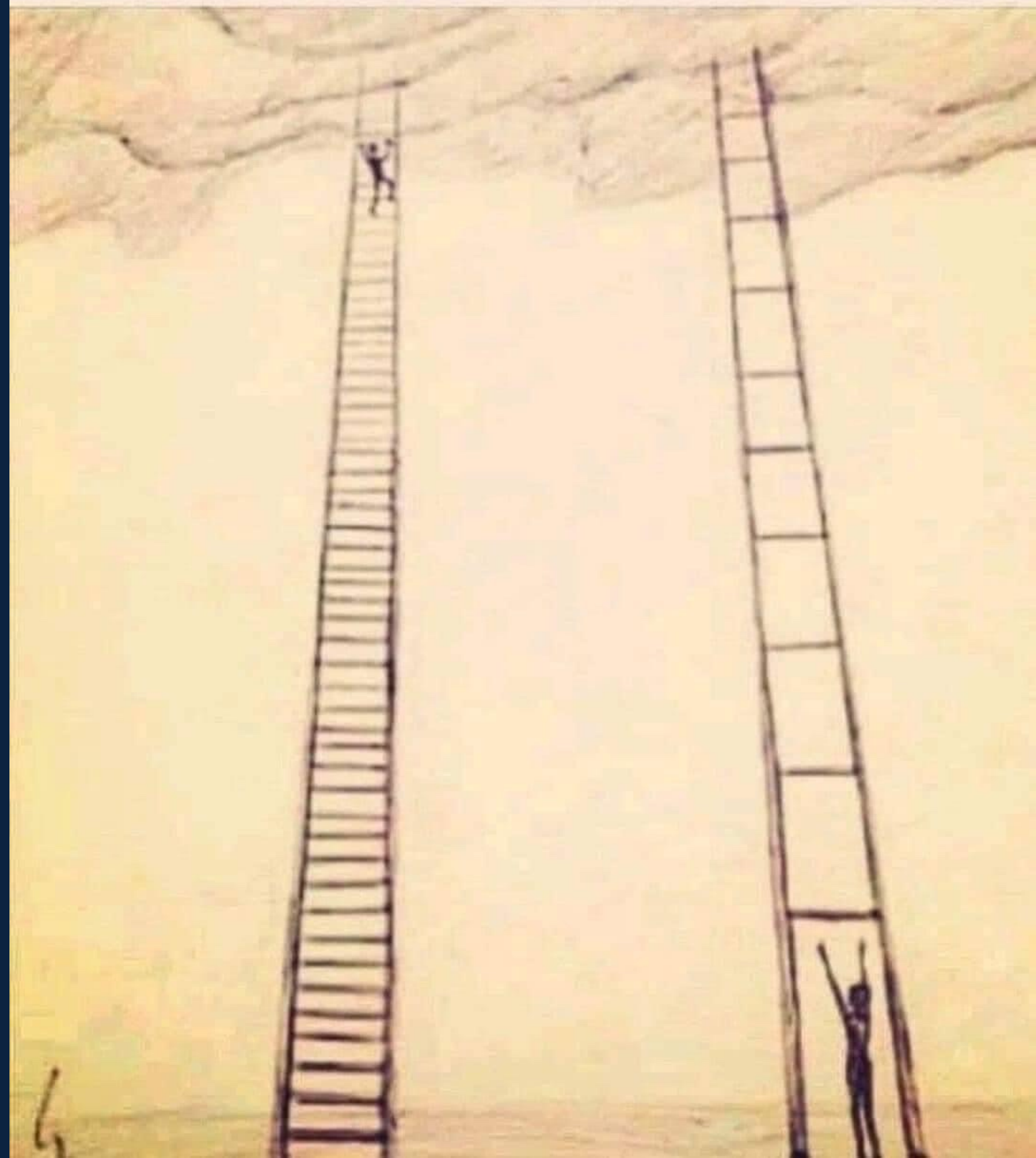
The Importance Of Small Steps!!!



W jaki sposób robić to “bez bólu”?

- Małymi krokami
- Zgodnie ze swoimi potrzebami

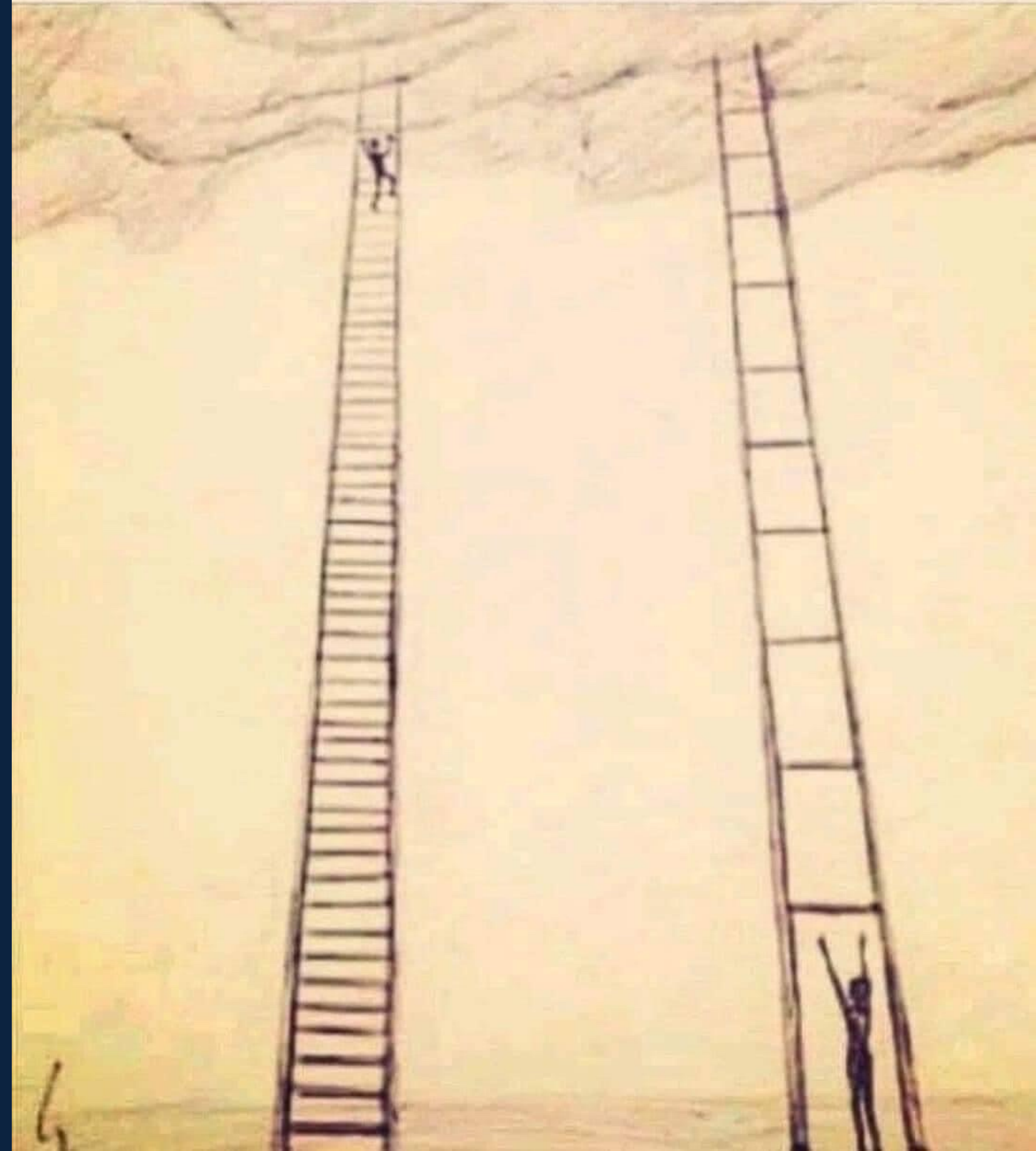
The Importance Of Small Steps!!!



W jaki sposób robić to “bez bólu”?

- Małymi krokami
- Zgodnie ze swoimi potrzebami
- Wykorzystując obecnie używane procesy
- Wykorzystując obecnie używane narzędzia

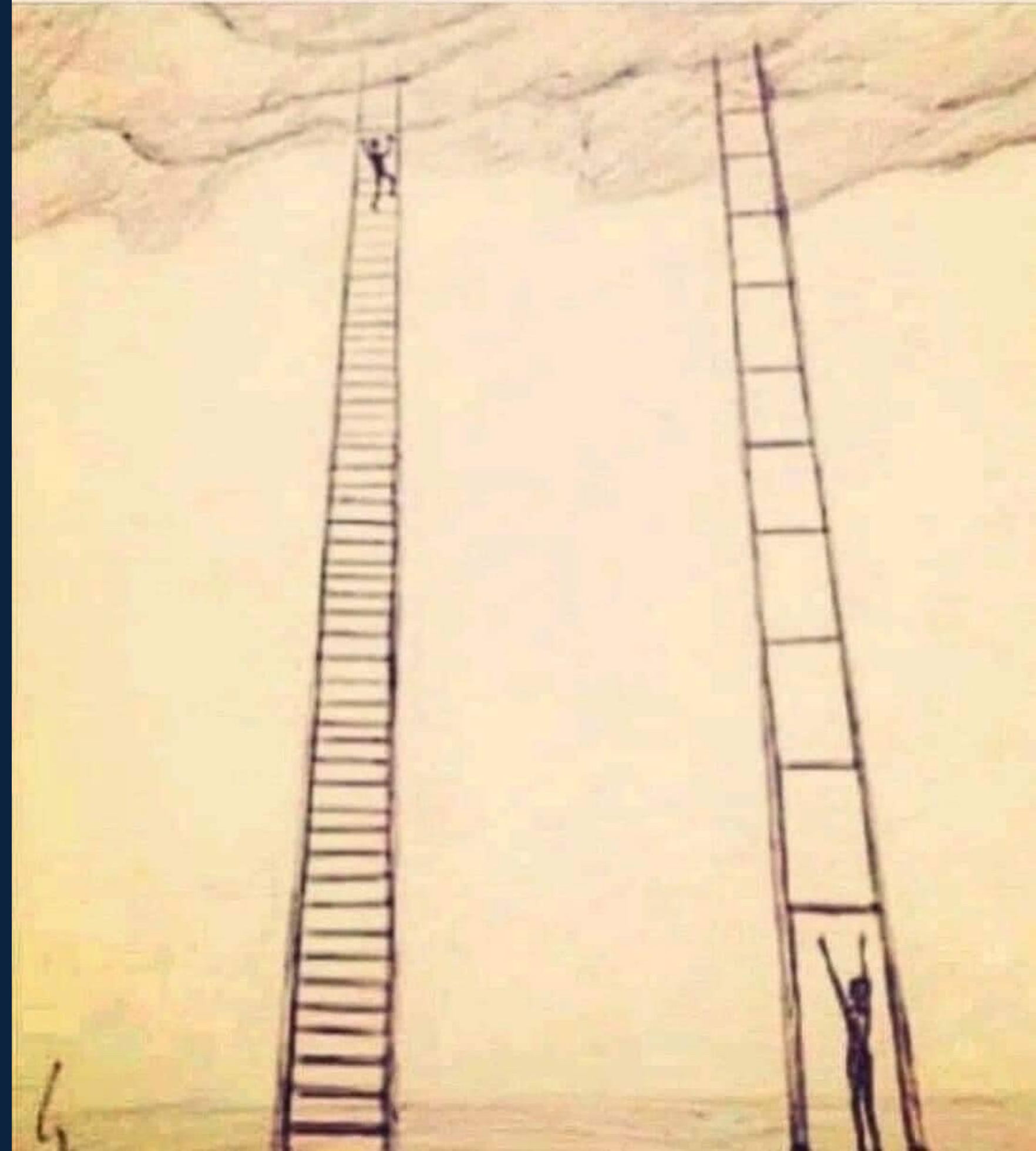
The Importance Of Small Steps!!!



W jaki sposób robić to “bez bólu”?

- Małymi krokami
- Zgodnie ze swoimi potrzebami
- Wykorzystując obecnie używane procesy
- Wykorzystując obecnie używane narzędzia
- Korzystając ze środowisk testowych

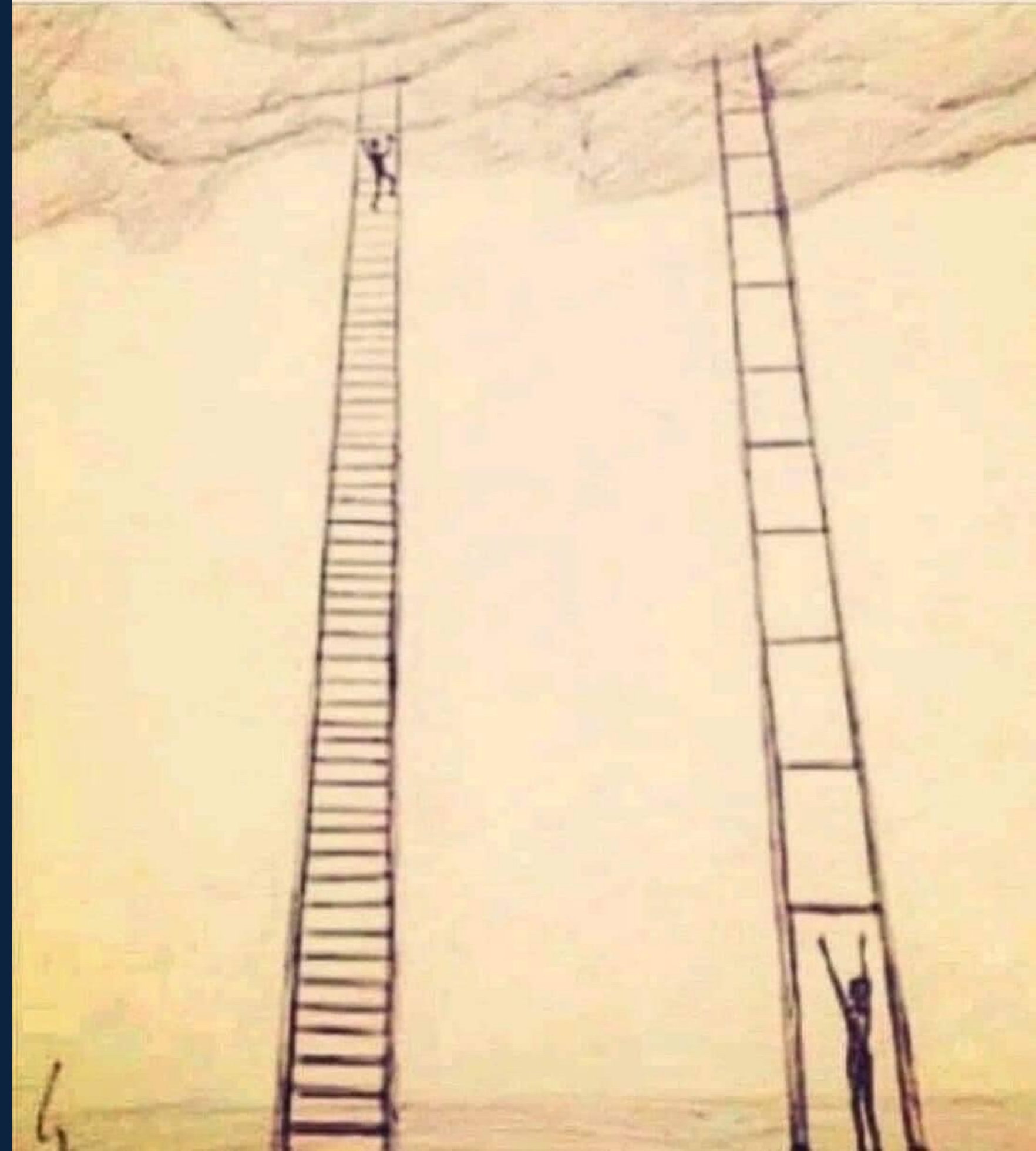
The Importance Of Small Steps!!!



W jaki sposób robić to “bez bólu”?

- Małymi krokami
- Zgodnie ze swoimi potrzebami
- Wykorzystując obecnie używane procesy
- Wykorzystując obecnie używane narzędzia
- Korzystając ze środowisk testowych
- Jak najszybciej

The Importance Of Small Steps!!!



Każdy projekt zasługuje na
odrobinę devops.



Dziękuję

Koniec





Q&A