we ♥ speed

WEB PERFORMANCE TESTING

A Seamless Integration into Web Development Pipelines

Nantes, 2024

HELLO!

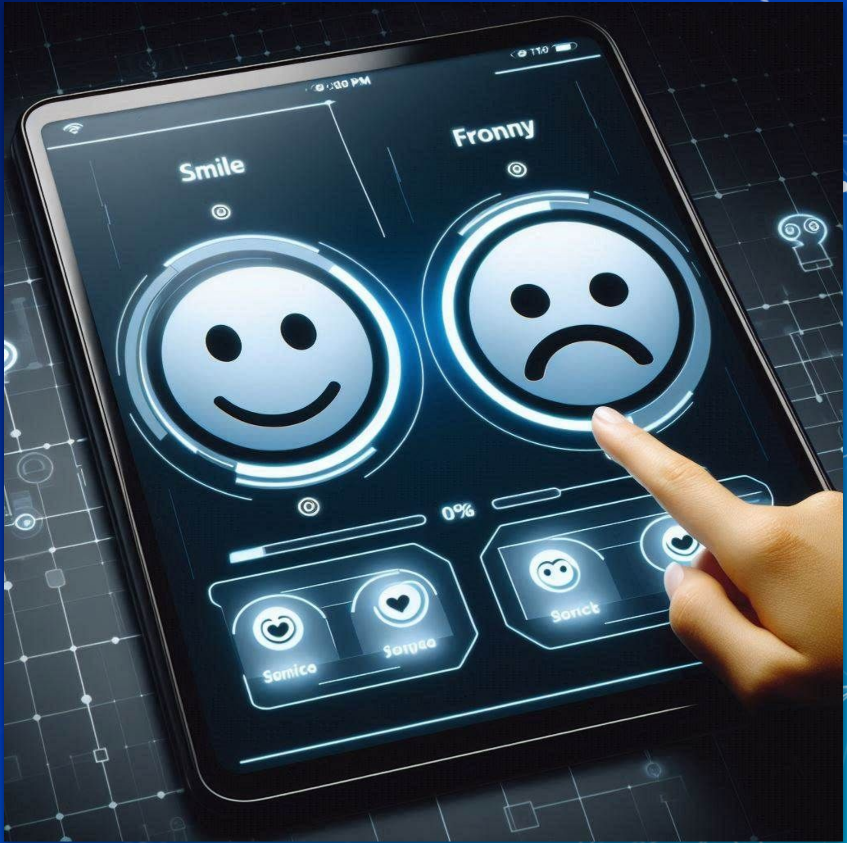
I am Estela Franco

Web Performance Specialist at
Schneider Electric

DISCLAIMER

!=

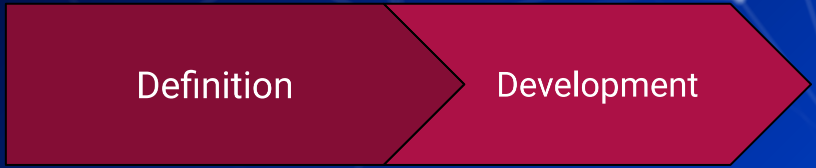
The problem



The problem

Definition

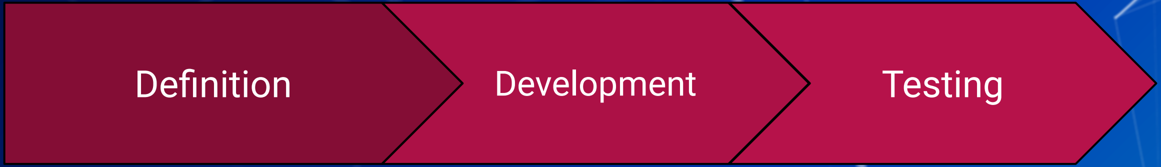
The problem



Definition

Development

The problem

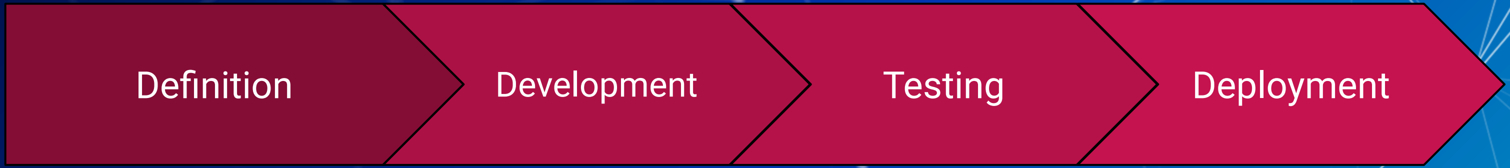


Definition

Development

Testing

The problem



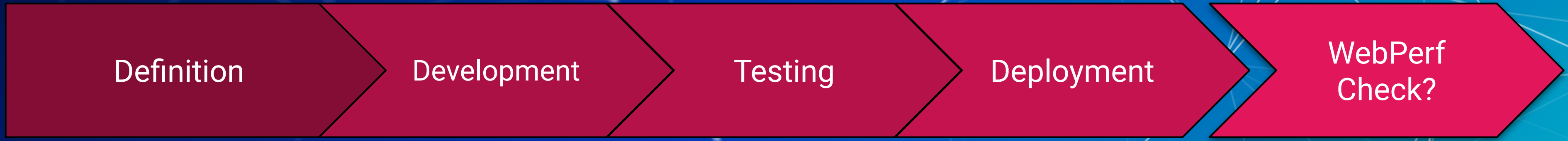
Definition

Development

Testing

Deployment

The problem



Definition

Development

Testing

Deployment

WebPerf Check?

WebPerf
Check?

Devs?

QA?

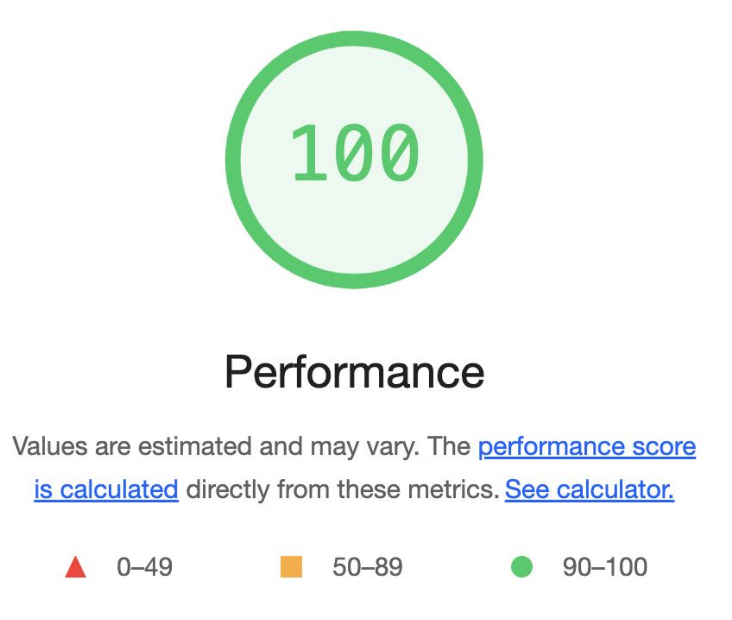
SEO?

Product?

Users?

No one?? 😢

Once upon a time...

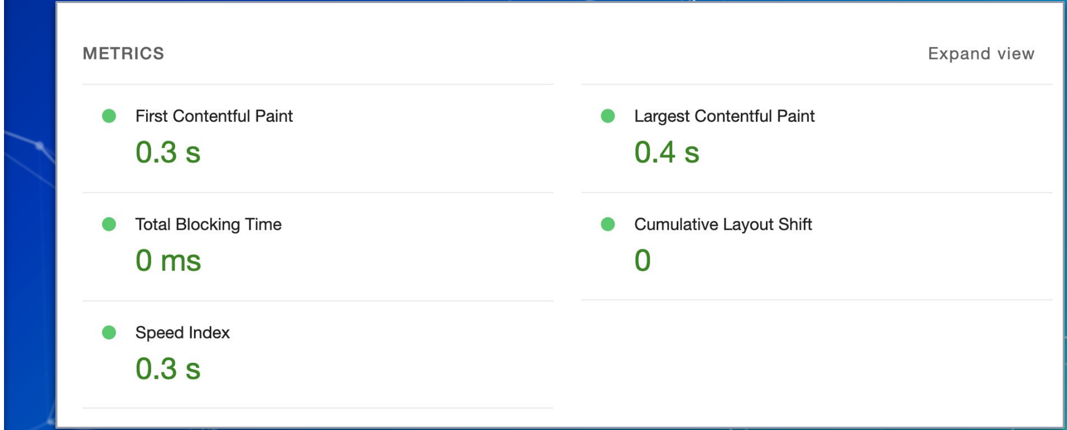


100

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100



METRICS

Expand view

● First Contentful Paint
0.3 s

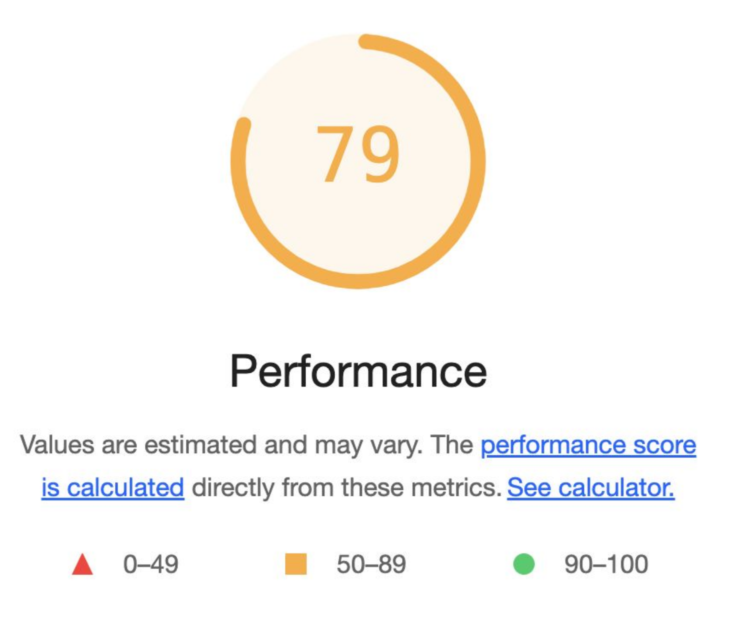
● Largest Contentful Paint
0.4 s

● Total Blocking Time
0 ms

● Cumulative Layout Shift
0

● Speed Index
0.3 s

But the following day...

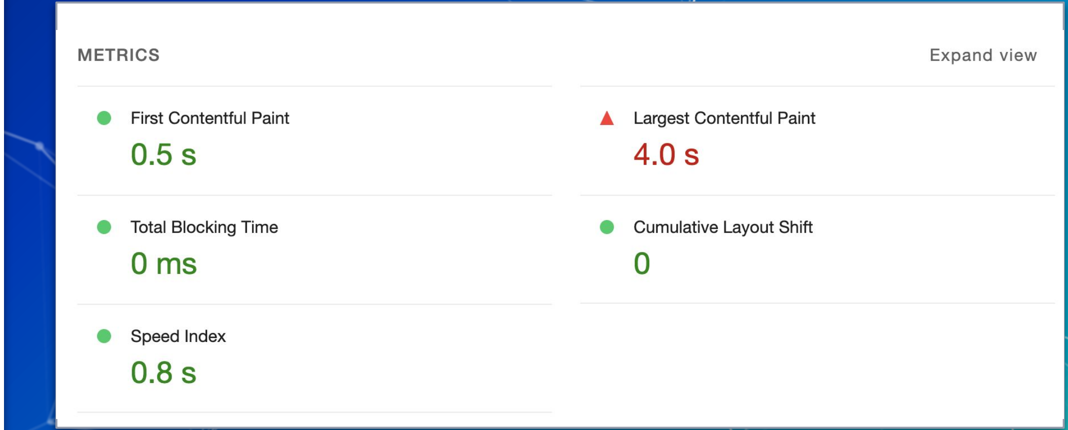


79

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100



METRICS

Expand view

● First Contentful Paint
0.5 s

▲ Largest Contentful Paint
4.0 s

● Total Blocking Time
0 ms

● Cumulative Layout Shift
0

● Speed Index
0.8 s

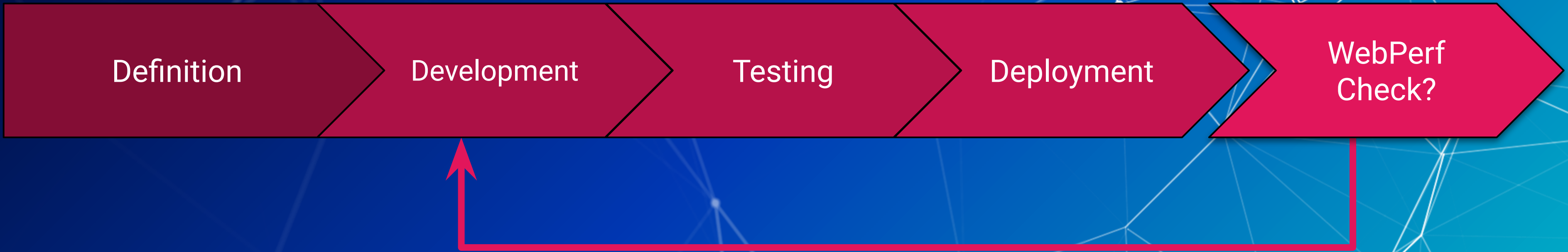
New hero image

~100kB → 7.2MB

New hero image

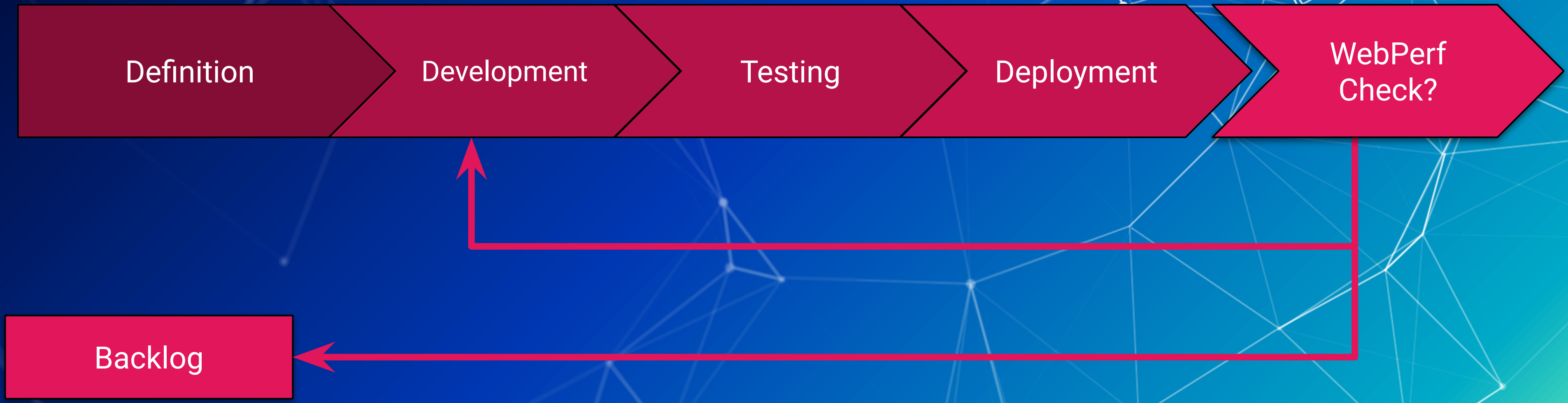
~100kB → ~~7.2MB~~
~90kB

The problem

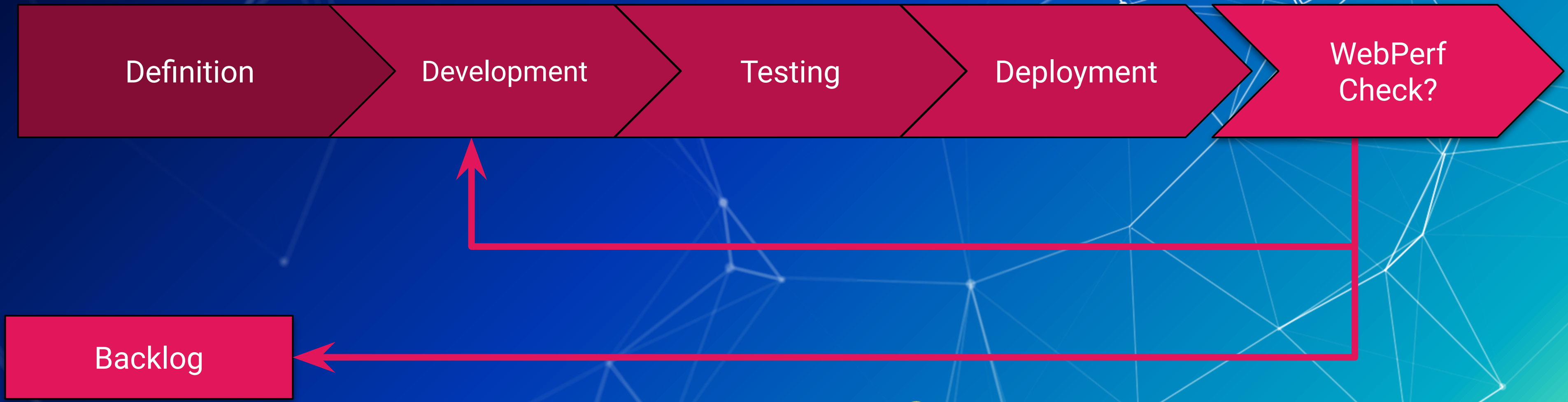


we ❤ speed

The problem


Definition
Development
Testing
Deployment
WebPerf Check?
Backlog

The problem



Definition

Development

Testing

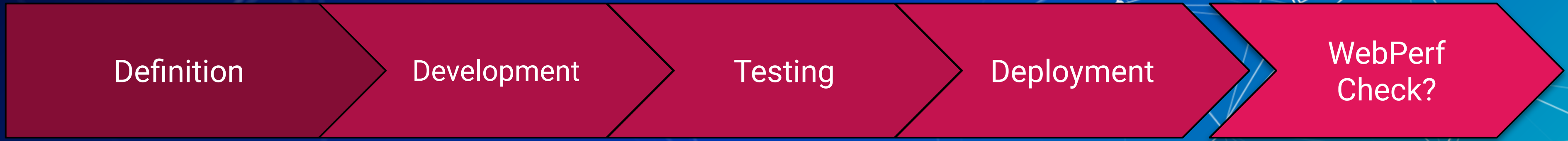
Deployment

WebPerf Check?

Backlog

⏳ 💰 😟☹️

The solution



Definition

Development

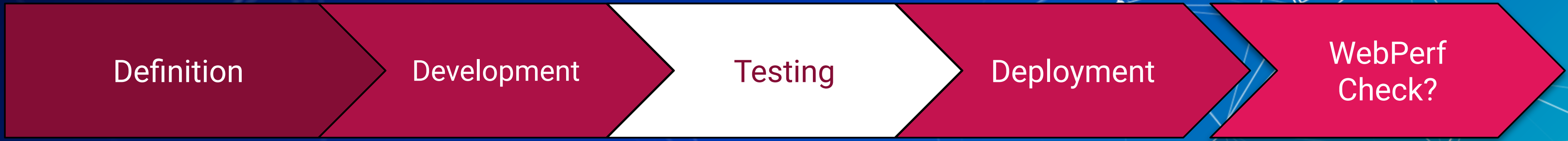
Testing

Deployment

WebPerf Check?

we speed

The solution



Definition

Development

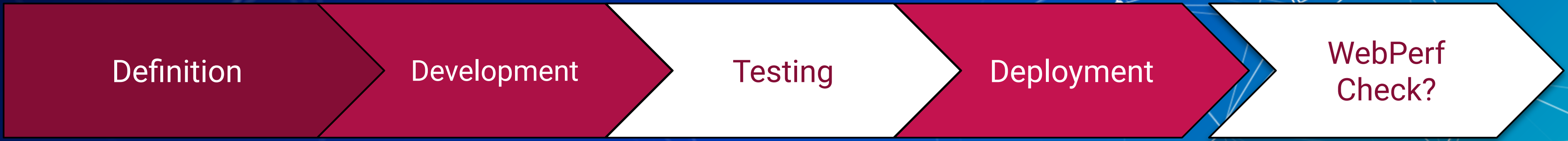
Testing

Deployment

WebPerf Check?

[...testing, "webPerf"]

The solution



[...testing, "webPerf"]

> "Don't worry, Estela! We run manual Lighthouse tests during the development process in our local machines.

we ♥ speed

Pros

- Helps to identify some of the web performance issues before the code is shipped.
- Improves the web performance culture.

Cons

- Manual work: how many runs per URL? How many URLs tested?
- Local environment **!=** standard environment

Pros

- Helps to identify some of the web performance issues before the code is shipped.
- Improves the web performance culture.

Cons

- Manual work: how many runs per URL? How many URLs tested?
- Local environment **!=** standard environme

It worked on my machine
¯_(ツ)_/¯

How to do this?

CONTINUOUS INTEGRATION (CI)

But first...

What is Git?

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

But first...

What is GitHub?

GitHub is a Git repository hosting service. It offers the distributed version control and source code management functionality of Git, plus its own features such as bug tracking, feature requests, task management, continuous integration, and wikis for every project.

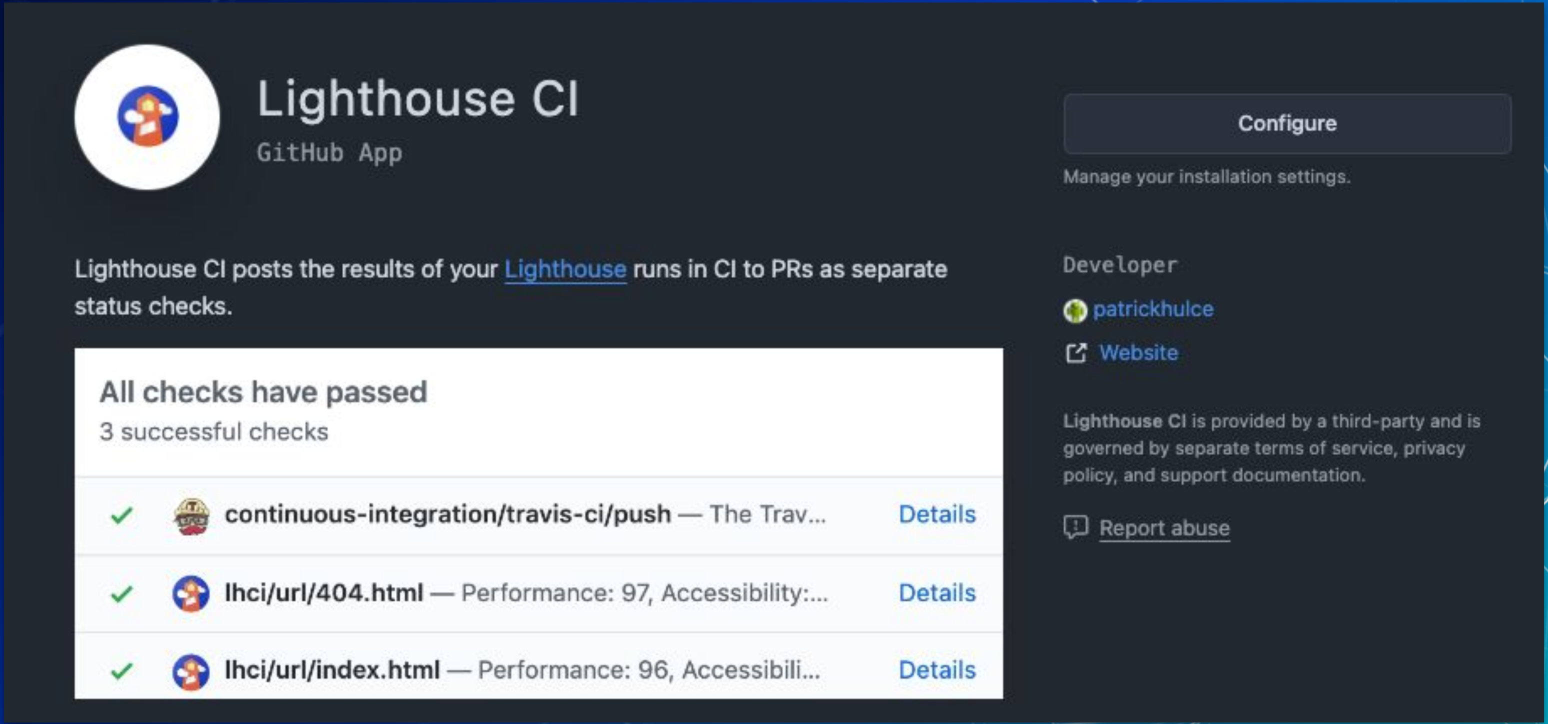
But first…

What is GitHub Actions?

GitHub Actions is a tool built into GitHub that helps automate tasks in software projects. It allows you to set up workflows that can automatically run when certain events happen, like when new code is added to a project.

LIGHTHOUSE CI

Lighthouse CI is a suite of tools that make continuously running, saving, retrieving, and asserting against Lighthouse results as easy as possible.



Lighthouse CI

GitHub App

Configure

Manage your installation settings.

Lighthouse CI posts the results of your Lighthouse runs in CI to PRs as separate status checks.

All checks have passed

3 successful checks

✓ continuous-integration/travis-ci/push — The Trav... Details

✓ lhci/url/404.html — Performance: 97, Accessibility:... Details

✓ lhci/url/index.html — Performance: 96, Accessibili... Details

Developer

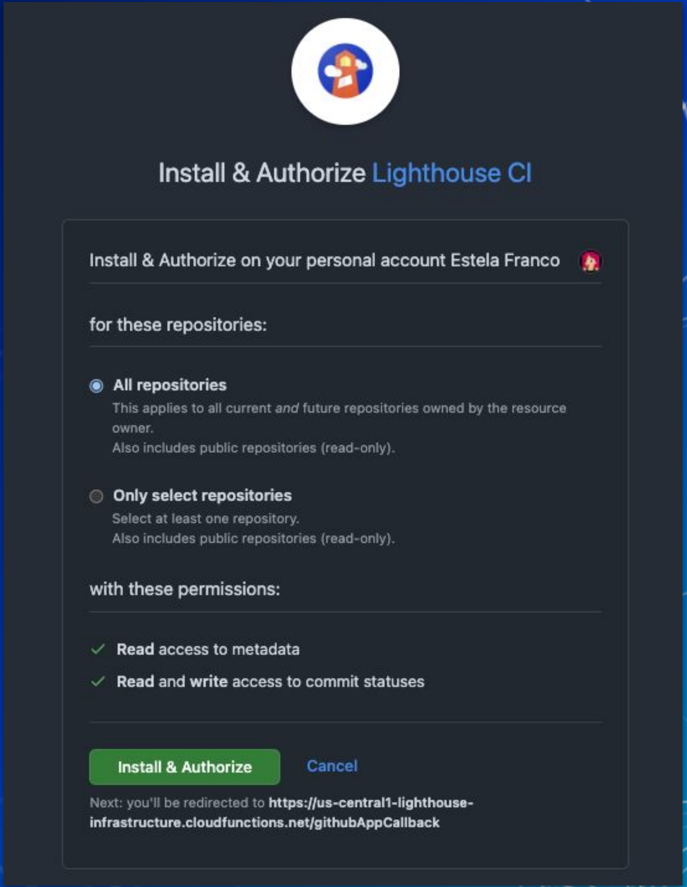
patrickhulce

Website

Lighthouse CI is provided by a third-party and is governed by separate terms of service, privacy policy, and support documentation.

Report abuse

https://github.com/apps/lighthouse-ci



Install & Authorize Lighthouse CI

Install & Authorize on your personal account Estela Franco

for these repositories:

● **All repositories**
This applies to all current *and* future repositories owned by the resource owner.
Also includes public repositories (read-only).

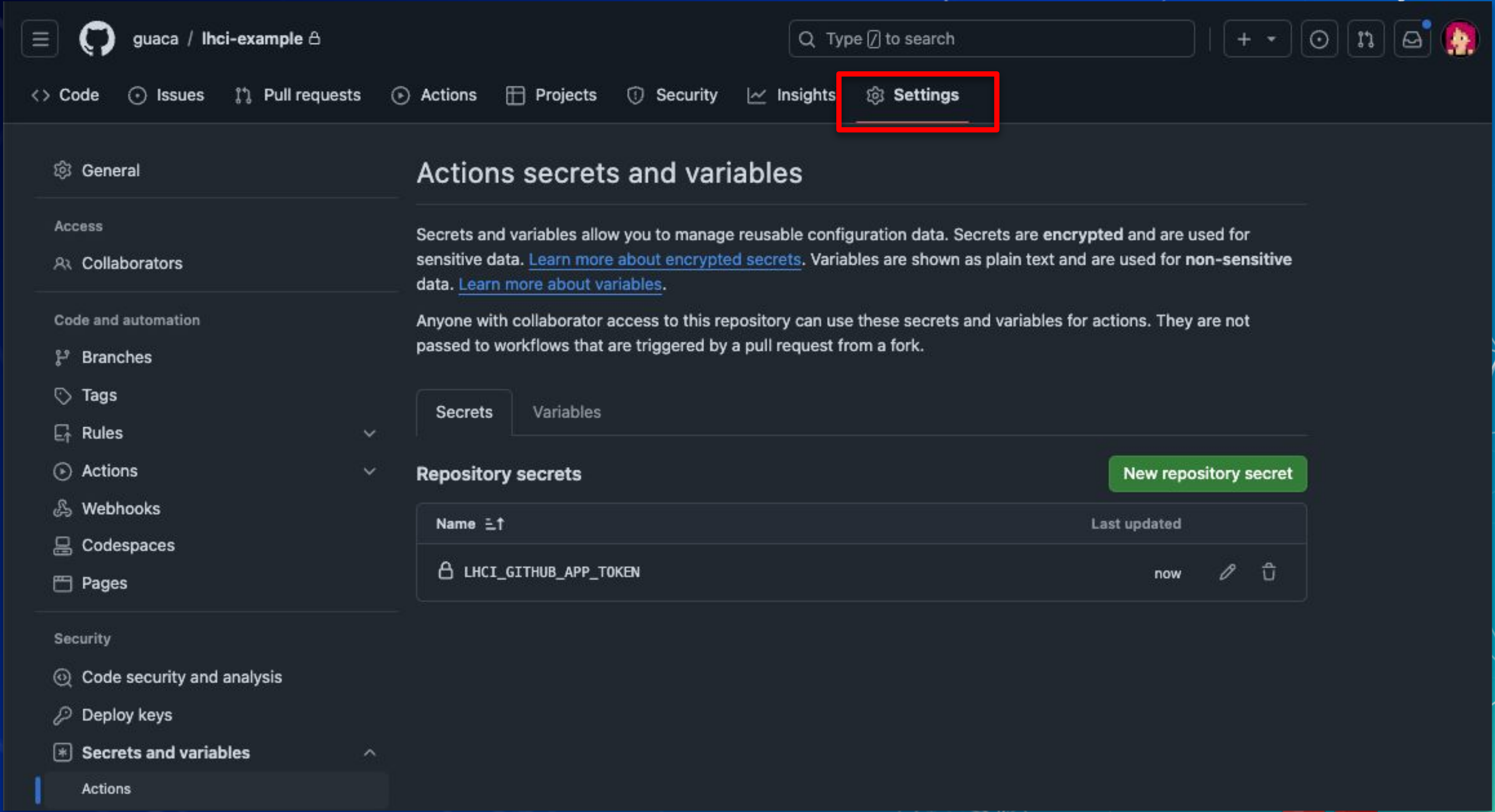
○ **Only select repositories**
Select at least one repository.
Also includes public repositories (read-only).

with these permissions:

✓ **Read** access to metadata

✓ **Read** and **write** access to commit statuses

Next: you'll be redirected to **https://us-central1-lighthouse-infrastructure.cloudfunctions.net/githubAppCallback**



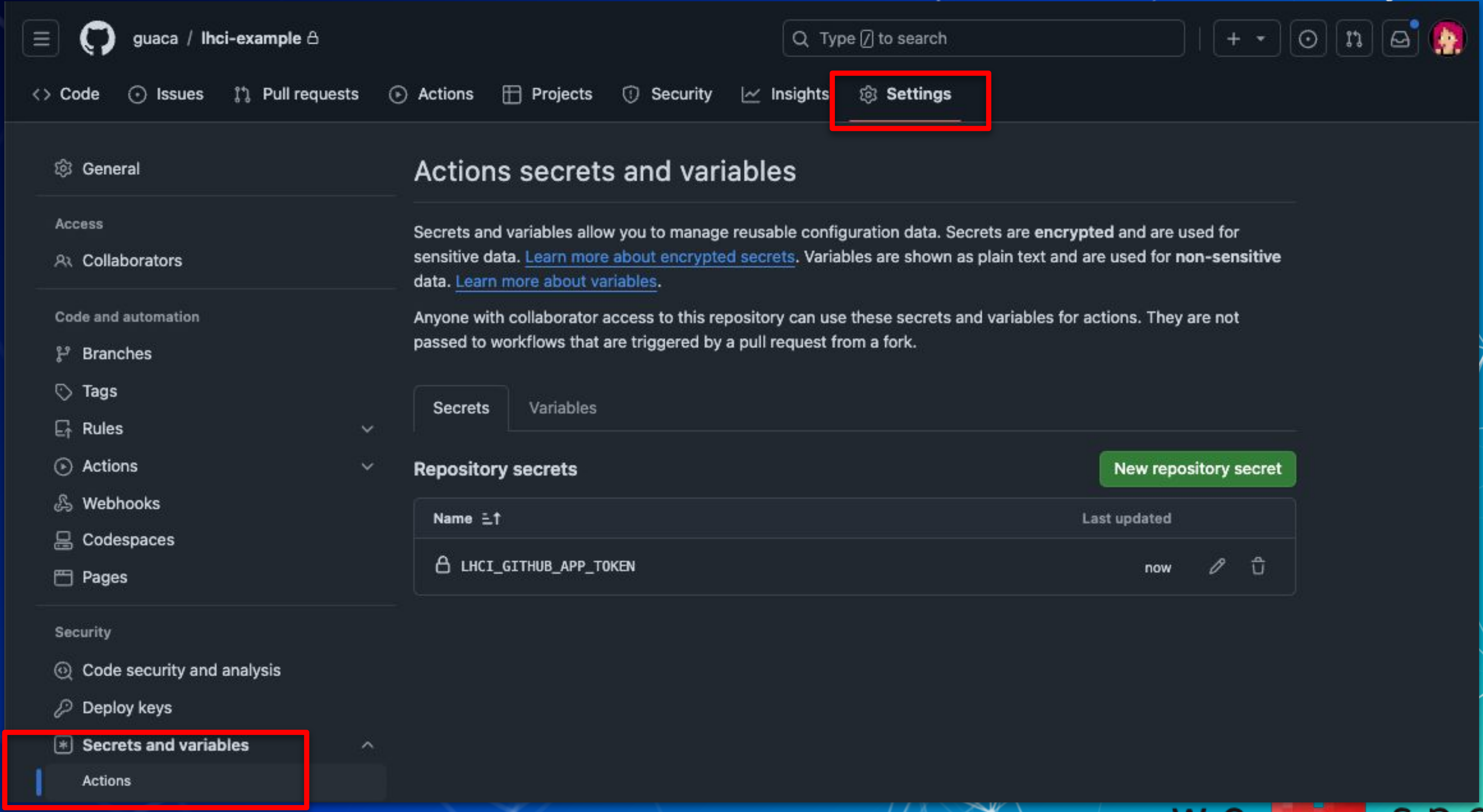
Actions secrets and variables

Secrets and variables allow you to manage reusable configuration data. Secrets are **encrypted** and are used for sensitive data. Learn more about encrypted secrets. Variables are shown as plain text and are used for **non-sensitive** data. Learn more about variables.

Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.

Repository secrets

Name	Last updated
LHCI_GITHUB_APP_TOKEN	now



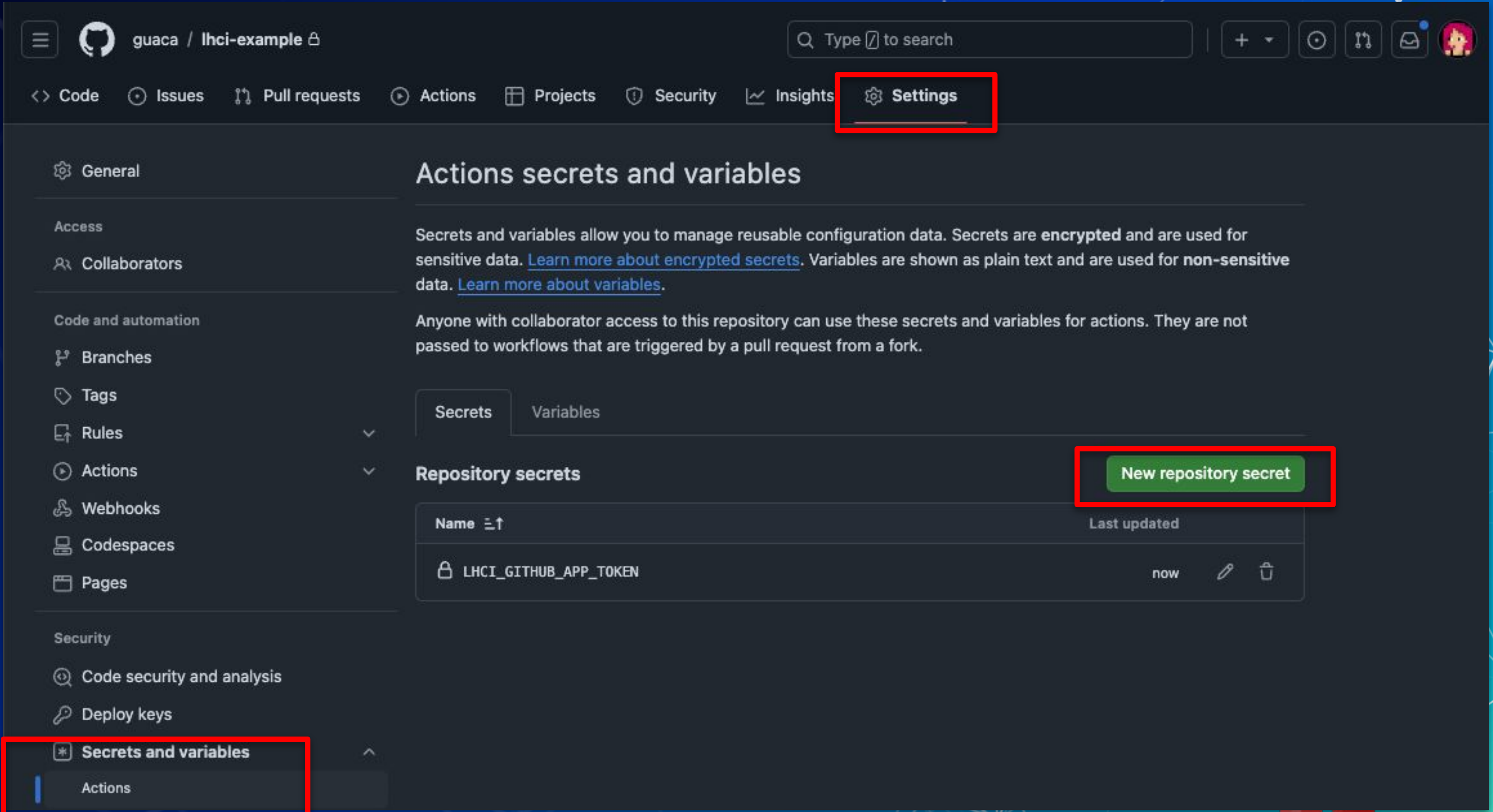
Actions secrets and variables

Secrets and variables allow you to manage reusable configuration data. Secrets are **encrypted** and are used for sensitive data. Learn more about encrypted secrets. Variables are shown as plain text and are used for **non-sensitive** data. Learn more about variables.

Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.

Repository secrets

Name	Last updated
LHCI_GITHUB_APP_TOKEN	now



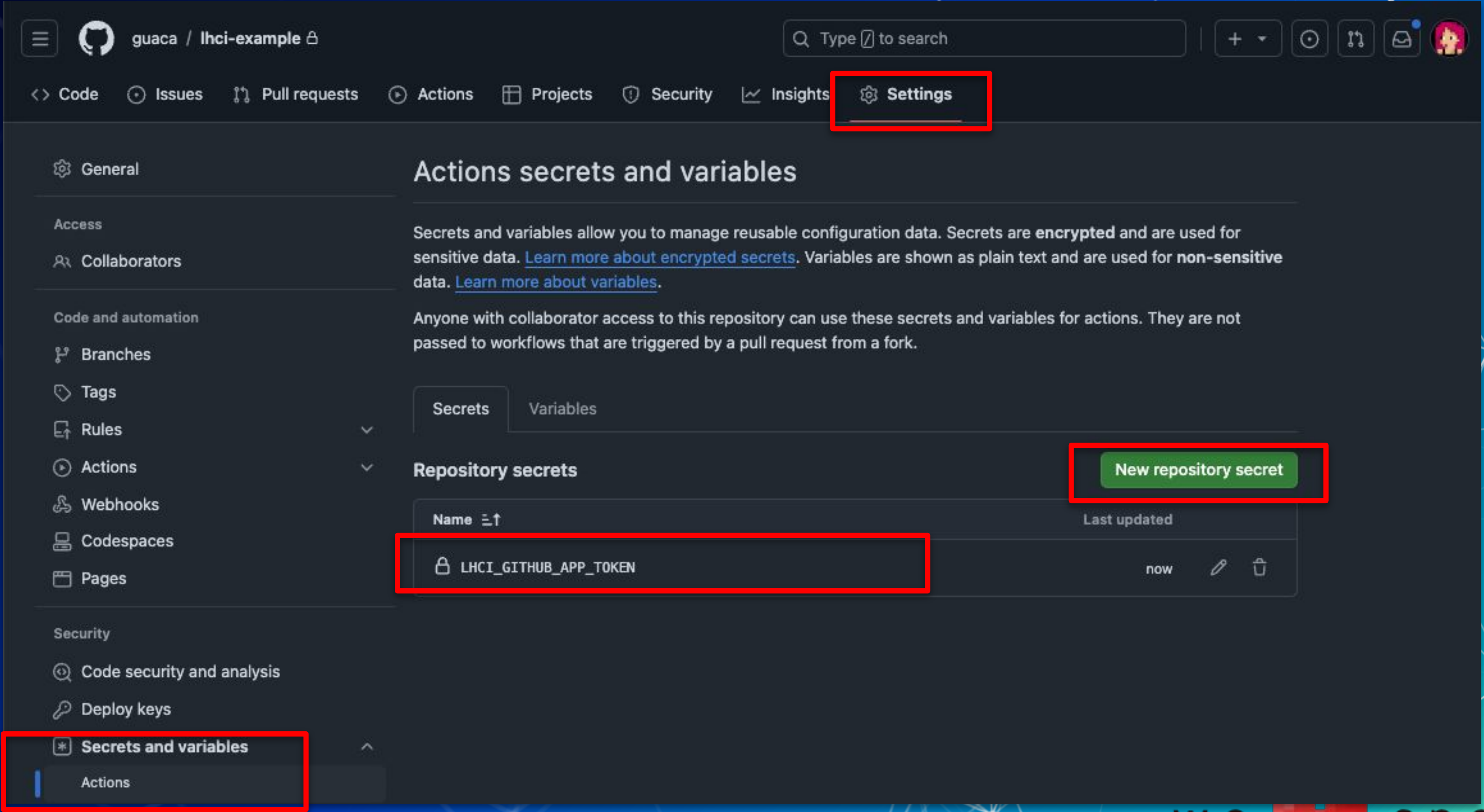
Actions secrets and variables

Secrets and variables allow you to manage reusable configuration data. Secrets are **encrypted** and are used for sensitive data. Learn more about encrypted secrets. Variables are shown as plain text and are used for **non-sensitive** data. Learn more about variables.

Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.

Repository secrets

Name	Last updated
LHCI_GITHUB_APP_TOKEN	now



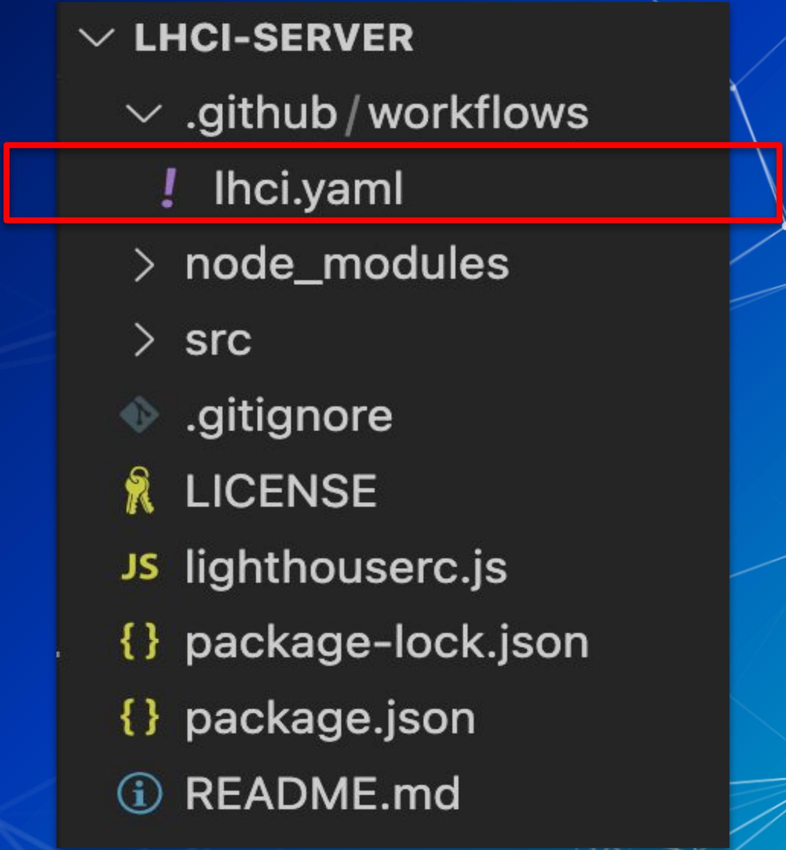
Actions secrets and variables

Secrets and variables allow you to manage reusable configuration data. Secrets are **encrypted** and are used for sensitive data. Learn more about encrypted secrets. Variables are shown as plain text and are used for **non-sensitive** data. Learn more about variables.

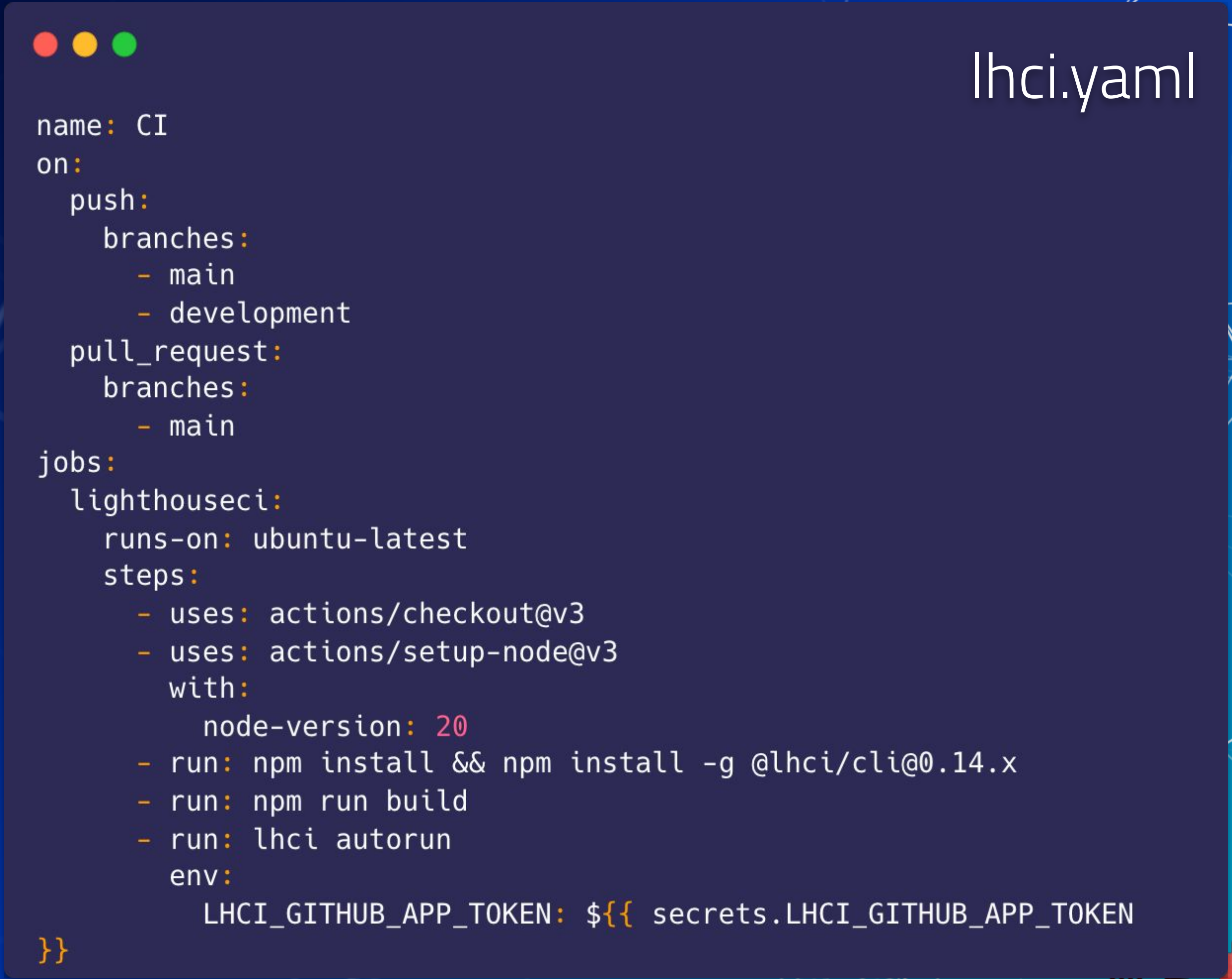
Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.

Repository secrets

Name	Last updated
LHCI_GITHUB_APP_TOKEN	now



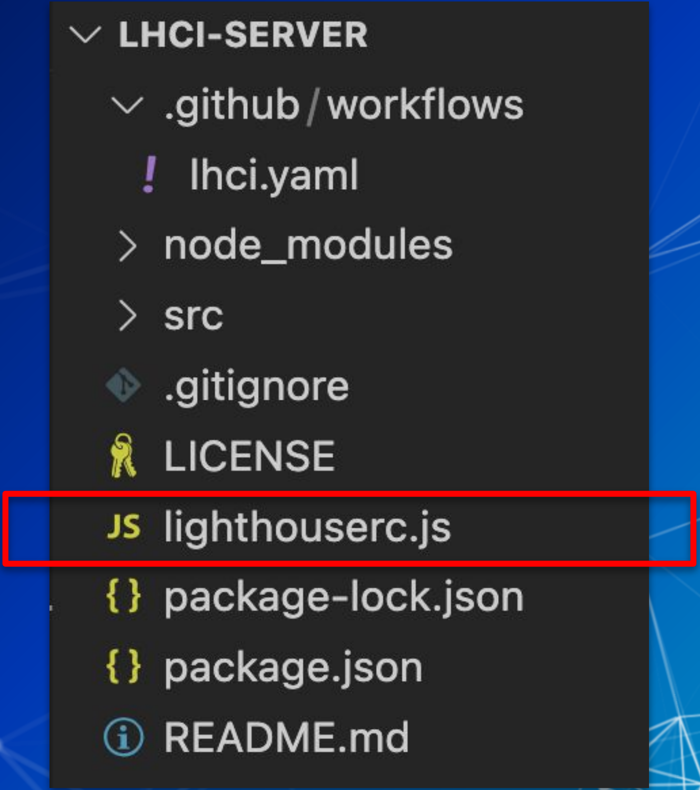
```
LHCI-SERVER
  .github / workflows
    lhci.yaml
  node_modules
  src
  .gitignore
  LICENSE
  lighthouserc.js
  package-lock.json
  package.json
  README.md
```



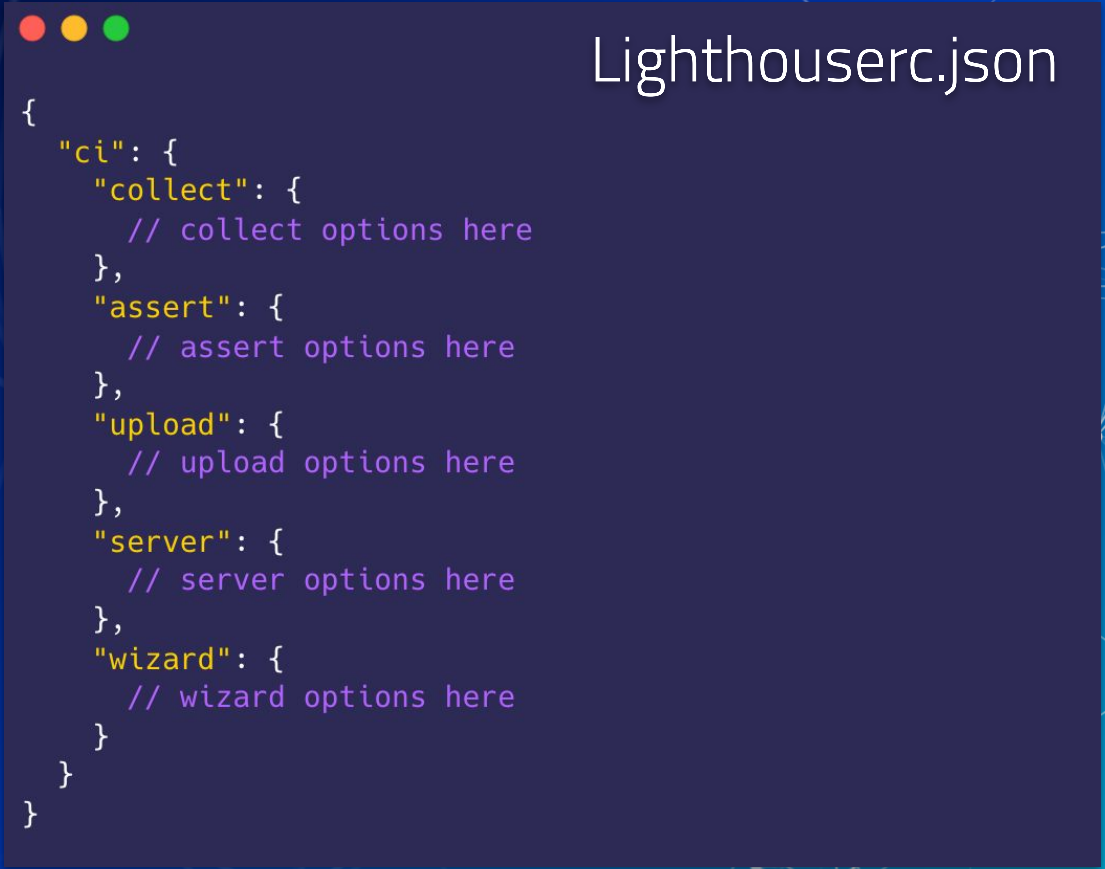
lhci.yaml

```yaml
name: CI
on:
  push:
    branches:
      - main
      - development
  pull_request:
    branches:
      - main
jobs:
  lighthouseci:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-node@v3
        with:
          node-version: 20
      - run: npm install && npm install -g @lhci/cli@0.14.x
      - run: npm run build
      - run: lhci autorun
        env:
          LHCI_GITHUB_APP_TOKEN: ${{ secrets.LHCI_GITHUB_APP_TOKEN
}}
```

https://github.com/GoogleChrome/lighthouse-ci/



```
LHCI-SERVER
  .github / workflows
    lhci.yaml
  node_modules
  src
  .gitignore
  LICENSE
  lighthouserc.js
  package-lock.json
  package.json
  README.md
```

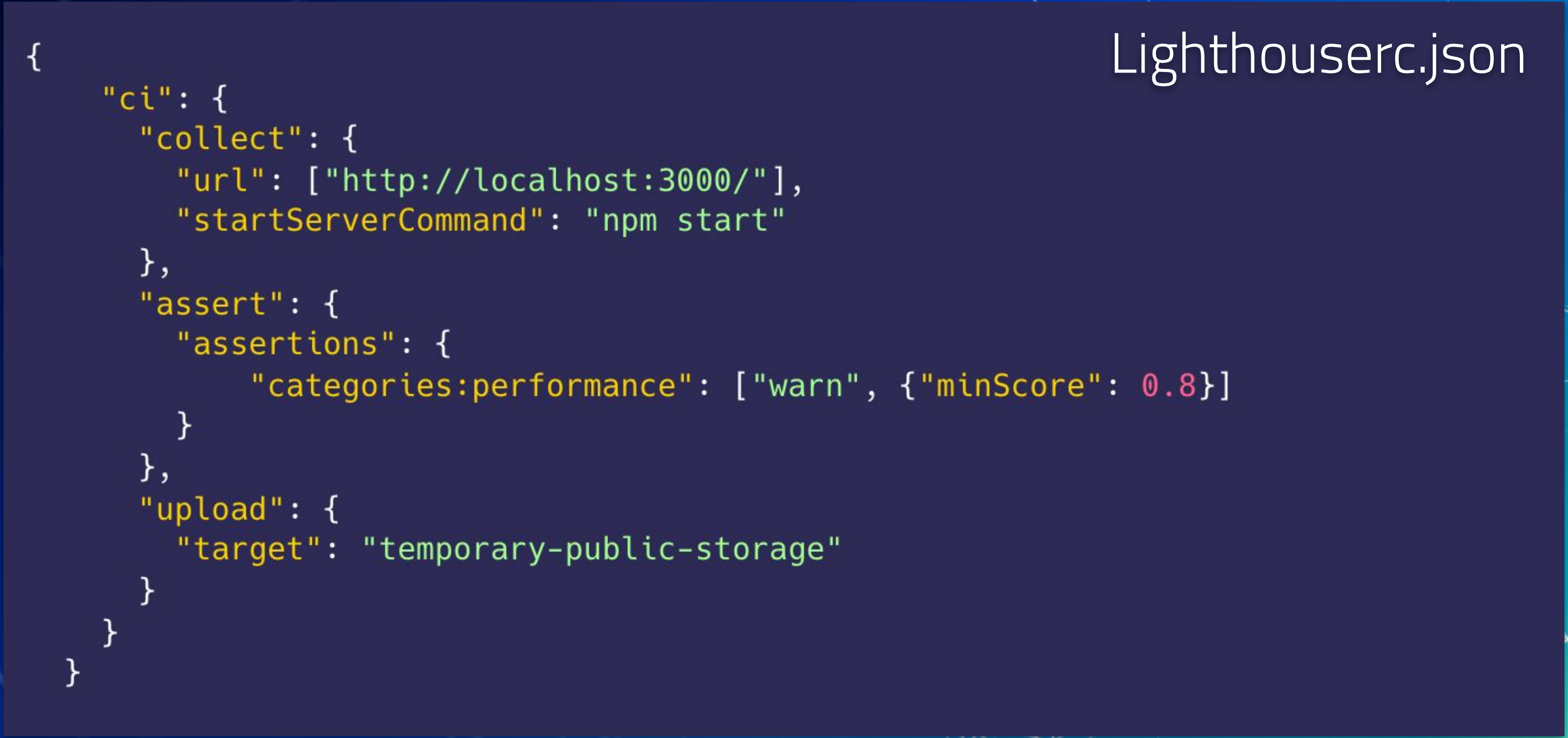


Lighthouserc.json

```
{
  "ci": {
    "collect": {
      // collect options here
    },
    "assert": {
      // assert options here
    },
    "upload": {
      // upload options here
    },
    "server": {
      // server options here
    },
    "wizard": {
      // wizard options here
    }
  }
}
```

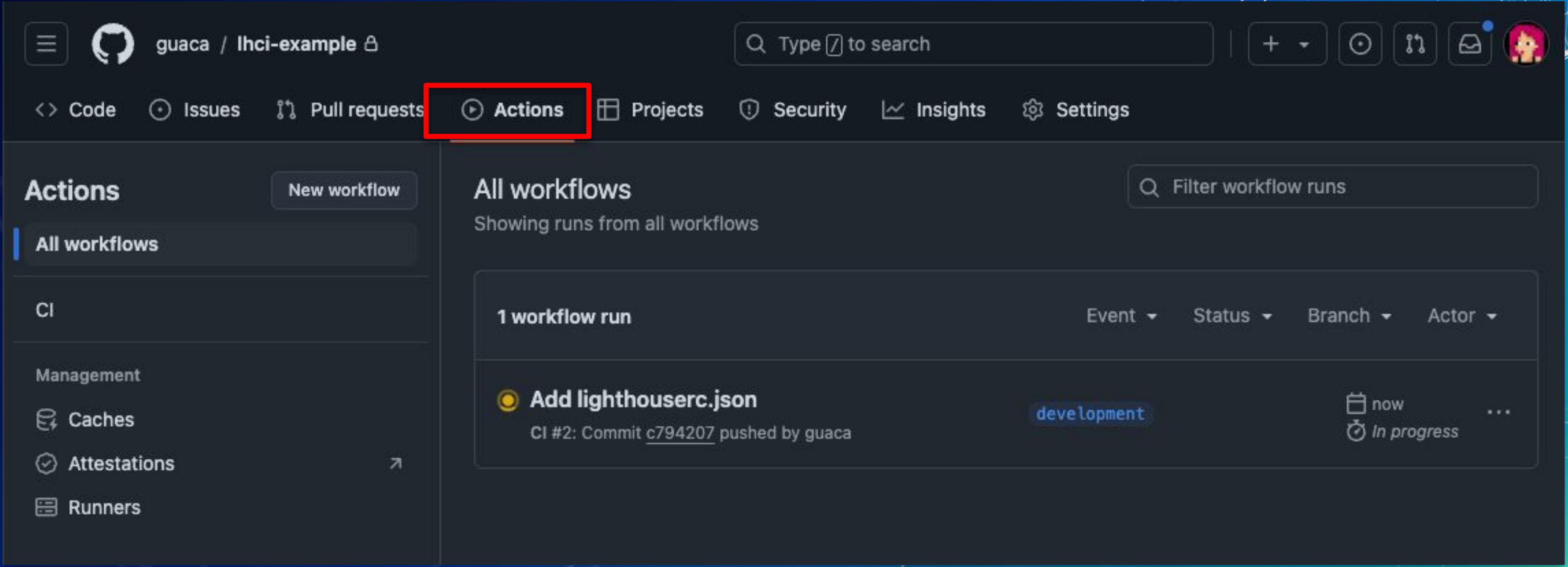
https://github.com/GoogleChrome/lighthouse-ci/blob/main/docs/configuration.md

Lighthouserc.json



```
{
    "ci": {
      "collect": {
        "url": ["http://localhost:3000/"],
        "startServerCommand": "npm start"
      },
      "assert": {
        "assertions": {
            "categories:performance": ["warn", {"minScore": 0.8}]
        }
      },
      "upload": {
        "target": "temporary-public-storage"
      }
    }
  }
```

See it in action!



All workflows

Showing runs from all workflows

1 workflow run

Add lighthouserc.json

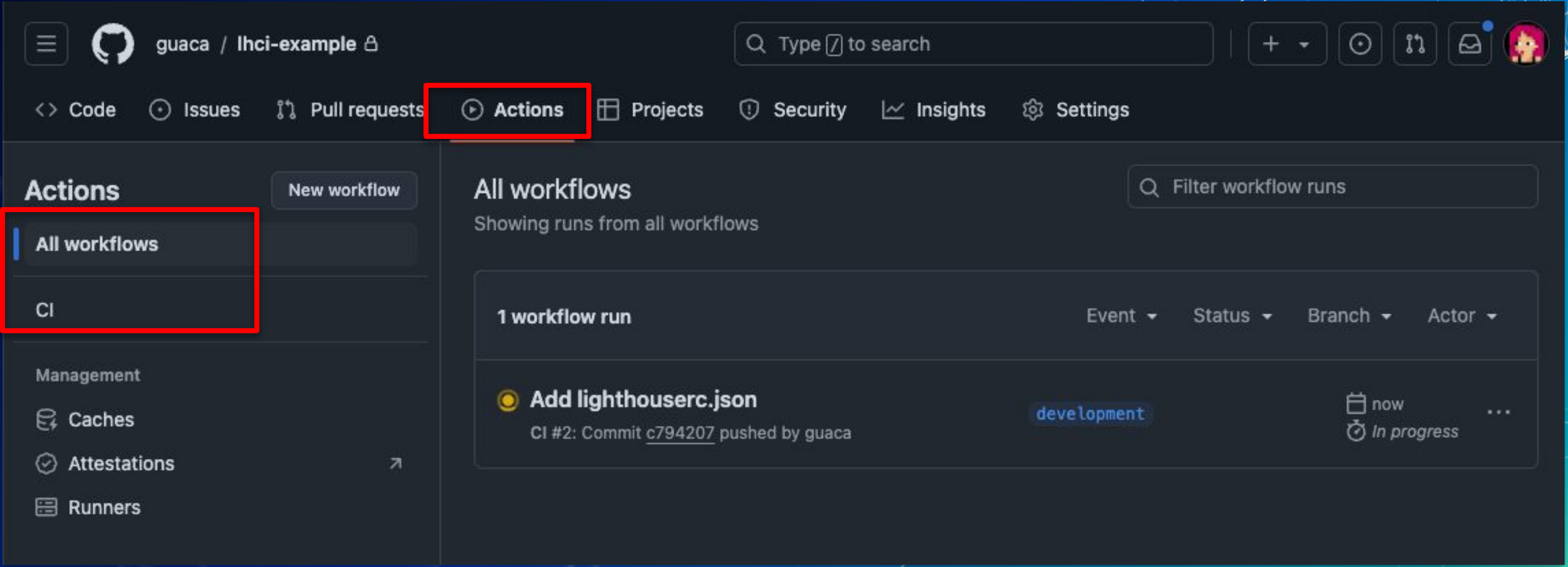
CI #2: Commit c794207 pushed by guaca

development

now

In progress

See it in action!



guaca / lhci-example

Type / to search

Code | Issues | Pull requests | Actions | Projects | Security | Insights | Settings

Actions

New workflow

- All workflows
- CI

Management

- Caches
- Attestations
- Runners

All workflows

Showing runs from all workflows

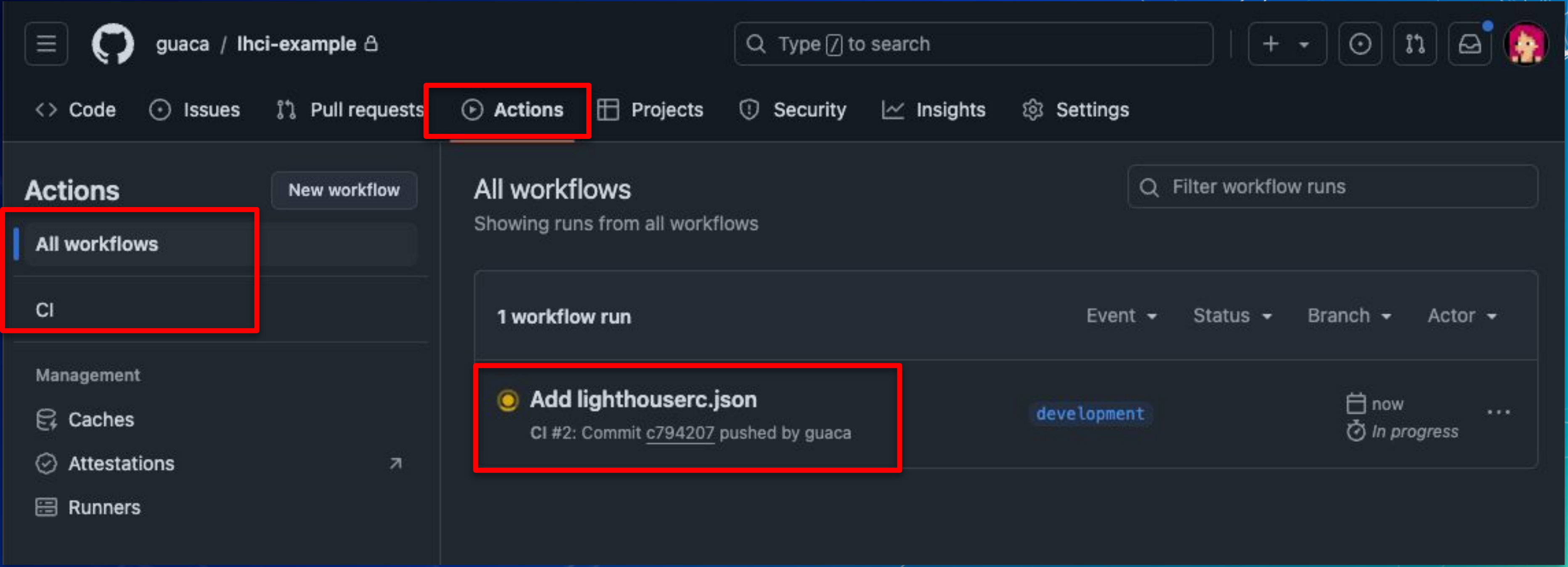
Filter workflow runs

1 workflow run — Event ▾ Status ▾ Branch ▾ Actor ▾

Add lighthouserc.json
CI #2: Commit c794207 pushed by guaca
development
now
In progress

we ❤ speed

See it in action!



Actions

New workflow

All workflows

CI

Management

Caches

Attestations

Runners

All workflows

Showing runs from all workflows

Filter workflow runs

1 workflow run

Event ▾ Status ▾ Branch ▾ Actor ▾

Add lighthouserc.json

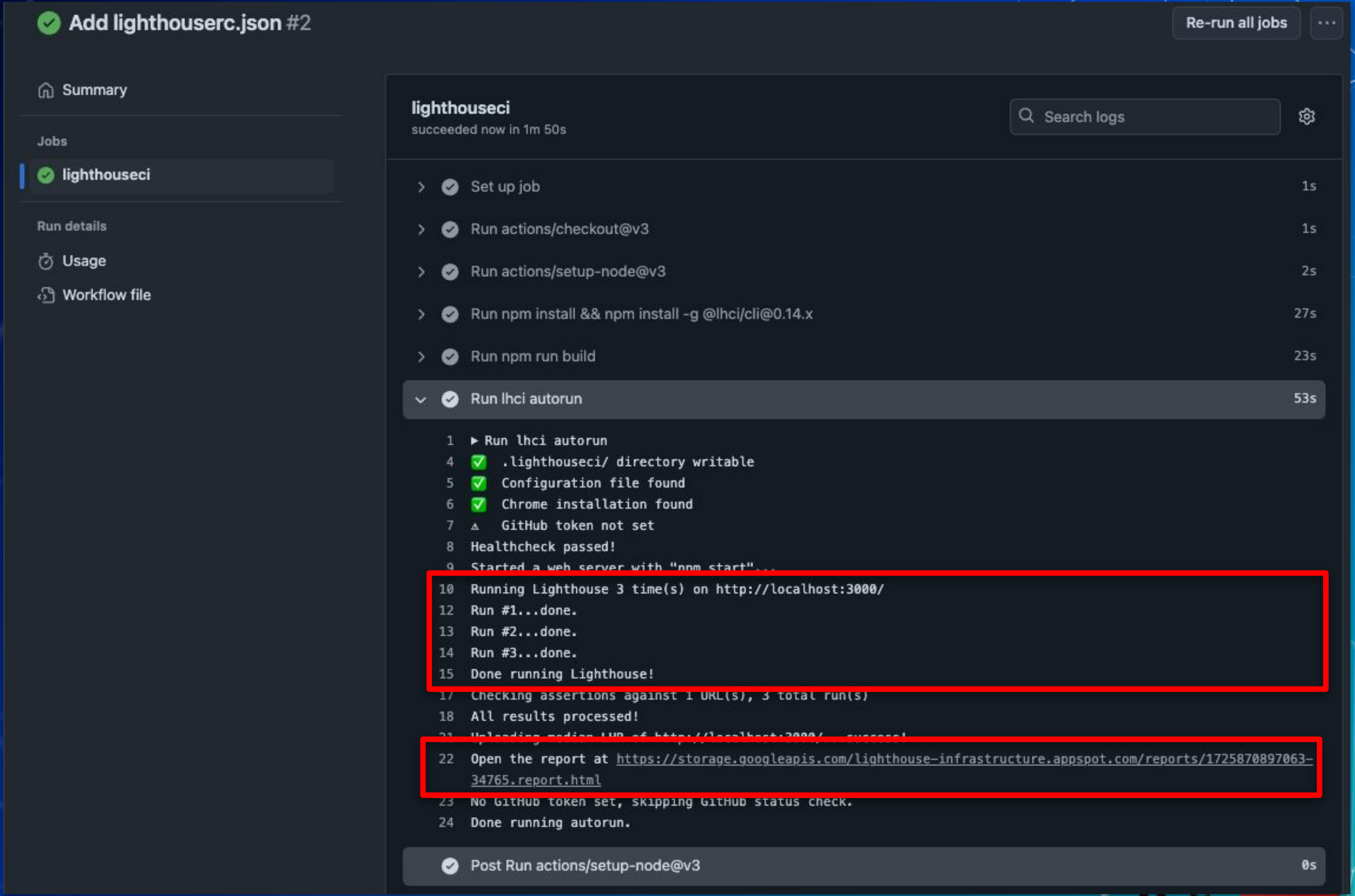
CI #2: Commit c794207 pushed by guaca

`development`

now

In progress

we speed



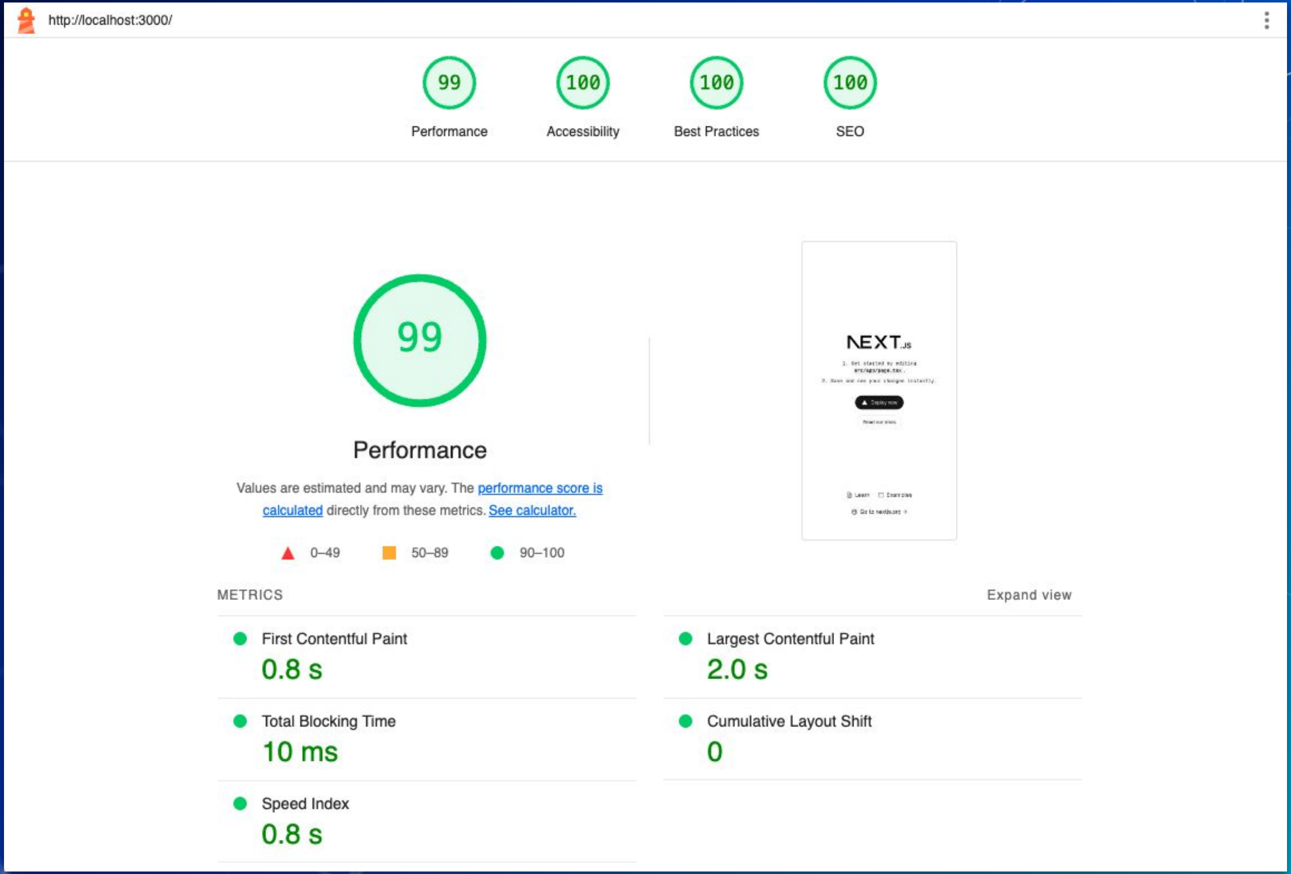
lighthouseci

succeeded now in 1m 50s

- Set up job 1s
- Run actions/checkout@v3 1s
- Run actions/setup-node@v3 2s
- Run npm install && npm install -g @lhci/cli@0.14.x 27s
- Run npm run build 23s
- Run lhci autorun 53s

```
▶ Run lhci autorun
✅  .lighthouseci/ directory writable
✅  Configuration file found
✅  Chrome installation found
⚠   GitHub token not set
Healthcheck passed!
Started a web server with "npm start"...
Running Lighthouse 3 time(s) on http://localhost:3000/
Run #1...done.
Run #2...done.
Run #3...done.
Done running Lighthouse!
Checking assertions against 1 URL(s), 3 total run(s)
All results processed!
Uploading median LHR of http://localhost:3000/...success!
Open the report at https://storage.googleapis.com/lighthouse-infrastructure.appspot.com/reports/1725870897063-34765.report.html
No GitHub token set, skipping GitHub status check.
Done running autorun.
```

- Post Run actions/setup-node@v3 0s



http://localhost:3000/

99 Performance | 100 Accessibility | 100 Best Practices | 100 SEO

99

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49   ■ 50–89   ● 90–100

METRICS   Expand view

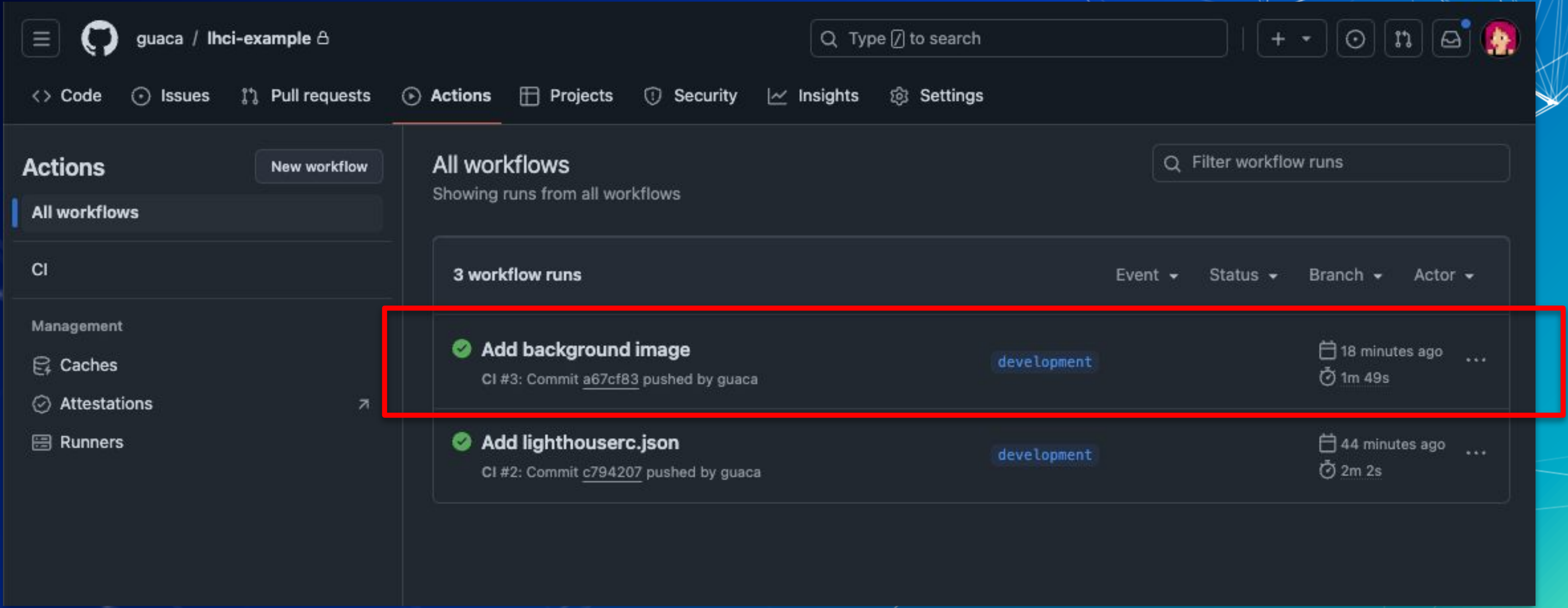
● First Contentful Paint
0.8 s

● Largest Contentful Paint
2.0 s

● Total Blocking Time
10 ms

● Cumulative Layout Shift
0

● Speed Index
0.8 s



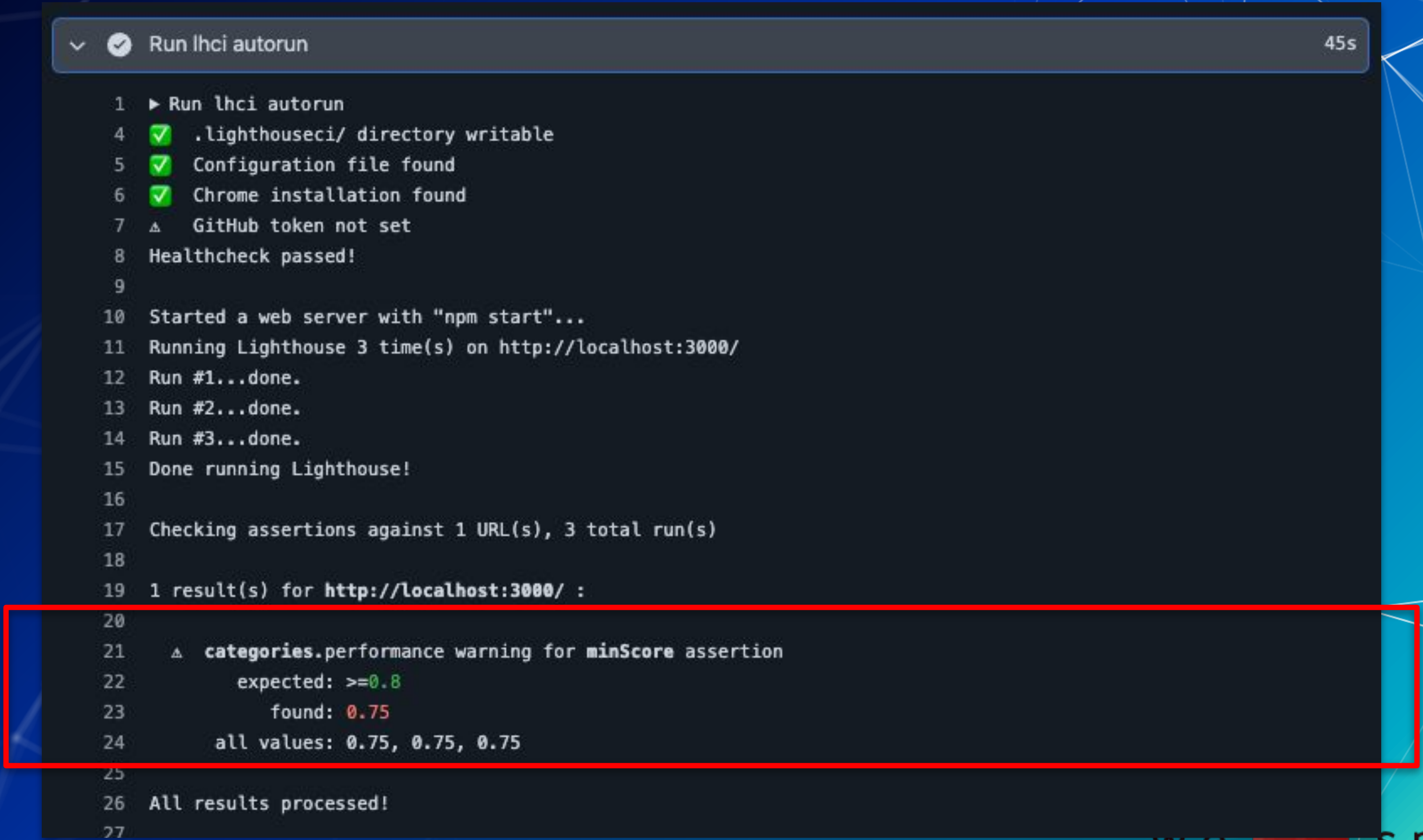
All workflows

Showing runs from all workflows

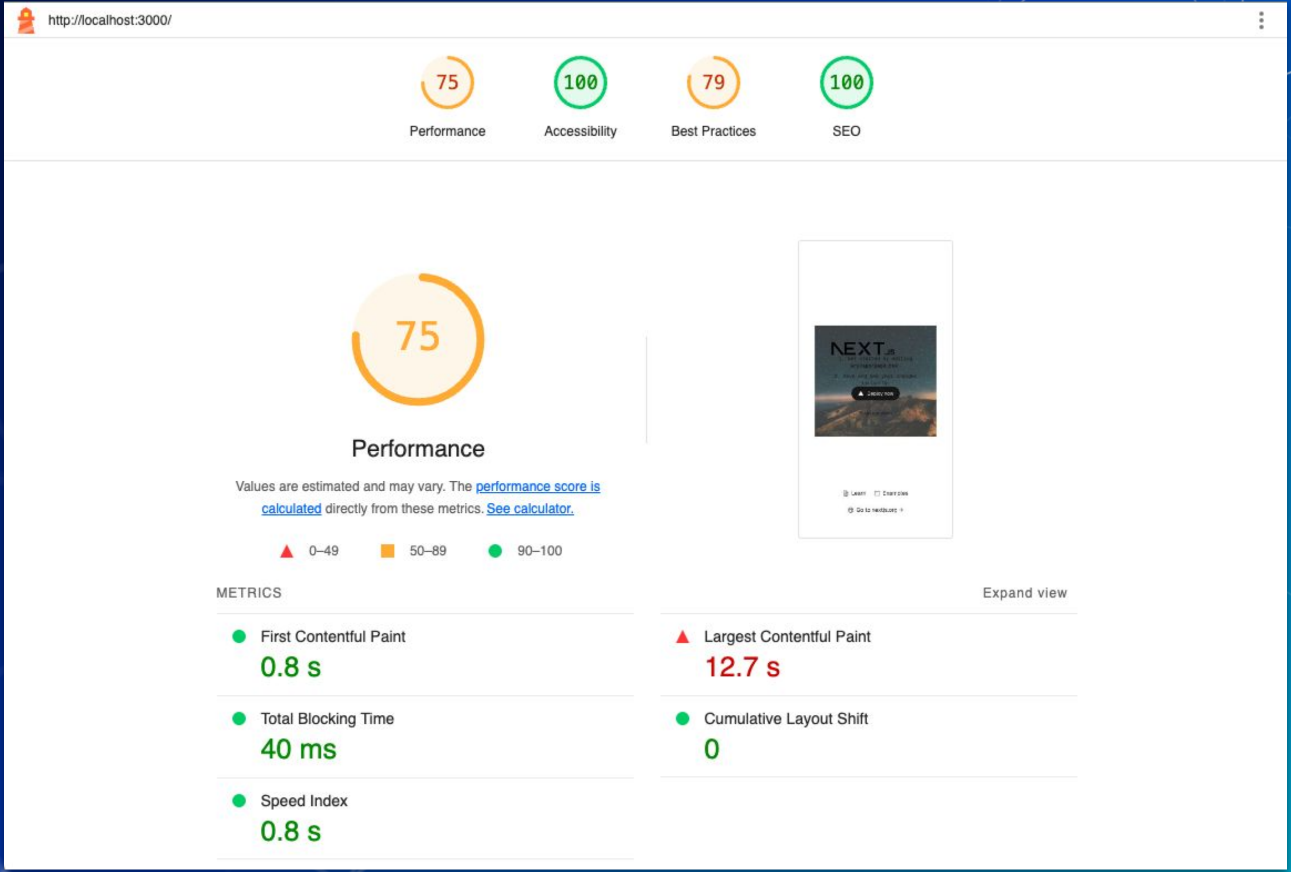
3 workflow runs

✓ **Add background image**
CI #3: Commit a67cf83 pushed by guaca
`development`
18 minutes ago
1m 49s

✓ **Add lighthouserc.json**
CI #2: Commit c794207 pushed by guaca
`development`
44 minutes ago
2m 2s



```
▶ Run lhci autorun
✅  .lighthouseci/ directory writable
✅  Configuration file found
✅  Chrome installation found
⚠  GitHub token not set
Healthcheck passed!

Started a web server with "npm start"...
Running Lighthouse 3 time(s) on http://localhost:3000/
Run #1...done.
Run #2...done.
Run #3...done.
Done running Lighthouse!

Checking assertions against 1 URL(s), 3 total run(s)

1 result(s) for http://localhost:3000/ :

  ⚠  categories.performance warning for minScore assertion
        expected: >=0.8
           found: 0.75
      all values: 0.75, 0.75, 0.75

All results processed!
```



http://localhost:3000/

75 Performance

100 Accessibility

79 Best Practices

100 SEO

75

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

METRICS

Expand view

● First Contentful Paint
0.8 s

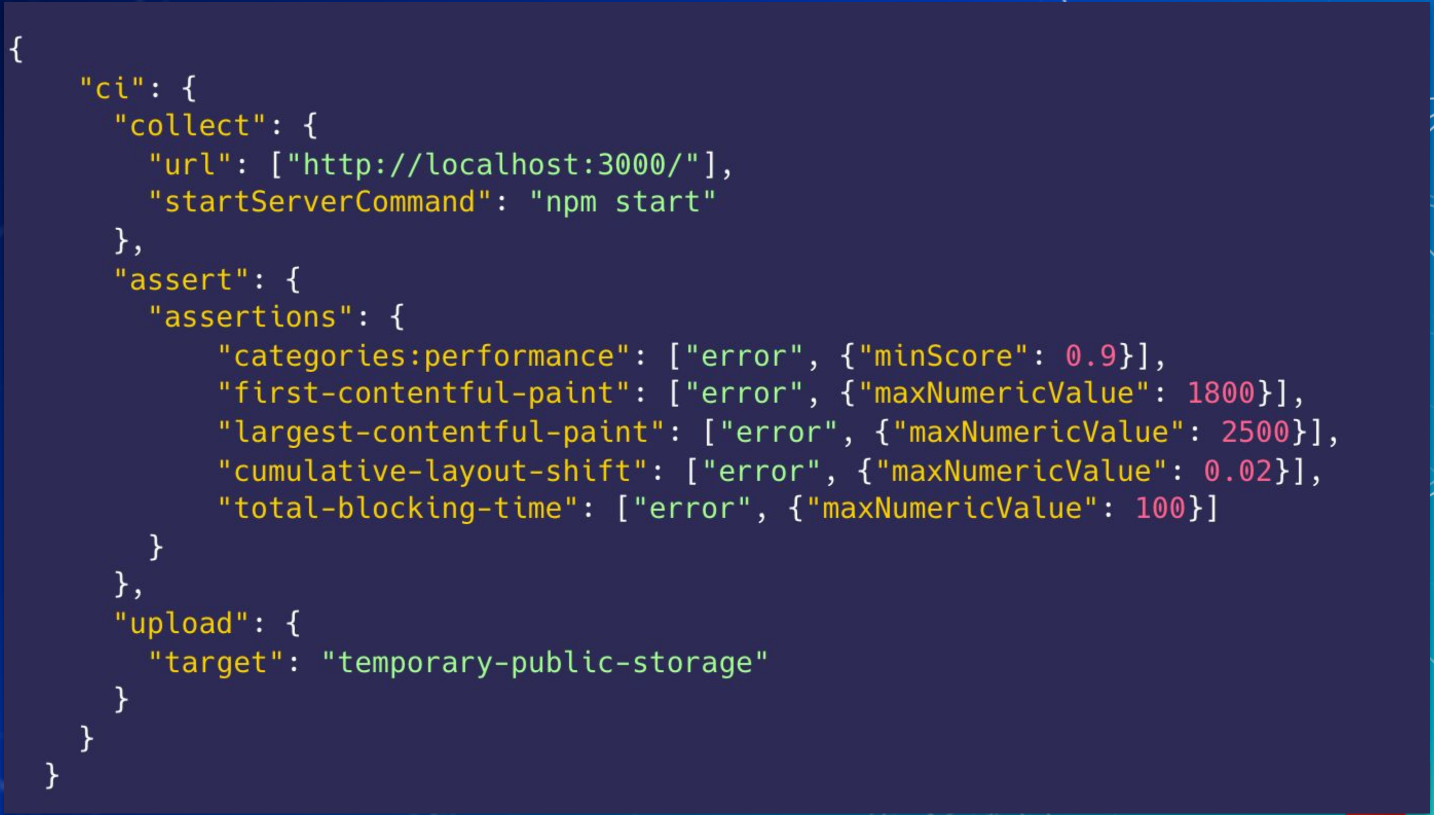
▲ Largest Contentful Paint
12.7 s

● Total Blocking Time
40 ms

● Cumulative Layout Shift
0

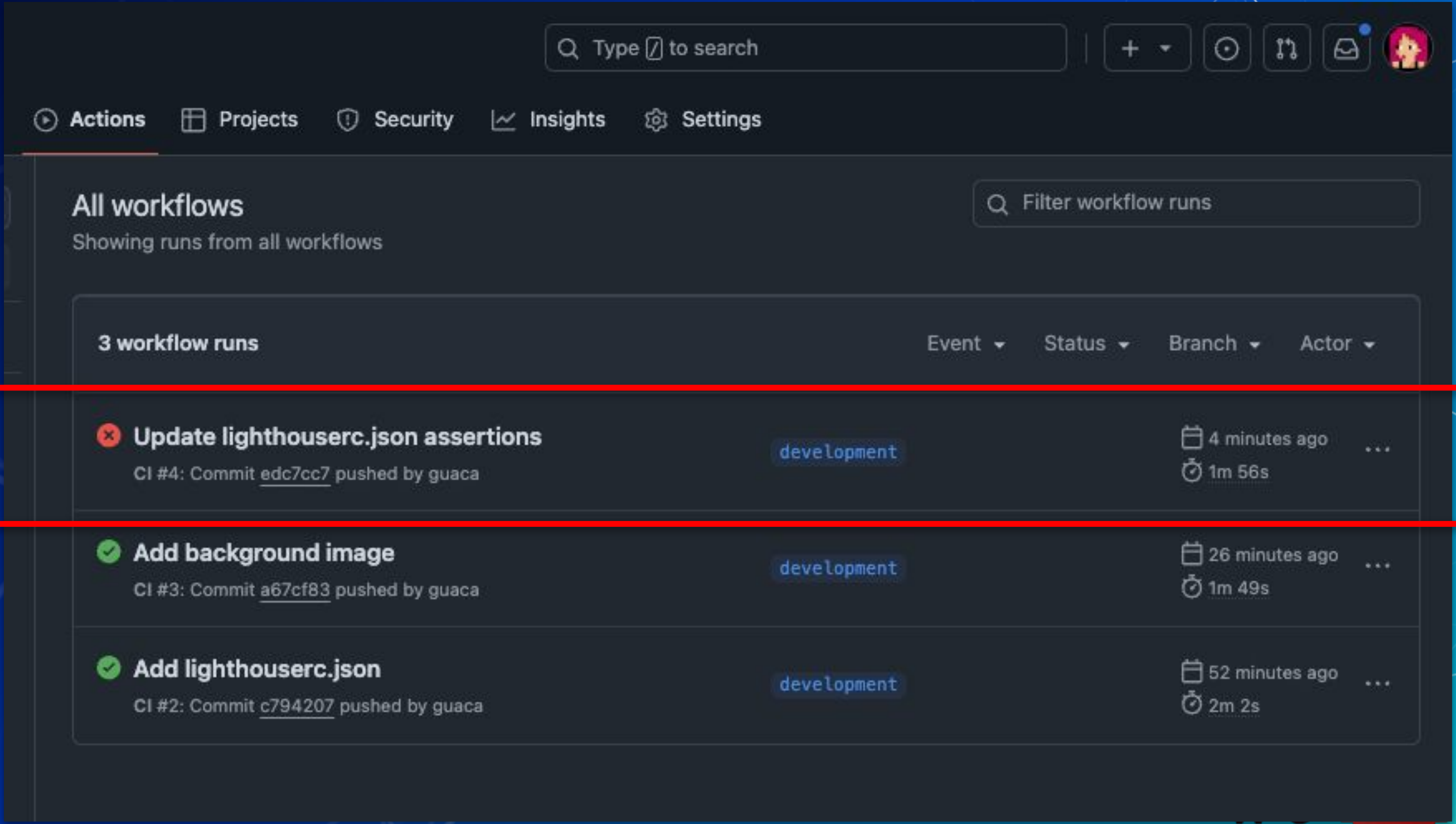
● Speed Index
0.8 s

Perf Budgets Matters



```
{
  "ci": {
   "collect": {
    "url": ["http://localhost:3000/"],
    "startServerCommand": "npm start"
   },
   "assert": {
    "assertions": {
      "categories:performance": ["error", {"minScore": 0.9}],
      "first-contentful-paint": ["error", {"maxNumericValue": 1800}],
      "largest-contentful-paint": ["error", {"maxNumericValue": 2500}],
      "cumulative-layout-shift": ["error", {"maxNumericValue": 0.02}],
      "total-blocking-time": ["error", {"maxNumericValue": 100}]
    }
   },
   "upload": {
    "target": "temporary-public-storage"
   }
  }
 }
```

Heads up!



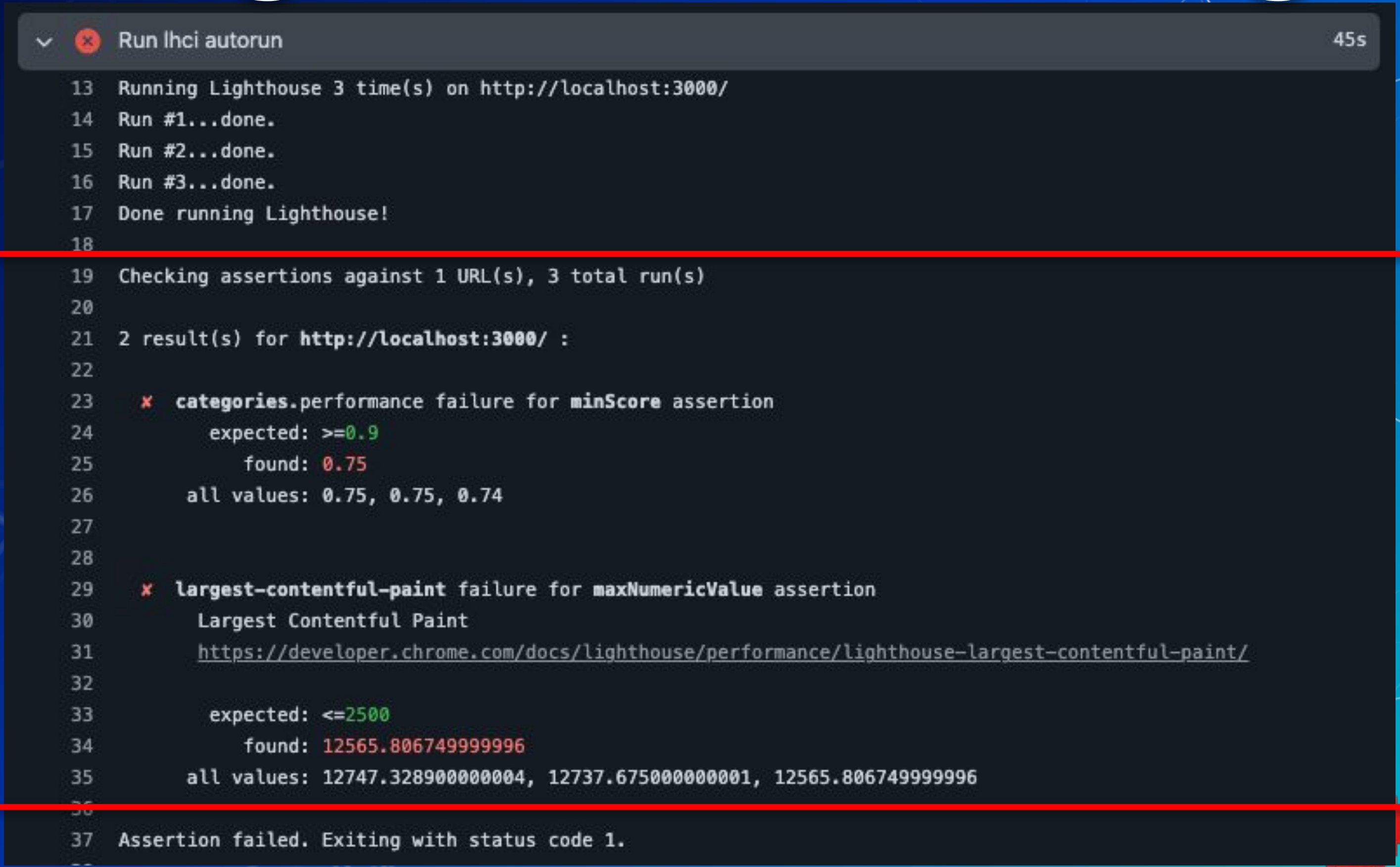
All workflows

Showing runs from all workflows

3 workflow runs

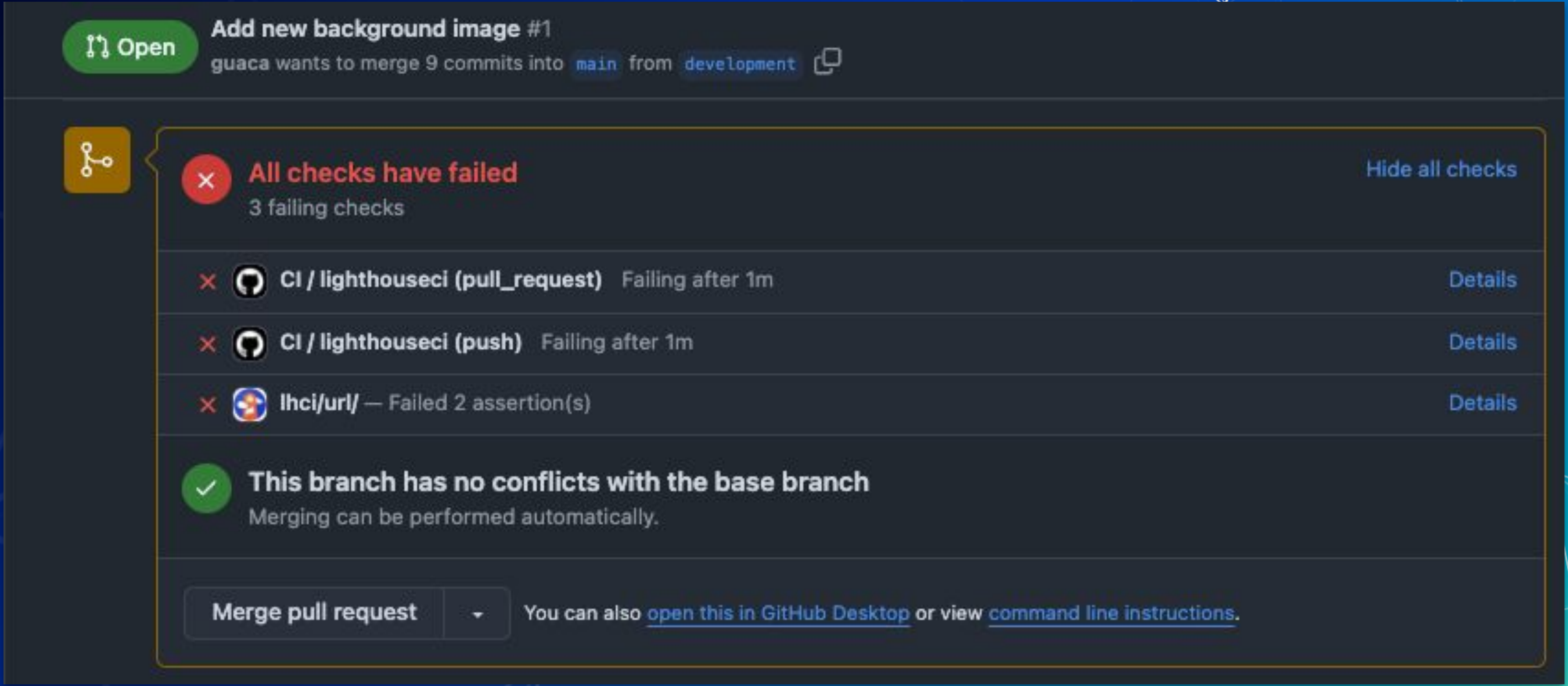
✗ **Update lighthouserc.json assertions** CI #4: Commit edc7cc7 pushed by guaca	development	4 minutes ago 1m 56s	...
✓ **Add background image** CI #3: Commit a67cf83 pushed by guaca	development	26 minutes ago 1m 49s	...
✓ **Add lighthouserc.json** CI #2: Commit c794207 pushed by guaca	development	52 minutes ago 2m 2s	...

Degradation is coming



Run lhci autorun 45s

```
Running Lighthouse 3 time(s) on http://localhost:3000/
Run #1...done.
Run #2...done.
Run #3...done.
Done running Lighthouse!

Checking assertions against 1 URL(s), 3 total run(s)

2 result(s) for http://localhost:3000/ :

  ✘  categories.performance failure for minScore assertion
        expected: >=0.9
           found: 0.75
      all values: 0.75, 0.75, 0.74


  ✘  largest-contentful-paint failure for maxNumericValue assertion
       Largest Contentful Paint
       https://developer.chrome.com/docs/lighthouse/performance/lighthouse-largest-contentful-paint/

        expected: <=2500
           found: 12565.806749999996
      all values: 12747.328900000004, 12737.675000000001, 12565.806749999996

Assertion failed. Exiting with status code 1.
```



Open Add new background image #1

guaca wants to merge 9 commits into `main` from `development`

All checks have failed

3 failing checks

Hide all checks

✕ CI / lighthouseci (pull_request) Failing after 1m Details

✕ CI / lighthouseci (push) Failing after 1m Details

✕ lhci/url/ — Failed 2 assertion(s) Details

✓ This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request ▾ You can also open this in GitHub Desktop or view command line instructions.

LIGHTHOUSE CI SERVER

Saves historical Lighthouse data, displays trends in a dashboard, and offers an in-depth build comparison UI to uncover differences between builds.

Requirements:

- → Node v16 LTS
- → Database Storage (sqlite, mysql, or postgresql)

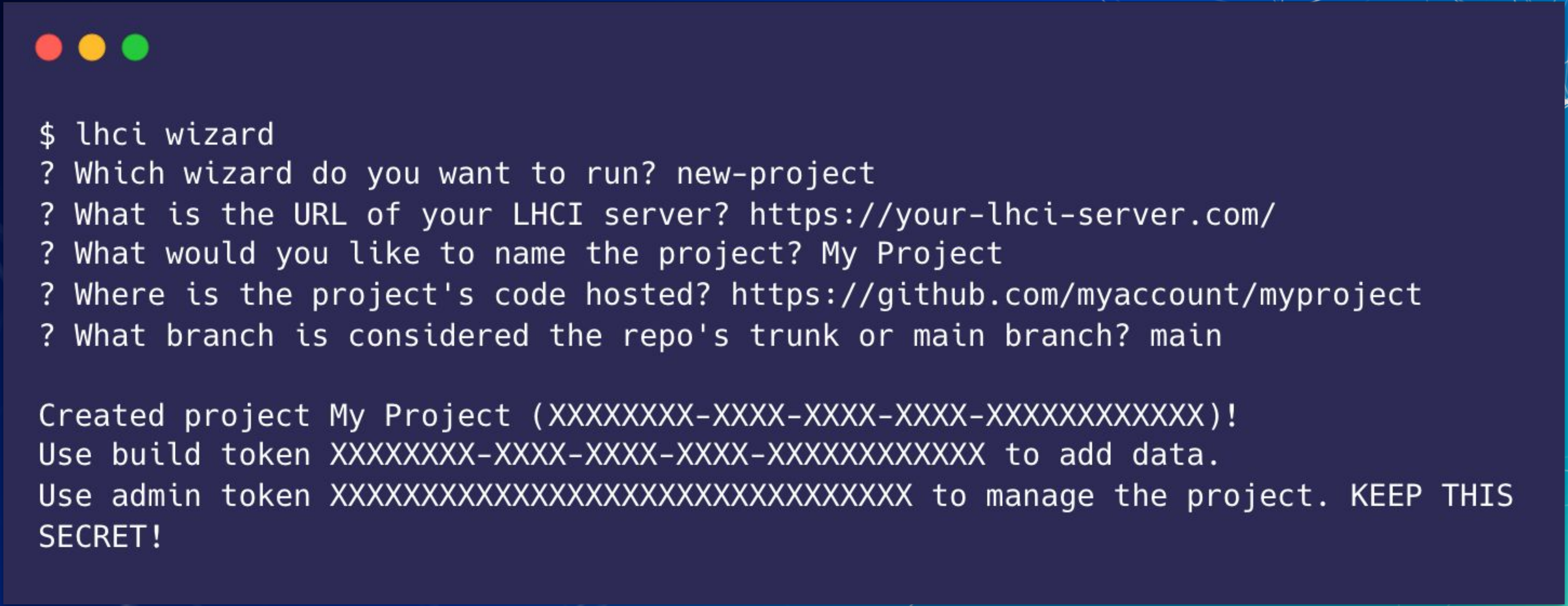
https://github.com/GoogleChrome/lighthouse-ci/blob/main/docs/server.md

Requirements:

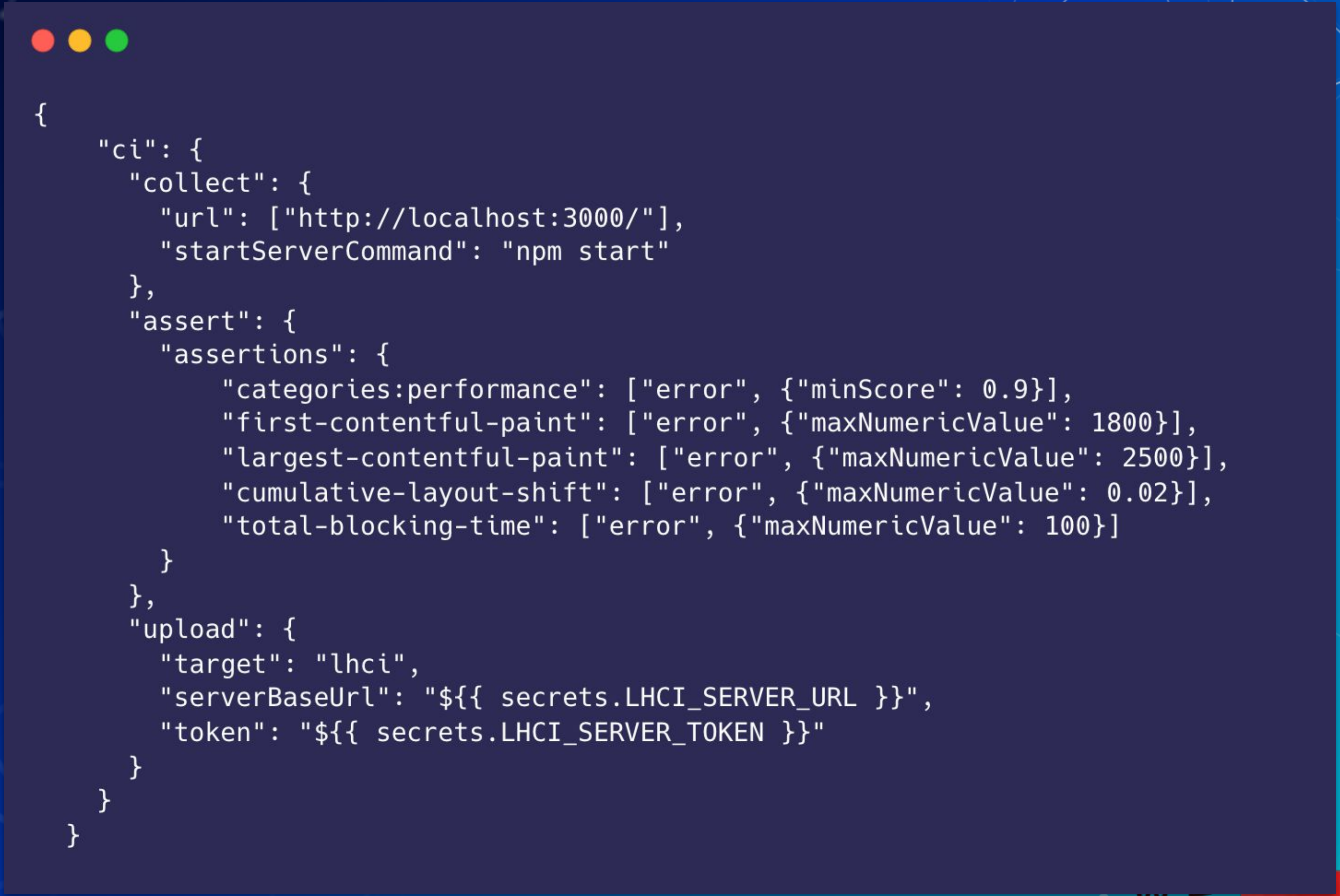
- → Node v16 LTS
- → Database Storage (sqlite, mysql, or postgresql)

https://hub.docker.com/r/patrickhulce/lhci-server

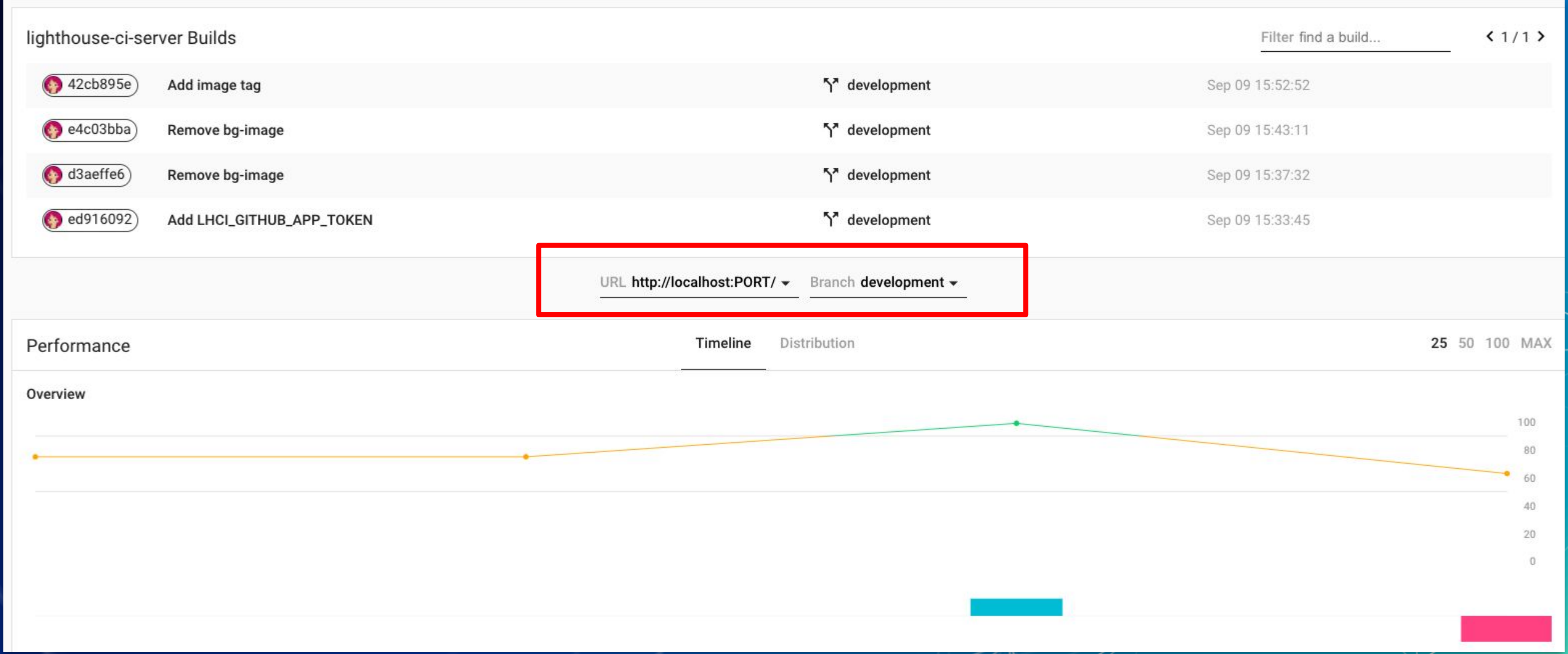
https://github.com/GoogleChrome/lighthouse-ci/blob/main/docs/server.md



```
$ lhci wizard
? Which wizard do you want to run? new-project
? What is the URL of your LHCI server? https://your-lhci-server.com/
? What would you like to name the project? My Project
? Where is the project's code hosted? https://github.com/myaccount/myproject
? What branch is considered the repo's trunk or main branch? main

Created project My Project (XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX)!
Use build token XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX to add data.
Use admin token XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX to manage the project. KEEP THIS
SECRET!
```



```
{
  "ci": {
   "collect": {
    "url": ["http://localhost:3000/"],
    "startServerCommand": "npm start"
   },
   "assert": {
    "assertions": {
      "categories:performance": ["error", {"minScore": 0.9}],
      "first-contentful-paint": ["error", {"maxNumericValue": 1800}],
      "largest-contentful-paint": ["error", {"maxNumericValue": 2500}],
      "cumulative-layout-shift": ["error", {"maxNumericValue": 0.02}],
      "total-blocking-time": ["error", {"maxNumericValue": 100}]
    }
   },
   "upload": {
    "target": "lhci",
    "serverBaseUrl": "${{ secrets.LHCI_SERVER_URL }}",
    "token": "${{ secrets.LHCI_SERVER_TOKEN }}"
   }
  }
 }
```



lighthouse-ci-server Builds

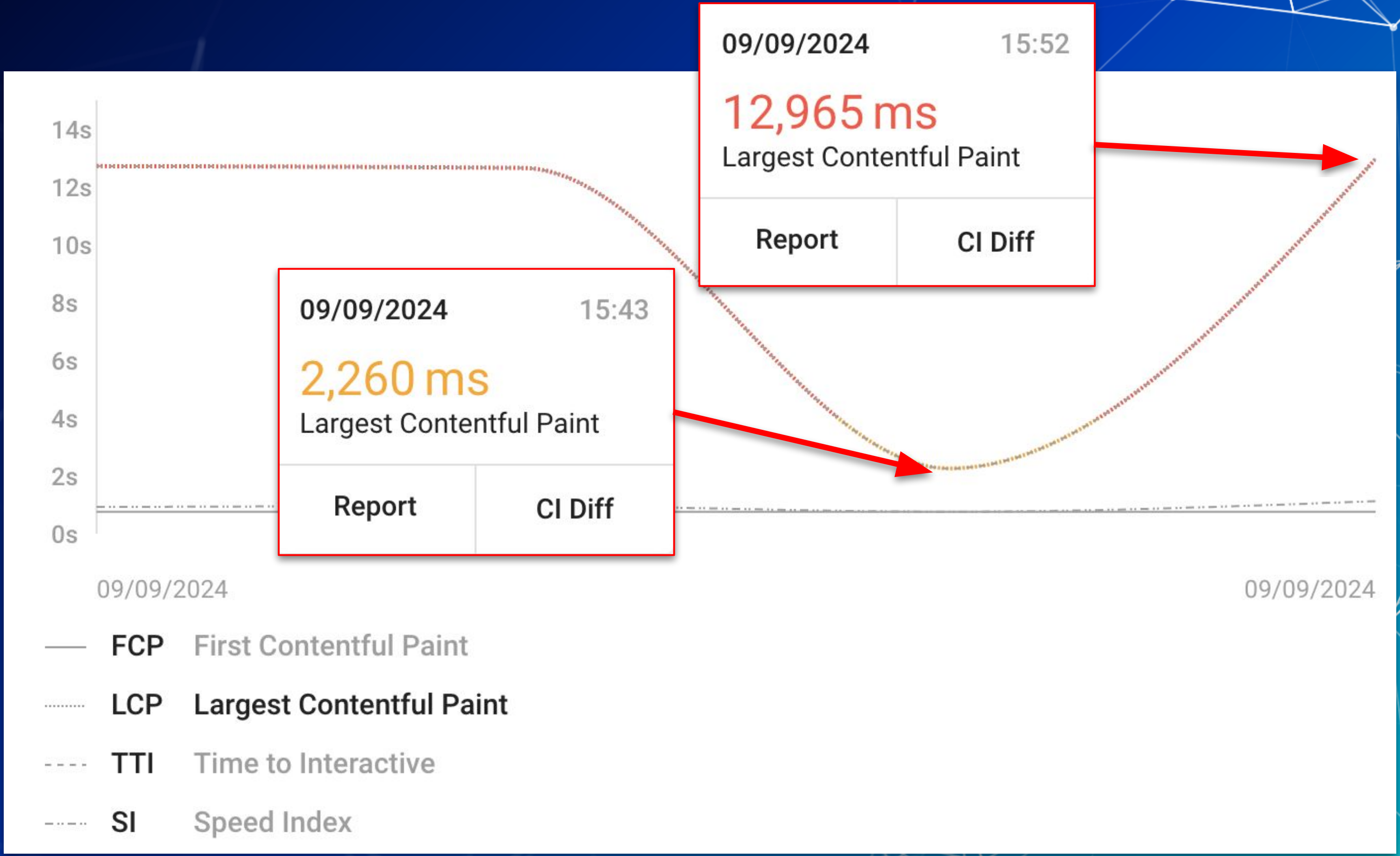
Filter find a build...

‹ 1 / 1 ›

42cb895e	Add image tag	development	Sep 09 15:52:52
e4c03bba	Remove bg-image	development	Sep 09 15:43:11
d3aeffe6	Remove bg-image	development	Sep 09 15:37:32
ed916092	Add LHCI_GITHUB_APP_TOKEN	development	Sep 09 15:33:45

URL http://localhost:PORT/ Branch development

Performance



09/09/2024 15:52

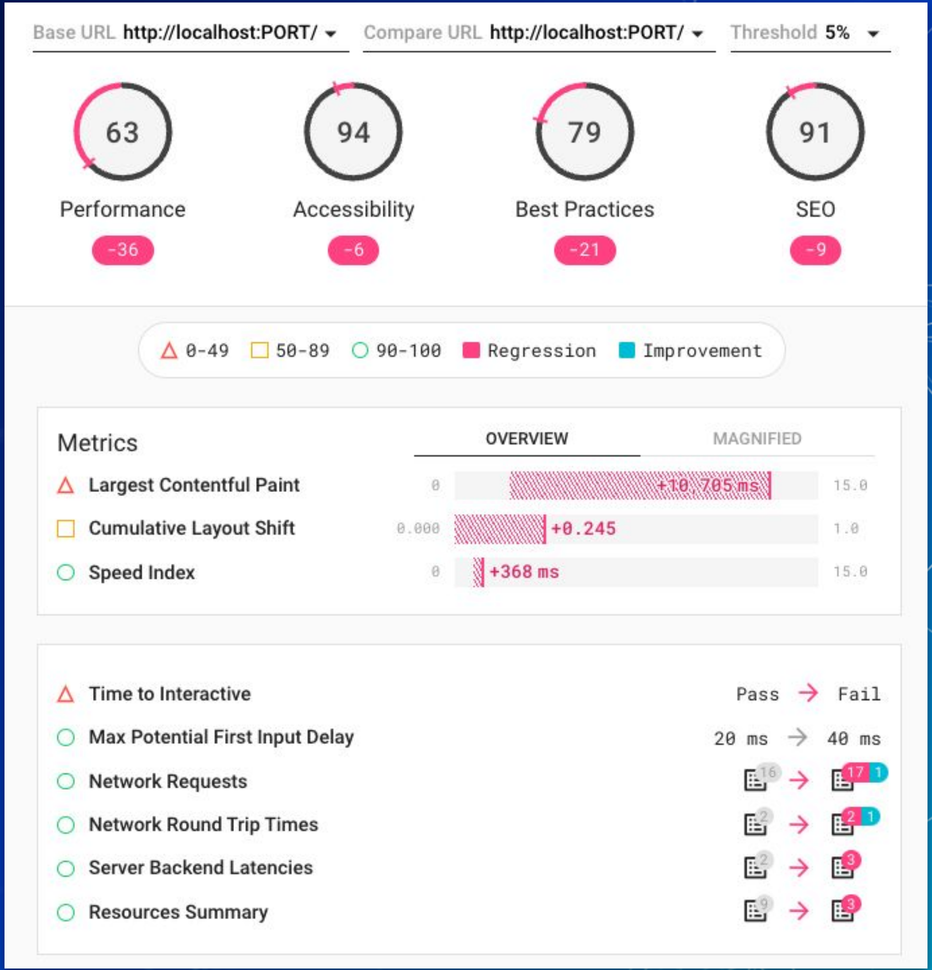
12,965 ms

Largest Contentful Paint

09/09/2024 15:43

2,260 ms

Largest Contentful Paint



Base URL http://localhost:PORT/ ▾ | Compare URL http://localhost:PORT/ ▾ | Threshold 5% ▾

63	94	79	91
Performance	Accessibility	Best Practices	SEO
-36	-6	-21	-9

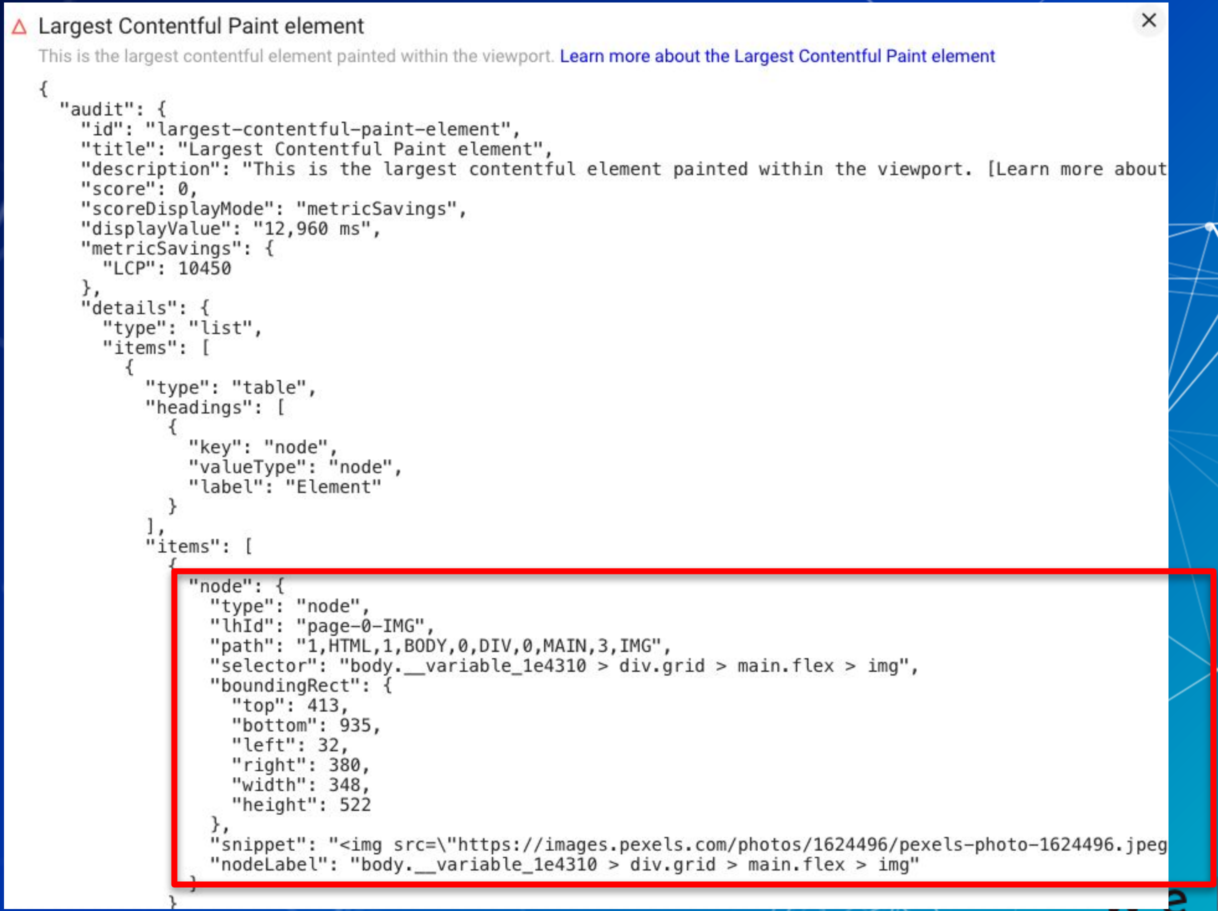
△ 0-49 □ 50-89 ○ 90-100 🟥 Regression 🟦 Improvement

Metrics

OVERVIEW | MAGNIFIED

△	Largest Contentful Paint	0	+10,705 ms	15.0
□	Cumulative Layout Shift	0.000	+0.245	1.0
○	Speed Index	0	+368 ms	15.0

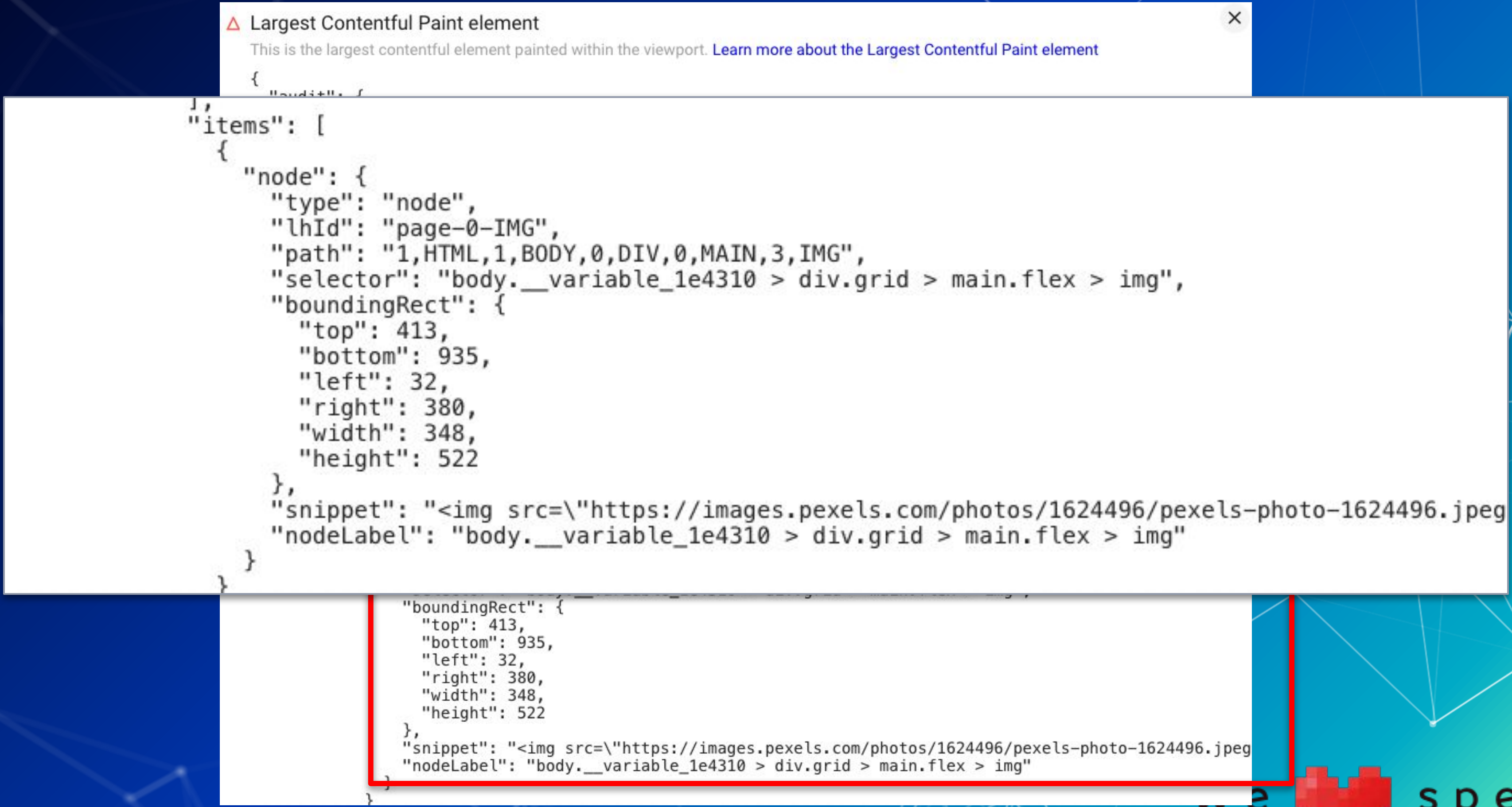
△	Time to Interactive	Pass	→	Fail
○	Max Potential First Input Delay	20 ms	→	40 ms
○	Network Requests	16	→	17 1
○	Network Round Trip Times	2	→	2 1
○	Server Backend Latencies	2	→	3
○	Resources Summary	9	→	3



△ Largest Contentful Paint element

This is the largest contentful element painted within the viewport. Learn more about the Largest Contentful Paint element

```
{
  "audit": {
    "id": "largest-contentful-paint-element",
    "title": "Largest Contentful Paint element",
    "description": "This is the largest contentful element painted within the viewport. [Learn more about
    "score": 0,
    "scoreDisplayMode": "metricSavings",
    "displayValue": "12,960 ms",
    "metricSavings": {
      "LCP": 10450
    },
    "details": {
      "type": "list",
      "items": [
        {
          "type": "table",
          "headings": [
            {
              "key": "node",
              "valueType": "node",
              "label": "Element"
            }
          ],
          "items": [
            {
              "node": {
                "type": "node",
                "lhId": "page-0-IMG",
                "path": "1,HTML,1,BODY,0,DIV,0,MAIN,3,IMG",
                "selector": "body.__variable_1e4310 > div.grid > main.flex > img",
                "boundingRect": {
                  "top": 413,
                  "bottom": 935,
                  "left": 32,
                  "right": 380,
                  "width": 348,
                  "height": 522
                },
                "snippet": "<img src=\"https://images.pexels.com/photos/1624496/pexels-photo-1624496.jpeg
                "nodeLabel": "body.__variable_1e4310 > div.grid > main.flex > img"
              }
            }
```



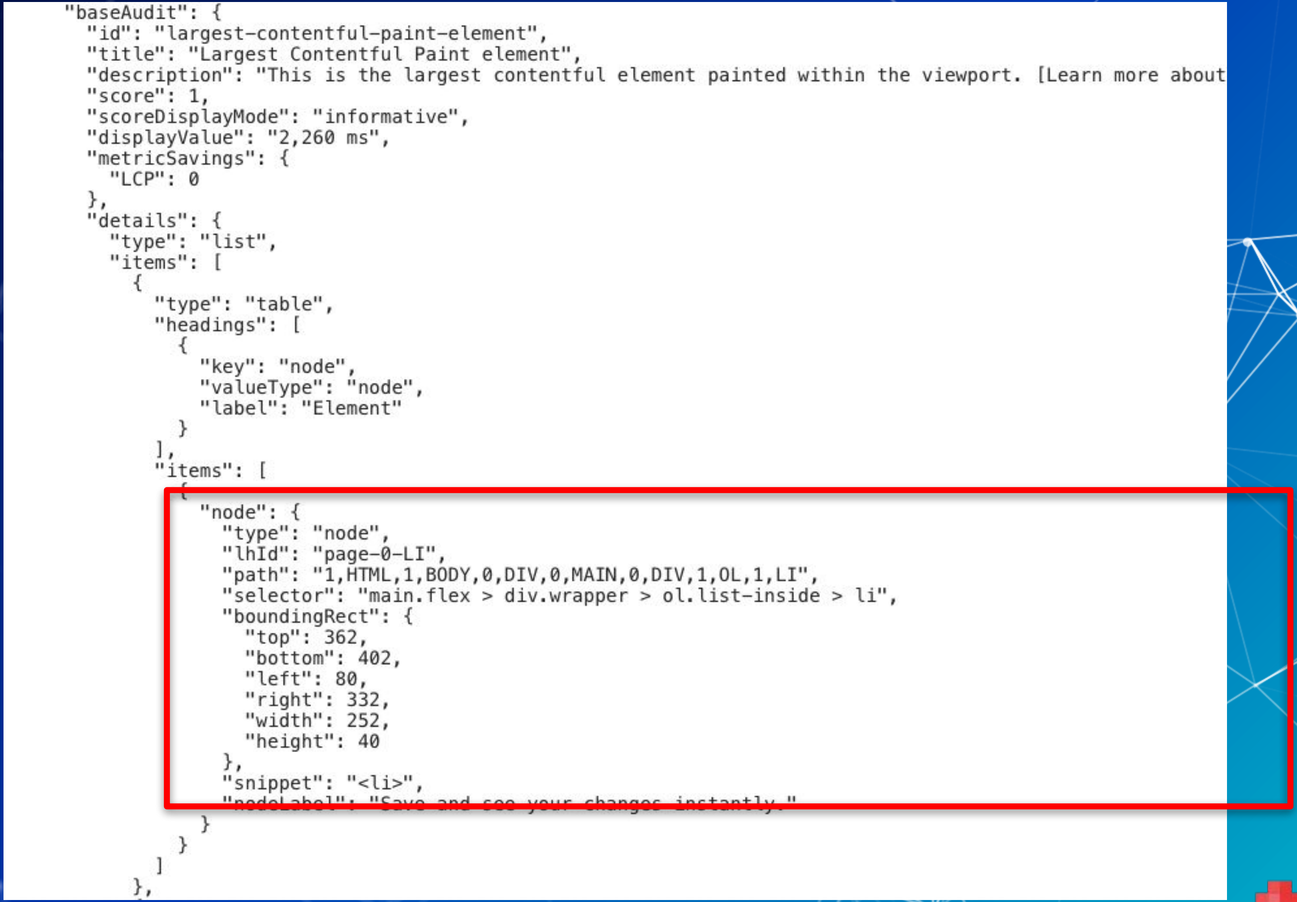
△ Largest Contentful Paint element

This is the largest contentful element painted within the viewport. Learn more about the Largest Contentful Paint element

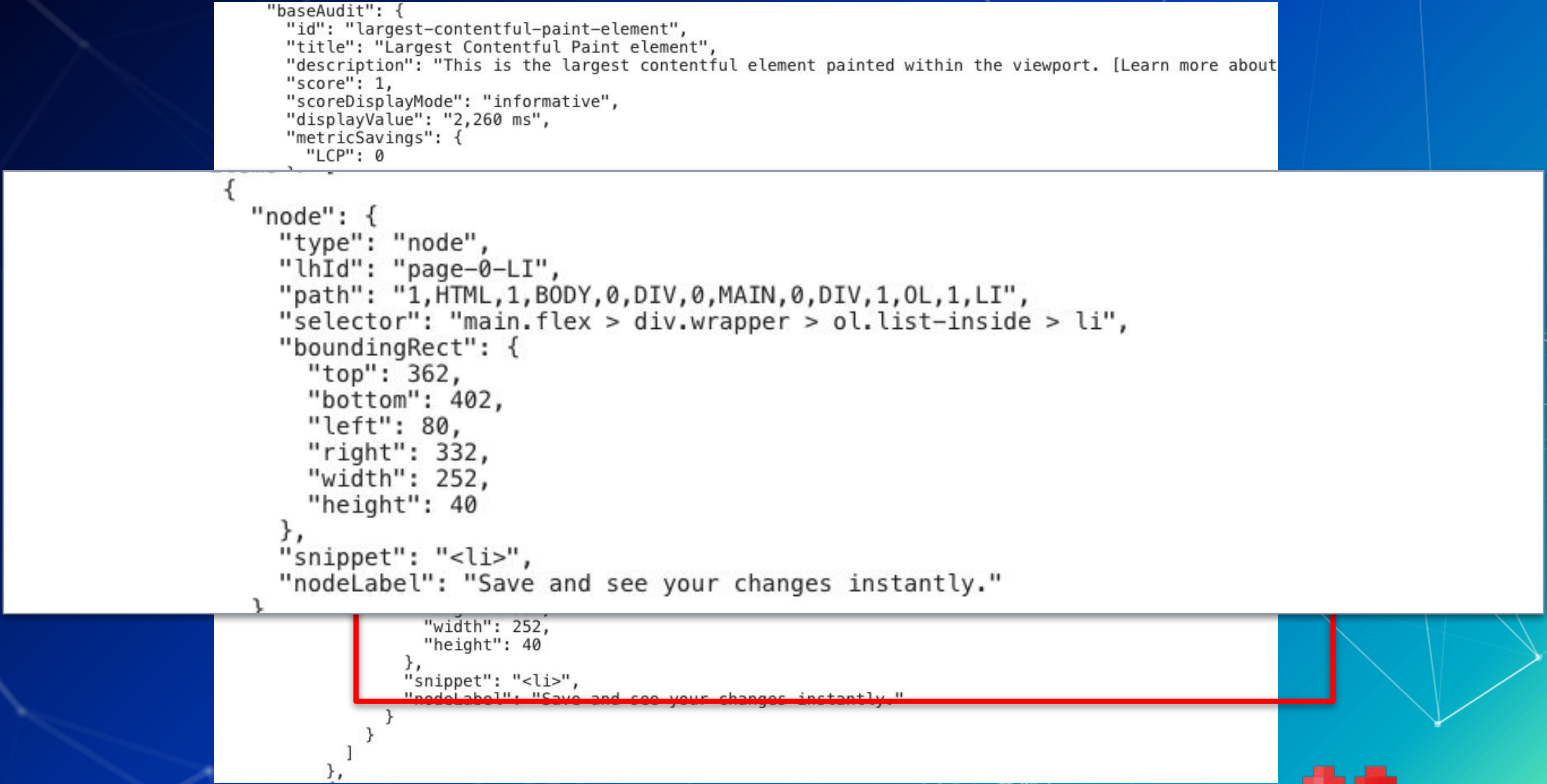
```
{
  "audit": {
```

```
],
"items": [
  {
    "node": {
      "type": "node",
      "lhId": "page-0-IMG",
      "path": "1,HTML,1,BODY,0,DIV,0,MAIN,3,IMG",
      "selector": "body.__variable_1e4310 > div.grid > main.flex > img",
      "boundingRect": {
        "top": 413,
        "bottom": 935,
        "left": 32,
        "right": 380,
        "width": 348,
        "height": 522
      },
      "snippet": "<img src=\"https://images.pexels.com/photos/1624496/pexels-photo-1624496.jpeg
      "nodeLabel": "body.__variable_1e4310 > div.grid > main.flex > img"
    }
  }
```

```
"boundingRect": {
  "top": 413,
  "bottom": 935,
  "left": 32,
  "right": 380,
  "width": 348,
  "height": 522
},
"snippet": "<img src=\"https://images.pexels.com/photos/1624496/pexels-photo-1624496.jpeg
"nodeLabel": "body.__variable_1e4310 > div.grid > main.flex > img"
}
}
```



```
    "baseAudit": {
      "id": "largest-contentful-paint-element",
      "title": "Largest Contentful Paint element",
      "description": "This is the largest contentful element painted within the viewport. [Learn more about
      "score": 1,
      "scoreDisplayMode": "informative",
      "displayValue": "2,260 ms",
      "metricSavings": {
        "LCP": 0
      },
      "details": {
        "type": "list",
        "items": [
          {
            "type": "table",
            "headings": [
              {
                "key": "node",
                "valueType": "node",
                "label": "Element"
              }
            ],
            "items": [
              {
                "node": {
                  "type": "node",
                  "lhId": "page-0-LI",
                  "path": "1,HTML,1,BODY,0,DIV,0,MAIN,0,DIV,1,OL,1,LI",
                  "selector": "main.flex > div.wrapper > ol.list-inside > li",
                  "boundingRect": {
                    "top": 362,
                    "bottom": 402,
                    "left": 80,
                    "right": 332,
                    "width": 252,
                    "height": 40
                  },
                  "snippet": "<li>",
                  "nodeLabel": "Save and see your changes instantly."
                }
              }
            ]
          },
```



```
    "baseAudit": {
      "id": "largest-contentful-paint-element",
      "title": "Largest Contentful Paint element",
      "description": "This is the largest contentful element painted within the viewport. [Learn more about
      "score": 1,
      "scoreDisplayMode": "informative",
      "displayValue": "2,260 ms",
      "metricSavings": {
        "LCP": 0
```

```
{
  "node": {
    "type": "node",
    "lhId": "page-0-LI",
    "path": "1,HTML,1,BODY,0,DIV,0,MAIN,0,DIV,1,OL,1,LI",
    "selector": "main.flex > div.wrapper > ol.list-inside > li",
    "boundingRect": {
      "top": 362,
      "bottom": 402,
      "left": 80,
      "right": 332,
      "width": 252,
      "height": 40
    },
    "snippet": "<li>",
    "nodeLabel": "Save and see your changes instantly."
  }
```

```
              "width": 252,
              "height": 40
            },
            "snippet": "<li>",
            "nodeLabel": "Save and see your changes instantly."
          }
        }
      ]
    },
```

"But Estela, [we don’t use Github / our project is too complex and uses several repositories].

How can we apply Web Performance Testing?

It depends...

Lighthouse-cli

sitespeed.io

WebPageTest by catchpoint

SpeedCurve

DebugBear

And many others...

Key Takeaways

- Web Performance Testing helps prevent degradation in your deployments to Prod.
- Web Performance Testing reduces the time spent troubleshooting unnoticed web performance issues (and improves DX).
- Web Performance Testing can be integrated into the web development process with LHCI.
- LHCI Server is a great tool to help you visualize trends and identify potential problems when degradation is observed.

Thank you!

- estelafranco.com
- toot.cafe/@guaca
- guaca
- *guaca (I'm trying to quit!)*

we ❤ speed