

In Depth Guide to Postgres' new CI

Nazir Bilal Yavuz
Open Source PG Team
Microsoft

About Me

- Nazir Bilal Yavuz
- Open Source Postgres Team @Microsoft
- Working on Postgres ~2 years
- Mainly build and CI stuff

What is CI

- Stands for Continuous Integration
- Automating the integration of new code changes
 - Build
 - Test
- Main benefits
 - Find bugs quicker
 - Fasten the development cycle

Why we need CI on Postgres

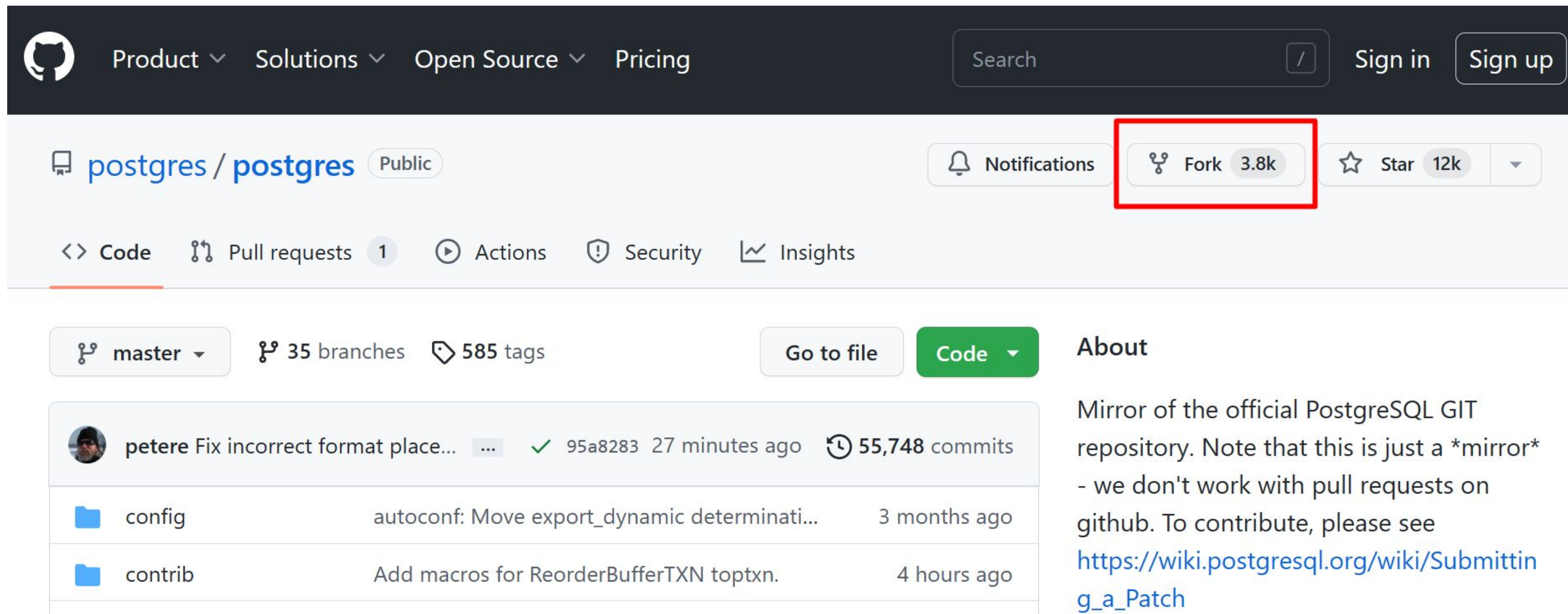
- Patch is sent but...
 - Your changes are not working on...
- Testing changes on various platforms
- Developer productivity
 - Find bugs fast
 - Shorter review period

New CI system

- Uses Cirrus CI (<https://cirrus-ci.org/>)
 - Continuous integration and deployment platform that enables developers to automate their software development workflows
- Steps to run CI:
 - Enable CI
 - Push changes
 - See results
 - PG16

1 - Fork the Postgres repository

- <https://github.com/postgres/postgres>



The screenshot shows the GitHub repository page for `postgres/postgres`. The repository is public and has 3.8k forks and 12k stars. The `Fork` button is highlighted with a red box. Below the repository name, there are tabs for `Code`, `Pull requests` (1), `Actions`, `Security`, and `Insights`. The `Code` tab is selected. Below the tabs, there are buttons for `master` (35 branches), `585 tags`, `Go to file`, and `Code`. The `About` section on the right explains that this is a mirror of the official PostgreSQL GIT repository and provides a link to the PostgreSQL wiki for submitting patches.

Product ▾ Solutions ▾ Open Source ▾ Pricing

Search / Sign in Sign up

postgres / postgres Public

Notifications Fork 3.8k Star 12k

<> Code Pull requests 1 Actions Security Insights

master ▾ 35 branches 585 tags Go to file Code ▾

petere Fix incorrect format place... 95a8283 27 minutes ago 55,748 commits

config	autoconf: Move export_dynamic determinati...	3 months ago
contrib	Add macros for ReorderBufferTXN toptxn.	4 hours ago

About

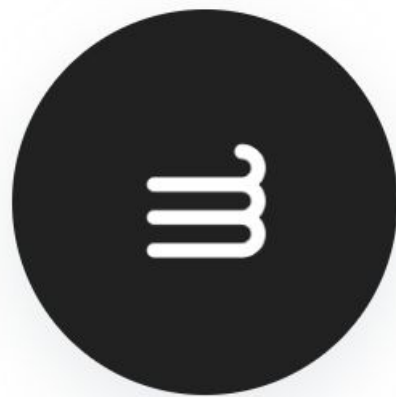
Mirror of the official PostgreSQL GIT repository. Note that this is just a *mirror* - we don't work with pull requests on github. To contribute, please see https://wiki.postgresql.org/wiki/Submittin_g_a_Patch

2 - Add Cirrus CI application to your GitHub account

- <https://github.com/marketplace/cirrus-ci>



[Marketplace](#) / [Apps](#) / Cirrus CI



Application

Cirrus CI

Set up a free trial

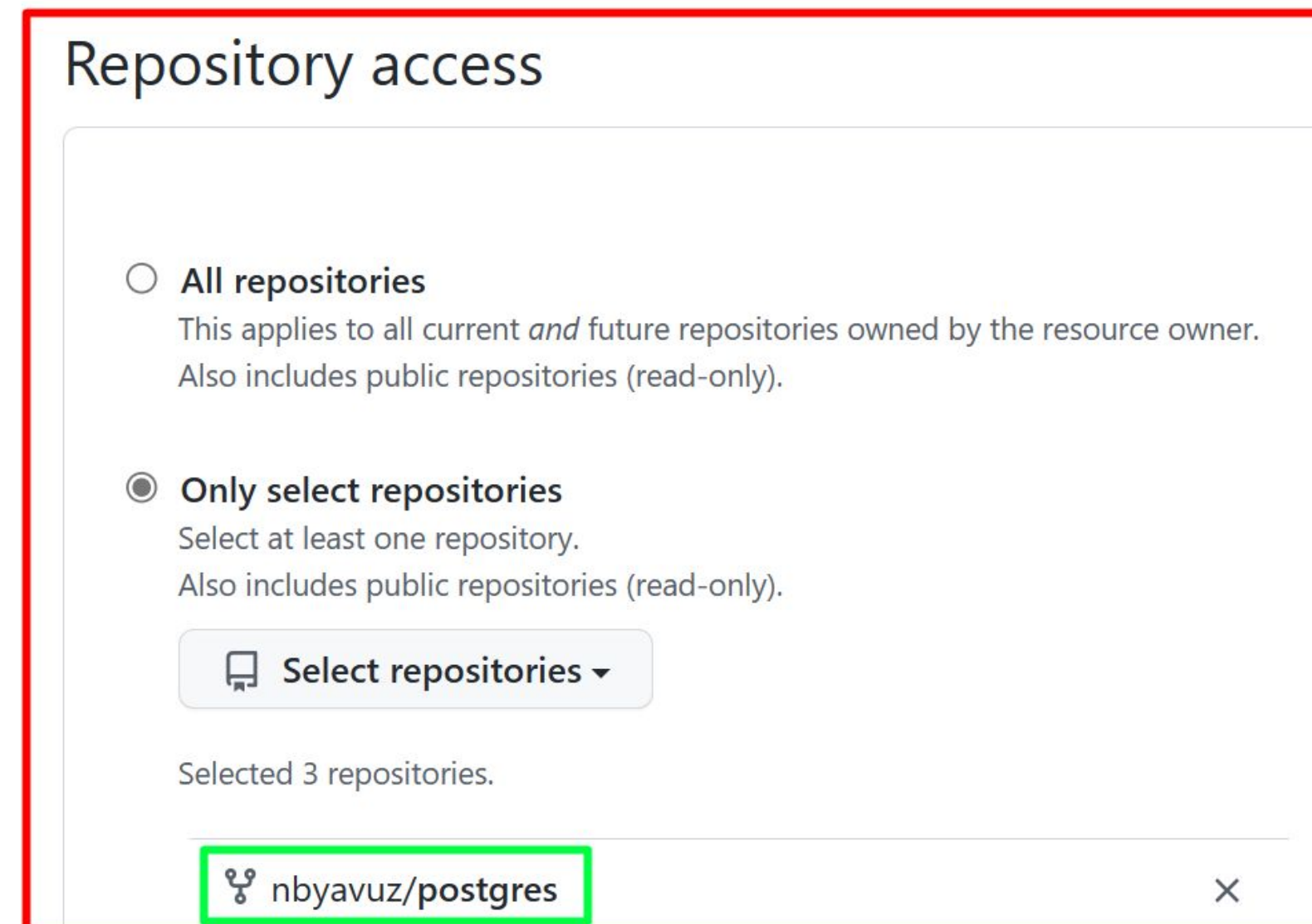
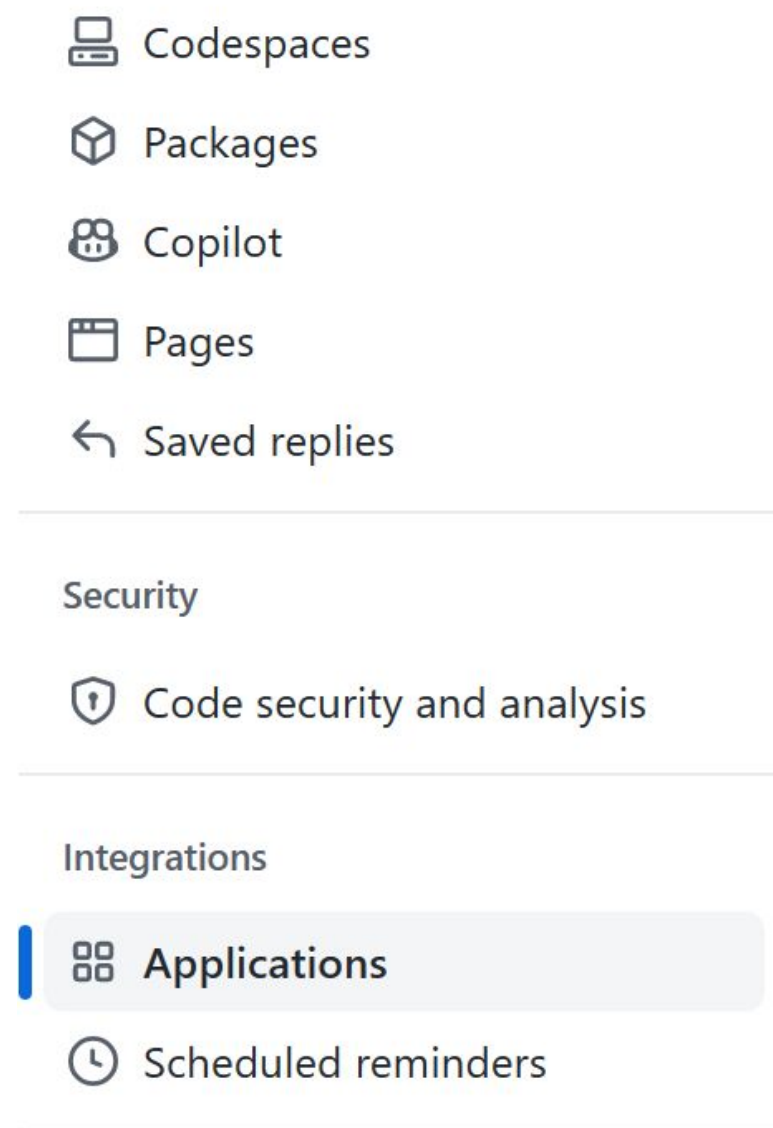
✓ GitHub has verified that the publisher controls the domain and meets other [requirements](#).

Cirrus CI makes your development cycle fast, efficient, and secure by leveraging modern cloud technologies. Cirrus CI scales with your team and makes shipping software faster and cheaper.

[Read more...](#)

3 - Cirrus CI should access to forked PG repository

- GitHub -> Settings -> Applications -> Cirrus CI



See results

Cirrus CI

postgres > postgres > master > Build for 442f870

Created 19 hours ago

Failed in 05:39

Integrate superuser check into has_rolreplication()

Commit [442f870](#) on branch [master](#).

TASKS (8)

HOOKS (0)

<div></div> <div>00:37</div> <div>SanityCheck</div> <div></div>
<div></div> <div>08:23</div> <div>FreeBSD - 13 - Meson</div> <div></div>
<div></div> <div>07:11</div> <div>Linux - Debian Bullseye - Autoconf</div> <div></div>
<div></div> <div>05:02</div> <div>Linux - Debian Bullseye - Meson</div> <div></div>
<div></div> <div>08:00</div> <div>macOS - Ventura - Meson</div> <div></div>
<div></div> <div>12:06</div> <div>Windows - Server 2019, VS 2019 - Meson & ninja</div> <div></div>

How CI works

- It reads **.cirrus.yml** file
 - <https://github.com/postgres/postgres/blob/master/.cirrus.yml>
- Explain in detail later

CI tips

- Cirrus terminal
- Gathering logs
- on_failure instruction
 - instruction
 - I will refer it as a key on yaml file

Cirrus terminal

- When Postgres is working on your local but you see failures on CI
- No need to run CI again



Gathering logs

- *_artifact instruction
- Analyze logs offline
- By default log, diff, regress logs
- Need to gather other files (e.g. docs)

```
testrun_artifacts:  
  paths:  
    - "build*/testrun/**/*.log"  
    - "build*/testrun/**/*.diffs"  
    - "build*/testrun/**/regress_log_*"   
  type: text/plain
```



on_failure instruction

- Only when we come across a failure
- Avoid unnecessary logging
- Faster CI runs for successful build

```
on_failure:  
  testrun_artifacts:  
    paths:  
      - "build*/testrun/**/*.log"  
      - "build*/testrun/**/*.diffs"  
      - "build*/testrun/**/regress_log_*"   
    type: text/plain
```

Let's debug!

.cirrus.yml file

- 6 main parts
- Basic example

In Summary

- Enable new CI on GitHub
- Push changes to trigger CI run
 - CI reads .cirrus.yml file
- See the results
- If debugging is needed
 - Use Cirrus terminal
 - Gather & investigate logs
 - Use on_failure if needed

Thank you!

Questions?

Discord

Comments

naziryavuz@microsoft.com

Tweet to me [@nbyavuz](https://twitter.com/nbyavuz)