CI for CADR

Synopsis

Currently CADR lacks Continuous Integration. As of current investigation, we decide to use a self-hosted runner at GitHub Actions using VMs from AWS or Azure, which will implement the Debian 10 environment for testing successfully. If we decide to use other CI platforms later during the project period, Jenkins or GitLab CI/CD will also be a valid choice.

Project Plan

Current Status

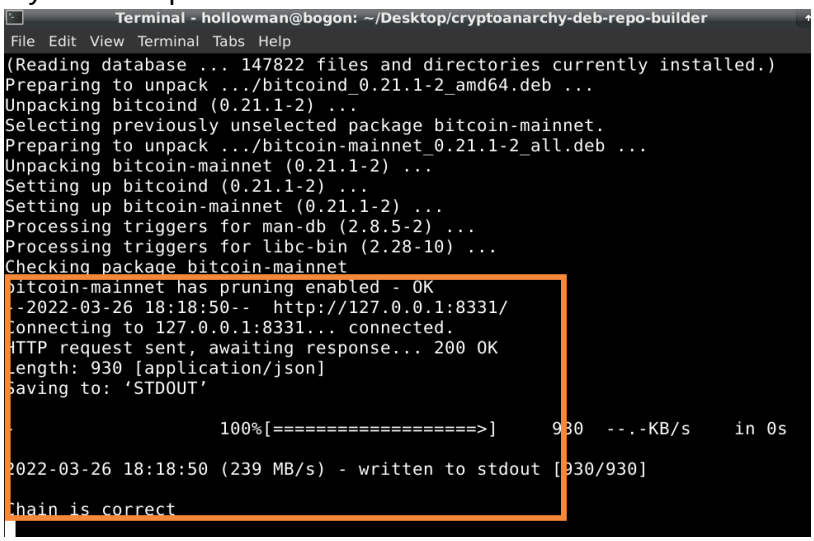
As of now, a demo for CI using the GitHub Actions has already been worked out. https://github.com/debian-cryptoanarchy/cryptoanarchy-deb-repo-builder/pull/198 The demo CI can be triggered manually, or by sending PRs as well as pushing directly to the master branch. The demo CI first builds the running podman environment, then with the podman environment, builds the CADR packages, and then uploads the built Debian packages to Artifacts. Finally test the CADR just built. However, tests in the CI have failed for lacking some dependencies. Moreover, GitHub Actions can do the CI with the containers' help, but the default runner has an issue that its disk space is too small to satisfy our test needs (Although it's OK to just test the regtest data).

Road of Discovery

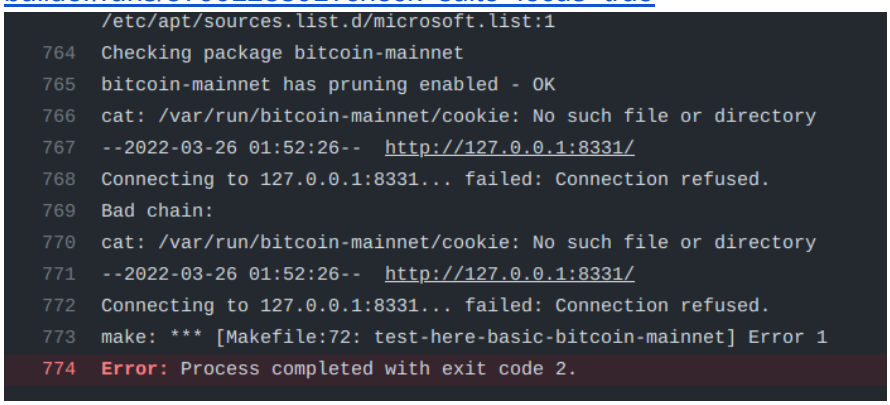
At first, I ignored the fact that there already exists a dockerfile for CADR running (although it was not for building) https://github.com/debian-cryptoanarchy/cryptoanarchy-deb-repo-builder/tree/master/docker , and setup my own dockerfile from scratch by adding dependencies when I encounter any errors. https://github.com/debian-cryptoanarchy/cryptoanarchy-deb-repo-builder/commit/1580e9e025f7ccb597b35753e004a7d5327f10ef

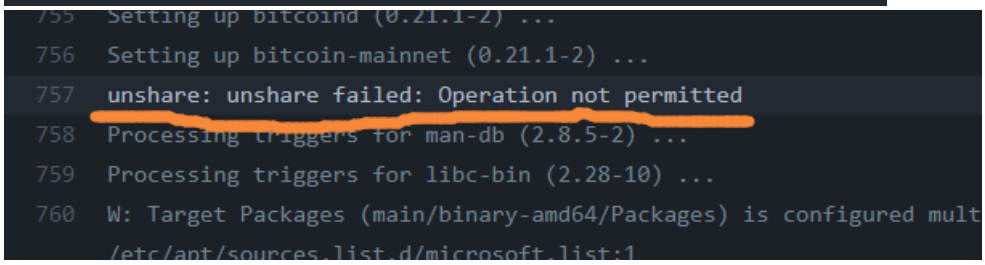
My own dockerfile turned out to work fine on GitHub Actions for the building process, but failed for the test process. https://github.com/HollowMan6/build-cryptoanarchy-deb-repo-builder/actions/runs/2042963652

Initially I thought it was some more dependency issues, since the test can work on my own computer!



When checking the logs I found that it's due to the unshare issue.
https://github.com/HollowMan6/build-cryptoanarchy-deb-repo-builder/runs/5700123801?check_suite_focus=true





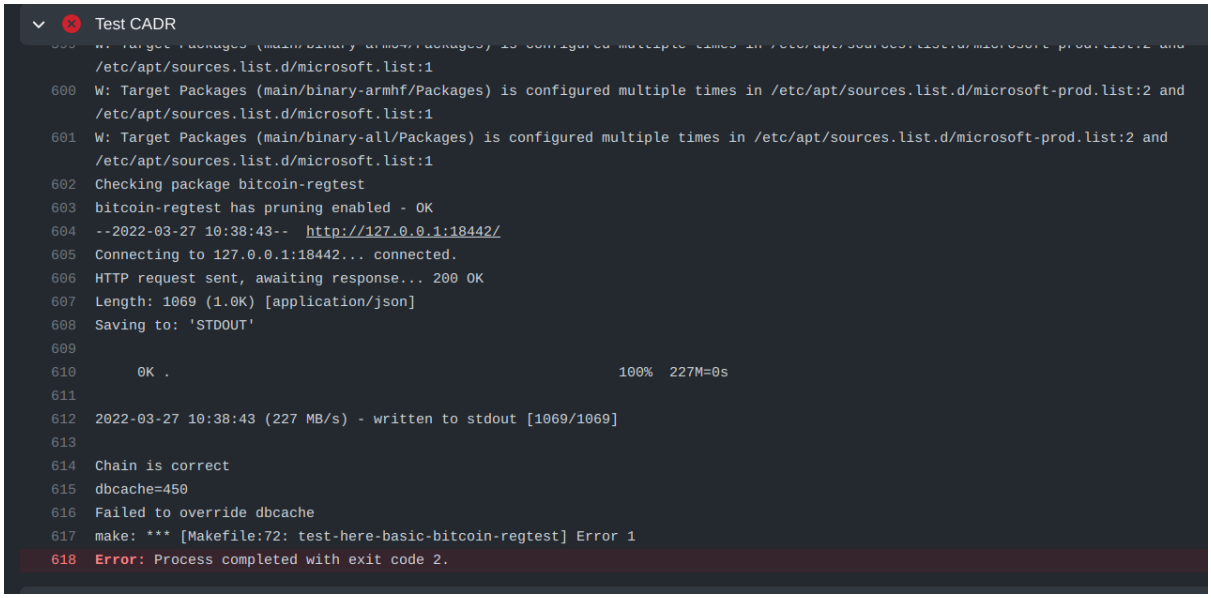
Then I noticed that adding a --privileged parameter can fix the unshare issue for docker. But then a systemd issue just came after it. Finally I noticed the dockerfile in the codebase that already exists for CADR, and just as how it works, I made the systemd to run as the first programme.

```
VOLUME [ "/sys/fs/cgroup" ]
CMD ["/lib/systemd/systemd"]
```

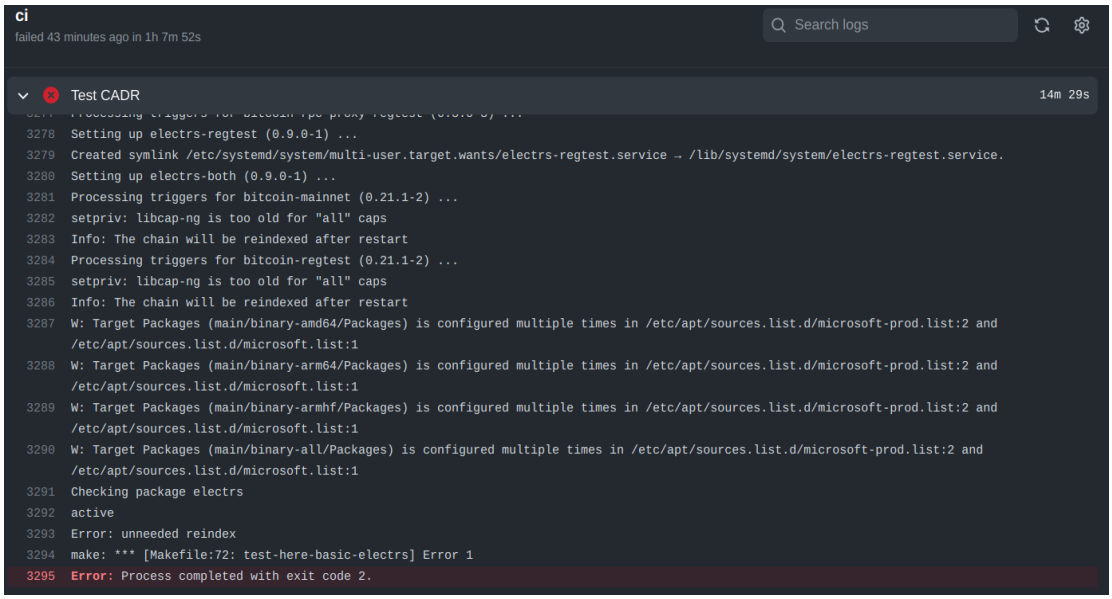
and add `--tmpfs /tmp --tmpfs /run --tmpfs /run/lock -v /sys/fs/cgroup:/sys/fs/cgroup:ro`
as additional parameters to share some host machine resources and make it as the daemon container. Operations are done using docker exec to attach to the container. But still it doesn't fix the systemd issue.

Finally I switched to podman and the systemd issue got fixed (it supports --systemd=true) because I occasionally found this article https://developers.redhat.com/blog/2019/04/24/how-to-run-systemd-in-a-container when I Googled the issue.

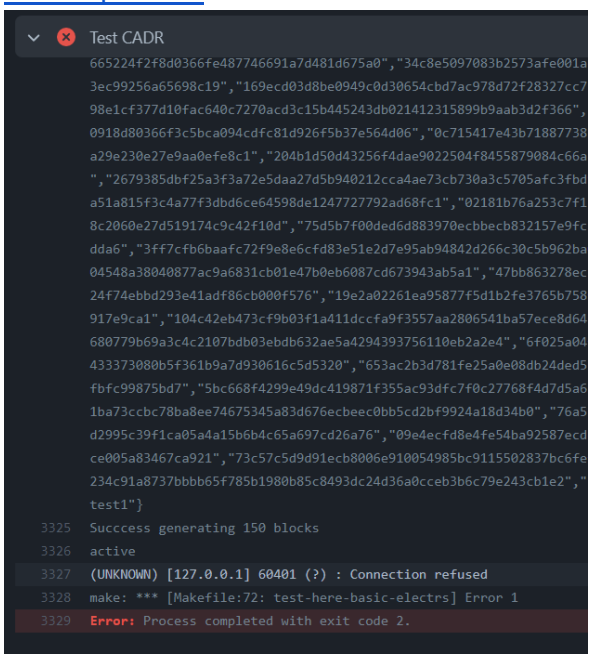
However, a new issue occurs https://github.com/HollowMan6/build-cryptoanarchy-deb-repo-builder/runs/5708913219?check_suite_focus=true suggesting failed to override dbcache.



Can't see any errors from the workflow logs https://github.com/HollowMan6/build-cryptoanarchy-deb-repo-builder/runs/5710646832?check_suite_focus=true , even though the missing bc dependency has been fixed and I can confirm that the command here https://github.com/debian-cryptoanarchy/cryptoanarchy-deb-repo-builder/blob/master/pkg_specs/bitcoin-@variant.sps#L88 works correctly in the container running on my PC. Moreover when I manually skip the test just mentioned, the test further shows that there is an unneeded reindex https://github.com/HollowMan6/build-cryptoanarchy-deb-repo-builder/runs/5710055525?check_suite_focus=true,



which suggests that it's related to issue 108 https://github.com/debian-cryptoanarchy/cryptoanarchy-deb-repo-builder/issues/108 maybe? Have opened an issue here: https://github.com/debian-cryptoanarchy/cryptoanarchy-deb-repo-builder/issues/202 I can skip that test with this commit https://github.com/HollowMan6/build-cryptoanarchy-deb-repo-builder/commit/4defba982028724de70b9ac26d27a59cb150c4c6 , but further error suggesting electrs is not available https://github.com/HollowMan6/build-cryptoanarchy-deb-repo-builder/runs/5713299561?check_suite_focus=true Can be fixed with this PR https://github.com/debian-cryptoanarchy/cryptoanarchy-deb-repo-builder/pull/203



Even if those issues are skipped, another platform will still be needed since the full test requires too much space https://github.com/HollowMan6/build-cryptoanarchy-deb-repo-builder/runs/5713729925?check_suite_focus=true

Only testing the regtest part will be fine with no space issue. https://github.com/HollowMan6/build-cryptoanarchy-deb-repo-builder/runs/5750416035?check_suite_focus=true

Future Plans

Find all dependencies for the tests to run smoothly.

In order to speed up the CI process, as well as use other alternative platforms, the dockerfile in the demo can also be moved to debcrafter such that whenever

debcrafter has a new push, a docker (podman) image can be persistently created for running CADR CI.

The demo mentioned above then can pull the docker (podman) image directly from the docker registry and thus don't need to rebuild it every time the CI runs.

Learn how to create GitHub self-hosted runners on AWS or Azure.

Other CI platforms can be considered. If we decide to use Jenkins, we can specify to use the docker (podman) image as the running environment https://www.jenkins.io/doc/book/pipeline/docker/ and it will be pretty much like GitHub Actions but with more flexibility. The GitLab CI/CD will also work just like Jenkins https://docs.gitlab.com/ee/ci/pipelines/pipeline_architectures.html by specifying the custom docker (podman) images. You can also check one of my projects that uses python docker (podman) image for GitLab CI/CD as an example: https://gitlab.com/HollowMan6/LZU-Auto-COVID-Health-Report/-/blob/main/.gitlab-ci.yml#L1

Jenkins support Jenkins agents, https://www.jenkins.io/doc/book/using/using-agents/ while GitLab CI/CD support GitLab Runner https://docs.gitlab.com/runner/ , which are all equivalent to GitHub self-hosted runners.

More CI platforms will be investigated during the project period, and will compare the pros and cons in detail, then finally choose one platform to implement.

Project Timeline

May 23, 2022 - Jun 5, 2022: Find the missing dependencies to run tests in the container environment to fix the tests running for the demo mentioned above.
Jun 6, 2022 - Jun 19, 2022: Decide which cloud service provider to use.
Jun 20, 2022 - July 3, 2022: Create the self-host runners for CI on GitHub Actions to run successfully.
July 4, 2022 - July 8, 2022: Mid-term evaluations.
July 11, 2022 - July 24, 2022: Write CI on Jenkins and GitLab CI/CD, check if they need self-host runners. If needed, create the runners and check their effectiveness.
July 25, 2022 - August 7, 2022: Also Investigate other possible CI platforms, Write CI on those platforms and check their effectiveness, whether they need self-hosted runners, form a detailed report. Finally decide which CI to use by cost-effectiveness.
August 8, 2022 - August 14, 2022: Time for documentation as well as any lacks.
August 15, 2022 - August 22, 2022: : Final evaluations.

Future Deliverables

I can help resolve issues as much as I can, pack more packages, write more documentations or tests so that eventually help stabilize CADR.

Benefits to Community

With CI the development for CADR will be much easier, any potential developers can now contribute directly to the repo and the reviewer can check the building result of

CI to determine whether the PR is valid, thus relieving the workload of maintainers a lot. With CI, new contributors will also feel comfortable as I learnt that many of them still don't know how to build CADR even with detailed documentation.

Biographical Information

I was the participant of the Google Summer of Code 2021 under openSUSE for Customize IBus project https://summerofcode.withgoogle.com/archive/2021/projects/6295506795364352.
This year I would like to apply for Summer of Bitcoin 2022 at Cryptoanarchy Debian since I have great interest to Bitcoin now, as well as I was too late when I showed my interest for becoming a mentor at openSUSE and the openSUSE doesn't encourage students to apply again under the same organization for GSoC 2022.
I continue to contribute at openSUSE even when GSoC 2021 ends, and because of my great contribution to openSUSE as the project maintainer of my GSoC 2021's work, I eventually become a member at openSUSE.
https://en.opensuse.org/User:Hollowman

I've already contributed to CADR with six PRs and one issue.

For the PR, the first one helps set bitcoind log file under /var/log:
https://github.com/debian-cryptoanarchy/cryptoanarchy-deb-repo-builder/pull/196 , which helps resolve https://github.com/debian-cryptoanarchy/cryptoanarchy-deb-repo-builder/issues/194

The second one helps set bitcoind nosettings enabled by default:
https://github.com/debian-cryptoanarchy/cryptoanarchy-deb-repo-builder/pull/197 , which helps resolve https://github.com/debian-cryptoanarchy/cryptoanarchy-deb-repo-builder/issues/193

The third one helps downgrade rustyline to 6.3.0 to fix compiling issue:
https://github.com/debian-cryptoanarchy/cadr-guide/pull/14

The forth one helps to remove bc dependency for overriding dbcache
https://github.com/debian-cryptoanarchy/cryptoanarchy-deb-repo-builder/pull/200 , which helps resolve https://github.com/debian-cryptoanarchy/cryptoanarchy-deb-repo-builder/issues/199

The fifth one helps to change all apt into apt-get in test
https://github.com/debian-cryptoanarchy/cryptoanarchy-deb-repo-builder/pull/201

The sixth one helps to fix the test connecting to electrs-regtest
https://github.com/debian-cryptoanarchy/cryptoanarchy-deb-repo-builder/pull/203

The issue reports a test failed for bitcoin-regtest with error unneeded reindex for test-here-basic-electrs: https://github.com/debian-cryptoanarchy/cryptoanarchy-deb-repo-builder/issues/202

I have lots of experience writing for GitHub Actions as well as GitLab CI/CD, you can check the list below for my projects and contributions related:
https://github.com/HollowMan6/mdbook-pdf/tree/main/.github/workflows
https://gitlab.com/HollowMan6/LZU-Auto-COVID-Health-Report/-/blob/main/.gitlab-ci.yml
https://gitlab.com/HollowMan6/Wechat-Timed-Message/-/blob/main/.gitlab-ci.yml
https://github.com/HollowMan6/Wechat-Timed-Message/tree/main/.github/workflows
https://github.com/marketplace/actions/send-message-to-wechat
https://gitlab.com/HollowMan6/Site-Notifications/-/blob/main/.gitlab-ci.yml
https://github.com/HollowMan6/Site-Notifications/tree/main/.github/workflows
https://github.com/VowpalWabbit/vowpal_wabbit/pull/3816

I've also participated in the GitHub Actions Hackathon 2021:
https://dev.to/hollowman6/github-issues-to-wechat-4klk

As well as Hacktoberfest 2021 related to GitHub Actions:
https://github.com/commercetools/commercetools-sync-java/pull/789
https://github.com/asyncapi/.github/pull/84

I also know some Rust, I have created a backend for mdBook written in Rust for generating PDF based on headless chrome and Chrome DevTools Protocol https://github.com/HollowMan6/mdbook-pdf as well as contribute to mdBook using Rust https://github.com/rust-lang/mdBook/pull/1738 .

I master Python and NodeJS, you can find many of my personal project written in those two languages: https://github.com/HollowMan6?tab=repositories

I have experience in various Linux packaging. For debian packaging, I have packed gnome-shell-extension-customize-ibus and set up a distribution repo for my GSoC 2021 project IBus Customization: https://github.com/openSUSE/Customize-IBus/tree/main/deb/debian
https://github.com/openSUSE/Customize-IBus/blob/main/Makefile#L127-L128
https://github.com/openSUSE/Customize-IBus/tree/package-repo#debianubuntu

Looking back into my undergraduate life, when I entered the honors class (Specialized Class for Fundamentals and Theories) at Lanzhou University majoring in Computer Science at first, I began to develop an enthusiasm for open source. I set up my own GitHub account (@HollowMan6). As I was dabbling in different realms to enrich my knowledge related to Computer Science, whenever I wrote some useful programs, I would set up a repository and post it on GitHub. I appreciate the open-source spirit and devote much time to contributing at GitHub. I have contributed to many notable projects and organizations, and NASA Mars 2020 Helicopter Mission used some projects that I have contributed. Many of my personal projects had been archived in the 2020 Arctic Code Vault Program by GitHub. My activities in open-source communities have also helped me win Linux Foundation Training (LiFT) Scholarship (China) in the Academic Aces category. Now I have 1.5k followers and over 500 personal project stars on GitHub.

I have also took global open-source internships four times in total, separately funded by Alibaba Summer of Code 2020 https://github.com/HollowMan6/My-Alibaba-Summer-of-Code#my-alibaba-summer-of-code, Google Summer of Code 2021 https://github.com/HollowMan6/My-Google-Summer-of-Code#my-google-summer-of-code , and the Institute of Software, Chinese Academy of Sciences (twice) for Open Source Promotion Plan - Summer 2020 & 2021 https://github.com/HollowMan6/My-OSPP-Summer#my-open-source-promotion-plan---summer . Some details can also be found on my CV. The most memorable one to me is the Google Summer of Code. The project I did there was to improve the UI usability of IBus. IBus is a framework popular in Asia that integrates with Linux to enable users who speak non-Latin languages to input. Google Summer of Code is a competitive program where tons of students with various project ideas are actively applying. To stand out among the applicants for application, I must be very promising to show that I could finish the project excellently. Although I had no earlier experience developing the GNOME Shell extension, I still want to do the project to benefit Linux users in Asia. Then during the application month, I devoted most of my free time to studying the IBus and GNOME Shell codebase, referring to their documentation, and successfully made a demo. It was a challenging time indeed, but it is worthwhile. Google eventually decided to sponsor my project during the program.

Now for the Summer of Bitcoin 2022, it's my first time to learn and develop for Bitcoin related stuff, so if I get selected, I will definitely gain lots of knowledge related to Bitcoin and contribute in good quality to the Cryptoanarchy Debian community both during and after the project period. Looking forward to hearing the good news.

Hollow Man