

Crystal in the Cloud: A Cloud Native Journey



Crystal 1.0 Conference – Thursday, July 8th



Taylor Carpenter
taylor@vulk.coop



Will Harris
will@vulk.co

Crystal in the Cloud: A cloud-native journey

Abstract: As part of an initiative to build a testing framework and suite of tests that help end-users to check usage of cloud native practices we chose Crystal as the language for the project. In this talk we will give an overview of the project and the networking+telecommunication domain we are working in. We will review the implementation of the CNF Test Suite. We will also discuss how we see Crystal has good alignment with cloud native and K8s development, such speeding up development velocity, reducing risk, and life-cycle dependency management and why we chose it.



Today's journey (25 minutes)

- Intro to team & CNF Test Suite project
- Going with Crystal
- Wins and obstacles with crystalizing CNF Test Suite
- Get involved
- Q&A



About us



Vulk Coop

- **Worker-owned software cooperative**
- Since 2013 in Austin, Texas USA
- [Austin Software Co-operative Meetup](#)
- **Twitter:** @vulkcoop
- **Email:** howl@vulk.coop



About us



Taylor Carpenter

- **CNF Working Group Co-Chair**
- Member, Vulk Coop in Austin, TX
- 20+ years of devops + development experience
 - Telecom, Finance, Healthcare
- Open Source advocate, using Linux since 1994
- Helping CNCF with CI/CD and Cloud Native Telecom initiatives since 2017
- **Twitter:** @ixx
- **GitHub:** @taylor



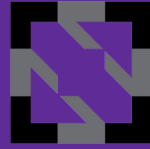
About us



Will Harris

- **CNF Test Suite Maintainer**
- Owner <https://will.vc/> based in Miami, FL
- 10+ years of experience building software at scale for Tech Giants (ex-LinkedIn), Venture backed startups, and Open Source Projects
- Helping CNCF with Cloud Native Telecom initiatives since 2019
- **Twitter:** @boujeehacker
- **GitHub:** @williscool





CLOUD NATIVE
COMPUTING FOUNDATION

CNCF's Telecom Industry Initiatives



What is CNCF?

The **Cloud Native Computing Foundation** (aka CNCF) is an organization under Linux Foundation focused on making cloud native computing ubiquitous for everyone.



**CLOUD NATIVE
COMPUTING FOUNDATION**

Learn more at <https://www.cncf.io/>



What is cloud native?

Set of practices, principles, and technology

- Loosely coupled
- Service discovery
- Microservices
- Coarse-grained packaging
- Designed for automation
- Declarative consumption model
- Immutable infrastructure



Cloud native benefits

Adopt cloud native technologies and practices to gain:

- **Better resource efficiency** to run the same number of services on fewer servers
- **Improved resiliency and availability** despite failures of individual CNFs, machines, or even data centers
- **Higher development velocity** with reduced risk
- **Interoperability improvements** to help with disaggregation and multi-vendor compatibility



Who are the end users?

Telecommunication and networking community

- Telecom service providers
- Network equipment manufacturers
- Application and platform developers



What's different about Telco software?

There is networking software everywhere

- Existing standards
- Long development cycle
- Low risk tolerance
- Works on my machine



CNF Test Suite

What is the CNF Test Suite?

An open source **test suite** for CNF developers and network operators to evaluate how well a network application, aka **Cloud Native Network Function (CNF)**, follows cloud native principles and best practices.

<https://networking.cloud-native-principles.org/>



Features

- Is **self-contained** with minimal requirements and necessary configuration
- Provides **suggestions** on how to improve a network application to better follow **cloud native best practices**
- Offers CLI help documentation
- Supports self-hosted and protected image repositories
- Supports multiple K8s workload resources and services
- Gives **fair assessment** with a flexible scoring system
 - Tests fail gracefully
 - Tests are skipped when prerequisites are not met
- Facilitates **integration of test results** into upstream systems



Implementation overview

- The test framework and tests are written in Crystal
- CNF Test Suite leverages **upstream tools** where possible such as LitmusChaos, Chaos Mesh, kubectl, dockerd, and Helm linter for testing CNFs
- The upstream tool installation, configuration, and versioning has been made **repeatable**
- Support any vanilla upstream **Kubernetes**



Test categories

Categories help organize the tests for discussion. Tests may exist in more than one category. **Cloud native principles** and capabilities (such as resilience) may apply to more than one category.

- Installable and Upgradeable
- Configuration and Lifecycle
- Hardware support
- Microservice
- Compatible
- State
- Security
- Scaling
- Observability
- Resilience



Current tests

Installable	Configuration	Microservice	Compatibility	State	Security	Scalability	Observability	Resilience
Published Helm chart	Liveness Readiness	Startup time	N/A	Volume HostPath Not Found	Privileged mode	Increase capacity	Traffic to Prometheus	CNF back up when container killed/dies
Valid Helm chart	Rolling version upgrade/downgrade	Image Size		No Local Volume Configuration		Decrease capacity		LitmusChaos Pod Network Loss
Helm Deploy	No nodePort Secrets Used					ClusterAPI enabled		LitmusChaos Pod Network Latency
Install Script Helm	Static: Hardcoded IPs in source code							LitmusChaos Pod CPU Hog
Rolling Update	Immutable Config Maps							LitmusChaos Disk Fill LitmusChaos Pod Delete

Contributors



@taylor



@wavell



@nupejosh



@lixuna



@denver
williams



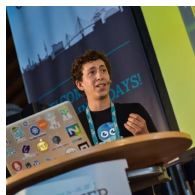
@williscool



@agentpoyo



@HashNuke



@xmulligan



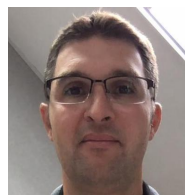
@michaels
pedersen



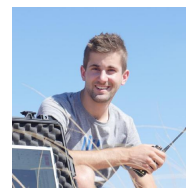
@petorre



@electro
cucaracha



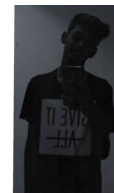
@sishbi



@clementnuss



@uditgaurav



@saksham
gurbhele

See more at <https://github.com/cncf/cnf-testsuite/graphs/contributors>



Going with Crystal - Why we chose Crystal over other languages



What are our language needs?

Why use Crystal in the infrastructure space?

- **Compiled** = Binary distribution
- Wiring JSON APIs together
- Wiring CLI output together



Default / common language choices in CNCF + K8s

CNCF Cloud Native Interactive Landscape



The Cloud Native Trail Map (png, pdf) is CNCF's recommended path through the cloud native landscape. The cloud native landscape (png, pdf), serverless landscape (png, pdf), and member landscape (png, pdf) are dynamically generated below. Please open a pull request to correct any issues. Greyed logos are not open source. Last Updated: 2021-07-06 05:30:18Z

You are viewing 243 cards with a total of 1,529,786 stars, market cap of \$10.5T and funding of \$4.3B.

Landscape Card Mode Serverless Members

Tweet 1567

No Grouping (243)

 agola Agola Sorint.Lab ★ 856	 Microsoft Azure AKS Engine for Azure Stack Microsoft MCap: \$2.1T ★ 4	 ALLUXIO Alluxio Funding: \$23M ★ 5,145	 aws Amazon Elastic Kubernetes Service Distro (Amazon EKS-D) Amazon Web Services MCap: \$1.8T ★ 894	 ANTREA Antrea Cloud Native Computing Foundation (CNCF) Funding: \$3M ★ 1,085	 APACHE Camel Apache Camel K The Apache Software Foundation ★ 582
 Apache Thrift™ Apache Thrift The Apache Software Foundation ★ 8,476	 APISIX APISIX The Apache Software Foundation ★ 5,445	 argo Argo Cloud Native Computing Foundation (CNCF) Funding: \$3M ★ 8,802	 ArtifactHUB Artifact Hub Cloud Native Computing Foundation (CNCF) Funding: \$3M ★ 574	 Athenz Athenz Cloud Native Computing Foundation (CNCF) Funding: \$3M ★ 610	 Microsoft Azure Azure (AKS) Engine Microsoft MCap: \$2.1T ★ 877
 beam	 beats	 BFE	 BOSH	 BRIGADE	 Buildkite

landscape.cncf.io

Go vs Crystal error messaging. (Go version)

Here are the logs I'm getting. It appears to be an issue with the request being passed down to the interceptor.

```
{"level":"info","caller":"logging/config.go:100","msg":"Successfully created the logger.","knative.dev/jsonconfig":  
{"level":"info","caller":"logging/config.go:101","msg":"Logging level set to info"}  
{"level":"info","caller":"logging/config.go:69","msg":"Fetch GitHub commit ID from kodata failed: open /var/run/ko/  
{"level":"info","logger":"eventlistener","caller":"logging/logging.go:46","msg":"Starting the Configuration eventli  
{"level":"info","logger":"eventlistener","caller":"eventlistenersink/main.go:71","msg":"EventListener pod started",  
{"level":"info","logger":"eventlistener","caller":"eventlistenersink/main.go:97","msg":"Listen and serve on port 80  
panic: runtime error: invalid memory address or nil pointer dereference  
[signal SIGSEGV: segmentation violation code=0x1 addr=0x0 pc=0x128e438]
```

```
goroutine 35 [running]:
```

```
github.com/tektoncd/triggers/pkg/interceptors/github.(*Interceptor).ExecuteTrigger(0xc00031e360, 0xc00047a300, 0x0,  
    /workspace/go/src/github.com/tektoncd/triggers/pkg/interceptors/github/github.go:56 +0x3c8  
github.com/tektoncd/triggers/pkg/sink.Sink.executeInterceptors(0x18685c0, 0xc00009e120, 0x180dc20, 0xc0005a4160, 0x  
    /workspace/go/src/github.com/tektoncd/triggers/pkg/sink/sink.go:179 +0x171  
github.com/tektoncd/triggers/pkg/sink.Sink.processTrigger(0x18685c0, 0xc00009e120, 0x180dc20, 0xc0005a4160, 0x7f904  
    /workspace/go/src/github.com/tektoncd/triggers/pkg/sink/sink.go:129 +0x1f9  
github.com/tektoncd/triggers/pkg/sink.Sink.HandleEvent.func1(0x18685c0, 0xc00009e120, 0x180dc20, 0xc0005a4160, 0x7f  
    /workspace/go/src/github.com/tektoncd/triggers/pkg/sink/sink.go:93 +0xd2  
created by github.com/tektoncd/triggers/pkg/sink.Sink.HandleEvent  
    /workspace/go/src/github.com/tektoncd/triggers/pkg/sink/sink.go:92 +0x79d
```



Go vs Crystal error messaging. (Go version)

Here are the logs I'm getting. It appears to be an issue with the request being passed down to the interceptor.

```
{"level":"info","caller":"logging/config.go:100","msg":"Successfully created the logger.", "knative.dev/jsonconfig":  
{"level":"info","caller":"logging/config.go:101","msg":"Logging level set to info"}  
{"level":"info","caller":"logging/config.go:69","msg":"Fetch GitHub commit ID from kodata failed: open /var/run/ko/  
{"level":"info","logger":"eventlistener","caller":"logging/logging.go:46","msg":"Starting the Configuration eventli  
{"level":"info","logger":"eventlistener","caller":"eventlistenersink/main.go:71","msg":"EventListener pod started",  
{"level":"info","logger":"eventlistener","caller":"eventlistenersink/main.go:97","msg":"Listen and serve on port 80  
panic: runtime error: invalid memory address or nil pointer dereference  
[signal SIGSEGV: segmentation violation code=0x1 addr=0x0 pc=0x128e438]
```

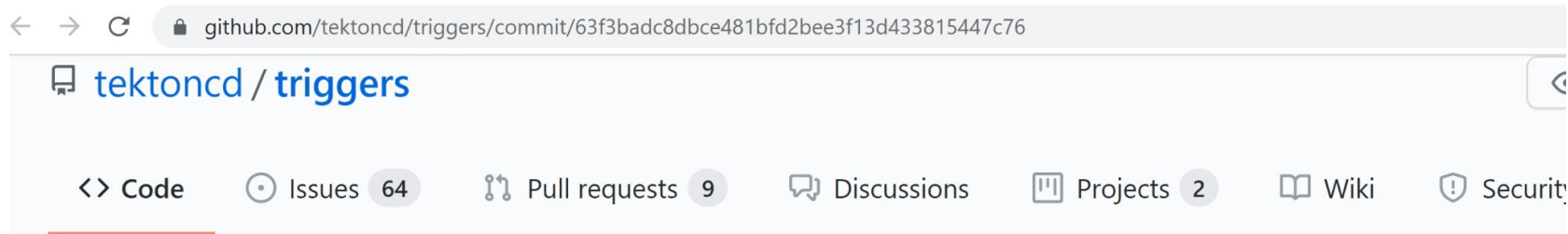
goroutine 35 [running]:

```
github.com/tektoncd/triggers/pkg/interceptors/github.(*Interceptor).ExecuteTrigger(0xc00031e360, 0xc00047a300, 0x0,  
    /workspace/go/src/github.com/tektoncd/triggers/pkg/interceptors/github/github.go:56 +0x3c8  
github.com/tektoncd/triggers/pkg/sink.Sink.executeInterceptors(0x18685c0, 0xc00009e120, 0x180dc20, 0xc0005a4160, 0x  
    /workspace/go/src/github.com/tektoncd/triggers/pkg/sink/sink.go:179 +0x171  
github.com/tektoncd/triggers/pkg/sink.Sink.processTrigger(0x18685c0, 0xc00009e120, 0x180dc20, 0xc0005a4160, 0x7f904  
    /workspace/go/src/github.com/tektoncd/triggers/pkg/sink/sink.go:129 +0x1f9  
github.com/tektoncd/triggers/pkg/sink.Sink.HandleEvent.func1(0x18685c0, 0xc00009e120, 0x180dc20, 0xc0005a4160, 0x7f  
    /workspace/go/src/github.com/tektoncd/triggers/pkg/sink/sink.go:93 +0xd2  
created by github.com/tektoncd/triggers/pkg/sink.Sink.HandleEvent  
    /workspace/go/src/github.com/tektoncd/triggers/pkg/sink/sink.go:92 +0x79d
```



Actual problem:

it's JSON... [tektoncd/triggers#63f3badc](https://github.com/tektoncd/triggers/commit/63f3badc8dbce481bfd2bee3f13d433815447c76)

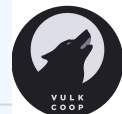


✓ fix interceptor panic when request body is nil

We were passing the original request where the body had been drained to the first interceptor meaning that the body to the first interceptor in the chain was always empty. Further, we were using the `GetBody` which can sometimes be empty leading to a panic.

fixes [#355](#)

Signed-off-by: Dibyo Mukherjee <dibyo@google.com>



Documentation on unhelpful error message

“Invalid memory address or nil pointer derefer



Description



“Invalid memory address or nil pointer dereference” error



Cause



There are a number of reasons, most commonly, an API may have been configured incor



Task management (Go)

<https://taskfile.dev>

Getting started

Create a file called `Taskfile.yml` in the root of your project. The `cmds` attribute should contain the commands of a task. The example below allows compiling a Go app and uses [Minify](#) to concat and minify multiple CSS files into a single one.

```
version: '3'

tasks:
  build:
    cmds:
      - go build -v -i main.go

  assets:
    cmds:
      - minify -o public/style.css src/css
```

Running the tasks is as simple as running:

```
task assets build
```



```
version: '3'

tasks:
  main-task:
    cmds:
      - task: task-to-be-called
      - task: another-task
      - echo "Both done"

  task-to-be-called:
    cmds:
      - echo "Task to be called"

  another-task:
    cmds:
      - echo "Another task"
```

Overriding variables in the called task is as simple as informing `vars` attribute:

```
version: '3'

tasks:
  greet:
    vars:
      RECIPIENT: '{{default "World" .RECIPIENT}}'
    cmds:
      - echo "Hello, {{.RECIPIENT}}!"

  greet-pessimistically:
    cmds:
      - task: greet
      vars: {RECIPIENT: "Cruel World"}
```



Enter Crystal

Wins

(and a few obstacles) with
crystalizing CNF Test Suite



Choosing Crystal

Idioms are more **human friendly**

- You can “have your cake and eat it too” especially for those that are familiar with Ruby from working the DevOps community



Go vs Crystal error messaging. (Crystal version)



[Feature] cnf_setup should install cnfs that use a manifest directory in airgapped mode #808

wavell opened this issue 15 days ago · 6 comments



agentpoyo commented yesterday

Collaborator



While in airgapped mode (no internet):

```
[drew@hercules cnf-testsuite]$ ./cnf-testsuite cnf_setup cnf-config=sample-cnfs/k8s-non-helm/cnf-testsuite.yr
Successfully created directories for cnf-testsuite
cnf setup airgapped mode
The connection to the server localhost:8080 was refused - did you specify the right host or port?
../../../../usr/share/crystal/src/json/parser.cr:125:5 in 'parse_exception'
../../../../usr/share/crystal/src/json/parser.cr:121:5 in 'unexpected_token'
../../../../usr/share/crystal/src/json/parser.cr:37:7 in 'parse_value'
../../../../usr/share/crystal/src/json/parser.cr:13:12 in 'parse'
../../../../usr/share/crystal/src/json.cr:131:5 in 'parse'
src/tasks/utils/kubectl_client.cr:187:7 in 'nodes'
src/tasks/utils/kubectl_client.cr:272:52 in 'schedulable_nodes_list'
src/tasks/utils/airgap.cr:353:12 in 'pods_with_tar'
src/tasks/utils/airgap.cr:188:5 in 'bootstrap_cluster'
src/tasks/utils/airgap.cr:165:5 in 'cache_images'
src/tasks/cnf_setup.cr:21:5 in '->'
../../../../usr/share/crystal/src/primitives.cr:255:3 in 'call'
lib/sam/src/sam/execution.cr:20:7 in 'invoke'
lib/sam/src/sam.cr:35:5 in 'invoke'
lib/sam/src/sam.cr:53:7 in 'process_tasks'
src/cnf-testsuite.cr:104:3 in '__crystal_main'
../../../../usr/share/crystal/src/crystal/main.cr:105:5 in 'main_user_code'
../../../../usr/share/crystal/src/crystal/main.cr:91:7 in 'main'
../../../../usr/share/crystal/src/crystal/main.cr:114:3 in 'main'
__libc_start_main
_start
???.
unexpected token '<EOF>' at 1:1
```



Task management (Crystal)

<https://github.com/imdrasil/sam.cr>

☰ README.md

Tasks are the main unit in `sam.cr`. Task has a name, a list of prerequisites and a list of actions (block of a code).

Sam extends the global context with own DSL. To define a task use `task` method which accepts the task name as the 1st argument.

```
task "name" do
end
```



If you want to define prerequisites, add the array with their names as the 2nd argument:

```
task "name", ["prereq1", "prereq2"] do
end
```

Executing a task

Sam does no magic with your `sam.cr` file - it is just a common `.cr` source file which allows you to recompile it with any possible code you want such amount of times you need. Therefore the most obvious way to execute any task is:

```
$ crystal sam.cr name
```



Package management (Crystal)

```
26 desc "The CNF Test Suite program enables interoperability of CNFs from multiple vendors running on top of Kubernetes supplied by differ
27 task "workload", ["automatic_cnf_install", "ensure_cnf_installed", "configuration_file_setup", "compatibility", "state", "security", "sc
28   VERBOSE_LOGGING.info "workload" if check_verbose(args)
29
30   total = CNFManager::Points.total_points("workload")
31   if total > 0
32     stdout_success "Final workload score: #{total} of #{CNFManager::Points.total_max_points("workload")}]"
33   else
34     stdout_failure "Final workload score: #{total} of #{CNFManager::Points.total_max_points("workload")}]"
35   end
36
37   if CNFManager::Points.failed_required_tasks.size > 0
38     stdout_failure "Test Suite failed!"
39     stdout_failure "Failed required tasks: #{CNFManager::Points.failed_required_tasks.inspect}"
40   end
41   stdout_info "CNFManager::Points::Results have been saved to #{CNFManager::Points::Results.file}".colorize(:green)
42 end
43
```



Successes

57 stars and counting on GitHub

- <https://github.com/cncf/cnf-testsuite>

Crystal helped bigtime

- Leverage Ruby experience
- Crystal is better at doing DSL because of Ruby heritage



CNF Test Suite in action

- <https://asciinema.org/a/DUI1XgA45N4OJ8sKZQ2hinCfn>
- Example CNF based on [CoreDNS](#) (a CNCF graduated project)
 - Modified to demonstrate **passed**, **skipped** and **failed** test results
- Run Workload Tests
 - Duration: 5-6 minutes

```
✓ PASSED: No privileged containers 🚫
Security final score: 5 of 5

✓ PASSED: Replicas increased to 3 🚦
✓ PASSED: Replicas decreased to 1 🚦
Scalability final score: 20 of 35

✓ PASSED: No IP addresses found
✓ PASSED: Helm liveness probe found 🟢
✓ PASSED: Helm readiness probe found 🟢
✓ PASSED: NodePort is not used
✓ PASSED: No hard-coded IP addresses found in the runtime k8s configuration
✓ PASSED: CNF Rollback Passed
No Secret Volumes or Container secretKeyRefs found for resource: (kind: "Deployment", name: "coredns-coredns")
❌ SKIPPED: Secrets not used 🟡

To address this issue please see the USAGE.md documentation

X FAILED: Found mutable configmap(s) 🚫
✓ PASSED: CNF for Rolling Update Passed
✓ PASSED: CNF for Rolling Downgrade Passed
✓ PASSED: CNF for Rolling Version Change Passed
Configuration Lifecycle final score: 40 of 46

✓ PASSED (by default): No install script provided
Successfully created directories for cnf-testsuite
✓ PASSED: Helm Chart exported chart Lint Passed 🚦
X FAILED: Published Helm Chart Not Found 🚫🔴
SKIPPED: Helm Deploy 🟡
Installability final score: 9 of 20

✓ PASSED: Image size is good 🚦🚦🚦
✓ PASSED: CNF had a reasonable startup time 🚦
Microservice final score: 10 of 10

✓ PASSED: pod_network_latency chaos test passed 🚦🚦🚦
✓ PASSED: Application pod is healthy after high CPU consumption 🚦🚦🚦
✓ PASSED: Replicas available match desired count after container kill test 🚦🚦🚦
```



Successes

Accessible code base

- <https://github.com/cncf/cnf-testsuite/blob/main/src/cnf-testsuite.cr>

Better developer experience

- <https://github.com/imdrasil/sam.cr>
<https://taskfile.dev/#/usage?id=getting-started>
- <https://nicedoc.io/oxequa/realize#user-content-quickstart>

Shorter onboarding time for contributors

- An end-user went from reporting an issue to being a contributor and created their first PR in under 2 weeks



Multi-platform support

Support all major operating systems

- Linux
- Windows(wsl)
- MacOS (from source)



Experience w/ upgrading process

Upgrading was (mostly) a breeze

- <https://github.com/cncf/cnf-testsuite/pull/325>

```
--      +-
31      # Download and install Crystal
32      - sudo apt update && sudo apt install -y libevent-dev
33      - wget https://github.com/crystal-
lang/crystal/releases/download/0.33.0/crystal-0.33.0-1-linux-x86_64.tar.gz
34      - tar -xvf crystal-*.tar.gz
35      - export PATH=$(pwd)/crystal-0.33.0-1/bin:$PATH
36      - crystal version
37      # Download and install kubectl
38      - curl -LO https://storage.googleapis.com/kubernetes-
release/release/$(curl -s https://storage.googleapis.com/kubernetes-
release/release/stable.txt)/bin/linux/amd64/kubectl && chmod +x kubectl &&
sudo mv kubectl /usr/local/bin/
@@ -43,6 +43,6 @@ jobs:
43      script:
44      - shards install
45      - crystal spec -v || travis_terminate 1;
46      - docker pull crystallang/crystal:0.32.1-alpine
47      - docker run --rm -it -v $PWD:/workspace -w /workspace
crystallang/crystal:0.32.1-alpine crystal build src/cnf-conformance.cr --
release --static --link-flags "-lxml2 -llzma"
```

```
--      +-
31      # Download and install Crystal
32      - sudo apt update && sudo apt install -y libevent-dev
33      + wget https://github.com/crystal-
lang/crystal/releases/download/0.35.1/crystal-0.35.1-1-linux-x86_64.tar.gz
34      - tar -xvf crystal-*.tar.gz
35      + export PATH=$(pwd)/crystal-0.35.1-1/bin:$PATH
36      - crystal version
37      # Download and install kubectl
38      - curl -LO https://storage.googleapis.com/kubernetes-
release/release/$(curl -s https://storage.googleapis.com/kubernetes-
release/release/stable.txt)/bin/linux/amd64/kubectl && chmod +x kubectl &&
sudo mv kubectl /usr/local/bin/
@@ -43,6 +43,6 @@ jobs:
43      script:
44      - shards install
45      - crystal spec -v || travis_terminate 1;
46      + docker pull crystallang/crystal:0.35.1-alpine
47      + docker run --rm -it -v $PWD:/workspace -w /workspace
crystallang/crystal:0.35.1-alpine crystal build src/cnf-conformance.cr --
release --static --link-flags "-lxml2 -llzma"
```



Warts and things we would like to see in the future

Not all fun and games

- Static types and JSON
- **unless** vs **if** in Crystal

Future

- Better Arm architecture support?
- gRPC in Crystal?
- Crystal protobufs?



Crystal dev help wanted

- Refactor code based on Crystal best practices
- Wrap upstream tools
- Help with static types and JSON



Get involved!



Ready to get started?

How you can get involved with the CNF Test Suite:

- Watch, **star** and/or fork the CNF Test Suite [GitHub repo](#)
- Documentation updates
- Suggest enhancements
- Report bugs
- Request/add new tests
- Request/add CNFs to be validated



Join the conversation

CNF Test Suite Contributor Meeting

- **Thursdays at 14:15 - 15:00 UTC**
- CNF Test Suite Meeting Details
- Mailing List: <https://lists.cncf.io/g/cnf-test-suite>

CNCF Slack Channels

- slack.cncf.io
 - [#cnf-testsuite-dev](#)
 - [#cnf-testbed-dev](#)
 - [#cnf-wg](#)

Connect with Vulk Coop

Stay connected

- Contact: <https://vulk.coop/contact/>
- Email: howl@vulk.coop
- Twitter: [@vulkcoop](https://twitter.com/vulkcoop)

Learn more about software cooperatives

- RSVP to Vulk Coop's monthly meetup
 - <https://www.meetup.com/Austin-Software-Co-operatives>



Contact us



Taylor Carpenter, Vulk Coop

taylor@vulk.coop

GitHub: @taylor



Will Harris, Vulk Coop

will@vulk.co

GitHub: @williscool

howl@vulk.coop



Q&A

Discussion

Feedback



Q&A



Thank You!