



# LF NETWORKING

## Developer & Testing Forum



# CLOUD NATIVE

## TELECOM INITIATIVE

For advanced users and contributors

*Best Practices, Certification, Test Catalog*



**tietoevry**

Martin Matyáš, Lead Cloud Engineer

<https://lfnetworking.org>





## Aspiration

# Help with adoption of Cloud Native technology for Networking applications

## How to reach it

- Define how the Cloud Native networking applications should be designed
  - By defining “**Best practices**”
- Provide **tooling for checking**/giving feedback for application Cloud Nativeness
  - By **automated test suite**
- Give **end users confidence** that the Cloud Native networking application is well designed and behaving
  - By providing **Certification**



# CNTi in Github

github.com/lfn-cnti

## **cnti**

Generic repo – issues, project

## **bestpractices**

Best practice definitions

## **certification**

Certification process

github.com/cnti-testcatalog

## **testsuite**

Test framework, test case  
implementation

# CNTi Best Practice

Documentation-oriented repository.

- Detailed reasoning
- Use cases
- Formalization of best practices



# CNTi Certification

Documentation-oriented repository.

- Certification criteria
- Participation forms
- Process description/steps

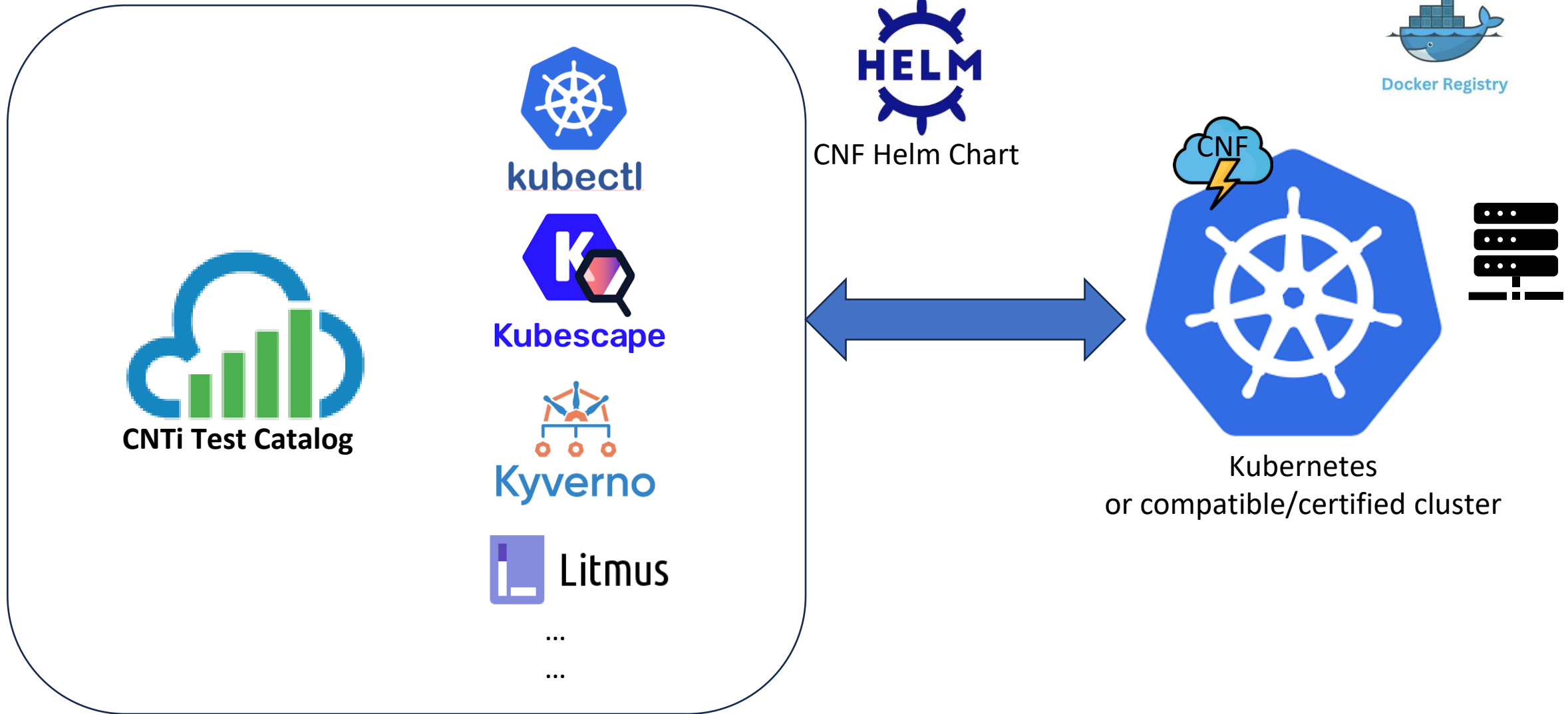


# CNTi Test Catalog

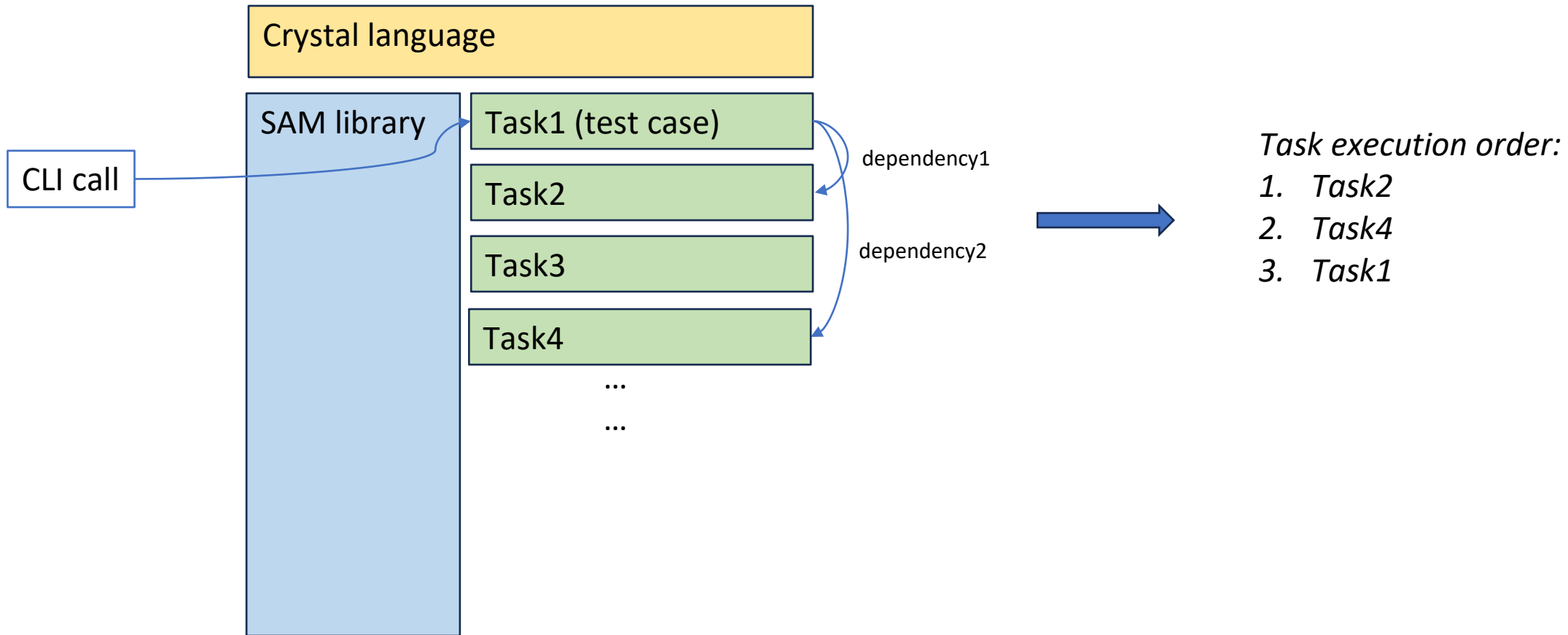
- Test suite for **evaluation CNFs** for Cloud Native Principles
- Collection of categorized **automated test cases**
- **Command-line** interface
- Leverages **upstream tools** where possible
- Crystal lang
  - compiled, **statically linked binary** with all needed included
- **Workload** tests (-> **CNF**)
- **Platform** tests (-> **K8s**)
- Detailed **test documentation**
  - description, reasoning, remediation



# CNTi Test Catalog



# CNTi Test Catalog Architecture





# CNTi Test Catalog

## directory structure

```
.
├── docs
├── embedded_files
├── example-cnfs
│   ├── coredns
│   ├── envoy
│   ├── ip-forwarder
│   ├── linkerd2
│   ├── nsm
│   ├── pantheon-nsm-nat
│   └── vpp-3c2n-csp-use-case
├── sample-cnfs
│   ├── k8s-multiple-deployments
│   ├── k8s-multiple-processes
│   ├── k8s-non-helm
│   ├── ..
│   └── ..
├── spec
│   ├── 5g
│   ├── cnf_testsuite_all
│   ├── fixtures
│   ├── platform
│   ├── utils
│   └── workload
├── src
│   ├── proto
│   ├── tasks
│   └── templates
├── tools
│   ├── airgapped_kind
│   ├── cluster-api-dev-setup
│   ├── curl_install_tester_docker_setup
│   ├── ephemeral_env
│   ├── github-runner
│   ├── reboot_daemon
│   └── registry
├── utils
│   └── airgap
```

# CNTi Test Catalog implementation aspects

- Code structure driven by crystal's sam library
  - Make-like approach
  - Based on tasks and dependencies among them
  
- Crystal language implications
  - Ruby-like syntax
  - Static type control
  - Usage of code blocks

# CNTi Test Catalog

## setup and test actions

- Test environment setup
  - `cnf-testsuite setup`
  
- Workload setup
  - `cnf-testsuite cnf_setup cnf-config=cnf-testsuite.yml`
  
- Test execution
  - `cnf-testsuite <test_or_group_name>`
  
- Collecting results
  - See `results/cnf-testsuite-results-<date>-<time>.yml`
  
- Cleanup
  - `cnf-testsuite cleanup_all`

# CNTi Test Catalog

## Useful tweaks

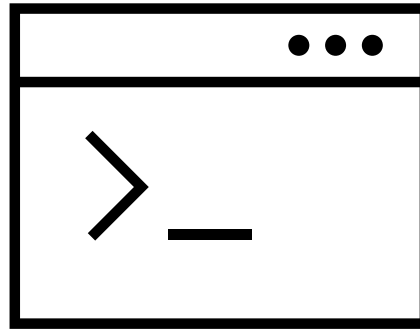
➤ Switching on logs

```
$ cnf-testsuite -l debug <test_or_group_name>
```

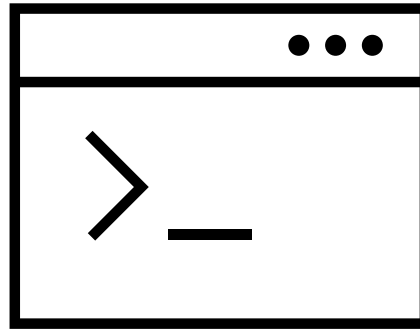
➤ Excluding one or more tests from a group

```
$ cnf-testsuite <group_name> ~<excluded_test_1> ~<excluded_test_2> ...
```

# Demo 1: non\_root\_containers



# Demo 2: Adding a new test



# CNTi Test Catalog

## Testing of tests

- Crystal's "spec" tests concept used to test tasks/tests
- Every test case has one or more spec test defined (usually a positive and a negative case)
- Spec tests use sample CNFs
- Organized in file structure and by tags
- Execution:

```
$ crystal spec <spec_file>[:line_nr]
$ crystal spec --tag <tag>
```
- All spec test are part of github actions verifying pull requests

# CNTi Challenges to address

- Attract more end users
- Attract more contributors
- Infrastructure for testing
- Improve github actions/gating
- Improve tooling
- Improve relations/dependencies among
  - Best Practices
  - Certification
  - Test Catalog



# CNTi Test catalog

## Challenges to address

- Test run duration, especially when scaling on more complex CNFs
- Usability improvements
  - Documentation structure, Documentation as a code
  - User interface
    - Better CLI
    - Better organized logs
    - Getting better information/symptoms from test execution
- Improve modularity
  - Robust **modular** k8s-oriented testing framework
  - Modules/Plugins for particular test suites (generic, telecom, other aspects)
- Add more tests

# Contribution topics for the Best practices

- **Formalizing existing best practices**
- **Join discussions about existing best practices and their weighting**
- **Help with improving best practice repo and designing relations between “best practices” and the Test catalog/Certification**

# Contribution topics for the Certification

- **Feedback from certification process is needed and would be well appreciated**
  - go through the **Certification 2.0 Beta** process with your CNF, submit result and provide feedback
- **Join discussions about Certification criteria**
- **Help to improve Certification documentation**

# Contribution topics for the Test Catalog

- **Join discussions about future architecture**
- **Join discussions about new tests**
- **Bring ideas for improvements**
- **Bring ideas about new tests**
- **Help with fixing bugs**
- **Help with new feature implementations**

# How to get in touch

- Subscribe to [LFN Tech](#) CNTi Slack channels
  - #cnti-general
  - #cnti-bestpractices
  - #cnti-testcatalog-testsuite
  - #cnti-testsuite-dev
  - #cnti-certification
- Participate on regular calls
  - Best Practices - Every other Monday at 8am PT
  - Test Catalog - Tuesdays at 8am PT
  - Certification - Every other Thursday at 8am PT
- Report issues/improvement ideas
  - Submit Issues in github repos
  - Initiate Discussions in github
- Submit pull requests

Entry point:

<https://lfnetworking.org/cloud-native-telecom-initiative/>





# Questions and maybe answers

