



KubeCon



CloudNativeCon

Europe 2020

Intro: CNCF's Telecom Initiatives

Dan Kohn, CNCF & Taylor Carpenter, Vulk Coop

Virtual

Agenda (35 Minutes)



Virtual

- Summary of CNCF's Telecom Initiatives
- Cloud Native Principles
- Community Collaboration
- Telecom User Group
- CNF Testbed
- CNF Conformance Test Suite
- How to Get Involved
- Q&A

CNCF's Imperative



Virtual

- Help Telcos navigate the cloud native and open source landscape
- Support and assist Telecom Operators in obtaining the benefits touted by cloud native technologies

Cloud Native Tools for Telecom



Virtual

Cloud Native Principles and Technologies



TELECOM
User Group

Upstream Initiatives



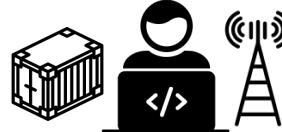
CNF Testbed



CNF Conformance



Operators



CNF Developers

End-users

Telecom User Group Summary



Virtual

A collaborative space that welcomes operators, vendors, and telecom developers who are using or aiming to use cloud native technologies, to:

- Share ideas, ask questions, and voice their needs and concerns
- Write up requirements, best practices and gap analysis
- Meet regularly via once-a-month Zoom calls
- Mailing list telecom-user-group@lists.cncf.io



TELECOM
User Group

<https://github.com/cncf/telecom-user-group>

CNF Testbed Summary



Virtual

An open source platform, toolchain, and set of examples, that supports:

- Reviewing emerging cloud native technologies in the Telecom domain
- Reproducing test results and use cases
- Building new prototypes and reference examples



CNF Testbed



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



kubernetes
KUBERNETES

BARE-METAL SERVER

HARDWARE

packet

CNF Conformance Summary



Virtual

An open source test suite for validating Cloud Native Network Function's (CNFs) and Telecom Platform's adherence to cloud native principles with the following goals:

- Provide test results for CNFs and Telecom platforms to help operators identify solutions which observe the [cloud native capabilities and principles](#) important to them
- Enable CNF Developers to demonstrate conformance to cloud native best practices
- Provide tools and feedback to help CNF Developers improve their CNFs and platforms



CNF Conformance

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

Public Telecom-focused Meetings



Virtual

- CNF Conformance/Testbed weekly developer meetings:
 - Thursdays at 14:15 - 15:00 UTC
 - **Next call on Thursday, August 20th at 14:15 UTC**
 - **Meeting details:** <https://github.com/cncf/cnf-conformance>
- CNCF Telecom User Group monthly meetings:
 - 1st Mondays (alternating start times, 15:00 UTC or 11:00 UTC)
 - **Next call on Monday, September 7th at 11:00 UTC**
 - **Meeting details:** <https://github.com/cncf/telecom-user-group>

Communication Channels



CloudNativeCon
Europe 2020

Virtual

- Subscribe to the Telecom User Group (TUG) mailing list:
 - telecom-user-group@lists.cncf.io
- Join slack.cncf.io channels:
 - [#cnf-conformance](#)
 - [#cnf-testbed](#)
 - [#tug](#)
- Email:
 - taylor@vulk.coop
 - dan@linuxfoundation.org



Feedback Welcome



KubeCon



CloudNativeCon
Europe 2020

Virtual

Provide feedback through the survey linked below:

- <https://www.surveymonkey.com/r/HLBN3XP>





KubeCon



CloudNativeCon

Europe 2020

Virtual

Cloud Native Guiding Principles

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

What is Cloud Native?



Virtual

Cloud native builds on existing principles and practices like DevOps processes, service-based architecture, agile software development, etc.

[Cloud Native Definition](#) as defined by CNCF outlines:

- Approaches that exemplify cloud native practices including microservices, immutable infrastructure, and declarative APIs
- The ability to make high-impact changes frequently and predictably with minimal toil

What are Cloud Native Principles?



Virtual

[The Cloud Native Networking Principles papers](#) provide depth and references for the principles, addressing questions such as:

- How are cloud native systems loosely coupled?
- How is immutable infrastructure provisioned?
- How are cloud native systems configured?
- What properties and characteristics are beneath these?
 - Eg. reproducible and repeatable infra is needed for immutability

A cloud native network function (CNF) is a cloud native application that implements or facilitates network functionality.

A cloud native network function consists of one or more microservices, and has been developed using [Cloud Native Principles](#) including **immutable infrastructure, declarative APIs, and a “repeatable deployment process.”**

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

What is a Cloud Native Telecom Platform?



Virtual

A platform implemented following [Cloud Native Principles](#) which supports running workloads (eg. CNFs building a large application) in a cloud native manner for its capabilities which include providing **service discovery**, **resource management**, **fault tolerance**, **declarative configuration**, and **resiliency** for both workloads as well as the platform.



KubeCon



CloudNativeCon

Europe 2020

Virtual

Community Collaboration

Community Collaboration



Virtual

Engaging and contributing in communities such as:

- [Linux Foundation Networking](#) (LFN)
 - ONAP, OVP 2.0
- [Cloud iNfrastructure Telco Task Force](#) (CNTT)
 - RA-2, RI-2, RC-2
- [XGVela](#) PaaS (platform as a service) initiative
- CNCF and K8s initiatives, groups, and projects

Many Efforts and Initiatives



Virtual

Requirements



Cloud native Principles:
[Definition](#), [Papers¹](#)

Core+Addons
Best Practices
SIGs

CNTT Reference Model
+ Architecture



Requirements to create a reference platform w/workloads meeting the model (high-level requirements and features)

Implementations

CNTT RI-2 (Testing)

XGVela



Communities can use the requirements as-is or with adjustments when implementing solutions for their end-users

Testing

CNTT RC-2 (Testing)



OPNFV

CNF Conformance



End-to-end testing
K8s Addon Specs: CSI, CRI, CNI, etc

Test cases can cover the requirements and be used for verification of the various implementations

Certification



CNF Conformance

OPNFV Verified

LFN's certification and badging program. Will use the results of several upstream testing sources. Will use multiple sources for requirements.

Bringing Value to End Users

Principles

Cloud Native Principles [1.2](#)

Community Collaboration

Initiatives



TELECOM
User Group



CNF Testbed



CNF Conformance

Output

Discussions
Gap analysis
White papers¹

Toolchain + framework:

- Infra provisioning
- Platform
- Examples + Use cases

Cloud native test cases:

- Platform
- Workload

End Users

Operators:
China Mobile, Bell
Canada

CNF Developers:
Pantheon, Intel,
Samsung

RM RA RI RC
Cloud iNfrastructure Telco Task Force
(CINTT)

OVP 2.0 The OpenFV Verified logo consists of the text "OVP 2.0" next to a blue rectangular seal with the words "OPENFV VERIFIED" and a small circular emblem.



KubeCon



CloudNativeCon

Europe 2020

Virtual

Telecom User Group

<https://github.com/cncf/telecom-user-group>

Telecom User Group Members



Virtual

arm



Charter®
COMMUNICATIONS



中国移动
China Mobile



CLOUD NATIVE
COMPUTING FOUNDATION



JUNIPER
NETWORKS



OLF
NETWORKING



NOKIA

SAMSUNG

swisscom

T Mobile™

verizon®

vodafone

VULK COOP

Telecom User Group

- A group for operators, vendors, and telecom developers who are using or aiming to use cloud native technologies
- Share ideas, ask questions, and voice their needs and concerns
- The TUG is not expected to do software development, but may write up requirements, best practices, gap analysis, or similar documents.
- We kicked off the TUG in a birds-of-a-feather (BoF) in 2019 in [Barcelona](#) and [Shanghai](#) and have moved to once-a-month Zoom [calls](#).



TELECOM
User Group

Telecom White Papers



Virtual

- Created the [Cloud Native Thinking for Telecommunications](#) white paper
- Contributing to and sourcing the [Cloud Native Networking papers](#)

<https://github.com/cncf/telecom-user-group>

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

Collaboration with CNTT, OVP & OPNFV



Virtual

- Provide feedback and improvements to the CNTT RA-2 requirements
- Help identify scope and gaps in meeting CNTT requirements in the RI-2 workstream
- Working with CNCF and K8s projects + initiatives to share learnings

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

<https://github.com/crosscloudci/k8s-infra>



KubeCon



CloudNativeCon

Europe 2020

Virtual

CNF Testbed

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

CNF Testbed Contributors



Virtual

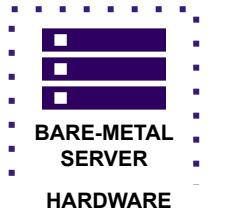
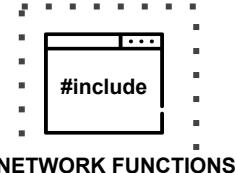


CNF Testbed



Virtual

- An open source [testbed](#)
 - Testing and reviewing emerging cloud native technologies in the Telecom domain
 - Providing fully reproducible use cases and examples
- Out-of-the box support deployments to on-demand hardware from the bare metal hosting company, [Packet](#)



packet

Technology Innovation Review Tool



Virtual

- Support using different technology options
- Keep things as simple as reasonable
- Use upstream community tooling (ex. Kubespray, NFVbench)
- Use cloud native principles where possible

Use Cloud Native Principles



Virtual

- Where possible use cloud native principles for all levels (hardware to use case)
 - Immutable hardware
 - Version control all configuration including underlay networking
 - Workload bootstrapping repeatable by automation/pipeline
- Highlight where gaps are missing and out-of-band procedures are used
- Bring focus to technology which is attempting to provide solutions to meet cloud native principles

Key Features of CNF Testbed

- Fully repeatable in Packet from scratch
- Uses in-band components as much as possible
 - Helm or kubectl for K8s use cases and examples
 - Terraform + Kubespray
 - K8s-native replacements for out-of-band host setup where available
- Examples covering minimal usage, service chaining, and various use cases
- Demonstrations of emerging technologies, such as:
 - Network Service Mesh
 - SR-IOV device plugins
 - K8s CPU manager addons
 - Multus

We Welcome Your Participation



Virtual

- Replicate this work using the code published at github.com/cncf/cnf-testbed with an API key from packet.com/cnf
- Package your internal network functions in containers (ideally following cloud native principles) and run on your instance of the testbed
 - We don't need to see the code but would love to see the results
- Create pull requests with improvements to the provisioning toolchain, to add support for running the CNF Testbed in own your lab, or another public bare metal provider
- Contribute new use cases and examples to the CNF Testbed ([issues](#) or [spec board](#))



KubeCon



CloudNativeCon

Europe 2020

Virtual

CNF Conformance

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

CNF Conformance Contributors



Virtual



**Taylor
Carpenter**
@taylor



**W.
Watson**
@wavell



**Joshua
Smith**
@nupejosh



**Lucina
Stricko**
@lixuna



**Denver
Williams**
@denverwilliams



**William
Harris**
@williscool



**Michael S.
Pedersen**
@michaels
pedersen



**Petar
Torre**
@petorre



**Victor
Morales**
@electro
cucaracha



**Drew
Bentley**
@agentpoyo

- An open source test suite for self-validating Cloud native Network Functions (CNFs) and the underlying Telecom platforms
- Provides visibility into how well CNFs and the platforms follow [cloud native principles](#)
- Based on the [K8s Conformance](#) program

<https://github.com/cncf/k8s-conformance>

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

CNF Conformance Goals



Virtual

- Provide operators with test results for CNFs and CNF Platforms to help with find solutions with the cloud native capabilities and principles important to them
- Provide an open source test suite to enable CNF Developers to demonstrate conformance to cloud native best practices
- Provide tools and feedback to CNF Developers to improve their CNFs and platforms

Platform & Workload Test Categories



Virtual

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.

- **Configuration and Lifecycle:** Declarative interfaces, resilience, availability, versioning
- **Installable and Upgradeable:** Such as via a [Helm](#) chart and/or [Kustomize](#) plugin
- **Hardware support:** device and resource access via declarative interfaces
- **Microservice:** Applications are broken into loosely coupled and independently deployable services which are organized around business capabilities (limited set of concerns)
- **Compatible:** They should work with any [Certified Kubernetes](#) product and any CNI-compatible network that meet their functionality requirements
- **Stateless:** State should be stored in a Custom Resource Definition or a separate database rather than requiring local storage
- **Security:** Privilege checks, separation of workloads, accessing host machines
- **Scaling:** They should support horizontal scaling (across multiple machines) and vertical scaling (between sizes of machines)
- **Observability:** They should externalize their internal states in some way that lends itself to metrics, tracing, and logging
- **Resilience:** CNFs should be resilient to failures inevitable in cloud environments.



KubeCon



CloudNativeCon

Europe 2020

Virtual

CNF Conformance Workload Tests

Initial Set of Workload Tests



Virtual

Microservice	Compatibility	Statelessness	Security	Scalability	Configuration	Observability	Installable	Resilience
Startup time	N/A	hostPath volumes	Privileged mode	Increase capacity	Liveness	N/A	Valid Helm chart	CNCF back up when container killed/dies
Size of container image				Decrease capacity	Readiness		Helm Deploy	Litmus Chaos Pod Network Loss
					Rolling version upgrade		Published Helm chart	Litmus Chaos Pod CPU Hog
					nodePort			
					Static: Hardcoded IPs in source code			



KubeCon



CloudNativeCon

Europe 2020

Virtual

CNF Conformance Platform Tests

Overview of Platform Testing

- Focus on K8s-based platforms
 - The Architecture **must** run conformant Kubernetes as defined by <https://github.com/cncf/k8s-conformance>
- All layers
 - Addons/extensions
 - Underlying hardware/network provisioning
- Integration between layers up to the workload

Requirement Sources



Virtual

- [CNTT K8s Reference Architecture Requirements](#)
- Direct Feedback from Operators and CNF Developers
- K8s community best practices, including [security](#)
- Cloud Native Principles at <https://networking.cloud-native-principles.org/>

Initial Set of Platform Tests



Virtual

Microservice	Compatibility	Statelessness	Security	Scalability	Configuration	Observability	Hardware	Resilience
N/A	Custom Resource Definition (CRD) in use	N/A	Cloud native container runtime (CRI)	Horizontal auto scaling with cluster-api	Cloud native userspace data planes	Cloud native logging	CPU Management policies	Chaos resilience with node failure
				Horizontal auto scaling with new nodes	Network Service Mesh deployed to platform	Cloud native monitoring	Cloud native Container Network Interfaces (CNI)	Chaos network test
				Supports K8s network policy				High-load signalling traffic



KubeCon



CloudNativeCon

Europe 2020

Virtual

CNF Examples

Current CNF Examples



Virtual

- [Example-cnfs](#)
 - CoreDNS
 - Envoy
 - IP-forwarder
 -  Linkerd2

Future CNF Examples to Test



Virtual

- EXAMPLE-CNFS.md

- free5GC (<https://www.free5gc.org/>)
- vBNG from ONAP BBS use-case
- Go-GTP based example
- Envoy with WASM
- 5G Packet Core
- Scaling P-GW

Service chains, use cases, and CNFs with dependent services



KubeCon



CloudNativeCon

Europe 2020

Virtual

Using the CNF Conformance Test Suite

Overview of Instructions & Usage



Virtual

- A [INSTALL Guide](#) is published on the GitHub repo:
 - Minimal [CNF Developer Instructions](#) provide a quick start for using the test suite
 - [Test developer / source install instructions](#) are available
- [Extensive usage instructions](#) with details for running individual tests are available
- Prereqs are outlined, including using your own K8s cluster, Kind, or how to deploy a CNF Testbed based cluster.
- Instructions on configuring the suite to use your CNF are covered, including a [example configuration file](#) which can be copied and modified.

Run All Tests with Your CNF



CloudNativeCon
Europe 2020

Virtual

- Start a full workload test run with: cnf-conformance all
- On the console message for each test will show PASSED or FAILED

```
✓ PASSED: No privileged containers
✓ PASSED: Replicas increased to 3 
✓ PASSED: Replicas decreased to 1 
✗ FAILURE: IP addresses found
✓ PASSED: Helm liveness probe found
✓ PASSED: Helm readiness probe found
✓ PASSED: CNF coredns-coredns Rolling Update Passed
✓ PASSED: NodePort is not used
✓ PASSED: No hard-coded IP addresses found in the runtime K8s
✓ PASSED: Helm Chart cnfs/coredns/helm_chart/coredns lint Passed
```

- A final score will be shown when the test completes

```
Resilience final score: 15 of 15
Final score: 94 of 100
```

- A log file with all the results will be created as well

Scoring System and Test Results



Virtual

- The CNF scoring system will be applicable to both Workload and Platform test results
- The scoring system structure and functionality is based on Sonobuoy K8s Conformance test:
 - Scoring can be updated and modified via a [yaml configuration file](#) which maps to each of the tests
 - The test results are saved to a yaml file which contains a summary for the entire test as well as individual results for each test
- The test results from CNF Conformance can be used downstream in:
 - LFN's OVP 2.0 badging program
 - CNTT RC-2 testing efforts
 - Operators and CNF Developers directly

Ready to Get Started?



Virtual

- Try using the CNF Conformance test suite, provide feedback and contribute
 - <https://github.com/cncf/cnf-conformance>
 - [CONTRIBUTING.md](#)
- Run CNF Conformance using the [CNF Developer Install and Usage guide](#)
- Pull Requests welcome for:
 - Suggestions to improve documentation
 - New CNF Examples to run in the test suite
 - New conformance test ideas/prototypes



KubeCon



CloudNativeCon

Europe 2020

Virtual

Collaborating and Contributing to Communities

Cross-Community Contributions

- Provide feedback and improvements to requirement sources
- Help identify technology options, provide reference examples, and find gaps in implementation efforts
- Provide a cloud native checkbox for testing and badging
- Share learnings across communities

<https://github.com/cncf/cnf-testbed>
<https://github.com/crosscloudci/k8s-infra>

From Principles to End User Value

Principles

Cloud native Principles [1,2](#)

Community Collaboration

Initiatives



TELECOM
User Group



CNF Testbed



CNF Conformance

Output

Discussions
Gap analysis
White papers¹

Toolchain + framework:

- Infra provisioning
- Platform
- Examples + Use cases

Cloud native test cases:

- Platform
- Workload

End Users

Operators:
China Mobile, Bell
Canada

CNF Developers:
Pantheon, Intel,
Samsung

RM RA RI RC
Cloud iNfrastructure Telco Task Force
(CINTT)

OVP 2.0





KubeCon



CloudNativeCon

Europe 2020

Virtual

How to Get Involved

Upcoming Meetings



Virtual

- **CNF Conformance/Testbed Dev weekly meeting:**
 - Thursday, August 20th at 14:15 - 15:00 UTC
- **CNCF Telecom User Group monthly meeting:**
 - Monday, September 7th at 11:00 UTC
- **Open Networking & Edge Summit 2020 (virtual)**
 - Wednesday, September 30th at 17:45 - 18:15 UTC
 - Birds of a Feather: CNCF Telecom User Group
 - <https://sched.co/bWQV>

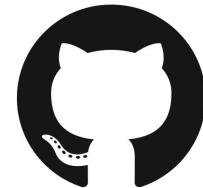
Communication Channels

- Mailing list:
 - telecom-user-group@lists.cncf.io
- slack.cncf.io channels:
 - [#cnf-conformance](#)
 - [#cnf-testbed](#)
 - [#tug](#)
- Email:
 - taylor@vulk.coop
 - dan@linuxfoundation.org
- Book a time to meet with Dan Kohn:
 - <https://calendly.com/dankohn>



GitHub & Twitter

- Telecom User Group
 - <https://github.com/cncf/telecom-user-group>
- CNF Testbed
 - <https://github.com/cncf/cnf-testbed>
 - <https://twitter.com/cnftestbed>
- CNF Conformance
 - <https://github.com/cncf/cnf-conformance>
 - <https://twitter.com/cnfconformance>



Feedback Welcome



Virtual

Provide feedback and signup for beta testing through the survey linked below:

- <https://www.surveymonkey.com/r/HLBN3XP>





KubeCon



CloudNativeCon

Europe 2020

Virtual

Q&A - Live

Taylor Carpenter, Vulk Coop

taylor@vulk.coop

GitHub: taylor

Twitter: ixx



KubeCon



CloudNativeCon

Europe 2020

Virtual

Thank you!

Taylor Carpenter, Vulk Coop

taylor@vulk.coop

GitHub: taylor

Twitter: ixx



KubeCon



CloudNativeCon

Europe 2020

Virtual



KEEP CLOUD NATIVE
CONNECTED

