

# CNFs vs. VNFs

Dan Kohn  
Executive Director, CNCF



























# Cloud Native Computing Foundation

- Non-profit, part of the Linux Foundation; founded Dec 2015

## Graduated

## Incubating

## Sandbox

 <b>kubernetes</b> Orchestration	 OPENTRACING Distributed Tracing API	 fluentd Logging	 gRPC Remote Procedure Call	 containerd Container Runtime	 rkt Container Runtime	 CNI Networking API	 envoy Service Mesh	 ROOK Storage	 spiffe Identity Spec	 SPIRE Identity	 Open Policy Agent Policy		
 <b>Prometheus</b> Monitoring	 JAEGER Distributed Tracing	 TUF Software Update Spec	 Notary Security	 Vitess Storage	 CoreDNS Service Discovery	 NATS Messaging	 LINKERD Service Mesh	 HELM Package Management	 cloudevents Serverless	 TELEPRESENCE Tooling	 HARBOR Registry	 OPENMETRICS Metrics Spec	 KV Key/Value Store

- Platinum members:

 Alibaba Cloud	 aws	 Azure	 CISCO	 DELL Technologies	 docker	 FUJITSU	 Google Cloud	 HUAWEI	
 IBM Cloud	 intel	 JD.COM	 MESOSPHERE	 ORACLE	 Pivotal	 redhat.	 SAMSUNG SAMSUNG SDS	 SAP	 vmware®



# TODAY THE LINUX FOUNDATION IS MUCH MORE THAN LINUX



## Security

We are helping global privacy and security through a program to encrypt the entire internet.



## Networking

We are creating ecosystems around networking to improve agility in the evolving software-defined datacenter.



## Cloud

We are creating a portability layer for the cloud, driving de facto standards and developing the orchestration layer for all clouds.



## Automotive

We are creating the platform for infotainment in the auto industry that can be expanded into instrument clusters and telematics systems.



## Blockchain

We are creating a permanent, secure distributed ledger that makes it easier to create cost-efficient, decentralized business networks.



## Web

We are providing the application development framework for next generation web, mobile, serverless, and IoT applications.



We are regularly adding projects; for the most up-to-date listing of all projects visit [tlfprojects.org](https://www.linuxfoundation.org/projects)



# KubeCon + CloudNativeCon

- China
  - [Shanghai](#): November 14-15, 2018
  - Sponsorships [open](#)
- North America
  - Seattle: December 11 - 13, 2018
  - Sponsorships [open](#)
- Europe
  - Barcelona: May 21 - 23, 2019



# Network Architecture Evolution

- › 1.0: Separate physical boxes for each component (e.g., routers, switches, firewalls)
- › 2.0: Physical boxes converted to virtual machines called Virtual Network Functions (VNFs) running on VMware or OpenStack
- › 3.0: Cloud-native Network Functions (CNFs) running on Kubernetes on public, private, or hybrid clouds







# Network Architecture 1.0

# Network Architecture Evolution

- › 1.0: Separate physical boxes for each component (e.g., routers, switches, firewalls)
- › 2.0: Physical boxes converted to virtual machines called Virtual Network Functions (VNFs) running on VMware or OpenStack







# Network Architecture 2.0



# Network Architecture Evolution

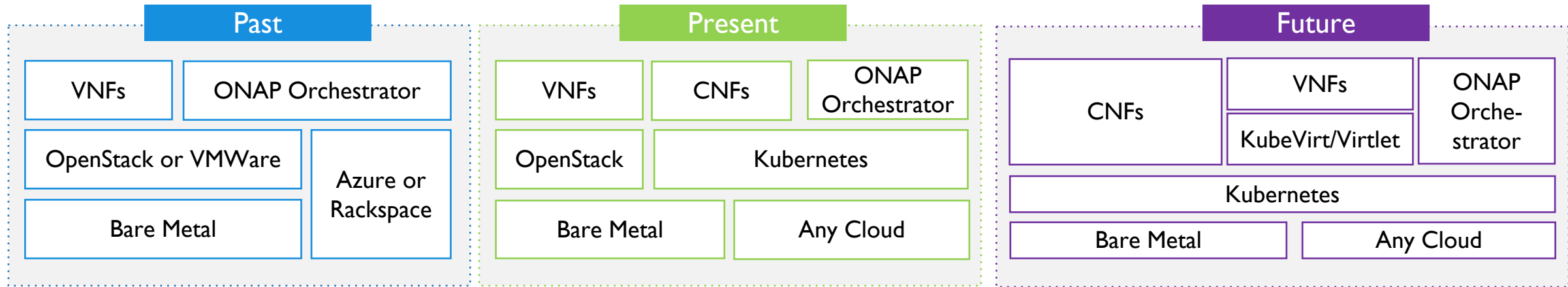
- › 1.0: Separate physical boxes for each component (e.g., routers, switches, firewalls)
- › 2.0: Physical boxes converted to virtual machines called Virtual Network Functions (VNFs) running on VMware or OpenStack
- › 3.0: Cloud-native Network Functions (CNFs) running on Kubernetes on public, private, or hybrid clouds





**Network Architecture 3.0**  
(hardware is the same as 2.0)

# Evolving from VNFs to CNFs



- › ONAP Amsterdam (Past) runs on OpenStack, VMware, Azure or Rackspace
- › ONAP Casablanca (Present) runs on Kubernetes and so works on any public, private or hybrid cloud
- › Virtual Network Functions (VNFs) are virtual machines that run on OpenStack or VMWare, or can be run on K8s via [KubeVirt](#) or [Virtlet](#)





# Major Benefits

1. Cost savings (with public, private, and hybrid clouds)
2. Development velocity
3. Resiliency (to failures of individual CNFs, machines, and even data centers)



# The challenge of transitioning VNFs to CNFs

- › Moving from network functionality from *physical* hardware to encapsulating the software in a *virtual* machine (P2V) is generally easier than *containerizing* the software (P2C or V2C)
- › Many network function virtualization VMs rely on kernel hacks or otherwise do not restrict themselves to just the stable Linux kernel userspace ABI
  - › They also often need to use DPDK or SR-IOV to achieve sufficient performance
- › Containers provide nearly direct access to the hardware with little or no virtualization overhead
  - › But they expect containerized applications to use the stable userspace Linux kernel ABI, not to bypass it



# Areas for More Discussion

- › The strength of no longer being locked into specific OSs
  - › Any version of Linux >3.10 is acceptable
- › [Multi-interface](#) pods vs. [Network Service Mesh](#)
- › Complete [parity](#) for IPv6 functionality and [dual-stack](#) support in K8s
- › Security, and specifically recommendations from [Google](#) and [Jess](#) that come into play when hosting untrusted, user-provided code
  - › Possible use of isolation layers such as [gVisor](#) or [Kata](#)
- › Scheduling container workloads with network-related hardware constraints (similar to what's been done for GPUs)
  - › Network-specific functionality like QOS





# Demo Plans Underway

- › VNFs vs. CNFs
  - › Working on a demo of boot-time and throughput of VNFs on OpenStack vs. CNFs on Kubernetes, where the networking code and underlying hardware is identical
  - › Will deliver opens source installers and Helm charts
- › Cloud-native Customer Premises Equipment (CCPE) Project
  - › Modify the ONAP vCPE [use case](#) and [VNF](#) deployment to show VNF vs. CNF deployments of chained network functions



# Roll-Out Plans

- › [Open Source Summit NA](#), Vancouver, August 28: Joint workshop by CNCF executive director Dan Kohn and LF Networking head Arpit Joshipura on Cloud-native Network Functions
- › [Open Network Summit Europe](#), Amsterdam, September 25: Marketing launch
- › [KubeCon + CloudNativeCon NA](#), Seattle, December 11: Planned demo
- › [Mobile World Congress](#), Barcelona, February 25: Major roll-out
- › Ongoing close collaboration with LF Networking and specific carriers providing feedback (AT&T, Bell Canada, Vodafone, etc.)



# The Networking aspects of Cloud Native

**Arpit Joshipura**  
GM Networking  
The Linux Foundation



## Industry Direction: Any Cloud + Portable Apps in Containers

- › Utilize the best of Cloud with the best of Telecom Networks
  - › Promise of Containers – allow for any cloud portability
  - › Promise of Network – full network automation & zero touch services
- › Telecom Network Transformation require a hybrid strategy
  - › Migration of VM to containers step by step approach to VNF/Workloads
  - › VM and Container Interworking in a Multi-VIM environment

## Two leading de-facto platforms –Networking & Cloud

 **LF** NETWORKING

&

 **CLOUD NATIVE**  
COMPUTING FOUNDATION

 **ONAP**  
OPEN NETWORK AUTOMATION PLATFORM

 **kubernetes**

- **Network Automation & Orchestration Platform**
- **ONAP has a multi-VIM strategy (Openstack, Vmware, Azure,..)**
- **Project within ONAP – OOM looking at Containers**

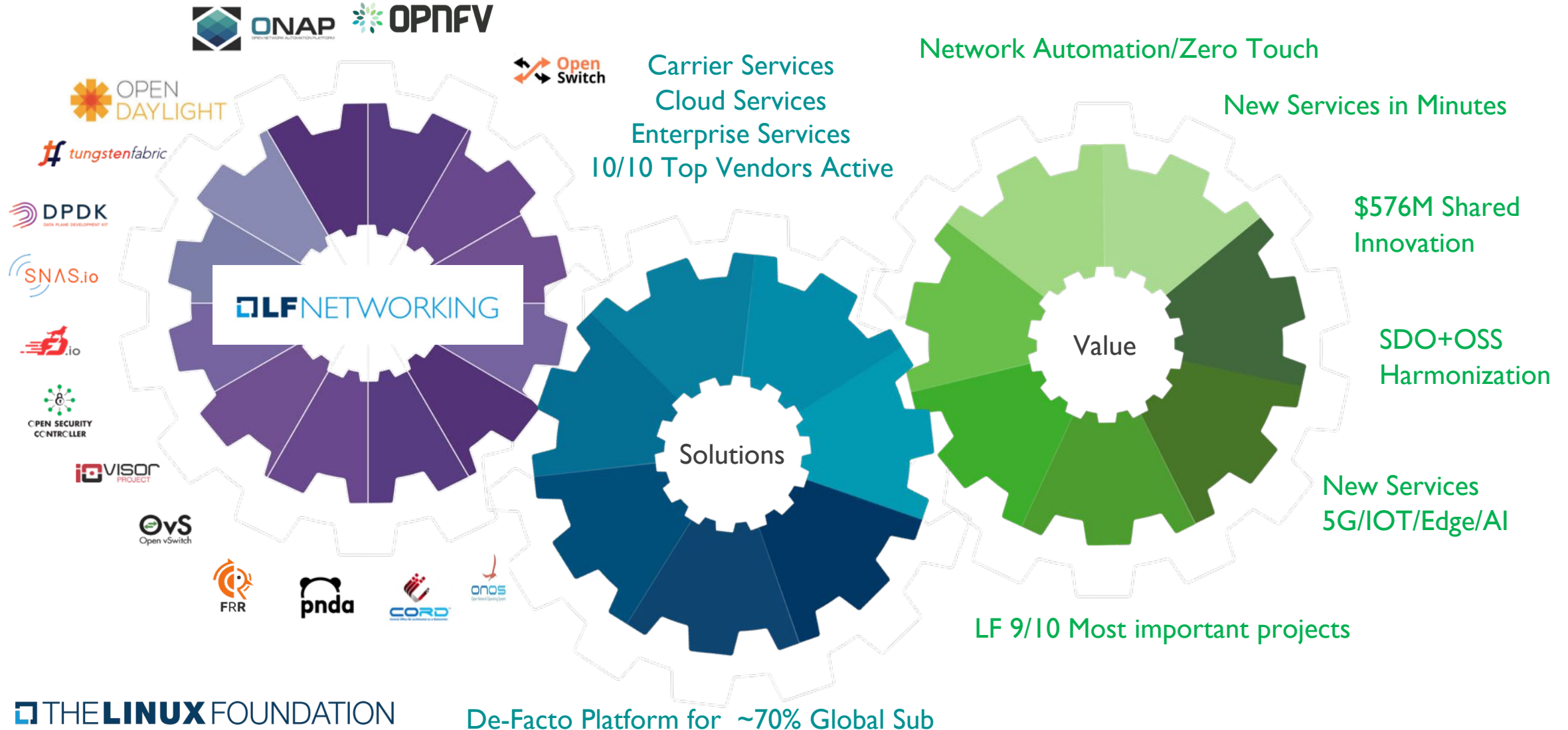
- **Cloud Automation & Orchestration Platform**
- **Project within CNCF looking at ONAP – Cross Cloud CI**

**Open Source projects at LF can bring the best of both worlds to the Telecom Industry**

 **THE LINUX** FOUNDATION

# Sustainable Innovation: Open Source Networking

Creating *De-Facto* Platforms to Enable Next Generation Solutions in Telecom, Enterprise & Cloud



# LF Networking Vision: Automating Cloud, Network, & IOT Services

**Services**

**Cloud Services**

**Residential Services**

**Enterprise Services**

**IOT Services**



**Software & Automation**

**Cloud Automation**

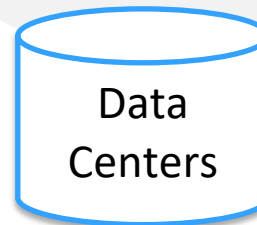
**LF NETWORKING**

(ONAP, OPNFV, ODL, FD.io, SNAS, PNDA)

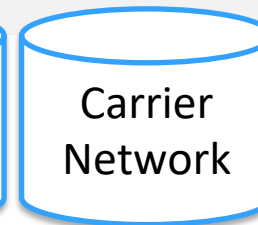
**IOT Automation**

**Infrastructure**

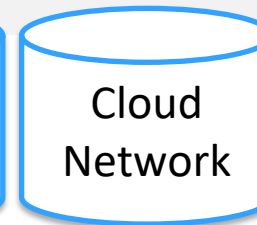
**Enterprise**  
Software Defined Data Centers (SDDC)



Data Centers



Carrier Network



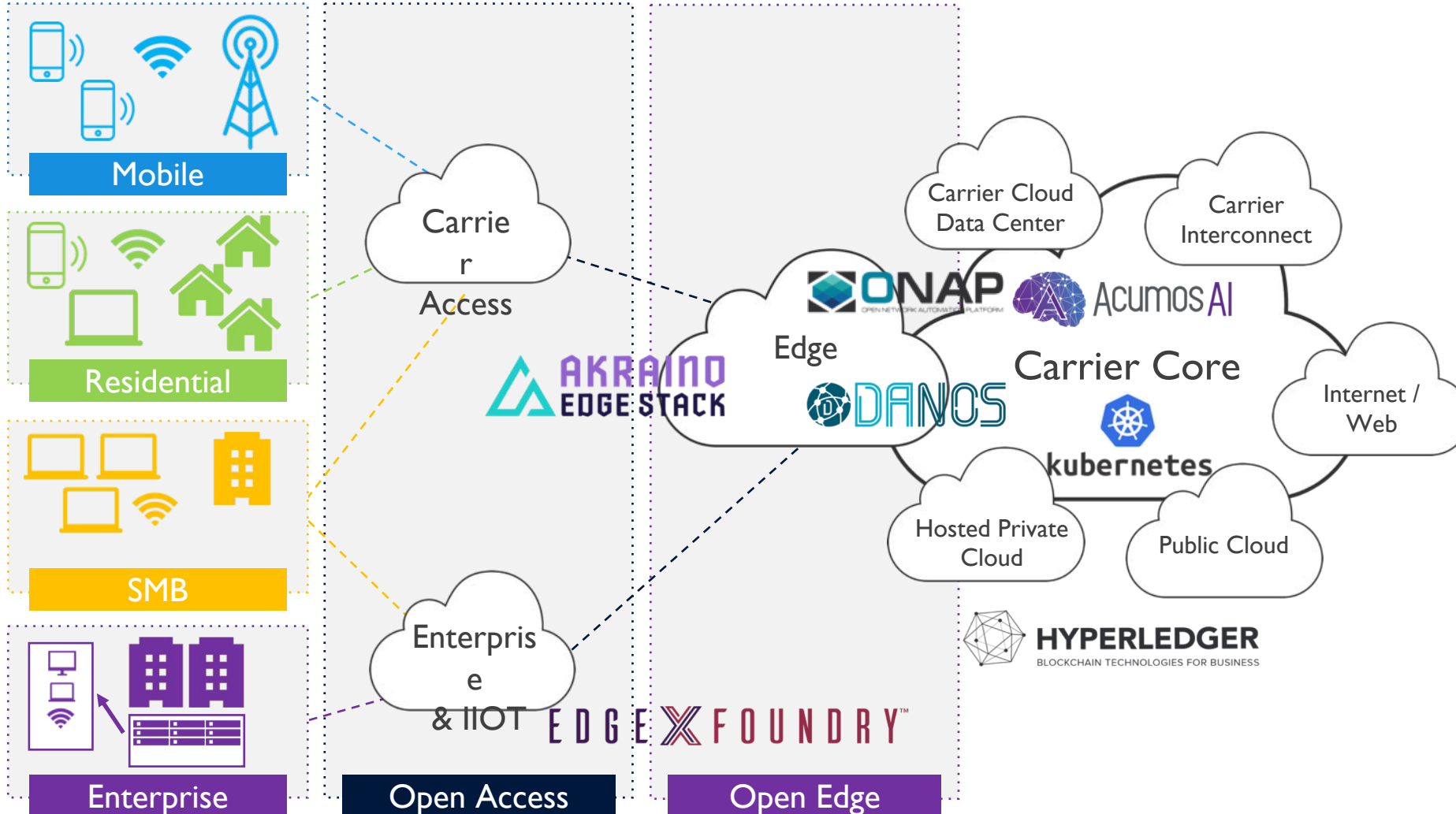
Cloud Network

**Public/Hybrid**  
Cloud Service Providers  
Cloud Hosting  
Private Cloud Providers  
Web Service Providers

Service Providers  
MSO/CableCo



# Bringing It All Together Core to Edge – LF Open Source Network + Disaggregation + Edge + IOT + AI + Cloud + Blockchain



Standards for Edge



Ref Implementation

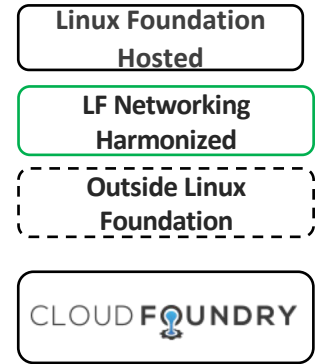
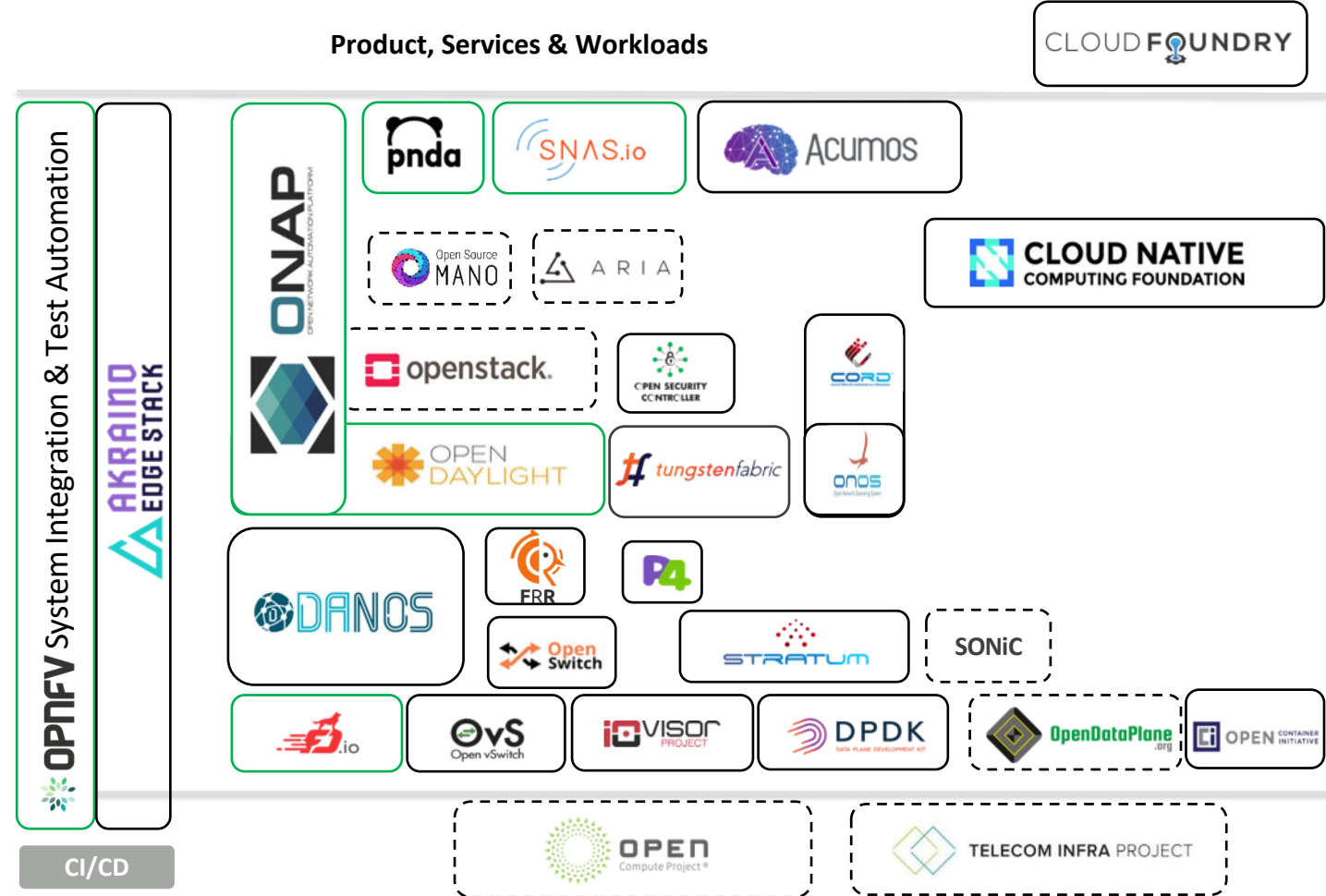
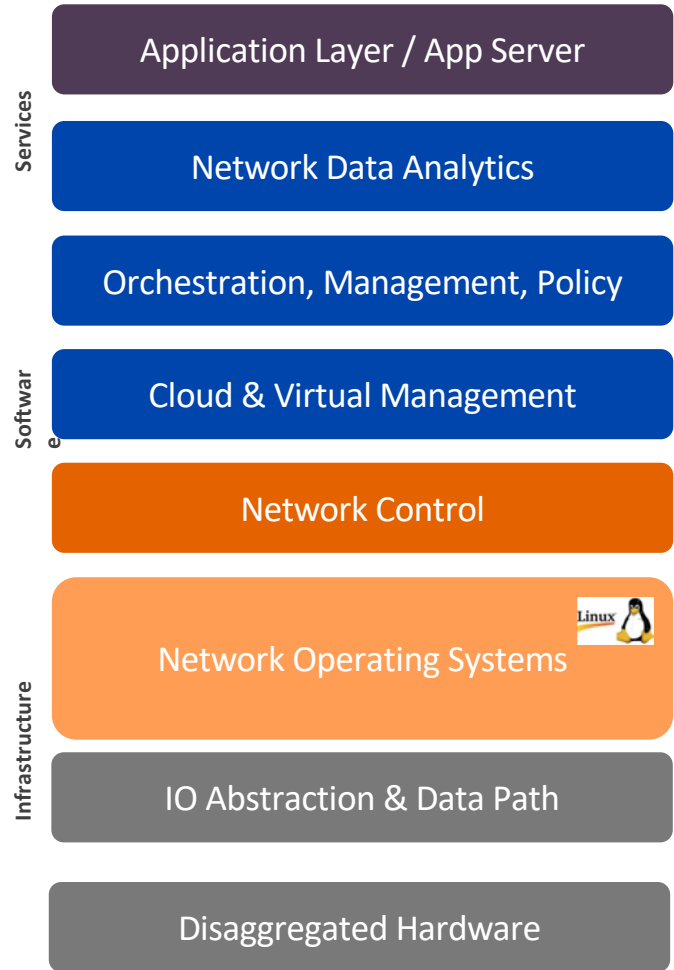


IoT, Gateway & Cloud Ref Arch

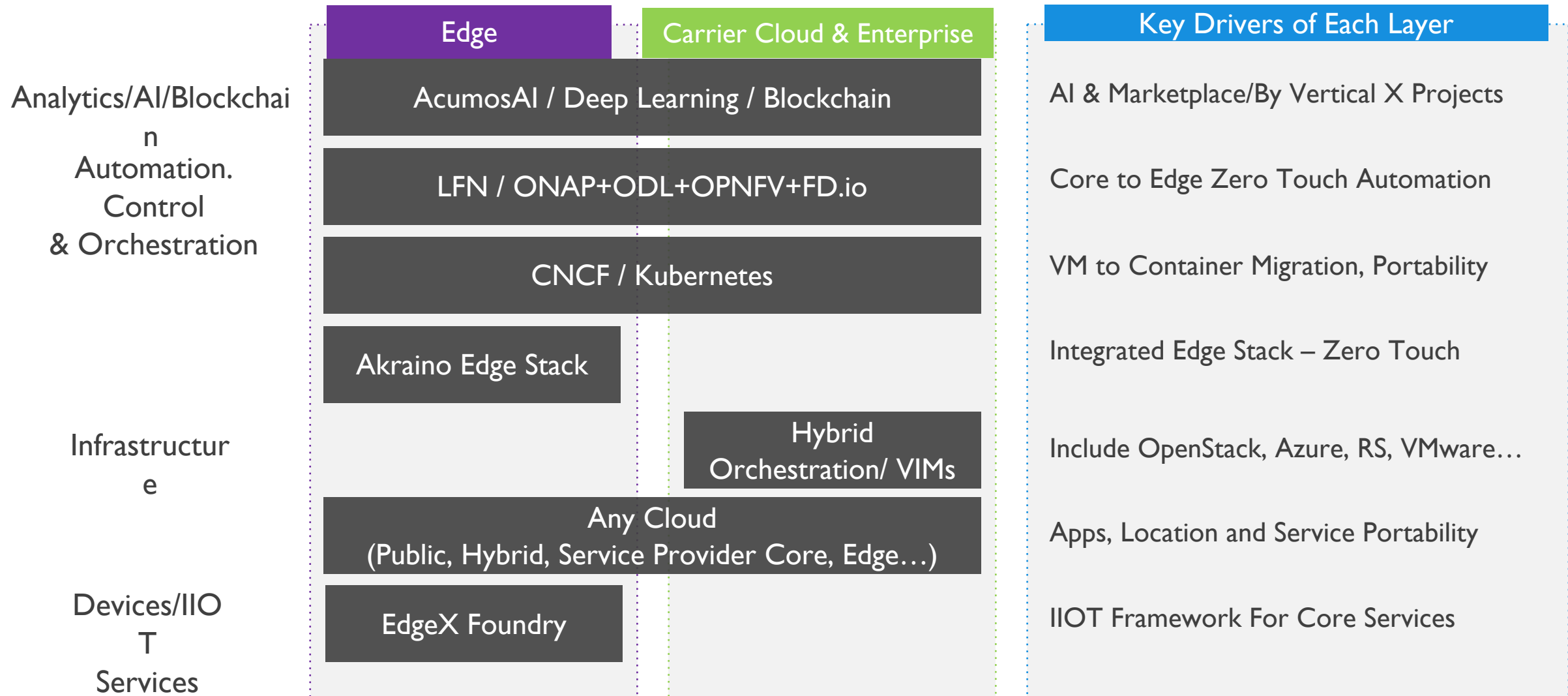


Other Edge Activities

# Open Source Networking Landscape Linux Foundation hosts 9/10 Top projects



# Linux Foundation Path to Open Source Harmonization 2.0

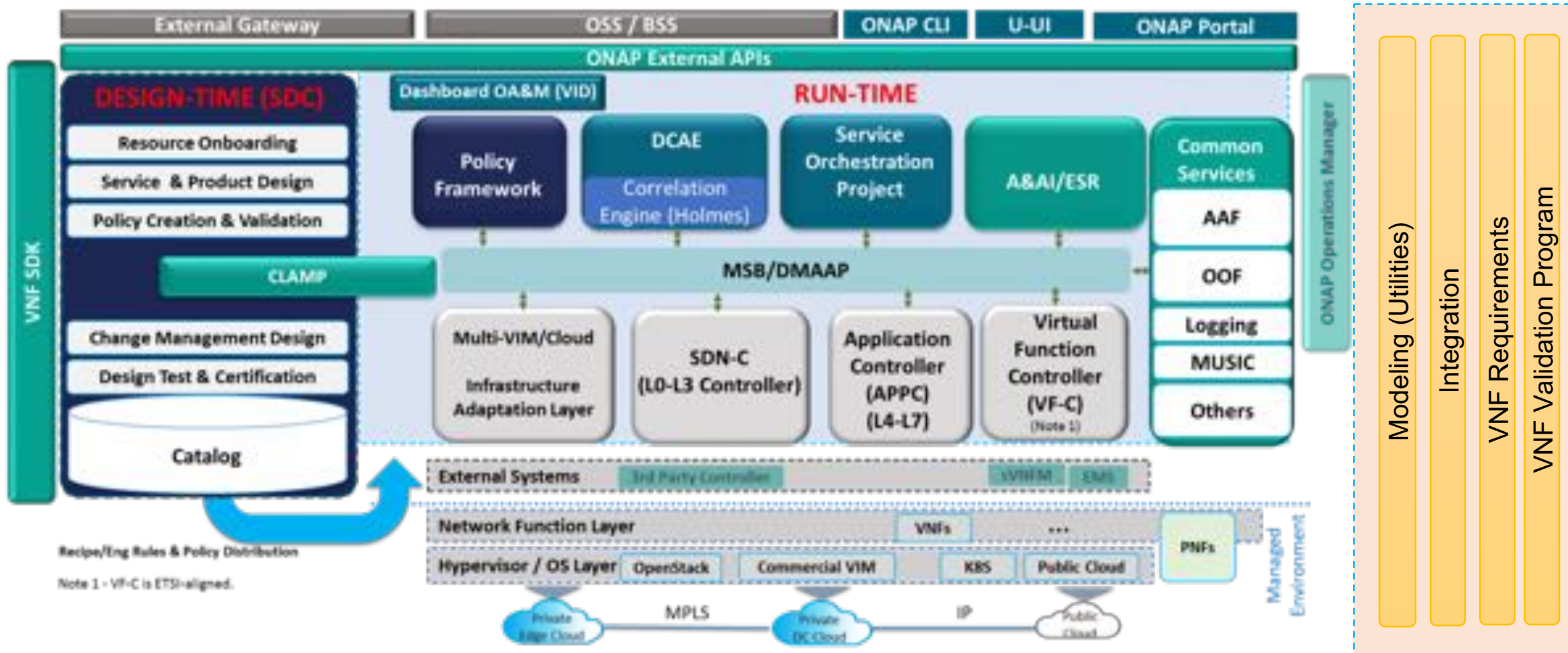


## The deep dive – VNFs on ONAP & Cloud Native journey

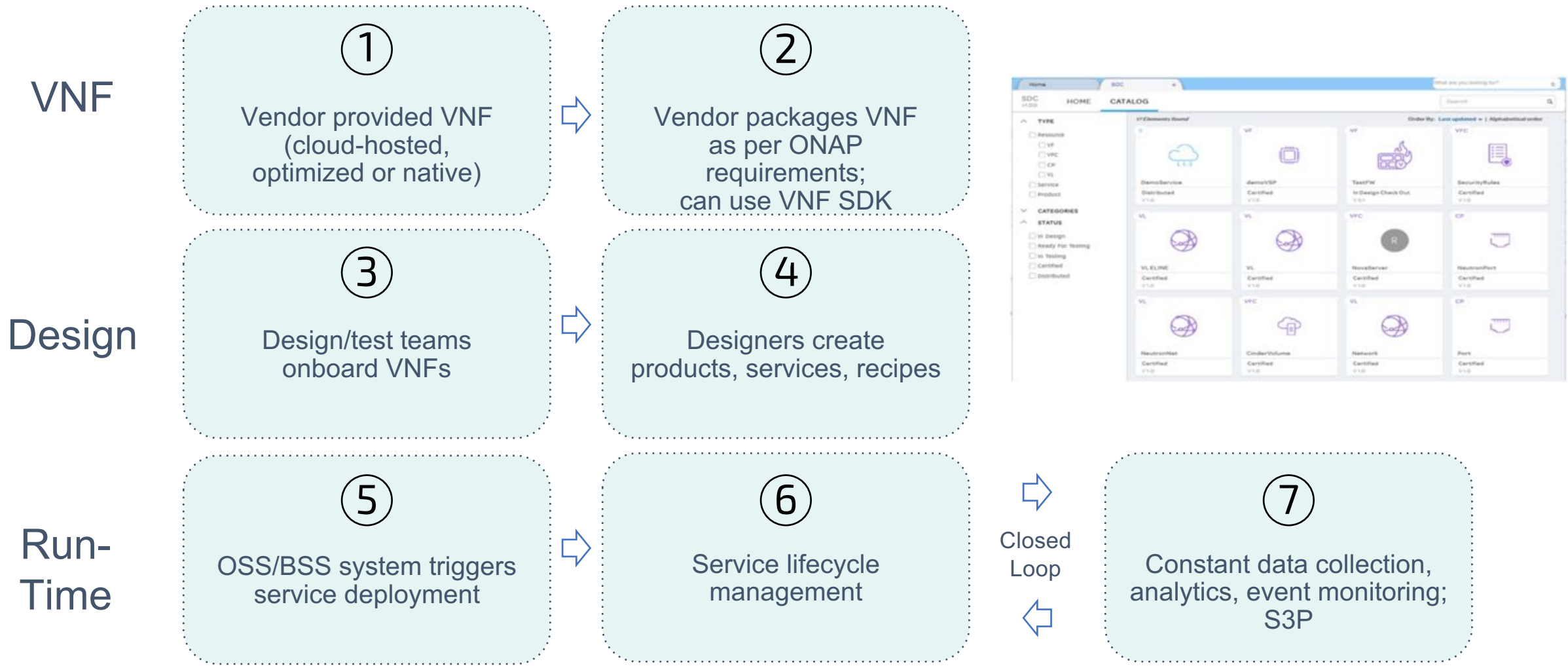




# ONAP Beijing Architecture



# A Day in the Life of an ONAP Service



Credit: Aarna Networks, ONAP Training course

# Kubernetes Gap Analysis & Transition plans

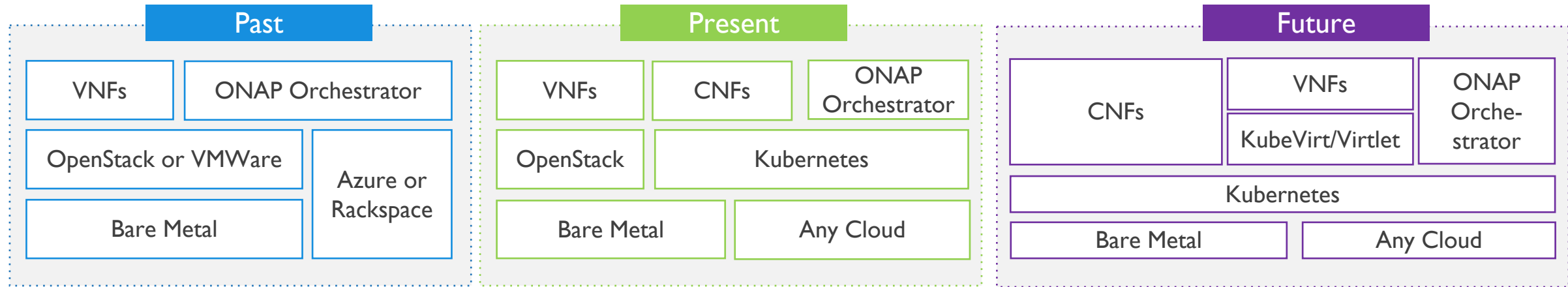
## Top 3 Areas of Investigation

1. Support for VNFs/Apps from different vendors (Ref guest OS and VNF ecosystem alignment)
2. Security – Access rights and privileges, known rules for admins etc
3. Network Specific Requirements – focus on performance, scale and capabilities
  - › Enabling scheduling container workloads with network-related hardware constraints (similar to what's been done for GPUs)
  - › Multi-homed containers (Multis, a CNI plugin is working on this)
  - › Functional parity when deployed with IPv6
  - › Network-specific functionality like QOS
  - › multiple vNIC for a given container, which is currently not supported.

## Transition plan

- › Demo of Any app, Any cloud with ONAP running on Kubernetes @ ONS opening keynote
- › Kubernetes as the Virtual Infrastructure Manager (VIM) for running the ONAP Orchestrator
- › Consider Kubernetes as the ONAP Application Controller (AppC)
- › As VNFs can be containerized, do so and manage them with Kubernetes – prioritize use cases (eg CDN, DNS)

## Evolving from VNFs to CNFs



- › ONAP Amsterdam (Past) runs on OpenStack, VMWare, Azure or Rackspace
- › ONAP Casablanca (Present) runs on Kubernetes and so works on any public, private or hybrid cloud
- › Virtual Network Functions (VNFs) are virtual machines that run on OpenStack or VMWare, or can be run on K8s via [KubeVirt](#) or [Virtlet](#)



# Today's Agenda

## **BACKGROUND AND VISION**

1:30 Introduction to VNFs and CNFs & Cross-cloud Dan Kohn

2:00 Networking & Telecom Automation: VNF to CNF journey Arpit Joshipura

## **REQUIREMENTS**

2:15 Cloud Native lessons and requirement: A view from end user - Telus, Sanah Tariq

2:30 (Dan/Arpit facilitate) Why Telecom and Cloud Native technologies are coming together – discuss challenges and requirements

## **BREAK**

## **PROJECTS AND ROADMAP**

3:30 Overview of projects solving the migration Roadmap to Cloud Native

3:50 Network Service Mesh (VPP/Ligato) Ed Warnicke

4:10 Cross-Cloud CI working group Taylor

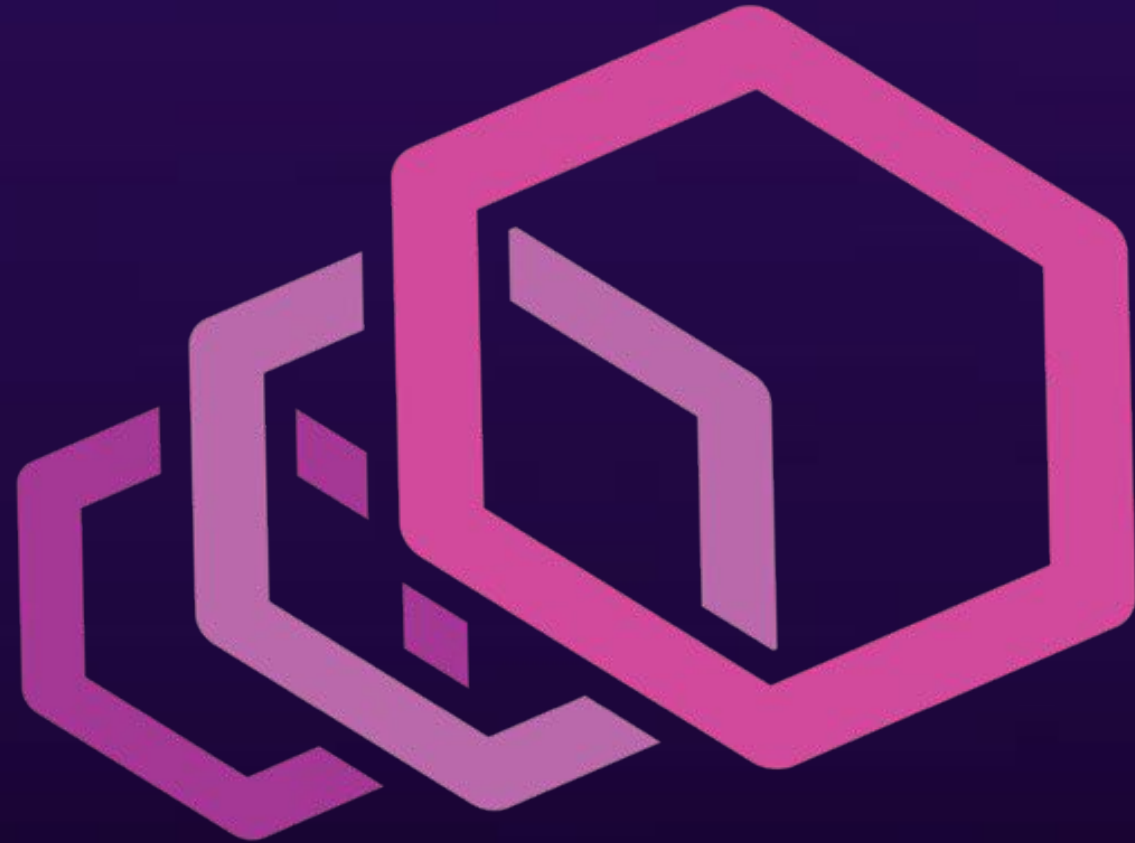
4:30 Wrap up and How to get involved

The background of the image is a dark blue field filled with a complex network of glowing blue lines. These lines connect numerous small, bright yellow circular nodes, creating a web-like structure that suggests a global network or data flow. The nodes and lines are more densely packed on the right side of the image and become sparser towards the left.

 THE **LINUX** FOUNDATION

# Envoy&Istio overview

Ihor Dvoretzkyi, [@idvoretzkyi](#),  
Developer Advocate, CNCF



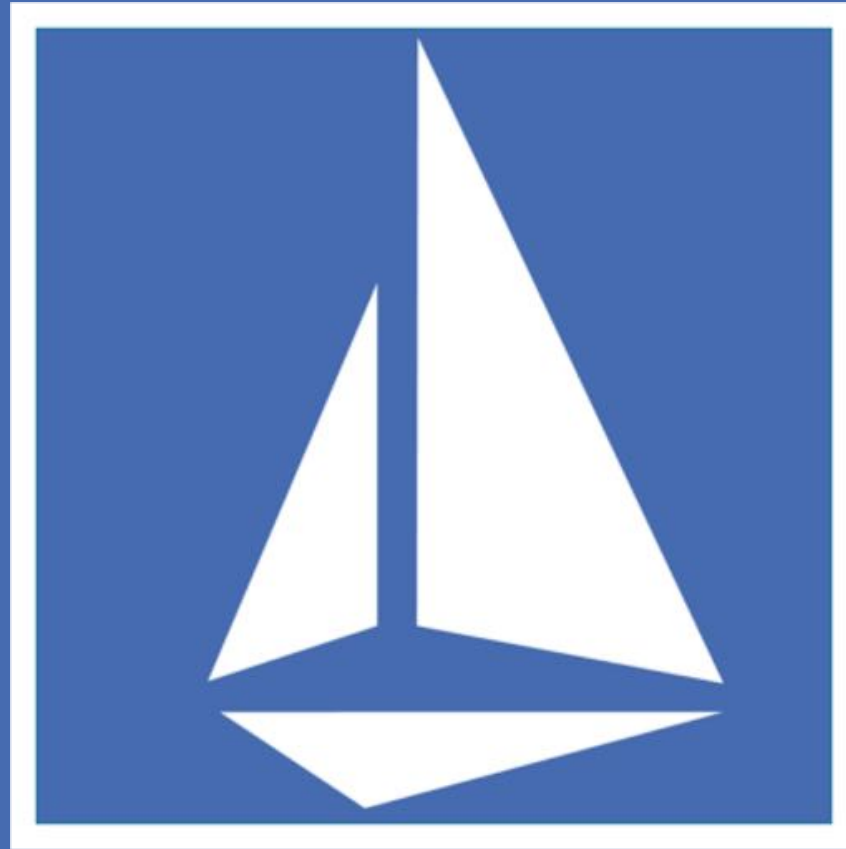


- **Inception** May '15
- **Open sourced** September '16: <https://github.com/envoyproxy/envoy>
- **Joined CNCF** September '17: <https://www.envoyproxy.io/>
- **Users/contributors** (partial list): Lyft, Google, IBM, Verizon, Apple, Microsoft, Pivotal, Red Hat, EBay, Stripe, VSCO, Tencent QQ, Twilio, Yelp ... and more all the time.
- **Integrations** (partial list): Istio (Google/IBM), Nomad (Verizon), Tigera, Covalent, Turbine Labs, Datawire ... more on the way.
- **Lyft deployment**: 10s of thousands of hosts, 100s of services, 3M + mesh RPS.

# State of Service-Oriented Architecture networking

- **Protocols** (HTTP/1, HTTP/2, gRPC, databases, caching, etc.).
- **Infrastructures** (IaaS, CaaS, on premise, etc.).
- Intermediate **load balancers** (AWS ELB, F5, etc.).
- **Observability** output (stats, tracing, and logging).
- Implementations (often partial) of **retry**, **circuit breaking**, **rate limiting**, **timeouts**, and other distributed systems best practices.
- **Authentication** and **Authorization**.
- Per language **libraries** for service calls.

- **Out of process architecture:** Let's do a lot of really hard stuff in one place and allow application developers to focus on business logic.
- **Modern C++ code base:** Fast and productive.
- **L3/L4 filter architecture:** A byte proxy at its core. Can be used for things other than HTTP (e.g., MongoDB, redis, stunnel replacement, TCP rate limiter, etc.).
- **HTTP L7 filter architecture:** Make it easy to plug in different functionality.
- **HTTP/2 first!** (Including **gRPC** and a nifty gRPC HTTP/1.1 bridge).
- **Service discovery and active/passive health checking.**
- **Advanced load balancing:** Retry, timeouts, circuit breaking, rate limiting, shadowing, outlier detection, etc.
- Best in class **observability:** stats, logging, and tracing.
- **Edge proxy:** routing and TLS.





# Istio overview

- Connect, manage and secure microservices
- Separation of concerns: developer and operations
- For both containerized and non-containerized workloads
- Leverage great functionality in Envoy, adding pluggable management and control planes
- Intelligent routing, load balancing, metrics collection, policy enforcement, end-to-end authentication
- Istio 1.0 announced in July'2018





# Istio components

- **Mixer** - Central component that is leveraged by the proxies and microservices to enforce policies such as authorization, rate limits, quotas, authentication, request tracing and telemetry collection.
- **Pilot** - A component responsible for configuring the proxies at runtime.
- **Citadel** - A centralized component responsible for certificate issuance and rotation.
- **Node Agent** - A per-node component responsible for certificate issuance and rotation.



# Istio and Envoy

- **Envoy** - Sidecar proxies per microservice to handle ingress/egress traffic between services in the cluster and from a service to external services
- Providing a rich set of functions like:
  - discovery
  - L7 routing
  - circuit breakers
  - policy enforcement
  - telemetry recording/reporting functions.

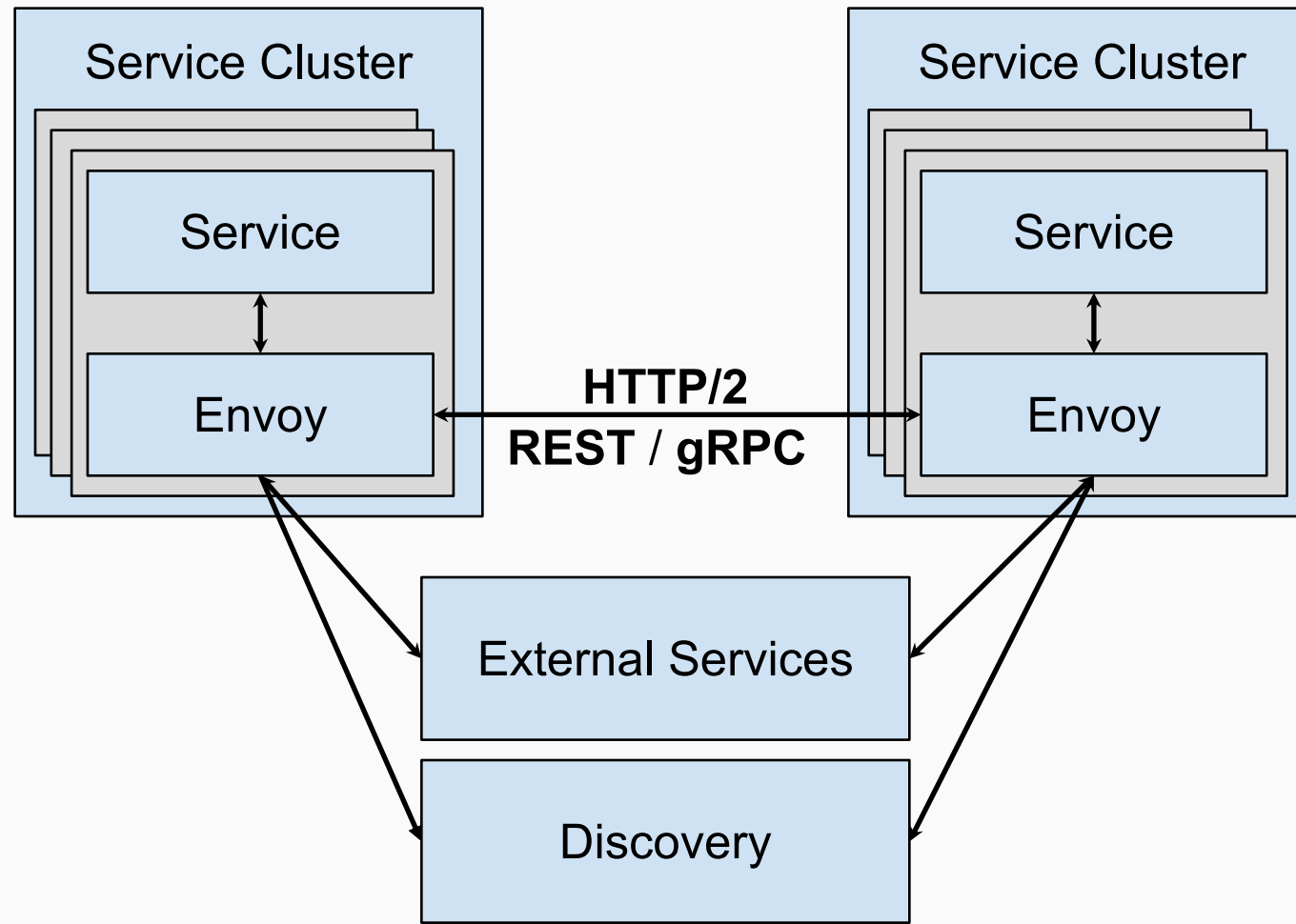
# More details

- **Envoy:**
  - <https://www.envoyproxy.io/>
  - <https://github.com/envoyproxy/envoy>
- **Istio:**
  - <https://istio.io/>
  - <https://github.com/istio/istio>



# Extra

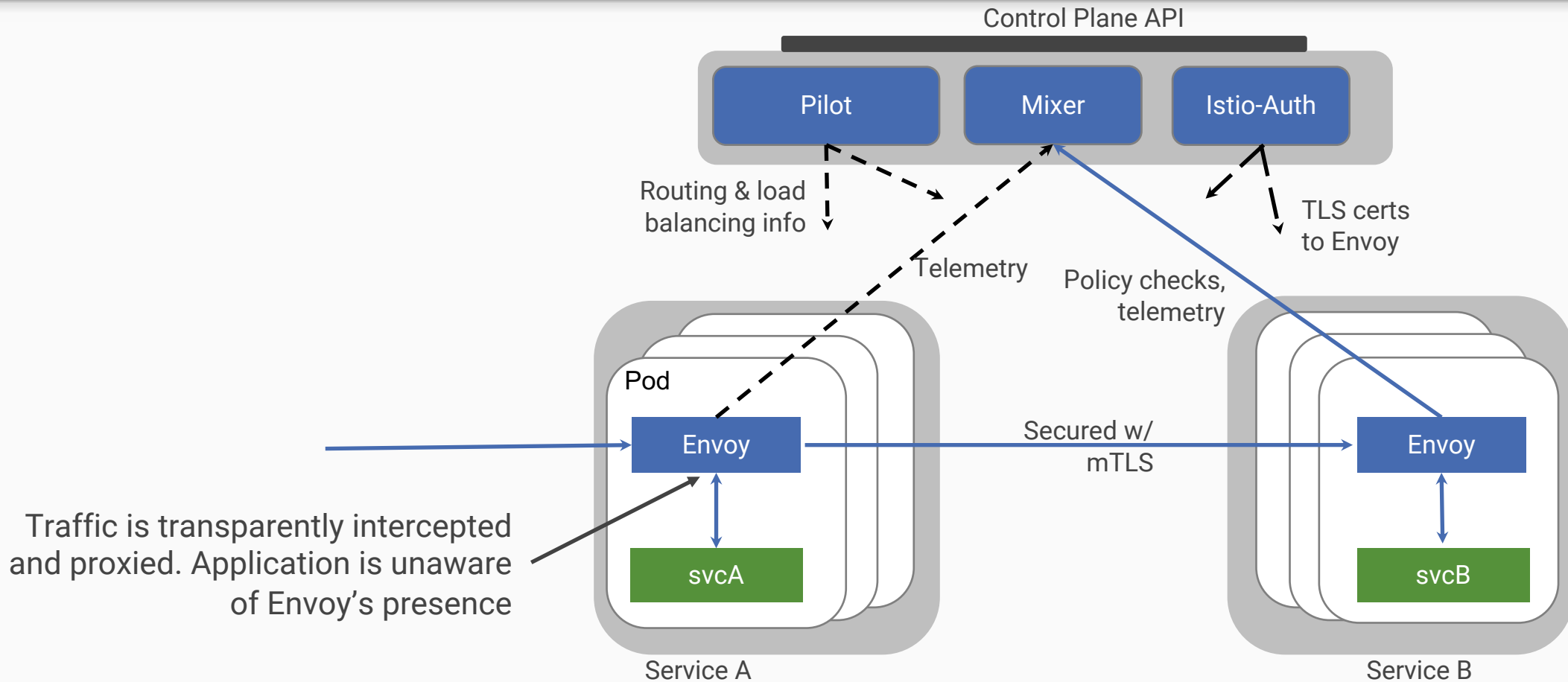
# Envoy overview

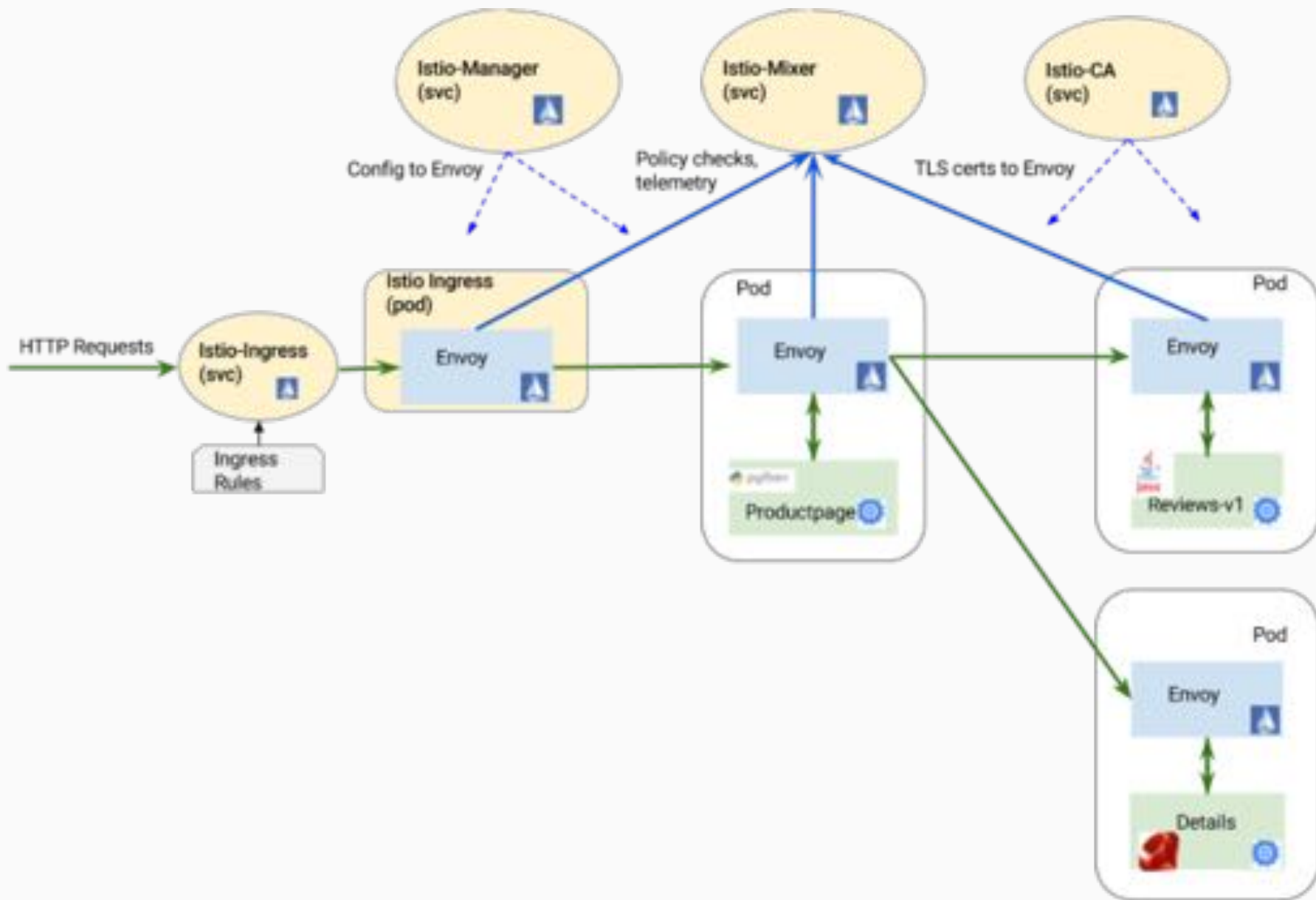






# Architectural overview





# Cross-Cloud CI Overview

Cloud-native Network Functions Seminar

August 28, 2018

# Presented by:

Taylor Carpenter, Vulk Coop

[taylor@vulk.coop](mailto:taylor@vulk.coop)

# Cross-Cloud CI Project Overview

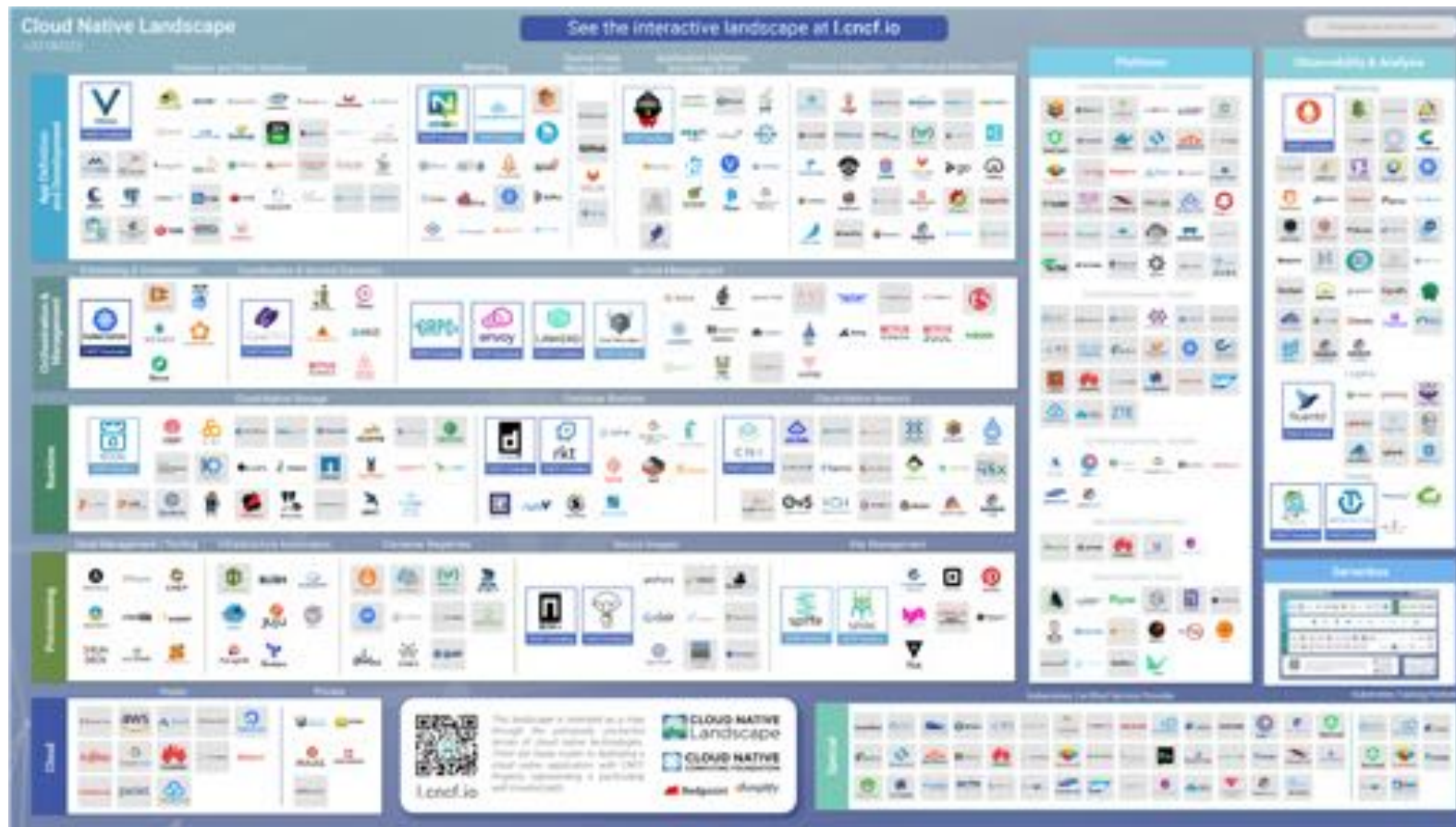
Why?





# Cross-Cloud CI Project Overview

**Why?** CNCF ecosystem is growing rapidly with new projects and cloud providers!



# Cross-Cloud CI Project Overview

## Why?

- The CNCF ecosystem is large, diverse and continues to grow. CNCF would like to ensure cross-project interoperability and cross-cloud deployments of all cloud native technologies and show the daily status of builds and deployments on a status dashboard.



# Cross-Cloud CI Project Overview

## Why?

- The CNCF ecosystem is large, diverse and continues to grow. CNCF would like to ensure cross-project interoperability and cross-cloud deployments of all cloud native technologies and show the daily status of builds and deployments on a status dashboard.

## What?

- Cross-cloud testing system
- Status repository server
- Status dashboard



# Build and provision CNCF projects

## Graduated



Kubernetes  
Orchestration



Prometheus  
Monitoring

## Incubating



OpenTracing  
Distributed Tracing API



Fluentd  
Logging



gRPC  
Remote Procedure Call



containerd  
Container Runtime



rkt  
Container Runtime



CNI  
Networking API



Envoy  
Service Mesh



Jaeger  
Distributed Tracing



Notary  
Security



TUF  
Software Update Spec



Vitus  
Storage



CoreDNS  
Service Discovery



NATS  
Messaging



Linkerd  
Service Mesh



Helm  
Package Management



Rook  
Storage



SPIFFE  
Identity Spec



SPIRE  
Identity



Open Policy Agent  
Policy



CloudEvents  
Serverless



Telepresence  
Tooling



Harbor  
Registry



OpenMetrics  
Metrics Spec



TiKV  
Key/Value Store



# Project CI artifacts and non-CNCF projects

Implemented

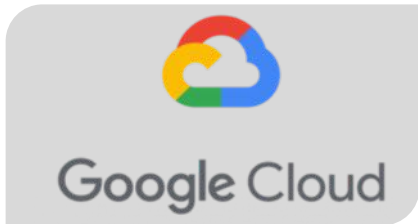
---



# Deploy to public/bare metal/private clouds

## Implemented

---









































































































## In Progress

---











































































































Last updated: 12 hours ago

Project	Build		Release	Deployments						
	Status	Stable Head		AWS	Azure	GCE	IBM Cloud	Bare Metal (Packet)	OpenStack	VMware vsphere
 <b>Kubernetes</b> Orchestration	 SUCCESS	v1.11.2		 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 SUCCESS	03a6d0b		 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
 <b>Prometheus</b> Monitoring	 SUCCESS	v2.3.2		 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 SUCCESS	c663477		 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
 <b>CoreDNS</b> Service Discovery	 SUCCESS	v1.2.0		 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 SUCCESS	b83ed01		 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
 <b>Fluentd</b> Logging	 SUCCESS	v1.2.4		 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 SUCCESS	fd8821a		 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
 <b>Linkerd</b> Service Mesh	 SUCCESS	1.4.6		 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 SUCCESS	b962702		 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
 <b>ONAP</b> Network Automation	 SUCCESS	v1.11		 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 SUCCESS	8ad7083		 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS



Last updated: 1 minute ago

Project	Build		Release	Deployments						
	Status		Stable Head	AWS	Azure	GCE	IBM Cloud	Bare Metal (Packer)	OpenStack	VMware vSphere
 <b>Kubernetes</b> Orchestration	 BUILDING		v1.11.2	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 BUILDING		03a6d0b	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
 <b>Prometheus</b> Monitoring	 BUILDING		v2.3.2	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 BUILDING		c663477	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
 <b>CoreDNS</b> Service Discovery	 BUILDING		v1.2.0	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 BUILDING		b83fd71	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
 <b>Fluentd</b> Logging	 BUILDING		v1.2.4	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 BUILDING		fd8827a	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
 <b>Linkerd</b> Service Mesh	 BUILDING		1.4.6	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 BUILDING		b96270	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
 <b>ONAP</b> Network Automation	 BUILDING		v1.1.1	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 BUILDING		8ad7083	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS



















































































































Created for CNCF by Cross-Cloud




 Report bug




Last updated: 1 minute ago






































Project	Build	Release	Deployments							
			Status	Stable Head	AWS	Azure	GCE	IBM Cloud	Bare Metal (Packer)	OpenStack
 <b>Kubernetes</b> Orchestration	 SUCCESS	v1.11.2	 RUNNING	 RUNNING	 RUNNING	 RUNNING	 RUNNING	 RUNNING	 RUNNING	 RUNNING
	 SUCCESS	03a6d06	 RUNNING	 RUNNING	 RUNNING	 RUNNING	 RUNNING	 RUNNING	 RUNNING	 RUNNING
 <b>Prometheus</b> Monitoring	 SUCCESS	v2.3.2	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 SUCCESS	c663477	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
 <b>CoreDNS</b> Service Discovery	 SUCCESS	v1.2.0	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 SUCCESS	b83bd71	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
 <b>Fluentd</b> Logging	 SUCCESS	v1.2.4	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 SUCCESS	fd8821a	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
 <b>Linkerd</b> Service Mesh	 SUCCESS	1.4.6	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 SUCCESS	b96270	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
 <b>ONAP</b> Network Automation	 SUCCESS	v1.1.1	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 FAILURE	8ad7083	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS

Created for CNCF by Cross-Cloud




 Report bug


Last updated: 1 minute ago

Project	Build	Release	Deployments						
			AWS	Azure	GCE	IBM Cloud	Bare Metal (Packer)	OpenStack	VMware vSphere
 <b>Kubernetes</b> Orchestration	 SUCCESS  SUCCESS	v1.11.2 03a6d0b	 SUCCESS  SUCCESS	 SUCCESS  SUCCESS	 SUCCESS  SUCCESS	 SUCCESS  SUCCESS	 SUCCESS  SUCCESS	 SUCCESS  SUCCESS	 SUCCESS  FAILED
	 <b>Prometheus</b> Monitoring	 SUCCESS  SUCCESS	v2.3.2 c663477	 RUNNING  RUNNING	 RUNNING  RUNNING	 RUNNING  RUNNING	 RUNNING  RUNNING	 RUNNING  RUNNING	 RUNNING  RUNNING
 <b>CoreDNS</b> Service Discovery		 SUCCESS  SUCCESS	v1.2.0 b857d01	 RUNNING  RUNNING	 RUNNING  RUNNING	 RUNNING  RUNNING	 RUNNING  RUNNING	 RUNNING  RUNNING	 RUNNING  RUNNING
	 <b>Fluentd</b> Logging	 SUCCESS  SUCCESS	v1.2.4 fd8871a	 RUNNING  RUNNING	 RUNNING  RUNNING	 RUNNING  RUNNING	 RUNNING  RUNNING	 RUNNING  RUNNING	 RUNNING  RUNNING
 <b>Linkerd</b> Service Mesh		 SUCCESS  SUCCESS	1.4.6 b96270	 RUNNING  RUNNING	 RUNNING  RUNNING	 RUNNING  RUNNING	 RUNNING  RUNNING	 RUNNING  RUNNING	 RUNNING  RUNNING
	 <b>ONAP</b> Network Automation	 SUCCESS  FAILED	v1.1.1 8ad7083	 RUNNING  RUNNING	 RUNNING  RUNNING	 RUNNING  RUNNING	 RUNNING  RUNNING	 RUNNING  RUNNING	 RUNNING  RUNNING

Created for CNCF by Cross-Cloud CI




 Report bug




Last updated: 1 minute ago

Project	Build	Release	Deployments						
			Status	Stable Head	AWS	Azure	GCE	IBM Cloud	Bare Metal (Packet)
 <b>Kubernetes</b> Orchestration	 SUCCESS	v1.11.2	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 SUCCESS	03a6d0b	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 FAILURE
 <b>Prometheus</b> Monitoring	 SUCCESS	v2.3.2	 FAILURE	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 SUCCESS	c663477	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 FAILURE
 <b>CoreDNS</b> Service Discovery	 SUCCESS	v1.2.0	 FAILURE	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 SUCCESS	b87ed87f	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 FAILURE
 <b>Fluentd</b> Logging	 SUCCESS	v1.2.4	 FAILURE	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 SUCCESS	fd8821a	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 FAILURE
 <b>Linkerd</b> Service Mesh	 SUCCESS	1.4.6	 FAILURE	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 SUCCESS	t96Z79f	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 FAILURE
 <b>ONAP</b> Network Automation	 SUCCESS	v1.1.1	 FAILURE	 SUCCESS	 SUCCESS	 FAILURE	 SUCCESS	 SUCCESS	 FAILURE
	 FAILURE	8ad7083	 NA	 NA	 NA	 NA	 NA	 NA	 NA



Failed Job #72130 triggered 5 minutes ago

```

Installing terminal-table 1.8.0
Fetching jenkins_api_client 1.5.1
Installing jenkins_api_client 1.5.1
Fetching vcr 4.4.0
Installing vcr 4.4.0
Fetching netmock 3.3.0
Installing netmock 3.3.0
Bundle complete! 34 Gemfile dependencies, 48 gems now installed.
Bundled gems are installed into `usr/local/bundle`
Post-install message from aruba:
Use an ruby 1.9.7
* Make sure you add something like that to your "Gemfile". Otherwise you will
  get cucumber > 3 and this will fail on ruby 1.9.7

gem "cucumber", "~> 1.3.39"

With aruba >= 1.9 there will be breaking changes. Make sure to read https://github.com/cucumber/aruba/blob/master/HISTORY.md for 1.9.8
$ cd -
~/build/onap/so
$ ./onapcli-service/bia/build_pipeline build_status --integration=INTEGRATION_NAME --release-type=INTEGRATION_RELEASE_TYPE so
Build status: success
$ ./onapcli-service/bia/build_pipeline download_container --integration=INTEGRATION_NAME --release-type=INTEGRATION_RELEASE_TYPE so
./onapcli-service/bia/crosscloud/ciservice/onap/build_pipeline.rb:79:in "head_container_image_tag": undefined method "split" for nil:NilClass (NoMethodError)

   from ./onapcli-service/bia/crosscloud/ciservice/onap/build_pipeline.rb:19:in "container_image_tag"
   from ./onapcli-service/bia/crosscloud/ciservice/onap/build_pipeline.rb:318:in "container_artifact_url"
   from ./onapcli-service/bia/crosscloud/ciservice/onap/build_pipeline.rb:399:in "download_container"
   from ./onapcli-service/bia/build_pipeline.rb:18:in "download_container"
   from /usr/local/bundle/gems/thor-0.20.4/lib/thor/command.rb:27:in "run"
   from /usr/local/bundle/gems/thor-0.20.4/lib/thor/invocation.rb:126:in "invoke_command"
   from /usr/local/bundle/gems/thor-0.20.4/lib/thor.rb:387:in "dispatch"
   from /usr/local/bundle/gems/thor-0.20.4/lib/thor/base.rb:446:in "start"
   from ./onapcli-service/bia/build_pipeline.rb:95:in "run"
===== Job failed, exit code 1
    
```

**container**

Duration: 2 minutes 31 seconds  
Runner #18

**Trigger**  
Total: 2032  
**DASHBOARD\_API\_HOST\_PORT**  
devapi.cnfd  
**CROSS\_CLOUD\_YML**  
<https://raw.githubusercontent.com/cross-cloud-configuration/master/cross-cloud.yml>

Commit 4086f43a   
Merge: Fix the structure in the doc for 'soap'

Failed Pipeline #12567 from master  
package

 Failed container





Last updated: 1 minute ago

Project	Build		Release	Deployments						
	Status		Stable Head	AWS	Azure	GCE	IBM Cloud	Bare Metal (Packet)	OpenStack	VMware vSphere
 <b>Kubernetes</b> Orchestration	 SUCCESS		v1.11.2	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 SUCCESS		03a6d0b	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 FAILURE
 <b>Prometheus</b> Monitoring	 SUCCESS		v2.3.2	 FAILURE	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 SUCCESS		c663477	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 FAILURE
 <b>CoreDNS</b> Service Discovery	 SUCCESS		v1.2.0	 FAILURE	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 SUCCESS		b87ed87f	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 FAILURE
 <b>Fluentd</b> Logging	 SUCCESS		v1.2.4	 FAILURE	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 SUCCESS		fd8821a	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 FAILURE
 <b>Linkerd</b> Service Mesh	 SUCCESS		1.4.6	 FAILURE	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS
	 SUCCESS		t96Z79f	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 SUCCESS	 FAILURE
 <b>ONAP</b> Network Automation	 SUCCESS		v1.1.1	 FAILURE	 SUCCESS	 SUCCESS	 FAILURE	 SUCCESS	 SUCCESS	 FAILURE
	 FAILURE		8ad7083	 FAILURE	 FAILURE	 FAILURE	 FAILURE	 FAILURE	 FAILURE	 FAILURE

Created for CNCF by Cross-Cloud CI




 Report bug


Failed Job #72184 triggered 5 minutes ago

```

waiting for 1 ready nodes.. 0 ready nodes, 0 registered. Retrying.
No resources found.
waiting for 1 ready nodes.. 0 ready nodes, 0 registered. Retrying.
No resources found.
waiting for 1 ready nodes.. 0 ready nodes, 0 registered. Retrying.
No resources found.
waiting for 1 ready nodes.. 0 ready nodes, 0 registered. Retrying.
No resources found.
waiting for 1 ready nodes.. 0 ready nodes, 0 registered. Retrying.
No resources found.
Detected 0 ready nodes, found 0 nodes out of expected 1. Your cluster may not be fully functional.
No resources found.
$ helm init --service-account tiller
Creating /root/.helm
Creating /root/.helm/repository
Creating /root/.helm/repository/cache
Creating /root/.helm/repository/local
Creating /root/.helm/plugins
Creating /root/.helm/starters
Creating /root/.helm/cache/archive
Creating /root/.helm/repository/repositories.yaml
Adding stable repo with URL: https://kubernetes-charts.storage.googleapis.com
Adding local repo with URL: http://127.0.0.1:8878/charts
$ helm HOME has been configured at /root/.helm.

Tiller (the Helm server-side component) has been installed into your Kubernetes Cluster.

Please note: by default, tiller is deployed with an insecure "allow unauthenticated users" policy.
For more information on securing your installation see: https://docs.helm.sh/using_helm/#securing-your-helm-installation
Happy Helming!
$ sleep 30
$ kubectl rollout status -n deployment/tiller-deploy --namespace=kube-system
waiting for deployment spec updates to be observed...
ERROR: Job failed: iteration took longer than 300 seconds
                
```

**Provisioning**

Duration: 60 minutes 6 seconds  
Runner #10

---

**Trigger**

Token: 4bdf

**CLOUD**  
vsphere

**SOURCE**  
12557

**PROJECT\_ID**  
54

**PROJECT\_BUILD\_PIPELINE\_ID**  
12557

**TARGET\_PROJECT\_ID**  
54

**TARGET\_PROJECT\_NAME**  
kubernetes

**TARGET\_PROJECT\_COMMIT\_REF\_NAME**  
master

**DASHBOARD\_API\_HOST\_PORT**  
devapi.cncf.io

**CROSS\_CLOUD\_YML**  
<https://raw.githubusercontent.com/cross-configuration/master/cross-cloud.yml>

Commit: def96cd

[Fix the decommissioning bug](#)



🕒 Last updated: 1 minute ago

Project	Build		Deployments						
	Status	Stable Head	AWS	Azure	GCE	IBM Cloud	Bare Metal (Packet)	OpenStack	VMware vSphere
 <b>Kubernetes</b> Orchestration	<span>🟢 SUCCESS</span>	v1.11.2	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>
	<span>🟢 SUCCESS</span>	03a6d0b	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🔴 FAILURE</span>
 <b>Prometheus</b> Monitoring	<span>🟢 SUCCESS</span>	v2.3.2	<span>🔴 FAILURE</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>
	<span>🟢 SUCCESS</span>	c663477	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🔴 FAILURE</span>
 <b>CoreDNS</b> Service Discovery	<span>🟢 SUCCESS</span>	v1.2.0	<span>🔴 FAILURE</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>
	<span>🟢 SUCCESS</span>	b87ed87f	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🔴 FAILURE</span>
 <b>Fluentd</b> Logging	<span>🟢 SUCCESS</span>	v1.2.4	<span>🔴 FAILURE</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>
	<span>🟢 SUCCESS</span>	fd8821a	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🔴 FAILURE</span>
 <b>Linkerd</b> Service Mesh	<span>🟢 SUCCESS</span>	1.4.6	<span>🔴 FAILURE</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>
	<span>🟢 SUCCESS</span>	t96Z79f	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🔴 FAILURE</span>
 <b>ONAP</b> Network Automation	<span>🟢 SUCCESS</span>	v1.1.1	<span>🔴 FAILURE</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🔴 FAILURE</span>	<span>🟢 SUCCESS</span>	<span>🟢 SUCCESS</span>	<span>🔴 FAILURE</span>
	<span>🔴 FAILURE</span>	8ad7083	<span>🟡 WARN</span>	<span>🟡 WARN</span>	<span>🟡 WARN</span>	<span>🟡 WARN</span>	<span>🟡 WARN</span>	<span>🟡 WARN</span>	<span>🟡 WARN</span>

Created for CNCF by Cross-Cloud CI

🔗 🐦 📧 📌 Report bug





passed Job #67042 triggered about 5 minutes ago

```

Starting k9s on display :80 with rev 1208x5804x14
Executing mount tests at log level INFO
-----
OpenShift ETE
-----
OpenShift ETE.Bootstrap
-----
OpenShift ETE.Bootstrap.Testsuites
-----
OpenShift ETE.Bootstrap.Testsuites.Health-Check :: Testing ecamp components are...
-----
Reside: P90 Health Check | PASS |
-----
OpenShift ETE.Bootstrap.Testsuites.Health-Check :: Testing ecamp compon... | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
-----
OpenShift ETE.Bootstrap.Testsuites | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
-----
OpenShift ETE.Bootstrap | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
-----
OpenShift ETE | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
-----
Output: /share/logs/ETE_5758/output.txt
Log: /share/logs/ETE_5758/log.html
Report: /share/logs/ETE_5758/report.html
$ #314
Job succeeded
    
```

**etd**

Duration: 40 minutes 58 seconds  
Runner: #18

---

**Trigger**

Token: 3c63

**KUBERNETES**  
YX8pVimYyc2vtbjogdJEKY2x1c3RlcmM5CCE

**CLOUD**  
openstack

**SOURCE**  
11218

**ORG**  
onap

**CHART\_REPO**  
onap-amsterdam

**PROJECT\_ID**  
53

**CHART**  
so

**PROJECT\_BUILD\_PIPELINE\_ID**  
11218

**TARGET\_PROJECT\_ID**  
53

**TARGET\_PROJECT\_NAME**  
so

**TARGET\_PROJECT\_COMMIT\_REF\_NAME**  
v1.3.1

**DEPLOYMENT\_NAME**



# Technology Overview

# CI System Technology Overview

## Unified CI/CD platform:

- GitLab

## Cross-cloud provisioning:

- Terraform, Cloud-init, and per cloud K8s configuration

## App deployments:

- K8s manifest management with Helm

## E2e tests:

- Custom containers + Helm

## Automated builds and deployments:

- Git + per project yaml configuration





# Dashboard Technology Overview

## Frontend:

- Vue.js

## Status repository:

- Elixir and Erlang

## Automated builds and deployments:

- Git + per project yaml configuration



Q&A

# How to Collaborate

## Attend CI WG meetings:

- <https://github.com/cncf/wg-ci>
- 4th Tuesday of the month at 11:00am Pacific Time
- Next Meeting is on Tuesday, Sept 28th

## Subscribe to the CNCF CI public mailing list:

- <https://lists.cncf.io/g/cncf-ci-public>

## Create issues on GitHub:

- <https://github.com/crosscloudci/cross-cloud/issues>

## Join the #cncf-ci channel on slack:

- Request invite at <https://slack.cncf.io/>
- [Cloud-native.slack.com](https://cloud-native.slack.com)



# Connect with Cross-Cloud CI



**@crosscloudci**



**@crosscloudci**



**crosscloudci@vulk.coop**



For more details and an in-depth demo, please contact Dan Kohn & Cross-Cloud CI team at CNCF booth at #OSSNA18

Also presenting at:

- KubeCon + CloudNativeCon China
  - November 13-14, Shanghai