Common NFVI Telco Taskforce

Technical F2F Work Shop – January 13-16, 2020

RI Workstream: Key Updates

Facilitator: Mike Fix, Rajesh Rajamani, Sridhar Rao, Cedric Ollivier

THE LINUX FOUNDATION

GSMA

Content & MVP Targets

THE LINUX FOUNDATION

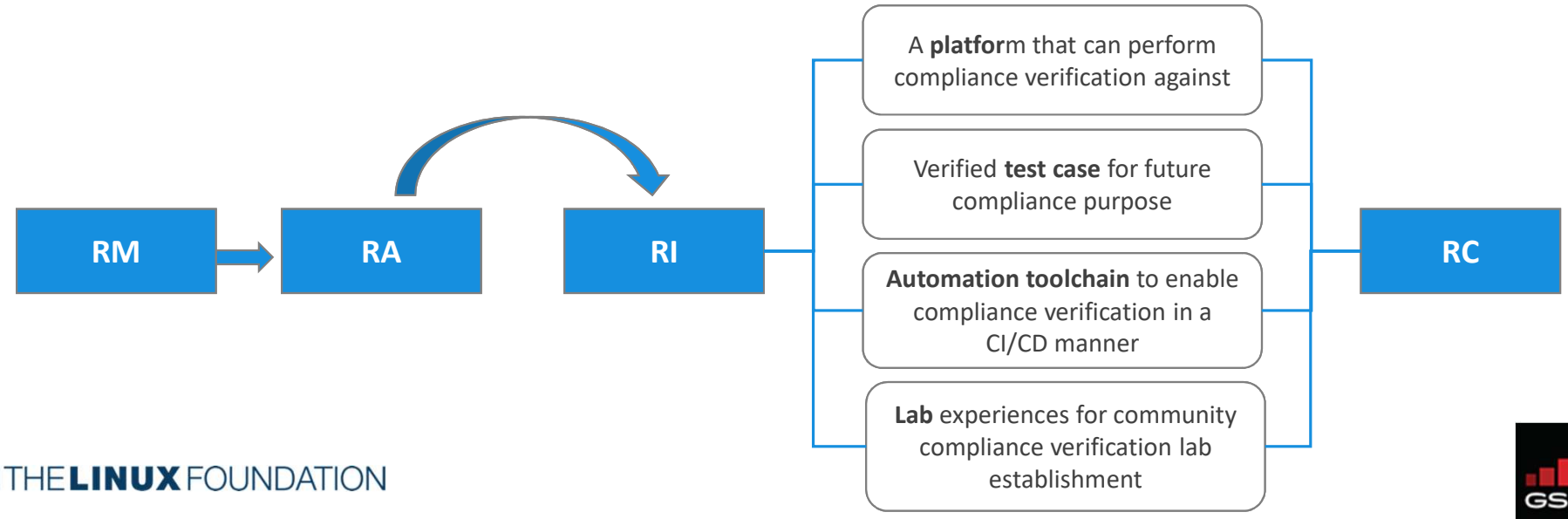
GSMA

WS Goals

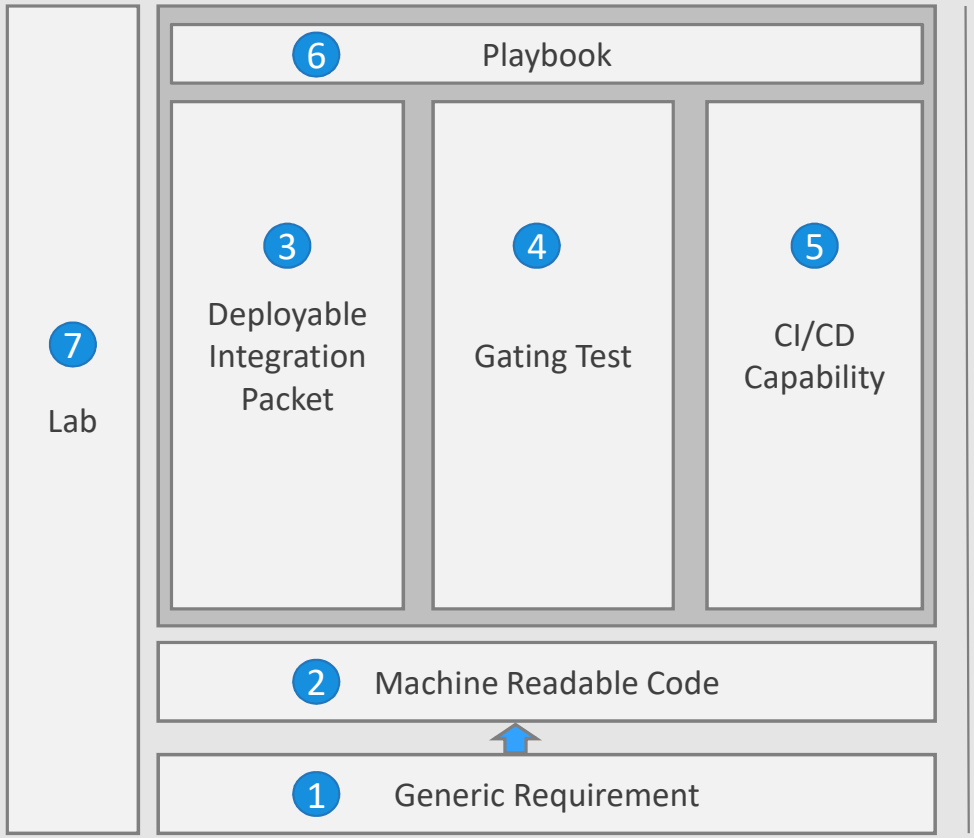
From Document to Code

Infrastructure Realization: As the first WS to translate documents into real code, RI is responsible to provide RC, and even further to the telco industry, with real deployable tools for compliance testing/POC/Lab setup

Snezka RI-Alpha Deliverables: Initial Content, Establish Lab, Install Software, Draft Cookbook



Approach



1. Provide generic requirement for the reference implementation
2. Provide practical machine readable code to reveal and enable these generic requirements
3. Provide deployable integration packet for implementation in accordance to RM and RA
4. Provide gating test for the implementation to make sure it meets RM & RA
5. Provide CI/CD capability for the implementation so as to benefit future release evolvement and compliance testing
6. Conclude in a playbook for all the necessary operations of the above capabilities
7. Establish lab environment to demonstrate the above capabilities & provide further guidance to compliance labs

Progress to Date | Key Accomplishments

Requirements

Initial content of requirements for Reference Implementation, including:

- Labs
- Tooling
- Installers, Releases
- Automation Requirements

Labs

RI Lab requirements specifies:

- Required support for various NFVI profiles defined by RM
- Capabilities required for Compute, Storage, ToR and Controller nodes
- Lab Topology & Power/cooling requirements

Cookbook

Initial operational guide ("Cookbook")

- Lab Access/Connectivity
- Integration of installers & components
- Deployment validations

RI Dev Chapter 7

- Link to Chapter 7

Pod Implementation

Intel POD 10

- Airship 1.0 & OpenStack Ocata
- OPNFV Iruya Release
- Performance benchmarking with Funtest
- 1 jump host, 3 Controllers & 2 compute

POD 15

- Same configuration as POD 10
- Currently facing infrastructure issue and being fixed

RI Requirement Updates

Progress: Initial Content Creation

Initial Content: https://github.com/cntt-n/CNTT/tree/master/doc/ref_impl/cntt-ri

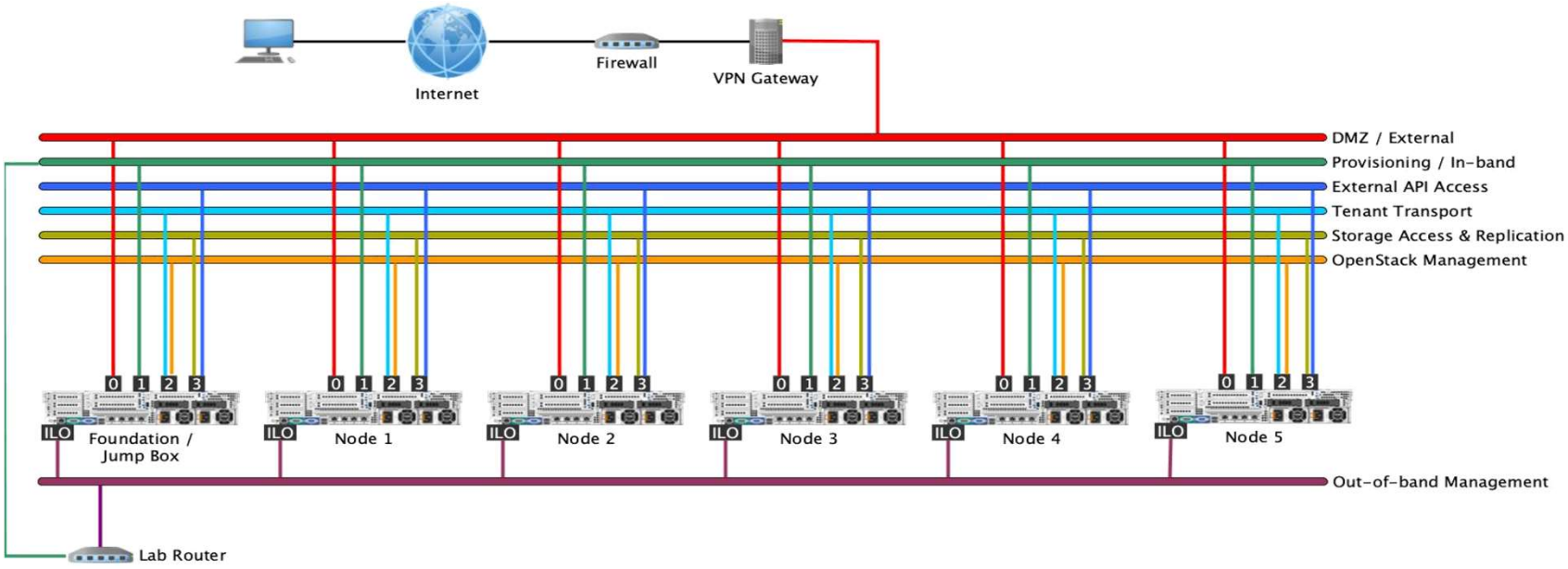
Chapter	Status
Chapter 01 - Overview	Complete
Chapter 02 - Reference Implementation Requirements	Complete
Chapter 03 - NFVI + VNF Target State and Specification	Still Developing Contents
Chapter 04 - Lab Requirements	Still Developing Contents
Chapter 05 - Installer Requirements	Still Developing Contents
Chapter 06 - Lab Operations	Still Developing Contents
Chapter 07 - Integration	Still Developing Contents
Chapter 08 - Gap analysis & Development	Still Developing Contents

Progress: Lab Requirements

Lab Requirements

- Must provide capabilities to support the NFVI Profiles specified by CNTT RM
- Must withstand component failures, provide for network path redundancy & Controller HA
- Must provide Jump Host, Compute & Storage hosts, Controllers & ToR with adequate compute, memory, storage, acceleration & network bandwidth capabilities (minimum 6 hosts, recommended 12 hosts in lab)
- Rack space requirement - at least 19U, sufficient for 10-16 servers
- Power requirement - ~ 10KWatts needed for max 16 hosts & 2 ToR

Recommended Lab topology



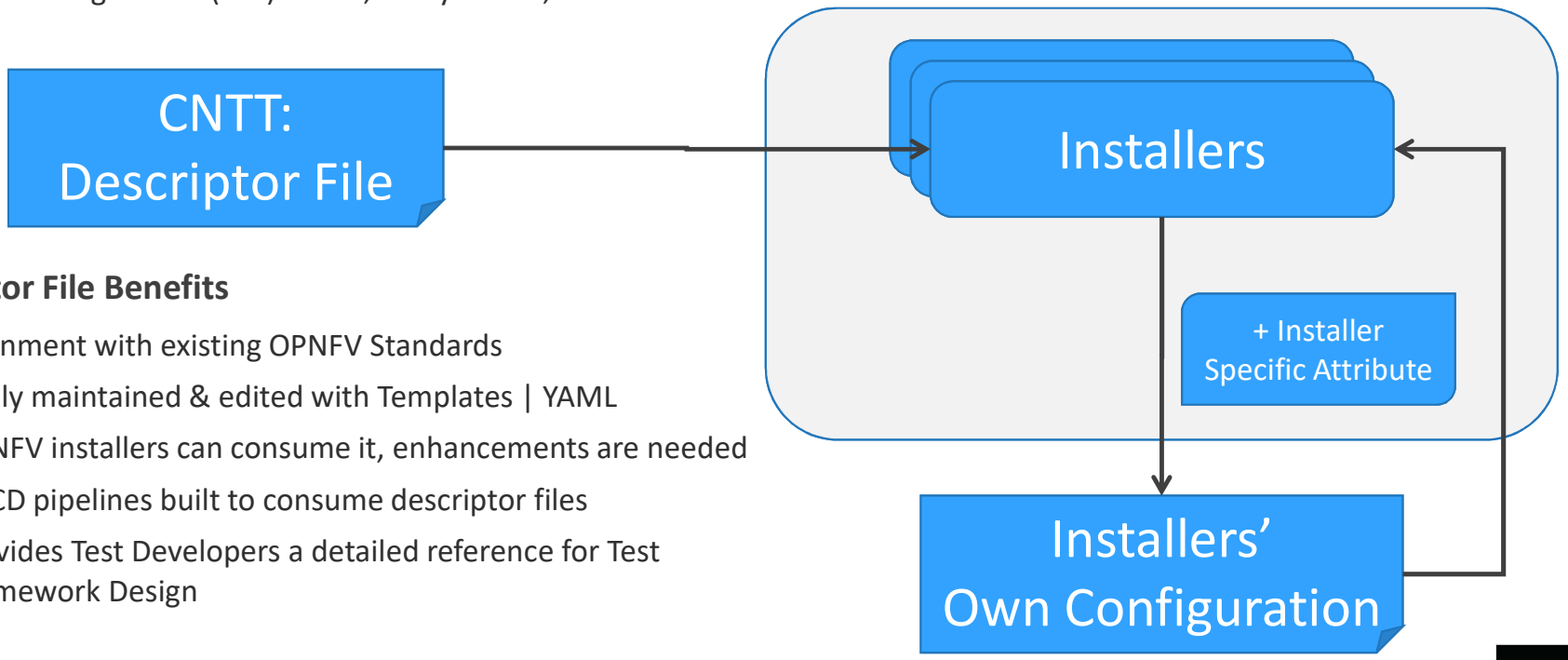
Rajesh

Progress: Installer Requirements

Descriptor File Contents *(Under Consideration)*

Hardware Configuration	Network Configuration (VLAN settings, IP)
Server Template	Storage Configuration
Server Instances	NTP, Proxy Server, etc.
Software Configuration (VM)	

Installer accepts the CNTT format descriptor file as input content



Descriptor File Benefits

- Alignment with existing OPNFV Standards
- Easily maintained & edited with Templates | YAML
- OPNFV installers can consume it, enhancements are needed
- CI/CD pipelines built to consume descriptor files
- Provides Test Developers a detailed reference for Test Framework Design

Installer translates content to own configuration

THE LINUX FOUNDATION

Sridhar

*** Currently under discussion**

GSMA

Progress: Installer Requirements (Continued)

Next Steps: New Content Need to Add

Content to be added as a prerequisite in a new commitment

- Hardware validation requirements(BIOS, RAID, CPU, Memory, IPMI, NIC)
- Network configuration on switch and router.(i.e.: vlan design and set for network plane for BMC and/or public access vlan)

Complementary information to add:

- Metadata (Lab Owner, Location, Purpose),
- Components (SDN-C, NFV features),
- Common Network Info (IP range, Subnets),Deployment Tools (installers) and Hardware Prerequisites (SRIOV, DPDK). (network information: in range or static? need to further discussion)

See GitHub for additional details: https://github.com/cntt-n/CNTT/blob/master/doc/ref_impl/cntt-ri/chapters/chapter05.md

Sridhar

NFVI Required State

Objective

Provide traceability to RM/RA requirements for RI realization & Manifest Validations, and validated for Compliance.

Requirements Sources

- RM Ch 5: https://github.com/cntt-n/CNTT/blob/master/doc/ref_model/chapters/chapter05.md
- RA Ch 2 Infra: https://github.com/cntt-n/CNTT/blob/master/doc/ref_arch/openstack/chapters/chapter02.md
- RA Ch 5 NFVI API: https://github.com/cntt-n/CNTT/blob/master/doc/ref_arch/chapters/chapter05.md

RI Traceability

Chp03:NFVI + VNF Target State & Specification documents required state of NFVI, as mapped to RM/RA requirements, **yielding the "readiness" of the NFVI:**

- Sec 3.2 VNF profile – describes what every workload must declare
  - instance type (B/N/C), flavor
  - network options, persistent storage and acceleration extensions
- Sec 3.3 NFVI SW profile – list of features defining the NFVI software layer
  - virtual compute (e.g., CPU allocation ratio, huge page etc),
  - storage (e.g., IOPS criteria and replication factor etc)
  - network optimization and acceleration options
- Sec 3.4 NFVI required status - architect requirements for VIM, API, tenants, LCM, security, & expected versions

Progress: Reference Implementation Cookbook

Cookbook Outline | Progress

Goal

Utilize Lab, Install, and Validation Experience with RI POC Lab for initial offering to create content

Purpose

Accelerate, NFVI+VNF Certifications through standardized lab setup & validations

Chapters Created

- Requirements Gathering
- Access and Connectivity (Lab)
- Available Installers
- Airship (as RI alpha installer)
  - Descriptor File Preparations
  - Deployment Installer & Install Steps
- Deployment Validations

Progress to Date

- CICD Toolchain in place
- Initial Draft Complete
- Inclusive of Descriptor Files, Install Manifests (AirShip), & Deployment Validations (Functest)

Next Release Plans

- Update Documentation:
  - CICD Toolchain (process & scripts)
  - Jenkins Setup & Job Creation
  - Compliance Validation (steps, process)
- Perform Validation & Optimize Accordingly

Link to Additional Information

Reference Implementation Achievements | Targets for Alpha

POD10 | POD15: Install State & Availability

	Pod-10	Pod-15
Owner	Trevor Cooper	Mike Fix
Purpose	CNTT testing PoC	CNTT CIRV
Pod-Topology	https://wiki.opnfv.org/display/pharos/Intel+POD10	https://wiki.opnfv.org/display/pharos/Intel+POD15
Status	Ready to Use. All dashboards up and running. Test Tools deployment in progress	Manifests are ready. Infra Issue is being addressed by Team-Intel.
Installer	Airship	Airship
Infrastructure Definition	https://github.com/opnfv/airship/tree/master/site/intel-pod10	https://github.com/opnfv/airship/tree/master/site/intel-pod15
OpenStack Installation Type	OOK: OpenStack on Kubernetes	OOK: OpenStack on Kubernetes
Networking support	Openvswitch and SRIOV	Openvswitch and SRIOV
Functest Compatibility	~90%-95%	NA
Performance Benchmarking	In-Progress	NA
Usecases	NI – SRIOV. Compute Intensive	NI – SRIOV. Compute Intensive
GAPS	Low on Storage	No SSD for data disk.
Deployed and Maintained By	Sridhar@OPNFV-Airship	Sridhar@OPNFV-Airship

Initial NFVI Installation – The Installer

Infrastructure defined using Airship Manifests, and deployed using Airship Installer.

- How to create manifests: https://wiki.opnfv.org/display/AIR/Airship+Manifest+Creation+For+New+Sites

Why Airship?

- Opensource, Declarative, Flexibile and Consistent
- OPNFV-Airship has simplified manifest creation
- OPNFV-Airship has also simplified IP planning (Internal)
- Airship has captured all aspects of deployment
  - Strategy, definition, networking (networks, NetworkLinks and NIC-Mapping), profiling, security, software, bootactions, baremetal-provisioning, LMA serverside (even RHOSP-13 Director does not support any serverside monitoring)

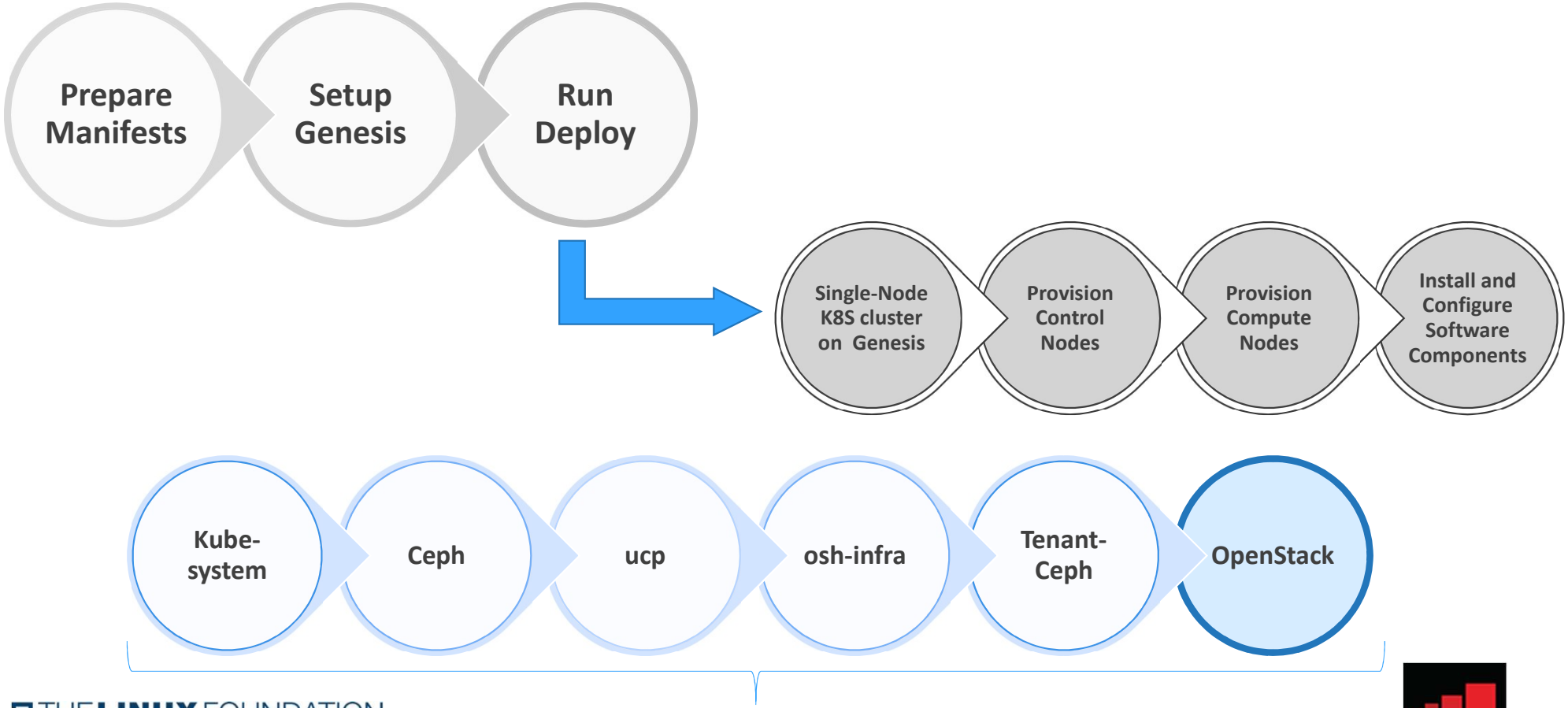
1.0 Gaps?

- Ubuntu Only, as MaaS is used
- SDN controller deployment is not supported
- MANO (VNFM/NFVO) deployment is not supported
- Airship 2.0 may fix these issues

Contacts with other, related, installer has proved to be unsuccessful

- Kayobe (similar to Airship – OOK)

Initial NFVI Installation – The Process, Summarized



The Software Components (Namespaces w.r.t K8s)

Initial NFVI Installation - Learnings

Physical-networking (connectivity and configuration), and Internet-Access Issues can cause delays

- Pings and LLDP tools can help in former
- At times the access to repos can be slow
  - **Offline repository, possibly in jumphost, would be helpful**

Getting Ceph right is very important

- It would help to have SSDs for data disks and preferably separate journal disks (NVMes)
- **In Intel-Pods, this is not the case**

Open IT tickets early and follow it up – provide as much details as possible

- **Different time zones (Deployment and IT/Admin) will not help**

Though, with its well-defined process, flow and detailed logs, troubleshooting any deployment failures is relatively easier, **OPNFV-Airship should come-up with Troubleshooting Guide**

- Some simple 'inspection' and 'validation' tools would be very helpful

OPNFV PDF is not used for manifest creation (yet)

- Shortcomings = Automation Challenges
- Enhancement to PDF/IDF/SDF is under consideration. Once done, auto manifest creation can be achieved

Release Notes | Issues

Release Notes, & Issues

Labs

- Intel POD10 and POD15, Pharos Compliance Labs utilized for Alpha release

Installer

- Installation performed manually using AirShip manifests.
- Target next release for consumption of detailed machine readable description files.

Software

- RI based on OpenStack Ocata vs. RA requirements for PIKE
- Installer manifests (i.e. OpenStack Helm (OSH)) supports Pike and Stein (today).
- All core services/capabilities installed, except Live Migration and Resize Services per OSH.

Validations

- Confirmed RA1 Ch 5 (API) feature capability is exposed per OSH Ocata
- Successfully audited installation and recommended next release changes: (1) increase Neutron Workers, (2) automatic removal of :80 reference in API end-points, and (3) use Nova 2.1 vs 2.0

Next Steps

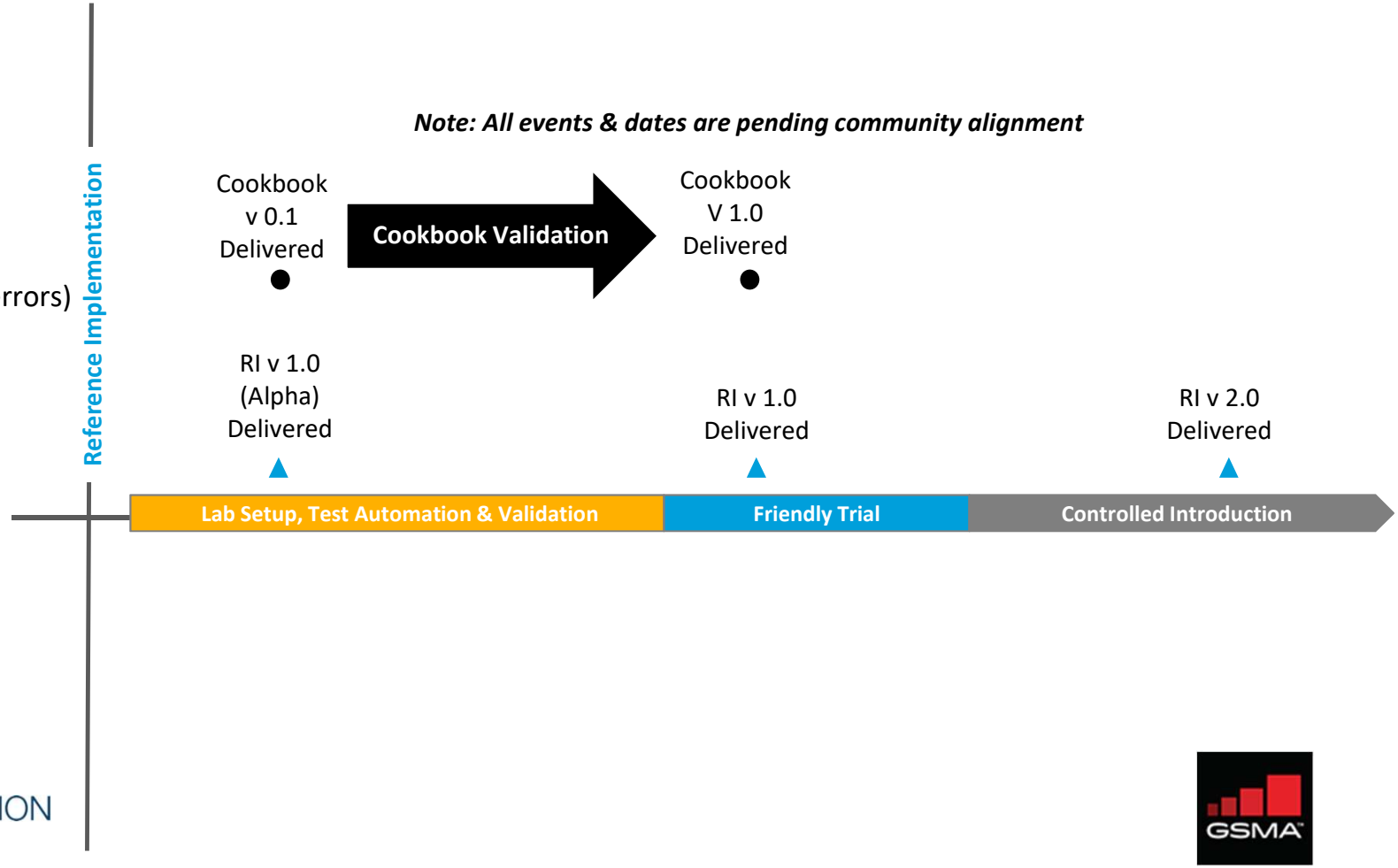
Current Status

Completed:

- RM | RA | RI Requirements
- Lab Requirements
- Initial Lab Secured
- S/W Deployed | Config
- Smoke Test | Sanity
- Continuous Deployment (with errors)

In-Progress

- Manifest > PDF & IDF
- Complete Lab PoC & Deliver Lab
- Create Cookbook & RI Topology artifact
- Implement PoC Key Learnings



THE LINUX FOUNDATION

Mike Fix

GSMA

Appendix

Initial NFVI Installation – Access

Dashboards	URL	Credentials
Horizon	dashboard-airship.intel-pod10.opnfv.org	admin/password123
Grafana	grafana-airship.intel-pod10.opnfv.org	grafana/password123
Kibana	kibana-airship.intel-pod10.opnfv.org	elasticsearch/password123
Nagios	nagios-airship.intel-pod10.opnfv.org	nagios/password123

Progress: Installer Requirements(1/4)

Ref	sub-category	Description
req.gen.ins.01	Installer	Installer **must** accept a descriptor file to finish deployment.
req.gen.ins.02	Installer	Installer implementation **must** validate the descriptor file with schema.
req.gen.ins.03	Installer	Any existing installer implementation **may** need adaption for the descriptor file.
req.gen.ins.04	Installer	Installer **may** support reporting the deployment progress status.
req.gen.des.01	Descriptor	Descriptor file **must** include hardware resource configuration, software configuration.
req.gen.des.02	Descriptor	Descriptor file **may** include additional extending configuration.

Reference Implementation

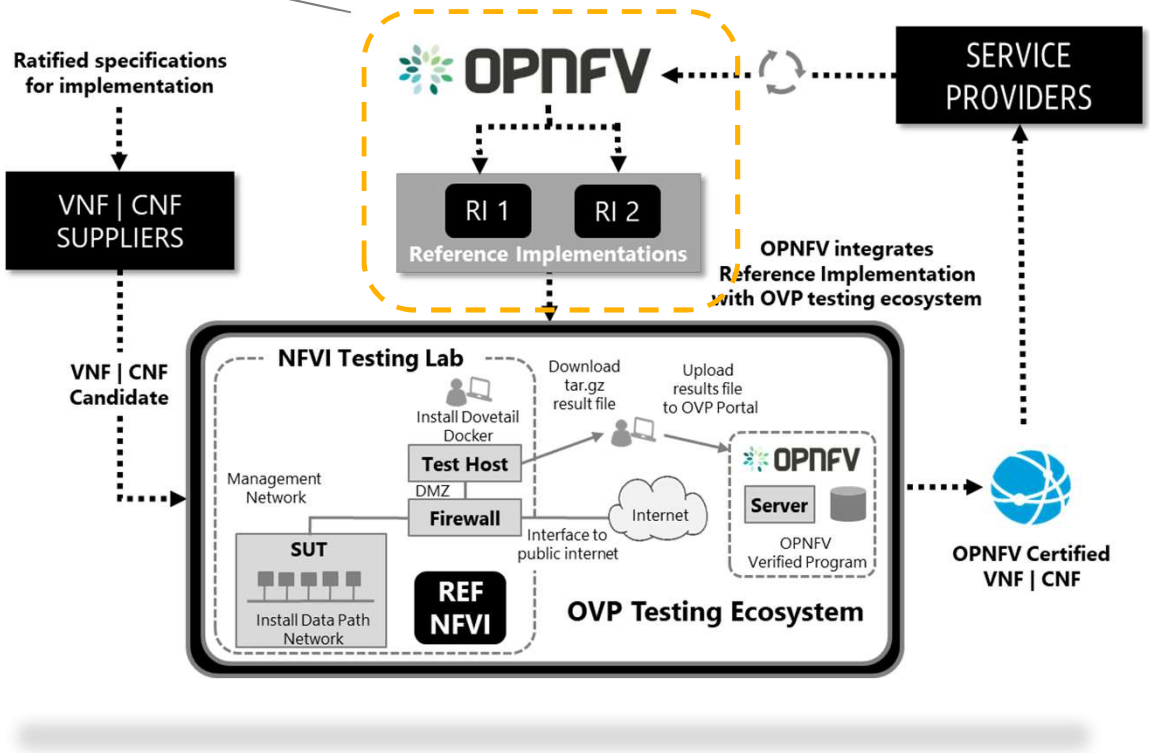
Objective
Implement & deploy based on the design & configurations of each Reference Architecture

Goals

- Provide CNTT verification labs with standard hardware/software for cloud platform & VNF certification
- Providing Reference Implementation leveraging Reference VNFs (“Golden VNFs”) for interoperability testing
- Collaborate with OPNFV on approach & tools, aligning with CNTT community requirements
- Establish hardware | software manifests

Target Delivery

January 2020 (v 1.0- Alpha)
Aligns with Reference Architecture # 1 (OpenStack)



THE LINUX FOUNDATION

GSMA

Reference Implementation Approach | Outcome

Actions Underway	Outcomes
Generate eco-system requirements for Reference Implementation, including: • Labs • Tooling • Installers, Releases • Automation Requirements	Efficient and effective quality approach while aligning with CNTT objectives.
Create detailed description file definitions for installers, VNF features, etc..	User follows the steps or procedures to achieve an RI identical to the lab utilized by CNTT
Create detailed Lab criteria & operations	Optimized approach and aligned with CVC lab setup, operations, & support
Create a cookbook for the first RI, which includes detailed steps for the deployment and configuration	User follows the steps or procedures to achieve an RI identical to the lab utilized by CNTT
Create a detailed design for automation for deployment & testing	Reference Implementation is integrated with CVC ecosystem, including test cases & framework
Create continuous integration & delivery pipeline for Reference Implementation	Alignment with OPNFV CI to generate CNTT Reference Implementation in a continuous manner
Perform gap analysis to determine required actions for existing eco-system within LFN/OPNFV community projects	Required test cases & test frameworks are identified & created in collaboration with upstream communities

THE LINUX FOUNDATION

GSMA

Reference Implementation: Lab Strategy & Plan

Goal

Select a secure, stable, and configurable lab, enabling automated reference implementation (RI) validations

Vision & Recommendation

- ✓ **Establish at least 2 community labs** (leverage testing & HW expertise)
  - ✓ Supports multiple RI validations in parallel
  - ✓ Readily available supplier test apparatus & expertise
  - ✓ Leverage geo diversity
- ✓ **Leverage Lab as a Service (UNH-IOL) labs**
- ✓ **Promote 3rd Party Vendor Labs for Scalability**

Qualifications – Must be Satisfied for Lab Selection

- **Available** with Outage Contingency (4-nine uptime)
- **Current & Stable** (current patch sets)
- Demonstrated **Integration with the OVP Ecosystem**
- **Secure** (physical, and logical)
- **Configurable with minimal downtime**
- **Effortless onboarding process**

RI-1 Lab Chosen (Alpha)

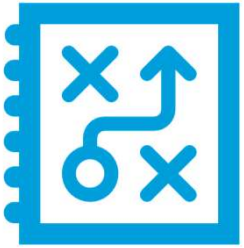
- Intel POD10 PoC & POD15 Target State
- 3 x Controller Nodes
- 10 x Compute Nodes (4 general + 2 per B/N/C)
- 1 x Jump Host
- 1 x Spine Switch and 2 x Leaf Switch in 1 x 48u Rack

THE LINUX FOUNDATION

GSMA

Reference Implementation Challenges

- Community resources (engagement & contributions)
- **OPNFV capabilities & alignment with CNTT objectives**
  - Ability to scale to demand
- Alignment with other communities
- Limited support structure for defects | issues (best-effort community Triage)
- Agreement & Support to build a normalized POD/Infrastructure Descriptor File for installers
- Closure on target location and quantity of code repos needed
- Consensus to create a long-term delivery Work Group (WG) for LCM
- Need automated BareMetal validations



CNTT will maintain ownership of the Reference Implementation until a satisfactory level of support, stability, & maturity is attained

THE LINUX FOUNDATION

GSMA

POD10 | POD15: Install State & Availability

	Pod-10	Pod-15
Owner	Trevor Cooper	TBD
Purpose	CNTT testing PoC	CNTT RI
Pod-Topology	https://wiki.opnfv.org/display/pharos/Intel+POD10	https://wiki.opnfv.org/display/pharos/Intel+POD15
Status	Ready to Use. All dashboards up and running. Test Tools deployment in progress	Manifests are Ready. Infra Issue is being addressed by Team-Intel.
Installer	Airship	Airship
Infrastructure Definition	https://github.com/opnfv/airship/tree/master/site/intel-pod10	https://github.com/opnfv/airship/tree/master/site/intel-pod15 [Coming Soon]
Openstack Installation Type	OOK: Openstack on Kubernetes	OOK: Openstack on Kubernetes
Networking support	Openvswitch and SRIOV	Openvswitch and SRIOV
Functest Compatibility	~90%-95%	Yet to be performed
Performance Benchmarking	In-Progress	Yet to be performed
CNTT RA/RM Compatibility	Yet to Assess. Lack to requirement specification	NA
Usecases	NI – SRIOV. Compute Intensive.	NI-SRIOV. Compute Intensive
GAPS	Low on Storage. Yet to assess CNTT RM/RA compatibility due to lack of manifest validation specification	No SSD for datadisk.
Deployed and Maintained By	Sridhar@OPNFV-Airship	Sridhar@OPNFV-Airship

POD 10 Configuration Details

Installer:

- Airship version v1.0
- Airship TreasureMap version v1.6

Openstack:

- Openstack version: Ocata
- Installation type: Openstack on Kubernetes

Other software versions & locations:

- https://github.com/airshipit/treasuremap/blob/v1.6/global/software/config/versions.yaml

Infrastructure definition / descriptor files:

- https://github.com/opnfv/airship/tree/master/site/intel-pod10

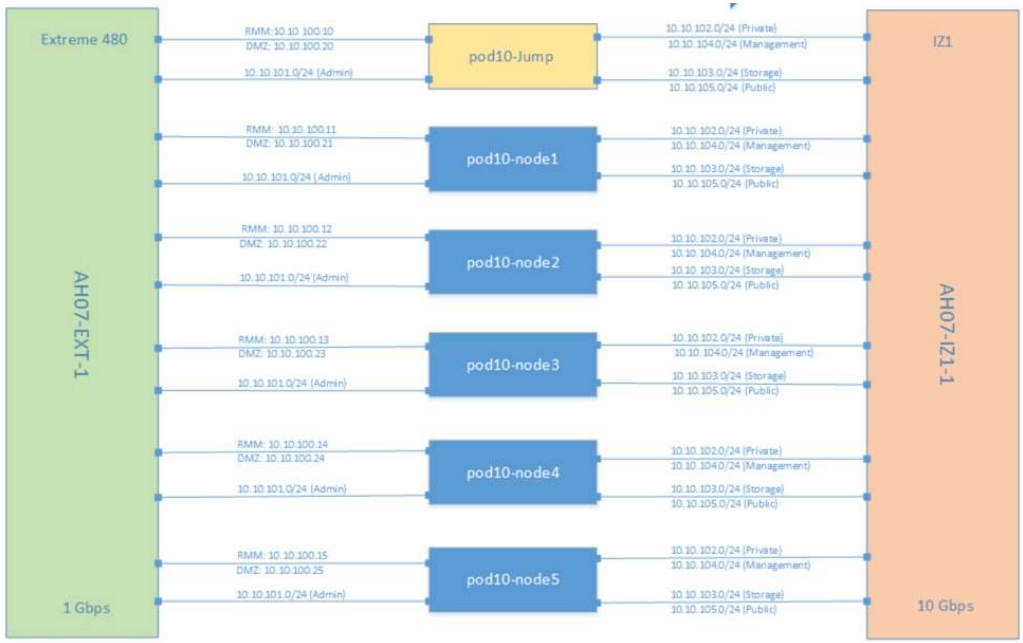
Hardware details:

Node description	Compute	Memory	Storage
pod10-jump	2xE5-2699	64GB	3TB (Sata) 180 (SSD)
pod10-node1	2xE5-2699	64GB	3TB (Sata) 180 (SSD)
pod10-node2	2xE5-2699	64GB	3TB (Sata) 180 (SSD)
pod10-node3	2xE5-2699	64GB	3TB (Sata) 180 (SSD)
pod10-node4	2xE5-2699	64GB	3TB (Sata) 180 (SSD)
pod10-node5	2xE5-2699	64GB	3TB (Sata) 180 (SSD)

Test Tooling:

- OPNFV Iruya Release
- https://wiki.opnfv.org/display/SWREL/Iruya
- OPNFV projects used for testing
  - Yardstick
  - Functest
    - Rally
    - Tempest
  - Dovetail

POD 10 Topology



THE LINUX FOUNDATION

GSMA

POD 15 Configuration Details

Installer:

- Airship version v1.0
- Airship TreasureMap version v1.6

Openstack:

- Openstack version: Ocata
- Installation type: Openstack on Kubernetes

Other software versions & locations:

- https://github.com/airshipit/treasuremap/blob/v1.6/global/software/config/versions.yaml

Infrastructure definition / descriptor files:

- https://github.com/opnfv/airship/tree/master/site/intel-pod15 [Coming Soon]

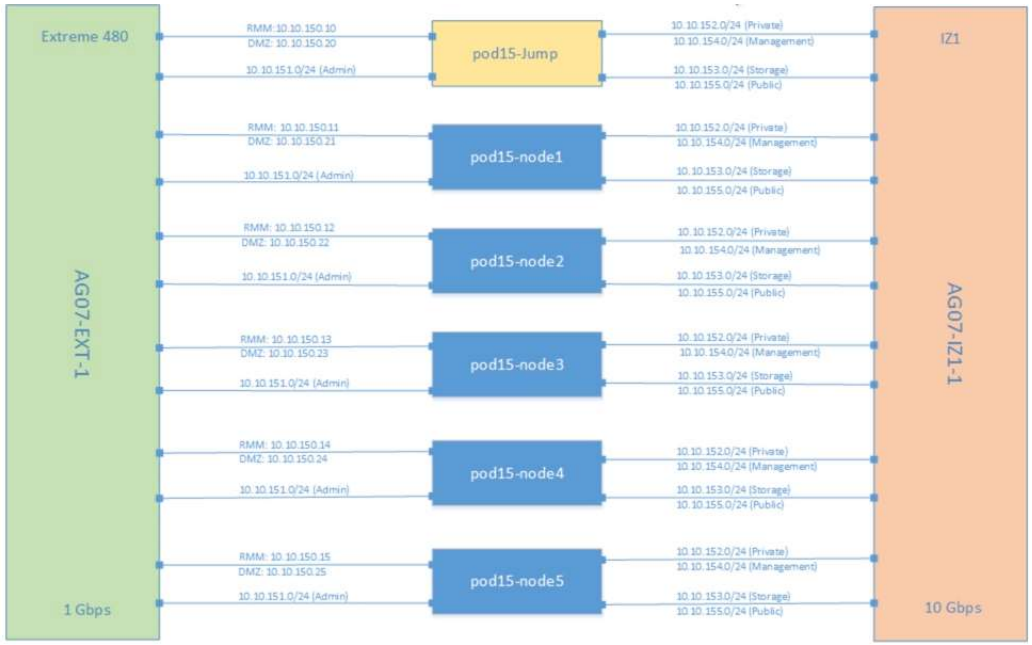
Hardware details:

Node description	Compute	Memory	Storage
pod15-jump	2xE5-2699	64GB	3TB (Sata) 180 (SSD)
pod15-node1	2xE5-2699	64GB	3TB (Sata) 180 (SSD)
pod15-node2	2xE5-2699	64GB	3TB (Sata) 180 (SSD)
pod15-node3	2xE5-2699	64GB	3TB (Sata) 180 (SSD)
pod15-node4	2xE5-2699	64GB	3TB (Sata) 180 (SSD)
pod15-node5	2xE5-2699	64GB	3TB (Sata) 180 (SSD)

Test Tooling:

- OPNFV Iruya Release
- https://wiki.opnfv.org/display/SWREL/Iruya
- OPNFV projects used for testing
  - -Yardstick
  - -Functest
    - Rally
    - Tempest
  - -Dovetail

POD 15 Topology



THE LINUX FOUNDATION

GSMA