



LF NETWORKING

Developer & Testing Forum

E2E Nephio R1 workload orchestration

**Vikas Kumar, Sandeep
Sharma, Aarna Networks**

<https://lfnetworking.org>



Kubernetes-based intent-driven automation of network functions and the underlying infrastructure

“Swimlanes”

1. Infrastructure
2. Workload (network function)
3. Workload configuration

Nephio R1 Components

PackageVariant
Controller

PackageRevision
Controller

ConfigSync

Nephio Management Cluster

Nephio Controllers

Cluster API

Cluster Bootstrap Controllers

Generic Specializers

ipam-specializer

Repository controller

NAD Controller

Package Approver

- **Concepts**

- Config Injection
- Package Specialization
- Condition Choreography

Config Injection



```
Spec:  
  upstream:  
    package: srcpkg  
    ...  
  downstream:  
    package: desiredpkg  
    ...  
  injectors:  
    - name: edge01  
    ...
```

1. Apply PackageVariant Object

Config injection is the process of injecting the contextual information to the downstream packagerevision resources.

This information is consumed by the KRM functions in the pipeline for configuring the packages.

The source of the configuration are in cluster objects.

This slide explains this with in-cluster workloadCluster objects containing the contextual information about target clusters, this information is used by KRM functions for allocation resources like IP addresses for example.



3. Injection procedure

2. Clone and create draft

1. process plugin points: parse through all the resources in packagerevision and find the resources with injection required.

2. Parse the pv.spec.injectors and find the kubernetes objects, edge01 of kind workloadCluster in this case.

3. Fetch the workloadcluster object and inject it to the packagerevision desiredpkg in the downstream repo

4. Update Kptfile with,
conditions:
- type: config.injection.WorkloadCluster.workload-cluster
status: "True"
message: injected resource "edge01" from cluster
reason: ConfigInjected

```
workload-cluster.yaml: |-  
  apiVersion: infra.nephio.org/v1alpha1  
  kind: WorkloadCluster  
  metadata:  
    name: workload-cluster  
  annotations:  
    config.kubernetes.io/local-config:  
    "true"  
    kpt.dev/config-injection: required
```

One KRM in resource list has config-injection required

```
edge01  
kind: workloadCluster
```

This object contains the contextual information of the cluster which will later be consumed by specializers fn/controllers

Nephio Cluster

Specialization Pipeline and Condition Choreography

pipeline:

mutators:

- image: gcr.io/kpt-fn/apply-replacements:v0.1.1
configPath: apply-replacements-namespace.yaml
- image: docker.io/nephio/upf-deploy-fn:v1.0.1
- image: docker.io/nephio/interface-fn:v1.0.1

Kptfile

interface-fn

dependency

upf-deploy-
ent-fn

status:

Conditions:
Type: interface
Status: not ready
Type: vlanclaim
Status: not ready
Type: ipclaim
Status: not ready
....

Kptfile

6. Functions Keep updating conditions in kpt file as resources are allocated and created

4. Reconciles the package revision, checks conditions and invokes corresponding functions

5. Consult the clustercontext object and allocate ip vlan via the backend controllers based on the kind of CNIs in the target cluster

1. saving package revision triggers pipeline.
2. Functions create inventory of for, owns and watch
3. Functions Update kptfile conditions and save.

Eventually all resources are allocated, and all conditions in the KPTfile are set to Ready by the condition choreography done by KRM fns/controllers. The result is the deployment of UPFDeployment curated CR in the edge01 cluster

ipclaim-fn

vlanclaim-fn

nad

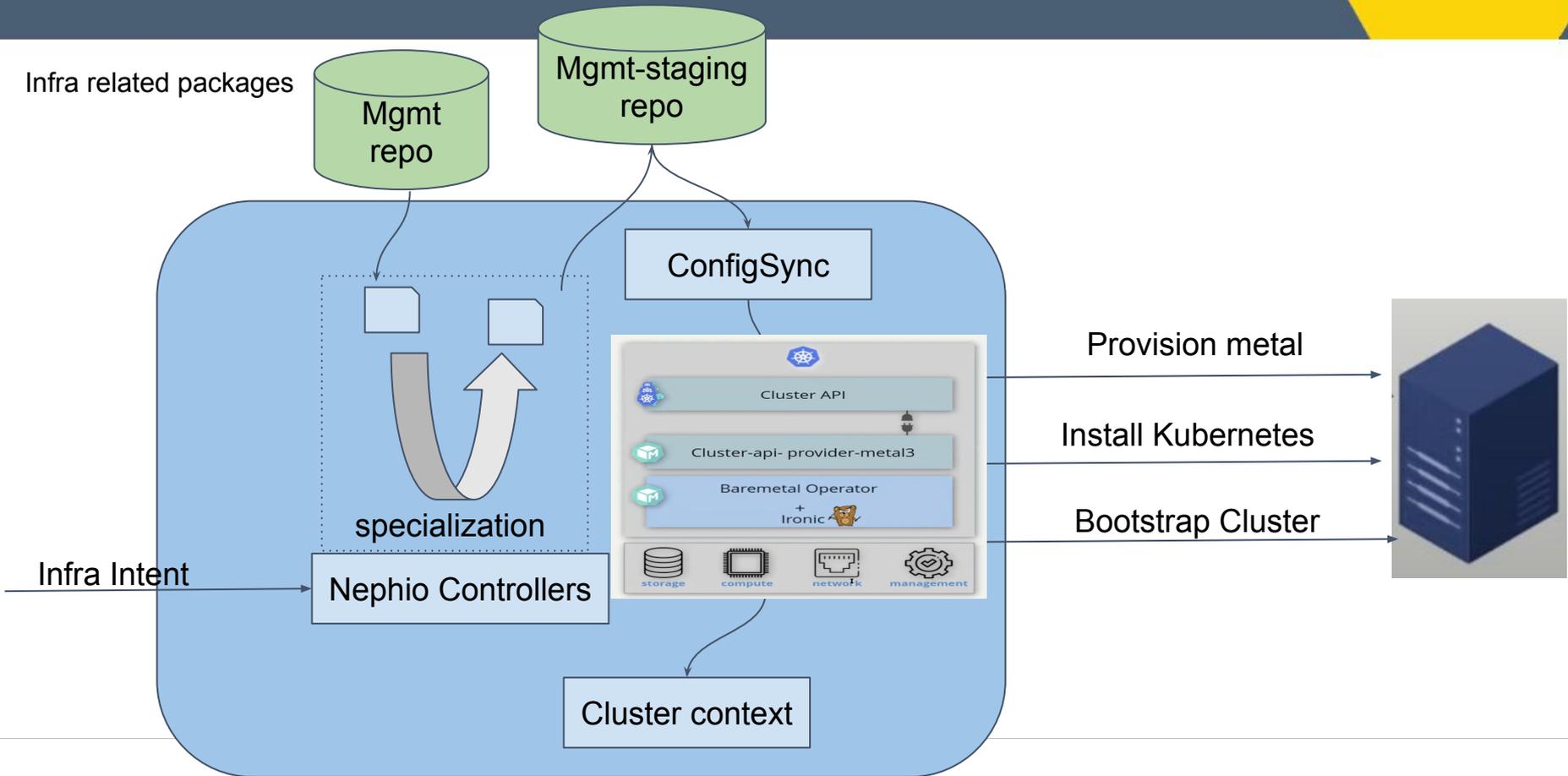
edge01
kind: workloadCluster

Ipam, vlan backend

Our Use Case

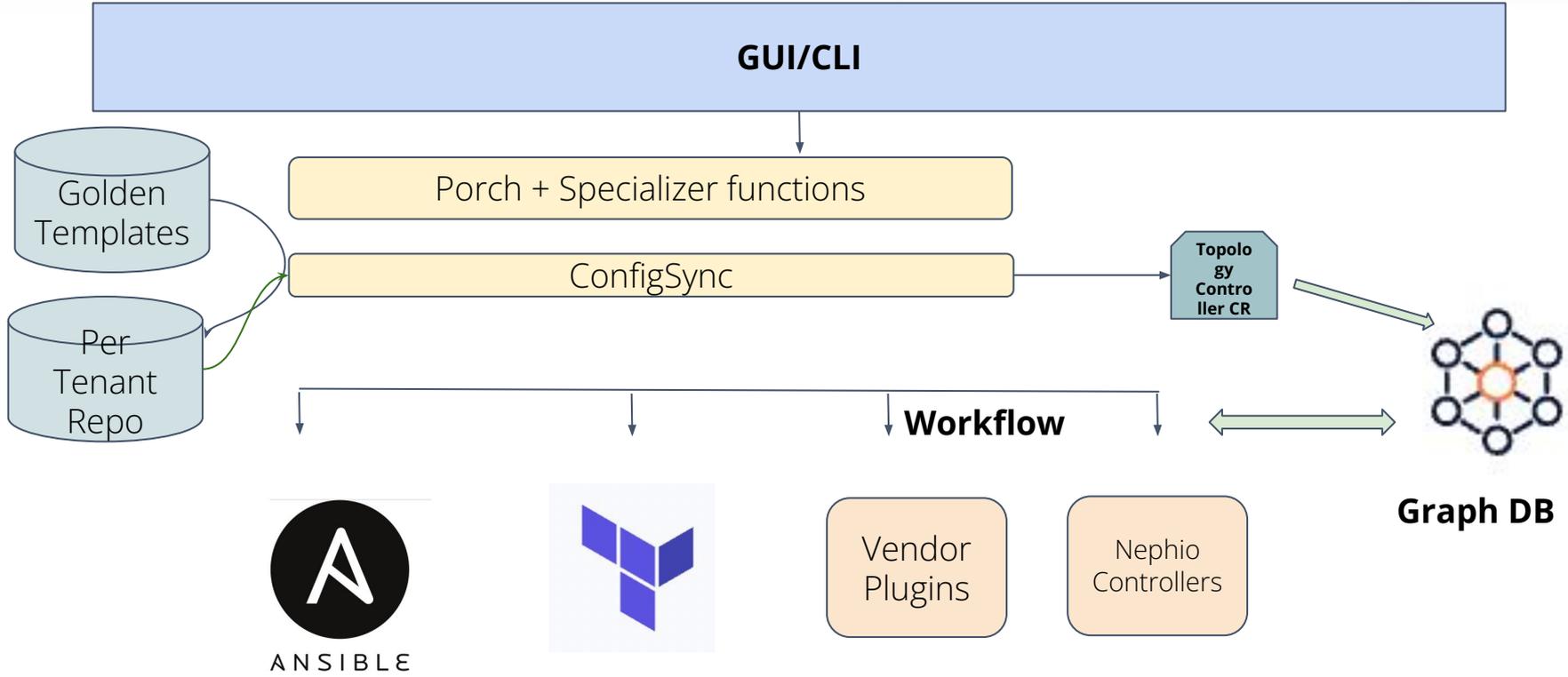
- Orchestrate, discover and manage
 - Infrastructure - metal, kubernetes clusters
 - Workloads - VMs, Containers, VMs as pods (kubevirt)

Bare-metal provisioning

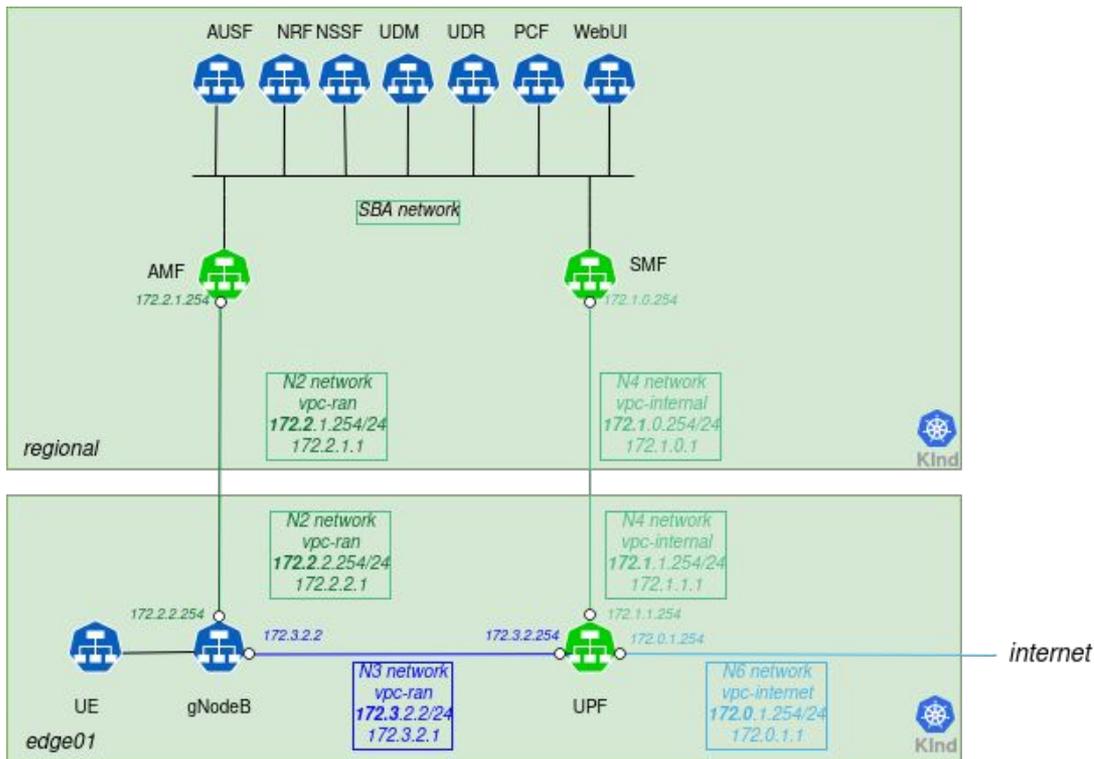


Workload Orchestration

E2E Service Intent



R1-Demo

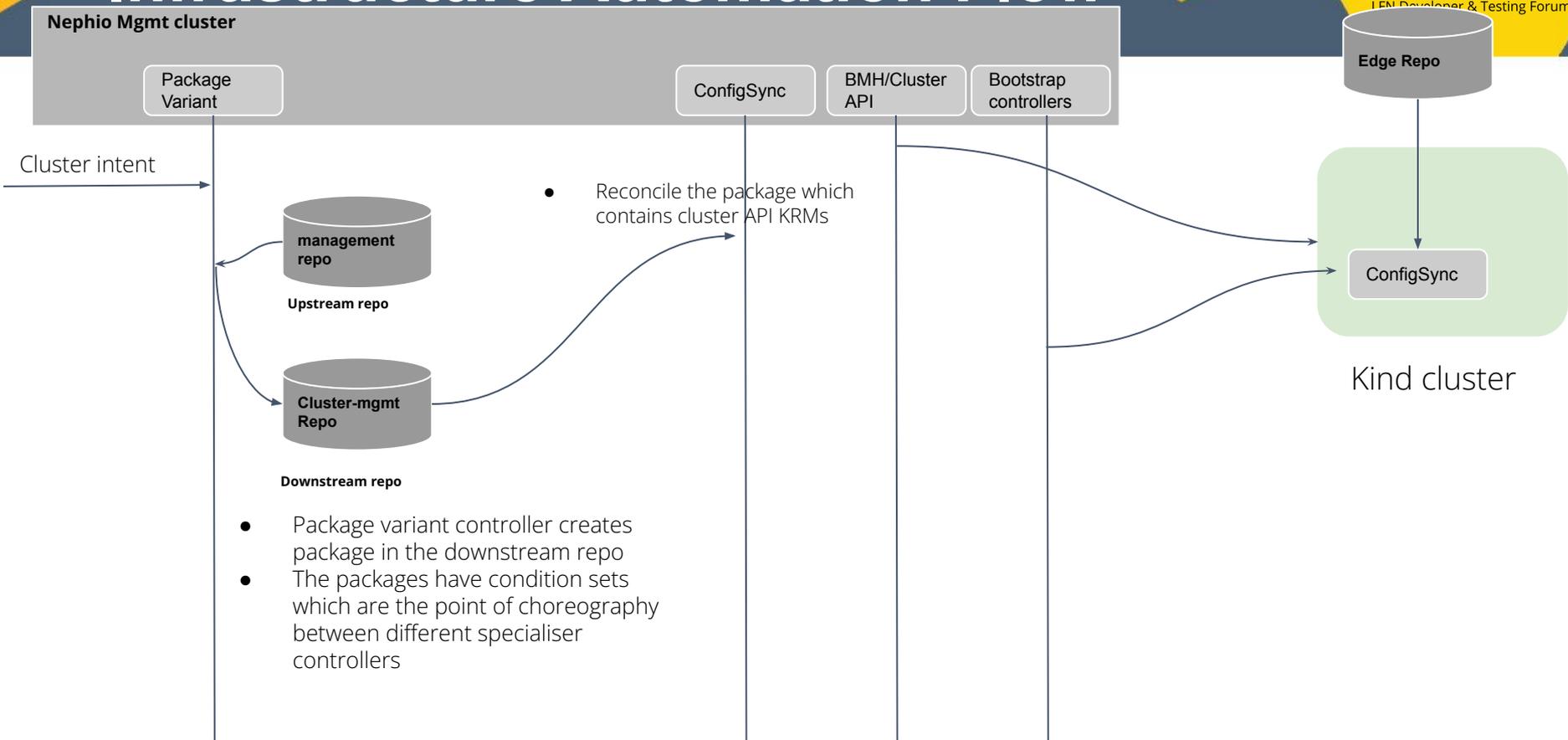


This demo is created by following this community link:

<https://github.com/nephio-project/docs/blob/main/user-guide/exercises.md>

Q&A, Thank You

Infrastructure Automation Flow



Workload Deployment Automation

Nephio Mgmt cluster

Package Variant

Specializers

Approver

Workload intent

management repo

Edge Repo

Base package pipeline and conditions.

- Ipam : NotReady
- NAD: Not Ready
- VendorDeployment KRM: NotReady

- Ipam : **Ready**
- NAD: Not Ready
- VendorDeployment KRM: NotReady

- Ipam : **Ready**
- NAD: **Ready**
- VendorDeployment KRM: NotReady

- Ipam : **Ready**
- NAD: **Ready**
- VendorDeployment KRM: **Ready**

Ipam specialiser gets ip allocated based on target cluster, and updates the package using KRM function

NAD injector updates the package by inserting the network attachment definition KRMs

This specializer can insert configmap or deployment objects based on the vendor NF requirement

Approve and publish package when all conditions are met.

NF

ConfigSync

Edge cluster

Edge Repo