

CNFO – Simplified AAI CNF Model

Konrad Bańka (Samsung)
Łukasz Rajewski (Orange)
Seshu Kumar (Huawei)

CNFO - Summary

Executive Summary - Provide CNF orchestration support through integration of K8s adapter in ONAP SO

- Support for provisioning CNFs using an external K8s Manager
- Support the Helm based orchestration
- leverage the existing functionality of Multi cloud in SO
- Bring in the advantages of the K8s orchestrator
- Set stage for the Cloud Native scenarios

Owners: Lukasz Rajewski (Orange), Seshu Kumar M (Huawei)

Business Impact - Enables operators and service providers to orchestrate CNFs based services along with the VNFs and PNFs

Business Markets - All operators and service providers that are intended to use the CNFs along with PNFs / VNFs

Funding/Financial Impacts - Reduction in the footprint of the ONAP for CNF support.

Organization Mgmt, Sales Strategies - *There is no additional organizational management or sales strategies for this requirement outside of a service providers "normal" ONAP deployment and its attendant organizational resources from a service provider.*

REQ-341
Guilin

REQ-458
Honolulu

REQ-627
Istanbul+

AAI CNF Model – Overview

- Currently no CNF Resources information is visible in ONAP AAI
- Some interfaces are already implemented (Multicloud-k8s Status/Query API) that allow retrieval of detailed resources information
- Initial implementation of CNF Model in AAI should be simple and allow user to know about resources available and where to get their exact status from
- Long-Term solution should design appropriate CNF Resources in AAI, providing only the most important data and relationships about them.

Simple AAI CNF Model – Istanbul

- Create additional AAI Object (k8s-resource) storing information about **ANY** Resource in K8s
- Data stored within this AAI object:
 - `id` [guid; **Primary Key**]
 - `name` [string; **Alternate Key**]
 - `group` [string; **Alternate Key**]
 - `version` [string; **Alternate Key**]
 - `kind` [string; **Alternate Key**]
 - `namespace` [string; **Alternate Key**; **Empty-allowed**]
 - `labels` [list<string>]
 - `k8s-resource-selflink` [string]
 - This field allows API consumer to specify query toward SO CNF Adapter to get full object data
- This object would be a child resource of tenant, accessible using composite key:
 - /cloud-infrastructure/cloud-regions/cloud-region/{cloud-owner}/{cloud-region-id}/tenants/tenant/{tenant-id}
 - /k8s-resources?name=vfw&namespace=vfirewall
 - /k8s-resources/k8s-resource/{k8s-resource-id}
- Relationship matrix:
 - TO **tenant** (PARENT of k8s-resource, k8s-resource **HostedOn** tenant, MANY2ONE)
 - TO **cloud-region** (k8s-resource **HostedOn** cloud-region, ONE2MANY)
 - FROM **generic-vnf** (generic-vnf **ComposedOf** k8s-resource, ONE2MANY)
 - FROM **vf-module** (vf-module **ComposedOf** k8s-resource, ONE2MANY)

Simple AAI CNF Model – Istanbul

Example object:

- Fields:

- id: “dd5310e1-526d-4fd3-9a7a-5321d637aa7b”
- name: “packetgen”
- group: “core”
- version: “v1”
- kind: “service”
- namespace: “vfirewall”
- labels: [“app=packetgen”, “release=test-release”, “k8splugin.io/rb-instance-id=goofy_merkle”]
- k8s-resource-selflink: <http://so-cnf-adapter:8090/TBD>

- Resource API Path:

- /cloud-infrastructure/... k8s-resources/k8s-resource/dd5310e1-526d-4fd3-9a7a-5321d637aa7b
- /cloud-infrastructure/.../k8s-resources?name=packetgen&group=core&version=v1&kind=service
 - may return more than one result

Simple AAI CNF Model – Jakarta

- Provide **the most meaningful data** directly into AAI (according to predefined schema)
- Define **high level AAI objects** to describe CNF Resources with meaningful data for Closed-Loops, OSS/BSS and Operators
- Source of data for K8s objects may differ depending on actual resource:
 - Some objects can change their state over time, they're mostly related to given CNF/*VF-module* (eg. Pods)
 - Some objects would be static by nature, they're mostly associated with Cloud-Region (eg. Namespaces, Worker Nodes, CNIs)

