Nephio Proofs of Concept

Please visit the Nephio GitHub for code and more information. This repository collects information and code for proofs of concepts (PoCs) presented to and contributed to the Nephio project.

Note: Issues should be opened in the nephio repository, using the prefix "nephio-pocs: " in the issue title.

Pull requests are welcome. Each PoC has its own directory in this repository, which should minimally contain a `README.md` file with a description and link, but may contain the full source code and documentation for the PoC. Additionally, each PoC should be listed in the summary below in alphabetical order and with a short one-paragraph description relating it directly to Nephio's goals:

- Candice: Allows local NETCONF and NETCONF-like workflows (to CNFs and adjacent PNFs) to be managed declaratively in Kubernetes.
- CNCK: Allows for network functions that are not designed to be cloud native (do not self-configure or self-orchestrate) to work better in a cloud native environment by handling dependency discovery and configuration.
- ENO: External Network Operator (ENO) is a framework that enables network automation in Kubernetes. Exposes a common API which allows the dynamic orchestration of networks on cluster and fabric level and therefore gives the ability to applications to consume a high number of different networks. ENO intends to run in workload clusters under Nephio context and will be responsible for the network provision inside the cluster as well as in the fabric.
- free5GC Operator: Deploys and manages free5GC's AMF, SMF, and UPF components on Kubernetes.
- Knap: Enables "network-as-a-service" for Kubernetes by allowing network functions to specify Multus network requirements in general terms without requiring local cluster- or node-specific knowledge. Knap then generates Multus CNI configs automatically.
- Nephio NF Controllers An example of network function controllers integrated with kpt.
- Nephio Package Deployment Controller Shows how to use Porch to render package variants across a set of cluster repositories, while injecting cluster specific context and reconfiguring each package based on that context.
- Nephio 5gc Topology Controller Builds on top of the NF controller and the pacakge deployment controller to render multiple network functions across multiple, different types of clusters.
- NF Injector Controller Builds on the Nephio 5gc Topology controller to inject a `UPFDeployment` resource and associated IPAM allocation requests.
- K8s IPAM Controller Provides an extensible IPAM system, along with a reference implementation. It builds on top of the 5gc topology controller and NF injector controller to perform IP allocations and inject the results back into the package.
- Nephio/Porch WebUI A prototype web UI for Porch that can be configured for use in Nephio.
- Planter: A meta-scheduler for Kubernetes that enables an all-at-once fire-and-forget declarative frontend for the complex lifecycle management of the workload-and-its-cluster interrelationships. Addresses the "bifurcation" (or "chicken-and-egg") problem that arises from vertically-integrated network function workloads.
- nf-deploy-fn Mutates and injects the IP allocationss in `UPFDeployment`. Runs as a kpt function in the kpt pipeline.
- nad-inject-fn Injects/generates Multus network attachement definitions based on a `UPFDeployment` and IP allocations. Runs as a kpt function in the kpt pipeline