

How CSPs Can Leverage Nephio to Integrate Telco Standards and Cloud Native Automation Principles Through Open Source

Author: Sana Tariq, Principal Technology Architect, Cloud, *TELUS Communications*

Challenge

Communication Service Providers (CSPs) must innovate faster to maintain a competitive edge in the digital marketplace. They must continue to own customer relationships through new services and applications that drive how the world communicates. If CSPs did not innovate, they would be reduced to data-pipes serving only connectivity. Innovation in the services layer is directly related to the stack that builds the telco network itself; it has to be agile, open, and software-centric, exposing/using APIs to the layers above it to drive service innovation.

In TELUS' journey to become a software centric organization, they made a significant amount of progress since the inception of 'NFV'. Standards evolved, like ETSI, 3GPP and O-RAN to support openness. Industry organizations went through massive transformations in achieving DevOps culture, shifting teams, mindsets, skill sets and tools. An ecosystem of cloud providers grew, with capabilities to support telco workloads, giving telecom service providers the opportunity to build sophisticated hybrid cloud strategies exploiting the benefits of regions, APIs, lower-cost and performance.

Applications that deliver critical services come from vendors evolved, from being too closed to somewhat open, from PNFs to VNFs to CNFs along the journey to being more cloud native.

Cloud Adoption

The importance of software and open source became extra clear as cloud adoption grew, with ~80% of cloud technologies are based on open source projects. These open source projects are the common denominator across multiple clouds helping to avoid vendor lock-in and enabling interoperability between different private and public clouds. These common open source projects and softwares are helping build a community of developers that can share skill sets. Open source is playing a huge role in standardization across this ecosystem.

Automation & Orchestration

Some progress was made with automation and orchestration, but much of what is done is "fire and forget". There has been little progress made within policy and closed loop to achieve fully managed day2+ ops. Automation is not cloud-native yet and we are not ready for scale. Cloud native automation will prepare us for the future of AI/ML, with self- healing, self- scaling fully autonomous networks.

Standardization

Telecom network functions evolved from a history of standardization that drove interoperability around the world with multiple devices, networks and vendors. Telecom has some of the most complex standards, RFCs, and protocols that are not understood by the webscale world; the code written for telco network functions was old and slow to evolve. There was a need to re-factor and re-develop applications as microservices. It was very difficult to rewrite legacy code fast.

Hence, we reached a stage where applications are packaged as containers but they are still monolithic in nature. Most CNFs are still very large in size and are not capable of dynamically scaling out across different component micro-services. They don't follow the cloud principles of failing fast and recovering faster; elsewhere.

From VNFs to CNFs

"We must Improve our CNFs and automation models to exploit the benefits of cloud."

From VNFs to CNFs (con't)

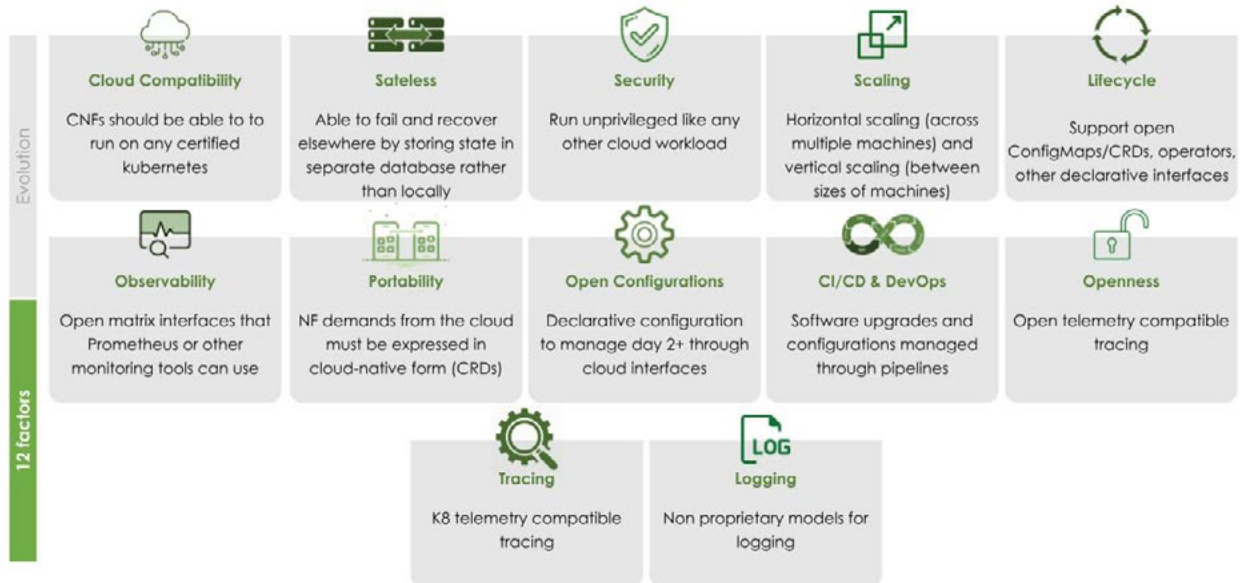
When we put monolithic applications on sophisticated private or public clouds, there is little value that we get from cloud resilience mechanisms, which say 'deploy small and grow as you need.' About ~80% of the cloud value in the form of APIs and environment services is unused. And with a large footprint we have a poor business case for public cloud. There is still little agility and cost savings with current cloud migrations.

A true microservices- based NF should have loosely coupled microservices that can each scale/heal dynamically. We should be able to achieve a highly available resilient instance with a small footprint. Its security, reliability, and management is also split into smaller modules.

We must Improve our CNFs and automation models to exploit the benefits of cloud.

Vision:

Cloud Native Automation + Nephio from a service provider's lens



Solution

The 12 factors that determine CNF are actually cloud-native, which TELUS has been including as a compliance requirement in every RFP. We have made a long way on 5G core with varying degrees of compliance depending on which ecosystem you are working with. ORAN is extremely immature at the moment to address these requirements but we are still working with our vendors.

Cloud features cannot be used adequately without complying to the above principles. Applications require re-design and refactoring. Service providers must continue to drive the industry in the direction of cloud. There are some disruptive vendors who have built applications with a "cloud first" mindset.

"The idea of Project Nephio is taking a different approach to solve the deployment and life cycle management of the entire stack, from infrastructure to network functions through a single consistent Kubernetes based framework."

Solution (con't)

Complex automation - NFs as pets

Simplified automation – NFs as cattle

The diagram shows a large number of CNFs from different vendors, each coming with its own automation solution, like VNFM for deploying, Lifecycle management, and EMS for FM/PM data monitoring.

Most VNFMs are a combination of standards based and proprietary APIs, which means there is a lot of custom and proprietary work required for each CNF deployment.

TELUS develops Network Service Descriptor (NSDs) (generally templates in Yang or TOSCA) in service orchestration to abstract prescriptive details for deployment. Life cycle management of these CNFs present even bigger challenges (i.e., writing policies in different languages across different tools and gathering monitoring data from legacy EMSs). This model presents a state of maturing complexity.

The idea of Project Nephio is taking a different approach to solve the deployment and life cycle management of the entire stack, from infrastructure to network functions through a single consistent Kubernetes based framework.

Nephio is really an extension of Kubernetes, one of the most successful open source projects that has stood the test of scale and time and proven its simplicity and flexibility.

"Nephio proposes to develop CRDs and Operators for infrastructure deployment/ configuration network function deployment and NF configuration."

- Some simple Enterprise applications can use just kubernetes for orchestration which can manage simple resource types like compute, network and storage.
- Telco network functions require additional customizations in the cloud (e.g., but SR-IOV dpdk, CPU pinning, Huge pages etc.) that are not natively expressed in Kubernetes. So the Telco industry attempted to solve it through external constructs like VNFMs to manage the CNF end to end.
- Over time, it was observed that using VNFMs was not a successful model with the heterogeneity and complexity it brings at operational levels.

Alternate Approach

An alternate approach could have been extending Kubernetes' general purpose control plane to "see" new resources as Custom Resources (CRs) and define their ideal state through custom resource document (CRDs), then use Operators to execute an infinite loop on managing the resources LCM all within Kubernetes native constructs. Operator uses Custom Resources (CR) that define the desired configuration and state of a specific application through Custom Resource Definitions (CRD) -- which is exactly what Nephio proposed.

Nephio proposes to develop CRDs and Operators for infrastructure deployment/ configuration network function deployment and NF configuration. These CRDs are managed through operators which reconcile the system to a desired state with an infinite loop - control loop (the intent). This infinite loop that can reconcile the current state to the intended or desired state and constantly manage drifts is the most powerful outcome of using kubernetes and operators. Kubernetes operators manage application logic and are part of the Kubernetes control plane. As such, they are controllers that execute loops to check the actual state of the cluster and the desired state, acting to reconcile them when the two states are drifting apart. This means developers do not need to write policies in tens of different tools and languages to achieve day 2+ management.

The Nephio model can derive simplicity and scale.

Telco standards need to evolve to converge with CNCF and the webscale world to allow cloud native transformation.

Service providers have a unique opportunity to work through open source communities to drive standards, public cloud providers, network function vendors and jointly collaborate to solve some of the cloud native problems together. Nephio's automation requirements will drive some transformation through CNFs design that can bring openness.

To learn more about these principles, please watch [Sana Tariq's ONE Summit presentation from November 2022](#).