Learning with Nephio R1 - Episode 1 - Series Introduction

Author: John Belamaric

Prerequisites: None


Video: https://youtu.be/1MyuFgRLStY

About this Series

Welcome to Nephio! We are glad you have joined us to learn about Nephio. This is the first in a series of tutorials designed to help you learn about Nephio - what is is, why it exists, and how it works. This series of short articles will include slides and often a video presentation as well, as well as exercises you can do in a Nephio demo environment. The idea is to allow you to get a basic understanding of Nephio and its concepts, and then provide additional material for deeper dives.

About Nephio

What exactly is Nephio, though? Perhaps the best place to start is with our mission statement:

Nephio’s goal is to deliver carrier-grade, simple, open, Kubernetes-based cloud native intent automation and common automation templates that materially simplify the deployment and management of multi-vendor cloud infrastructure and network functions across large scale edge deployments. Nephio enables faster onboarding of network functions to production including provisioning of underlying cloud infrastructure with a true cloud native approach, and reduces costs of adoption of cloud and network infrastructure.

That’s a lot to digest. To simplify it, let’s break it down into the “What”, the “How”, and the “Why”.

For “What”, we see “automation” for “the deployment and management of multi-vendor cloud infrastructure and network functions across large scale edge deployments”. The first thing to notice is that Nephio aims to automate “deployment and management”. Nephio will handle the full life cycle of managed entities - provisioning as well as updating over time and eventually turning them down. Nephio will allow changes to existing deployments, and manage those changes through the same automation tool chain as that used for deployment.

The next key point is that Nephio will work for both network functions and the infrastructure (clusters, nodes, interfaces, networks, interconnects, cloud services, and more) that is required to run those network functions. By using the same platform for management of the network functions, and their underlying infrastructure, we enable automation that can start from the network function requirements, and use those to derive the necessary infrastructure to support the network function. As the requirements change (for example, a need for increased capacity), the automation can determine the changes needed in the underlying infrastructure and implement those as well.

Nephio also needs to be multi-vendor. This means it is not tied to any specific network function vendor, nor is it tied to a particular cloud platform. In fact, you do not need a cloud provider at all - you may want to implement this with your own private cloud. A key challenge in this is ensuring that vendors continue to be able to differentiate, without breaking the general purpose automation. The goal of Nephio is not to produce a one-size-fits-all platform that pushes vendors to the lowest common denominator. Instead, we intend to foster an ecosystem for innovation. This means taking the common, non-differentiating aspects of vendors’ solutions and sharing the development and maintenance burden of those across the community by developing common schemas, tool kits, libraries, and controller frameworks. This enables cloud, network function, and orchestration vendors to focus on the unique and critical value they can bring.

The finally piece of the “what” is that this all must work “across large scale edge deployments”. This is a critical consideration, and has far reaching consequences. Some are obvious - the user interfaces, automation code and other tooling must be able to handle many thousands of instances of network functions without slowing to a crawl, for example. However, the massive scale also implies something about the organizations that are building these topologies. They are not built by a single developer or even a single team. There are many, many teams that must interact and coordinate their efforts to build out and maintain a large topology. When seen through a “single instance” lens, many of the approaches we take in Nephio may seem overly complex. However, they are intended to enable a much more robust platform for managing network functions across a large, complex organization, not just a large number of clusters.

The mission also covers the “How” with the phrase “Kubernetes-based cloud native intent automation”. What is “Kubernetes-based” doing here? By explicitly stating that we will build on top of Kubernetes, we are focusing the community efforts on where Nephio creates value. We are firmly stating that Nephio will not waste time and effort on software platform infrastructure that can be readily supplied by Kubernetes: API machinery, authentication and authorization, schema definition machinery, basic client and controller runtime libraries and tooling, and many other core services.

Now, what does “cloud native intent automation” mean? This phrase, especially in the context of “Kubernetes-based”, describes a specific technique for automation: the use of declarative intent with continuous reconciliation. Rather than the user telling the automation platform exactly how to execute a change to a system, the user simply declares the intended end state for the system. The automation needs to be smart enough and complete enough to figure out how to “get there from here”. It must observe the existing state, compare it to the intended state, and resolve (reconcile) any differences by modifying the system. This process can be repeated indefinitely, resulting in a system that converges towards stability, even under changing conditions.

Finally, the mission covers the “Why”, including both “faster onboarding of network functions to production” and “reduces costs of adoption of cloud”. It doesn't really qualify the “faster”, but the goal is really a dramatic reduction. Rather than taking months, it should take hours or minutes. When the broader technology industry moved from manual provisioning of physical server to virtual machines, and eventually to cloud-based VMs, it changed the way people develop and build services. We expect similar benefits if network topologies can be more easily created, modified, and destroyed. These changes can lead to increased agility in delivering services to customers. The "Why" will be covered in much more depth in the next episode.

About the Nephio Community
Nephio is an open source project of the Linux Foundation. We do our best to be a friendly, welcoming community. We have several different “special interest groups” (SIGs) that meet regularly, as well as mailing lists and a Slack instance. Please join us!

- Learning page - [https://nephio.org/learn](https://nephio.org/learn)
- Site - [https://nephio.org/](https://nephio.org/), [https://nephio.org/about/](https://nephio.org/about/)
- Wiki - [https://wiki.nephio.org/](https://wiki.nephio.org/)
- Docs - [https://github.com/nephio-project/docs](https://github.com/nephio-project/docs)
- Blog Postings - [https://nephio.org/blog/](https://nephio.org/blog/)
- Slack - [https://nephio.slack.com/](https://nephio.slack.com/) (join via this invitation)
- Project Github - [https://github.com/nephio-project](https://github.com/nephio-project)
- Project email distro - [https://lists.nephio.org](https://lists.nephio.org)
  - nephio-tsc (for TSC members and interested parties)
  - nephio-dev (for all)
- SIG lists: sig-netarch, sig-automation, sig-release