



Red Hat

Sample Telco book title that is quite a
mouthful indeed with a lot of words

The subtitle of the document

Version 10.1, 2021-12-09

Copyright

Copyright goes here!

Introduction

Red Hat is building a Telco platform to serve network needs across Core, Edge, Lite and Far Edge. In the infancy of this journey, the platform is the entry path for 5G Core cloud-native network functions (NFs) and brings forth the introduction of CNCF-based stack for Cloud-Native Network Functions (CNFs).

In addition, Red Hat is also building a Kubernetes-based Container as a Service (CaaS) Platform with Platform as a Service (PaaS) services to support 5G Services-based architecture.



A note!



Caution!



Tip!



Warning!

Scope

This document, and the current platform configuration, are currently limited in scope to Wireless network elements.

This document covers the requirements for tenants to run their application on Red Hat's OpenShift Network Functions Virtualization Infrastructure (NFVI) platform. While Red Hat OpenStack was initially designed to support "Direct-Port" VNFs, this document's intent is to provide guidance as the Partner community evolves their software to support containerized cloud-native applications.

Refactoring

Network Functions (NFs) break their software down into the smallest sets of microservices possible. Running monolithic applications inside of a container is not the operating model recommended by Red Hat.

It is hard to move a 1000LB boulder. However, it is easy when that boulder is broken down into many pieces. All Cloud-native network functions (CNFs) break apart each piece of the functions/services/processes into separate containers. These containers can still be within OpenShift PODs, and all of the functions that perform a single task must be within the same namespace.

There is a quote that describes this best from [Lewis and Fowler](#): "the microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and independently deployable by fully automated deployment machinery."

Pods

Pods are the smallest deployable units of computing that can be created and managed in OpenShift.

A Pod can contain one or more running containers at a time. Applications / containers running in the same Pod have access to several of the same namespaces. For example, each application has access to the same network namespace, meaning that one running container can communicate with another running container over 127.0.0.1:<port>.

The same applies to storage volumes, since the containers are in the same Pod, they have access to the same storage namespace, and can both mount the same volume. Of course, care must be taken in this configuration to prevent any data corruption.

Sample code block

```
---
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: performance-addon-operator
  namespace: openshift-performance-addon-operator
```

spec:

channel: "4.9"

name: performance-addon-operator

source: performance-addon-operator

sourceNamespace: openshift-marketplace