CNF Architectures for Edge and Dataplane

Jarrod Lucia  F5 Inc
Innovation Architect | Service Provider 5G Telco and Cloud | APCJ Regional Lead
I NEED TO KNOW WHY MOVING OUR APP TO THE CLOUD DIDN’T AUTOMATICALLY SOLVE ALL OUR PROBLEMS.

YOU WOULDN’T LET ME RE-ARCHITECT THE APP TO BE CLOUD-NATIVE. JUST PUT IT IN CONTAINERS.

YOU CAN’T SOLVE A PROBLEM JUST BY SAYING TECHY THINGS. KUBERNETES.
Today’s Challenges for Telco and Edge Deployments

INCREASING TRAFFIC AND COMPLEXITY OF NETWORKS BREAKS EXISTING MODELS

Increasing CapEx and OpEX
- Need higher network capacity at lower cost
- Need to increase revenue from 5G services and new edge opportunities

Scaling and automating functions
- Digital transformation needs to be simplified
- Deployment and upgrade cycles too slow

Maintaining and improving network security
- Larger threat surface
- Increasing DDoS attacks
Definition of Cloud Native

KEY ASPECTS OF BEING CLOUD NATIVE

Agility  Resilience  Observability
Transformation to Containerization

VNF

- App 1
- Bins/Lib
- Guest OS

- App 2
- Bins/Lib
- Guest OS

- App 3
- Bins/Lib
- Guest OS

CNF

- App 1
- App 2
- App 3
- Bins/Lib
- Bins/Lib
- Bins/Lib

Container Engine

- Operating System
- Infrastructure

Machine Virtualization

- Hypervisor
- Infrastructure
- Laptop

Containerization
- Secure by Design
- Modular Architecture
- Monitoring, Visibility and Analytics
- Frictionless Service Consumption
- Governance
- Telco 3GPP Compliant

Key Composition of a Microservices Strategy

- Insight on apps utilization & performance
- SLA and resiliency and availability monitoring
- Service oriented monitoring
- Cost consumption monitoring
- Application performance (APM) monitoring
- API driven ecosystem
- Consumable full application services (e.g. traffic management and security)
- Support Blue/Green deployment model
- Seamless integration into CI/CD pipeline

- Organization structure alignment

- Embed Security control in the beginning
- Secure image registries
  - Trusted signed images
  - Container Vulnerability Scan
- Scan for unintentional private embedded data.
- Identity and access management
  - mTLS
  - Authentication Proxy
  - OpenID Connect
- RBAC – Role-based Access Control
- Least privilege access model

- Portable and platform vendor independent
- Agile and scalable architecture – Works everywhere, infrastructure independent
- Environment elasticity
- API driven ecosystem
- Run in consistent and predictable manner
- Highly resilient architecture and application services

- Standardization
  - Tooling and language framework
  - Deployment pipeline
  - Alignment with standards bodies (if possible – e.g. OCI, CNCF)
- Automation
  - Consistent deployment patterns
  - API definition and strategy
  - Monitoring (e.g. metrics and dashboard)

- Organization structure alignment
Cloud-Native is the evolution of Network Functions delivery

LONG TERM EVOLUTION TO IMPROVE OPERATIONS

**Network Functions** are elements that enrich network traffic (CGNAT, Firewall, DNS Cache, etc.)

- **1980s - 2000s**
  - Physical Network Functions
    - Network Functions implemented as hardware appliances & chassis with tightly coupled software

- **2010s**
  - Virtual Network Functions
    - Network Functions implemented as virtual machines running on Networks Functions Virtualization platforms

- **2020s - Future**
  - Cloud-Native Network Functions
    - Network Functions implemented as containers that can be controlled by Kubernetes

Network Functions are elements that enrich network traffic (CGNAT, Firewall, DNS Cache, etc.)
Comparison of Form Factor Architectures

WHERE ARE THE BENEFITS OF CLOUD-NATIVE OVER EXISTING ARCHITECTURES?

<table>
<thead>
<tr>
<th></th>
<th>Physical</th>
<th>Virtualised</th>
<th>Cloud-Native</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network performance</td>
<td>Dedicated performance enhanced with ASIC and FPGA technologies</td>
<td>Raw throughput typically limited by hypervisor performance</td>
<td>Raw throughput typically limited by commercial CPU performance</td>
</tr>
<tr>
<td>Resource utilisation</td>
<td>Dedicated Network Resource</td>
<td>Resource over-provisioning not desirable for telco workloads</td>
<td>Resource utilization based on available scheduling</td>
</tr>
<tr>
<td>Deployment speed</td>
<td>Deployment takes days - weeks</td>
<td>Deployment takes minutes</td>
<td>Deployment takes seconds</td>
</tr>
<tr>
<td>Automation</td>
<td>Proprietary and fragmented automation</td>
<td>Automation is complex with multiple interacting control planes</td>
<td>Kubernetes provides a common, consistent control plane</td>
</tr>
</tbody>
</table>
Creating a Cloud-Native Network Function

JUST PORTING EXISTING ARCHITECTURES HAS LIMITED BENEFITS

Monolithic CNF

- Orchestrated through Kubernetes control plane
- Slow to deploy
- Difficult to scale data plane
- Complex to upgrade and enhance
- Inefficient use of compute resources
Creating a True Cloud-Native Network Function

DECOMPOSING THE TRADITIONAL NETWORK FUNCTIONS MONOLITH

Network Function

Monolithic software

Orchestrated through Kubernetes control plane
Fast scaling of stateless dataplane
Distributed state supports scaling and resilience
Control functions rapidly added and upgraded
Lightweight, efficient components
Data-plane consolidation remains critical for efficient NFV deployment at scale.

Data-plane consolidation remains important for CNF.
F5 CNF Architecture Overview

Control Node
- Helm v3 Chart (config package)
- Pod (Controller) - F5 Controller
- API Server
- CRD
- Listens for events related to F5

Workload Node
- Pod (Data Classifier)
  - f5-tmm
    - Containerized traffic mgmt
    - micro-kernel (tmm) providing dataplane functions
  - SRIOV
  - DPU
  - SNiC

Pod – (Dataplane Stateless Management)
- Pod (Telemetry / Observability)
- SRIOV
- DPU
- SNiC

Cloud-Native Functions
- CUPS architecture
- Multiple dataplane pods for increased scale
- Dataplane Stateless management
- Automation / Kubernetes Control Plane
- Boosted Dataplane Performance Capability
CNF Key Characteristics

BEST OF SUITE CONSOLIDATED CLOUD-NATIVE NETWORK FUNCTIONS

- Control Plane – User / Data Plane Separation

- Containerized micro-services that communicate via APIs

- Small footprint with horizontal scaling
CNF Key Characteristics

BEST OF SUITE CONSOLIDATED CLOUD-NATIVE NETWORK FUNCTIONS

- Orchestrated by Kubernetes API
- Integrations with popular Kubernetes distributions
- Boosted performance - hardware offload (DPU / FPGA / SmartNIC)