Open source Approach to Design and Deployment of Microservices based VNF

Prem Sankar G, Ericsson
http://twitter.com/premsankar
Agenda

- Microservices – Intro, design patterns
- ETSI NFV Architecture
- VNF Characteristics
- Opensource Ecosystem
- Container Orchestration Engine Integration
What is Microservices?

Principles that drive Microservices architecture

- Services must be loosely coupled so that they can be developed, deployed and scaled independently
- Organized around business capabilities
- API Focused
- Smart endpoints and dumb pipes
- Decentralized Governance
- **Decentralized Data Management**
- Infrastructure Automation (infrastructure as code)
- Design for failure
- Evolutionary Design
Network Functions - Flavors
Microservices Design Patterns

• Reference – Chris Richardson
  http://microservices.io/patterns/microservices.html
Microservices Design Patterns for VNF

• Decomposition Pattern
  • Business Capability/Subdomain context
• Service discovery Pattern
• API Gateway Pattern
• Database per service Pattern
  • Keep each microservice’s persistent data private to that service and accessible only via its API
  • RDBMS design approaches
    • Private-tables-per-service – each service owns a set of tables that must only be accessed by that service
    • Schema-per-service – each service has a database schema that’s private to that service
    • Database-server-per-service – each service has its own database server.
• Circuit Breaker
Key drivers for VNF

- Elasticity
- Fault Tolerance
- Agility/
- API-based/End points
- Infra and Location agnostic
- Central Orchestration
- Predictability
- Business needs
  - 5G, IOT
ETSI NFV Architecture
VNF Lifecycle

- Workflow Manager and Orchestrator
  - Continuous Integration
    - Design and Creation
  - Continuous Deployment
    - Deployment and Execution
  - Monitor
    - Operation and Maintenance
Opensource Ecosystem

• Workflow Manager and Orchestrator
  • ONAP/TOSCA

• Design and Deployment
  • CI/CD Pipelines
    • Jenkins
    • Fabric8 (Jenkins, Kubernetes and Docker)
    • Other options – goCD, concourse.ci
    • Ansible/Puppet/Chef

• Execution/NFVi
  • Kubernetes with ODL, OPNFV

• Monitoring
  • Prometheus and other alternatives (https://prometheus.io/docs/introduction/comparison/)

• Tracing
  • Zipkin

• Logging
  • ELK stack
Container Orchestration Engine - OpenDaylight

VNF - Mixed deployment

- OpenStack APIs
- Kubernetes APIs
- Neutron Plugin or Gluon/Proton
- CNI Plugin
- OpenStack
- Kubernetes
- OpenDaylight
- App
- App
- App
- Container RT
- KVM
- Container RT
- OVS
- Linux
- Linux
- Linux
- Linux
- Linux
- Datacenter Hardware
- VM Application
- Container Application
Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.

-- Melvyn Conway, 1967