PaaS-driven Interoperation between OSM and Kubernetes-based Edge Platforms

Nikos Psaromanolakis, Staff Research Engineer
Vasileios Theodorou, Senior Research Engineer
Intracom Telecom
Overview

• Motivation

• Edge PaaS Approach
  • Edge Management Platform (EMP)
  • OSM Extensions & EMP
  • Updated OSM Information Model
  • Architecture & Implementation

• Demo
Motivation

• Enhancement of NFV MANO towards incorporation of a Platform-as-a-Service (PaaS) delivery model for Edge services
  • Foster automation, interoperability & maintainability
  • Minimize management overhead
    • No need to continuously collect detailed state/raw data from all edges
    • Low level management decisions can take place locally → optimized on local infra
  • Minimize deployment speed and stimulate reusability of services
    • Abstraction of service modelling to a usable yet flexible level
OSM KNF vs Edge PaaS Approach

**OSM KNF steps:**
- Build helm chart (outside of OSM)
- Reference helm charm inside of KNF
- Package KNF
- Upload KNF (inside VNFD) which contains helm chart to OSM

**Edge PaaS Approach:**
- Describe the PaaS service natively inside OSM Information Model (VNF Descriptor)
  - Model expressiveness (auto scaling/healing policies, QoS, exposed application ports…)
  - Management translation and overhead is transferred to EMP (Kubernetes + Interfacing layer)
- Updates on the deployed PaaS services can be made dynamically (without the involvement of OSM)
The “Edge Management Platform”

- **Edge Management Platform (EMP)** is managing Edge Cloud resources & services
- Enhanced VIM entity for edge clouds, which additionally:
  - Manages the lifecycle of edge resources
  - Exposes **PaaS capabilities** for the provisioning of Edge services
Cloud/Core to Edge Slicing Interoperation

VIM

Virtual Networking

Cloud/Core NFVI

PaaS-based Interoperation

(Encrypted) Intercommunication Tunnel

Extended NVF MANO (incl. PaaS features)

MANO & Control Plane

Network Slice #1 (Data Plane)

Network Slice #2 (Data Plane)

K8s technologies for Virtual Networking

- Open vSwitch CNI plugin
  - https://github.com/k8snetworkplumbingwg/ovs-cni
- Antrea
  - https://antrea.io/
- Multus CNI
  - https://github.com/k8snetworkplumbingwg/multus-cni
- k-vswitch
  - https://github.com/k-vswitch/k-vswitch
  - ...

© ETSI 2021

OSM#12 Ecosystem Day, 1 Dec. 2021
OSM Information Model (IM) Extension

• Extended the OSM Information Model (OSM-IM SOL005) for handling PaaS aspects
  • PaaS deployment flavor
  • Auto-scaling/healing parameters
  • Inter-connection configurations
  • Exposed application ports
  • ...

© ETSI 2021
Updated (Extended) OSM Information Model

```
vnfd:vnfd-catalog:
  vnfd:
    - connection-point:
      - name: eth0
        description: MEC PaaS descriptor that includes a PaaS with scaling policy
        id: demo_paas_vnfd
        mgmt-interface:
          - cp: eth0
            name: demo_paas_vnfd

    paas:
      - description: MEC Demo PaaS descriptor with scaling policy
        id: Demo_PaaS
        name: Demo_PaaS
        paas-interfaces:
          - external-connection-point-ref: eth0
            name: ens0
          quality: HIGH
          type: EXTERNAL
        paas-requirements:
          count-max: '5'
          count-min: '1'
          location: Peania_19002_Athens
          ports:
            - name: port_1
              port: '31510'
        scaling-policy:
          monitoring-variable: cp
          policy-name: minimize_cost
          type-paas: CONTAINER
          paas-service-name: demo_paas
          short-name: demo_paas_vnfd
          version: '1.0'
```
OSM extensions & EMP

• Extensions of all (end-to-end) involved OSM modules (v8) for realization of EMP features have been implemented:
  • LIGHT-UI, NBI, LCM
  • Databases (MySQL and MongoDB)
  • Resource Orchestration Module (RO)
    • Plugins that implement all methods (Instantiate, Update, Terminate & Delete etc.) to interact with:
      • EMP-K8S controller
      • OpenNESS (MEC software toolkit)

• Implemented an EMP controller that runs on top of Kubernetes:
  • Register, De-register, Deploy, Update and Delete Edge PaaS services
    • When a PaaS is instantiated, the EMP controller creates K8s objects:
      • Deployment, Services and Horizontal Pod Autoscaler (HPA)
OSM – EMP Architecture

- **Extended OSM:**
  - EMP registration / de-registration
  - PaaS interaction:
    - Instantiation/termination of PaaS edge services
    - Continuous (PaaS-level) status monitoring

- **EMP Controller:**
  - Translates OSM requests to K8S “language”
  - Register, De-register, Deploy, Update & Delete PaaS services
  - Communicates with an SDN controller to control (inter-)networking
• **Edge PaaS Service Catalogue:**
  - Register PaaS services locally available (i.e., deploy-able)
  - Maintain list of available auto-scaling policies
    - Extendable
    - Parameterizable
    - Configurable

```json
  "id": "6183e35ea1e1338f4c1f0697",
  "application_ports": [80],
  "image": "k8s.gcr.io/hpa-example",
  "name": "demo-paas",
  "type": "Container",
  "autoscaling_policies": [  
    {  
      "policy": "minimize_cost",
      "monitoring_metrics": [  
        {  
          "limit": "1Gi",
          "metric_name": "memory",
          "request": "0.5Gi",
          "util_percent": 50.0,
          "is_default": true
        },
        {  
          "limit": "500m",
          "metric_name": "cpu",
          "request": "200m",
          "util_percent": 60.0,
        }
      ]
    }
  ]
```
Edge PaaS Approach (Workflow example)

I have come up with better auto-scaling model parameters!

Data Scientist

PaaS #1 utilization parameter (in the PaaS Catalogue) updated

Inform OSM on number of PaaS instances & updated policy
Time for the live Demo!
THANK YOU!

https://osm.etsi.org/