Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: August 18, 2014

Z. Cao
H. Deng
China Mobile

February 14, 2014

Service Function Controller/Orchestration Requirements and Templates
draft-cao-sfc-control-orchestration-00

Abstract

Service Function Chain architecture further enables the modularity of network functions; network service functions can be split and chained together to compose complicated services. Network automation relies on a specific orchestrator to automatically deploy an end2end service or application. If this end2end service or application has a specific requirement to chaining several service functions, there is a need of an interface from the application to inform the orchestrator. This document investigates the problem.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 18, 2014.

Copyright Notice

Table of Contents

1. Motivation

Service function chaining is a broad term used to describe a common model for delivering multiple service functions in a specific order. Service function chaining de-couples service delivery from the underlying network topology and creates a dynamic services plane that addresses the requirements of cloud and virtual application delivery. Packets and/or flows that require services to be applied are classified and redirected to the appropriate service functions. Additionally, context can be shared between the network and the services. Service function chaining has also been discussed in other forums e.g. ETSI and 3GPP, and the ETSI NFV group is discussing service function chaining as part of network function virtualization.

Service Function Chain architecture further enables the modularity of network functions; network service functions can be split and chained together to compose complicated services. Network automation relies on a specific orchestrator to automatically deploy an end2end service or application. If this end2end service or application has a specific requirement to chaining several service functions, there is a need of an interface from the application to inform the orchestrator.

Orchestrator is an important function within the cloud management system and Network Function Virtualization (NFV) system [Heat][Cfn]. End-to-end service utilizes the NFV orchestrator to map the VNFs to the virtualization infrastructure, instantiating VNFs at appropriate locations to realize the intended service, allocating and scaling hardware resources to VNFs.

Establishing the VNF forwarding paths (forwarding path is service function chain) is also the task of the orchestrator. This document discusses the orchestration requirement and template examples.

2. Terminology

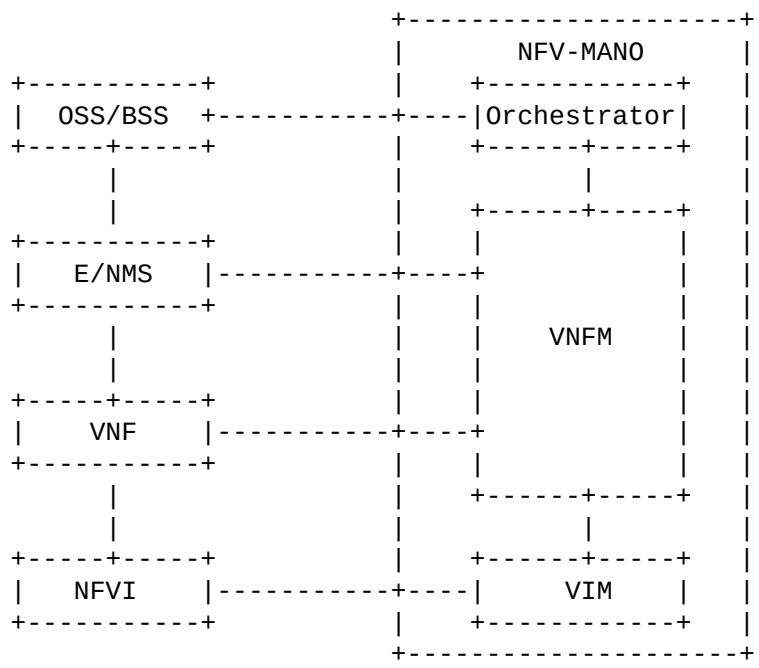
NF (Network Function): A functional building block within an operator's network infrastructure, which has well-defined external interfaces and a well-defined functional behaviour. Note that the totality of all network functions constitutes the entire network and services infrastructure of an operator/service provider. In practical terms, a Network Function is today often a network node or physical appliance. [Quoted from ETSI NFV]

Virtualised Network Function (VNF): An implementation of an executable software program that constitutes the whole or a part of an NF that can be deployed on a virtualisation infrastructure.

Service Classifier: A component that performs traffic classification. Classification is the precursor to the start of a service chaining path. Meta-data could assist traffic classification. Service classification is different from DPI component where only service related information in packets is retrieved and classified.

3. SFC Orchestration Framework

The following figure is the architecture of NFV.



NFV Architecture

The NFV-MANO consists of Orchestrator, VNFM(VNF Manager) and VIM (Virtual Infrastructure Manager). VNFM is the component to interact with the VNFs, and VIM is the to manage the NFV Infrastructure (NFVI), including computing, storage and NETWORKING.

Figure 1 is the service function chain orchestration architecture. In this architecture, Application Deployer is the entity or administrator to deploy a specific end-to-end network system. For example, it can be a Mobile Operator to deploy a virtualized packet core system with Gi-LAN services [I-D.liu-sfc-use-cases]. In the network auto deployment, the Application Deployer specifies the requirements of the Service Functions (certain vNF) and their forwarding paths (chaining relations). Thereafter, the Orchestrator informs the VNFM and VIM to instantiate SFs and Infrastructure Nodes respectively.

Note: the VNFM and VIM will interact with the SFC Controller Node in SFC Architecture [I-D.quinn-sfc-arch], or the VIM or VNFM will take the job of SFC Controller Node themselves.

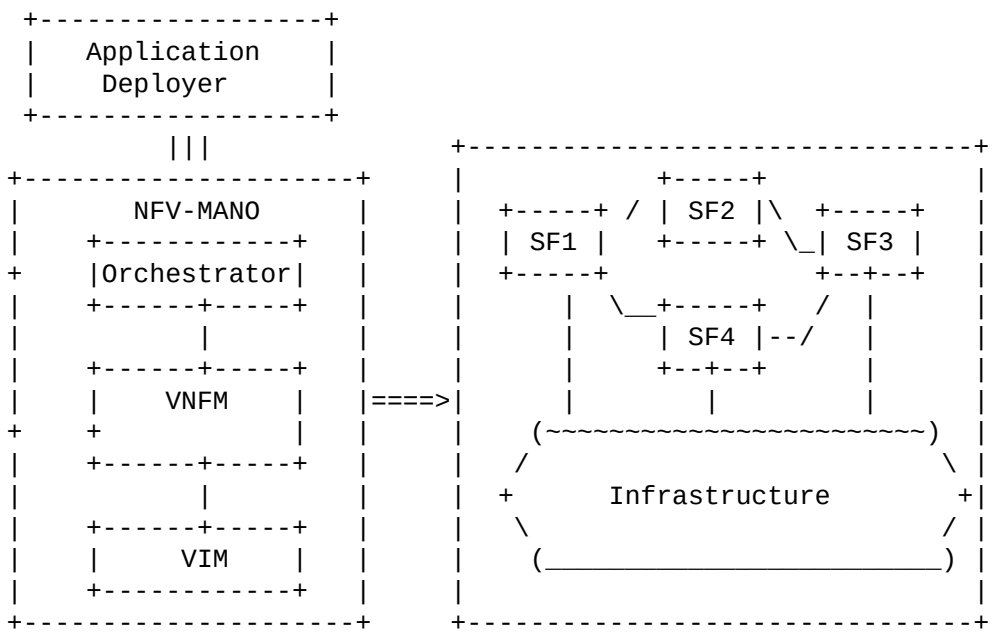


Figure 1: SFC Orchestration

4. Requirements

The requirements of SFC Orchestration Templates are listed as below.

1. RESTful Style API. This is the common practice in both AWS Cloudformation and Openstack Heat.

2. Components of the SFC Orchestration.

   A. Template Version: version number and time

   B. Template Description: text description of the template.

   C. Parameters: necessary parameters of SF, e.g., username and password of Database node, or SSH access account information.

   D. Mapping Information: Mapping of the vNF/SF to the infrastructure nodes, and it will contain the memory, storage, network information.

   E. Resources: containing the Elasticity, Properties, Security Group information.

3. Chaining information.

A. Chaining Identity.

B. Chain name.

C. Forwarding Context: the matching criteria of of this chain.

D. Next Hop Context: the information information of the next SF on the chain.

5. Orchestration Templates including SFC Information

According to the discussion in the previous section, an example of the SFC orchestration template can be depicted as below:

```
{
  "SFCTemplateFormatVersion" : "2014-02-12",

  "Description" : "This template installs a singe instance deployment for video
traffic optimization. It also specifies the chaining information for this instance",

  "Parameters" : {

    "KeyName": {
      "Description" : "Name of an existing KeyPair to enable SSH access to the instances",
      "Type": "String",
      "MinLength": "1",
      "MaxLength": "255",
      "AllowedPattern" : "[\\x20-\\x7E]*",
      "ConstraintDescription" : "can contain only ASCII characters."
    },

  "SSHLocation" : {
      "Description" : "The IP address range that can be used to SSH to the VM instances",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern": "(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\\d{1,2})",
      "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
    }
  }

  "Mappings" : {
    "CloudOSInstanceType2Arch" : {
      "t1.micro"    : { "Arch" : "64" },
      "m1.small"    : { "Arch" : "64" },
      "m2.xlarge"   : { "Arch" : "64" },
      "m3.xlarge"   : { "Arch" : "64" },
```

```
      "c1.medium"   : { "Arch" : "64" },
    },

% The following information informs that this SF engages in a video acceleration chain
% indexed with ID "0X7516AB", and was instructed to forward packets with
% Destination IP ==20.20.20.1, Port=XXXX to the next SF on the chain
% with node ID == "0x abcdabcdabcdabcd" AND IP addr == "12.34.56.78"

    "ChainInstance" :{
      "ChainID": "0X7516AB",
      "Description": "Video Acceleration",
      "ProtocolType": "8080",
      "ForwardingContext":"DIP==20.20.20.1, Port=XXXX",
      "NextHopNodeUUID": "0X abcdabcdabcdabcd",
      "NextHopIP": "12.34.56.78",
    }
  }

  "Resources": {}
  "Outputs": {}
}
```

6. IANA Considerations

To be analyzed.

7. Security Considerations

To be analyzed.

8. References

8.1. Normative References

[RFC3775] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

8.2. Informative References

[Cfn] "AWS CloudFormation, http://aws.amazon.com/cloudformation/", .

[Heat] "OpenStack Orchestration, https://wiki.openstack.org/wiki/Heat.", .

[I-D.liu-sfc-use-cases]
Will, W., Li, H., Huang, O., Boucadair, M., Leymann, N., Cao, Z., Hu, J., and C. Pham, "Service Function Chaining (SFC) Use Cases", draft-liu-sfc-use-cases-02 (work in progress), February 2014.

[I-D.quinn-sfc-arch]
Quinn, P. and A. Beliveau, "Service Function Chaining (SFC) Architecture", draft-quinn-sfc-arch-04 (work in progress), January 2014.

[I-D.quinn-sfc-nsh]
Quinn, P., Guichard, J., Fernando, R., Surendra, S., Smith, M., Yadav, N., Agarwal, P., Manur, R., Chauhan, A., McConnell, B., and C. Wright, "Network Service Header", draft-quinn-sfc-nsh-01 (work in progress), February 2014.

[NFVE2E] "Network Functions Virtualisation: End to End Architecture, http://docbox.etsi.org/ISG/NFV/70-DRAFT/0010/NFV-0010v016.zip", .

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3118] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages", RFC 3118, June 2001.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

Authors' Addresses

Zhen Cao
China Mobile
Xuanwumenxi Ave. No. 32
Beijing 100053
China

Email: zehn.cao@gmail.com, caozhen@chinamobile.com

Hui Deng
China Mobile
Xuanwumenxi Ave. No. 32
Beijing 100053
China

Email: denghui@chinamobile.com