Network Working Group M. Hasan
Internet-Draft Cisco Systems, Inc.
Intended status: Informational A. Chari
Expires: December 09, 2013 D. Fahed
France Telecom - Orange Labs
M. Morrow
Cisco Systems, Inc.
June 07, 2013

Programmatic Interfaces to On-demand Network Services
draft-masum-chari-i2rs-netservices-00.txt

Abstract

One of the major features or requirements of Cloud is on-demand CRUD (Create / Read / Update / Delete) of Cloud resources or associated resources. The on-demand feature dictates that resources are CRUD in a time-frame in the order of seconds or few minutes since the arrival of the resource CRUD request. Many network resources cannot be CRUD in that time-frame. With the support of programmable networking (or SDN) in the model of I2RS will facilitate programming of network resources rapidly, thus facilitating CRUD of Cloud or related network resources on-demand. Network resources associated with many network services can be either a Cloud resource or directly associated with a Cloud resource. These resources should be CRUD on-demand in Cloud timescale. In this draft, employing Hybrid (virtual) Cloud as a use case and using I2RS as the "model", we will define few requirements for programmable on-demand interfaces to network services or I2NS.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

A Cloud service provider (CSP) offers services to tenants (enterprises, enterprise departments, employees, end consumers) out of one or more Cloud DCs, where a tenant can acquire or release (CRUD: Create, Read, Update, Delete) compute, storage or network resources on-demand and anytime. The CSP exposes Cloud service interfaces to tenants which are used by tenants to CRUD resources. Tenant facing Cloud service interfaces (CSI-T) are inherently

programmatic interfaces or API, supporting mostly REST-based API (such as Openstack Nova or Quantum or AWS EC2 API). The CSI-T exposes resources in Cloud abstractions. A Cloud controller or management framework (or CCF, which itself can be a collection of a number of systems, such as Openstack and SDN controller) maps or realizes the abstractions in the underlying infrastructure by further invoking underlying functions, interfaces or APIs (let us call this realization interfaces). For example, a request to create a virtual server via a CSI-T (such as in Openstack <service endpoint>/<API version>/<tenant ID>/servers, where "servers" is the Cloud resource abstraction in Openstack for virtual servers) can be mapped to Libvirt API and other calls to programmatically launch a VM. A CSI-T corresponding to CRUD of Cloud network resources will be mapped to underlying network resource or service programming interfaces. Consider, For example, a tenant creates a virtual Cloud via a CSI-T interface (see [SHC]), which can be mapped by a CCF to an MPLS VPN (VRF). In true Cloud fashion this virtual Cloud can be created or deleted on-demand and anytime or components of the virtual Cloud, such as on-premises (in private Cloud or intranet) or off-premises (in public Cloud DC) site or resource set or subnetwork can be CRUD on-demand elastically. As a result of this the associated VRF and its components (such as route targets) can be updated on-demand. Other examples are rerouting within the virtual Cloud and QoS guarantee requested on-demand (via CSI-T) based on dynamic conditions within the virtual Cloud or based on application requirements.

The CSI-T interfaces for Cloud services (such as Openstack or EC2 API) are programmatic allowing on-demand and elastic CRUD of resources (with Cloud timescale). The realization interfaces (such as Libvirt or other interfaces for compute and storage resource CRUD) are also programmatic. In a Cloud environment (together with the compute and storage services) the network or network service related interfaces also has to be programmatic so that CSI-T interfaces can be mapped to network service related interfaces on-demand. There is a need for defining such realization interfaces (let us call it interfaces to network services for Cloud or I2NS) that are programmatic and standardized. Employing a virtual Cloud use case, we outline the requirements behind the I2NS. We outline the potential interfaces that can be defined. The requirements and recommendations for the I2NS are general enough to apply to various Cloud related or other use cases requiring on-demand programmatic interfaces.

1.1. Acronyms and Definitions

Tenant: An enterprise, enterprise department, enterprise user or end consumer.

CRUD: Create, Read, Update, Delete (of resources, entities).

CSI-T: Tenant facing Cloud Service Interfaces.

PPVPN: Provider Provided VPN - VPN that is configured and managed by a service provider on behalf of a tenant.

PE: Provider Edge router.

CE: Customer Edge router.

VRF: Virtual Routing and Forwarding instance.

CSP: Cloud Service Provider - Owns or operates a public and hybrid Cloud.

ECRT: BGP Extended Community Route Target.

CCF: Cloud Controller or Management Framework.

I2NS: Interfaces to on-demand Network Services.

2. I2NS

The programmatic and elastic (dynamic) interfaces should be defined for network services or features in a way so that elastic (dynamic, incremental and decremental) invocation of those interfaces will be possible. In a typical configuration and provisioning model affecting elastic changes in the network or network services is either impossible or cumbersome. It is also time-consuming. In a Cloud environment elastic changes has to happen in a time-frame of few seconds to minutes. Consider for example, configuring MPLS VPN VRF in a typical environment. The VRF, import and export statements are configured a priori and applied. When, for example, there is need for adding new import or export route targets (RT), typically the entire VRF configuration block has to be updated and then change applied. Simple elastic programmatic interfaces that will affect changes in Cloud timescale and in finer granularity are needed. Consider for example a virtual Cloud which is mapped to an MPLS VPN in the network. As in a Cloud, resources and other entities (such as enterprise or CSP sites or regions or workgroups) can be added to or deleted from the virtual Cloud ondemand and anytime. These entities then should be reachable (if addedd) or unreachable (if deleted) from

within the virtual Cloud. As a result, the MPLS VPN (VRF) has to be updated on-demand (with export and import route functions).

Employing MPLS VPN network service as a use case, we provide examples of programmatic and elastic interfaces for network services suitable in a Cloud environment. The outline below is conceptual and can be considered as requirements to define exact (CRUD) interfaces. The interfaces should be defined in a way so that incremental, elastic and granular CRUD is possible. This I-D does not propose any particular syntax, protocol or language binding or model. Rather the interfaces are described in English, which can used used as a starting point or requirement to define the standard ones.

CreateMPLSL3VPNVRF (<name>, <PE>, <RD>) --> (returns) <VRFID>

CreateMPLSL3VPNImportRT (<list of RT>) --> <ImportRTID>

CreateMPLSL3VPNImportRTFilter (<ip prefix>) --> <ImportRTFilterID>

CreateMPLSL3VPNExportRT (<list of RT>) --> <ExportRTID>

CreateMPLSL3VPNExportRTFilter (<ip prefix>) --> <ExportRTFIlterID> and <ECRT> (ECRT: BGP Extended Community RT)

UpdateMPLSL3VPN (<VRFID>, <IMPORT or EXPORT>, <ImportRTID or ExportID or ImportFilterID or ExportFilterID>)

UpdateMPLSL3VPNImportRT (<importRTID>, <list of RT>)

UpdateMPLSL3VPNExportRT (<exportRTID>, <list of RT>)

UpdateMPLSVPNImportRTFilter (<ImportFilterID, <ip prefix>)

UpdateMPLSL3VPNExportRTFilter (<exportFilterID>, <ip prefix>).

UpdateMPLSL3VPNPE2CEInterface (<interface>, <VRFID>).

UpdateMPLSL3VPNPE2CENeighbor (<VRFID>, <neighbor IP>).

UpdateMPLSL3VPNPE2PENeighbor (<neighbor IP>).

Similarly read and delete operations.

Consider Figure 1, where it is shown that an enterprise (tenant) makes use of public Cloud services, which provides a service to CRUD a virtual Cloud that may span multiple Clouds (private and public), multiple enterprise sites and resources residing in both on-premises (in tenant intranet or private Cloud) and off-premises (in public

Cloud locations). Consider now that a tenant T1 making use of this service creates a virtual Cloud (via relevant CSI-T interfaces for CRUD of the virtual Cloud) as follows:

- Incorporate on-premises resource set ONPR-App1-0, public Cloud DC location CSP-DC-Loc1, and resource set OFPR-DMZ1. As a result following I2NS interfaces can be invoked (by a CCF) programmatically, on-demand and elastically:

-

   1. On PE-TS1 invoke CreateMPLSL3VPNVRF (VCL1, PE-TS1, <RD>) --> (returns) VRFID1 (where <RD> for example is 100:100).

   2. On PE-CL1: CreateMPLSL3VPNVRF (VCL1, PE-CL1, 100:100) --> (returns) VRFID2.

   3. On PE-TS1: CreateMPLSL3VPNExportRTFilter (<ip prefix of ONPR-App1-0>). This will create an extended community RT (ECRT1, such as 100:200) corresponding to IP prefix of ONPR-App1-0.

   4. On PE-TS1: UpdateMPLSL3VPN (VRFID1, EXPORT, ECRT1) (which should result in sending ECRT1 to all the other relevant PE via MP-BGP; in the example to PE-CL1).

   5. On PE-CL1: CreateMPLSL3VPNImportRT (ECRT1) --> iRT1.

   6. On PE-CL1: UpdateMPLSL3VPN (VRFID2, IMPORT, iRT1).

   7. On PE-CL1: CreateMPLSL3VPNExportRTFilter (<ip prefix of OFPR-DMZ1>) --> ECRT2.

   8. On PE-CL1: UpdateMPLSL3VPN (VRFID2, EXPORT, ECRT2).

   9. On PE-TS1: CreateMPLSL3VPNImportRT (ECRT2) --> iRT2.

   10. On PE-TS1: UpdateMPLSL3VPN (VRFID1, IMPORT, iRT2).

   11. On PE-TS1: UpdateMPLSL3VPNPE2CENeighbor (VRFID1, <CE11 IP>).

   12. On PE-TS1: UpdateMPLSL3VPNPE2PENeighbor (<PE-CL1 IP>).

   13. etc.

- When anytime later OFPR-DMZ-2 (in CSP DC-Loc2) is added to VCL1, following will be updated on-demand:

-

1. On PE-CL2: CreateMPLSL3VPNVRF (VCL1, PE-CL2, 100:100) --> VRFID3.

2. On PE-CL2: CreateMPLSL3VPNExportRTFilter (<ip prefix of OFPR-DMZ-2>) --> ECRT3.

3. On PE-CL2: UpdateMPLSL3VPN (VRFID3, EXPORT, ECRT3).

4. Update PE-CL2, PE-TS1, PE-CL1 with import RT of ECRT1, ECRT2 and ECRT3. In addition update neighbors accordingly.

Figure 1 shows an example of virtual (hybrid) Cloud together with the network topology.

```
   -------------------------
   | O ONPR-DB  O Other    | T1 Site 1       FIGURE 1
   | |          | Resources|
   | O FW -------          |
   | |                     |   HSW: Hypervisor Switch
   | O ONPR-App1-0         |   VLB/FW: Virtual Load-balancer/Firewall
   | |                     |   ER: Edge Router, CE: Customer Edge
   | O FW       O ONPR-    |   VM: Virtual Machine, PE: Provider Edge
   | |          | App-D-0  |
   | O ONPR-DMZ |          |
   | |          |          |   T2 Site 1       T1 Site 5      T1 Site 7
   |---------------------- | -------------  -------------   ------------
   | Tenant DC Network   | | |  |        |  |  |        |   |  |       |
   |---------------------- | |  |        |  |  |        |   |  |       |
   |    |                  | |  |        |  |  |        |   |  |       |
   |----O CE11-------------| |--O CE21---|  |--O CE15---|   |--O CE17--|
        |                       |              |               |
        -------------------------              -----------------
                 |                                     |
                 O PE-TS1------------------------------O PE-TS2
                 |   SP Private (IP/MPLS) MAN/WAN      |
   CSP           O PE-CL1------------------------------O PE-CL2 CSP
   DC-Loc1       |                                     |        DC-Loc2
   |-------------O ER-CL1-------------------| |--------O ER-CL2--------|
   |             |                          | |        |               |
   |            -------------------------   | | --------------------- |
   |            | CSP DC 1 Core/Aggr    |   | | |CSP DC 2 Core/Aggr  | |
   |            ---------O Access SW1----   | | ------O Access SW2---| |
   |                     |                  | |       | \              |
   |         ------------|-----------       | |       |   \            |
   |         |                      |       | |       |     \          |
   |      ---O HSW 1 ----------     O HSW 2 | | HSW 3 O       O HSW 4  |
   |      |           |       |     |       | |       |       |        |
   |      |           |       |     |       | |       |       |        |
   |      |           O VFW   |     |       | |       O VFW   |        |
   |      |           | T11   |     |       | |       | T12   |        |
   | ---------   ----------  ---- ------    | | -----------  --------- |
   |  |     |     |     |      |     |      | |  |      |       |      |
   |  O ... O     O ... O      O ... O ...  | |  O  ... O       O ...  |
   |  T1    T1    T1    T1     T2    T1     | |  T1     T1      T1     |
   |  VM11  VM12  VM13  VM14   VM21  VM15   | |  VM16   VM17    VM18   |
   |     OFPR-      OFPR-           OFPR-  | |     OFPR-       OFPR-  |
   |     DMZ1       App1-1          App-D-1| |     DMZ-2      App-D-2|
   ------------------------------------------ --------------------------
```

3. Security Considerations

Typical security considerations of network service updates will apply. In an on-demand dynamic Cloud environment certain security issues may be amplified. For example, uncontrolled BGP updates as resources are CRUD very dynamically (by a rogue entity). The dynamic application of configuration may amplify the situation of mistaken route leaking from one VPN to another. Hence proper steps should be taken.

4. IANA Considerations

There is no IANA consideration.

5. Conclusion

There is need for defining programmatic interfaces for various network services and features to effect on-demand and elastic changes in the network in a fast timescale as required in a Cloud environment. Employing virtual hybrid Cloud as a use case with its realization over a MPLS VPN network, we have outlined programmatic interfaces to network services (I2NS) as it pertains to MPLS VPN. With I2RS as the base framework, the I2NS for MPLS VPN and other Cloud relevant network services should be defined.

6. Acknowledgements

The authors would like to acknowledge contributions of Mohammad R. Rahman and Darren Y. Hu of UC Davis, California, for their valuable input for the programmatic interfaces.

7. References

7.1. Normative References

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.

[RFC4360] Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended Communities Attribute", RFC 4360, February 2006.

7.2. Informative References

[NIST] Mell, P. and T. Grance, "The NIST Definition of Cloud Computing", 800-145 NIST, September 2011, <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>.

[RFC2685] Fox, B. and B. Gleeson, "Virtual Private Networks Identifier", RFC 2685, September 1999, <http://tools.ietf.org/html/rfc2685>.

[SHC] Hasan, M., Chari, A., and B. et.al., "A framework for controlling Multitenant Isolation, Connectivity and Reachability in a Hybrid Cloud Environment", , September 2012, <http://tools.ietf.org/html/draft-masum-chari-shc-00>.

Authors' Addresses

Masum Z. Hasan
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
USA

Email: masum@cisco.com

Abdelhadi Chari
France Telecom - Orange Labs
2, avenue Pierre Marzin
Lannion , 22307
France

Email: abdelhadi.chari@orange.com

David Fahed
France Telecom - Orange Labs
2, avenue Pierre Marzin
Lannion , 22307
France

Email: david.fahed@orange.com

Monique Morrow
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
US

Email: mmorrow@cisco.com