Brownfield: Adding a vMX into AWS, then using Contrail Command Fabric manager to configure it as an SDN Gateway

Last edited by **Simon Green** 3 minutes ago

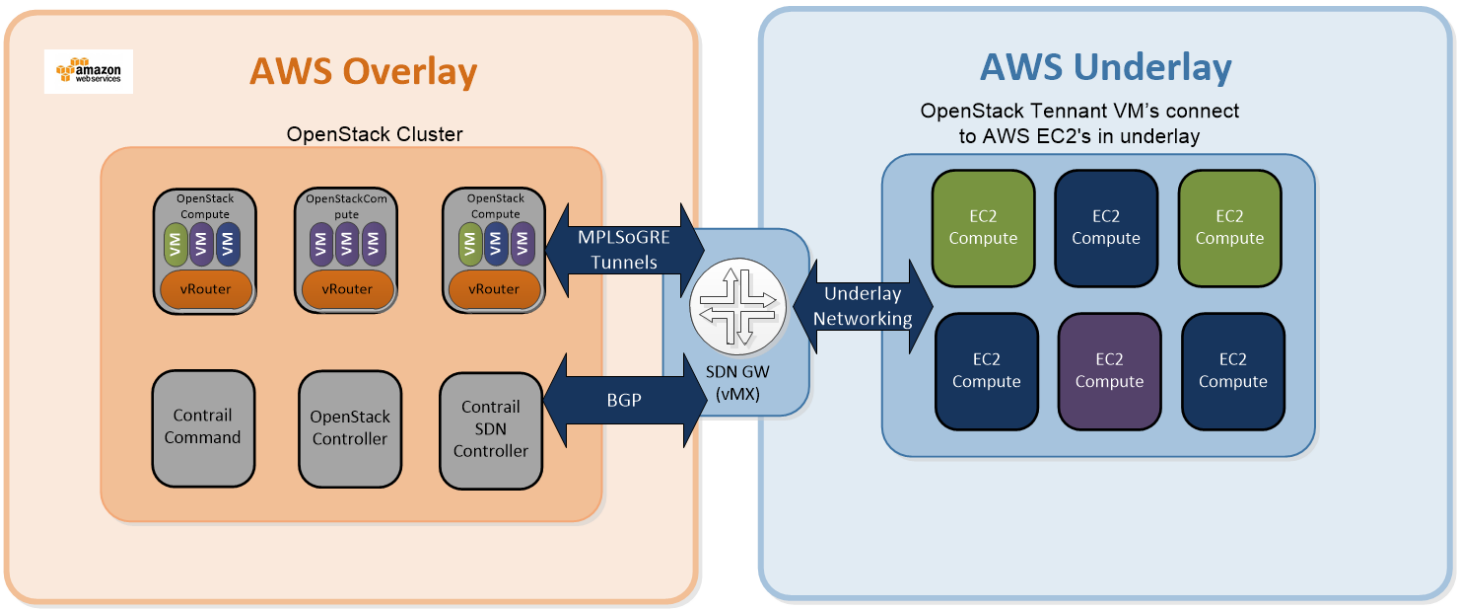
Brownfield - Adding an SDN Gateway (vMX) to an existing AWS setup using CloudFormation then using Contrail Commands Fabric to configure it

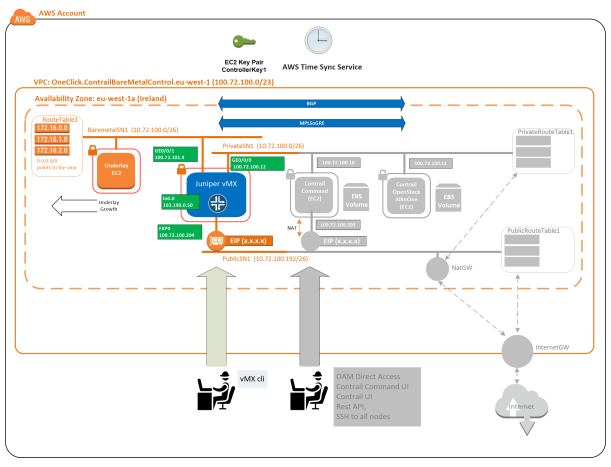
---

1. Here we will take the OpenStack all in one we deployed previously and add a vMX using CloudFormation
2. Then use Contrail Command Fabric manager to configure it as an SDN gateway (brownfield discovery).
3. We will then stretch an OpenStack SDN network out to the vMX and configure VRF routing to allow traffic between OpenStack (SDN networks in overlay) and a test AWS EC2 instance sat within an AWS Subnet (underlay subnet).
4. We will then test connectivity

Note: While here we are adding an SDN gateway to an existing OpenStack deployed into AWS. This procedure could also be applied to a Kubernetes or an OpenShift cluster that had been imported into Contrail Command. It could also be used on premise with physical MXs.

Note: To use Contrail Fabric Manager the cluster will need to be imported into Contrail Command. In this case as we deployed OpenStack using Contrail Command this is already the case. For other clusters deployed outside of Command you can import them directly, the OpenShift wiki has an example of a cluster import. The Contrail docs also show examples.





One Time Setup:

1. this wiki assumes you have already ran though the OpenStack all in one wiki and have a working Openstack up with two test networks net1 and net2. https://svl-ssd-git.juniper.net/sre/aws-one-click-deployers/wikis/Deploy-a-simple-OpenStack-test-setup-into-AWS-with-Contrail-Command
2. You need an AWS account. Login to the AWS console with your browser.
3. In your AWS account console you need to an EC2 SSH key pair named "ContrailKey" (unless you change the name in the stack parameters Note: You should be. able to deploy this stack in most regions within AWS, I'm using Ireland here.

Deploy the vMX CloudFormation stack

1. Click this link to launch the CloudFormation stack into your account (Ireland) Launch Stack

   Note: if you right click and copy this link, then share it with your customers via email. It will work for them as well

->NEXT
complete the field idVPC (The id of your existing VPC)
complete the field idSDNGatewayPublicSubnet1 (The id of your existing public subnet)
complete the field idControllersPrivateSubnet1 (The id of your existing public subnet)
complete the field idControllersPrivateSubnetRouteTable1 (The id of your Contrail Controller route table so we can inject routes for vMX ge-0/0/0 and loopback traffic)
complete the field idControllersSecurityGroup (The id. of the Contrail Controller security group so we can add rules into it for VMX traffic)
->NEXT
->NEXT
->Deploy Stack

If you prefer to use AWS cli to deploy the stack, here is the command with example parameter values

```
aws cloudformation create-stack \
--stack-name OpenStack-SDN-GW-Brownfield \
--disable-rollback \
--template-url https://s3-eu-central-1.amazonaws.com/contrail-one-click-deployers/vMX-SDN-G
--parameters \
ParameterKey=SGSubnet1,ParameterValue="100.72.100.0/23" \
ParameterKey=idVPC,ParameterValue="vpc-0b544eb12e2ba8ff2" \
ParameterKey=idSDNGatewayPublicSubnet1,ParameterValue="subnet-082c1e76f99fc0085" \
ParameterKey=idControllersPrivateSubnet1,ParameterValue="subnet-03ffa548c4f10827b" \
ParameterKey=idControllersPrivateSubnetRouteTable1,ParameterValue="rtb-05213ec54c1299773" \
ParameterKey=idControllersSecurityGroup,ParameterValue="sg-0fd225a20ff064161" \
ParameterKey=SDNGatewayFXP0PublicIP,ParameterValue="100.72.100.204" \
ParameterKey=SDNGatewayFXP0PublicIPSN,ParameterValue="26" \
ParameterKey=SDNGatewayAZ1PrivateIPGE000,ParameterValue="100.72.100.12" \
ParameterKey=SDNGatewayAZ1PrivateIPGE000SN,ParameterValue="26" \
ParameterKey=SDNGatewayAZ1PrivateIPGE001,ParameterValue="100.72.101.4" \
ParameterKey=SDNGatewayAZ1PrivateIPGE001SN,ParameterValue="26"
```

Check and if needed set the tunnel preference to MPLSoGRE

We have already set the tunnel preference. order to MPLS Over GRE, VxLAN, MPLS Over UDP. We did this in the OpenStack wiki. The AWS vMX AMI needs MPLSoGRE as first choice. MPOLSoUDP is either not supported or not functional, depending on the. AMI you use to deploy.
Should you need to change it manually, here are the steps.

1. Open your tungsten fabric UI towards the Contrail SDN controller within your OpenStack All-0In-One. You can get the link from the OpenStack CloudFormation outputs. example: https://[contrail-command-public-ip]:8143
2. Settings->Infrastructure->Global Configuration->Edit->Encapsulation Priority Order=MPLS Over GRE, VxLAN, MPLS Over UDP->Save

Changes to your OpenStack setup: (patch 1)

Patch1: OpenStack Device Manager
For Contrail Fabric to support the vMX we have to patch device manager.
To make this easier I've placed a patched image 2003 on Docker Hub for you. Steps below install it into OpenStack.
At the end of the wiki I'll also provide the steps for patching the image yourself, just in case you are running a different SW release.

1. ssh to your OpenStack SDN controller node (uses the OpenStack CloudFormation outputs to get the ssh command)

paste the steps below:

```
sudo bash
#edit /etc/contrail/config/docker-compose.yaml
--      image: "hub.juniper.net/contrail/contrail-controller-config-devicemgr:2003.33"
++      image: "docker.io/simonjohngreen/contrail-controller-config-devicemgr:2003.33-v3"

cd /etc/contrail/
docker-compose -f ./config/docker-compose.yaml down
docker-compose -f ./config/docker-compose.yaml up -d
```

```
contrail-status
docker ps | grep device | grep 2003.33-v3
```

Changes to your vMX config: (patch 2)

The vMX will currently throw an error complaining that its referencing a policy that it has not creating.
The steps below will fix this.

1. ssh to your vMX (uses the vMX CloudFormation outputs to get the ssh command)

paste the below cli:

```
configure
set groups __contrail_simon__ policy-options policy-statement _contrail_net1-l2-6-import te
set groups __contrail_simon__ policy-options policy-statement _contrail_net1-l2-6-import te
set groups __contrail_simon__ policy-options policy-statement _contrail_net1-l2-6-import te
set groups __contrail_simon__ policy-options policy-statement _contrail_net1-l2-6-import te
set groups __contrail_simon__ policy-options policy-statement _contrail_net1-l2-6-import th
set groups __contrail_simon__ policy-options policy-statement _contrail_net1-l2-6-export te
set groups __contrail_simon__ policy-options policy-statement _contrail_net1-l2-6-export te
set groups __contrail_simon__ policy-options policy-statement _contrail_net2-l2-7-import te
set groups __contrail_simon__ policy-options policy-statement _contrail_net2-l2-7-import te
set groups __contrail_simon__ policy-options policy-statement _contrail_net2-l2-7-import te
set groups __contrail_simon__ policy-options policy-statement _contrail_net2-l2-7-import th
set groups __contrail_simon__ policy-options policy-statement _contrail_net2-l2-7-export te
set groups __contrail_simon__ policy-options policy-statement _contrail_net2-l2-7-export te
set groups __contrail_simon__ policy-options community target_64512_1002 members target:645
set groups __contrail_simon__ policy-options community target_64512_1001 members target:645
set groups __contrail_simon__ policy-options community target_64512_8000001 members target:
set groups __contrail_simon__ policy-options community target_64512_1003 members target:645
set apply-groups __contrail_simon__
commit
exit
```

Use Contrail Fabric Manager to configure the SDN Gateway

Connect to the contrail command UI (use the OpenStack CloudFormation outputs for the link)

```
Infrastructure->Fabric->Create->Existing_Fabric->Provision

Name: ContrailSDNGateway
Device Credentials:
     username: jnpr
     password: EfrtGF5EDF_d54ERrf
Management Subnets:
     CIDR: 100.72.100.204/32 (note this is the public which container fxp0)
Loopback Subnets:
     CIDR: 192.90.0.55/32
->Next
Device Discovery:

Assign Roles:
     highlight, on the right click assign roles. complete as follows:
                                                                                Physical Role:
                                                                                Routing Bridgi
                                                                                        DC-Gat
                                                                                        Route

->autoconfigure
->next
->finish
```

To confirm that Fabric Manager has finished configuring you can check bgp status on the vMX (this takes 4-5 minutes before its configured)

```
show bgp summary
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active
100.72.100.11         64512         11          6       0       0          16 Establ
```

Extend the OpenStack net1 to the physical router

Connect to the contrail command UI

```
extend net1 to the vmx
     overlay->virtual networks->net1->edit->
                                          ->advanced->extend to the pyysical route (our
                                          ->route target 64512:1001
                                          ->external
```

There are many different ways to configure a vMX to breakout a VRF to underlay, we will use rib groups.

Create a rib group to allow the net1 l3 vrf (overlay) to route to inet.0. (underlay)

```
configure
set policy-options prefix-list overlay 192.0.0.0/16
set policy-options policy-statement leak-inet-to-vrf term 1 from protocol static
set policy-options policy-statement leak-inet-to-vrf term 1 from prefix-list overlay
set policy-options policy-statement leak-inet-to-vrf term 1 then accept
set policy-options policy-statement leak-inet-to-vrf term 2 then reject
set routing-options static rib-group inet-to-vrf
set routing-options static route 0.0.0.0/0 next-hop 100.72.100.1
set routing-options static route 100.72.101.0/26 next-hop 100.72.101.1
set routing-options rib-groups inet-to-vrf import-rib inet.0
set routing-options rib-groups inet-to-vrf import-rib _contrail_net1-l3-6.inet.0
set routing-options rib-groups inet-to-vrf import-policy leak-inet-to-vrf
commit
exit
```

Testing the setup

So at this point your OpenStack instances (in overlay) will be able to reach your EC2 instances within AWS (in underlay)

```
to connect to the test instances within the OpenStack all in one.
docker exec -it nova_libvirt virsh list
docker exec -it nova_libvirt virsh console 3
```

Get the ip address of the bare metal test instance from the console and ping it from an OpenStack instances in net1.

```
Example:
ping 100.72.101.49
PING 100.72.101.49 (100.72.101.49): 56 data bytes
64 bytes from 100.72.101.49: seq=0 ttl=63 time=1.944 ms
64 bytes from 100.72.101.49: seq=1 ttl=63 time=1.559 ms
```

if you would like to connect to the vMX bare metal instance directly and test in the other direction on the vmx

```
start shell
paste your ssh key into file  key.txt
chmod 0400 key.txt
ssh -i key.txt ubuntu@100.72.101.49
ping 192.0.2.3
PING 192.0.2.3 (192.0.2.3) 56(84) bytes of data.
64 bytes from 192.0.2.3: icmp_seq=1 ttl=62 time=4.66 ms
64 bytes from 192.0.2.3: icmp_seq=2 ttl=62 time=1.49 ms
```

Debugging hints

In order to watch device manager deploy the ansible. On the OpenStack all-in-one instance run this command

```
docker logs -f $(docker ps | grep devicemgr | cut -d" " -f1)
```

All Done

Notes: How to build your own patched image for device-manager instead of using mine

```
docker exec -i -t -e LINES=$(tput lines) -e COLUMNS=$(tput cols)  $(docker ps | grep device

edit /opt/contrail/fabric_ansible_playbooks/roles/cfg_overlay_evpn/templates/juniper_junos_
--      set groups {{cfg_group}} routing-instances {{contrail_l2_ri}} instance-type {{ri_ty
++      set groups {{cfg_group}} routing-instances {{contrail_l2_ri}} instance-type virtual

exit
docker login --username=[your dockerhub login] docker.io
docker commit $(docker ps | grep devicemgr | cut -d" " -f1)  contrail-controller-config-dev
docker tag $(docker image list | grep 2003.33-v3 | awk '{print $3}') docker.io/simonjohngre
docker push docker.io/[your docker hub login]/contrail-controller-config-devicemgr:2003.33-

#now get out deployment to pick up the patched 2003.33 image.
#edit /etc/contrail/config/docker-compose.yaml
--      image: "hub.juniper.net/contrail/contrail-controller-config-devicemgr:2003.33"
++      "docker.io/contrail-controller-config-devicemgr:2003.33-v3"
cd /etc/contrail/
docker-compose -f ./config/docker-compose.yaml down
docker-compose -f ./config/docker-compose.yaml up -d
contrial-status
docker ps | grep device | grep 2003.33-v3
```