# Milestone 2 : Routing as a Service

Submitted By :
Anupam Godse (angodse@ncsu.edu)
Ketul Patani (kpatani@ncsu.edu)
Prithvi Sharan (psharan@ncsu.edu)
Yash Kamdar (yjkamdar@ncsu.edu)

MILESTONE 1 FEEDBACK :

1. Review design choices
2. Incorporate multi tenancy

PRE MILESTONE 2 FEEDBACK :

1. Proposed a new architecture and discussed its merits and demerits
2. Incorporated multi tenancy

NEW DESIGN OVERVIEW :

1. Every tenant is assumed to have a VM (we call it Management VM) from where our solution is designed to be run. Management VM has a full view of the entire topology of the customer's network deployment, and serves as a point of command and control.
2. From the Management VM, the customer is able to perform(mainly) the following tasks :-
    a. Spawn Client VPC.
    b. Create subnets and attach VMs to said subnet.
    c. Create a Transit VPC which performs the core routing functionality.
    d. Peer Client VPC to Transit VPC.

CORE FUNCTIONALITIES OVERVIEW :

1. Spawn Client VPC :-

    a. Spawn VPC generates a VPC which be default comes with 2 Spine Router VMs
    b. Spine VMs act as the gateway for traffic that needs to traverse VPC boundaries, such as inter VPC traffic, external traffic (external wrt the hypervisor).
    c. Routing within a Client VPC is taken care of by Spine routers hosted within the VPC.

2. Create subnets and attach VMs to said subnets :-

    a. Within the scope of a VPC, the customer is given the functionality to define and create subnets.
    b. Spawning a subnet, results in the generation of a namespace (we call it Leaf NS) and a pure L2 Linux bridge.
    c. The Linux bridge serves as a point of aggregation of VMs that can be hosted on the newly created subnet.
    d. The leaf namespace created also has a link using which it connects to the Linux bridge, and acts as a DHCP server, leasing IP addresses to VMs that are also connected to the bridge.
    e. The leaf namespace also also connects itself to the 2 Spine VMs hosted within the VPC via L3 links. At this point, a static route is added on each of the Spine VM for the leaf subnet with the next hop being that of the corresponding leaf namespace.

3. Create a Transit VPC which performs the core routing functionality :-

    a. Transit VPC is used to connect subnets residing between multiple VPCs.
    b. Spawning a Transit VPC creates 2 Transit Router VMs.
    c. BGP sessions are built between each of the Spine Router in the Client VPC and the Transit Router in the Transit VPC (4 BGP sessions per VPC pair).

4. Peer Client VPC to Transit VPC :-

    a. As discussed in the previous section, BGP is used to peer Client VPCs to Transit VPCs.
    b. Transit VPCs are also used to peer Client VPCs running on multiple hypervisors : in this case for a given client, each hypervisor would host a Transit VPC and a BGP session is built between these Transit VPCs over a GRE tunnel.
    c. Path redundancy is provided due to the presence of 2 Transit Router VMs in each Transit VPC.

INTERFACE DEFINITIONS :

1. Northbound Interface :

    a. The customer is exposed to a YAML file that needs to be filled to perform any of the above functionalities described. 1 YAML file/operation.

2. Southbound Interface :

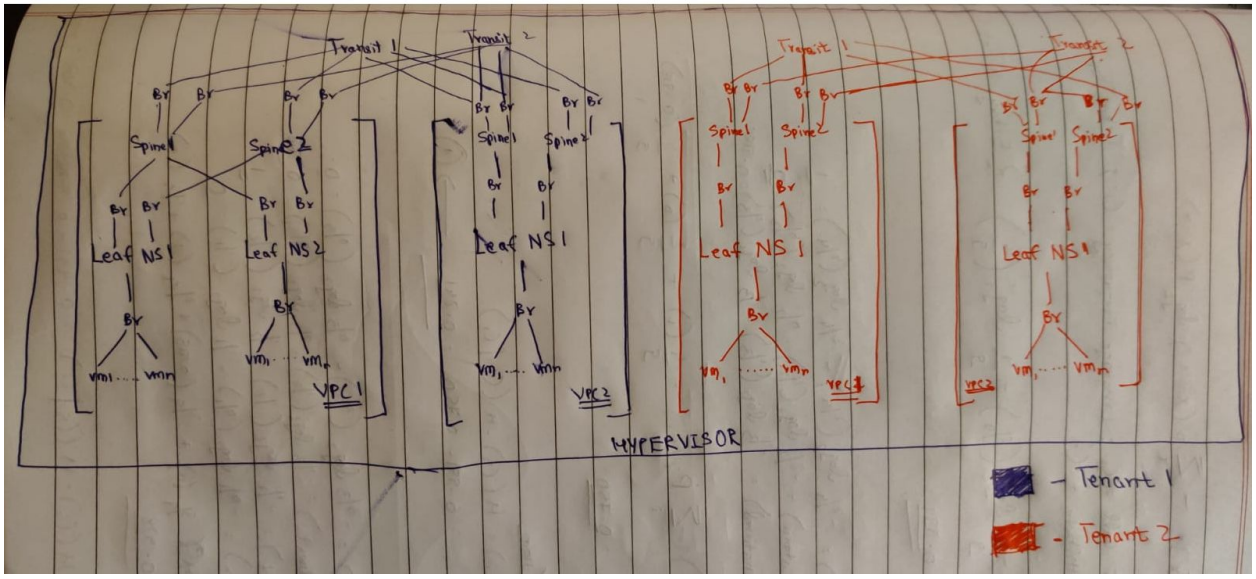
    a. The system is configured via shell scripts that directly run commands over the target Namespaces, VMs and Hypervisors.

3. Logic Layer :

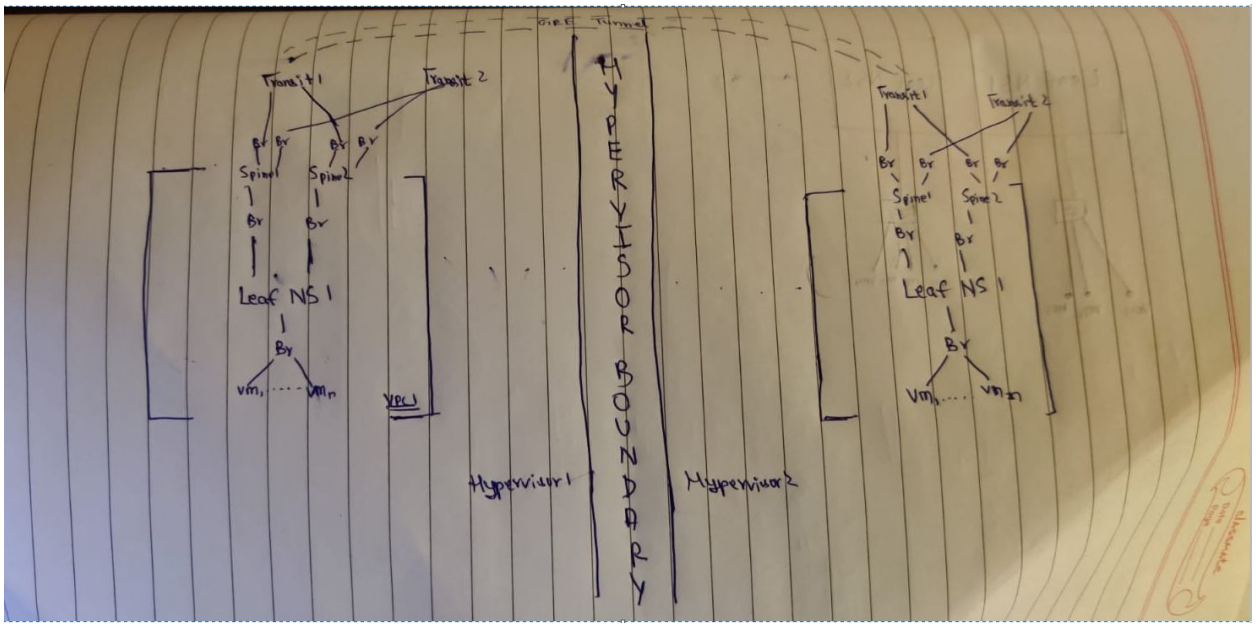
    a. Ansible acts as the rendezvous point between the Northbound and Southbound APIs.

PROJECT CURRENT SHAPE :

1. Implemented functional features of providing an easy to use routing platform.
2. Provided a management backplane to each networking VM which is accessible via the Management VM.
3. Provided a central point of deployment for the customer - which in later stages can also be used as the point of log collection, authentication and authorization.
4. Yet to implement additional functional features of fine tuning BGP path attributes to steer traffic via custom paths.
5. Yet to implement additional management features.




**Single Hypervisor Multi Tenant Communication**




**Multi Hypervisor Single Tenant Communication**
**(Concept can be extended to multi tenant scenario too)**