

Calico and BGP

Anirban Sen Chowdhary



Project Calico, a ***Tigera*** open-source project that provides a layer 3 network implementation, aimed at scalable datacenter deployments.

This slides explains describes how **calicoctl** is used for managing BGP.



What is BGP?



BGP which is also known as **Border Gateway Protocol** is a standardized exterior gateway protocol that is exclusively designed to exchange routing and reachability information among autonomous systems (AS) on the Internet.

This **BGP** protocol is often classified as a path vector protocol but is sometimes also classed as a distance-vector routing protocol.

It makes routing decisions based on paths, network policies, or rule-sets configured in a network.



Why Calico with BGP ?



Calico was designed to use the tools and techniques of the public Internet for scale-out network fabrics because it can run a large networks, with all the tools and techniques over the time. So we need to use the same tools as the scale-out world approaches Internet scale end point networks.

A VM based cloud in a modern era used to host thousands servers in a pod easily, and tens of thousands or even low hundreds of thousands of VMs end points as said in Calico terminology in that same pod. A container based cloud might increase the end point count by an order of magnitude or two.

In the **Calico** design, this would equate thousands of routers, and potentially millions of routes or end points. These numbers are not consistent with using an **IGP** or any other similar technology, but easily achieved by **BGP**, especially when we use route reflection to improve the router scaling number.

So we can see easily, **BGP** is the only option for this component of routing in a **Calico** network.



The Linux kernel generally knows about the **BGP** protocol.

Calico knows about the Linux kernel and how it works. So Calico can interact and ask kernel about other computers on the network to find the IP addresses in the system where it is and all the traffic for those IPs comes to the system.

It routes into Linux Kernel on each hosts and into physical fabric if required.

Calico use **BGP** as a routing protocol that tells other Calico nodes (and the rest of the infrastructure) where workloads are at any given point in time.



Summarizing in a point to use **BGP** to advertise the end points in a Calico network :

- Its very simple
- It is industry current best practice and trends following up today
- The only protocol that will sufficiently scale



Configuring BGP Peers :

There are many commands available in **calicoctl** for managing BGP. We will be looking into all those commands in our next few slides.

It is intended primarily for users who are running on private cloud and would like to peer Calico with their underlying infrastructure.

This slides covers configuration with **calicoctl** commands as examples:

- * Global default node AS Number
- * The full node-to-node mesh
- * Global BGP Peers
- * Node-specific BGP Peers



Global default node AS Number :

The **global default node AS Number** is the AS Number used by the **BGP** agent on a Calico node when it has not been explicitly specified. Setting this value simplifies configuration when your network topology allows all of your Calico nodes to use the same AS Number.

Example

To set the default AS number to 67111, run the following **calicoctl** command on any node:

```
$ calicoctl config set asNumber 67111
```

To view the current default value, run the command without specifying an AS number, the command will output the current value.

```
$ calicoctl config get asNumber 67111
```




The full node-to-node mesh :

The **full node-to-node mesh** option provides the mechanism which automatically configure peering between all Calico nodes. When enabled, each Calico node automatically sets up a **BGP** peering with every other Calico node in the network. By default this is enabled.

It is basically a simple mechanism for auto-configuring the BGP network in small scale deployments.

Example

If you need to turn the full BGP node-to-node mesh back on run the following command on any node:

```
$ calicoctl config set nodeToNodeMesh on
```

To view whether the BGP node-to-node mesh is on or off, enter the command without specifying the parameter, the command will output the current state.

```
$ calicoctl config get nodeToNodeMesh on
```




Global BGP Peers :

A **global BGP peer** is a **BGP** agent that peers with every Calico node in the network. A typical use case for a global peer might be a mid-scale deployment where all of the Calico nodes are on the same L2 network and are each peering with the same Route Reflector

Example

To add a global BGP peer at IP address 192.20.30.40 with AS number 67111 run the following command on any node:

```
$ cat << EOF | calicoctl create -f -  
apiVersion: v1  
kind: bgpPeer  
metadata:  
  peerIP: 192.20.30.40  
  scope: global  
spec:  
  asNumber: 67111  
EOF
```



Node-specific BGP Peers :

When different network topologies come in to play, a case where each Calico node peers with a Route Reflector in the Top of Rack (ToR) switch. In this case the BGP peerings are configured on a **per-node basis** also can be called as **node-specific peers**.

In the AS Per Rack model, each Calico node in a rack will be configured with a node-specific peering to the ToR Route Reflector.

Example

To add a BGP peer at IP address **aa:bb::ff** with AS number 67111 , peering with Calico node “node1”, run the following command on any node:

```
$ cat << EOF | calicoctl create -f -
apiVersion: v1
kind: bgpPeer
metadata:
  peerIP: aa:bb::ff
  scope: node
  node: node1
spec:
  asNumber: 67111
EOF
```




For more information visit

<https://www.projectcalico.org/>

<https://docs.projectcalico.org/v2.6/introduction/>

<https://blog.tigera.io/tagged/calico>



Thank You