

# Automate Leaf Configs using LeafBuilder (CVP Configlet Builder)

Version 5.3

[What is LeafBuilder?](#)

[Why do we need this?](#)

[Where can you use this?](#)

[LeafBuilder Requirements/Installation](#)

[LeafBuilder Limitations and Work in Progress](#)

[Getting Started with LeafBuilder](#)

[Steps to use the LeafBuilder](#)

[Assumptions](#)

[Location of latest Leaf Builder Configlet](#)

## What is LeafBuilder?

It's a CVP Configlet builder which helps building Leaf device configs based on inputting few parameters. The objective is to make Leaf Spine topology a template and use this Builder to config each of the Leaf Switches without manually configuring each one of them. Currently, it creates config for Leaf Spine Interface/IP config, BGP , MLAG configs , VXLAN, Underlay/Overlay SVIs, VRF, EVPN configs etc.

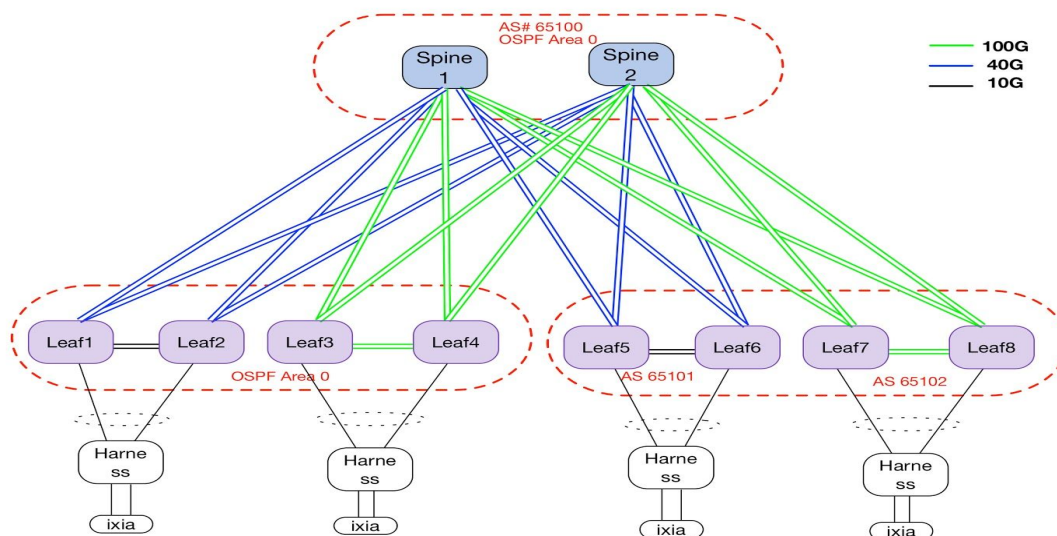
## Why do we need this?

It helps in building complex leaf device configs with BGP/OSPF (as Leaf Spine protocols, MLAG, Vxlan, EVPN etc) in matter of few seconds without having to manually configure each leaf device.

## Where can you use this?

Anywhere with Leaf Spine topology scaling > 2 PODs which requires manual config of Leaf Spine protocols, IP addresses, MLAG on all the Leaf switches. You can use it for <2 PODs as well, but, you could probably do that easily with manual config as well given the number of Leaf switches are less.

See the sample topology where i have used this LeafBuilder to configure the Leaf switches. I had 8 leaf switches which requires, Leaf Spine Interface configs, BGP config, MLAG config etc. All of that was done using this LeafBuilder simply providing only 8 inputs in the LeafBuilder form. (please see below form with filled details)



## LeafBuilder Requirements/Installation

What do you need in order to use this LeafBuilder configlet builder?

- CVP with latest version
- Download/extract netaddr python package source “netaddr-0.7.10.tar.gz” and “tar zxvf netaddr-0.7.10.tar.gz”.
- Install netaddr package “cd <netaddr-0.7.10>, python setup.py install”
- Create symlink in CVP Python library package for netaddr “ln -s /usr/lib/python2.7/site-packages/netaddr /cvpi/pythonlab/Lib/netaddr”

## LeafBuilder Limitations and Work in Progress

1. Current version supports leaf Spine IP, BGP/OSPF , MLAG Peer links, Port-channels towards Harness switch and SVIs with scale
2. Leaf to Spine Interfaces are provided without / for e.g if your Leaf Interface connecting to Spine is “49/1” , you input the Leaf to Spine Interface as “49” and /1 will be added by the script

## Getting Started with LeafBuilder

Steps to use the LeafBuilder

1. Design your topology with IP addresses/connections as per the xls template (if you have changed connections/IP addresses, you may need to edit this LeafBuilder script to adjust accordingly.
2. Import this LeafBuilder into your CVP node using CVP Configlet Import option
3. Go to Network Provisioning view,
  - a. You can selectively run this LeafBuilder for each Leaf switch
  - b. Fill the form with leaf-spine config details as per the template you have designed in Step 1 above for each switch
4. Once LeafBuilder generates the config and the same can be applied by running the task created

## Description of LeafBuilder Fields

1. Devices - Select the switch you want to generate config for (by default all switches will be selected, make sure to deselect the rest and select only the specific leaf switch you are generating config for)
2. Leaf Spine Protocol - Radio button to select OSPF or BGP
3. Leaf ASN - AS Number for the Leaf
4. Switch Number - As per the sample topology above, pick up the switch position based on where it is placed (e.g Leaf1=1 and Leaf2=2 , Leaf 8=8 or more)
5. Number of Spines - specify the number of Spines in your topology
6. Leaf to Spine Base Interface - Leaf interface to which Spines are connected, if 2 Spines and interfaces 49/1 and 50/1 are connected to Spines, just enter "49" - without quotes
7. Leaf to Spine Base IP Address - Leaf Ip address for connected interface towards Spine (/31 netmask assumed here)
8. Select MLAG Yes or No
9. If Yes, Peer Link IP Address - Ip address for the MLAG Peerlink
10. Select Underlay SVI - Yes or No
11. If Yes, Underlay SVI Start - SVI start for Underlay & Underlay SVI Start IP - SVI IP (starting) for Underlay, Number of Underlay SVI
12. Select Vxlan Config - Yes or No
13. If Yes, Enter Vtep Primary IP and Secondary (Varp) IPs
14. Select VRF required for Overlay Yes or No
15. If Yes, Number of Overlay VRFs required
16. Select Overlay SVI - Yes or No
17. If Yes, Select Re-Circ Interface required or not for Vxlan routing in T2 and Arad devices
18. Overlay Vxlan SVI start , Overlay Vxlan SVI IP starting range
19. Number of Overlay SVI per VRF
20. Bridging Only Overlay VLAN Start, Number of Bridging only Overlay VLANs (If Selected
21. Proving Spine Lo0 IP address if EVPN Config Yes Selected

## Assumptions

- LeafBuilder assumes the interfaces for Leaf Spine/MLAG peer links, Leaf to Harness switches etc. and subnet mask for IP addresses provided. If you want to change those, you need to edit the LeafBuilder script to change the same in appropriate section