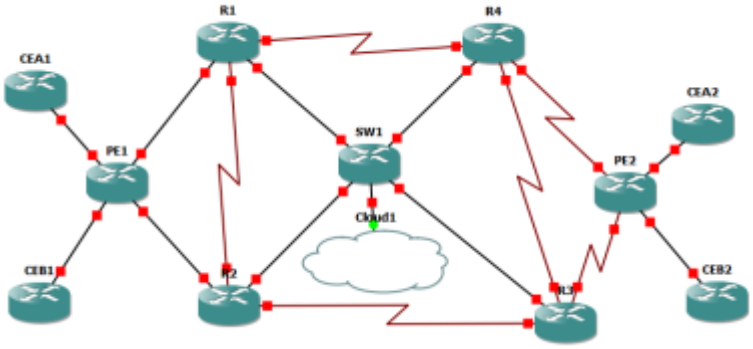
Table of Contents

L3 VPN Example

This is a very detailed example on how to build a L3 VPN, but could be anything, from scratch. This includes service types, templates, relations, scenarios, jobs, etc…

L3 VPN Create

The use case for which we are going to develop an end to end workflow is of provisioning a L3VPN. The Topology is as shown below.



- SW1, R1, R2, R3 and R4 are P Routers in a CSP named CSP
- PE1 and PE2 are PE routers
- The CE routers CEA1 for CustA and CEB1 for CustB are connected to PE1
- Similarly, the CE routers CEA2 for CustA and CEB2 for CustB are connected to PE2

CustA want a L3VPN from CEA1 to CEA2 and CustB want a L3VPN from CEB1 to CEB2.

To go about delivering L3VPNs in CSP network, we have to perform following steps.

1. Create Service Type
2. Create Template
3. Create Relation
4. Create Job
5. (optional) Create Form.

Some populating the database steps are necessary before we start with Service Type. They are:

1. Create Client Type
2. Create Site Type
3. Add Service Class
4. Add Node Class(es) and Node Type(s)
5. Add IPv4 Plan(s) and Assign it to Client Type

Step 1

- Create Client Type

- Provide details below :
  - Client Type : CSP
  - Description : Communications Service Provider

Step 2

- Create Site Type
- Create three Site Types.

1. for CE (with Static Routing and No Redundancy)
2. for PE
3. for CE (generic)

Static_No_Redundancy

- Site Type: Static_No_Redundancy
- Caption: Static_No_Redundancy

PE

- Site Type: PE
- Caption: PE

CE

- Site Type: CE
- Caption: CE

Step 3

- Create Service Class
- Now edit the Site Type “Static_No_Redundancy”
- And add the two Site Types PE and CE recently created to Service Class by clicking the button “New”

Step 4

Create Node Class and Node Types We need to create two Node class and node types for PE and CE respectively as below:

PE_C7600

- Client Type : CSP
- Service Class : PE

- Node Class: PE_C7600

CE_ISR3800

- Client Type : CSP
- Service Class : CE
- Node Class: CE_ISR3800

Step 5

Create Node Types

C7600

- Client Type: CSP
- Node Class: PE_C7600
- Node Type: C7600

ISR3800

- Client Type: CSP
- Node Class: CE_ISR3800
- Node Type: ISR3800

Now, for both the Node Types, Pls edit/update the following details

For Node Type: C7600

- Parameter: Template
- Function: literal
- Value1: C7600

- Parameter: Node_Position
- Function: literal
- Value1: NA

- Parameter: Domain
- Function: literal
- Value1: BOM

- Parameter: Enable_secret
- Function: literal
- Value1: netyce01 (must match what is on the real device's enable password)

For Node Type: ISR3800

- Parameter: Template
- Function: literal
- Value1: isr3800

- Parameter: Node_Position
- Function: literal
- Value1: ZA

- Parameter: Domain
- Function: literal
- Value1: BOM

- Parameter: Enable_secret
- Function: literal
- Value1: netyce01 (must match what is on the real device's enable password)

Step 6

Create IPv4 Plans
Click New and provide following details:

Loopbacks

PE_Loopback

- First IP Plan is for Loopbacks (for P, CE and PE routers)
- Plan Size: 24
- Description: PE_Loopback

Subnets

Under Subnets section, create following subnets

- Subnet Name: P_Loopbacks
- Subnet size: 32
- Start IP: 0.0.0.1
- End IP: 0.0.0.5

Subnet Plans

Under Subnet Plans section, create a new “Loopback Reference” plan with loopback reference as “lo0”

CE_Loopback

- Subnet Name: CE_Loopback

- Subnet size: 32
- Start IP: 0.0.0.6
- End IP: 0.0.0.10

Under Subnet Plans section, create a new “Loopback Reference” plan with loopback reference as “lo0”

PE_Loopback

- Subnet Name: PE_Loopbacks
- Subnet size: 32
- Start IP: 0.0.0.11
- End IP: 0.0.0.255

Under Subnet Plans section, create a new “Loopback Reference” plan with loopback reference as “lo0”

PE_CE

2nd IPV4 plan is for PE_CE links

- Plan Size: 24
- Description: PE_CE

Subnets

Under Subnets section, create following subnets

- Subnet Name: PE_CE
- Subnet size: 30
- Start IP: 0.0.0.0
- End IP: 0.0.0.255

Subnet Plans

Under Subnet Plans section, create a new “Point-to-point” plan

Template

We need to create several templates
1) vrf template
2) port templates
3) sub templates

4) Main templates

vrf template

template	template type	vendor type	model name
vrf_template	port	Cisco_IOS	vrf

Template text:

```
ip vrf <vrf_Name>
rd <Vrf_rd>
route-target both <Vrf_rt>
```

> Note: As with all templates, do not forget to set your new template to the 'production' state

port templates

template	template type	vendor type	model name
AddVrf	port	Cisco_ios	Ethernet

Template text:

```
#reload PE_CE_nets
interface <Port_type><Slot_id>/<Port_id>
ip vrf forwarding <Vrf_name>
ip address <If_ip_N> <Net_mask>
no shut
```

Sub templates

template	type	vendor
addMPbgp	automation	Cisco_IOS

Template text:

```
router bgp <Rtg_bgp_as>
address-family ipv4 vrf <Vrf_name>
redistribute static
redistribute connected
!
!
!
```

Service Type

Following Service Types need to be created
AddCE: To Add a CE with its loopback address
AddPE: To Add a PE with its loopback address
Create: To Create a L3VPN on a PE to PE basis

AddCE

Client Type: CSP
Service Class: CE
Service Type: api
Service Task: AddCE

Seq	Exec	Class	Scope	Match	Value	Alias	Comments
1	LOCATE	CLIENT	CURRENT	CLIENT_CODE	(client)	<customer>	
2	ADD	SITE	<customer>	SITE_CODE	(site)	<ce.site>	
3	ADD	SERVICE	<ce.site>	CURRENT	AddCE	<ce.svc>	
4	ADD	NODE	<ce.svc>	NODE_TYPE	ISR3800	<ce.node>	
5	ASSIGN	NODE	<ce.node>	NODE_NAME	(ce.name)		
6	ASSIGN	NODE	<ce.node>	TEMPLATE	ISR		
7	ADD	PORT	<ce.node>	LOOPBACK	Lo0	<ce.lo0>	
8	ADD	SUBNET	<ce.svc>	NET_NAME	CE_Loopback	<ce.subnet>	
9	LOCATE	ADDRESS	<ce.subnet>	ADDRESS_FIRSTFREE		<ce.add>	
10	ASSIGN	ADDRESS	<ce.add>	PORT	<ce.lo0>		

AddPE

Client Type: CSP
Service Class: PE
Service Type: api
Service Task: AddPE

Seq	Exec	Class	Scope	Match	Value	Alias	Comments
1	LOCATE	CLIENT	CURRENT	CLIENT_CODE	(client)	<client>	
2	ADD	SITE	<client>	SITE_CODE	(site)	<site>	
3	ADD	SERVICE	<site>	CURRENT	(pe.site)	<addpe.svc>	
4	ASSIGN	SERVICE	<addpe.svc>	SERVICE_NAME	AddPE		
4	ADD	NODE	<addpe.svc>	NODE_TYPE	C7600	<pe.node>	
5	ASSIGN	NODE	<pe.node>	NODE_NAME	(pe.name)		
6	ASSIGN	NODE	<pe.node>	TEMPLATE	c7600		
7	ADD	PORT	<pe.node>	LOOPBACK	Lo0	<pe.lo0>	
8	ADD	SUBNET	<addpe.svc>	NET_NAME	PE_Loopbacks	<pe.loopback>	
9	LOCATE	ADDRESS	<pe.loopback>	PICK_FORCED	(pe.lo0.ip)	<pe.lo0.ip>	
10	ASSIGN	ADDRESS	<pe.lo0.ip>	PORT	<pe.lo0>		
11	LOCATE	PORT	<pe.node>	PORT_TEMPLATE_FIRSTNOTOPO	AddVrf	<pe.port>	
12	ADD	SUBNET	<addpe.svc>	NET_NAME	PE_CE	<pe.ce.subnet>	
13	LOCATE	NODE	GLOBAL	NODE_NAME	(ce.node)	<ce.node>	

Seq	Exec	Class	Scope	Match	Value	Alias	Comments
14	LOCATE	PORT	<ce.node>	PORT_TEMPLATE_FIRSTNOTOPO	ISR3800	<ce.port>	
15	ADD	LINK	<pe.port>	PORT	<ce.port>	<pe.ce.link>	
16	ASSIGN	SUBNET	<pe.ce.subnet>	LINK	<pe.ce.link>		
17	ADD	VRF	<ce.node>	VRF_NAME	(vrf)	<ce.vrf>	
18	ADD	VRF	<pe.node>	VRF_NAME	(vrf)	<pe.vrf>	
19	ASSIGN	SUBNET	<pe.ce.subnet>	VRF	<ce.vrf>		
20	ASSIGN	SUBNET	<pe.ce.subnet>	VRF	<pe.vrf>		

Create

Client Type: CSP
Service Class: PE
Service Type: api
Service Task: Create

Seq	Exec	Class	Scope	Match	Value	Alias	Comments
1	LOCATE	CLIENT	CURRENT	CLIENT_CODE	(client)	<client>	
2	ADD	SITE	<client>	SITE_CODE	(site)	<site>	
3	LOCATE	SERVICE	GLOBAL	NODE	(pe.node)	<pe.l3vpn.svc>	
4	LOCATE	NODE	GLOBAL	NODE_NAME	(pe.node)	<pe.node>	
5	LOCATE	PORT	<pe.node>	PORT_TEMPLATE_FIRSTNOTOPO	AddVrf	<pe.port>	
6	ADD	SUBNET	<pe.l3vpn.svc>	NET_NAME	PE_CE	<pe.ce.subnet>	
7	LOCATE	NODE	GLOBAL	NODE_NAME	(ce.node)	<ce.name>	
8	LOCATE	PORT	<ce.node>	PORT_TEMPLATE_FIRSTNOTOPO	ISR3800	<ce.port>	
9	ADD	LINK	<pe.port>	PORT	<ce.port>	<pe.ce.link>	
10	ASSIGN	NODE	<pe.node>	NODE_POSITION	NA		
11	ASSIGN	NODE	<pe.node>	NODE_POSITION	ZA		
12	ADD	VRF	<ce.node>	VRF_NAME	(vrf)	<ce.vrf>	
13	ADD	VRF	<pe.node>	VRF_NAME	(vrf)	<pe.vrf>	
14	ASSIGN	VRF	<vrf>	VRF_TEMPLATE	vrf_template		
15	ASSIGN	VRF	<vrf>	SUBNET	<pe.ce.subnet>		

Relations

Relations are required to fetch information out of the database. Here this use case assumes we have PE, CE and VRF information with us beforehand

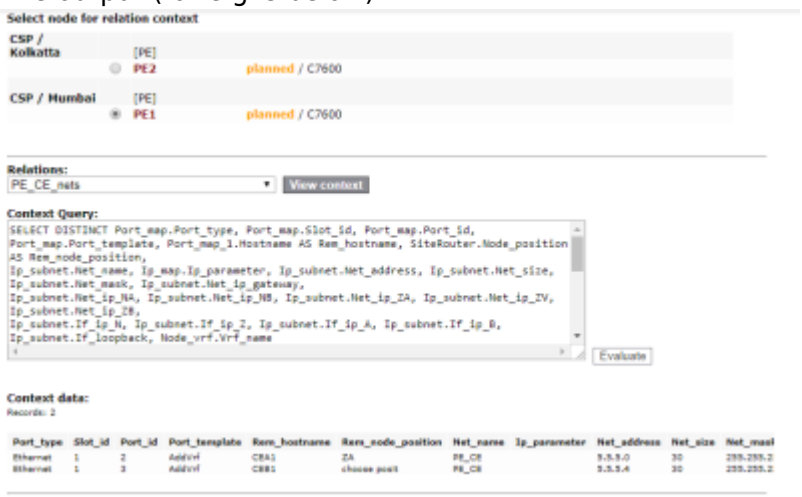
Thus with the variables at hand, how can we retrieve other parameters which the Job will need. For e.g, we shall need the PE Interface Name as an input, this will be retrieved using relations

Name:PE_CE_nets
Description: Show all connected ptp subnets to CE's from PE.
SQL:

```
SELECT DISTINCT Port_map.Port_type, Port_map.Slot_id, Port_map.Port_id,
Port_map.Port_template,
    Port_map_1.Hostname AS Rem_hostname, SiteRouter.Node_position AS
Rem_node_position,
    Ip_subnet.Net_name, Ip_map.Ip_parameter, Ip_subnet.Net_address,
```

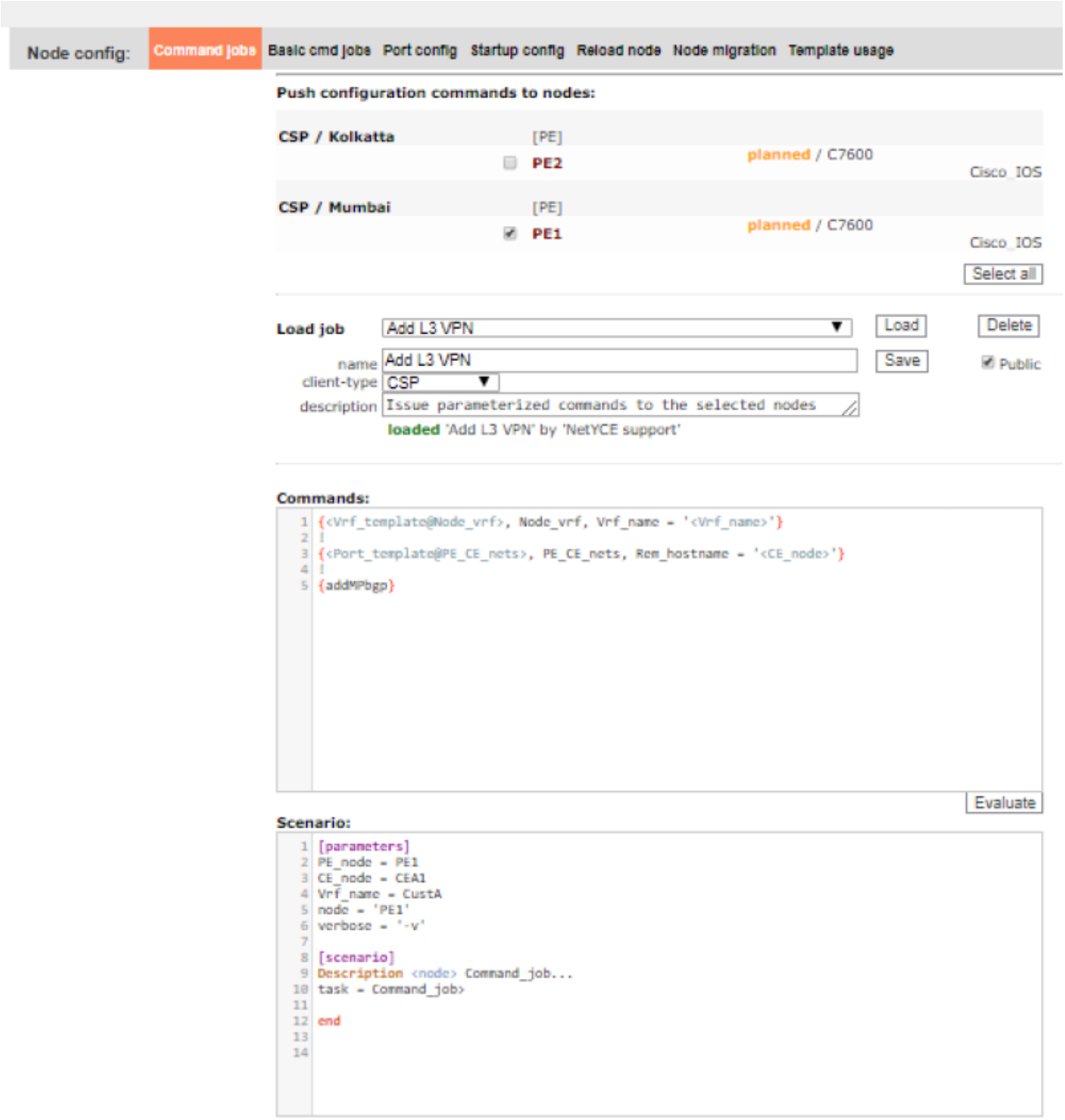
```
Ip_subnet.Net_size,
    Ip_subnet.Net_mask, Ip_subnet.Net_ip_gateway, Ip_subnet.Net_ip_NA,
Ip_subnet.Net_ip_NB,
    Ip_subnet.Net_ip_ZA, Ip_subnet.Net_ip_ZV, Ip_subnet.Net_ip_ZB,
Ip_subnet.If_ip_N,
    Ip_subnet.If_ip_Z, Ip_subnet.If_ip_A, Ip_subnet.If_ip_B,
Ip_subnet.If_loopback,
    Node_vrf.Vrf_name
FROM (((((Port_map
INNER JOIN Topo_map ON Port_map.Interface_id = Topo_map.Interface_id)
INNER JOIN Topo_map AS Topo_map_1 ON Topo_map.Topo_id = Topo_map_1.Topo_id)
INNER JOIN Port_map AS Port_map_1 ON Topo_map_1.Interface_id =
Port_map_1.Interface_id)
INNER JOIN SiteRouter ON Port_map_1.Hostname = SiteRouter.Hostname)
INNER JOIN Port_map AS Port_map_2 ON SiteRouter.Hostname =
Port_map_2.Hostname)
INNER JOIN (Ip_map
INNER JOIN Ip_subnet ON Ip_map.Subnet_id = Ip_subnet.Subnet_id) ON
Port_map_2.Interface_id = Ip_map.Interface_id
LEFT JOIN Node_vrf ON Ip_subnet.Vrf_id = Node_vrf.Vrf_id
WHERE Port_map.Hostname = '<hostname>'
    AND Port_map_1.Interface_id != Port_map.Interface_id
    AND Ip_subnet.Net_name = 'PE_CE'
ORDER BY Port_map_1.Hostname
```

The output (for e.g is below) :



And if we scroll to right to see the remaining columns
We can spot the IP addresses assigned and the VRF name too.

Command Job



As shown in the snapshot, The Command Job has two sections
1) Commands:
2) Scenario

The Commands section has below:

```
{<Vrf_template@Node_vrf>, Node_vrf, Vrf_name = '<Vrf_name>'}
!
{<Port_template@PE_CE_nets>, PE_CE_nets, Rem_hostname = '<CE_node>'}
!
{addMPbgp}
```

The {<Vrf_template@Node_vrf>, Node_vrf, Vrf_name = '<Vrf_name>'}

Here <Vrf_template@Node_vrf> is substituted with the lines in the Vrf_template by querying the relation Node_vrf.
Whilst Node_vrf returns a lot of columns, we are interested in Vrf_template and Node_vrf variable's values which are substituted in the Vrf_template's execution, PROVIDED Vrf_name = value_of_vrf_specified_in_Scenario
In other words
<Vrf_template@Node_vrf> will give the assigned vrf template which will be called upon. The variables within that template are filled with the values in Node_vrf (the second part of the function) per line. The last part in the function (optional) will filter the query output with the assigned value.

For e.g, this is the output we expect from the above line:

```
ip vrf CustA
rd  172.31.0.11:1
route-target both 65001:1
!
```

Here in the {<Port_template@PE_CE_nets>, PE_CE_nets, Rem_hostname = '<CE_node>'} line, we substitute the Port_template value from the relation PE_CE_nets
We also then expand the lines in the Port_template and start substituting parameters in PE_CE_nets PROVIDED Rem_hostname = value_of_CE_hostname_specified_in_Scenario

For e.g, this is the output we expect from the above line:

```
interface Ethernet1/2
ip vrf forwarding CustA
ip address  5.5.5.1  255.255.255.252
no shut
```

In the {addMPbgp} line, we are calling addMPbgp template. All the values in the variables within addMPbgp template are substituted by either explicit input provided in the Scenario.
For e.g, this is the output we expect from the above line:

```
router bgp 65001
address-family ipv4 vrf CustA
redistribute static
redistribute connected
!
```

Thus we have shown how to deploy L3 VPN on a PE for a set of parameters...

Scheduling the Job

This is the last part of the howto where the rubber meets the road. Here, the jobs are scheduled as per the time of the day, day of the week OR
NoW
And the device gets configured as per schedule.

We can see the session transaction log in Operate ⇒ Job Logs:

```
PE1#show privilege
Current privilege level is 15
PE1#show privilege
Current privilege level is 15
PE1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
PE1(config)#ip vrf CustA
PE1(config-vrf)#rd 172.31.0.11:1
PE1(config-vrf)#route-target both 65001:1
PE1(config-vrf)#!
PE1(config-vrf)#interface Ethernet1/2
PE1(config-if)#ip vrf forwarding CustA
PE1(config-if)#ip address 5.5.5.1 255.255.255.252
PE1(config-if)#no shut
PE1(config-if)#!
PE1(config-if)#router bgp 65001
PE1(config-router)# address-family ipv4 vrf CustA
PE1(config-router-af)#  redistribute static
PE1(config-router-af)#  redistribute connected
PE1(config-router-af)#  !
PE1(config-router-af)# !
PE1(config-router-af)#!
PE1(config-router-af)#end
PE1#
```

Modify and delete

Here are two more articles that explain the modification and deletion of the L3VPN.