Table of Contents

L3 VPN Example	3
L3 VPN Create	3
Step 1	3
Step 2	4
Step 3	4
Step 4	4
Step 5	5
Step 6	6
Template	7
vrf template	8
port templates	8
Sub templates	8
Service Type	9
Relations1	.0
Command Job	.1
Scheduling the Job	.3
Modify and delete	.4

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 "loo"

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 "loo"

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 "loo"

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

templatetypevendoraddMPbgpautomationCisco_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></customer>	
2	ADD	SITE	<customer></customer>	SITE_CODE	(site)	<ce.site></ce.site>	
3	ADD	SERVICE	<ce.site></ce.site>	CURRENT	AddCE	<ce.svc></ce.svc>	
4	ADD	NODE	<ce.svc></ce.svc>	NODE_TYPE	ISR3800	<ce.node></ce.node>	
5	ASSIGN	NODE	<ce.node></ce.node>	NODE_NAME	(ce.name)		
6	ASSIGN	NODE	<ce.node></ce.node>	TEMPLATE	ISR		
7	ADD	PORT	<ce.node></ce.node>	LOOPBACK	Lo0	<ce.lo0></ce.lo0>	
8	ADD	SUBNET	<ce.svc></ce.svc>	NET_NAME	CE_Loopback	<ce.subnet></ce.subnet>	
9	LOCATE	ADDRESS	<ce.subnet></ce.subnet>	ADDRESS_FIRSTFREE		<ce.add></ce.add>	
10	ASSIGN	ADDRESS	<ce.add></ce.add>	PORT	<ce.lo0></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></client>	
2	ADD	SITE	<client></client>	SITE_CODE	(site)	<site></site>	
3	ADD	SERVICE	<site></site>	CURRENT	(pe.site)	<addpe.svc></addpe.svc>	
4	ASSIGN	SERVICE	<addpe.svc></addpe.svc>	SERVICE_NAME	AddPE		
4	ADD	NODE	<addpe.svc></addpe.svc>	NODE_TYPE	C7600	<pe.node></pe.node>	
5	ASSIGN	NODE	<pe.node></pe.node>	NODE_NAME	(pe.name)		
6	ASSIGN	NODE	<pe.node></pe.node>	TEMPLATE	c7600		
7	ADD	PORT	<pe.node></pe.node>	LOOPBACK	Lo0	<pe.lo0></pe.lo0>	
8	ADD	SUBNET	<addpe.svc></addpe.svc>	NET_NAME	PE_Loopbacks	<pe.loopback></pe.loopback>	
9	LOCATE	ADDRESS	<pe.loopback></pe.loopback>	PICK_FORCED	(pe.lo0.ip)	<pe.lo0.ip></pe.lo0.ip>	
10	ASSIGN	ADDRESS	<pe.lo0.ip></pe.lo0.ip>	PORT	<pe.lo0></pe.lo0>		
11	LOCATE	PORT	<pe.node></pe.node>	PORT_TEMPLATE_FIRSTNOTOPO	AddVrf	<pe.port></pe.port>	
12	ADD	SUBNET	<addpe.svc></addpe.svc>	NET_NAME	PE_CE	<pe.ce.subnet></pe.ce.subnet>	
13	LOCATE	NODE	GLOBAL	NODE_NAME	(ce.node)	<ce.node></ce.node>	

Seq	Exec	Class	Scope	Match	Value	Alias	Comments
14	LOCATE	PORT	<ce.node></ce.node>	PORT_TEMPLATE_FIRSTNOTOPO	ISR3800	<ce.port></ce.port>	
15	ADD	LINK	<pe.port></pe.port>	PORT	<ce.port></ce.port>	<pe.ce.link></pe.ce.link>	
16	ASSIGN	SUBNET	<pe.ce.subnet></pe.ce.subnet>	LINK	<pe.ce.link></pe.ce.link>		
17	ADD	VRF	<ce.node></ce.node>	VRF_NAME	(vrf)	<ce.vrf></ce.vrf>	
18	ADD	VRF	<pe.node></pe.node>	VRF_NAME	(vrf)	<pe.vrf></pe.vrf>	
19	ASSIGN	SUBNET	<pe.ce.subnet></pe.ce.subnet>	VRF	<ce.vrf></ce.vrf>		
20	ASSIGN	SUBNET	<pe.ce.subnet></pe.ce.subnet>	VRF	<pe.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></client>	
2	ADD	SITE	<client></client>	SITE_CODE	(site)	<site></site>	
3	LOCATE	SERVICE	GLOBAL	NODE	(pe.node)	<pe.l3vpn.svc></pe.l3vpn.svc>	
4	LOCATE	NODE	GLOBAL	NODE_NAME	(pe.node)	<pe.node></pe.node>	
5	LOCATE	PORT	<pe.node></pe.node>	PORT_TEMPLATE_FIRSTNOTOPO	AddVrf	<pe.port></pe.port>	
6	ADD	SUBNET	<pe.l3vpn.svc></pe.l3vpn.svc>	NET_NAME	PE_CE	<pe.ce.subnet></pe.ce.subnet>	
7	LOCATE	NODE	GLOBAL	NODE_NAME	(ce.node)	<ce.name></ce.name>	
8	LOCATE	PORT	<ce.node></ce.node>	PORT_TEMPLATE_FIRSTNOTOPO	ISR3800	<ce.port></ce.port>	
9	ADD	LINK	<pe.port></pe.port>	PORT	<ce.port></ce.port>	<pe.ce.link></pe.ce.link>	
10	ASSIGN	NODE	<pe.node></pe.node>	NODE_POSITION	NA		
11	ASSIGN	NODE	<pe.node></pe.node>	NODE_POSITION	ZA		
12	ADD	VRF	<ce.node></ce.node>	VRF_NAME	(vrf)	<ce.vrf></ce.vrf>	
13	ADD	VRF	<pe.node></pe.node>	VRF_NAME	(vrf)	<pe.vrf></pe.vrf>	
14	ASSIGN	VRF	<vrf></vrf>	VRF_TEMPLATE	vrf_template		
15	ASSIGN	VRF	<vrf></vrf>	SUBNET	<pe.ce.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 the second s										
CSP / Kolkatta		(PE)								
	(PE2		planned / C760	0					
CSP / Hu	mbai	[PE]								
	0	PE1		planned / C760	0					
Relations										
PE_CE_n	ets			 View co 	mberet					
Context O	werv:									
SELECT OF	C 7 7 1 1 1 7 7			Barris and Charle	Ed. Doot over Door	+ 64				
DEPARTURE IN COMPANY		Port ma	D. POPT TYPE.	FOFT BAD, NACT	ad. POPT Hap.Mor					
Port map.	Port_te	mplate,	Port map 1.H	lostname AS Res	hostname, SiteR	outer.Node	position			
Port_map. AS Rem_no	Port_te de_posi	mplate, tion,	Port_map_1.H	lostname AS Re	_bostname, SiteR	outer.Node	position			
Port_map. AS Rem_no Ip_subnet	Port_te xde_posi .Net_na	mplate, tion, me, Ip_	Port_map_1.H map.Ip_parame	ter, Ip_subnet	LNet_address, Ip	outer.Node _subnet.Ne	_position			
Port_map. AS Rem_no Ip_subnet Ip_subnet	Port_te xde_posi .Net_na .Net_ma	mplate, tion, me, Ip_ sk, Ip_	Port_map_1.H map.Ip_parame subret.Net_ip	ter, Ip_subnet _gateway,	id, Port_map.ror _hostname, SiteRi t.Net_address, Ip	outer.Node	position			
Port_map. AS Rem_no Ip_subnet Ip_subnet	Port_te de_posi .Net_na .Net_na .Net_ip	mplate, tion, me, Ip_ sk, Ip_ JAA, Ip	Port_type, Port_map_1.H map.Tp_parame subret.Net_ip _subret.Net_i	ter, Ip_subnet _gateway, p_NB, Ip_subnet	t.Net_address, Ip_ et.Net_ip_IA, Ip_	outer.Node _subnet.Net subnet.Net	_position t_size, _ig_ZV,			
Port_map. AS Rem_no Ip_subnet Ip_subnet Ip_subnet	Port_te de_posi .Net_na .Net_ina .Net_ip	mplate, tion, me, Ip_ sk, Ip_ _NA, Ip_ _Z8,	Port_map_1.H Map.Ip_parame subnet.Het_ip _subnet.Het_i	ter, Ip_subnet gateway, p_NB, Ip_subnet To subset I	<pre>_id, Fort_Map.ror _hostname, SiteRi t.Net_address, Ip_ et.Net_ip_IA, Ip_;</pre>	_subnet.Net	_position t_size, _ig_ZV,			
Port_map. AS Rem_no Ip_subnet Ip_subnet Ip_subnet Ip_subnet Ip_subnet	Port_te wde_posi .Net_na .Net_ina .Net_ip .Net_ip .If ip	mplate, tion, mt, Ip_ sk, Ip_ _NA, Ip _Z8, N, Ip_s	Port_map_1.H Nop.Ip_paramet subnet.Net_ip _subnet.Net_i Unnet.If_ip_2 Node_vef_Vef	<pre>cort_map.siot (ostname AS Res gateway, p_N8, Ip_subnet.If . Ip_subnet.If</pre>	_io, Fort_Map.For _hostname, SiteR t.Net_address, Ip_ et.Net_ip_IA, Ip_; f_ip_A, Ip_subnet		_position t_size, _ig_ZV,			
Port_map. AS Rem_mo Ip_subnet Ip_subnet Ip_subnet Ip_subnet Ip_subnet	Port_te ode_posi i.Net_na i.Net_ip i.Net_ip i.Net_ip i.If_ip i.If_ioo	port_ma mplate, tion, me, Ip_ ak, Ip_ _NA, Ip_ _28, N, Ip_s pback,	Port_type, Port_map_1.H map.Ip_parame subnet.Net_ip _subnet.Net_i Node_vrf.Vrf_	<pre>cort_map.siot (ostname AS Res gateway, p_N8, Ip_subnet. , Ip_subnet.If name</pre>	sh, Port_map.nor _hostname, SiteR t.Net_address, Ip_ st.Net_ip_IA, Ip_; f_ip_A, Ip_subnet		_position t_size, _ig_ZV, _	Feelenter		
Port_map. AS Rem_no Ip_subnet Ip_subnet Ip_subnet Ip_subnet Ip_subnet Ip_subnet	Port_te de_posi .Net_na .Net_ip .Net_ip .Net_ip .If_ip .If_loo	port_ma mplate, tion, me, Ip_ sk, Ip_ _NA, Ip_ _Z8, N, Ip_s pback,	Port_type, Port_map_1.H map_1p_paramet subnet.Net_ip ubnet.If_ip_2 Node_vrf.Vrf_	rort_map.sion (ostname AS Res _gateway, p_N5, Ip_subnet. , Ip_subnet. name	<pre>_hostname, SiteB _hostname, SiteB t.Net_address, Ip et.Net_ip_IA, Ip_subnet f_ip_A, Ip_subnet</pre>	w, outer.Node _subnet.Net subnet.Net .If_1p_0,	position t_size, _ig_ZV, *	Evaluate		
Port_map. AS Rem_nc Ip_subnet Ip_subnet Ip_subnet Ip_subnet Ip_subnet	Port_te wde_posi .Net_na .Net_ip .Net_ip .Net_ip If ip_	port_ma mplate, tion, me, Ip_ sk, Ip_ _NA, Ip_ ZB, N, Ip_s pback,	Port_type, Port_map_1.H map.Ip_parame subnet.Net_ip _subnet.Net_i Node_vrf.Vrf_	<pre>rort_map.sion (satname AS Res _gatesay, p_NS, Ip_subre , Ip_subret.If name</pre>	io, port_empiror _hostname, SitaRi t.Net_eddress, Ip et.Net_ip_IA, Ip_subnet	Node _subnet.Net subnet.Net .If_1p_0,	position t_size, _ig_ZV, *	Evaluate		
Port_map. AS Rem_no Ip_subnet Ip_subnet Ip_subnet Ip_subnet Ip_subnet Ip_subnet Records: 2	ata:	more_ma mplate, tion, sk, Ip_ sk, Ip_ _NA, Ip_ _28, N, Ip_s pback, I	p.rort_type, Port_mapl.H map.Ip_parame subnat.Net_p _subnat.Net_t _subnat.Net_t Node_vrf.Vrf_	rort_map.sion (schimme AS Res (ter, Ip_subnet) _gateway, p_NS, Ip_subnet. I, Ip_subnet.If name	Jul port_emp.our /hostname, SiteR t.Met_address, Ip et.Net_ip_IA, Ip_ f_ip_A, Ip_subset	Node subnet.Net subnet.Net .If_1p_0,	_position t_size, _ip_IV,	Evaluate		
Port_map. AS Rem_no Ip_subnet Ip_subnet Ip_subnet Ip_subnet Ip_subnet Ip_subnet Ip_subnet Records: 2	shami shami	Port_kd	Port_map_l.H Rop.Tp_portame_l.H subset.Het_l _subset.Het_i ubset.If_ip_2 Node_vrf.VrT_	<pre>rer_map.suc. isstname.AS Ren ter, Ip_suber _patemay, p_NB, Ip_suber , Ip_subert.It name</pre>	Rem_mode_position		position t_size, .:dp_ZV, *	Evaluate Net_address	Net size	Net mas
Port_map. AS Rom_no Ip_subnet Ip_subnet Ip_subnet Ip_subnet Ip_subnet Ip_subnet Ip_subnet Ip_subnet P_subnet P_subnet Ip_subnet I	shami de_post t.Net_na t.Net_na t.Net_ip t.Net_ip t.Net_ip t.If ip_ t.If ip_ t.If is_ t.If is_	Port_id Port_id 2	Port_coper. Port_map_l.H map_Ip_parame submat.Net_p _submat.Net_p _bonet.Het_i Node_vrf.Vrf_ Port_template _addref	ror_map.subnet sottame AS Re- patenay, pJHS, Ip_subnet. I, Ip_subnet. Rem_hostname CRA1	Ray node_position the state of the state of		position t_size, _ig_ZV, * }	Evaluate	Net_size	Net_mail 235.255.2

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

	Net_ip_gateway	Net_ip_NA.	Net_ip_NB	Net_ip_ZA	Rel_ip_ZV	Net_ip_28	N_QUE	15_10_2	IL IA.A	1.00.0	2f_loophack	Wf_name
5.252							5.5.5.1	5,5,5,2	5,5,5,1	5,5,5,2		Conta.
5,252							5.5.5.5	5.5.5.6	5.5.5.5	5.5.5.6		Cust5

Command Job

Node config:	Command jobs	Basic cmd jobs	Port config	Startup c	onfig R	eload node	Node mig	ration Templa	ate usage	
		Push configur	ration com	mands to	o nodes	:				
		CSP / Kolkatt	ta		[PE] PE2			planned / C	7600	Cisco_105
		CSP / Mumba	i	×	[PE] PE1			planned / C	27600	Cisco_IOS
		Load job name client-type description	Add L3 VPN Add L3 VPN CSP Issue para Ioaded 'Add	N T ameteriz d L3 VPN	ed com	ands to 1 YCE suppo	the selec	ted nodes	Load Save	Delete Public
		Commands: 1 (<vrf_ter 2 I 3 (<port_ter 4 I 5 (addMPbgp</port_ter </vrf_ter 	nplate@Node_ omplate@PE_C >}	_vrf>, No	ode_vrf,	Vrf_name nets, Rem_	= '«Vrf_n hostname	ame>'} = ' <ce_node>'</ce_node>	'}	
		Scenario: 1 [paramete 2 PE_node - 3 CE_node - 4 Vrf_name 5 node = 'F 6 verbose - 7 8 [scenario 9 Descripti 10 task = Co 11 12 end 13 14	ers] - PE1 - CEA1 - CustA >E1' - '-v' b] command_job>	Command_j	job					Evaluate

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 \Rightarrow 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.

From: https://wiki.netyce.com/ - **Technical documentation**

Permanent link: https://wiki.netyce.com/doku.php/guides:user:I3vpn:I3vpn



Last update: 2022/04/29 07:42