L2VPN Service Automation with Anuta ATOM

Highly Scalable Vendor Agnostic Service Orchestration

Key Capabilities

- Supports IETF YANG & OpenConfig Models
- Ensures Service Compliance
- Automated delivery of
 Model-driven Service UI
- Supports brownfield network
 discovery
- Supports complete lifecycle management of services
- Dry-run mode to validate command generation
- Vendor Agnostic
- Integration into Internal &
 External IPAM, VLAN, RD &
 RT Manager
- Workflow Integration for Pre & Post Validations
- Quickly scale up to 1M+ devices across 45+ vendors

It is a well-known fact that Service Providers & Enterprises take time to introduce new services to the market. The networks end up playing an antagonist in most cases, purely due to how they are managed. In this digital age, as the world is getting ready to embrace the high speeds of 5G, existing service provisioning processes do not scale or meet the agile market requirements. With the growing OTT competitors' who preach rapid service delivery and intolerance among customers on the lack of timely delivery of services, it is evident that network service fulfillment is the elephant in the room.

anuta networks

Multi-vendor networks, manual processes, varied configuration data sets, the introduction of new vendor platforms, siloed automation, integration hassles, and proprietary vendor solutions contribute to this ineffective and passive delivery of services.

Anuta ATOM is built ground up to support agile network service delivery. It introduces network service orchestration as a replacement for the existing methodologies of service provisioning. ATOM utilized YANG models as the underpinning to facilitate end-to-end complex service provisioning in heterogeneous networks. ATOM's service models go beyond the initial provisioning of services and manage the complete lifecycle to ensure full control. It goes a long way to accommodate the dynamic nature of today's businesses. The multi-vendor support of 150+ platforms across 45+ vendors enables Anuta ATOM to be the force that propels service delivery in organizations.

The service models in Anuta ATOM is built on the following principles to form the network abstraction layer:

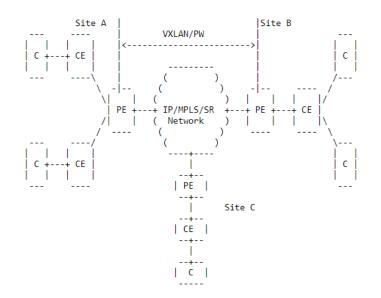
- 1. Offers a model-driven view of services and their configurations
- 2. Offers a model-driven view of devices by supporting both Native YANG & OpenConfig
- 3. Maintains the device & service configured state to support CRUD operations
- 4. Built-in exception handling to address unexpected issues during service provisioning
- 5. Dry run mode enables complete visibility to network teams before command provisioning
- 6. Supports atomic transaction to ensure data consistency across the network
- 7. Supports integration into other ecosystem tools such as IPAM
- 8. Integrated to ATOM's workflow engine for Pre & Post Service validations

Anuta ATOM supports several out-of-box service models. This catalog includes a wide range of services for Service Providers, Branch, Campus, and Datacenter networks. Some of the IETF & custom service models delivering VPN services are discussed below.

IETF L2VPN Automation

Anuta ATOM offers out-of-box support for IETF L2VPN. ATOM'S L2VPN model is based on RFC 8466 YANG data model for a Layer 2 provider-provisioned VPN service. The YANG data model supports Virtual Private Wire Services (VPWS) and multipoint Virtual Private LAN Services (VPLS) that uses Pseudowire signaled using LDP and BGP. The model defines service configuration elements that can be used in communication protocols between customers and network operators.

The L2VPN service model is architected as a collection of sites that exchange traffic over a shared infrastructure. The provisioned model will deliver an end-to-end layer 2 connectivity between two or more customer sites.



ATOM's L2VPN Service model offers lifecycle management through an abstracted interface to request, configure, and manage L2VPN service components. The configuration of network elements may be done using the CLI or other southbound interfaces such as NETCONF in conjunction with ATOM's device models based on CLI, Native YANG, or OpenConfig.

ATOM L2VPN Service Model Design

The L2VPN service model is structured such that the Service Provider can list the multiple circuits that it serves for the same customer. The YANG module is divided into two primary containers: *sites* & *vpn-services*.

Sites

A site represents a connection of a customer office to one or more VPN services. The site container stores information regarding detailed implementation arrangements made with the customer. In ATOM, the following critical characteristics of a site are captured.

- 1. Site-Id: This field provides a unique identification of the site in the overall network infrastructure.
 - 1. *Site-VPN-Flavor*: Defines the way that the VPN multiplexing is done, e.g., whether the site belongs to a single VPN site or a multi-VPN site. The following options are supported.
 - i. Site-Vpn-Flavor
 - ii. Site-Vpn-Flavor-Multi
 - iii. Site-Vpn-Flavor-Single
 - iv. Site-Vpn-Flavor-Nni

By default, the site belongs to a single VPN.

- 2. Bundling-Type: The following options are supported
 - i. Bundling-Type
 - ii. All2one-bundling
 - iii. One2one-Bundling
 - iv. Multi-Svc-Bundling

By default, each L2VPN can be associated with only one CE-VLAN, i.e., one-to-one bundling is used.

3. Default-Ce-Vlan-Id: The default VLAN Id set at the site level.

In the example illustrated in this brief, there are two sites, site-1, and site-3.

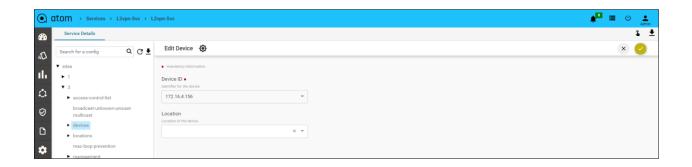
● atom → Services :	> L2vpn-Svc >	2vpn-Svc
Service Details		
D Search for a config	Q C 🖣	Edit Site 🔕
▶ sites vpn-profiles		mandatory information
vpn-protites		Site-Id Identifier of the site.
		1
Ð		Site-Vpn-Flavor Defines the way that the VPN multiplexing is done, e.g., whether the site belongs to a sit
2		Site-Vpn-Flavor-Single Site-Vpn-Flavor-Nni
3		Description Transit description of a site and using this as site name
		site1
		Bundling-Type Bundling type. By default, each LZVPN can be associated with only one CE-VLAN, i.e., o
		Bundling-Type All2one-Bundling One-2one-Bundling Multi-Svc-Bundling
		Default-Ce-Vian-Id Default-CE-VIAN IG set at the site level. 899



atom > Services > L2vpn-Svc > L2v	pn-Svc	单 🔲 🙂	Admin
Service Details		:	S :
Search for a config Q C 🕹	Edit Site 🧑	×	0
 interimentation vpr-profiles vpr-services 			

 Devices: It denotes the Provider Edge (PE) devices to connect the Customer Edge (CE) devices. One or more PE devices can be created as part of the devices list. In the example, site-1 hosts PE device 172.16.4.99, while site-3 hosts PE device 172.16.4.156

•	atom > Services > L2vpn-Svc > L	2vpn-Svc	📌 🔲 😃 🚊
-	Service Details		s 🛨
D.	Search for a config Q 💆	Edit Device 🔞	× 🕗
ıĿ	▼ sites ▼ 1	mandmay information Device ID	
٥	 access-control-list broadcast-unknown-unicast- multicast 	Identifier for the device. 172.16.4.99	
Ø	devices locations	Location Location of the device.	
	mac-loop-prevention	× •	



Creating a site in ATOM L2VPN service acts as a resource model that does not generate any commands. The site holds the device details involved in the L2VPN service.

3. Site-bearers: Once the sites are created listing the devices involved, the physical port connected to the CE device is defined.

In the example below, the device 172.16.4.99 under Site-1 uses port xe-0/1/9

	atom > Services > L2vpn-Svc > L2	ypn-Svc	*	=		Admin
a	Service Details					<u>د</u> ±
ı.E	Search for a config Q C 🛓	Edit Bearer 🔞		*	< (2
ıl.	▼ sites ▼ 1	enadersy information Bearerid •				
٩	 access-control-list broadcast-unknown-unicast- 	Unique identifier for a heaver. This identifies alkal be mapped to the heaver enformance an				
Ø	multicast devices	Device IP • device in the second seco				
۵	 locations mac-loop-prevention 	172.16.4.99 *				
	management service site-bearers	Portide ● Port of the PF locate from the given bearse: xe-0/1/9 X ▼				

Device 172.16.4.156 under Site-3 uses port xe-0/0/4

● atom → Services → L2vpn-Svc → L	2mp-Svc	* ••	ا ر	,	Admin
Service Details				\$	Ŧ
Search for a config Q C 🛓	Edit Bearer 🔞		×	C	
vites 11. ► 1	mandatury information Bearer-Id				
access-control-list	Usique identifier for a besier. This identifies shall be mapped to the besier reflexence on				
broadcast-unknown-unicast- multicast	Device IP • device junterena.				
devices locations	172.16.4.156 *				
mac-loop-prevention management service	Port-Id Port of the PE soute for the given beaver. x+O(0)(4 X *				

VPN Service

The VPN service in the L2VPN service model captures the generic information about the VPN service. It denotes the VRF for a customer on the Provider Edge (PE), allowing the use of overlapping IP addresses. The *vpn-service* list is composed of the following main attributes:

- 1. Vpn-Id: A vpn-id list uniquely references VPN services that a service provider offers. It reflects a VRF name on the device.
- 2. *Customer-Name*: This field identifies the customer served by the Service Provider.
- 3. *Vpn-Svc-Type*: This identifies the offered flavor of L2VPN services. ATOM supports VPLS and VPWS, which is the default value.
- 4. *Svc-Topo*: This identifies the type of VPN service topology. The topology could be one of Any-to-Any, Hub-Spoke, Hub-Spoke-Disjoint & Vpn-Topology. Any-to-Any appears as the default value.
- 5. *Ce-Vlan-Preservation & Ce-Vlan-Cos-Preservation*: Left to false by default, these fields allow a choice of CE-VLAN ID & CoS Preservation. This attribute is required when the customer edge is using VLAN header information between its locations.
- 6. Carriers Carrier: Utilized when VPN customers offer VPN services themselves to their customers.

The configuration parameters have been populated to create an L2VPN service "Achme-VPWS1" in the example below.

atom > Services > L2vpn-Svc >	L2vpn-Svc		💒 🙂 🛓
Service Details			1 1
Search for a config Q C	Create Vpn-Service 🔕	<u>[</u> 2,	× 🥝
▶ sites	Entities +	-mandatory information	
vpn-profiles vpn-services	 Vpn-Service - Achme- VPWS1 	Vpn-Id Defines a service identifies	
► ACME	Frame-Delivery	Achme-VPWS1	
	 Vpn-Nodes Vpn-Node - 51 	Vpn-Svc-Type Service type By default, the service type is 'yow'.	
	▼ Vpn-Targets	vpwa × 👻	
	Vpn-Target - 50.1.1.50:1020	Customer-Name	
	 Vpn-Node - 50 	custome name	
	 Vpn-Targets Vpn-Target - 50.1.1.50:1020 	Svc-Topo Defines the service topology, e.g., very-te-enry, "hull-speker.	
		Vpn-Topology Hub-Spoke-Disjoint Hub-Spoke Any-To-Any	
		Description Tental description of a spe-service.	
		Ce-Vlan-Preservation • Promoves the CE-VLAN (3 hom inguess to egress, i.e., the CE-VLAN tag of the egress tr	
		Ce-Vian-Cos-Preservation CE VLAN tog of the set the CE-VLAN tog of the regress feature are id	

ATOM's L2VPN service model has been customized to ensure a smooth provisioning experience. The *vpn-service* list includes *vpn-nodes* list where route distinguisher (RD) and route targets (RT) for each PE device can be defined.

The *vpn-node* captures *rd* values to distinguish routes from each VRF and support type-0, type-1, and type-2 formats. The *vpn-targets* capture the rt values in type-0, type-1, and type-2 forms and supports options of import, export, and both to facilitate route sharing among VRFs. In the screenshot below, the RT values are defined as <IP-address: ASN>

Continuing the L2VPN service "Achme-VPWS1" creation, VPN-Node 50 captures the RD and RT values for the device 172.16.4.99, while VPN-Node 51 captures the RD and RT values for the device 172.16.4.156

atom > Services > L2vpn-Svc > L2	vpn-Svc		💒 🙂 🚊 🖉 🔔
Service Details			1 1
Search for a config Q C 🛓	Create Vpn-Service	ß	× 🕗
 altes vp.profiles vgm.acvides ACME Achme.VPWS1 	 Entitles Vpn-Service - Achmer- VPWS1 Fineme-Belweyt Vpn-Nodes Vpn-Target Vpn-Target Vpn-Target Vpn-Target So.1.1.SO.1020 	• • • • • • • • • • • • • • • • •	

atom > Services > L2vpn-Svc > L2vpn-Svc		≜ ® ≡	∎ U		Admin	
Service Details				\$	<u>+</u>	
	<u>B</u> .		×	G	2	
 • sites yop-profiles • yop-services • AcME • AcME • Achme-VPWS1 • Achme-VPWS1 • Yop-Nedes • Yop-Targets <l< td=""><td>evandatory information Route-Target • Route-Target-Type • Route-Target-Target-Target-Type • Route-Target</td><td></td><td></td><td></td><td></td><td></td></l<>	evandatory information Route-Target • Route-Target-Type • Route-Target-Target-Target-Type • Route-Target					

• atom > Services > L2vpn-Svc	L2vpn-Svc		📌 🔳 🙂 🛓
Service Details			≗ ±
Search for a config Q C	Create Vpn-Service	<u></u> <u></u> .	× 🥝
III + sites vp-pofiles vp-sorties AcAre AchrevPrWs1 *********************************	 Appriservice - Achmeryprysi Appriservice - Achmeryprysi Appriservice - Achmeryprysi Appriservice - Achmeryprysice Appriservice -	• wandwing wandwing P 11.1511020 P 1 P 0	

atom > Services > L2vpn-Svc > L	L2ypn-Svc	📌 🔳 😃 🛓
Service Details		<u>*</u> 2
Search for a config Q C 🛓	Create Vpn-Service 🙆 🕒	× 🥝
 sites ymp-profiles ymm-ternices Achree-VPWS1 	Entities - - - - VpnS Gervice - Achme- VPWS1 Route Target - Incom Target subscription Route Target - Incom Target subscription - Yop-Nodes 50.1.5001020 South Target subscription Vpn-Target subscription Poper Target subscription Route Target subscription Vpn-Target subscription Poper Target subscription Route Target subscription Vpn-Target subscription Poper Target subscription Route Target subscription Vpn-Target subscription Route Target subscription Route Target subscription	

ATOM offers a one-click creation of services. The *vpn-services* list, when applied, ATOM generates the corresponding commands for the devices involved in the service. The device could be one or a combination of legacy CLI based devices, Native YANG, or OpenConfig based to complete an end-to-end service provisioning. In the example below, Juniper MX devices supporting Native YANG are provisioned with VPWS flavor for illustration purposes.

•	tom > Services > L2vpn-Sv	/c → L2v	pn-Svc								📌 🔳 🕛 💄
a	Service Details										\$ ±
٦,	Search for a config Q	c₹	G	/ 1	ô 1	Selected 1				2 Of 2 Search	٩
	▶ sites			Vpn-Id 🛧		Vpn-Svc-Type	Customer-Name	Svc-Topo	Description	Ce-Vlan-Preservation	Ce-Vlan-Cos-Preservation
ıh	vpn-profiles			ACME		vpws		any-to-any		\otimes	\otimes
	▼ vpn-services		 Image: A second s	Achme-VPWS1		vpws		any-to-any		\otimes	\otimes
٥	► ACME										
Θ	 Achme-VPWS1 										
	 cloud-accesses 										
D	 extranet-vpns 										
	frame-delivery										
\$	 ie-profiles 										
	vpn-nodes										
	▼ 50-172.16.4.99										
	maximum-routes										
	status										
	▼ vpn-targets										
	1020										
	51-172.16.4.156										

The below snapshot shows the NETCONF payload generated and provisioned on both 172.16.4.99 & 172.16.4.156.

Create: vpn-service Achme-VPWS1 ×
Task.ID FZn4xh03_kQ5uyFLqAVY888A Time Taken 22/10/2020_111:56:41 - 22/10/2020_111:56:52 (10 seconds)
Logs Commands
Operation JunosConfRoot.configuration Device Name/IP GRE-MIX-599.anutacorp.com / 172.16.4.99
Status Provisioned
Commands <configuration xmins="http://yang.juniper.net/junos/conf/root" xmins.ro="umisetf.parames.rmfns.netconf.base.1.0"> <!--- filler ---></configuration>
<routing-instances xmlns="http://yang.juniper.net/junos/confi/routing-instances"> <1- filler -></routing-instances>
subtree CREATE /controller.devices/devices/2016.4.99/atom-junos-mount-junipermx19481110/junos-conf-root.configuration/junos-conf-routing-instances/not
<instance create*="" ncoperation*=""></instance>
<name>Achme VPWS1</name>
<instance type="">/Zrypn/instance type></instance>
<route distinguisher=""></route>
<rd>type>50.1.1.50.10204/rd type></rd>
<vr></vr> vrFtarget>
<community-target:50.1.1.50:1020- community-<="" td=""></community-target:50.1.1.50:1020->
< - filler →

Operation	JunesConfRootconfiguration
Device Name/IF	2 gre01~MX-4.156 / 172.16.4.156
Status	Provisioned
Commands	<configuration conf="" filer="" junos="" parame.xmlns.netconf.base:1.0*<-="" root*xmlns.net="" th="" xmlnss*http:="" yang.juniper.net="" →<=""></configuration>
	<routing instances="" xmlns="http://yang.juniper.net/junos/conf/routing-instances"><!--</th--></routing>
	- subtree:CREATE /controller:devices/device=172.16.4.156/atom-junos-mount-junipermx194R110/junos-conf-root.configuration/jun</th
	<instance nc:operation="create"></instance>
	<name>Acbme VPWS1r/name></name>
	<instance-typex instance-type="" zvpn<=""></instance-typex>
	<route-distinguisher></route-distinguisher>
	<rd>type>50.1.1.51:1020</rd>
	
	<community>target-50.1.1.50.1020</community>
	< - filler ->

The below screenshot shows the service details on ATOM side.

```
Vpn-Service
 <vpn-services xmlns="urn:ietf:params:xml:ns:yang:ietf_l2vpn_svc">
    <vpn-service>
       <cloud-accesses/>
       <extranet-vpns/>
       <frame-delivery>
          <multicast-gp-port-mapping>dynamic-mapping</multicast-gp-port-mapping>
       </frame-delivery>
       <ie-profiles/>
       <vpn-nodes>
          <vpn-node>
             <vpn-targets>
                <vpn-target>
                   <route-target>50.1.1.50:1020</route-target>
                   <route-target-type>both</route-target-type>
                </vpn-target>
             </vpn-targets>
             <rd>50.1.1.51:1020</rd>
             <vpn-node-id>51</vpn-node-id>
             <device-ip>172.16.4.156</device-ip>
             <node-role>any-to-any-role</node-role>
          </vpn-node>
          <vpn-node>
             <vpn-targets>
                <vpn-target>
                   <route-target>50.1.1.50:1020</route-target>
                   <route-target-type>both</route-target-type>
                </vpn-target>
             </vpn-targets>
             <rd>50.1.1.50:1020</rd>
             <vpn-node-id>50</vpn-node-id>
             <device-ip>172.16.4.99</device-ip>
             <node-role>any-to-any-role</node-role>
             <maximum-routes/>
          </vpn-node>
       </vpn-nodes>
       <carrierscarrier>false</carrierscarrier>
       <ce-vlan-cos-preservation>false</ce-vlan-cos-preservation>
       <ce-vlan-preservation>false</ce-vlan-preservation>
       <svc-topo>any-to-any</svc-topo>
       <vpn-id>Achme-VPWS1</vpn-id>
       <vpn-svc-type>vpws</vpn-svc-type>
    </vpn-service>
 </vpn-services>
```

The below screenshot shows the commands provisioned on both the Juniper MX devices.

• <u>1</u> 172.16.4.99:22 × +
admin@GRE-vMX-5-99>
admin@GRE-vMX-5-99> show configuration display set match Achme
set routing-instances Achme-VPWS1 instance-type l2vpn
<pre>set routing-instances Achme-VPWS1 route-distinguisher 50.1.1.50:1020 set routing-instances Achme-VPWS1 vrf-target target:50.1.1.50:1020</pre>
admin@GRE-vMX-5-99>
• <u>1</u> 172.16.4.99:22 × • <u>2</u> 172.16.4.156:22 × +
admin@gre01-vMX-4.156> show configuration display set match Achme set routing-instances Achme-VPWS1 instance-type l2vpn
set routing-instances Achme-VPWS1 route-distinguisher 50.1.1.51:1020
set routing-instances Achme-VPWS1 vrf-target target:50.1.1.50:1020
admin@gre01-vMX-4.156>

Site Network Access

The *site-network-accesses* is a characteristic under the *site*. It defines the list of ports to the site and its properties. It is also the subcontainer where the physical port or bearer is bound to the *vpn-service*. The site-network-access represents a logical ethernet connection to a site with multiple site-network-accesses, like in multi-homing. The following essential characteristics are grouped under site-network-access.

1. Bearer: Gives a reference back into the site-bearer created above and the PE device port.



- 2. *Connection*: Offers the choice of encapsulation type & ethernet interface type. Options under each category is listed below.
 - i. Encapsulation-Type
 - a. Ethernet
 - b. Vlan
 - ii. Eth-Inf-Type
 - a. Tagged
 - a. qinq
 - b. vxlan
 - c. dot1q
 - d. qinany
 - e. Priority-tagged
 - b. Untagged
 - c. Lag

A tagged interface with qinq with tag type as C-S-Vlan is chosen for Site-1, Device-172.16.4.99, for illustration in the example below.

Search for a config C Search for a config Search for a config C Search for a config C Search for a config <th>📌 🔳 U 🔮</th>	📌 🔳 U 🔮
territoria a control territoria a control territoria a control last bradicast unknown uniclast multicast devices focations make copp prevention make copp prevention make copp prevention multicast territoria a control to be defined service	2
1 Type • access control-list Type • bradcast unknown unicast- multicast Ster-Network Access-Type Punt Ta Pulat • devices ServiceId • • devices ServiceId • • devices ServiceId • • management service Achne VFWSI • site devicesion upps • atteners type in stratute, the service type in types. • atteners type in stratute, the service type in types. • atteners type in stratute, the service type in types. • atteners type in stratute, the service type in types. • atteners type in stratute, the service type in types. • atteners type in stratute, the service type in types. • atteners type in types.	× 🥝
yop policies Network-Access-Id tenter of Annopole access tenter of Annopole access yop profiles yop services Remote Carrier-Name Remote carrier name, The Yende access determines only wh. X * Description	

atom > Services > L2vpn-Svc > L3	2vpn-Svc		💒 🖽 🎍
Service Details			1 <u>+</u>
Search for a config Q C 🛓	Create Site-Network-Access	© (b.	× 🥝
 * sites * sites * 1 * access control·list bradcast unknown uncast- multicast * devices * locations * locations * locations * management service * atterbarres * 2-172.16.4.90 atterbarrety * site retwork accesses 	 Entities • - Site-Network-Access - 20 - 20 Bare Connection Tagged-Interface Qring 	Instructionary and functional accesses types. For example, this could mean no dial in secsess	

atom > Services > L2vpn-Svc > L2vpn-Svc		🔎 🔳 🕛 🔔
Bervice Details		1 ±
Search for a config Q C 🛓 Create Site-Network-A	access 🗿 📴	× 🥝
	Encapsulation type. By default, the encapsulation type is set to sthemer. Encapsulation-Type Ethomet Vlan Ethologi Tume	

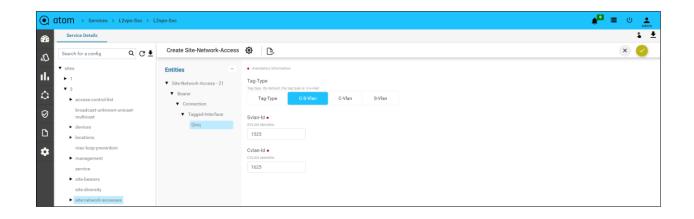
atom > Services > L2vpn-Svc > L2vpn-	1-Svc		≜ [⊡] ≡	ባ	A	idmin
Service Details					\$	Ŧ
	Create Site-Network-Access	(b)		×	•	
v sites ↓ 1 → access-control-list	Entities - • Site-Network-Access - 20 - 20 • Bearer • Connection • Tagged-Interface Qing	wandstory referention Tag Type Tag Type C+S Vitan C+Vian S+Vian SVian-(+ SVAN-(4- SVAN-(4- C+ViaN identifier 1625				

A tagged interface with qinq with tag type as C-S-Vlan is chosen for Site-3, Device-172.16.4.156, for illustration in the example below.

atom > Services > L2vpn-Svc > L2vpn-Svc		💒 🙂 🚊 🙂 🌲
Bervice Details		1 ±
© Search for a config Q C ₹ Create Site-Network-Access	© [<u>}</u> .	× 🧭
 stes i <li< td=""><td> * mandatary stammation Network-AccessId • Tart Tart Tart Remote-Carrier-Name Benote carrier name, "ne wonter carrier name parameter must be configured any with Tart Tart<</td><td></td></li<>	 * mandatary stammation Network-AccessId • Tart Tart Tart Remote-Carrier-Name Benote carrier name, "ne wonter carrier name parameter must be configured any with Tart Tart<	

•	atom > Services > L2vpn-Svc > L2	2vpn-Svc		💒 🖽 🖕
æ	Service Details			<u>*</u> 2
Ð	Search for a config Q C 🛓	Create Site-Network-Access	• E.	× 🥝
	 access-control-list broadcast-unknown-unicast- multicast devices locations 	 Entities • - Site Network Access - 21 Beare Connection Tagged-Interface Qing 	• • • • • • • • • • • • • • • • • • •	

atom > Services > L2vpn-Svc > L2vpn-Svc		•	= (Ŀ	Admin
Barrice Details ↓ Search for a config ↓ Create Site-Network-Access	s 🗿 [<u>B</u> .		×		• ±
 altes access control·list broadcast-unknown unicast- multicast devices 	• mandatory lationation Encapsulation-Type Encapsulation-Type Ethernet Tuberse Eth-In-Type Tagged Untagged Lag				



The below snapshot shows the NETCONF payload generated and provisioned on both 172.16.4.99 & 172.16.4.156.

Create:	e-network-access 20 ×		
Task ID	KKre54TJL#SYU-W560p1KMkQ		
Time Taken	22/10/2020, 18:52:07 - 22/10/2020, 18:52:25 (17 seconds)		
Logs	Commands		
Operation	JunosConfRoot.configuration		
Device Nam	ne/IP_GRE-wMX-5-99.anutacorp.com / 172.16.4.99		
Status	Provisioned		
Commands	<configuration xmlns="http://yang.juniper.net/junos/conf/root" xmlns:nc="umiset/params:xmlns:netconf/base1.0"> <1- filer →</configuration>		
	<interfaces create"="" xmlns="http://yang.juniper.net/junos/conf/interfaces'><{- filer-></td><td></td></tr><tr><td></td><td>< subtree.UPDATE /controller.devices/device=172.16.4.99/atom-junos-mount.junipermx194R110/junos-conf-root.configuration/junos-conf-interfaces.interfaces/interface=xe-0%2F1%2F9-></td><td></td></tr><tr><td></td><td><interface></td><td></td></tr><tr><td></td><td><name>ze0/1/9</name></td><td></td></tr><tr><td></td><td><unt no:operation="></interfaces>		
	<name>1625</name>		
	<encapsulation>vlan.coc</encapsulation>		
	-vlan taga>		
	<outer>1525</outer>		
	<inner>1625</inner>		
	<− filler ->		
	<routing-instances xmlns="http://yang.juniper.net/junos/conf/routing-instances"> < - filler></routing-instances>		
	subtree:UPDATE /controller:devices/device=172.16.4.99/atom-junos mount_junipermx194R110/junos-conf-rout.configuration/junos conf-routing-instances.routi		
	<instance></instance>		
1	<name>AchmeVPWS1</name>		
1	<pre>cprotocola nc:operation="create"></pre>		
	«12-vpn»		

Create:	Create: site-network-access 21			
Task ID	NAC-WritingTHoub1v_2GVDhg			
Time Taken	n 22/10/2020, 18:55:29 - 22/10/2020, 18:55:38 (8 seconds)			
Logs	Commanda			
Operation	JunosConfiguration			
Device Nan	ame/IP gre01+vMX-4.156 / 172.16.4.156			
Status	Provisioned			
Commande	ds cconfiguration xmIns="http://yang.junjper.net/junoa/conf/root" xmIns:ns="umiset/paramaxmins:netconf/base:1,0"> d= filer →			
	cited galaxies and an analysis of the second se			
	subtree: UPDATE / controller.devices/devices172.16.4.156/atom-junos-mount jurgiermx194R110/junos-conf-root configuration/junos-conf-interfaces.interfaces.interfaces/interfa			
	<interface></interface>			
	<name>xe-0/0/4</name>			
	<ur> <unit no:operation="create"></unit> </ur>			
	<name>1025</name>			
	<encapsulation>vlan-ccc</encapsulation>			
	«vlan taga»			
	<outer>1525c/outer></outer>			
	«inner»1625 <td></td>			
	<routing instances="" xmlns="http://yang.juniper.net/junos/conf/routing instances"><!--- filler ---></routing>			
	subtree:UPDATE /controller.devices/device=172.16.4.156/atom-junos-mountjunipermx194R110/junos-conf-rout.configuration/junos-conf-routing-instances.routing-instances/instances/cheme-VPWS1 ->			
	-instance-			
	<name>Achme-VPWS1</name>			
	<pre>constant = constant = consta</pre>			
	-džypn>			

The below screenshot shows the service details on ATOM side.

Site-Network-Access

```
<site-network-accesses xmlns="urn:ietf:params:xml:ns:yang:ietf_l2vpn_svc">
  <site-network-access>
      <access-control-list/>
      <bearer>
        <always-on>true</always-on>
         <connection>
            <tagged-interface>
              <ging>
                  <tag-type>c-s-vlan</tag-type>
                  <svlan-id>1525</svlan-id>
                  <cvlan-id>1625</cvlan-id>
               </ging>
               <type>qinq</type>
            </tagged-interface>
            <encapsulation-type>ethernet</encapsulation-type>
            <eth-inf-type>tagged</eth-inf-type>
         </connection>
         <bearer-reference>2</bearer-reference>
         <port-id>xe-0/1/9</port-id>
      </bearer>
      <device-ip>172.16.4.99</device-ip>
      <network-access-id>20</network-access-id>
      <service-id>Achme-VPWS1</service-id>
      <type>point-to-point</type>
      <vpn-svc-type>vpws</vpn-svc-type>
   </site-network-access>
</site-network-accesses>
```

Site-Network-Access

```
<site-network-accesses xmlns="urn:ietf:params:xml:ns:yang:ietf_l2vpn_svc">
  <site-network-access>
      <access-control-list/>
     <bearer>
        <always-on>true</always-on>
        <connection>
            <tagged-interface>
               <qinq>
                  <tag-type>c-s-vlan</tag-type>
                  <svlan-id>1525</svlan-id>
                  <cvlan-id>1625</cvlan-id>
               </qinq>
               <type>dot1q</type>
            </tagged-interface>
            <encapsulation-type>ethernet</encapsulation-type>
            <eth-inf-type>untagged</eth-inf-type>
         </connection>
        <bearer-reference>3</bearer-reference>
         <port-id>xe-0/0/4</port-id>
      </bearer>
      <device-ip>172.16.4.156</device-ip>
      <network-access-id>21</network-access-id>
      <service-id>Achme-VPWS1</service-id>
      <type>point-to-point</type>
      <vpn-svc-type>vpws</vpn-svc-type>
   </site-network-access>
</site-network-accesses>
```

The below screenshot shows the commands provisioned on both the Juniper MX devices.

• <u>1</u> 172.16.4.99:22 × • <u>2</u> 172.16.4.156:22 × +
admin@GRE-vMX-5-99> show configuration display set match Achme
set routing-instances Achme-VPWS1 protocols l2vpn encapsulation-type ethernet-vlan
set routing-instances Achme-VPWS1 protocols l2vpn no-control-word
set routing-instances Achme-VPWS1 protocols l2vpn site site1 site-identifier 1
set routing-instances Achme-VPWS1 protocols l2vpn site site1 interface xe-0/1/9.1625 remote-site-id 20
set routing-instances Achme-VPWS1 instance-type l2vpn
set routing-instances Achme-VPWS1 interface xe-0/1/9.1625
set routing-instances Achme-VPWS1 route-distinguisher 50.1.1.50:1020
set routing-instances Achme-VPWS1 vrf-target target:50.1.1.50:1020
admin@GRE-vMX-5-99>

• <u>1</u> 172.16.4.99:22 × • <u>2</u> 172.16.4.156:22 × +
admin@GRE-vMX-5-99> show configuration display set match xe-0/1/9
set interfaces xe-0/1/9 flexible-vlan-tagging
set interfaces xe-0/1/9 encapsulation flexible-ethernet-services
set interfaces xe-0/1/9 unit 1144 encapsulation vlan-ccc
set interfaces xe-0/1/9 unit 1144 vlan-id 1144
set interfaces xe-0/1/9 unit 1502 encapsulation vlan-ccc
set interfaces xe-0/1/9 unit 1502 vlan-tags outer 1501
set interfaces xe-0/1/9 unit 1502 vlan-tags inner 1502
set interfaces xe-0/1/9 unit 1625 encapsulation vlan-ccc
set interfaces xe-0/1/9 unit 1625 vlan-tags outer 1525
set interfaces xe-0/1/9 unit 1625 vlan-tags inner 1625
set routing-instances ACME protocols l2vpn site sitel interface xe-0/1/9.1502 remote-site-id 11
set routing-instances ACME interface xe-0/1/9.1502
set routing-instances Achme-VPWS1 protocols l2vpn site sitel interface xe-0/1/9.1625 remote-site-id 20
set routing-instances Achme-VPWS1 interface xe-0/1/9.1625
set routing-instances VPWS-1144 protocols evpn interface xe-0/1/9.1144 vpws-service-id local 1144
set routing-instances VPWS-1144 protocols evpn interface xe-0/1/9.1144 vpws-service-id remote 1144
set routing-instances VPWS-1144 interface xe-0/1/9.1144

• 1172.16.4.99:22 • 2172.16.4.156:22 • + admin@gre01-vMX-4.156> show configuration | display set | match Achme set routing-instances Achme-VPWS1 instance-type l2vpn set routing-instances Achme-VPWS1 interface xe-0/0/4.1625 set routing-instances Achme-VPWS1 route-distinguisher 50.1.1.51:1020 set routing-instances Achme-VPWS1 vrf-target target:50.1.1.50:1020 set routing-instances Achme-VPWS1 protocols l2vpn encapsulation-type ethernet-vlan set routing-instances Achme-VPWS1 protocols l2vpn no-control-word set routing-instances Achme-VPWS1 protocols l2vpn site site-4.156 site-identifier 3 set routing-instances Achme-VPWS1 protocols l2vpn site site-4.156 interface xe-0/0/4.1625 remote-site-id 21 admin@gre01-vMX-4.156>

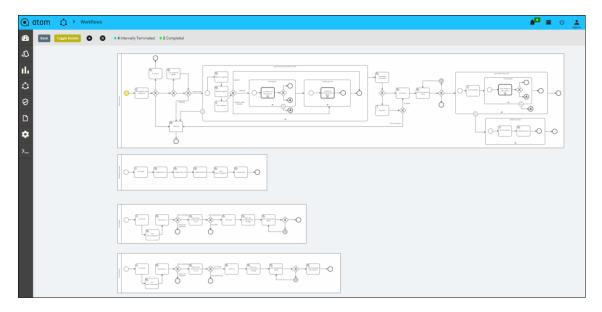
● <u>1</u> 172.164.99:22 × ■ <u>2</u> 172.16.4.156:22 × +
admin@gre01-vMX-4.156> show configuration display set match xe-0/0/4
set interfaces xe-0/0/4 flexible-vlan-tagging
set interfaces xe-0/0/4 encapsulation flexible-ethernet-services
set interfaces xe-0/0/4 unit 457 encapsulation vlan-ccc
set interfaces xe-0/0/4 unit 457 vlan-tags outer 456
set interfaces xe-0/0/4 unit 457 vlan-tags inner 457
set interfaces xe-0/0/4 unit 1625 encapsulation vlan-ccc
set interfaces xe-0/0/4 unit 1625 vlan-tags outer 1525
set interfaces xe-0/0/4 unit 1625 vlan-tags inner 1625
set routing-instances ACME interface xe-0/0/4.457
set routing-instances ACME protocols l2vpn site site-4.156 interface xe-0/0/4.457 remote-site-id 4
set routing-instances Achme-VPWS1 interface xe-0/0/4.1625
set routing-instances Achme-VPWS1 protocols l2vpn site site-4.156 interface xe-0/0/4.1625 remote-site-id 21
admin@gre01-vMX-4.156>

Service Compliance

Anuta ATOM's service modeling capabilities not only allows a smooth vendor agnostic provisioning, but also maintains the state and sanity of the services through its service compliance feature. The service compliance module detects the configuration drift to device configurations, performs service inventory to understand the nature of the changes and notifies upon deviation. ATOM also generates the difference in configuration along with the commands to be reconciled to ensure service compliance. The user is given the authority to overwrite the device or server (ATOM) service model configurations.

Workflow Integration

Every service provisioning is governed by a method-of-procedure in organizations. Integrating ATOM's service models into ATOM workflow engine, helps to execute a set of Pre-Checks, followed by the choice of services to be provisioned and Post-Checks to ensure a smooth and successful service provisioning. Below workflow executes an instance of VPN service with pre & post validations.



Additional Resources

Video-on-demand on ATOM L2VPN Service Automation

To learn how Anuta Network's ATOM Multi-Vendor Service Orchestration can accelerate your network automation journey, contact us at <u>https://www.anutanetworks.com</u>