



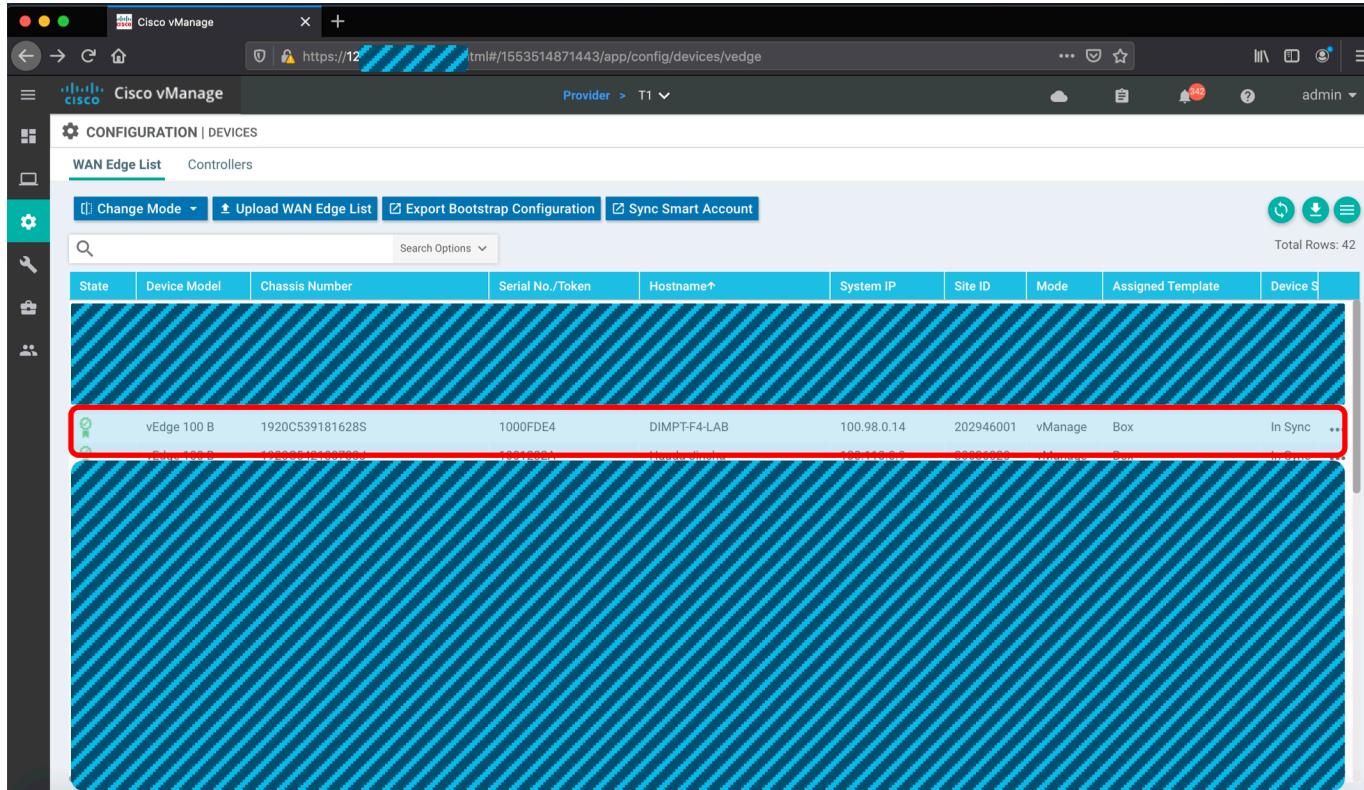
sdwan-tools demo

A Cisco SDWAN tool for quickly config the device both in CLI template and feature template, both single tenant and multi-tenant.

Weibo Rao

November 2019

vManage GUI, the device list



The screenshot shows the Cisco vManage web interface. The title bar reads "Cisco vManage". The URL in the address bar is "https://127.0.0.1:1553514871443/app/config/devices/vedge". The top navigation bar includes "Provider > T1", a user icon for "admin", and a notification bell with 342 notifications. On the left, there's a sidebar with icons for Home, Configuration, Devices, WAN, and Settings. The main content area is titled "CONFIGURATION | DEVICES" and "WAN Edge List". It features a search bar and several buttons: "Change Mode", "Upload WAN Edge List", "Export Bootstrap Configuration", and "Sync Smart Account". Below these are two rows of data. The first row is highlighted with a red border and contains the following information:

State	Device Model	Chassis Number	Serial No./Token	Hostname↑	System IP	Site ID	Mode	Assigned Template	Device S
In Sync	vEdge 100 B	1920C539181628S	1000FDE4	DIMPT-F4-LAB	100.98.0.14	202946001	vManage	Box	In Sync

vManage uses template to config and manage vEdge devices.

This demo will read the data of device 1920C539181628S from vmanage, and save to 1920C539181628S.json

And then make a little change of the json data, then push back to vManage.

The content of CLI template ‘Box’

The screenshot shows the Cisco vManage web interface. In the top navigation bar, it says "Cisco vManage". Below the header, there are tabs for "Provider > T1" and a user dropdown for "admin". On the left, there's a sidebar with icons for Device, Feature, Configuration, and Templates. Under Configuration, "CONFIGURATION | TEMPLATES" is selected. A sub-menu shows "Device Model" set to "vEdge 100 B" and "Template Name" set to "Box". The main content area is titled "CLI Configuration" and contains the following text:

```
36 ցւյալ-ըստ
37 39 առաջ կապահանգար
38 40 առաջ ստատիկ
41 !
42 vpn 0
43   ինտերֆեյս {{WAN_Port_Name}}
44   {{WAN_TYPE_INFO}}
45   {{cellular_interface}}
46 !
47
48 {{VPN1}}
49 {{VPN2}}
50 {{VPN3}}
51 {{VPN4}}
52
53
54
55
56 քուածականություն
57 օպ-վիզուալիտե
58 ֆլու-վիզուալիտե
59 !
60 vpn 512
61   ինտերֆեյս ge0/0
62
```

A red box highlights the line "{{VPN1}}". At the bottom right of the configuration window, there is a "Cancel" button.

Some user defines "huge" vars in the template. For example, the var {{VPN1}} contains all the configurations of VPN1, including interface, ip address, policy and so on.

They may developed their own user portal to config the device. They sell each VPN to the end user, so each VPN means an end user.

However, in some cases, customer only want to change part of configurations, such as from DHCP to static ip, the portal is not so flexible to do this simple change.

As you can see as the next page, user can not input multi-line contents in the GUI form.

The GUI can not input multi-line contents

Update Device Template

Variable List (Hover over each field for more information)

Chassis Number	1920C539181628S
System IP	100.98.0.14
Hostname	DIMPT-F4-LAB
host-name	DIMPT-F4-LAB
system-ip-address	100.98.0.14
siteid	202946001
organization-name	"C [REDACTED] T1"
vbond-address	22 [REDACTED]
controller-name	"C [REDACTED]
admin-name	admin
admin-password	[REDACTED]
cellular	omp no shutdown graceful-restart advertise connected advertise sta
WAN_Port_Name	ge0/4
WAN_TYPE_INFO	shaping-rate 2048ip address 10.43.11.11/24 tunnel-interface encap
cellular_interface	[REDACTED]
VPN1	[REDACTED]

Update **Cancel**

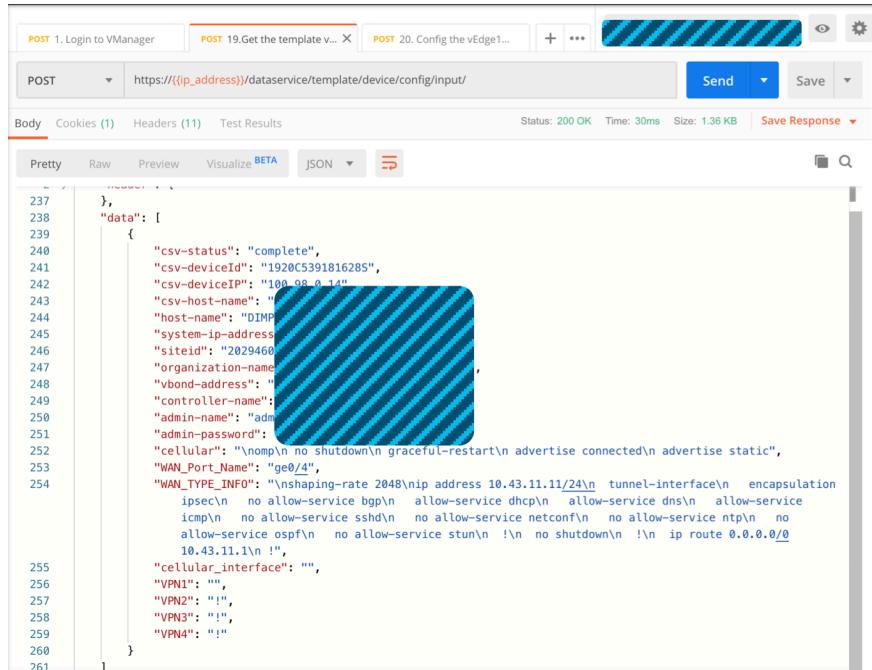
In vManage's GUI, it is hard to input multi-line contents.

When customer make changes in this form, they usually get the error message, such as follow:

Server error: shaping-rate 2048 ip address
10.43.11.11/24 syntax error: element does not
exist Error: on line 57: shaping-rate 2048 ip
address 10.43.11.11/24 tunnel-interface
encapsulation ipsec no allow-service bgp allow-
service dhcp allow-service dns allow-service icmp
no allow-service sshd no allow-service netconf no
allow-service ntp no allow-service ospf no allow-
service stun ! no shutdown ! ip route 0.0.0.0/0
10.43.11.1 !

Uses Postman to do the work

1. Get the data from vManage



POST 1. Login to VManager POST 19.Get the template v... X POST 20. Config the vEdge1...

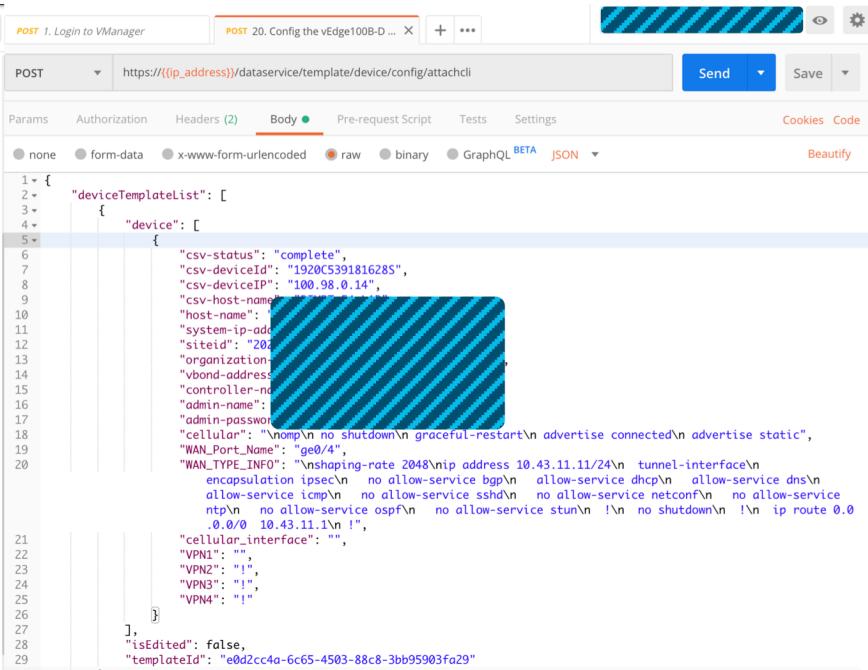
Body Cookies (1) Headers (11) Test Results

Status: 200 OK Time: 30ms Size: 1.36 KB Save Response

Pretty Raw Preview Visualize BETA JSON

```
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
    },
    "data": [
        {
            "csv-status": "complete",
            "csv-deviceId": "1920C5391816285",
            "csv-deviceIP": "100.98.0.14",
            "csv-host-name": "DIMPY-PC-01",
            "host-name": "DIMPY-PC-01",
            "system-ip-address": "100.98.0.14",
            "siteid": "2029460",
            "organization-name": "Cisco Systems Inc.",
            "vbond-address": "10.43.11.1",
            "controller-name": "DIMPY-PC-01",
            "admin-name": "admin",
            "admin-password": "admin",
            "cellular": "no nmp\n no shutdown\n graceful-restart\n advertise connected\n advertise static",
            "WAN_Port_Name": "ge0/4",
            "WAN_TYPE_INFO": "nshaping-rate 2048\nip address 10.43.11.11/24\n tunnel-interface\n encapsulation ipsec\n no allow-service bgp\n allow-service dhcp\n allow-service dns\n allow-service icmp\n no allow-service sshd\n no allow-service netconf\n no allow-service ntp\n no allow-service ospf\n no allow-service stun\n !\n no shutdown\n !\n ip route 0.0.0.0/0\n 10.43.11.1\n !",
            "cellular_interface": "",
            "VPN1": "",
            "VPN2": "!",
            "VPN3": "!",
            "VPN4": "!"
        }
    ]
}
```

2. Edit and then push back to vManage



POST 1. Login to VManager POST 20. Config the vEdge100B-D ... X

POST https://{{ip_address}}/dataservice/template/device/config/attachcli

Params Authorization Headers (2) Body Pre-request Script Tests Settings Cookies Code

Body (JSON)

```
1 {
  "deviceTemplateList": [
    {
      "device": [
        {
          "csv-status": "complete",
          "csv-deviceId": "1920C5391816285",
          "csv-deviceIP": "100.98.0.14",
          "csv-host-name": "DIMPY-PC-01",
          "host-name": "DIMPY-PC-01",
          "system-ip-address": "100.98.0.14",
          "siteid": "2029460",
          "organization-name": "Cisco Systems Inc.",
          "vbond-address": "10.43.11.1",
          "controller-name": "DIMPY-PC-01",
          "admin-name": "admin",
          "admin-password": "admin",
          "cellular": "no nmp\n no shutdown\n graceful-restart\n advertise connected\n advertise static",
          "WAN_Port_Name": "ge0/4",
          "WAN_TYPE_INFO": "nshaping-rate 2048\nip address 10.43.11.11/24\n tunnel-interface\n encapsulation ipsec\n no allow-service bgp\n allow-service dhcp\n allow-service dns\n allow-service icmp\n no allow-service sshd\n no allow-service netconf\n no allow-service ntp\n no allow-service ospf\n no allow-service stun\n !\n no shutdown\n !\n ip route 0.0.0/0\n 10.43.11.1\n !",
          "cellular_interface": "",
          "VPN1": "192.168.1.1",
          "VPN2": "192.168.1.1",
          "VPN3": "192.168.1.1",
          "VPN4": "192.168.1.1"
        ]
      ],
      "isEdited": false,
      "templateId": "e0d2cc4a-6c65-4503-88c8-3bb95903fa29"
    }
  ]
}
```

All jobs are done manually, it is inefficient and prone to errors. It's why this script was created.

Demo: setup the environment

The sdwan_env.py defines the server info in a list.

```
(venv) werao@WERA0-M-40KA Cisco-sdwan-tools % cat sdwan_env.py
# -*- coding: utf-8 -*-

server_list = [
{
    "server_name": "chusdwan",
    "hostname": "192.168.1.100",
    "port": 443,
    "username": "admin",
    "password": "C1sco12345",
    "tenant": [
        {
            "name": "T1"
        },
        {
            "name": "Agg-GW"
        },
        {
            "name": "DIMPT"
        }
    ],
    "single_tenant": false
},
{
    "server_name": "dcloudrtp",
    "hostname": "192.168.1.101",
    "port": 443,
    "username": "amdemo1",
    "password": "C1sco12345",
    "tenant": "single_tenant_mode"
}
]
```

This script supports both single tenant mode and multi tenant mode.

In my demo environment, it is multi-tenant mode.

```
(venv) werao@WERA0-M-40KA Cisco-sdwan-tools % python sdwan_tools.py set env
2019-11-12 21:44:47,688 - DEBUG - Start of program
2019-11-12 21:44:47,689 - DEBUG - Current environment is : server chusdwan tenant T1
Please choose the server you want to connect.
(0) chusdwan
(1) dcloudrtp
Choose your server:0
Please choose the tenant.
(0) T1
(1) Agg-GW
(2) DIMPT
Choose your tenant:0
Set env to chusdwan
Hostname: chusdwan
(venv) werao@WERA0-M-40KA Cisco-sdwan-tools % cat current_env.json
{"server_name": "chusdwan", "hostname": "192.168.1.100", "port": 443, "username": "admin", "password": "C1sco12345", "tenant": [{"name": "T1"}]}
(venv) werao@WERA0-M-40KA Cisco-sdwan-tools %
```

Demo: get the device config json data

Run python sdwan_tools get 1920C539181628S, get the debug info as follows.
The output file is 1920C539181628S.json

```
(venv) werao@WERO-M-40KA Cisco-sdwan-tools % python sdwan_tools.py get 1920C539181628S
2019-11-11 21:54:32,908 - DEBUG - Start of program
2019-11-12 21:54:32,909 - DEBUG - Current environment is : server tenant T1
2019-11-12 21:54:32,919 - DEBUG - Starting new HTTPS connection (1): 12.0.0.1:443
2019-11-12 21:54:32,996 - DEBUG - https://12.0.0.1:443 "POST /j_security_check HTTP/1.1" 200 0
2019-11-12 21:54:33,025 - DEBUG - https://12.0.0.1:443 "GET /dataservice/client/token HTTP/1.1" 404 0
2019-11-12 21:54:33,026 - DEBUG - Start of get_request(https://12.0.0.1:443/dataservice/tenant {'Content-Type': 'application/json'})
2019-11-12 21:54:33,048 - DEBUG - https://12.0.0.1:443 "GET /dataservice/tenant HTTP/1.1" 200 None
2019-11-12 21:54:33,051 - DEBUG - Start of get_request(https://12.0.0.1:443/dataservice/tenant {'Content-Type': 'application/json'}
{"header": {"generatedOn": "1573566867916", "viewKeys": {"uniqueKey": ["tenant-id"], "preferenceKey": "grid-Tenant"}, "columns": [{"title": "Sub Domain", "property": "subDomain", "width": 110, "dataType": "string"}, {"title": "vBond Address", "property": "vBondAddress", "width": 110, "dataType": "string"}, {"title": "Organization Name", "property": "orgName", "width": 110, "dataType": "string"}, {"title": "Description", "property": "desc", "width": 110, "dataType": "string"}], "fields": [{"property": "tenantId", "dataType": "string"}, {"property": "subDomain", "dataType": "string"}, {"property": "orgName", "dataType": "string"}, {"property": "desc", "dataType": "string"}]}, {"data": [{"orgName": "sam1SpInfo": "", "vBondAddress": "1.1.1.1", "subDomain": "", "oldIdpMetadata": "", "configDBClusterServiceName": "", "mode": "off", "idpMetadata": "", "createdAt": "1553514659696", "tenantId": "1553514659696", "spMetadata": "", "state": "READY", "desc": "Aggregation Gateway", "tenantName": "Agg-GW", "vBndlAddress": "1.1.1.1", "vBndlPort": "443", "spMeta": "", "spMetaState": "READY", "spDesc": "Tenant DIMP1", "spOrgName": "sam1SpInfo", "vBndlAddress": "1.1.1.1", "subDomain": "", "oldIdpMetadata": "", "configDBClusterServiceName": "1553514871443", "mode": "off", "idpMetadata": "", "createdAt": "1553514871450", "tenantId": "1553514871443", "spMetadata": "", "state": "READY", "desc": "Tenant DIMP2", "tenantName": "DIMP2", "vBndlAddress": "1.1.1.1", "vBndlPort": "443", "spMeta": "", "spMetaState": "READY", "spDesc": "Tenant DIMP1"}, {"data": [{"orgName": "sam1SpInfo": "", "vBondAddress": "1.1.1.1", "subDomain": "", "oldIdpMetadata": "", "configDBClusterServiceName": "1553514871443", "mode": "off", "idpMetadata": "", "createdAt": "1553514871450", "tenantId": "1553514871443", "spMetadata": "", "state": "READY", "desc": "Aggregation Gateway", "tenantName": "Agg-GW", "vBndlAddress": "1.1.1.1", "vBndlPort": "443", "spMeta": "", "spMetaState": "READY", "spDesc": "Tenant DIMP2", "spOrgName": "sam1SpInfo", "vBndlAddress": "1.1.1.1", "subDomain": "", "oldIdpMetadata": "", "configDBClusterServiceName": "1553514871443", "mode": "off", "idpMetadata": "", "createdAt": "1553514871450", "tenantId": "1553514871443", "spMetadata": "", "state": "READY", "desc": "Tenant DIMP1"}]}]
2019-11-12 21:54:33,051 - DEBUG - Start of post_request(https://12.0.0.1:443/vsessionid {'Content-Type': 'application/json'}
()}
```

Demo: make some changes to the json file.

The original file

```
1920C539181628S.json
```

```
1 {
2     "csv-status": "complete",
3     "csv-deviceId": "1920C539181628S",
4     "csv-deviceIP": "100.98.0.14",
5     "csv-host-name": "DIMPT-F4-LAB",
6     "host-name": "DRAFT-F4-LAB",
7     "system-ip-address": "100.98.0.14",
8     "siteid": "202946001",
9     "organization-name": "",
10    "vbond-address": "2",
11    "controller-name": "",
12    "admin-name": "admin",
13    "admin-password": "",
14    "cellular": "\nnoip\nno shutdown\ngraceful-restart\nadvertise connected\nadvertise static",
15    "WAN_Port_Name": "ge0/4",
16    "WAN_TYPE_INFO": "\nshaping-rate 2048\nip address 10.43.11.11/24\n tunnel-interface\n encapsulation ipsec\n no allow-service bgp\n allow-service dhcp\n",
17    "cellular_interface": "",
18    "VPN1": "",  
19    "VPN2": "",  
20    "VPN3": "",  
21    "VPN4": ""
22 }
```

The changed file

```
1920C539181628S.json
```

```
1 {
2     "csv-status": "complete",
3     "csv-deviceId": "1920C539181628S",
4     "csv-deviceIP": "100.98.0.14",
5     "csv-host-name": "DIMPT-LAB-F4-TEST",
6     "host-name": "DIMPT-LAB-F4-TEST",
7     "system-ip-address": "100.98.0.14",
8     "siteid": "202946001",
9     "organization-name": "",
10    "vbond-address": "",
11    "controller-name": "",
12    "admin-name": "ad",
13    "admin-password": "",
14    "cellular": "\nnoip\nno shutdown\ngraceful-restart\nadvertise connected\nadvertise static",
15    "WAN_Port_Name": "ge0/4",
16    "WAN_TYPE_INFO": "\nshaping-rate 2048\nip address 10.43.11.11/24\n tunnel-interface\n encapsulation ipsec\n no allow-service bgp\n allow-service dhcp\n",
17    "cellular_interface": "",
18    "VPN1": "\nvpn 100\ninterface ge0/2\n ip address 100.77.1.1/28\n no shutdown\n !\n",
19    "VPN2": "",  
20    "VPN3": "",  
21    "VPN4": ""
22 }
```

Demo: push the data back

Run python sdwan_tools push 1920C539181628S, get the debug info as follows. (You can turn off debug, by uncomment # logging.disable(logging.CRITICAL)

The job is successfully done.

```
(venv) werao@WERA0-M-40KA Cisco-sdwan-tools % python sdwan_tools.py get 1920C539181628S
2019-11-12 21:54:32,908 - DEBUG - Start of program
2019-11-12 21:54:32,909 - DEBUG - Current environment is : server tenant T1
2019-11-12 21:54:32,919 - DEBUG - Starting new HTTPS connection (1): 120.52.12.13:443
2019-11-12 21:54:32,996 - DEBUG - https://120.52.12.13:443 "POST /j_security_check HTTP/1.1" 200 0
2019-11-12 21:54:33,025 - DEBUG - https://120.52.12.13:443 "GET /dataservice/client/token HTTP/1.1" 404 0
2019-11-12 21:54:33,026 - DEBUG - Start of get_request(https://120.52.12.13:443/dataservice/tenant {'Content-Type': 'application/json'})
2019-11-12 21:54:33,048 - DEBUG - https://120.52.12.13:443 "GET /dataservice/tenant HTTP/1.1" 200 None
2019-11-12 21:54:33,051 - DEBUG - Start of get_request(https://120.52.12.13:443/dataservice/tenant {'Content-Type': 'application/json'})
{"header": {"generatedOn": "1573566867916", "viewKeys": {"uniqueKey": ["tenant-id"], "preferenceKey": "grid-Tenant"}, "columns": [{"title": "Sub Domain", "property": "subDomain", "width": 10, "dataType": "string"}, {"title": "Vbond Address", "property": "vbondAddress", "width": 110, "dataType": "string"}, {"title": "Organiization Name", "property": "orgName", "width": 110, "dataType": "string"}, {"title": "Description", "property": "desc", "width": 110, "dataType": "string"}], "fields": [{"property": "tenantId", "dataType": "string"}, {"property": "subDomain", "dataType": "string"}, {"property": "orgName", "dataType": "string"}, {"property": "desc", "dataType": "string"}]}, "data": [{"orgName": "sampleSpInfo", "subDomain": "1.1.1.1", "oldIdpMetadata": "", "configDBCclusterServiceName": "", "mode": "off", "idpMetadata": "", "createdAt": "1553514659617", "@rid": 39, "name": "Agg-GW", "tenantId": "1553514659609", "spMetadata": "", "state": "READY", "desc": "Adgregation Gateway", "orgName": "sampleSpInfo", "vbondAddress": "1.1.1.1", "subDomain": "1.1.1.1", "oldIdpMetadata": "", "configDBCclusterServiceName": "", "mode": "off", "idpMetadata": "", "createdAt": "1553514871450", "@rid": 284, "name": "I1", "tenantId": "1553514871443", "spMetadata": "", "state": "READY", "desc": "Tenant I", "orgName": "sampleSpInfo", "vbondAddress": "1.1.1.1", "subDomain": "1.1.1.1", "oldIdpMetadata": "", "configDBCclusterServiceName": "", "mode": "off", "idpMetadata": "", "createdAt": "1559543230932", "@rid": 142575, "name": "DIMP-T", "tenantId": "1559543230903", "spMetadata": "", "state": "READY", "desc": "Tenant DIMP-T"}]}
2019-11-12 21:54:33,051 - DEBUG - Start of post_request(https://120.52.12.13:443/dataservice/tenant/1553514871443/vssessionid {'Content-Type': 'application/json'}
()}
```

Demo: The job status

The job logs are the same as the script's output.

Demo: The device hostname was changed

The screenshot shows the Cisco vManage web interface. The top navigation bar includes the Cisco logo, 'Cisco vManage', and a 'Provider > T1' dropdown. Below the header is a toolbar with icons for cloud, file, help, and user status (350 notifications). The main area is titled 'CONFIGURATION | DEVICES' with a 'WAN Edge List' tab selected. The table displays various device details: State, Device Model, Chassis Number, Serial No./Token, Hostname, System IP, Site ID, Mode, Assigned Template, and Device S. A search bar and options for 'Change Mode', 'Upload WAN Edge List', 'Export Bootstrap Configuration', and 'Sync Smart Account' are available at the top of the table. The table shows 42 total rows. The last row in the table, representing a vEdge 100 B device, has its 'Hostname' field highlighted with a red box and contains the value 'DIMPT-LAB-F4-TEST'.

State	Device Model	Chassis Number	Serial No./Token	Hostname↑	System IP	Site ID	Mode	Assigned Template	Device S
Up	vEdge 100 B	1920C539181628S	1000FDE4	DIMPT-LAB-F4-TEST	100.98.0.14	202946001	vManage	Box	In Sync

Demo: vpn 100 was configured

The screenshot shows the Cisco vManage interface with the title "Cisco vManage" at the top left. The top navigation bar includes "Provider > T1" and a user icon for "admin". On the left, there's a sidebar with icons for Home, Configuration, Devices, Change Mode, and a search bar. The main area is titled "CONFIGURATION | DEVICES" and "WAN Edge List". A modal window titled "Running Configuration" is open, displaying the configuration for a specific device. The configuration text is as follows:

```
vpn u
interface ge0/4
ip address 10.43.11.11/24
tunnel-interface
encapsulation ipsec
no allow-service bgp
allow-service dhcp
allow-service dns
allow-service icmp
no allow-service sshd
no allow-service netconf
no allow-service ntp
no allow-service ospf
no allow-service stun
allow-service https
!
no shutdown
shaping-rate 2048
!
ip route 0.0.0.0/0 10.43.11.1
!
vpn 100
interface ge0/2
ip address 100.77.1.1/28
no shutdown
!
```

A red box highlights the "vpn 100" configuration block. In the bottom right corner of the modal, there is a "Close" button. To the right of the modal, a table titled "Sync Status" is visible, showing "Total Rows: 42" and several rows of sync status information.

Demo: Config diff tool on vManage audit log

Config Diff

Inline Diff Side by Side

Old Configuration		New Configuration	
1	viptela-system:system	1	viptela-system:system
2	personality	2	personality
3	device-model	3	device-model
4	chassis-number	4	chassis-number
5	host-name	5	host-name
6	system-ip	6	system-ip
7	vmanage-system-ip	7	vmanage-system-ip
8	site-id	8	site-id
9	admin-tech-on-failure	9	admin-tech-on-failure
10	no route-consistency-check	10	no route-consistency-check
11		11	
12		12	
13		13	
14		14	
15		15	

Changes:

- Line 5: host-name DIMPT-F4-LAB → DIMPT-LAB-F4-TEST
- Line 7: vmanage-system-ip 169.254.10.51 → 169.254.10.51
- Line 10: no route-consistency-check → no route-consistency-check

Redacted Lines:

- Line 11: Redacted
- Line 12: Redacted
- Line 13: Redacted
- Line 14: Redacted

Buttons:

- Close

Config Diff

61	allow-service https	61	allow-service https
62	!	62	!
63	no shutdown	63	no shutdown
64	shaping-rate 2048	64	shaping-rate 2048
65	!	65	!
66	ip route 0.0.0.0/0 10.43.11.1	66	ip route 0.0.0.0/0 10.43.11.1
67	!	67	!
68	vpn 100	68	vpn 100
69	interface ge0/2	69	interface ge0/2
70	ip address 100.77.1.1/28	70	ip address 100.77.1.1/28
71	no shutdown	71	no shutdown
72	!	72	!
73	!	73	!
68	vpn 512	74	vpn 512
69	interface ge0/0	75	interface ge0/0
70	ip address 192.168.254.1/24	76	ip address 192.168.254.1/24
71	no shutdown	77	no shutdown
72	!	78	!
73	!	79	!

Buttons:

- Close

Multi-line text convert

```
multilin_str = """vpn 0
dns 114.114.114.114 primary
interface ge0/4
ip dhcp-client
pppoe-client ppp-interface ppp1
no shutdown
!
interface ppp1
ppp authentication chap
hostname 86152074
password cxxxxx
!
tunnel-interface
encapsulation ipsec
color biz-internet restrict
no allow-service bgp
allow-service dhcp
allow-service dns
allow-service icmp
no allow-service sshd
no allow-service netconf
no allow-service ntp
no allow-service ospf
no allow-service stun
allow-service https
!
mtu    1492
no shutdown
!
"""
str_list = multilin_str.splitlines()
for string in str_list:
    print(string, end="\n")
```

This script helps to convert multi-line string to the data format of vManage json data.

```
"vpn 0\n dns 114.114.114.114 primary\n interface ge0/4\n ip
dhcp-client\n pppoe-client ppp-interface ppp1\n no
shutdown\n !\n interface ppp1\n ppp authentication chap\n
hostname 86152074\n password cxxxxx\n !\n tunnel-
interface\n encapsulation ipsec\n color biz-internet restrict\n
no allow-service bgp\n allow-service dhcp\n allow-service
dns\n allow-service icmp\n no allow-service sshd\n no allow-
service netconf\n no allow-service ntp\n no allow-service
ospf\n no allow-service stun\n allow-service https\n !\n mtu
1492\n no shutdown\n !\n!"
```

