Netmiko

In [1]:
```
!uname -n ; id
```

```
af40b2462788
uid=1000(jovyan) gid=100(users) groups=100(users)
```

In [2]:
```
# !pip install netmiko
```

VM環境情報

ベンダーサポート

cumulusはないのでlinuxを指定する

---

a10 accedian adtran_os alcatel_aos alcatel_sros allied_telesis_awplus apresia_aeos arista_eos aruba_os aruba_osswitch aruba_procurve avaya_ers avaya_vsp broadcom_icos brocade_fastiron brocade_fos brocade_netiron brocade_nos brocade_vdx brocade_vyos calix_b6 cdot_cros centec_os checkpoint_gaia ciena_saos cisco_asa cisco_ftd cisco_ios cisco_nxos cisco_s300 cisco_tp cisco_viptela cisco_wlc cisco_xe cisco_xr cloudgenix_ion coriant dell_dnos9 dell_force10 dell_isilon dell_os10 dell_os6 dell_os9 dell_powerconnect dell_sonic dlink_ds eltex eltex_esr endace enterasys ericsson_ipos extreme extreme_ers extreme_exos extreme_netiron extreme_nos extreme_slx extreme_tierra extreme_vdx extreme_vsp extreme_wing f5_linux f5_ltm f5_tmsh flexvnf fortinet generic generic_termserver hp_comware hp_procurve huawei huawei_olt huawei_smartax huawei_vrpv8 ipinfusion_ocnos juniper juniper_junos juniper_screenos keymile keymile_nos linux mellanox mellanox_mlnxos mikrotik_routeros mikrotik_switchos mrv_lx mrv_optiswitch netapp_cdot netgear_prosafe netscaler nokia_sros oneaccess_oneos ovs_linux paloalto_panos pluribus quanta_mesh rad_etx raisecom_roap ruckus_fastiron ruijie_os sixwind_os sophos_sfos supermicro_smis tplink_jetstream ubiquiti_edge ubiquiti_edgerouter ubiquiti_edgeswitch ubiquiti_unifiswitch vyatta_vyos vyos watchguard_fireware yamaha zte_zxros zyxel_os

---

In [3]:
```
import netmiko
import requests
import json
import csv
import pandas as pd
```

In [6]:
```
device = {
    'device_type':'linux',
    'ip':'172.17.0.2',
    'username':'root',
    'password':'cumupasswd',
    'secret':'cumpupasswd',
}
```

SSHを張ってコマンド投入

In [7]:
```
with netmiko.ConnectHandler(**device) as con:
    con.enable()
    output_text = con.send_command("net show system")
    print(output_text)
```

```
Hostname..... leaf01
Build..... Cumulus Linux 4.3.0
Uptime..... 1:08:00.530000

Model..... Cumulus VX
Memory..... 4GB
Vendor Name..... Cumulus Networks
Part Number..... 4.3.0
Base MAC Address. 02:42:AC:11:00:02
Serial Number.... 02:42:AC:11:00:02
Product Name..... Containerised VX
```

- 整形しやすいようにjsonで取得する（cumulusにはjson出力コマンドがある）

In [8]:
```
with netmiko.ConnectHandler(**device) as con:
    con.enable()
    output = con.send_command("net show system json")
    print(output)
```

{"memory": 4029192, "uptime": "1:08:16", "hostname": "leaf01", "platform": {"detected": true, "vendor": "Cumulus", "model": "VX"}, "build": "Cumulus Linux 4.3.0", "os-version": "4.3.0", "disk": null, "lsb-release": "DISTRIB_ID=\"Cumulus Linux\"\nDISTRIB_RELEASE=4.3.0\nDISTRIB_DESCRIPTION=\"Cumulus Linux 4.3.0\"\n", "os-description": "Cumulus Linux 4.3.0", "eeprom": {"idString": "TlvInfo", "totalLength": 69, "version": 1, "tlv": {"Part Number": {"index": 36, "length": 5, "code": "0x22", "value": "4.3.0"}, "Serial Number": {"index": 55, "length": 17, "code": "0x23", "value": "02:42:AC:11:00:02"}, "MAC Addresses": {"index": 43, "length": 2, "code": "0x2A", "value": "1"}, "CRC-32": {"index": 74, "length": 4, "code": "0xFE", "value": "0xBC1F0CA0"}, "Product Name": {"index": 29, "length": 2, "code": "0x21", "value": "Containerised VX"}, "Device Version": {"index": 33, "length": 1, "code": "0x26", "value": "3"}, "Base MAC Address": {"index": 47, "length": 6, "code": "0x24", "value": "02:42:AC:11:00:02"}, "Vendor Name": {"index": 11, "length": 16, "code": "0x2D", "value": "Cumulus Networks"}}, "validTlvInfoHeader": "true"}}

- APIの取得方法も確認

NVIDIAのcumulus linux APIリファレンス

https://docs.nvidia.com/networking-ethernet-software/cumulus-linux-42/System-Configuration/HTTP-API/

In [28]:
```
!curl -X POST -k -u root:cumupasswd -H "Content-Type: application/json" -d '{"cmd": "show system json"}' https://172.17.0.2:8080/nclu/v1/rpc
```

{"memory": 4029192, "uptime": "1:33:18.270000", "hostname": "leaf01", "platform": {"detected": true, "vendor": "Cumulus", "model": "VX"}, "build": "Cumulus Linux 4.3.0", "os-version": "4.3.0", "disk": null, "lsb-release": "DISTRIB_ID=\"Cumulus Linux\"\nDISTRIB_RELEASE=4.3.0\nDISTRIB_DESCRIPTION=\"Cumulus Linux 4.3.0\"\n", "os-description": "Cumulus Linux 4.3.0", "eeprom": {"idString": "TlvInfo", "totalLength": 69, "version": 1, "tlv": {"Part Number": {"index": 36, "length": 5, "code": "0x22", "value": "4.3.0"}, "Serial Number": {"index": 55, "length": 17, "code": "0x23", "value": "02:42:AC:11:00:02"}, "MAC Addresses": {"index": 43, "length": 2, "code": "0x2A", "value": "1"}, "CRC-32": {"index": 74, "length": 4, "code": "0xFE", "value": "0xBC1F0CA0"}, "Product Name": {"index": 29, "length": 2, "code": "0x21", "value": "Containerised VX"}, "Device Version": {"index": 33, "length": 1, "code": "0x26", "value": "3"}, "Base MAC Address": {"index": 47, "length": 6, "code": "0x24", "value": "02:42:AC:11:00:02"}, "Vendor Name": {"index": 11, "length": 16, "code": "0x2D", "value": "Cumulus Networks"}}, "validTlvInfoHeader": "true"}}

取得した情報を整形する

In [9]:
```
output = json.loads(output)
output
```

Out[9]:

```
{'memory': 4029192,
 'uptime': '1:08:16',
 'hostname': 'leaf01',
 'platform': {'detected': True, 'vendor': 'Cumulus', 'model': 'VX'},
 'build': 'Cumulus Linux 4.3.0',
 'os-version': '4.3.0',
 'disk': None,
 'lsb-release': 'DISTRIB_ID="Cumulus Linux"\nDISTRIB_RELEASE=4.3.0\nDISTRIB_DESCRIPTION="Cumulus Linux 4.3.0"\n',
 'os-description': 'Cumulus Linux 4.3.0',
 'eeprom': {'idString': 'TlvInfo',
  'totalLength': 69,
  'version': 1,
  'tlv': {'Part Number': {'index': 36,
    'length': 5,
    'code': '0x22',
    'value': '4.3.0'},
   'Serial Number': {'index': 55,
    'length': 17,
    'code': '0x23',
    'value': '02:42:AC:11:00:02'},
   'MAC Addresses': {'index': 43, 'length': 2, 'code': '0x2A', 'value': '1'},
   'CRC-32': {'index': 74, 'length': 4, 'code': '0xFE', 'value': '0xBC1F0CA0'},
   'Product Name': {'index': 29,
    'length': 2,
    'code': '0x21',
    'value': 'Containerised VX'},
   'Device Version': {'index': 33, 'length': 1, 'code': '0x26', 'value': '3'},
   'Base MAC Address': {'index': 47,
    'length': 6,
    'code': '0x24',
    'value': '02:42:AC:11:00:02'},
   'Vendor Name': {'index': 11,
    'length': 16,
    'code': '0x2D',
    'value': 'Cumulus Networks'}},
  'validTlvInfoHeader': 'true'}}
```

取得したい項目一覧

In [10]:
```
# Hostname..... leaf01
# Model..... Cumulus VX
# Memory..... 4GB
# Vendor Name..... Cumulus Networks
# Serial Number.... 02:42:AC:11:00:02
# Product Name..... Containerised VX
```

In [11]:
```
df = {}
df['vender'] = output['eeprom']['tlv']['Vendor Name']['value']
df['product'] = output['eeprom']['tlv']['Product Name']['value']
df['hostname'] = output['hostname']
df['memory'] = output['memory']
df['serial'] = output['eeprom']['tlv']['Serial Number']['value']
df['os-version'] = output['os-version']
```

データフレームに格納

In [24]:
```
exp_df = pd.DataFrame(list(df.items()), columns=['System info', 'Value'])
exp_df
```

Out[24]:

	System info	Value
0	vender	Cumulus Networks
1	product	Containerised VX
2	hostname	leaf01
3	memory	4029192
4	serial	02:42:AC:11:00:02
5	os-version	4.3.0

Excelに出力

In [25]:
```
exp_df.to_excel('./system_info.xlsx', index=False)
```

In [26]:
```
# CSVにも出力
exp_df.to_csv('./system_info.csv', index=False)
```

In [ ]:
```

```

その他メモ

ターミナル操作

input関数で入力操作を行う。

無限ループでコマンド操作を行い input に exit が入るとループから抜ける処理

In [19]:
```
with netmiko.ConnectHandler(**device) as con:
    con.enable()
    while True:
        prompt = con.find_prompt()
        command = input(prompt)
        if(command == ''):
            continue
        elif (command == 'exit'):
            break
        else:
            output = con.send_command(command)
            print(output)
```

```
Hostname..... leaf01
Build..... Cumulus Linux 4.3.0
Uptime..... 1:13:47.040000

Model..... Cumulus VX
Memory..... 4GB
Vendor Name..... Cumulus Networks
Part Number..... 4.3.0
Base MAC Address. 02:42:AC:11:00:02
Serial Number.... 02:42:AC:11:00:02
Product Name..... Containerised VX
```

```
--- /etc/hostname	2022-04-11 02:19:33.128254000 +0000
+++ /run/nclu/netmisc/etc_hostname	2022-04-11 02:21:35.614706000 +0000
@@ -1 +1 @@
-yabuki-cum
+yabuki-cumulus



net add/del commands since the last "net commit"
================================================

User  Timestamp                  Command
----  -------------------------  ------------------------------
root  2022-04-11 02:21:35.625976  net add hostname yabuki-cumulus

--- /etc/hostname	2022-04-11 02:19:33.128254000 +0000
+++ /run/nclu/netmisc/etc_hostname	2022-04-11 02:21:35.614706000 +0000
@@ -1 +1 @@
-yabuki-cum
+yabuki-cumulus



net add/del commands since the last "net commit"
================================================

User  Timestamp                  Command
----  -------------------------  ------------------------------
root  2022-04-11 02:21:35.625976  net add hostname yabuki-cumulus

Hostname..... yabuki-cumulus
Build........... Cumulus Linux 4.3.0
Uptime.......... 1:14:35.930000

Model........... Cumulus VX
Memory.......... 4GB
Vendor Name..... Cumulus Networks
Part Number..... 4.3.0
Base MAC Address. 02:42:AC:11:00:02
Serial Number.... 02:42:AC:11:00:02
Product Name..... Containerised VX
```

In [ ]:

マルチデバイスに対して同一処理を行う

サンプルをめも

In [29]:
```
### 'devices.txt'から複数機器にSSHする
# from netmiko import ConnectHandler
#
# with open('devices.txt') as routers:
#     for IP in routers:
#         Router = {
#             'device_type': 'cisco_ios',
#             'ip': IP,
#             'username': 'roger',
#             'password': 'cisco'
#         }
#
#         net_connect = ConnectHandler(**Router)
#
#         print ('Connecting to ' + IP)
#         print('-'*79)
#         output = net_connect.send_command('sh ip int brief')
#         print(output)
#         print()
#         print('-'*79)
#
### セッションを閉じる
# net_connect.disconnect()
```

In [30]:
```
### 'devices.txt'から複数機器にSSHする
# from netmiko import ConnectHandler
#
# with open('devices.txt') as routers:
#     for IP in routers:
#         Router = {
#             'device_type': 'cisco_ios',
#             'ip': IP,
#             'username': 'roger',
#             'password': 'cisco'
#         }
#
#         net_connect = ConnectHandler(**Router)
#
#         hostname = net_connect.send_command('show run | i host')
#         hostname.split(" ")
#         hostname,device = hostname.split(" ")
#         print ("Backing up " + device)
#
#         filename = '/home/roger/python-scripts-for-network-engineers/backups/' + device + '.txt'
#         # to save backup to same folder as script use below line and comment out above line
#         # filename = device + '.txt'
#
#         showrun = net_connect.send_command('show run')
#         showvlan = net_connect.send_command('show vlan')
#         showver = net_connect.send_command('show ver')
#         log_file = open(filename, "a")    # in append mode
#         log_file.write(showrun)
#         log_file.write("\n")
#         log_file.write(showvlan)
#         log_file.write("\n")
#         log_file.write(showver)
#         log_file.write("\n")
#
### セッションを閉じる
# net_connect.disconnect()
```

In [ ]:

In [ ]: