

# Janog44 Hackathon

## 自動棚卸しチーム



ビッグローブ株式会社

滝口敏行

前野洋史

及川優星

船山歩

君塚遼

株式会社ZTV

常喜俊憲

# 背景

ネットワーク機器の資産の棚卸しにて(BIGLOBE)

- エクセル管理、手で数えるのが怠い・・・
- SFPの数をいちいち数えるの面倒臭い。。。
- 利用中の機器、モジュール、トランシーバーの総数がわからない・・・
- 半年とか、年に数回必ずやってくる苦行・・・
- 積み重なる資産シールの山。。。
- このためだけに現地のDCへ。。。

ハッカソンに参加してみたい(ZTV)

- ”棚卸”に共感を覚え、”自動化”に惹かれてJOIN



# 目的

自動で資産の棚卸しをしたい！！

- ネットワーク機器ならリモートから情報が収集できるはず
- SNMPだけでは収集できる情報に限りがあるが、SHOWコマンドもできるようにすれば、色々情報取れるはず！



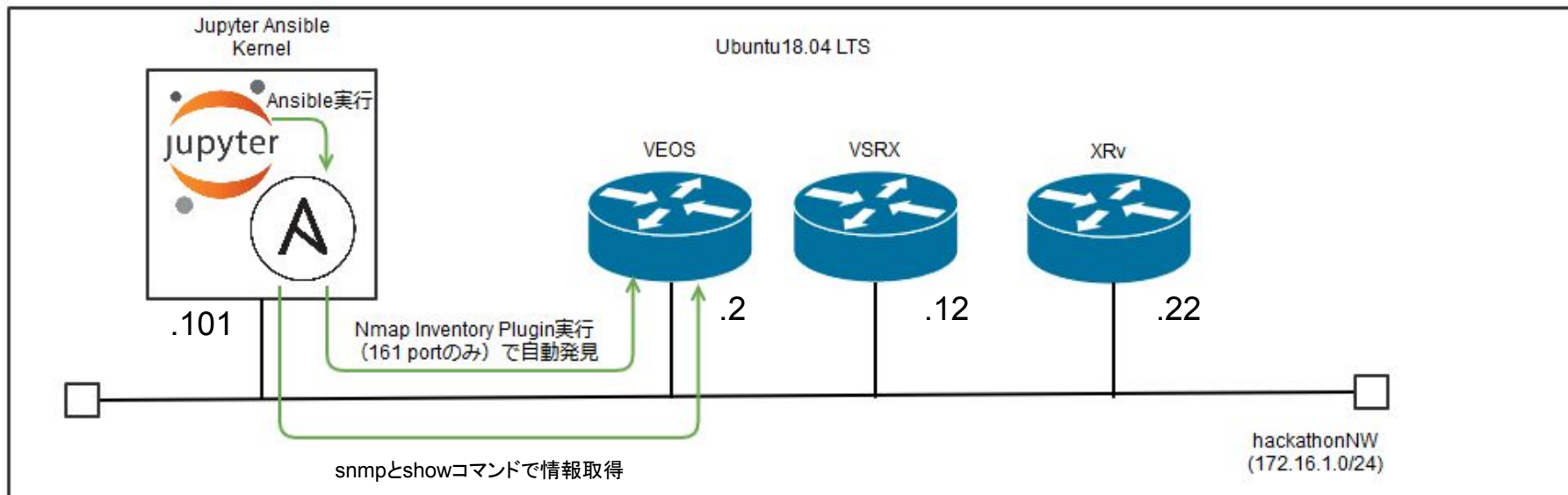
# 作ってみた

## 実装してみた機能

- 機器自動発見
- SNMPおよびSHOWコマンドを自動収集
- 実行結果をパース
- パースした情報をもとに集計



# 会場のdemo構成



## addressでnmapしたいアドレスレンジを指定する

```
In [2]: #inventory
plugin: nmap
strict: True
address: 172.16.1.0/27
exclude:
- 172.16.1.1
# - 172.16.1.2
- 172.16.1.3
- 172.16.1.4
- 172.16.1.5
- 172.16.1.6
- 172.16.1.7
- 172.16.1.8
- 172.16.1.9
- 172.16.1.10
- 172.16.1.11
# - 172.16.1.12
- 172.16.1.13
- 172.16.1.14
- 172.16.1.15
- 172.16.1.16
- 172.16.1.17
- 172.16.1.18
- 172.16.1.19
- 172.16.1.20
- 172.16.1.21
# - 172.16.1.22
- 172.16.1.23
- 172.16.1.24
- 172.16.1.25
- 172.16.1.26
- 172.16.1.27
- 172.16.1.28
- 172.16.1.29
- 172.16.1.30
- 172.16.1.31
- 172.16.1.32
```

```
▶ In [9]: #task
set_fact:
  junos_show: "{{ junos | json_query(query) | parse_cli_textfsm('/home/notebook/notebooks/ntc-templates/templates/juniper_junos_show_chas
vars:
  query: "results[].stdout | [0]"
```

```
TASK [set_fact] *****
fatal: [172.16.1.22]: FAILED! => {
  "msg": "parse_cli_textfsm input should be a string, but was given a input of <type 'NoneType'>"
}
fatal: [172.16.1.2]: FAILED! => {
  "msg": "parse_cli_textfsm input should be a string, but was given a input of <type 'NoneType'>"
}
ok: [172.16.1.12] => {
  "ansible_facts": {
    "junos_show": [
      {
        "CPU": "",
        "DESCRIPTION": "VSRX",
        "FANTRAY": "",
        "FPC": "",
        "HARDWARE": "Chassis",
        "MACHINETYPE": "",
        "NIC": "",
        "PARTNUM": "",
        "PIC": "",
        "PORT_PARTNUM": "",
        "PORT_SN": "",
        "SN": "d84a8b121d1f",
        "VID": ""
      }
    ]
  }
}
```

# demo

仮想ルータなので、空の情報しか取れませんでした……







# データの集計①

ansibleでshowコマンドを叩いて収集したネストされまくったデータがこちらです。

```
"stdout": "Mon Jul 24 11:00:34.718 JST\nNAME: \"0/RSP1\", DESCR: \"TEST_CISCO_CISCO\"\nPID: TEST-CISCO-RSP    , VID: V01, SN: TESTIOS0001\nNAME: \"0/3\", DESCR: \"TEST_CISCO_CISCO\"\nPID: TEST-CISCO-99X100GE-TR    , VID: V01, SN: TESTIOS0002\nNAME: \"0/3-PORT-1\", DESCR: \"TEST_CISCO\"\nPID: QSFP-100G-LR4-S    , VID: V02, SN: TESTIOS0003\nNAME: \"0/3-PORT-2\", DESCR: \"TEST_CISCO\"\nPID: QSFP-4X10G-LR-S    , VID: V02, SN: TESTIOS0004\nNAME: \"0/3-PORT-3\", DESCR: \"TEST_CISCO\"\nPID: QSFP-100G-SR4-S    , VID: V03, SN: TESTIOS0005\nNAME: \"0/3-PORT-4\", DESCR: \"TEST_CISCO\"\nPID: QSFP-40G-SR4    , VID: V03, SN: TESTIOS0006\nNAME: \"0/FC0\", DESCR: \"TEST_CISCO_CISCO\"\nPID: TEST-CISCO-SFC-T    , VID: V01, SN: TESTIOS0007\nNAME: \"0/FC1\", DESCR: \"TEST_CISCO_CISCO\"\nPID: TEST-CISCO-SFC-T    , VID: V01, SN: TESTIOS0008\nNAME: \"0/FC2\", DESCR: \"TEST_CISCO_CISCO\"\nPID: TEST-CISCO-SFC-T    , VID: V01, SN: TESTIOS0009\nNAME: \"0/FC3\", DESCR: \"TEST_CISCO_CISCO\"\nPID: TEST-CISCO-SFC-T    , VID: V01, SN: TESTIOS0010\nNAME: \"0/FC4\", DESCR: \"TEST_CISCO_CISCO\"\nPID: TEST-CISCO-SFC-T    , VID: V01, SN: TESTIOS0011\nNAME: \"Rack 0\", DESCR: \"TEST_CISCO X Line Card Slot Chassis\"\nPID: TEST-CISCO    , VID: V01, SN: TESTIOS0012\nNAME: \"0/FT0\", DESCR: \"TEST_CISCO_CISCO\"\nPID: TEST-CISCO-FAN    , VID: V01, SN: TESTIOS0013\nNAME: \"0/FT1\", DESCR: \"TEST_CISCO_CISCO\"\nPID: TEST-CISCO-FAN    , VID: V01, SN: TESTIOS0014\nNAME: \"0/PT0\", DESCR: \"TEST_CISCO\"\nPID: TEST-CISCO-AC-PEM    , VID: V03, SN: TESTIOS0015\nNAME: \"0/PT0-PM0\", DESCR: \"TEST_CISCO\"\nPID: PWR-XKW-AC    , VID: V03, SN: TESTIOS0016\nNAME: \"0/PT0-PM1\", DESCR: \"TEST_CISCO\"\nPID: PWR-XKW-AC    , VID: V03, SN: TESTIOS0017\n"
```

※テストデータです

## データの集計②

json\_queryとntc-templates(正規表現)で

イイ感じのjsonにパース



pandasでcsvに変換して表を出力

→demo

**Pandas**



```

In [212]: #python
          ##Juniper
          import json
          import csv
          import pandas as pd

          json_file_junos = "/home/notebook/notebooks/junos.json"
          csv_file_junos = "/home/notebook/notebooks/junos.csv"
          perse_dict1 = jperse['ansible_facts']['perse_data1']

          f = open(json_file_junos,"w")
          json.dump(perse_dict1, f, ensure_ascii=False, indent=4, sort_keys=True, separators=(',', ':'))
          f.close()

          df = pd.read_json(json_file_junos)
          df.to_csv(csv_file_junos)

          df[['DESCRIPTION', 'SN']]
    
```

Out[1]:

	DESCRIPTION	SN
0	JNPTEST001 [MX999999]	TEST1
1	Midplane X	TESTJP01
2	BUILTIN RE TEST	BUILTIN
3	Control Board	TESTJP02
4	LC9999	TESTJP03
5	LC 9999 PMB	TESTJP04
6	4xQSFP28 SYNCE	BUILTIN
7	LC9999	TESTJP99
8	LC 9999 PMB	TESTJP07
9	4xQSFP28 SYNCE	BUILTIN
10	Power Supply AC	TESTJNP0005
11	Power Supply AC	TESTJNP0006

# まとめ

## やったこと

- ・面倒な棚卸を自動化したい
- ・ansibleを使ってSNMPおよびSHOWコマンドを自動収集
- ・json\_query、textfsm、pandasを使って成形

## やってみたいこと

- ・シリアルナンバー、機器名以外も取得したい
- ・機器ごとの集計などしたい
- ・イントラネット内のPC等も棚卸がある。。。

# 【おまけ】デモ用のDockerfile/サンプルnotebook

以下のリポジトリにあります。

<https://github.com/as2518/janog44hackathon>