



# Cisco VIRL+ NSO + pyATS/Genie で実践する NetDevOps

モダンなツールでネットワーク試験を自動化

Shunsuke Sasaki, Technical Solutions Architect

Cisco Systems G.K.

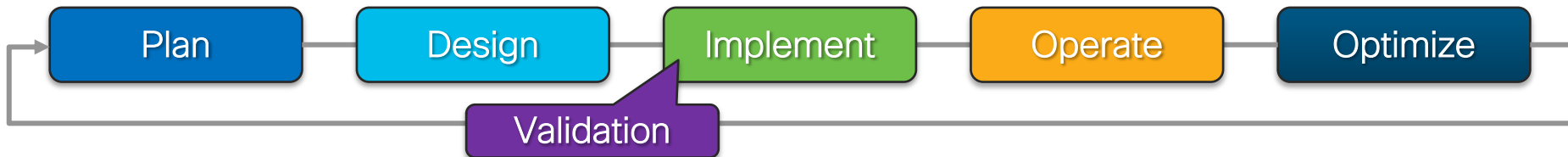
Oct 19, 2018 | Open Networking Conference Japan 2018

# 本日も話しする内容

- ネットワーク試験の自動化 インTRODクシヨン
- Cisco pyATS/Genie とは
- Cisco VIRL + NSO + pyATS/Genie デモ
  - ネットワーク試験自動化ユースケース

# 運用ライフサイクルとネットワーク自動化

## Full Lifecycle of Network Operations



Day-0

### プランニングとネットワーク デザインの自動化

- トポロジ最適化
- 帯域などのキャパシティ管理
- 機器リソース管理自動化

Day-0.5

### テストの自動化

- 単体 / 結合テスト
- HW テスト
- SW テスト
- SMU / Patch テスト

Day-1

### 設定の自動化

- 開通作業の自動化
- 回線追加作業(サービスオーダー)の自動化
- 増設作業の自動化
- アップグレード・マイグレーション作業自動化
- ゼロタッチプロビジョニング

Day-2

### 運用の自動化

- 監視
- ログ収集
- テレメトリ情報分析
- Feedback to Operate
- Feedback to Plan/Design

# ネットワークテスト自動化への期待

## 現状のテストプロセス

テスト項目作成



手順書の作成



テストベッドの準備・配線

テストを実行

ログの収集・分析・確認



合否判定

## 課題

- Excel シートでの手動管理。しばしば変更履歴が不明に。
- Excel シートかテキストファイル      ➤ テストごとに作成
- テストごとに物理環境を用意しセットアップ
- ステップごとに手動で実行
- コンソール出力を目視確認、エクセルで整理
- 人間が判定、結果を Excel シートに記入

# ネットワークテスト自動化への期待

## 現状のテストプロセス

テスト項目作成



手順書の作成



テストベッドの準備・配線

テストを実行

ログの収集・分析・確認

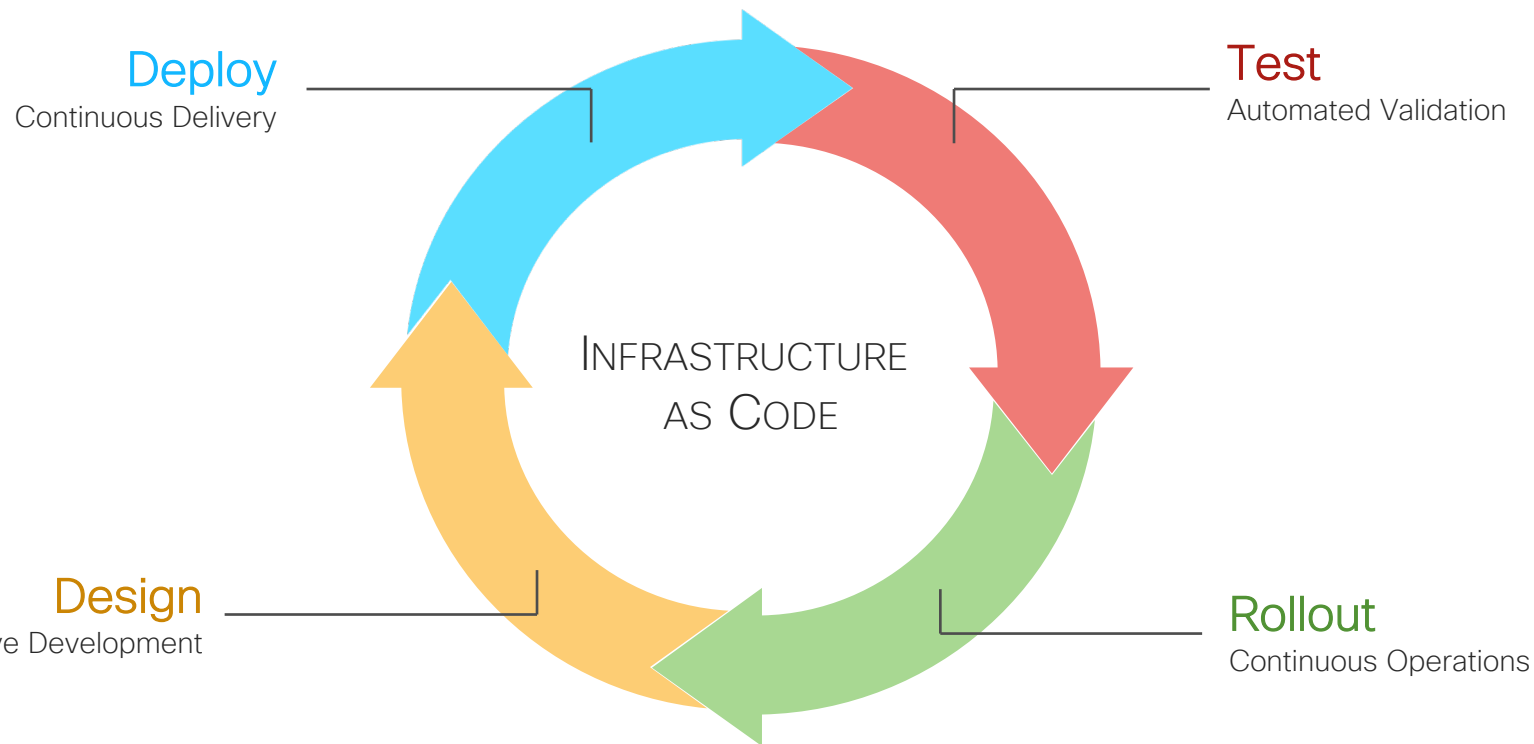


合否判定

## 望ましい姿

- オンラインツールによるバージョン管理
- コード化: Python 等
- コードの再利用性
- 共通の物理テスト環境
- 自動実行・スケジューリング
- ログの自動収集、レポート可視化
- 判定基準にそって自動的に判定

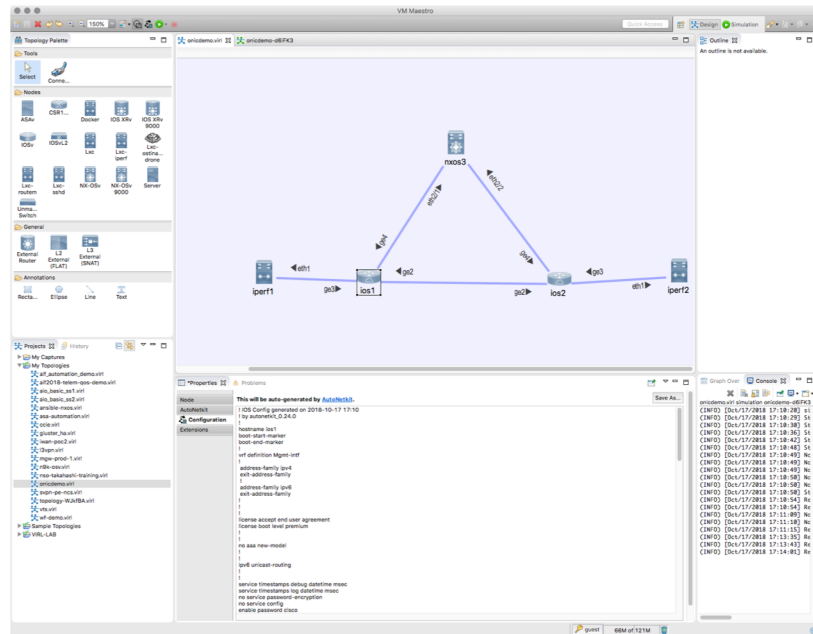
# 目指すべき姿: NetDevOps



# Cisco VIRL – Virtual Internet Routing Lab

- <http://virl.cisco.com/>

- 企業 IT/ネットワークにはより高い信頼性が求められている
  - ネットワークも Dev > Test > Prod の CI/CD サイクルが重要に
- Dev/Test 段階で VIRL によるシミュレーションを活用
  - マルチベンダーをサポート・アプリケーションを含む複雑なトポロジーを模擬可能
- NetDevOps エンジニアにとって最適な “virlutils” CLI ツールが利用可能
  - <https://github.com/CiscoDevNet/virlutils>
- 商用サポート版の Cisco Modeling Labs (CML) もある

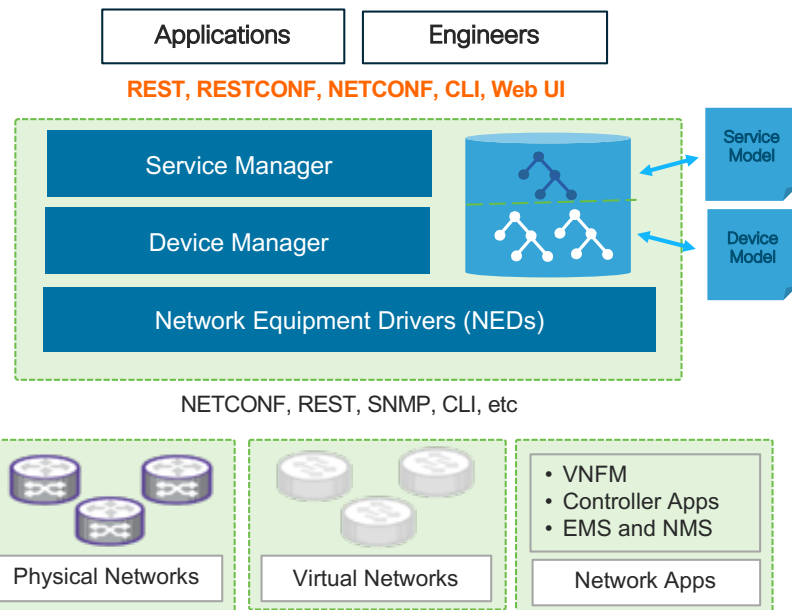


# Cisco NSO – Network Services Orchestrator

• <https://developer.cisco.com/docs/nso>

• “Single API and CLI for the Network”

- ネットワーク全体のコンフィグを保持し、単一の API または CLI でアクセス可能
- YANG モデルベース・高い柔軟性と拡張性
- マルチベンダーサポート(& TAC)

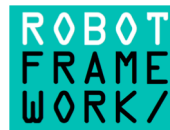




# pyATS – Python Automation Test Systems

- <https://github.com/CiscoTestAutomation>
- Python ベースのテストフレームワーク (Apache 2.0 License)
- 充実した周辺エコシステム
  - デバイス抽象化ライブラリ “**Unicon**”
  - ネットワーク試験向けライブラリ群 “**Genie**”
  - テスト記述言語として “**Robot Framework**”
- ネットワーク試験自動化に必要なあらゆる機能を備える
  - テストベッド・トポロジ管理
  - ネットワーク機器へ接続・コマンド実行・コンフィグ投入
  - テストに対応した Trigger を実行し結果を Verifier で確認
  - ネットワーク・ステートの自動プロファイリング – テスト実行時に before と after のステートを比較
- **FREE**

```
# Verify OSPF neighbor counts
Verify Ospf neighbors agg3
    verify count "5" "ospf neighbors" on device "agg3"
Verify Ospf neighbors agg4
    verify count "5" "ospf neighbors" on device "agg4"
Verify Ospf neighbors core1
    verify count "4" "ospf neighbors" on device "core1"
Verify Ospf neighbors core2
    verify count "4" "ospf neighbors" on device "core2"
```



*Unicon  
Connection Library*

# Cisco pyATS/Genie テスト自動化ソリューション

## Robot

- Keyword driven test automation
- English-like test scripts

## Genie Libs

- Feature model implementation
- Triggers, Verifications, Parsers, Connectors, etc

## Genie Library Framework

- Basis for agnostic automation libraries
- Stimulus, Event & Activities based automation

## pyATS Core Test Infrastructure

- Topology & Test definition
- Execution & Reporting

# pyATS - Getting Started

pyATS/Genie は Python Package Index (PyPI) から利用可能

- <https://pypi.python.org/pypi/pyats/>

## システム要件:

- Mac/Linux Environment (including WSL)
- Python 3+ virtual environment

```
## create a new python virtual environment
$ python3 -m venv ~/pyats

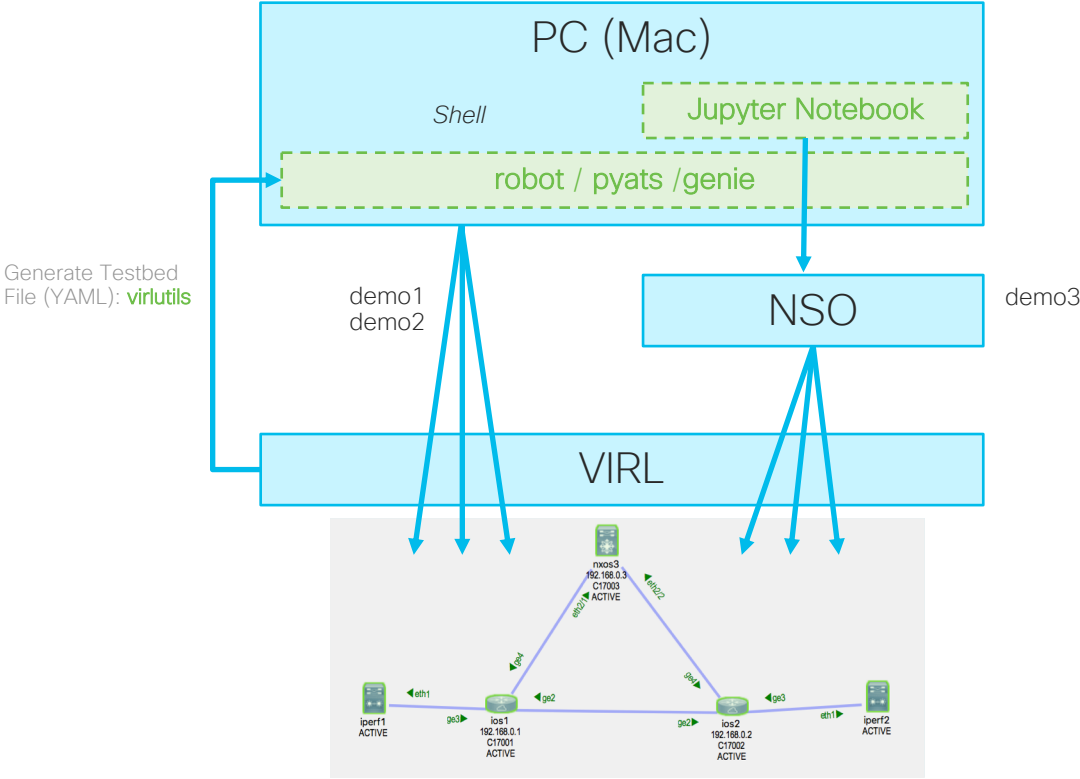
## install in your new environment
$ source ~/pyats/bin/activate
$ pip install pyats genie genie.libs.robot
```

Demo

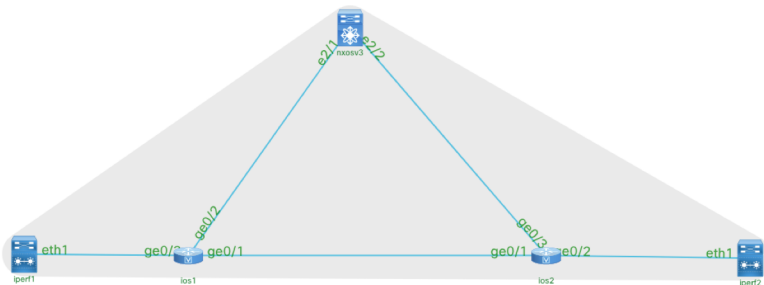
# Demo

- 環境とツールのご説明
- Use Case 1
  - **Unicon Library** を利用したプログラミング的デバイス操作と試験
- Use Case 2
  - **Genie Parser Library** を利用したデバイスステータスの取得と検証
- Use Case 3
  - **NSO** を利用したネットワーク全体の設定変更と試験 & ロールバック

# デモ環境



# pyATS - YAML による Testbed の記述



```
testbed:
  name: 3-router-topo

devices:
  csr:
    connections:
      console:
        ip: 10.94.242.171
        protocol: telnet

  n9k:
    connections:
      console:
        ip: 10.94.242.172
        protocol: telnet

  asr9k:
    connections:
      console:
        ip: 10.94.242.173
        protocol: telnet
```

```
topology:
  csr:
    interfaces:
      GigabitEthernet2:
        link: csr-to-n9k
      GigabitEthernet3:
        link: csr-to-asr9k

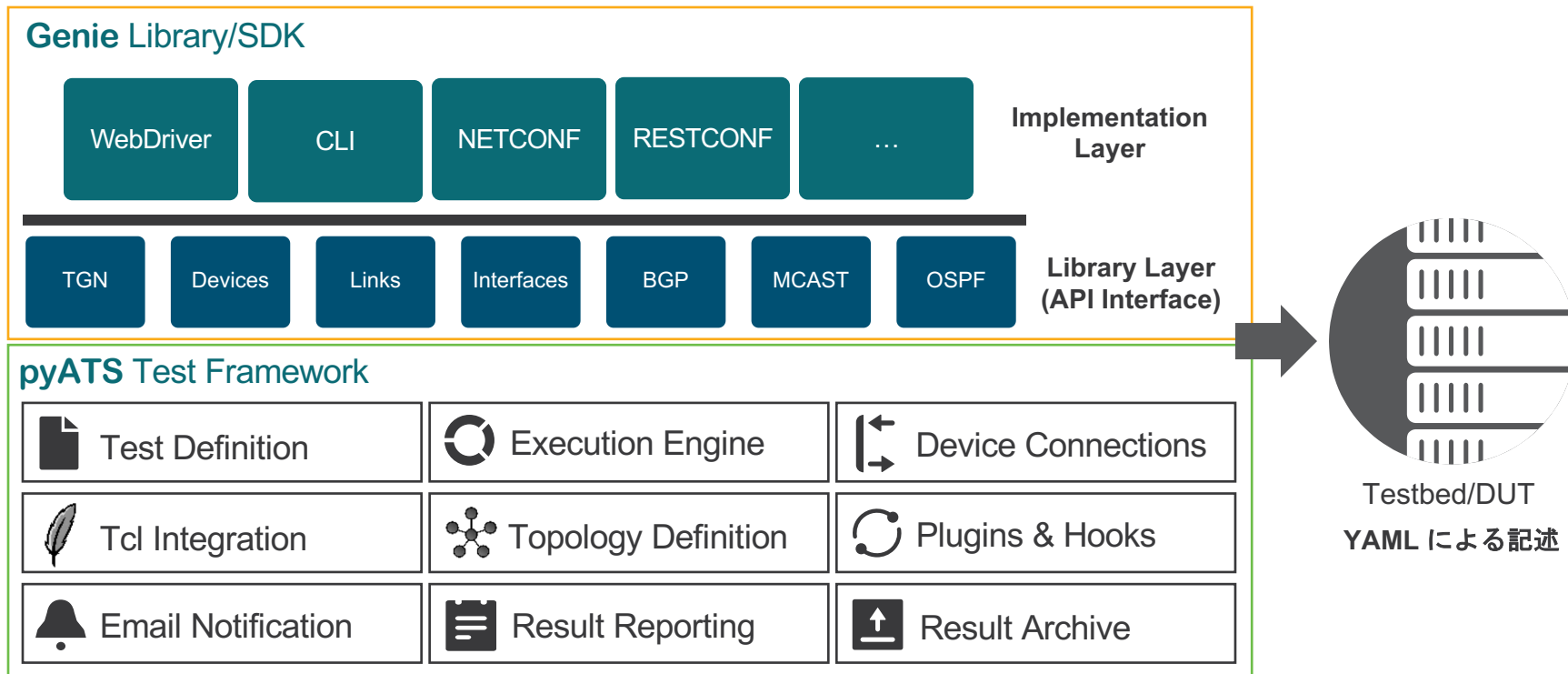
  n9k:
    interfaces:
      Ethernet2/1:
        link: csr-to-n9k
      Ethernet2/2:
        link: n9k-to-asr9k

  asr9k:
    interfaces:
      GigabitEthernet0/0:
        link: csr-to-asr9k
      GigabitEthernet0/1:
        link: n9k-to-asr9k
```

pyATS/Genie まとめ



# pyATS + Genie - Test Automation Ecosystem



# pyATS/Genie まとめ

- Cisco VIRL + NSO + pyATS/Genie を組み合わせたデモをご紹介します
- それぞれ単体でも動きます！ **Better Together**
  - API ドキュメントも充実しており使い方はユーザ次第！
- すでに利用しているテストツール群と簡単に連携できます
  - Git, Ansible, Jenkins, etc...
- pyATS/Genie は FREE !! まずは気軽にお試してください！

# pyATS Resources

- DevNet pyATS Lab: <https://github.com/CiscoDevNet/pyats-labs>
- DevNet: pyATS/Genie - <https://developer.cisco.com/site/pyats/>
- Framework Documentation: <https://developer.cisco.com/site/pyats/docs/>
- Package Documentation: <https://developer.cisco.com/site/pyats/docs/packages/>
- Community Forum: <https://communities.cisco.com/community/developer/pyats>
- GitHub Folder: <https://github.com/CiscoTestAutomation>
- DockerHub: <https://hub.docker.com/r/ciscotestautomation/pyats/>



Thank you



INTUITIVE

# Speaker Introduction



**Shunsuke Sasaki**



[shusasaki@cisco.com](mailto:shusasaki@cisco.com)



[@radiantmarch](https://twitter.com/radiantmarch)



[github.com/radiantmarch](https://github.com/radiantmarch)



[developer.cisco.com](https://developer.cisco.com)



[@CiscoDevNet](https://twitter.com/CiscoDevNet)



[facebook.com/ciscocodevnet/](https://facebook.com/ciscocodevnet/)



<http://github.com/CiscoDevNet>

# これまでの Network Engineer



## Network Skills

- Spanning-Tree
- Routing Protocols
- QoS
- VPN Design
- VOIP
- Fibre Channel
- Security Policy
- MPLS

## Programming Skills

- TCL
- EEM
- Expect Scripts

# これからの NetDevOps Engineer



## Network Skills

- Layer 2 & 3 Fundamentals
- Quality of Service
- Security and Segmentation
- Linux Networking
- Container Networking
- Cloud Networking
- IOT Networking
- Model Driven Programmability
- Network Function Virtualization

## Platform Skills

- Linux Administration
- Container Fundamentals
- Micro Service Platforms
- Cloud Fundamentals

## Programming Skills

- Data Formats (ex: JSON/YAML)
- Python and APIs (ex: REST)
- Source Control (ex: git)
- Configuration Management (ex: Ansible)

# NetDevOps Engineer が使いこなすべきツール群

## Distributed Source Control

(git, Subversion, Mercurial, GitHub, BitBucket, GitLab)

## Build Server

(GitLab, Jenkins, Team City, Drone)

## Configuration Management

(Ansible, Puppet, NSO, NAPALM, DIY)

## Network Test Tooling

(PyATS, TRex, Robot, Behave)

## Telemetry & Monitoring

(ELK, Grafana, Pipeline, UTM)

CLI

SNMP

NETCONF/  
RESTCONF

gRPC

REST APIs

YANG/Native Data Model

Configuration Data

Operational Data

Network Device

**Network  
Virtualization  
Platforms**  
(VIRL/CML, NFVIS,  
Vagrant)

## Development Environment

(Vagrant, NSO, VIRL/CML)

## Test Environment

(VIRL/CML)

## Production Environment



# NetDevOps Engineer が使いこなすべきツール群

(Cisco 製品 & プロジェクト)

Distributed Source Control

Build Server

Configuration Management

Network Service Orchestrator  
(NSO)

Network Test Tooling

PyATS, TRex

Telemetry & Monitoring

Pipeline, UTR

CLI

SNMP

NETCONF/  
RESTCONF

gRPC

REST APIs

Network Virtualization  
Platforms

NFVIS  
VIRL/CML  
Cisco VIM  
VNF

Network Device

IOS XE, IOS XR, NX-OS, Firepower, UCS  
DNA Center, APIC, Meraki, Viptela, and more

Development Environment

NSO, VIRL, VNFs

Test Environment

VIRL/CML

Production Environment

# NetDevOps Engineer のための Cisco DevNet リソース



- Cisco DevNet: <https://developer.cisco.com/>
- Cisco DevNet [日本]: [https://www.cisco.com/c/m/ja\\_jp/developer.html](https://www.cisco.com/c/m/ja_jp/developer.html)
- VIRT: <https://developer.cisco.com/site/virt/>
- NSO for evaluation: <https://developer.cisco.com/docs/nso/#!getting-nso/getting-nso>
- DevNet: pyATS/Genie - <https://developer.cisco.com/site/pyats/>

