



10時より開始します

*Cisco Community Expert Series Community Live*

回線切替じゃ不十分？

90分で体感する DevNetで広げるインフラ管理の可能性！

【生実演】 QoS最適化もAutomate ~要素検証、導入ステップ~

吉田 聡志 (Satoshi Yoshida)

トレノケート株式会社

December 8<sup>th</sup>, 2021



# ご参加ありがとうございます



Download the Presentation!

本日の資料はこちらからダウンロードいただけます  
<https://community.cisco.com/t5/-/-/ec-p/4498289>

今すぐ登録

資料のダウンロード

# 音声ブロードキャストについて

[音声ブロードキャスト (Audio Broadcast) ] ウィンドウが自動的に表示され、コンピュータのスピーカーから音声がかかります。

[音声ブロードキャスト (Audio Broadcast) ] ウィンドウが表示されない場合は、[ 通話 (Communicate) ] メニューから [音声ブロードキャスト (Audio Broadcast) ] を選択します。

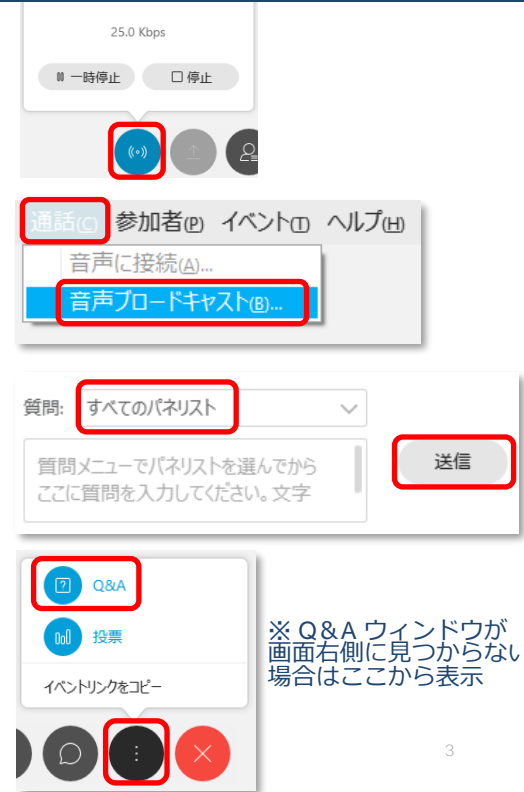
イベントが開始されると自動的に音声がかかります。

音声接続に関する詳細はこちらをご参照ください。

解決しない場合は、Q&A ウィンドウより

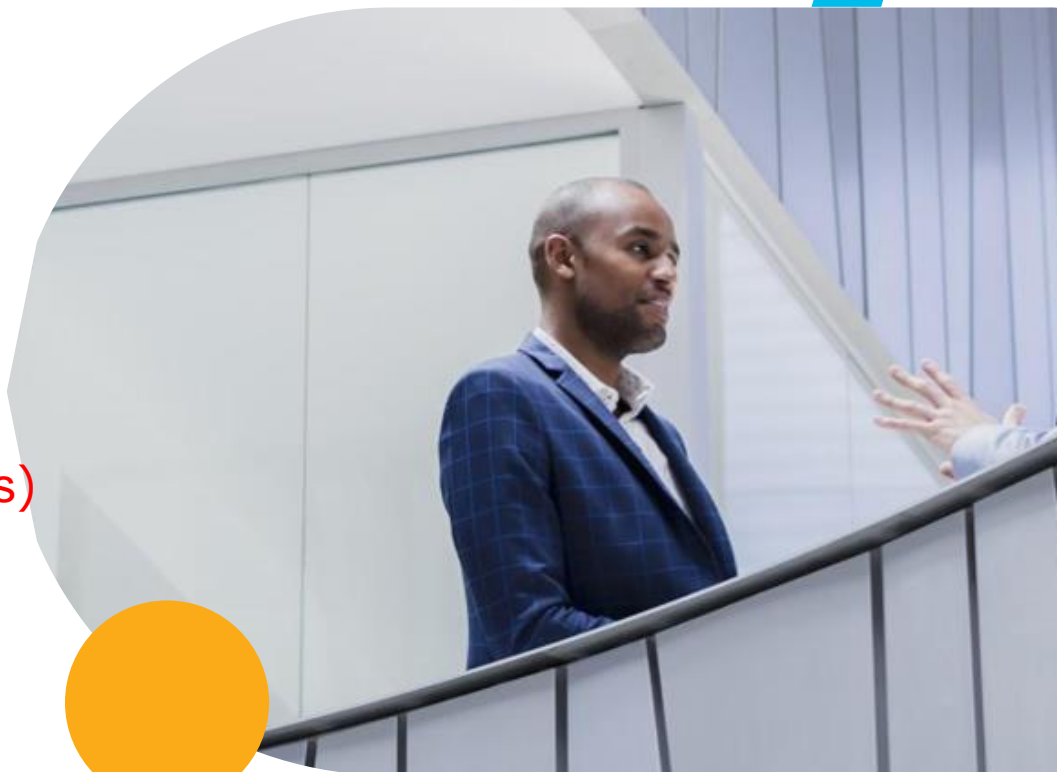
[すべてのパネリスト (All Panelists)] 宛にお知らせください。

<https://community.cisco.com/t5/-/-/ta-p/3129991>



# ご質問方法

Community Live中のご質問は、  
画面右側の Q&A ウィンドウから  
**すべてのパネリスト (All Panelists)**  
宛に送信してください



# 本日のエキスパートご紹介



**吉田 聡志 (Satoshi Yoshida)**  
トレノケート株式会社



Download the Presentation!

本日の資料をダウンロードしてお使いください

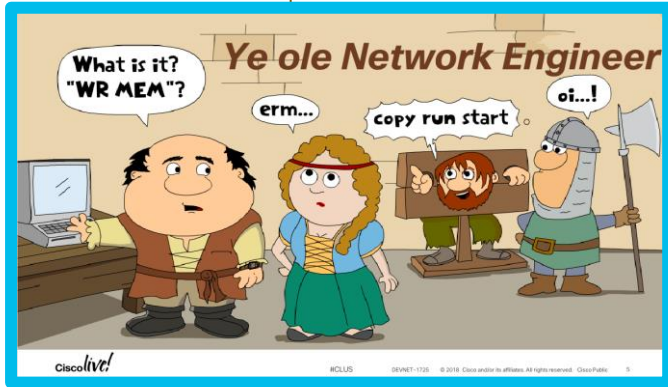
<https://community.cisco.com/t5/-/-/ec-p/4498289>

# はじめに

# セミナーの趣旨ご案内 (1/2)

- 内容
  - DevNetできることを紹介するデモ
  - 題材: 回線トラブル時の自動QoS変更
- もっと詳しく知りたい方のために
  - 詳細な手順書を2021年12月内 公開予定
    - [https://blog.trainocate.co.jp/blog/cisco\\_s01\\_040](https://blog.trainocate.co.jp/blog/cisco_s01_040)
  - 録画公開、発表資料ダウンロード

# How to be a Network Engineer in a Programmable Age



**Stone Age**  
Spanning Tree  
VLANs

**Bronze Age**  
SDN  
Routing Protocols  
WAN Design  
IP-magedon

**The Renaissance**  
SDN  
OpenFlow  
Controllers  
Overlays  
MP-BGP  
VXLAN  
Micro-Segmentation  
White Box

**Programmable Age**  
Cloud  
Python  
REST / APIs  
NETCONF / YANG  
"Fabrics"  
Network Function  
Virtualization (NFV)  
Containers  
DevOps  
NetDevOps!

**The Four Ages of Networking....**

CiscoLive! #CLUS DEVNET-1725 © 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public 18

**Carl's 3 Step Approach to Network Programmability**

**Phase 1**

- Python
- REST APIs
- JSON/XML/YAML
- git/GitHub

**Phase 2**

- Linux Skills
- Config Management
- Docker
- NETCONF/YANG

**As Needed**

- Network Controllers
- IOT Networking
- Cloud Networking
- "DevOps"

**Phase 3**

- Linux Networking
- Container Networking
- Network Function Virtualization

**DEVNET Sandbox**

CiscoLive! #CLUS DEVNET-1725 © 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public 40

CiscoLive!

#CLUS

DEVNET-1725



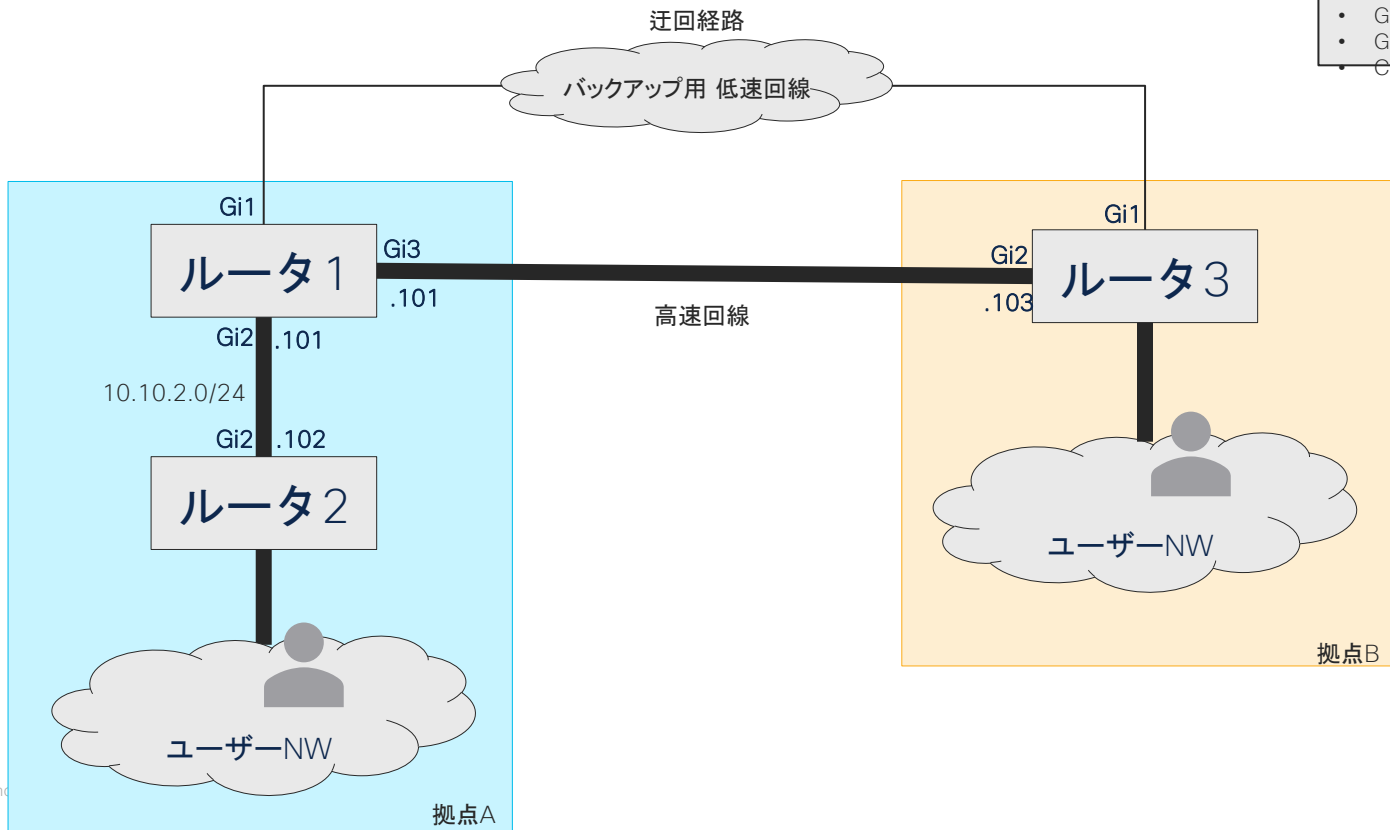
# セミナーの趣旨ご案内 (2/2)

- 推奨視聴スタイル
  - DevNet村の風景を展望台から眺める気持ちで
    - どんな可能性が広がっているか、遠くまで見渡してください



# デモンストレーション環境 構成図

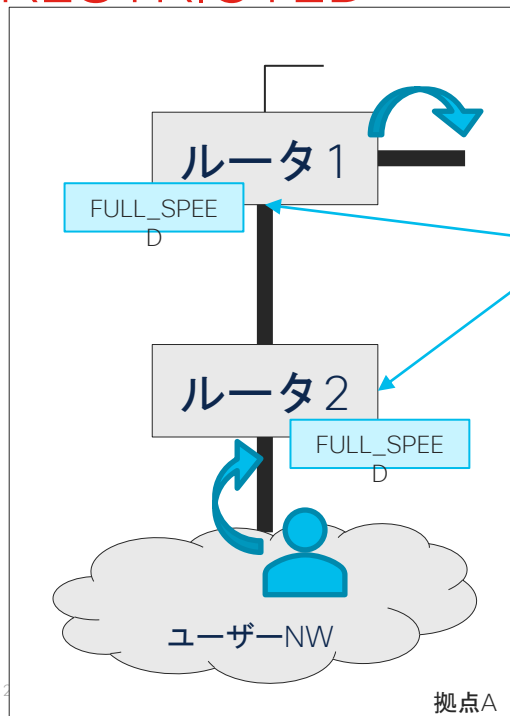
- 使用ソフトウェア、サービス
- Microsoft Azure
- Cisco CSR 1000v IOS-XE 16.12
- Python 2/3
- Visual Studio Code
- Teraterm
- Postman
- Git
- Github
- Cisco Webex Teams



# QoS ポリシー 構成状況

## 正常時 FULL\_SPEED

## RESTRICTED



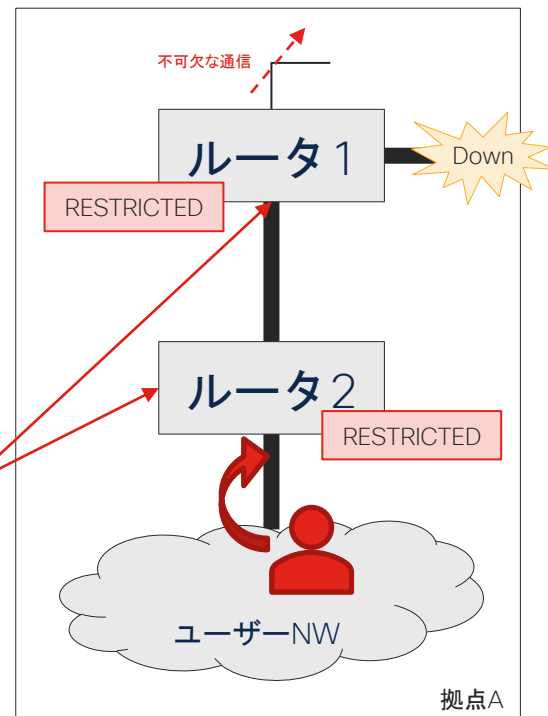
構成:

- それぞれのルータには2つのポリシーが作成済み
- IFのservice-policyアサインをトラブル発生時に変更する

```
policy-map FULL_SPEED
  音声を優先
  ビデオを優先
  潤沢な帯域を贅沢に活用
interface Gigabitethernet 2
  service-policy FULL_SPEED
```

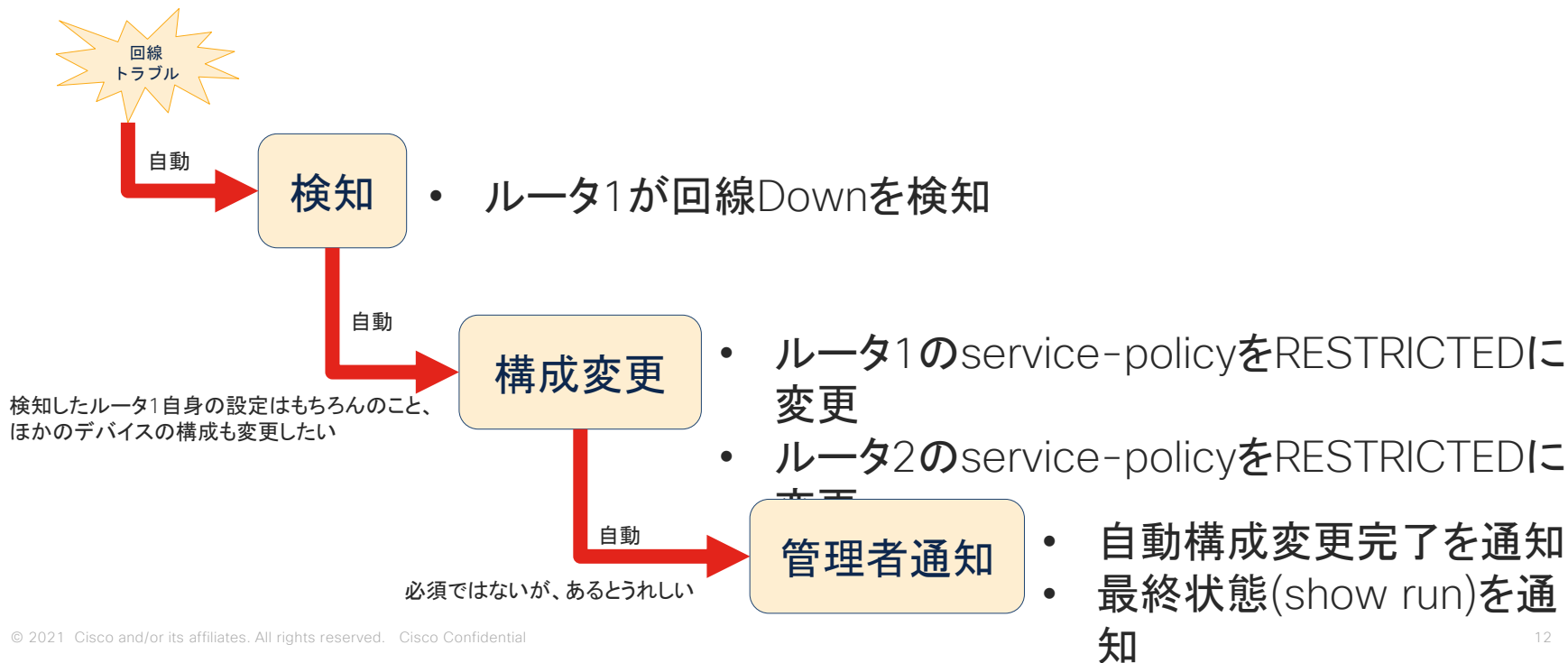
```
policy-map RESTRICTED
  管理通信を優先
  ビデオ会議は、がまん
  業務に不可欠な通信のみを死守
interface Gigabitethernet 2
  service-policy RESTRICTED
```

## トラブル時

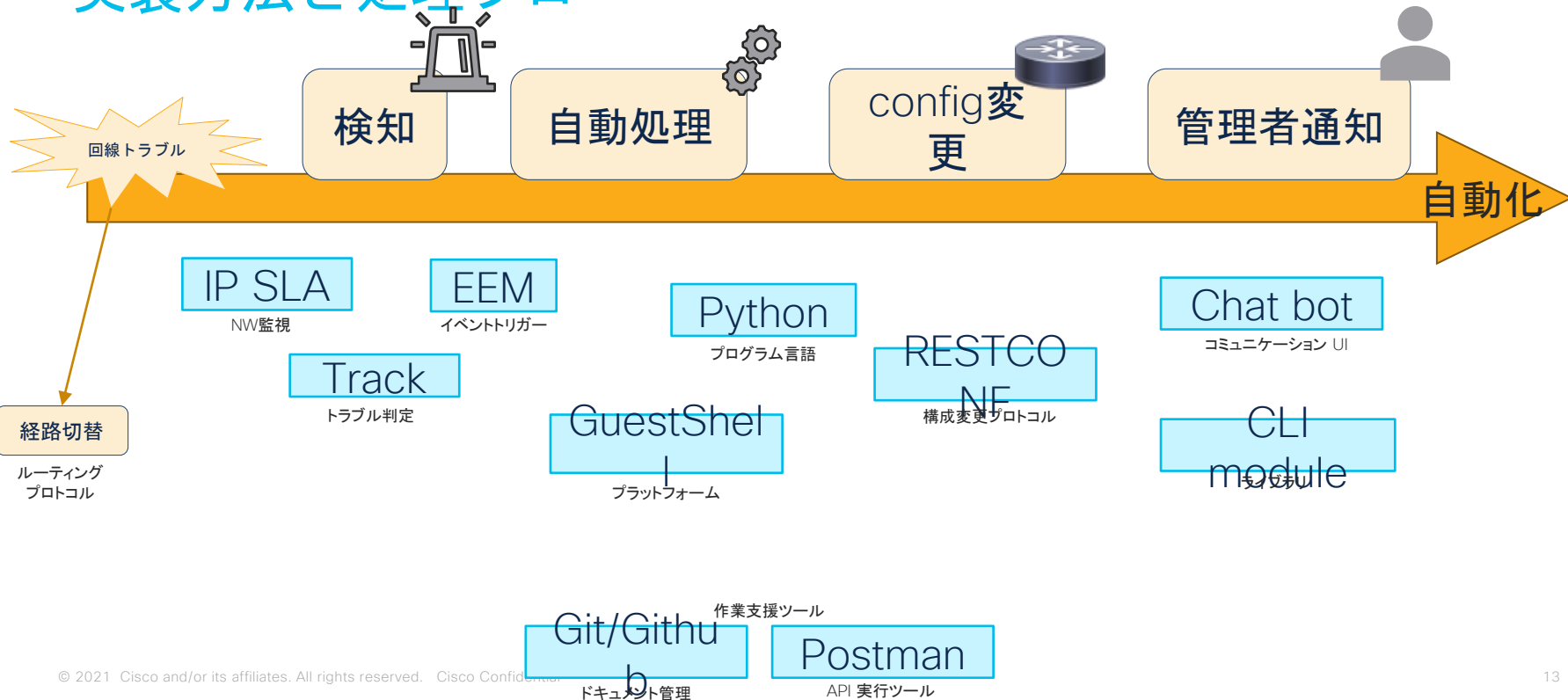


# やりたいこと

## 自動化の要件定義



# やりたいこと 実装方法と処理フロー

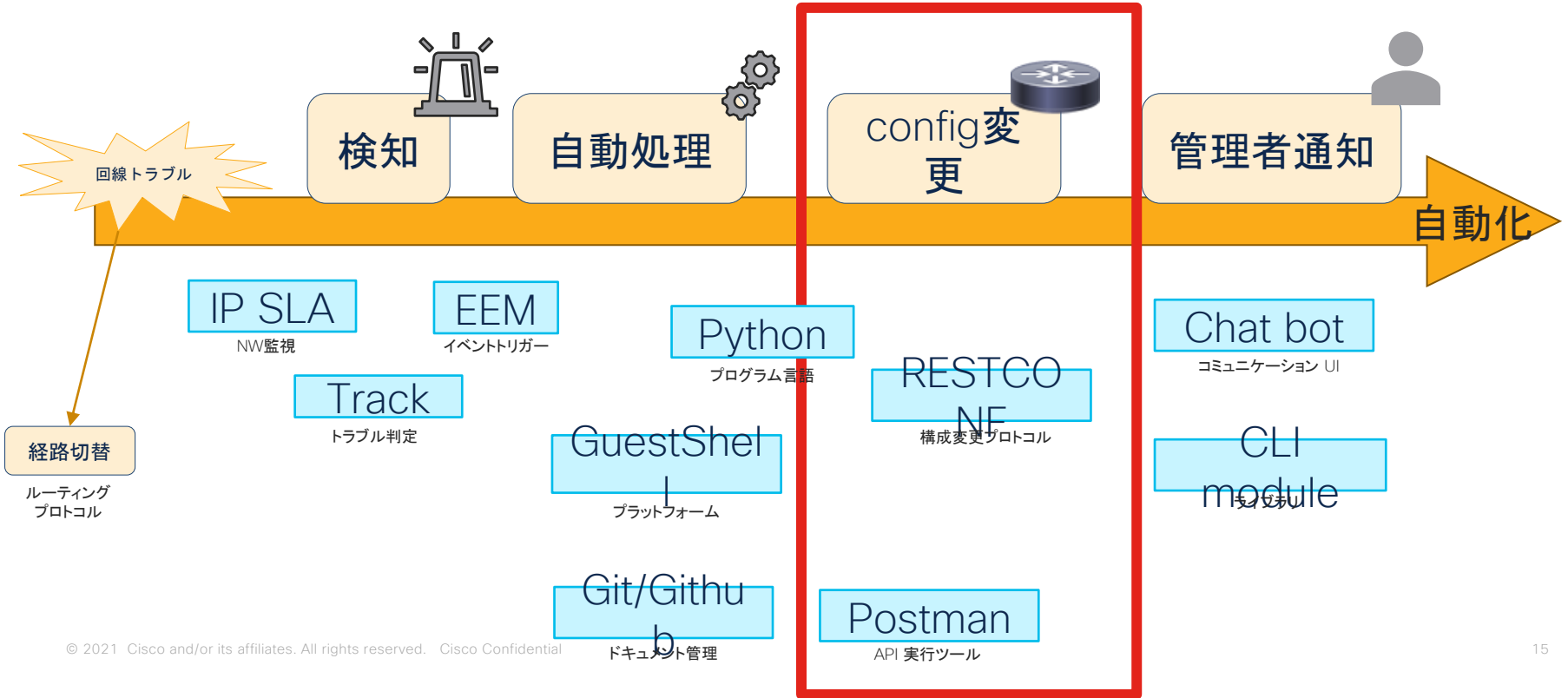


# RESTCONF

Cisco IOS XE で API を動かそう

# 実現したいこと

## このセクションのトピック



# RESTCONF とは

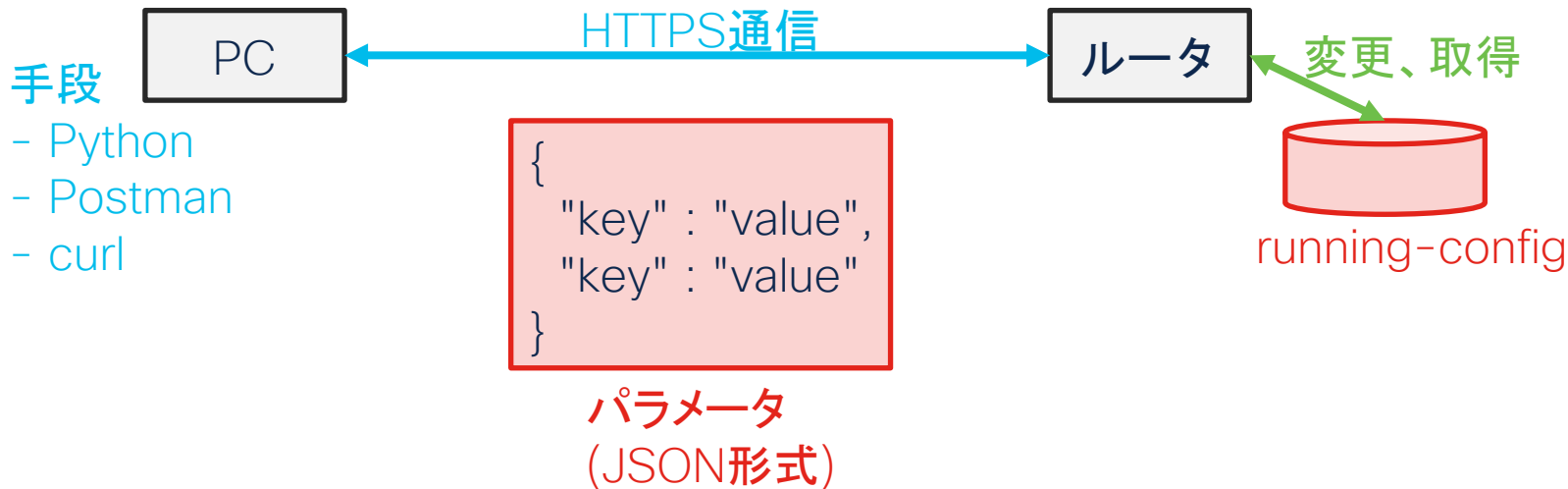
## Minimum 座学

- デバイスを外部から操作する手段のひとつ
- NETCONFとREST APIの良いトコ取り
  - NETCONF (YANGデータモデル) の DB設計を踏襲し
  - REST API の手軽さで利用する
- ルータの running-config を HTTP通信で操作できる



# RESTCONF とは

## Minimum 座学



# RESTCONFのためのPostmanの設定

The screenshot shows the Postman interface for a RESTCONF PATCH request. The main window displays the following details:

- Method:** PATCH
- URL:** https://Router1.az.traino.cat/restconf/data/Cisco-IOS-XE-native:native/interface
- Body Type:** raw
- Body Content:**

```
1  {
2    "Cisco-IOS-XE-native:interface": {
3      "GigabitEthernet": [
4        {
5          "name": "1",
6          "Cisco-IOS-XE-policy:service-policy": {
7            "output": "RESTRICTED"
8          }
9        },
10       {
11         "name": "2",
12         "Cisco-IOS-XE-policy:service-policy": {
13           "input": "RESTRICTED"
14         }
15       }
16     ]
17   }
18 }
```

The Authorization modal window shows the following configuration:

- Authorization Method:** Basic Auth
- Username:** demo
- Password:** .....
- Show Password:**

The following table lists the request headers:

<input checked="" type="checkbox"/>	Authorization ⓘ	Basic ZGVtbzpqYSQkdzByZA==	
<input checked="" type="checkbox"/>	Content-Type	application/yang-data+json	
<input checked="" type="checkbox"/>	Accept	application/yang-data+json	
<input checked="" type="checkbox"/>	Content-Length ⓘ	<calculated when request is sent>	
<input checked="" type="checkbox"/>	Host ⓘ	<calculated when request is sent>	
<input checked="" type="checkbox"/>	User-Agent ⓘ	PostmanRuntime/7.26.8	

# Web API や Postman を学ぶために WebベースAPI基礎とPostmanの使い方

- Community Live 2019 録画資料
- <https://community.cisco.com/t5/-/-/ba-p/4185061>



# RESTCONF 有効化手順

## CLI コマンド

```
conf t
```

```
restconf
```

```
ip http secure-server
```

# 演 デモンストレーション

## RESTCONFでIOS XEのconfigを操作する

- IOS XEでRESTCONFを有効化
- Postmanの設定確認
- PostmanでRESTCONFを実行
- PythonでRESTCONFを実行

# まとめ

## RESTCONF

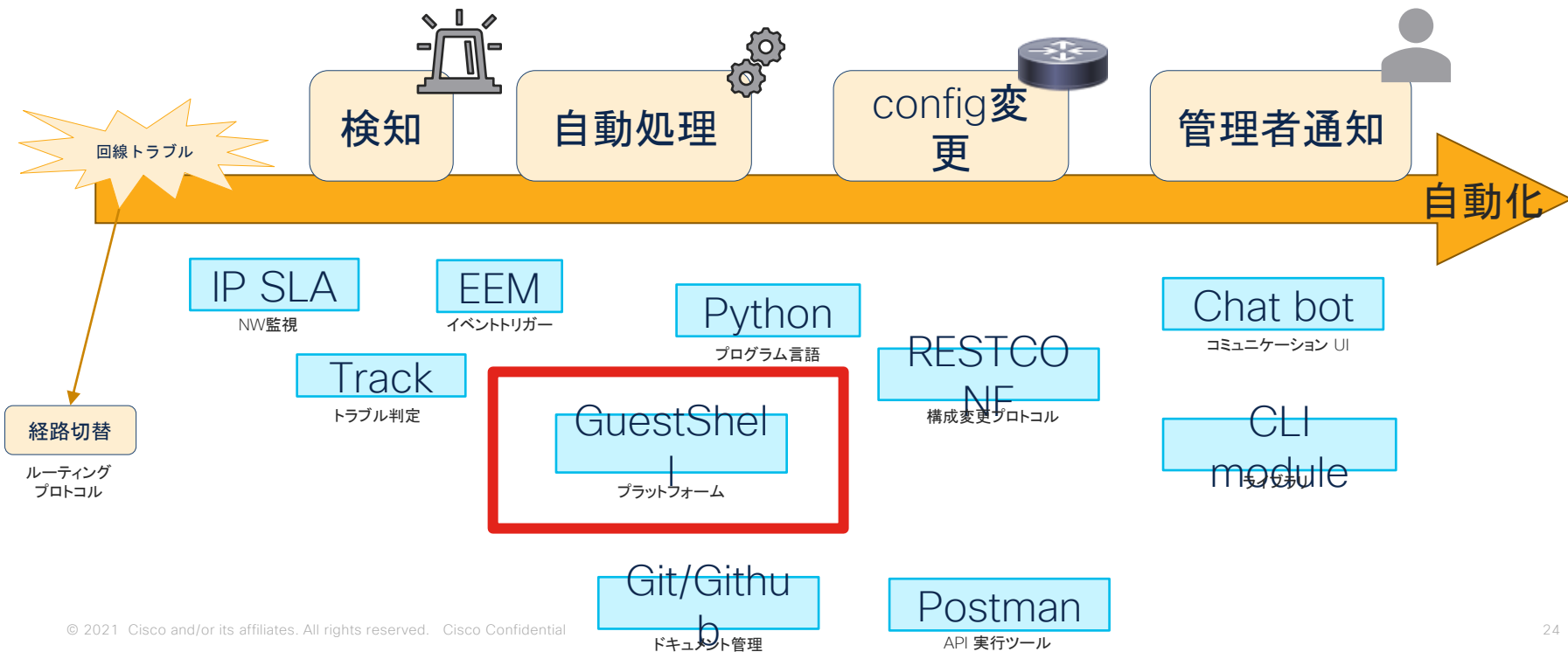
- RESTCONFの有効化手順は簡単
- PythonからIOS XEのconfigを変更できる
- Postmanは便利
  - RESTCONFの動作確認
  - Pythonソースコード生成
  - 検証作業中のconfig準備

# GuestShell

Cisco ルータ を Linux にしよう

# 実現したいこと

## このセクションのトピック





# 投票 ①

GuestShellを知っていますか？

- A) 現場で使ってますよ
- B) 用途は知っているが、自分で使う機会には恵まれません...
- C) 単語だけはどこかで見たような気がする
- D) そんなん初耳だよ

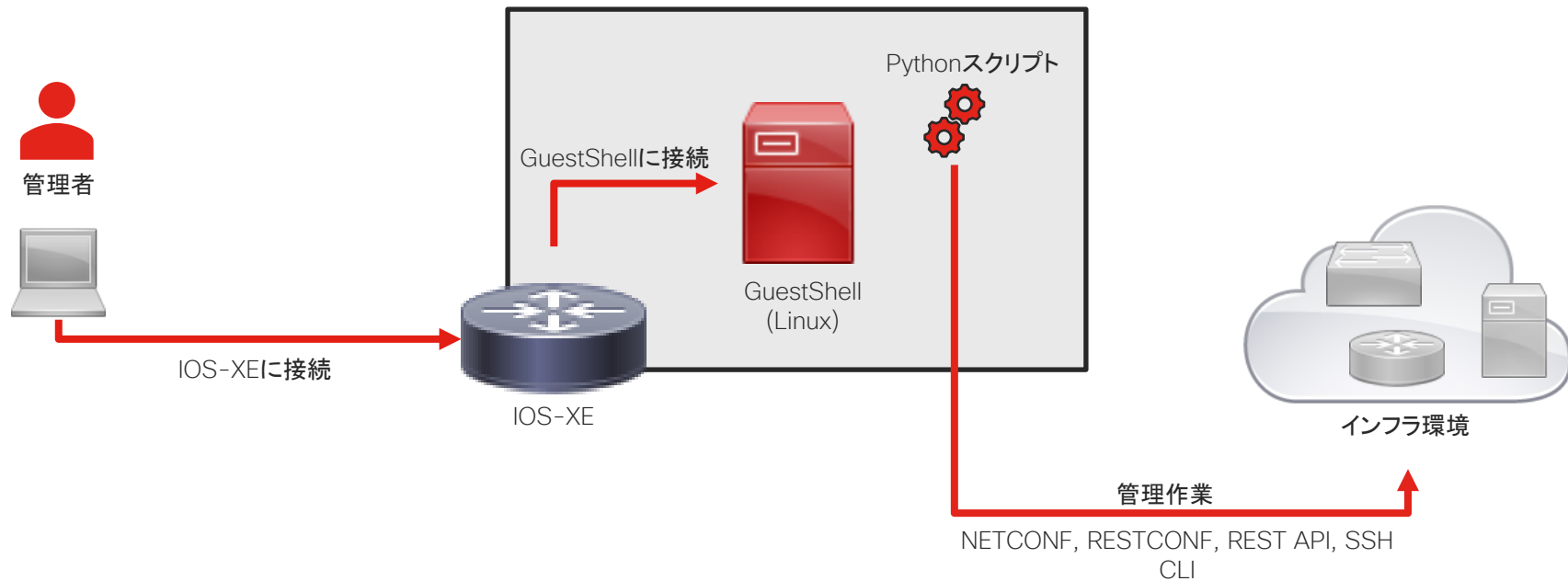
# GuestShell とは

## Minimum 座学

- Ciscoデバイスに内蔵されたLinux環境
  - 仮想マシン相当
- 24/365稼働なプラットフォーム
- インフラ管理のために利用できる

# GuestShell とは

## Minimum 座学



# GuestShell とは

## Minimum 座学

```
Router1#
Router1#show version | include Version
Cisco IOS XE Software, Version 16.12.01a
Cisco IOS Software [Gibraltar], Virtual XE Software (X86_64_LINUX_IOSD-UNIVERSAL
K9-M), Version 16.12.1a, RELEASE SOFTWARE (fc2)
licensed under the GNU General Public License ("GPL") Version 2.0. The
software code licensed under GPL Version 2.0 is free software that comes
GPL code under the terms of GPL Version 2.0. For more details, see the
Router1#
Router1#
Router1#
Router1#
Router1#guestshell
[guestshell@guestshell ~]$
[guestshell@guestshell ~]$ whoami
guestshell
[guestshell@guestshell ~]$ pwd
/home/guestshell
[guestshell@guestshell ~]$
```

# GuestShell 有効化手順 (1/4)

## iox 有効化

conf t

iox

# GuestShell 有効化手順 (2/4)

## VirtualPortGroup Interface 構成

```
conf t
```

```
interface VirtualPortGroup0
```

```
    ip address 172.22.22.254 255.255.255.0
```

# GuestShell 有効化手順 (3/4)

## app-hosting 構成

```
conf t
```

```
app-hosting appid guestshell
```

```
app-vnic gateway0 virtualportgroup 0 guest-interface 0
```

```
    guest-ipaddress 172.22.22.1 netmask 255.255.255.0
```

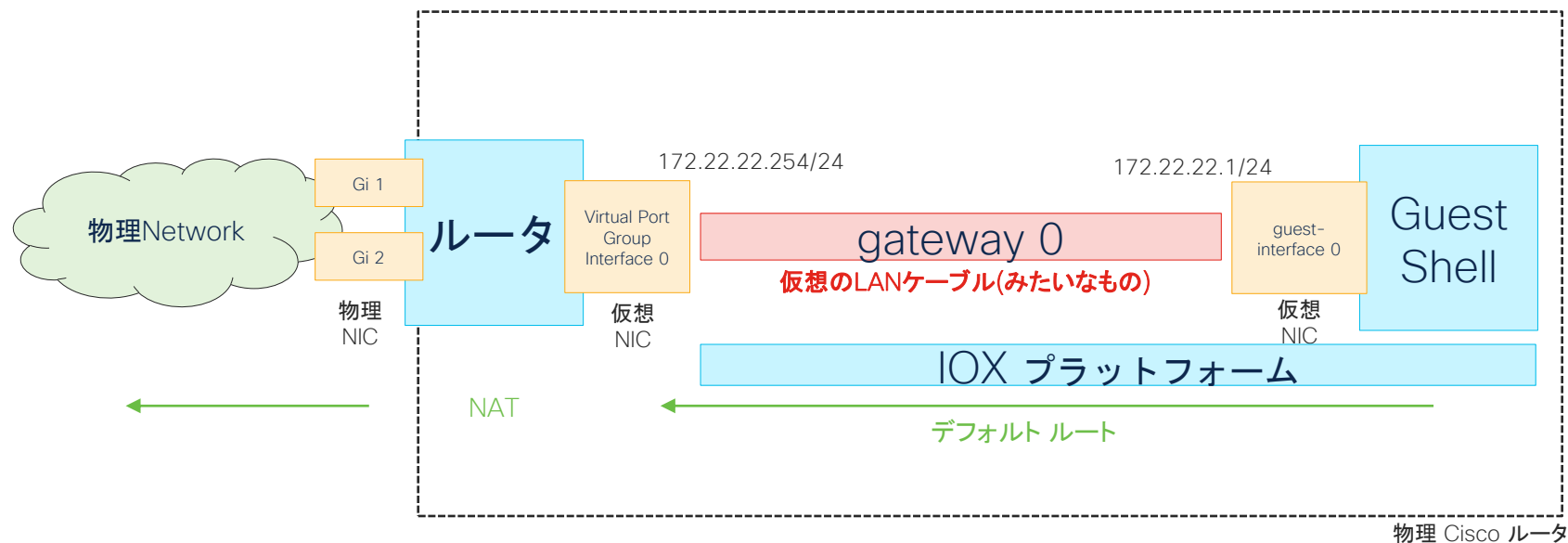
```
    exit
```

```
app-default-gateway 172.22.22.254 guest-interface 0
```

```
name-server0 8.8.8.8
```

# GuestShell 有効化手順 (3/4)

## app-hosting 構成コマンドの考え方





# GuestShell 有効化手順 (4/4)

(オプション) NAT構成 GuestShellが外部と通信する場合は、NAT or Route Advertise が必要

```
conf t
```

```
interface VirtualportGroup 0
```

```
    ip nat inside
```

```
interface GigabitEthernet 1
```

```
    ip nat outside
```

(省略) ACLやVRF、ip nat ~ は環境にあわせて適宜構成してください

# 演 デモンストレーション

## GuestShellを使用する

- GuestShell 起動準備
  - iox 有効化
  - VirtualPortGroup Interfaceの構成
  - app-hostingの構成
  - NATの構成
- guestshell 起動
- guestshell 確認

# まとめ

## GuestShell

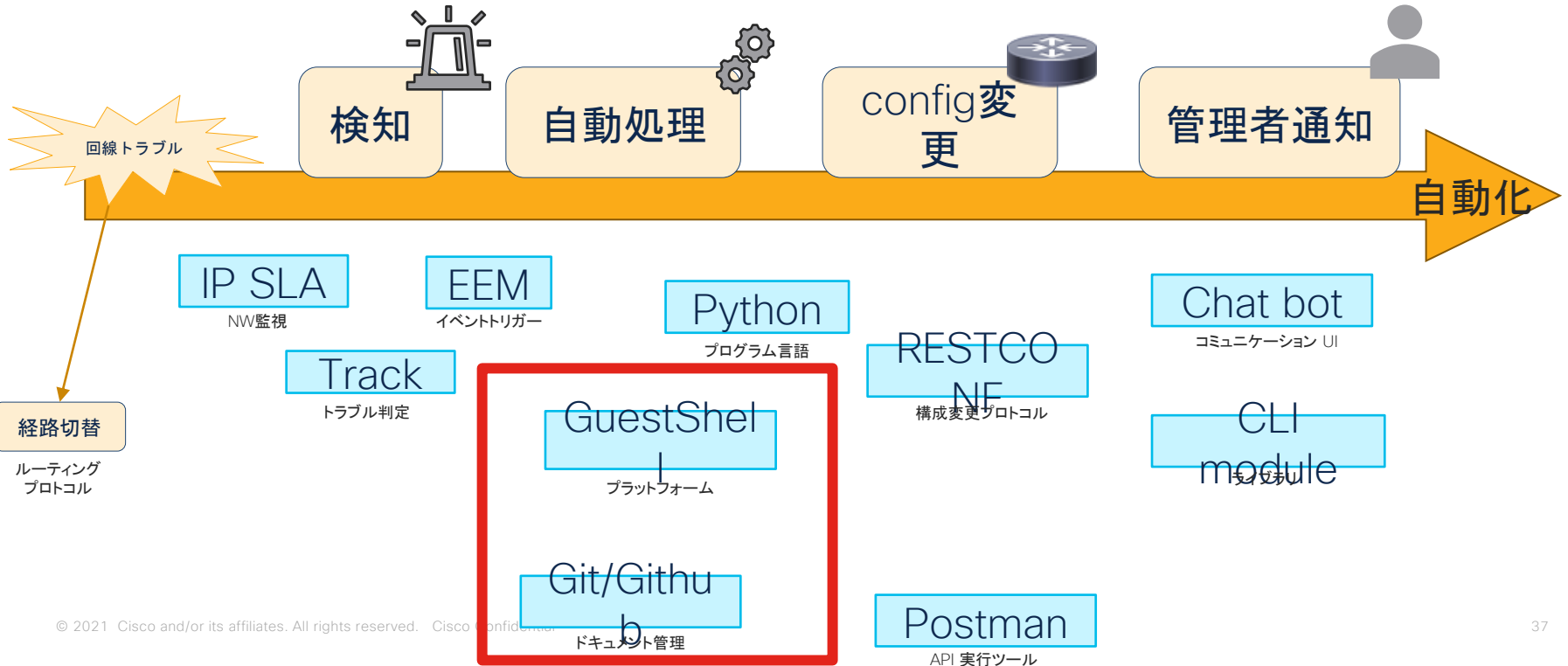
- GuestShell は Linux
- インフラ管理プラットフォーム

# GuestShellの活用

GuestShell に便利ツールをインストールしよう

# 実現したいこと

## このセクションのトピック



# GuestShell の活用

## Minimum 座学

- インフラ管理に役立つ便利ツールは多数存在する
- Python
- Git
- 手軽にGuestShell (Linux Cent OS) にインストールできる

# GuestShell の活用

## Minimum 座学

- Python
  - 学習が容易で人気の高いプログラム言語
  - ライブラリが充実している
- Python requests ライブラリ
  - RESTCONFに必要なHTTPメッセージを簡単に生成
- Git
  - バージョン管理システムとして人気があるソフトウェア
  - リモートリポジトリとしてのGitHubは非常に有名

## 投票 ②

バージョン管理ツール(Git, Subversion, Mercurialなど)

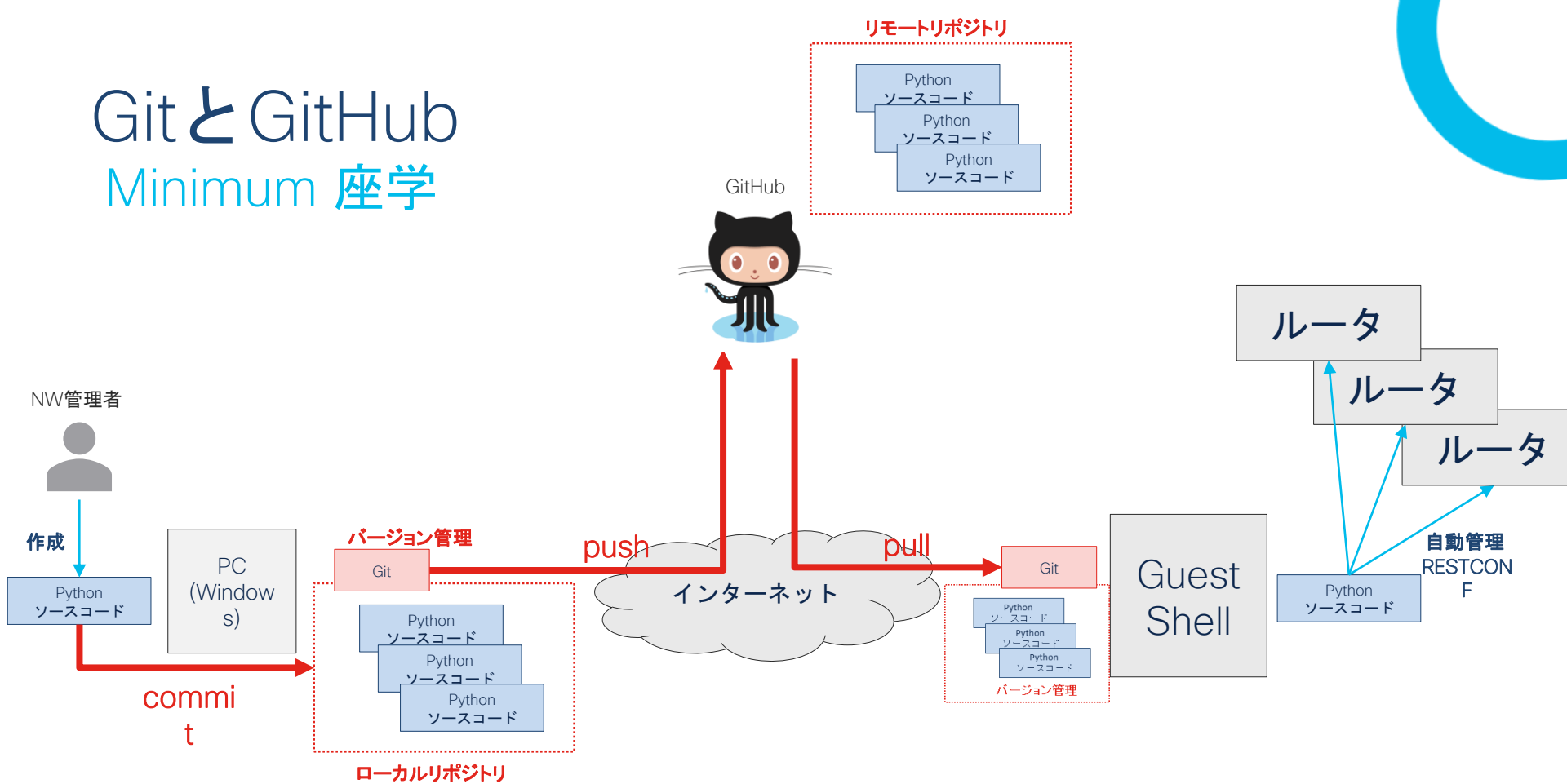
について、  
みなさんの利用状況を教えてください

- A) 職場でもバリバリ活用している
- B) 個人として使っているが、仕事ではイマイチ
- C) 興味本位で触って挫折した過去を持つ
- D) この時まで見て見ぬフリを貫いた自分を誇りに思う



# GitとGitHub

## Minimum 座学



# 演 デモンストレーション

## GuestShell に便利ツールをインストール

- Python3をインストール

```
$ sudo yum install python36
```

- Python requests ライブラリをインストール

```
$ sudo python3 -m pip install requests
```

- Gitをインストール

```
$ sudo yum install git
```

# まとめ

## GuestShell の活用

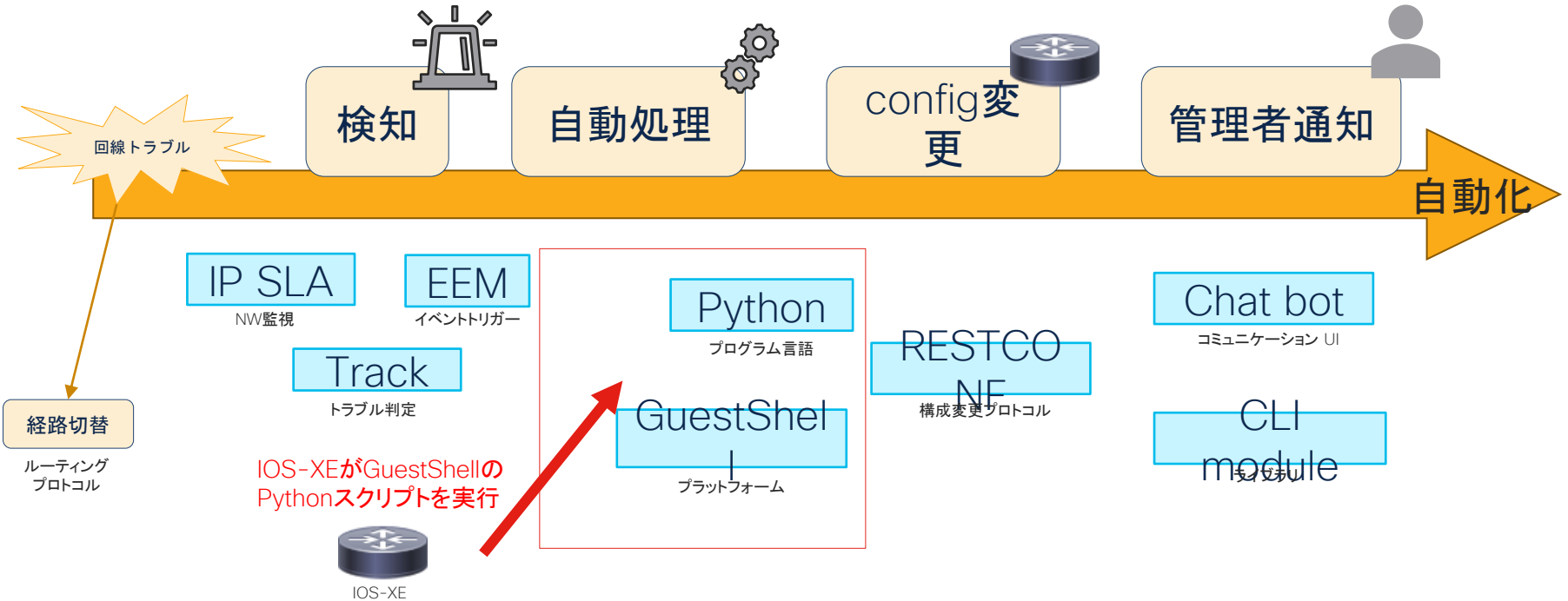
- GuestShell は Linux
- なので、yumが使える

# Python実行

## 幕間

# 実現したいこと

## このセクションのトピック



# Python 実行方法

## Minimum 座学

- LinuxでPythonを実行するコマンド

```
$ python <Pythonスクリプトファイル名>
```

- IOS-XEでGuestShellのコマンドを実行するコマンド

```
# guestshell run <GuestShell上で実行したいLinuxコマンド>
```

# Python を実行する (1/2)

## GuestShellでPythonを実行する

```
# guestshell
```

```
$ python3 ./devnet_demo/app/setQoS.py
```

# Python を実行する (2/2)

## IOS-XE からPythonを実行する

特権モードでguestshellコマンドを実行する

```
# guestshell run python3 "./devnet_demo/app/setQoS.py"
```

※ スクリプトファイルは、./home/guestshell からの相対パスとして指定する

guestshell のカレント ディレクトリはこの時は無関係になる



# まとめ

## IOS-XE からPythonを実行する

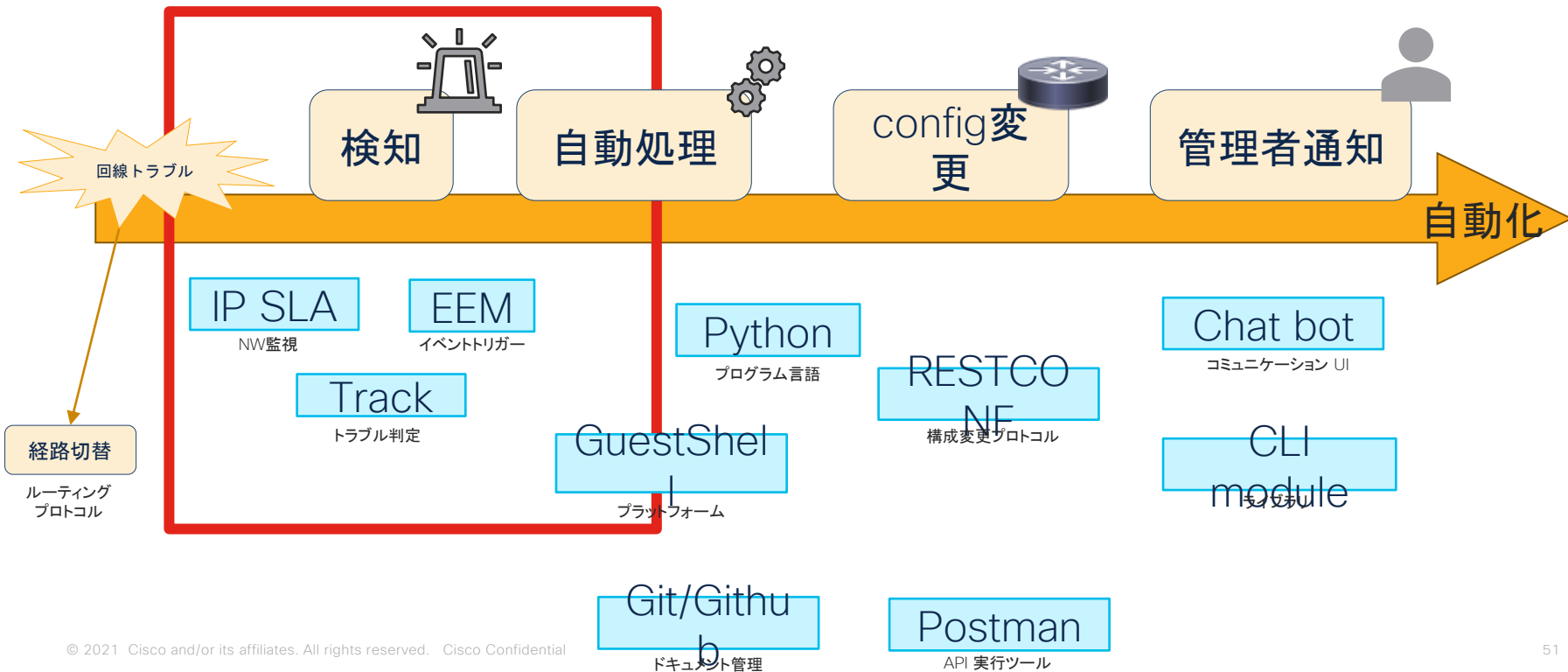
- IOS-XEからGuestShellを操作できる
  - 使用例: Pythonスクリプトを実行する

# IP SLA + Track + EEM

回線障害時に Python スクリプトを起動しよう

# 実現したいこと

## このセクションのトピック



# Ciscoデバイスの自動化コンポーネント

## Minimum 座学

- IP SLA
  - 条件を指定した通信環境測定ができるCiscoデバイスの機能
  - Pingの宛先を指定して疎通確認するのがもっともシンプルな使い方
  - Trackと連携できる
- Track
  - 通信環境の閾値を指定してステータスを監視できるCiscoデバイスの機能
  - IP SLAで測定しているPingが失敗したときにDown判定できる
  - EEMと連携できる
- EEM
  - 指定したイベントが発生した際に、指定した動作を自動実行するCiscoデバイスの機能
  - IP SLAで測定しているPing失敗を監視するTrackがDownしたときに、Pythonを実行するコマンドを実行する

# 回線トラブル検知と自動処理 (1/4)

## IP SLAを構成する

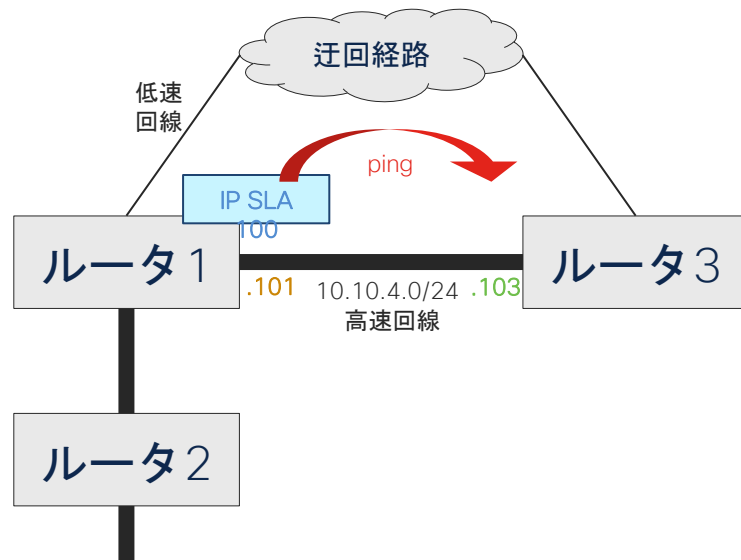
```
conf t
```

```
ip sla 100
```

```
icmp-echo 10.10.4.103 source-ip  
10.10.4.101
```

```
frequency 5
```

```
ip sla schedule 100 life forever start-time  
now
```

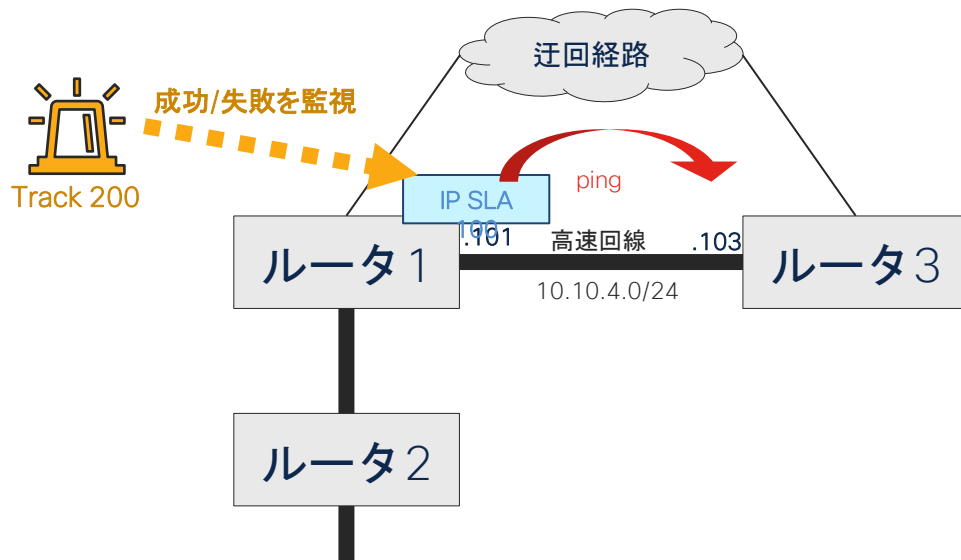


# 回線トラブル検知と自動処理 (2/4)

## Track を構成する

conf t

```
track 200 ip sla 100 reachability
```



# 回線トラブル検知と自動処理 (3/4)

## EEM を構成する

conf t

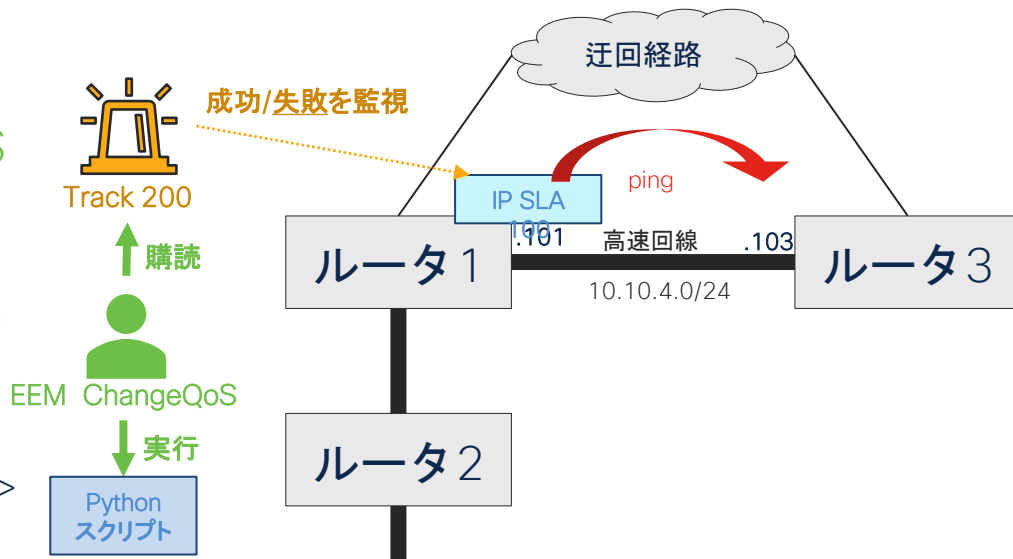
```
event manager applet ChangeQoS
```

```
event track 200 state down
```

```
action 1.0 <実行したいIOSコマンドや>
```

```
action 2.0 <Syslogに出力する文言を>
```

```
action 3.0 <シーケンシャルに記述する>
```



※ IOSのアクセスレベルは ユーザーモード から開始される

# 正しく動作するaction設定を選択してください

## 投票 ③

A

```
event manager applet ChangeQoS
  action 1.0 cli command "guestshell run python3 setQoS.py"
```

B

```
event manager applet ChangeQoS
  action 1.0 cli command "enable"
  action 2.0 cli command "guestshell run python3
/devnet_demo/app/setQoS.py"
```

C

```
event manager applet ChangeQoS
  action 1.0 cli command "enable"
  action 2.0 cli command "guestshell run
python3 ./devnet_demo/app/setQoS.py"
```

D

```
event manager applet ChangeQoS
  action 1.0 cli command "enable"
  action 2.0 cli command "configure terminal"
  action 3.0 cli command "guestshell run
python3 ./devnet_demo/app/setQoS.py"
```



# 回線トラブル検知と自動処理 (4/4)

## EEM を構成する

```
conf t
```

```
event manager applet ChangeQoS
```

```
event track 200 state down
```

```
action 1.0 syslog msg "***** ChangeQoS Applet fired *****"
```

```
action 2.0 cli command "enable"
```

```
action 3.0 cli command "guestshell run  
python3 ./devnet_demo/app/setQoS.py"
```

# 演 デモンストレーション

## IP SLA + Track + EEM

- IP SLAを構成する
- Trackを構成する
- EEMを構成する
  
- 回線トラブル時にPythonスクリプトでQoSが変更されることを確認する

# まとめ

## IP SLA + Track + EEM

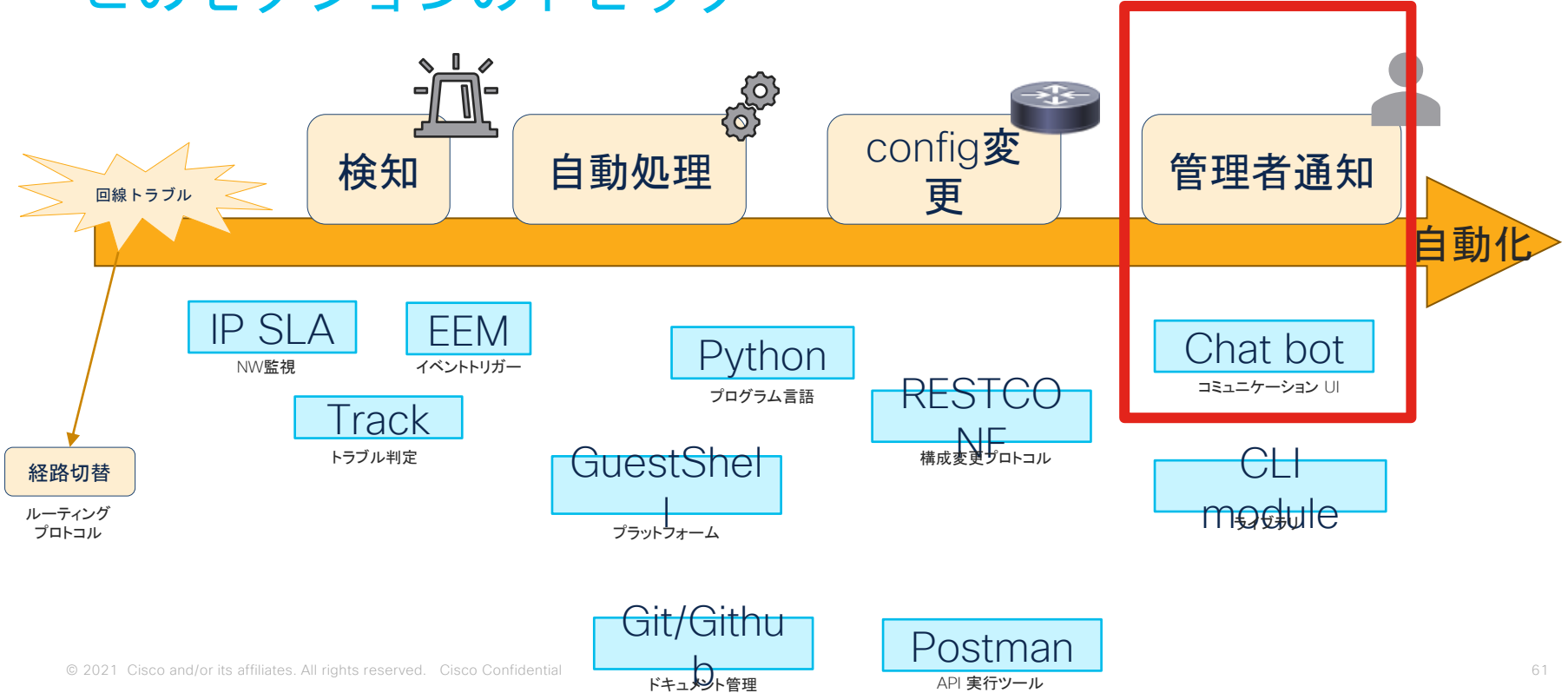
- IP SLAで疎通をチェック
- Trackでトラブルを検知
- EEMでPythonを実行

# (おまけ) Webex Teams API

チャット bot で管理者に通知しよう

# 実現したいこと

## このセクションのトピック



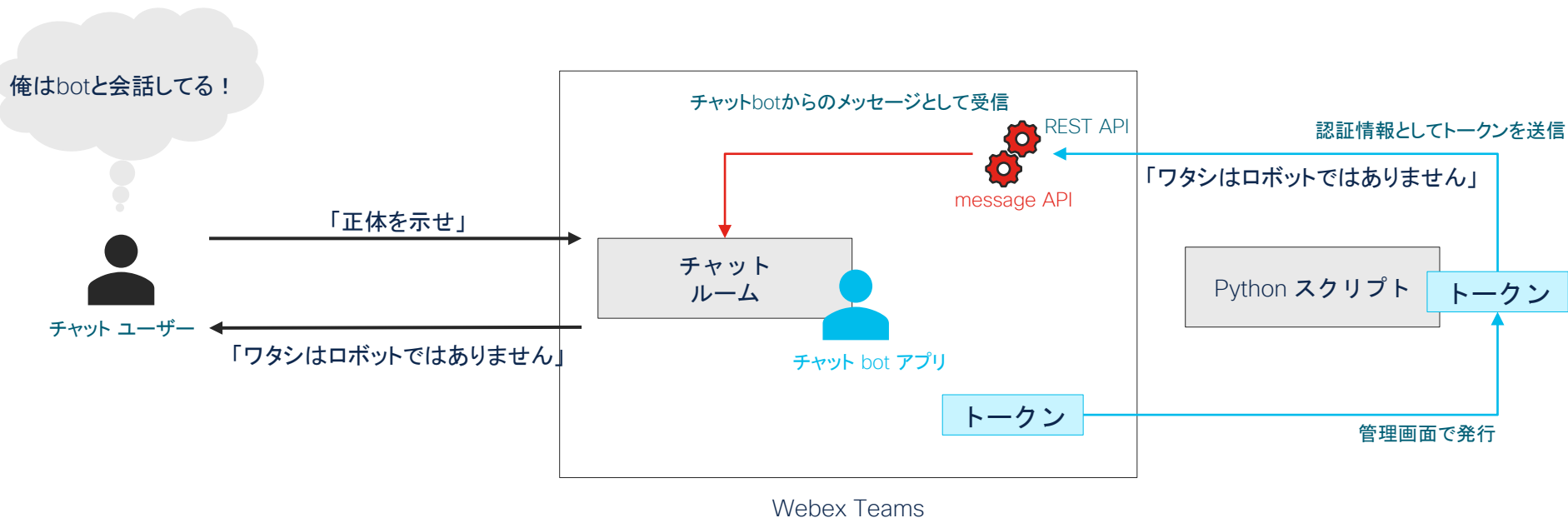
# Webex Teams APIとチャット bot

## Minimum 座学

- Webex Teams
  - チャット ツール
- Webex Teams API
  - チャット機能をプログラムから利用するためのAPI
    - チャットルームを作成する、メッセージを送受信する、など
  - WebベースのREST APIとして、PythonやPostmanで利用できる
- チャット bot
  - チャットをインタフェースとして利用する自動化システム
  - その正体は、チャット機能のAPIを呼び出しているプログラム
  - 難しそうに見えるが、実はすごく簡単につくれる

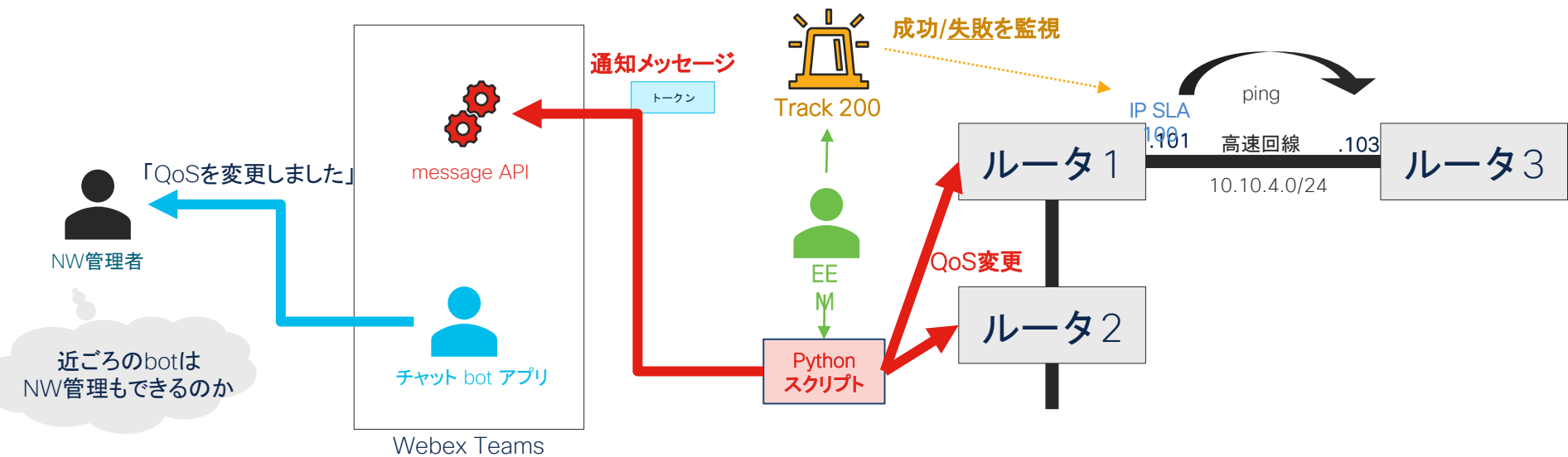
# Webex Teams APIとチャット bot

## チャット bot のイメージ



# 演 デモンストレーション

## Webex Teams チャット bot 構成





# 演 デモンストレーション

## Webex Teams チャット bot 構成

- チャット bot を作成する
- チャット bot の認証情報(トークン)を取得する
- PostmanでWebex TeamsのAPIを使用する
- QoS変更Pythonスクリプト実行時にチャットbotで通知する

# まとめ

## Webex Teamsチャットbot

- チャット bot は誰でも作成できる
- あたかもbotがNW管理を代行したようにみえるが、  
実際はPythonスクリプトが処理をしている
  - チャットbot自体は単なるインタフェースに過ぎない

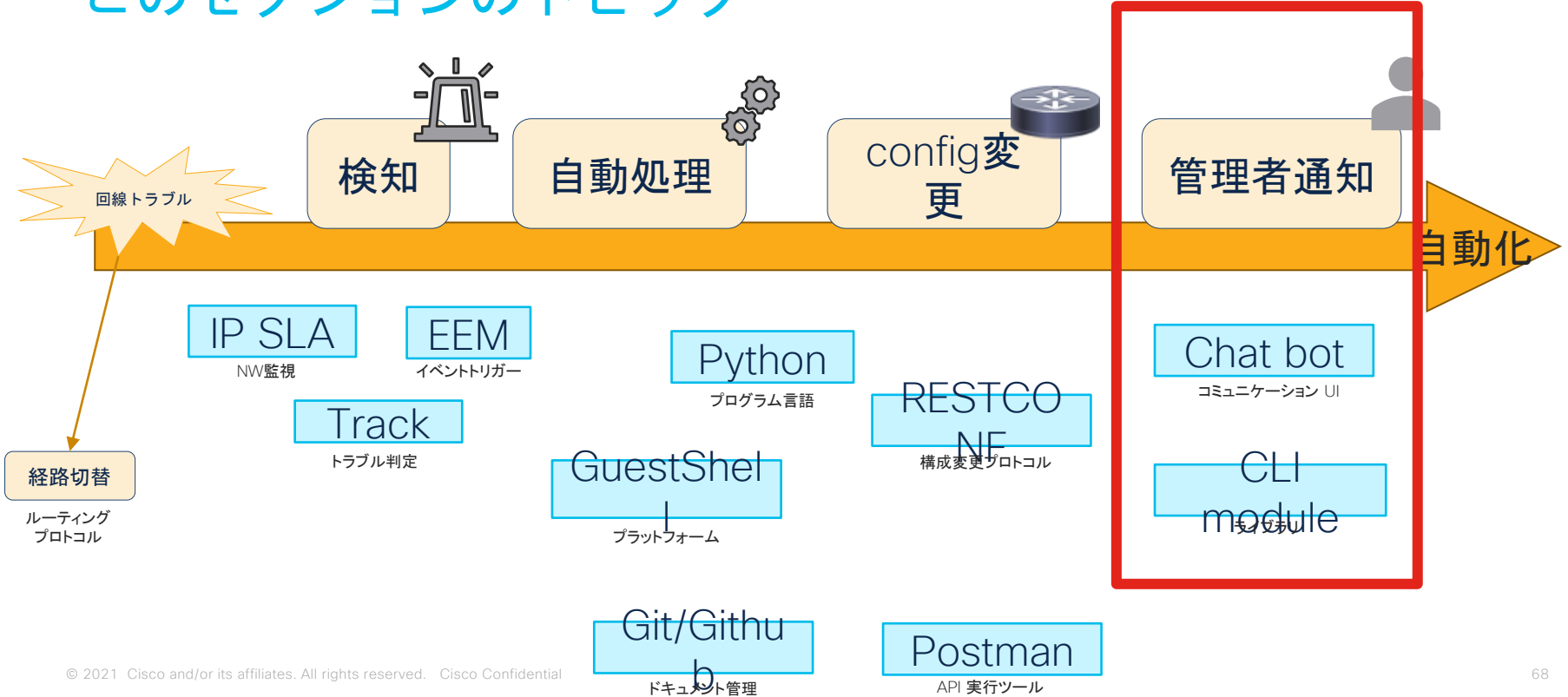
(おまけ)

# GuestShellからIOS-XEを操作

Pythonでshow runを取得して管理者  
に通知しよう

# 実現したいこと

## このセクションのトピック



# GuestShellからIOS-XEを操作

## Minimum 座学

- CLI ライブラリ
  - GuestShellのPythonがIOSコマンドを実行できる

```
import cli
```

```
cli.execute("<実行したいIOSコマンド>")
```

# 演 デモンストレーション

## GuestShellからIOS-XEを操作

- cli ライブラリをインポートする
- show runを取得する
- Webex Teamsにメッセージを送信する

# 最後に...

# まとめ

## デモンストレーション 振り返り

RESTCONF (Postman, Python)

GuestShell (app-hosting, Linux)

Git + GitHub

IP SLA + Track + EEM





# 今後のために (1/4)

## 機能拡張を検討するなら...

- Python エラーハンドリング
  - 基本文法、テクニック
    - try/except 例外処理
  - REST API 知識
    - HTTP 応答ステータスコード
- 周辺インフラ 構成変更
  - Ansible
    - NW負荷の高い処理をサスペンド
  - 会議システム
    - ビデオ画質 最大レートを制限

# 今後のために (2/4)

## 学習をするなら

- DevNet Sandbox
  - 無料で使える検証環境
  - <https://developer.cisco.com/site/sandbox/>
- DevNet Learning Labs
  - 学習用のコンテンツ (有志による日本語訳)
  - <https://developer.cisco.com/learning/labs/tags/Japanese>

# 今後のために (3/4)

## 学習をするなら

- Pythonプログラミング1 基本文法編
  - Pythonの基本文法や簡単なアプリケーション実装のための必須知識を習得
  - [https://www.trainocate.co.jp/reference/course\\_details.aspx?code=PRC0103G](https://www.trainocate.co.jp/reference/course_details.aspx?code=PRC0103G)
- REST APIによるインフラ管理実践
  - インフラ管理作業を自動化するためのテクニックである REST API を学習
  - [https://www.trainocate.co.jp/reference/course\\_details.aspx?code=NFC0455G](https://www.trainocate.co.jp/reference/course_details.aspx?code=NFC0455G)

# 今後のために (4/4)

## 学習をするなら

- Developing Applications and Automating Workflows using Cisco Core Platforms (DEVASC)
  - Ciscoのプラットフォームをベースとした基本的なネットワークアプリケーションの実装と自動化ワークフローを実装する方法の学習
  - [https://www.trainocate.co.jp/reference/course\\_details.aspx?code=CSC0434V](https://www.trainocate.co.jp/reference/course_details.aspx?code=CSC0434V)
- Implementing Automation for Cisco Enterprise Solutions (ENAI) v1
  - プログラマビリティと自動化をCiscoのエンタープライズキャンパスとWANに統合し、より効率的なワークフローと俊敏性の高いネットワークを作成
  - [https://www.trainocate.co.jp/reference/course\\_details.aspx?code=CSC0438V](https://www.trainocate.co.jp/reference/course_details.aspx?code=CSC0438V)
- 【来年リリース予定：CCIE対策コース】 Cisco High-End Lab BOOT CAMP 3

# Thank You

# Q&A

画面右側の Q&A ウィンドウから、  
すべてのパネリスト (All Panelists) 宛  
に送信してください。



# 次回の オンラインセミナー予定



## NCS5500 / NCS540 トラブルシューティング

2021 年 12 月 22 日 (水) 10:00 - 11:30

**藤代 大輝 (Taiki Fujishiro)**

シスコシステムズ

グローバルカスタマーエクスペリエンスセンター

テクニカル コンサルティング エンジニア



# [明日より] シスコ コミュニティ 特別キャンペーンが始まります！

シスコ コミュニティ  
Blue Thanks キャンペーン



仲間にThanksを送ろう！  
シスコ コミュニティ  
Blue Thanks キャンペーン

シスココミュニティへコンテンツを投稿したり、コミュニティメンバーの投稿にいいね！を送ることで感謝の気持ちを表しましょう！参加された方へ抽選でプレゼントを差し上げます。

詳細は以下をご覧ください

<https://community.cisco.com/t5/-/-/ba-p/4513538>





# 今冬も Community Helping Community プログラム実施中です！

あなたの行動が社会貢献に！

## Community Helping Community

コミュニティにおけるみなさんの活動 1 件ごとに、シスコが 1 ドルの寄付金を [Save the Children](#) に出資します。

さまざまな活動がカウントされます。得意な分野でご参加ください！寄付金が 1 万ドルになったら活動の募集を終了します。

詳細は以下をご覧ください

<https://community.cisco.com/t5/-/-/ba-p/4510284>



お役立ててください！

# シスココミュニティの歩き方

**ついに公開！**

**シスココミュニティの歩き方**

2020年2月版

利用方法や小技集など  
All in One 役立ちガイド

詳しくはこちら



<https://community.cisco.com/t5/-/-/ta-p/4021064>



ご参加ありがとうございました。

Community Liveと Cisco Communityの  
各アンケートにも ぜひ ご協力ください。

