

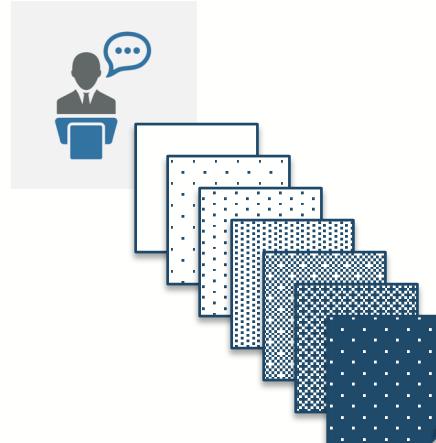
ネットワークCIパイプラインの構築

Shishio Tsuchiya

shtsuchi@arista.com

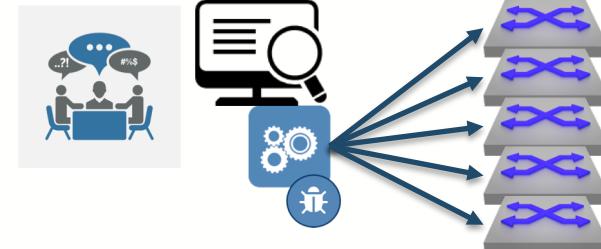
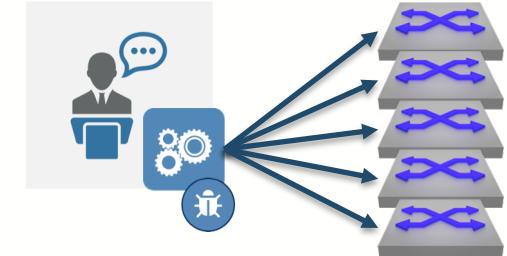
ネットワークCIの重要性

- CI/CDとはContinuous Integration／Continuous Deliveryの略で継続的なインテグレーションと継続的なデリバリー
- 自動化する前の取り組み
 - 作業ミスの防止
 - » 手順書に確認項目を追加
 - 運用や作業工数が肥大化



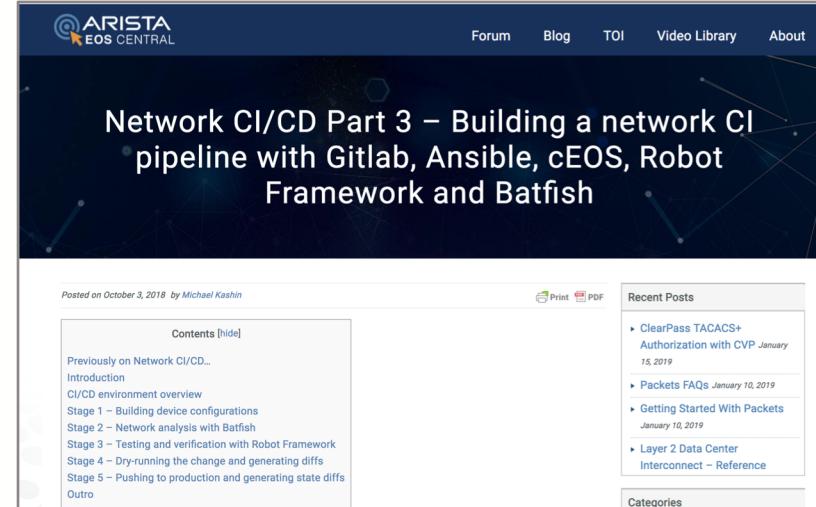
ネットワークCIの重要性

- ・自動化すればミスはなくなるのか？
 - 自動化コードのミス=>バグ
 - 自動化コードのテストが必要
- ・ソフトウェア業界の自動化の取り組み
 - 自動化テストも自動化
 - 繙続インテグレーション・パイプラインの構築
 - テスト自動化→テスト内容もコードに



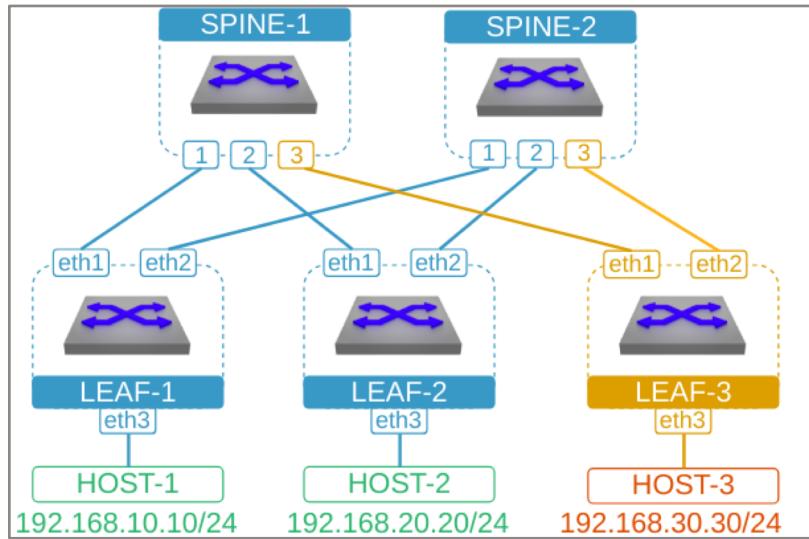
ネットワークCI/CD

- Network CI/CD Part 3 – Building a network CI pipeline with Gitlab, Ansible, cEOS, Robot Framework and Batfish
 - <https://eos.arista.com/network-ci-part-3/>
- こちらの環境を例に説明する



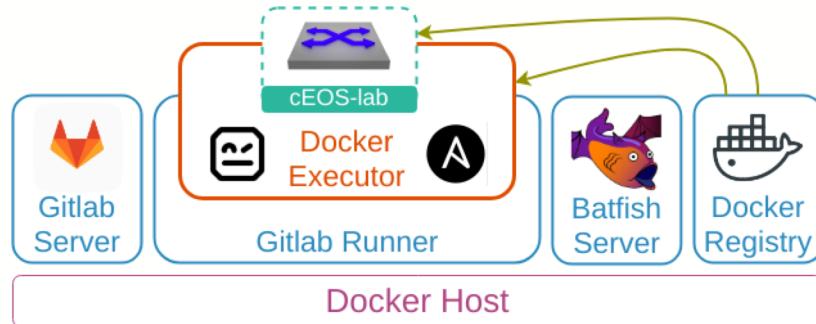
本セッションでのネットワーク構成

- L3 Closネットワーク
- cEOS-labで構築



本セッションでのCI環境

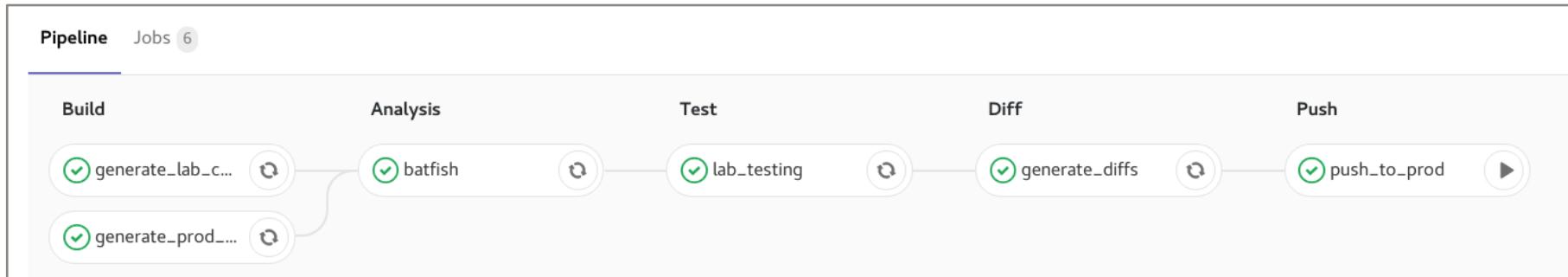
- Gitlab Server
 - Gitリポジトリの提供とCI/CDサーバとWeb UI
- Gitlab Runner
 - CI/CDパイプラインのワークノード。JobをGitlabサーバから受け取り、実行し、結果をGitlabサーバへ変更
- Batfishサーバでコンフィグの正常性を確認
- cEOS-lab、Robot FrameworkのDockerイメージをプライベートリポジトリへ登録



- 全てコンテナで作成
- CIパイプライン中の環境でもcEOS-lab x5で最大合計10個のcEOS-lab
- (参考: 4vCPU、16GBのCentOS)

CIパイプライン

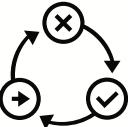
- YAMLでパイプラインを記述
- 5段階のステージ
 - Build: Ansibleによる機器コンフィグの自動作成(本番へは適用せず、オフラインでコンフィグ生成)
 - Analysis: Batfishを使ったルーティングの正常性確認
 - Test: cEOS-labとRobot Frameworkを使ったラボ環境での正常性確認
 - Diff: Ansibleで本番環境の現在のコンフィグと、適用するコンフィグの差分を確認
 - Push: Ansibleによる本番環境への設定投入



Ansibleとは

<https://www.ansible.com/>

- Red Hatのオープンソースの構成管理ツール
- 読みやすい
- タスク内容を記述するPlaybookはテキスト形式(YAML)
- エージェンスレス
- Chef/Puppetとは違いクライアントにエージェントが要らない
- 他のオープンソースとの連携が可能



Batfishとは

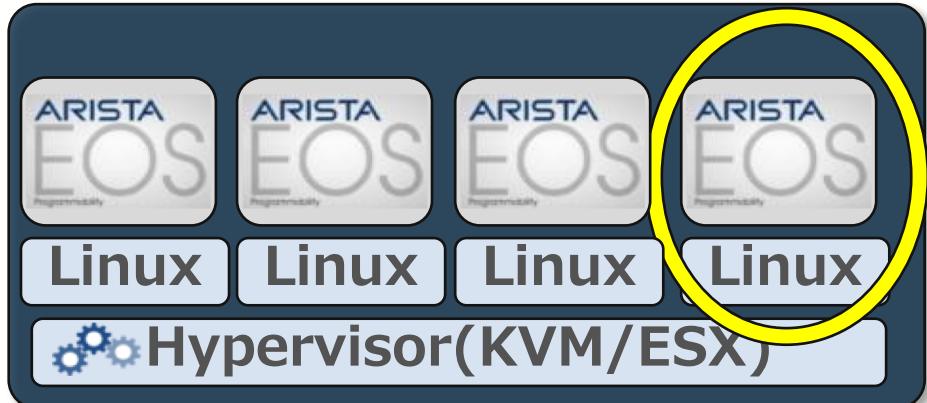
<https://www.batfish.org/>

- マルチベンダーのネットワーク検査ツール
 - コンフィグレーションの一貫性
 - 到達性
 - セキュリティ
 - 変更による解析
- これらをオフラインで実施する事が出来る
- Arista/Cisco/Juniper/Palo Alto/AWS/iptablesなど多くをサポート
- Microsoft Research, UCLA, USCなどにより開発、現在は INTENTIIONETによりサポート



vEOS-labとcEOS-lab

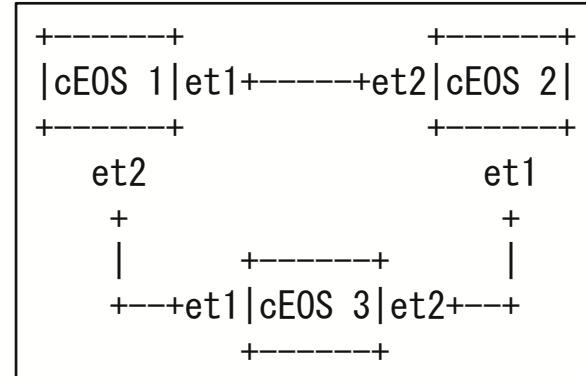
- vEOS-lab
 - KVM、ESX、Vbox、Vagrantへ対応
 - コンフィグ、コマンドの確認
 - 自動化の開発環境での活用
 - ソフトウェアのデータプレーンの為、性能は出ない
 - メモリ約1.5GB、推奨2GB
- cEOS-lab
 - vEOS-labのコンテナバージョン
 - 小さなリソースで稼働
 - メモリ約600MB
- どちらもユーザ登録のみで無料で使用可能



Docker network topology builder docker-topo

<https://github.com/networkop/docker-topo>

- コンテナ環境でのネットワーク構築ツール
- YAML形式で書かれた設定ファイルでトポロジーを作成/コンフィグ指定や保存が可能
 - vEOS-lab
 - cEOS-lab
 - CVP
- vrnetlabによる各社VMイメージ
 - <https://github.com/plajjan/vrnetlab>
 - CRS1K
 - vMX
 - XRv *動作確認済



links:

- endpoints: ["cEOS-1:eth1", "cEOS-2:eth2"]
- endpoints: ["cEOS-1:eth2", "cEOS-3:eth1"]
- endpoints: ["cEOS-2:eth1", "cEOS-3:eth2"]

VERSION: 2

driver: veth

PREFIX: 3-node

CONF_DIR: ./config

CEOS_IMAGE: ceosimage:latest

PUBLISH_BASE: 9000

Robot Framework

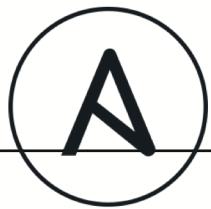
<https://robotframework.org/>

- Robot Frameworkは受け入れ試験/受け入れテスト駆動開発(ATDD)/ロボットプロセスオートメーション(RPA)で使われるオープンソース
- 簡単なSyntaxと複数のフォーマット
 - プレーンテキスト
 - reStructuredText(reST) など
- ライブラリはPython、Javaで作成可能
- レポートとログは、HTMLで出力
- データ・ドリブンのテストケースに対応
- 低いオーバーヘッド=>実行時間は早い
- プログラムやスクリプトの得意でない人にも、使いやすい



Arista Robot Frameworkライブラリ	
Library	Introduction
Connect To	スイッチにeAPIで接続
Get Running Config	指定したデバイスのrunning-configを取得
List Extensions	特定のデバイスにインストールしているExtentionのリストを取得
Configure	コンフィグモードから指定したコマンドを設定
Run Cmds	特定のデバイスで、eAPIコマンドを実行

Build:Ansibleによる機器コンフィグの自動作成



- Spinex2.Leafx3 5台分のコンフィグを作成

```
PLAY [clos] *****
TASK [ipam : Assign IP address information from IPAM] *****
ok: [Spine-1]

TASK [build : Generate configuration from templates] *****
changed: [Spine-1]
changed: [Spine-2]
changed: [Leaf-1]
changed: [Leaf-3]
changed: [Leaf-2]

PLAY RECAP *****
Leaf-1 : ok=1 changed=1 unreachable=0 failed=0
Leaf-2 : ok=1 changed=1 unreachable=0 failed=0
Leaf-3 : ok=1 changed=1 unreachable=0 failed=0
Spine-1 : ok=2 changed=1 unreachable=0 failed=0
Spine-2 : ok=1 changed=1 unreachable=0 failed=0

Uploading artifacts...
./outputs: found 7 matching files
Uploading artifacts to coordinator... ok id=7 responseStatus=201 Created token=nBaTP7Fp
Job succeeded
```

Pipeline Jobs 6

Build

generate_lab_c...

generate_prod...

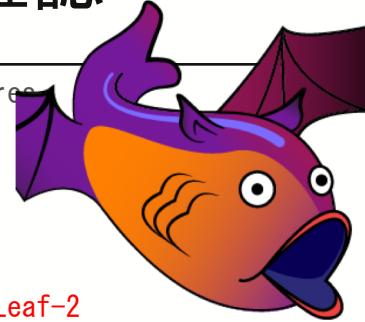
Analysis

batfis

Analysis: Batishを使ったルーティングの正常性確認

- Leaf3の検査
 - コントロールプレーン
 - » BGPピアが設定されているか
 - データプレーン
 - » Leaf1/Lea2のホップ数が2か
 - 障害時の動作
 - » パスが1つでホップ数も同等か

```
INFO: Progress: Searching for unused and undefined data structures
INFO: No undefined data structures found
INFO: No unused data structures found
INFO: Progress: Analyzing control plane properties
INFO: All leaves have at least one peering with each spine
INFO: Leaves only peer with spines
INFO: Progress: Analyzing traceroute from Leaf-3 to Leaf-1 and Leaf-2
INFO: Traceroute Progress: ACCEPTED
INFO: Number of paths 2 == 2 number of spines
INFO: Traceroute traversed exactly two hops
INFO: Traceroute traversed exactly two hops
INFO: Traceroute Progress: ACCEPTED
INFO: Number of paths 2 == 2 number of spines
INFO: Traceroute traversed exactly two hops
INFO: Traceroute traversed exactly two hops
INFO:
Progress: analysing failure conditions
INFO: Progress: Analyzing traceroute from Leaf-3 to Leaf-1 and Leaf-2
INFO: Traceroute Progress: ACCEPTED
INFO: Number of paths 1 == 1 number of spines
INFO: Traceroute traversed exactly two hops
INFO: Traceroute Progress: ACCEPTED
INFO: Number of paths 1 == 1 number of spines
INFO: Traceroute traversed exactly two hops
```



Analysis

batfish

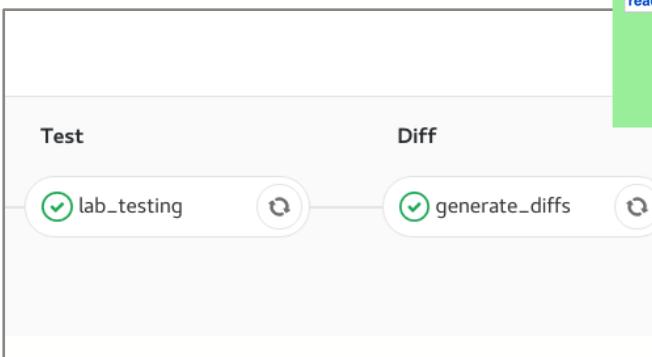
Test

✓

Test: cEOS-labとRobot Frameworkを使ったラボ環境での正常性確認

- Batfishで確認出来ないものをcEOS-labとRobot Frameworkで確認
- BGP/Linux端末からのPing

Test Details							LOG			
Totals	Tags	Suites	Search							
Type:	<input checked="" type="radio"/> Critical Tests <input type="radio"/> All Tests									
Status:	6 total, 6 passed, 0 failed									
Total Time:	00:00:08.311						Elapsed	Start / End		
Name	Documentation	Tags	Crit.	Status	Message					
Combined Validation Report. Leaf-3 dataplane. BGP Session	Check if BGP peerings are Established		yes	PASS			00:00:00.761	20190723 10:10:32.343	20190723 10:10:33.104	
Combined Validation Report. Leaf-3 dataplane. BGP Updates	Check if loopbacks are known via BGP		yes	PASS			00:00:01.011	20190723 10:10:33.105	20190723 10:10:34.116	
Combined Validation Report. Leaf-3 dataplane. Control Plane Snapshot	Capture the current state of BGP RIB		yes	PASS			00:00:01.006	20190723 10:10:34.117	20190723 10:10:35.123	
Combined Validation Report. Leaf-3 dataplane. E2E connectivity	Check if Host-3 can reach other hosts		yes	PASS			00:00:05.392	20190723 10:10:35.309	20190723 10:10:40.701	
Combined Validation Report. Leaf-3 dataplane. Remote leaves reachability Leaf1	Leaf1's SVIs Ping loss < 100%		yes	PASS			00:00:00.073	20190723 10:10:35.168	20190723 10:10:35.241	
Combined Validation Report. Leaf-3 dataplane. Remote leaves reachability Leaf2	Leaf2's SVIs Ping loss < 100%		yes	PASS			00:00:00.068	20190723 10:10:35.241	20190723 10:10:35.309	

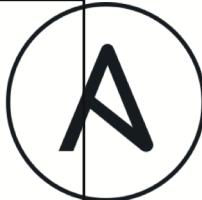


Diff: Ansibleで本番環境の現在のコンフィグと、適用するコンフィグの差分を確認

- 変更前の差分を確認。
- 差分はファイルに保存

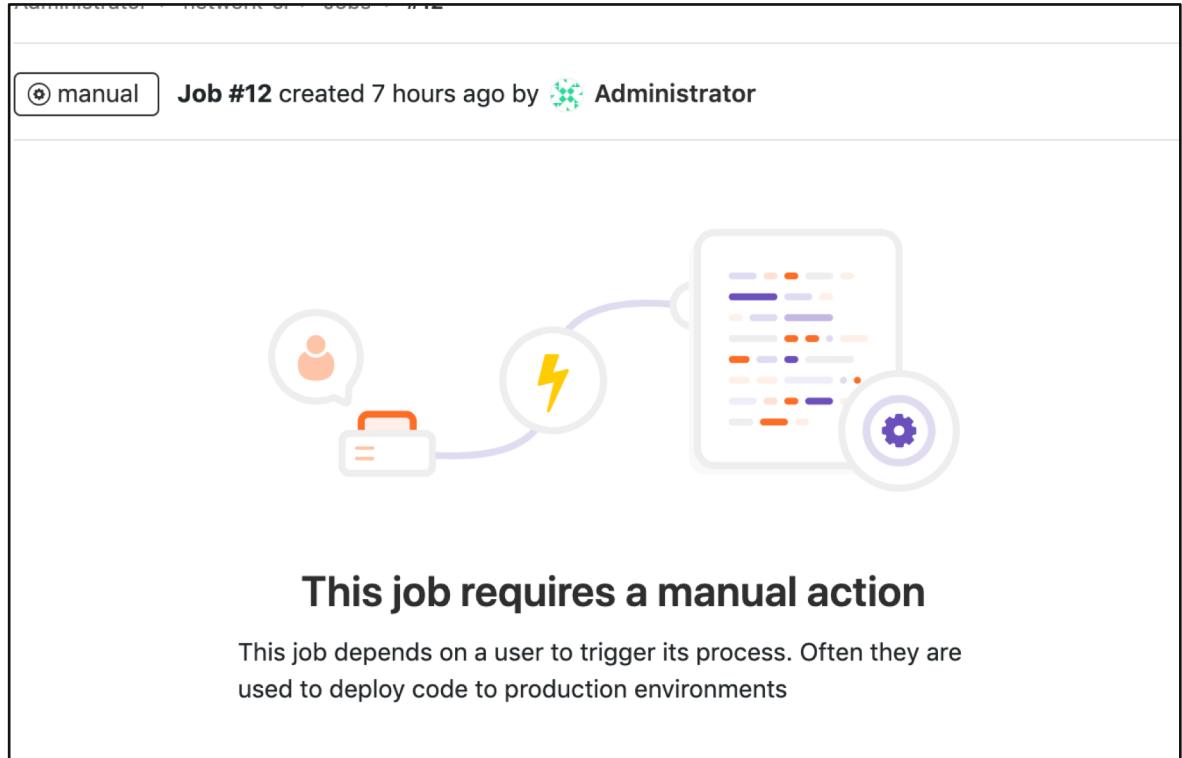
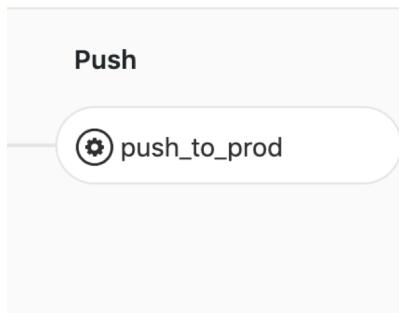
```
$ ansible-playbook --diff --check -e PROD_IP=$PRIMARY_IP -e buildenv=$BUILDENV -e outpath=$CI_PROJECT_DIR/outputs/diffs -e confdir=$CONF_DIR diff.yml

PLAY [clos] *****
TASK [Push new configuration] *****
--- system:/running-config
+++ session:/ansible_1563844259-session-config
@@ -10,6 +10,8 @@
!
username ***** privilege 15 secret sha512
$6$H9hv1.QIXjv.WbOI$ghbukD6MpBy.ONKkImqSv3tCA5Wz1FOeXgTbgw1QCRa8EWSiMRY5xrtzs57FvZXudZI/J3/NJwkBXLsFDG7gu0
!
+vlan 30
+!
interface Ethernet1
description Spine-1:Ethernet3:10.0.254.4/31
no switchport
@@ -21,8 +23,31 @@
ip address 10.0.255.5/31
!
```



Push: Ansibleによる本番環境への設定投入

- 本番環境の実施はマニュアルにて行う



まとめ

- Gitlab/Batfish/cEOS-lab/docker-topo/Robot Frameworkを用いてネットワークCI/CDパイプラインの手法を共有した
- 実際の構築/検証にて使われている手法と比較議論がしたい

Thank You

www.arista.com