

NOKIA

containerlabで叶える ネットワークエンジニアの “あるある願望”

ノキアソリューションズ&ネットワークス
小川 怜 (Ogawa Satoshi)



1. 本発表について

自己紹介

- 名前: 小川 怜 (Ogawa Satoshi)
- 所属: ノキアソリューションズ&ネットワークス合同会社
IP製品事業部 ソリューションアーキテクト
- 過去の発表:
 - JANOG26 : IPv6時代のIPv4を考える
 - ENOG76 : クイックスタートcontainerlab
 - JANOG53 : gNMIcを活用したマルチベンダー環境でのテレメトリ技術の実践
- ✓ email : satoshi.ogawa@nokia.com
- ✓ Slack(janog-meeting) : @Satoshi Ogawa
- ✓ Twitter : saogawa



ぜひ率直なご感想・ご意見をいただけると嬉しいです

1. 本発表について

発表内容

1. 概要
2. containerlab(clab)とは？
3. clab(と関連ツール)で実現するNWエンジニアのあるある願望
4. デモンストレーション
5. まとめ・議論

1. 本発表について

発表内容

1. 概要

2. containerlab(clab)とは？

3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

4. デモンストレーション

5. まとめ・議論

第1版

当時...



1. 概要・背景

「試したい」が「すぐできる」に変わる時代

- ラボ構築は、もはや“面倒な作業”ではありません
- containerlabと周辺ツールの進化によって、NWラボの「できたらいいな」が、現実になる
- “とりあえずやりたい時にやってみる”が、一番の学びになる
- 若手エンジニアが“触れない壁”を壊す鍵、それがcontainerlab

containerlab によって広がる仮想ラボの世界と、その具体的な実現手法をご紹介します

1. 本発表について

発表内容

1. 概要

2. containerlab(clab)とは？

3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

4. デモンストレーション

5. まとめ・議論

2. containerlabとは？

概要

特徴と主要メリット

- マルチベンダー & オープンソース
Arista、Cisco、Juniper、Nokia、SONiC、Linuxなど業界主要OSをサポート
- 軽量高速機能
コンテナベースでわずか数秒で環境構築
- 使い捨て・自動再現
環境の破棄と再構築・共有（GitHubとの親和性大）が容易
- IaC(コードとしてのインフラ)
YAMLでトポロジーを定義
- コンテナ版&VM版NOSをサポート
Vrnetlab統合によりVM型OSも同一フレームワークで実行可能



他のツール群との組み合わせによる相乗効果も

2. containerlabとは？

仮想ラボ構築までの流れ

1. Docker準備 → containerlabインストール

シングルシェルスクリプトで非常に簡単インストール

2. 必要なNOSイメージを準備

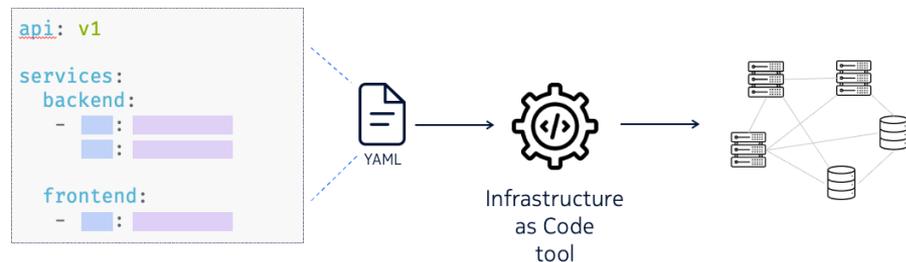
Arista, Cisco, Juniper, Nokiaなどのベンダーサイトから入手

3. トポロジフィアルをYAML形式で作成

4. deployコマンドで即起動

数秒で完全なネットワーク環境が利用可能に

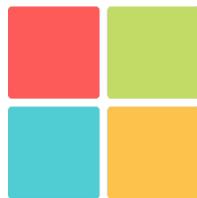
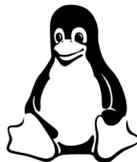
5. SSH, CLI, API等で実機同様に操作



2. containerlabとは？

仮想ラボ構築までの流れ：インストール

```
curl -L http://containerlab.dev/setup | sudo bash -s "all"
```



CONTAINERlab

- <https://containerlab.dev/install/>

最新Docker/GitHub CLI/containerlab がシングルラインシェルで完了！

2. containerlabとは？

仮想ラボ構築までの流れ：トポロジーYAML



トポロジーYAMLファイル例:

```
1  name: srl02
2
3  topology:
4    nodes:
5      srl1:
6        kind: nokia_srlinux
7        image: ghcr.io/nokia/srlinux:23.7.1
8      srl2:
9        kind: nokia_srlinux
10       image: ghcr.io/nokia/srlinux:23.7.1
11
12     links:
13       - endpoints: ["srl1:e1-1", "srl2:e1-1"]
14
15
```

2. containerlabとは？

仮想ラボ構築までの流れ：デプロイ

```
zeno@srexperits:~/srexperits$
```

1. 本発表について

発表内容

1. 概要

2. containerlab(clab)とは？

3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

4. デモンストレーション

5. まとめ・議論

3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

願望リスト

3-1. 仮想ラボをもっと身近に。OS環境に縛られず、すぐ試せるフェーズへ

containerlab × WSL : Windows, MAC Appleシリコン, CodeSpaceへの環境での仮想ラボ構築

3-2. 動かして、確かめて、理解する : 触りたいラボに即時アクセス

containerlab × GitHubレポジトリからYAMLファイル読み込み

3-3. 動かして、確かめて、理解する : k8sデータセンターネットワーク

containerlab × Kubernetes/KIND連携 : 最小構成でNW + K8s動作確認

3-4. 「今このこのこれを見たい」に応える進化するキャプチャ環境

containerlab × EdgeShark

3-5. “この1台じゃ足りない”を超える。手持ちサーバをつなげて広がる検証の可能性

containerlab × clabernetes : 手持ちサーバをクラスタ化して仮想NWラボ構築

3-6. コードとラボが同居する。効率も理解もスピードアップ。すべては一つの画面で

containerlab × VS-Code Extension : GUIベースでラボ設計・操作、CLIも1画面で

3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

3-1. 仮想ラボをもっと身近に。OS環境に縛られず、すぐ試せるフェーズへ

containerlab × WSL : Windows, MAC Appleシリコン, CodeSpaceへの環境での仮想ラボ構築

こうゆうことはありませんか？

- 「自宅のWindowsマシンでネットワークラボを試したいけど、Linux前提のツールばかりで動かない…」
- 「Mac Appleシリコンに対応してる？ 仮想化ハードル高そう…」
- 「ラボを作りたいだけなのに、OS設定やVM準備で心が折れる…」

解決策 = containerlab x 柔軟な実行環境

- containerlabは、Dev Containers(WSLやMac Intel/Mxチップ)さらには**GitHub Codespacesでも動作可能。**
- クラウドやローカル環境に縛られず、**どこでも、すぐに仮想ラボが構築**できます。
- 「試したい」のハードルが劇的に下がり、**日常PCで商用ネットワークのような検証環境**を手軽に再現可能です。



3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

3-1. 仮想ラボをもっと身近に。OS環境に縛られず、すぐ試せるフェーズへ

containerlab × WSL : Windows, MAC Appleシリコン, CodeSpaceへの環境での仮想ラボ構築

「GitHub Codespace」って何？

クラウド上で開発環境を提供されるサービス

- **即時アクセス** : 数秒で完全に設定された開発環境(例:マイVS-Code)にアクセス可能
- **高いカスタマイズ性** : 好みの設定や拡張機能を簡単に導入可能
- **どこでも開発作業可能** : インターネット接続があれば、どこからでも作業が可能

ローカルマシンに依存せず、開発環境を迅速に構築可

3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

3-1. 仮想ラボをもっと身近に。OS環境に縛られず、すぐ試せるフェーズへ

containerlab × WSL : Windows, MAC Appleシリコン, CodeSpaceへの環境での仮想ラボ構築

「GitHub Codespace」って何？

NW屋さんの表現

クラウド上で開発環境を提供されるサービス

- **即時アクセス** : 数秒で完全に設定された **NWラボ** 環境にアクセス可能。
- **高いカスタマイズ性** : 好みの **トポロジー** や **containerlab** 拡張機能を簡単に導入可能。
- **どこでもCLI作業可能** : インターネット接続があれば、**ブラウザで** どこからでも **CLI作業** が可能

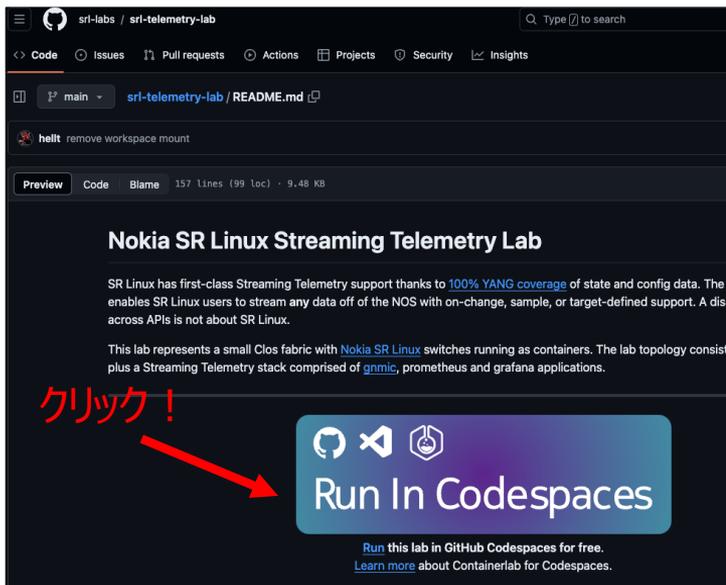
ローカルマシンに依存せず、**NWラボ** を迅速に構築可

3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

3-1. 仮想ラボをもっと身近に。OS環境に縛られず、すぐ試せるフェーズへ

containerlab × WSL : Windows, MAC Appleシリコン, CodeSpaceへの環境での仮想ラボ構築

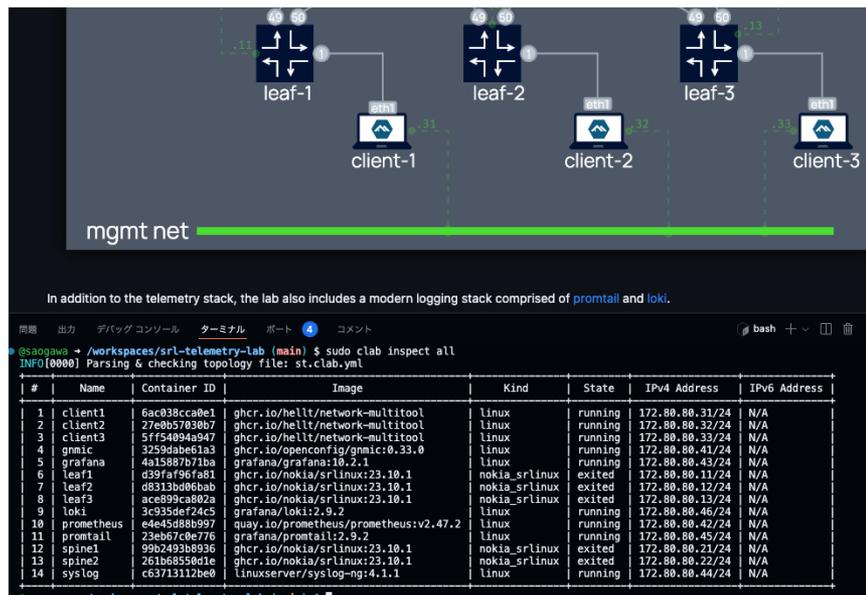
GitHub Codespace対応 - 1クリックでラボ構築！(ブラウザのみで利用可能)



クリック！

Run In Codespaces

Run this lab in GitHub Codespaces for free.
[Learn more about Containerlab for Codespaces.](#)



In addition to the telemetry stack, the lab also includes a modern logging stack comprised of `promtail` and `loki`.

```
bash
```

#	Name	Container ID	Image	Kind	State	IPv4 Address	IPv6 Address
1	client1	6ac038cca0e1	ghcr.io/hellt/network-multitool	linux	running	172.80.80.31/24	N/A
2	client2	27e0b57030b7	ghcr.io/hellt/network-multitool	linux	running	172.80.80.32/24	N/A
3	client3	5ff54094a947	ghcr.io/hellt/network-multitool	linux	running	172.80.80.33/24	N/A
4	gnmic	3259da0ee1a3	ghcr.io/openconfig/gnmic:v0.33.0	linux	running	172.80.80.41/24	N/A
5	grafana	4a15887b71ba	grafana/grafana:10.2.1	linux	running	172.80.80.43/24	N/A
6	leaf1	d39fa96fa81	ghcr.io/nokia/srlinux:23.10.1	nokia_srlinux	exited	172.80.80.11/24	N/A
7	leaf2	d8313bd06bab	ghcr.io/nokia/srlinux:23.10.1	nokia_srlinux	exited	172.80.80.12/24	N/A
8	leaf3	ace899ca802a	ghcr.io/nokia/srlinux:23.10.1	nokia_srlinux	exited	172.80.80.13/24	N/A
9	loki	3c935def74c5	grafana/loki:2.9.2	linux	running	172.80.80.46/24	N/A
10	prometheus	e4e45d88b997	quay.io/prometheus/prometheus:v2.47.2	linux	running	172.80.80.42/24	N/A
11	promtail	23eb67c0e776	grafana/promtail:2.9.2	linux	running	172.80.80.45/24	N/A
12	spine1	9902493b8936	ghcr.io/nokia/srlinux:23.10.1	nokia_srlinux	exited	172.80.80.21/24	N/A
13	spine2	2619e8559e1e	ghcr.io/nokia/srlinux:23.10.1	nokia_srlinux	exited	172.80.80.22/24	N/A
14	syslog	c63713112be0	linuxserver/syslog-ng:4.1.1	linux	running	172.80.80.44/24	N/A

3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

3-2. 動かして、確かめて、理解する：触りたいラボに即時アクセス

containerlab × GitHub：ラボ共有・仮想ラボの1クリック起動

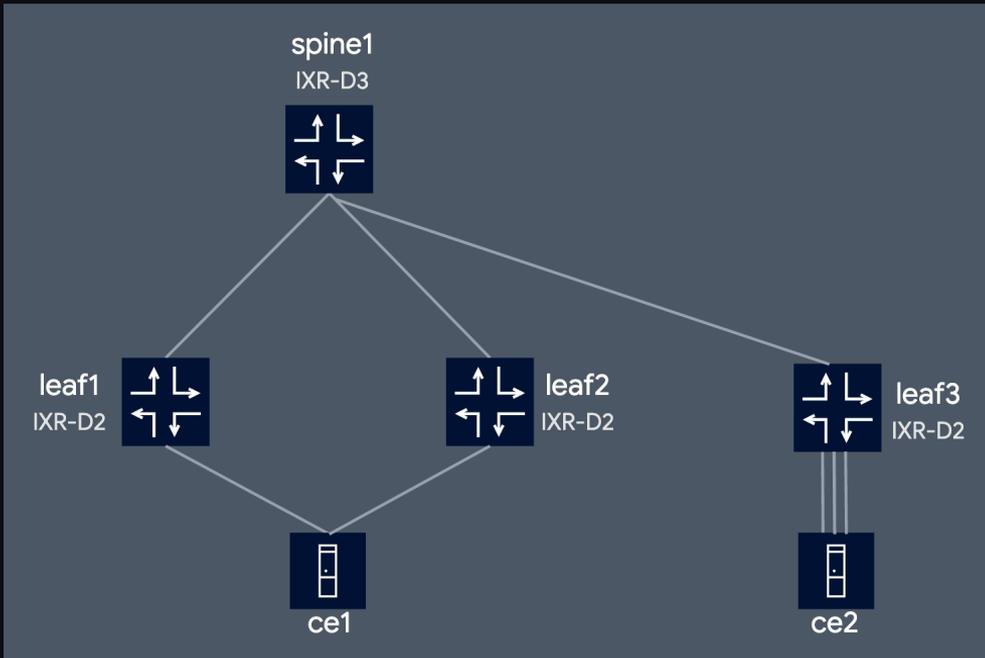
こうゆうことありませんか？

- 「試してみたい構成が載ってるけど、手元に再現するのに時間がかかる…」
- 「Topology YAMLファイルを手でコピーして、編集して、やっと起動…」「おっと関連していたスクリプトが..)」
- 「検証や勉強を始める前に、環境準備だけで疲れてしまう」

解決策 = containerlab x GitHub

- containerlabは、GitHub上のレポジトリにある `.clab.yml` ファイルをそのまま読み込んで即ラボ起動できます。
- `containerlab deploy --topo <GitHubのURL>` という1行で、環境構築が完了。
- **触りたいラボに、すぐ触れる**、ラボ環境へのと検証の距離が一気に縮まります。

The topology comprises a spine, three leaf(PEs) routers, and two Alpine Linux hosts(CEs). A multi-homed CE is connected to leaf1, while another is linked to leaf3 for testing purposes.



 **aaakpinar** Alperen

 **hellt** Roman Dodin

Languages

 100.0%



3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

3-2. 動かして、確かめて、理解する：触りたいラボに即時アクセス

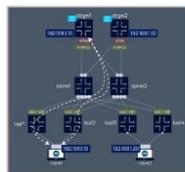
containerlab × GitHub：ラボ共有・仮想ラボの1クリック起動

GitHubで公開されている仮想ラボ環境

<https://github.com/topics/clab-topo>

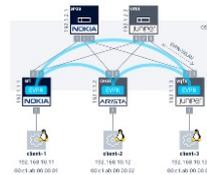
The screenshot shows the GitHub 'Containerlab topology' topic page. It features a search bar with the topic name, a description of the topic, and a list of repositories. The top repository is 'srl-labs / srl-telemetry-lab', which is starred 174 times. Below it, 'srl-labs / netbox-nrx-clab' is listed with 43 stars. The page also includes navigation tabs for Explore, Topics, Trending, Collections, Events, and GitHub Sponsors.

Nokia EVPN Interop



[srl-labs/nokia-evpn-lab](https://github.com/srl-labs/nokia-evpn-lab)

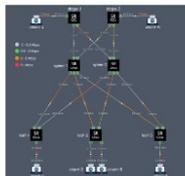
Multivendor EVPN



[srl-labs/multivendor-evpn-lab](https://github.com/srl-labs/multivendor-evpn-lab)

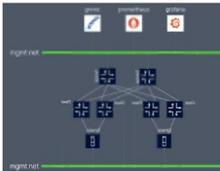


SR Linux & SROS Telemetry



[srl-labs/srl-sros-telemetry-lab](https://github.com/srl-labs/srl-sros-telemetry-lab)

SR Linux Oper-Group



[srl-labs/opergroup-lab](https://github.com/srl-labs/opergroup-lab)

3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

3-3. 動かして、確かめて、理解する：k8sデータセンターネットワーク

containerlab × Kubernetes/KIND連携：最小構成でNW + K8s動作確認

こうゆうことありませんか？

- 「Kubernetesとネットワークのつながりを勉強したいが、構築が重すぎる…」
- 「手元でBGPやEVPNとKubernetesを同時に動かす方法が分からない」
- 「クラウドネイティブの資料は読むけど、“実際どうつながってるか”がつかめない」

解決策 = containerlab x K8s/KIND連携

- containerlabは、Kubernetesクラスタ（KIND）を1つの仮想NW環境に構築することが可能です。
- **コンテナルータ+K8sを1台の端末で動かせる軽量検証環境**を構築できます。

3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

3-3. 動かして、確かめて、理解する：k8sデータセンターネットワーク

containerlab × Kubernetes/KIND連携：最小構成でNW + K8s動作確認

そもそも「k8s-kind」って何？

- kind = Kubernetes IN Docker の略。Docker コンテナの中で K8s ノードを動かせるオープンソースプロジェクト
- k8s-kind は、「Kubernetes のテスト用ノード」を containerlab 上に作るための仕組み

何が嬉しいんだっけ？

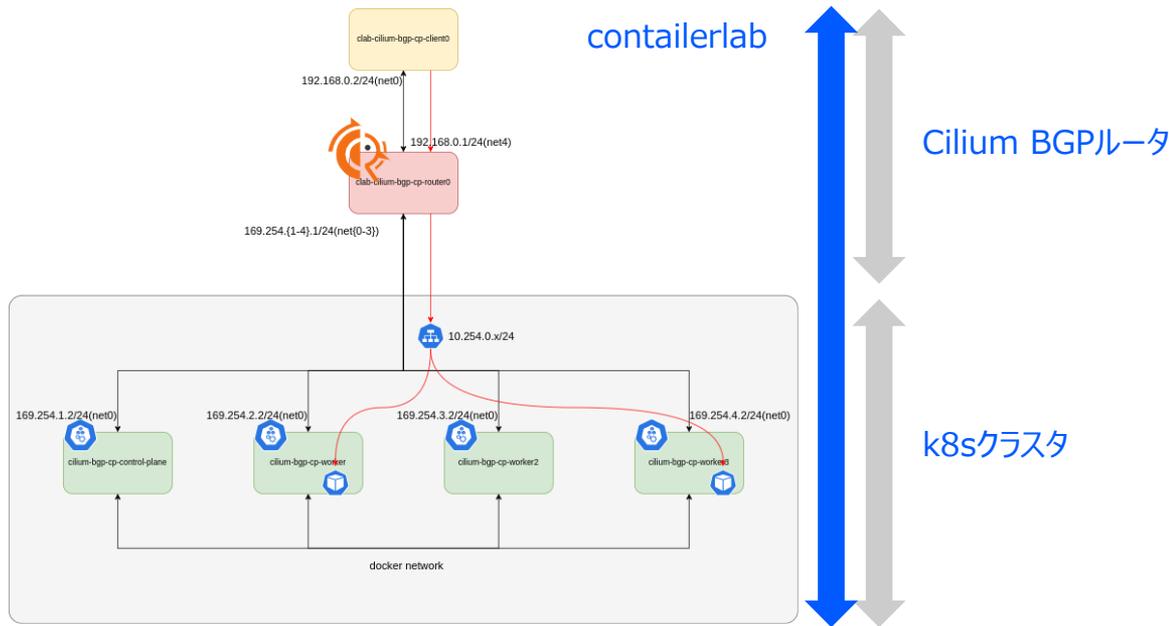
- SR Linux + Kubernetes の統合検証ができる
- クラウドネイティブネットワークファンクション（CNF）の通信をシミュレーションできる
- Kubernetes ベースのアプリがネットワーク機器とどう通信するかを見れる
→ k8s-kindで「Kubernetesクラスタの構築コスト(インストール手順の学習、構築時間の手間)を最小化」

```
name: sr1-k8s-lab
topology:
  nodes:
    sr11:
      kind: sr1
      image: ghcr.io/nokia/srlinux:23.10.1
    kube1:
      kind: k8s-kind
      image: kindest/node:v1.27.3
```

3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

3-3. 動かして、確かめて、理解する：k8sデータセンターネットワーク

containerlab × Kubernetes/KIND連携：最小構成でNW + K8s動作確認



引用元: KindとcontainerlabでCiliumのBGP Control Planeを試す
<https://terassyi.net/posts/2023/09/04/try-cilium-bgp-cp.html>

3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

3-4. 「今このこれを見たい」に応える進化するキャプチャー環境

containerlab × EdgeShark

こうゆうことありませんか？

- 「このタイミングのこのリンクだけ見たい！リアルタイムに！」
- 「仮想環境でパケットキャプチャするのに、何重ものミラーリング設定がめんどい」
- 「ちょっと見るのにpcapのファイル操作とかしたくない」

解決策 = containerlab × Edgeshark (エッジシャーク)

- EdgeSharkは、コンテナ間のLinuxネットワーク名前空間接続における通信を、リアルタイムかつGUIで可視化できるオープンソースのパケットキャプチャツール
- ContainerlabもNW名前空間でリンク接続しているため相性がよい
- CLIや複雑な設定不要で、“見たいところをクリック”するだけで使い慣れたWiresharkでリアルタイムにキャプチャー可能

初期(といっても1年前くらい?)のcontainerlabキャプチャー方法

```
ssh $containerlab_host_address ¥  
    "ip netns exec $lab_node_name tcpdump -U -nni $if_name -w -" | wireshark -k -i -
```

3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

3-4. 「今このこれを見たい」に応える進化するキャプチャー環境

containerlab × EdgeShark

The screenshot shows a web browser window with the URL `learn.srlinux.dev/blog/2024/vlans-on-sr-linux/`. The main content is a network diagram illustrating a setup with two clients and two switches. The diagram shows:

- `client1` connected to `sr1` (Switch 1) via interface `1`.
- `sr1` connected to `sr2` (Switch 2) via interface `10`.
- `sr2` connected to `client2` via interface `1`.

Below the diagram, there is a video feed of a man in a blue shirt, likely the presenter. The browser's address bar and various icons are visible at the top.

3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

3-5. “この1台じゃ足りない”を超える。手持ちサーバをつなげて広がる検証の可能性

containerlab x clabernetes(クラバネティーズ)：手持ちサーバをクラスタ化して仮想NWラボ構築

こうゆうことありませんか？

- 「1台のノートPCでは、ルータ数を増やすとCPUやメモリが足りない」
- 「大規模構成を試したくても、環境をスケールさせる方法がわからない」
- 「古いサーバが何台かあるけど、活用方法がない…」

解決策 = containerlab x clabernetes (クラバネティーズ)

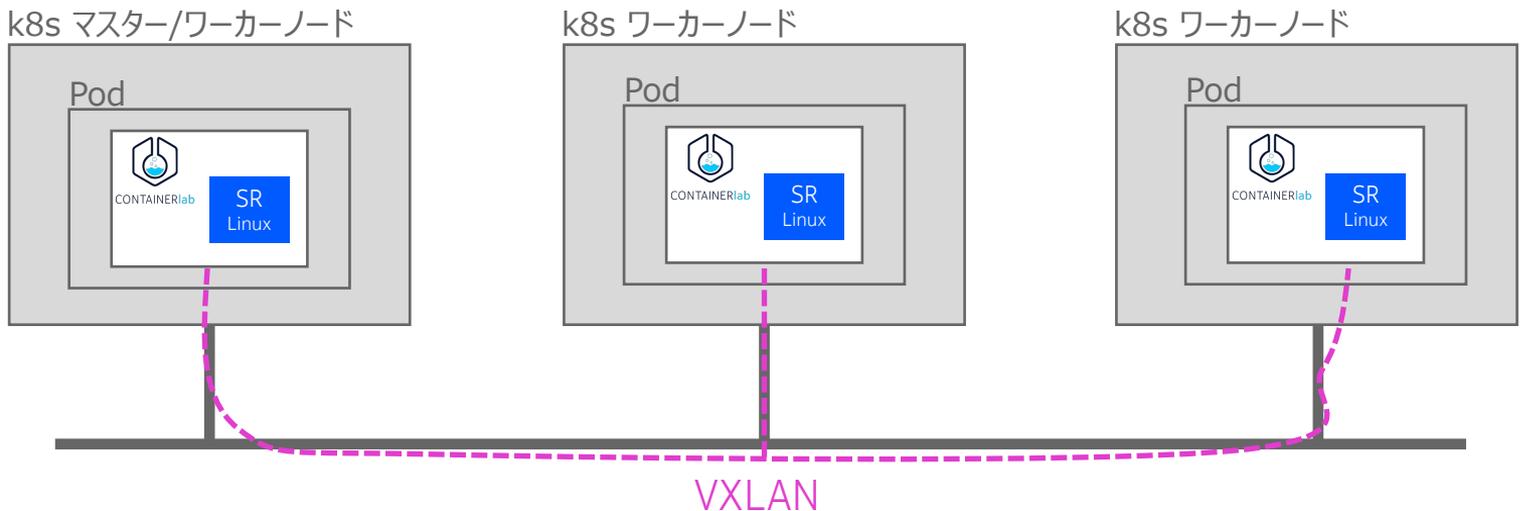
- clabernetes は、k8sクラスタ上に containerlabノードを分散して展開できるオープンソースプロジェクトです。
- これにより、1台の物理 PC やサーバーのリソースを超える規模のラボを構築する際に、複数台のサーバーを活用して仮想ネットワークラボをクラスタ化できます。

3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

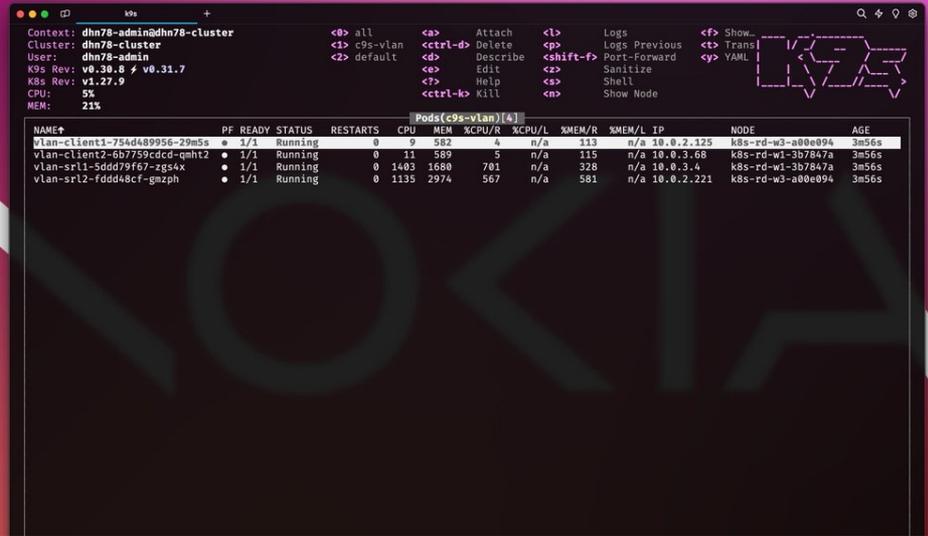
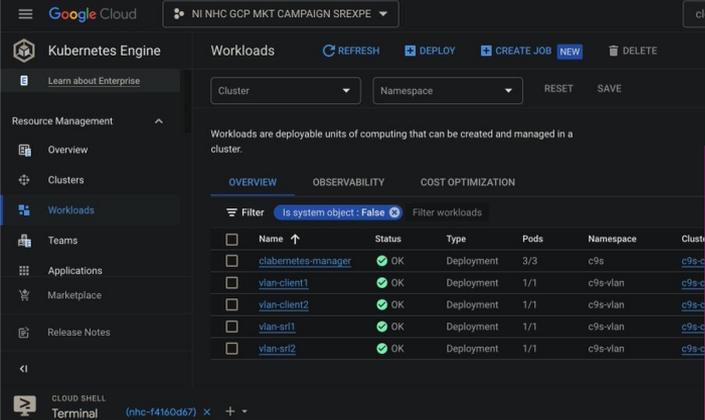
3-5. “この1台じゃ足りない”を超える。手持ちサーバをつなげて広がる検証の可能性

containerlab x clabernetes(クラバネティーズ)：手持ちサーバをクラスタ化して仮想NWラボ構築

動作イメージ = containerlabと同じYAMLトポロジーでデプロイ



containerlabとの機能互換性を高めるため 1ポッド=1clab=1ノード ※現実装



\$ kubectl get pods --namespace c9s-vlan -o wide

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
client1-5c4698f68c-v4z2n	1/1	Running	0	19m	10.244.1.15	c9s-worker	<none>	<none>
client2-6dfc49bc8f-hpkd4	1/1	Running	0	19m	10.244.2.15	c9s-worker2	<none>	<none>
srl1-78bdc85795-l9bl4	1/1	Running	0	19m	10.244.1.14	c9s-worker	<none>	<none>
srl2-7fffcd79-vxfn9	1/1	Running	0	19m	10.244.2.16	c9s-worker2	<none>	<none>



3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

3-6. コードとラボが同居する。効率も理解もスピードアップ。すべては一つの画面で
containerlab × VS-Code Extension : GUIベースでラボ設計・操作、CLIも1画面で

こうゆうことありませんか？

- 「YAMLでラボを書いて、別端末でCLI叩いて、別のツールでトポロジ図…分かれてて面倒」
- 「トポロジを変更 → 再デプロイ → SSHで入って確認…効率悪い」
- 「CLI操作時には、YAMLトポロジによる記述ではなく、ノード、リンク、視覚的かつ直感的に操作したい」

解決策 = containerlab × VS-Code Extension

- containerlabのVS Code Extensionを使えば、ラボ構成の作成・編集・可視化・操作まですべてを1画面で完結
 - YAMLファイルの作成補完（スキーマ支援）
 - ノードへの一括SSH接続や
 - EdgeSharkによるキャプチャー環境
 - TopoView による視覚的編集・確認
- 開発と検証を同じ場所で行えるため、自動化ツール(python/Ansible等)の開発・修正も圧倒的に速くなります。

3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

3-6. コードとラボが同居する。効率も理解もスピードアップ。すべては一つの画面で
containerlab × VS-Code Extension : GUIベースでラボ設計・操作、CLIも1画面で

そもそも「VS-Code」って何？

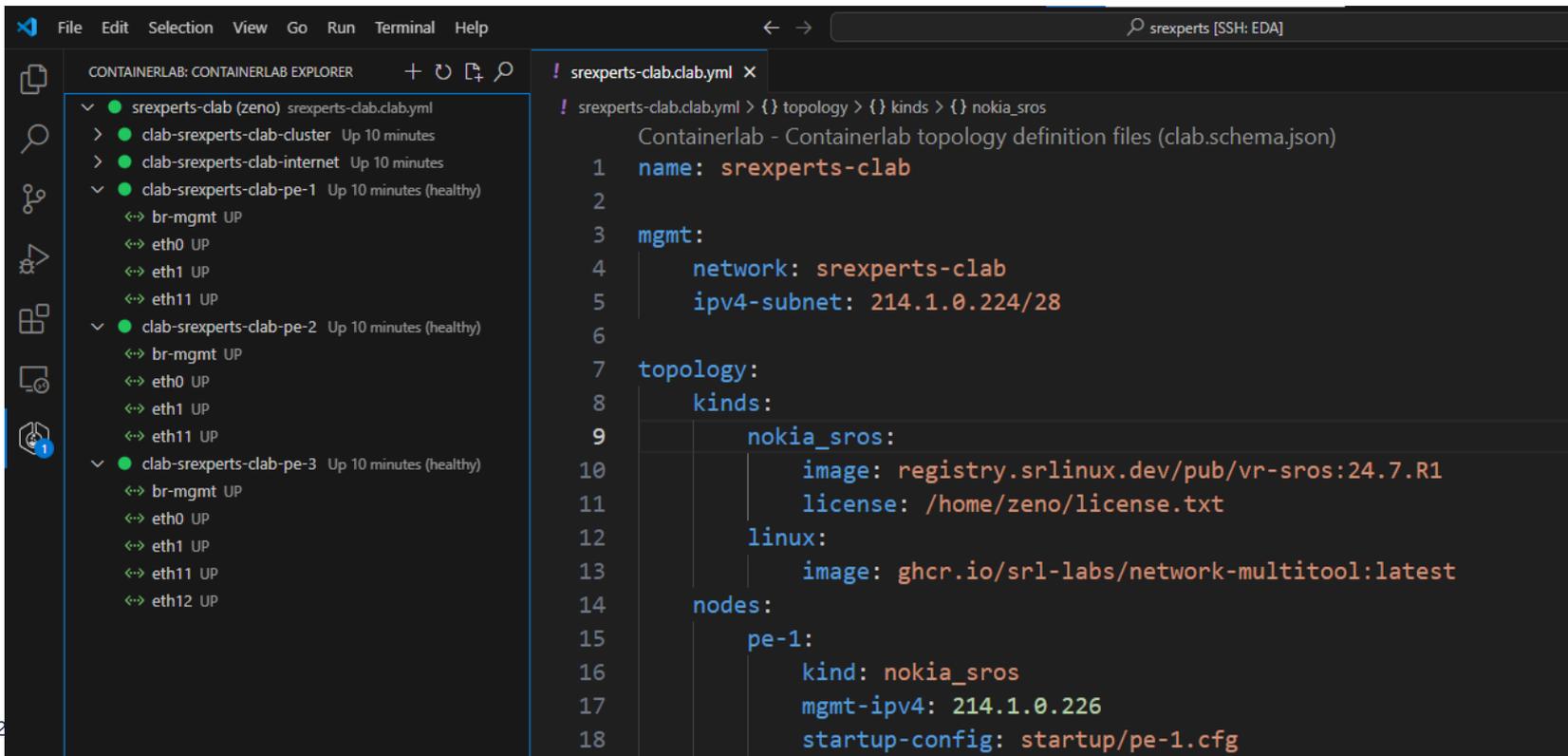
1. コードを書くためのアプリ
→ 多くのプログラミング言語や設定ファイル (YAML, Python, etc.) を編集できる「高性能なメモ帳」のような存在(IE)
2. 多くのプラットフォーム・OSで同じUI・機能を利用可能
→ Windows/macOS/Linux/Webで同じUI/拡張機能を利用することが可能。
3. 軽くて無料、しかも高機能
→ 無料なのに、コード補完・Git連携・ターミナル統合など、開発に必要な多くの拡張機能

何が嬉しいんだっけ？

- VS Codeの拡張機能である containerlab Extension も、上記全OSで動作し、containerlabを補助的に扱える GUI ツールとして使うことが可能。
- VS CodeをベースにしているCursorでも利用可能。共通ウィンドウでConfigの自動補完やバルクコンフィグを生成AIで出力させながら検証を進められる。

3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

3-6. コードとラボが同居する。効率も理解もスピードアップ。すべては一つの画面で
containerlab × VS-Code Extension : GUIベースでラボ設計・操作、CLIも1画面で



1. 本発表について

発表内容

1. 概要

2. containerlab(clab)とは？

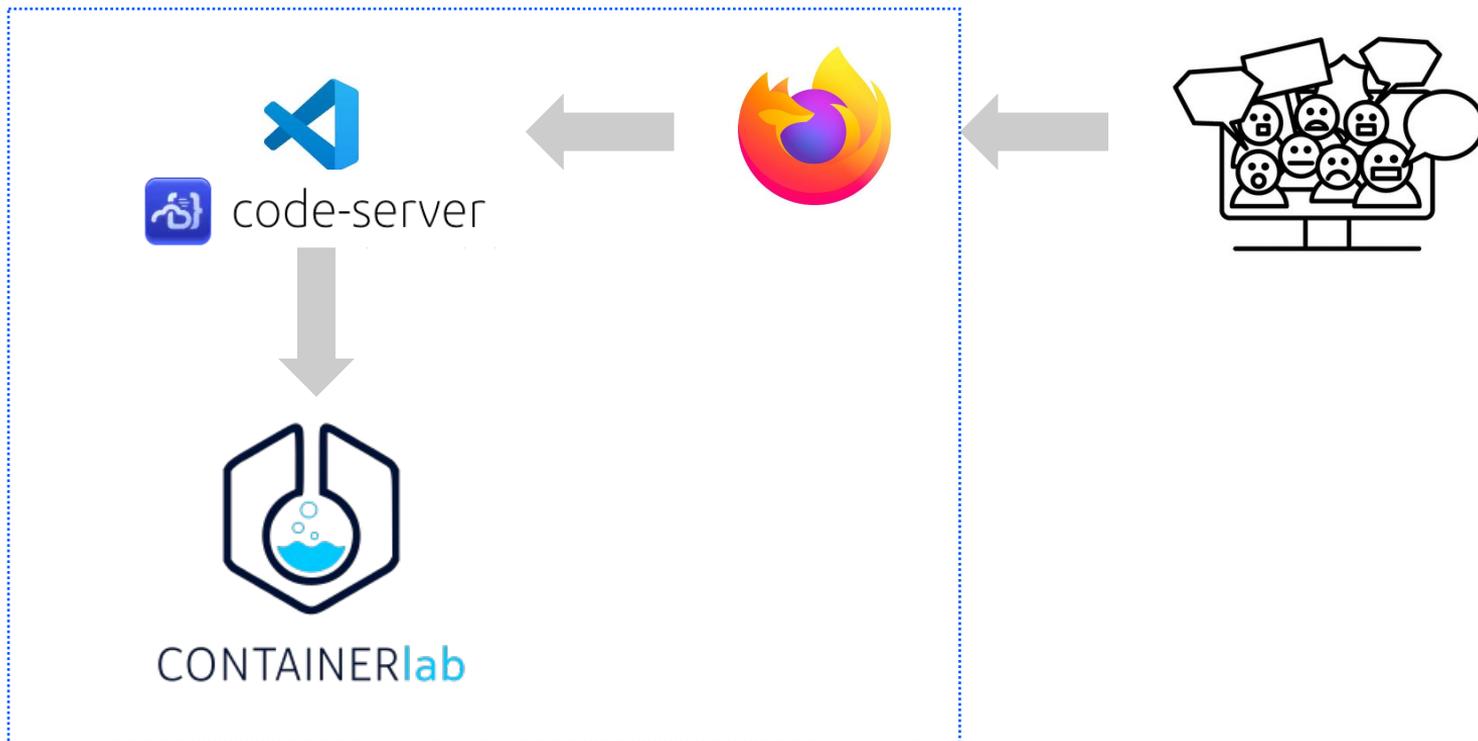
3. clab(と関連ツール)で実現するNWエンジニアのあるある願望

4. デモンストレーション

5. まとめ・議論

4. デモンストレーション

デモ環境と構成



4. デモンストレーション

The screenshot displays the Containerlab web interface. On the left, a sidebar lists 'RUNNING LABS' including 'st (nokia)' and various components like 'client1-3', 'gnmic', 'grafana', 'leaf1-3', 'loki', 'prometheus', 'promtail', 'spine1-2'. The main area shows a network topology diagram with nodes 'leaf1', 'spine1', 'leaf2', 'prometheus', 'grafana', and 'promtail'. The terminal window at the bottom shows system information for 'leaf1' and a prompt 'A:leaf1#'. The terminal output includes:

```
Serial Number      : Sim Serial No.  
System HW MAC Address: 1A:06:05:FF:00:00  
OS                 : SR Linux  
Software Version   : v24.10.1  
Build Number       : 492-gf8858c5836  
Architecture       : x86_64  
Last Booted        : 2025-07-30T01:45:11.816Z  
Total Memory       : 196542621 kB  
Free Memory        : 176098855 kB
```

Below the terminal output, it shows '--{ running }--[]--' and 'A:leaf1#'. The terminal title bar indicates 'Current mode: running' and 'admin (25) Wed 03:14AM'.



1. 本発表について

発表内容

1. 概要
2. containerlab(clab)とは？
3. clab(と関連ツール)で実現するNWエンジニアのあるある願望
4. デモンストレーション
- 5. まとめ・議論**

5. まとめ・議論 願望

本発表では containerlab の最新機能と関連ツールによって実現(?!)できるかもしれないNW技術者のあるある願望について紹介させていただきました。

議論したい点:

- 「ちょっと試したいだけ」のとき、皆さんはどんな環境を使っていますか？
→ 使っているツール名だけでなく、よくある“用途”や“つまずきポイント”もぜひ共有ください。
- NWラボに「こういう機能があったら助かる！」というアイデアはありますか？
→ 実装されていなくてもOK。「○○をGUIでできたら」「物理機器とつなげたい」など想像ベースで！
- NWとアプリ開発の境界が曖昧になってきた今、ラボに求められる“柔軟性”とは何でしょう？
→ CLI以外の操作/クラウド連携、CI/CD統合、チーム開発視点など、どこに価値があるか？意見ください。
- 学生&若手エンジニアが“気軽にラボを始める”には、どんなサポートがあると良いと思いますか？
→ GUI？チュートリアル？サンプル構成？「自分が最初にこれがあれば…」という視点で教えてください。NOSライセンス？

NOKIA

Let's enjoy コンテナラボ！

NOKIAブースで限定ステッカー配ってます！