
在庫管理とネットワーク監視を統合した 社内システムを開発した話

2021年6月25日

株式会社新潟通信サービス
近藤 勇希

自己紹介

- ・近藤 勇希
- ・株式会社新潟通信サービス
- ・2015年入社 of 27歳です。



目次

1. はじめに
2. 開発スタート
3. デモ
4. 今後の課題

目次

1. はじめに
2. 開発スタート
3. デモ
4. 今後の課題

開発のきっかけは、こんな感じでした

H社長



-
- 開発のきっかけは、こんな感じでした

H社長



うちのネットワーク監視システムなんとかならない？
(独自のシステム作れない？)

● 開発のきっかけは、こんな感じでした

H社長



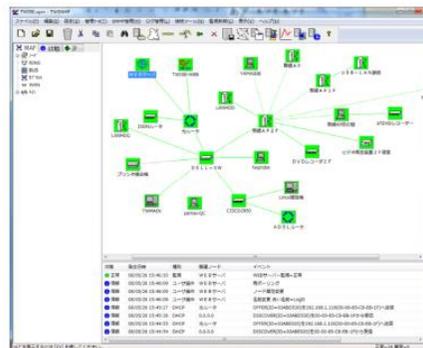
- ・現状ポート監視ができていない
機器のポートも監視したいな
- ・簡単なネットワーク図も欲しいな
- ・在庫管理システムと連動
できないかな

etc...

うちのネットワーク監視システムなんとかならない？
(独自のシステム作れない？)

●現状の監視

(1) NTT局舎の主要機器は【TWSNMP】でポートを監視



(2) エンドユーザーへの提供機器は【死活監視システム】で監視



これらの監視を行っている機器は【在庫管理システム】で管理

(各機器の管理、構成管理など)

別々のシステムで監視をしている結果

監視の登録漏れがあったり・・・

ポート単位での監視ができていない機器があったり・・・



実際のトラブル時

ぱっと情報が見えない・・・



どこのポートだ??
この機器の対向どこだ??
何の機器だ??

既存の在庫管理DB



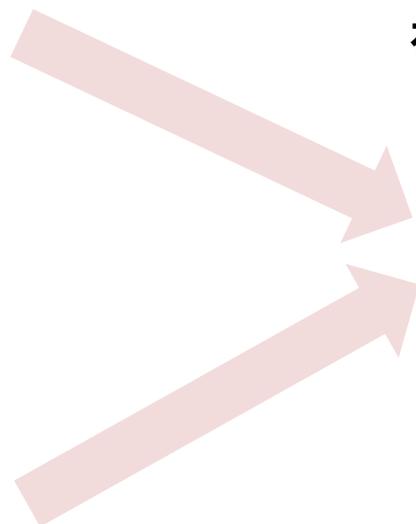
機器の管理データ
構成管理データ

既存の監視データ



監視用IPアドレス
監視ポート
その他もろもろ

在庫管理・ネットワーク監視DB



在庫管理・ネットワーク監視DB

機器の管理 + 構成管理 + 監視管理



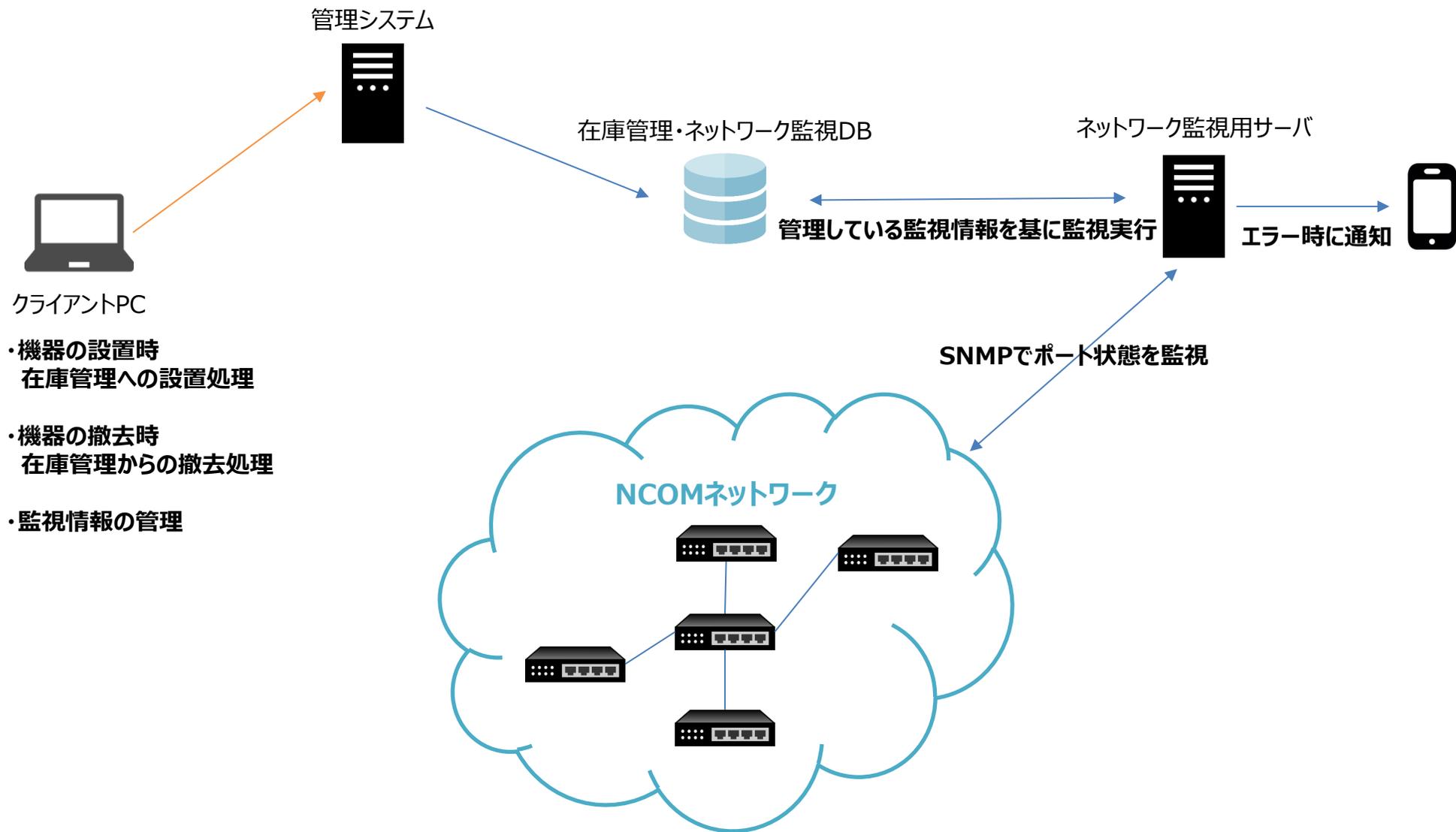
目標

弊社のネットワークに関わる全てを一元管理していく仕組みを作っていく

目次

1. はじめに
2. 開発スタート
3. デモ
4. 今後の課題

● 全体構成



クライアントPC

- ・機器の設置時
在庫管理への設置処理
- ・機器の撤去時
在庫管理からの撤去処理
- ・監視情報の管理

■ やること

1. 既存の在庫管理とネットワーク監視情報を合わせたDBを構築
2. SNMPで状態を取得してみる
3. 構築したDBを利用し、監視してみる
4. 簡易的なネットワーク図を作ってみる
5. 監視情報を操作する画面を作ってみる

■ やること

1. 既存の在庫管理とネットワーク監視情報を合わせたDBを構築
2. SNMPで状態を取得してみる
3. 構築したDBを利用し、監視してみる
4. 簡易的なネットワーク図を作ってみる
5. 監視情報を操作する画面を作ってみる

1. 既存の在庫管理とネットワーク監視情報を合わせたDBを構築 (1)

既存の在庫管理のDB

在庫管理番号:1001
スイッチA



スイッチAに対して、持っているデータ

- * メーカー、型番、シリアル等の基本情報
- * 拠点Aに設置
- * スイッチAには、SFP1が入っている
- * SFP1は拠点BのSFP100と対向関係

どのポートにSFPが挿入されているか（監視すべきポート）のデータを持っていなかった。

ポートの監視をしていくためには、データが足りないので・・・

1. 既存の在庫管理とネットワーク監視情報を合わせたDBを構築 (2)

- 監視機器、ポートの洗い出し

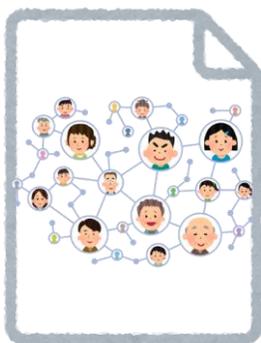
既存の在庫管理DB



マージして



NCOMネットワーク資料



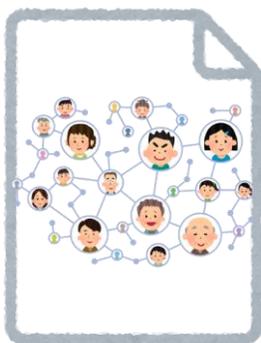
1. 既存の在庫管理とネットワーク監視情報を合わせたDBを構築 (2)

● 監視機器、ポートの洗い出し

既存の在庫管理DB



NCOMネットワーク資料



マージして



* 監視対象の機器は、どの機器管理番号の機器か

* 監視IPアドレスはなにか

* どのポートが監視対象か

* 監視対象のポートにはどの機器が入っているのか



● 洗い出したデータセット

機器管理番号	監視対象機器	設置場所	監視IP	ポート	挿入SFP	対向SFP
1001	スイッチA	拠点A	10.10.10.1	1	3001	4001
				2	3002	4002
				3	3003	4003
				10	3004	4004
				25	3005	4005
1002	スイッチB	拠点B	10.10.10.2	1	4001	3001
				2	4002	3002
				3	4003	3003
				4	4004	3004
				5	4005	3005

●洗い出したデータセット

機器管理番号	監視対象機器	設置場所	監視IP	ポート	挿入SFP	対向SFP
1001	スイッチA	拠点A	10.10.10.1	1	3001	4001
				2	3002	4002
				3	3003	4003
				10	3004	4004
				25	3005	4005
1002	スイッチB	拠点B	10.10.10.2	1	4001	3001
				2	4002	3002
				3	4003	3003
				4	4004	3004
				5	4005	3005

DB化



在庫管理・ネットワーク監視DBの用意ができた

■ やること

1. 既存の在庫管理とネットワーク監視情報を合わせたDBを構築

2. **SNMPで状態を取得してみる**

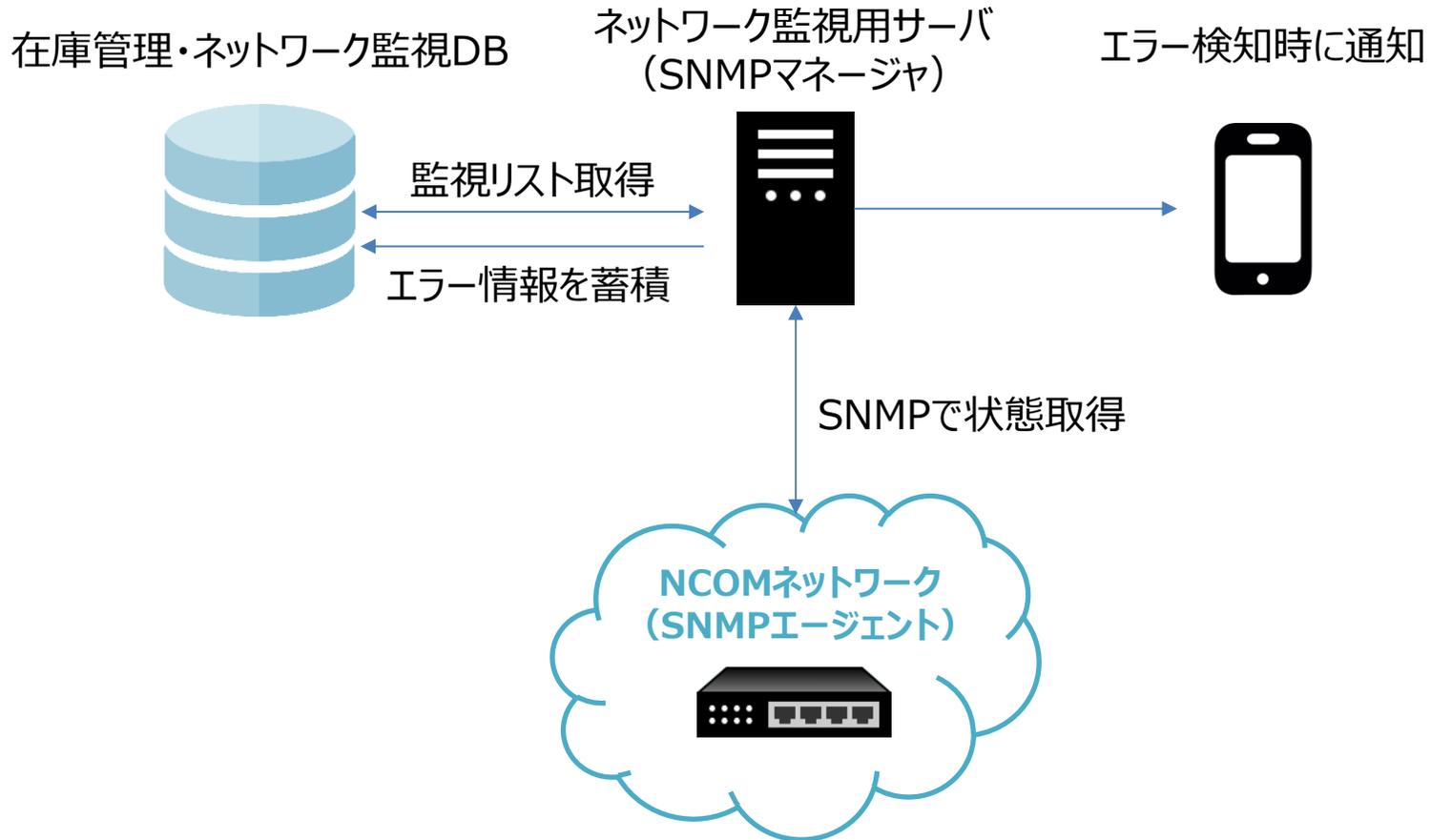
3. 構築したDBを利用し、監視してみる

4. 簡易的なネットワーク図を作ってみる

5. 監視情報を操作する画面を作ってみる

2. SNMPで状態を取得してみる（1）

● 構成



2. SNMPで状態を取得してみる (2)

- SNMPを利用して、ポート状態を取得

ネットワーク監視用サーバ
(SNMPマネージャ)



SNMP Get Request [oid=1.3.6.1.2.1.2.2.1.8.1]



SNMP Get Response [UP]



スイッチA



現在のポート1の状態は、UPですよ。

スイッチAのポート1の状態を知りたい。
oid=1.3.6.1.2.1.2.2.1.8.1の情報をちょうだい！

■ やること

1. 既存の在庫管理とネットワーク監視情報を合わせたDBを構築
2. SNMPで状態を取得してみる
- 3. 構築したDBを利用し、ポートの監視を試みる**
4. 簡易的なネットワーク図を作ってみる
5. 監視情報を操作する画面を作ってみる

3. 構築したDBを利用し、ポートの監視してみる（1）

- 在庫管理・ネットワーク監視DBの情報を基にポート状態を取得する

現在設置されている機器かつ監視情報が登録されているものを監視リストとして抽出

* 抽出した監視リスト

機器管理番号	監視IP	メーカー	型番	監視ポート
1001	10.10.10.10	D-link	aaa	11/18/25/26/27/28
1002	10.10.10.20	KTI	bbb	6
1003	10.10.10.30	アライドテレシス	ccc	1/2/3/4/5/6/7/8/9/10/15
...
1501	10.10.10.150	D-link	ddd	17/18/19
1503	10.10.10.170	FXC	fff	8

3. 構築したDBを利用し、監視してみる (2)

抽出した監視リストに対し、SNMPでポート状態を取得

ネットワーク監視用サーバ
(SNMPマネージャ)



1分毎にSNMPで状態取得

SNMP Get Request

SNMP Get Response



```
snmpwalk -OnQ -v 2c -c password
10.10.10.1 .1.3.6.1.2.1.2.2.1.8.11
.1.3.6.1.2.1.2.2.1.8.11 = up
snmpwalk -OnQ -v 2c -c password
10.10.10.1 .1.3.6.1.2.1.2.2.1.8.18
.1.3.6.1.2.1.2.2.1.8.18 = up
snmpwalk -OnQ -v 2c -c password
10.10.10.1 .1.3.6.1.2.1.2.2.1.8.25
.1.3.6.1.2.1.2.2.1.8.25 = up
.....
snmpwalk -OnQ -v 2c -c password
10.10.10.2 .1.3.6.1.2.1.2.2.1.8.6
Timeout: No Response
```

3. 構築したDBを利用し、監視してみる (2)

抽出した監視リストに対し、SNMPでポート状態を取得

ネットワーク監視用サーバ
(SNMPマネージャ)



1分毎にSNMPで状態取得

SNMP Get Request

SNMP Get Response



```
snmpwalk -OnQ -v 2c -c password  
10.10.10.1 .1.3.6.1.2.1.2.2.1.8.11  
.1.3.6.1.2.1.2.2.1.8.11 = up  
snmpwalk -OnQ -v 2c -c password  
10.10.10.1 .1.3.6.1.2.1.2.2.1.8.18  
.1.3.6.1.2.1.2.2.1.8.18 = up  
snmpwalk -OnQ -v 2c -c password  
10.10.10.1 .1.3.6.1.2.1.2.2.1.8.25  
.1.3.6.1.2.1.2.2.1.8.25 = up  
.....  
snmpwalk -OnQ -v 2c -c password  
10.10.10.2 .1.3.6.1.2.1.2.2.1.8.6  
Timeout: No Response
```



在庫管理番号	監視IP	メーカー	型番	監視ポート
1002	10.10.10.20	KTI	bbb	6

確実に型番bbbにポート6は存在するのに応答がない??



確認してみると

確認してみると

.1.3.6.1.2.1.2.2.1.8.101 = up

.1.3.6.1.2.1.2.2.1.8.102 = down

.1.3.6.1.2.1.2.2.1.8.103 = down

.1.3.6.1.2.1.2.2.1.8.104 = down

.1.3.6.1.2.1.2.2.1.8.105 = down

.1.3.6.1.2.1.2.2.1.8.106 = UP

SNMPで取得した際のポートの表記が違った・・・

他の機器でも情報を取得してみると
型番でポートの表記の違いが。

K社製のとある機器

.1.3.6.1.2.1.2.2.1.8.101 = up

A社製のとある機器

.1.3.6.1.2.1.2.2.1.8.5001 = up

他の機器でも情報を取得してみると
型番でポートの表記の違いが。

K社製のとある機器

.1.3.6.1.2.1.2.2.1.8.101 = up

A社製のとある機器

.1.3.6.1.2.1.2.2.1.8.5001 = up



監視実行時に自動で型番毎のポート表記に合わせるよう対応

(3) ポート表記を型番毎に対応させて、状態を取得

```
snmpwalk -OnQ -v 2c -c password
10.10.10.1 .1.3.6.1.2.1.2.2.1.8.11
.1.3.6.1.2.1.2.2.1.8.11 = up
snmpwalk -OnQ -v 2c -c password
10.10.10.1 .1.3.6.1.2.1.2.2.1.8.18
.1.3.6.1.2.1.2.2.1.8.18 = up
snmpwalk -OnQ -v 2c -c password
10.10.10.1 .1.3.6.1.2.1.2.2.1.8.25
.1.3.6.1.2.1.2.2.1.8.25 = up
.....
snmpwalk -OnQ -v 2c -c password
10.10.10.2 .1.3.6.1.2.1.2.2.1.8.106
.1.3.6.1.2.1.2.2.1.8.106 = up
```



一通り、弊社で取り扱う機器のポート状態取得ができることを確認。
(一部除く)

ポート状態の監視もOK！！

3. 構築したDBを利用し、監視してみる (3)

●エラー検知・復旧検知

ネットワーク監視用サーバ
(SNMPマネージャ)



SNMP Get Request

SNMP Get Response[DOWN]



```
snmpwalk -OnQ -v 2c -c password  
10.10.10.2 .1.3.6.1.2.1.2.2.1.8.106  
.1.3.6.1.2.1.2.2.1.8.106 = down
```

監視対象ポートが、down

3. 構築したDBを利用し、監視してみる (3)

●エラー検知・復旧検知

ネットワーク監視用サーバ
(SNMPマネージャ)



SNMP Get Request

SNMP Get Response[DOWN]



```
snmpwalk -OnQ -v 2c -c password  
10.10.10.2 .1.3.6.1.2.1.2.2.1.8.106  
.1.3.6.1.2.1.2.2.1.8.106 = down
```

監視対象ポートが、down



エラー情報をDBへ蓄積

- 10.10.10.2のポート6がダウンしたよ

3. 構築したDBを利用し、監視してみる (3)

- エラー検知・復旧検知

- ② エラー情報 + 設置場所、対向先をメールで一齐に通知

ネットワーク監視用サーバ
(SNMPマネージャ)



10.10.10.2のポート6がダウンしたぞー！
設置場所は* * *
対向は* * *だよ！

3. 構築したDBを利用し、監視してみる (3)

●エラー検知・復旧検知

②エラー情報 + 設置場所、対向先をメールで一斉に通知

ネットワーク監視用サーバ
(SNMPマネージャ)



10.10.10.2のポート6がダウンしたぞー！
設置場所は* * *
対向は* * *だよ！

③エラー検知機器に対し、1分毎に復旧チェックを行う

ネットワーク監視用サーバ
(SNMPマネージャ)



復旧したか？？？？

SNMP Get Response[UP]

3. 構築したDBを利用し、監視してみる (3)

●エラー検知・復旧検知

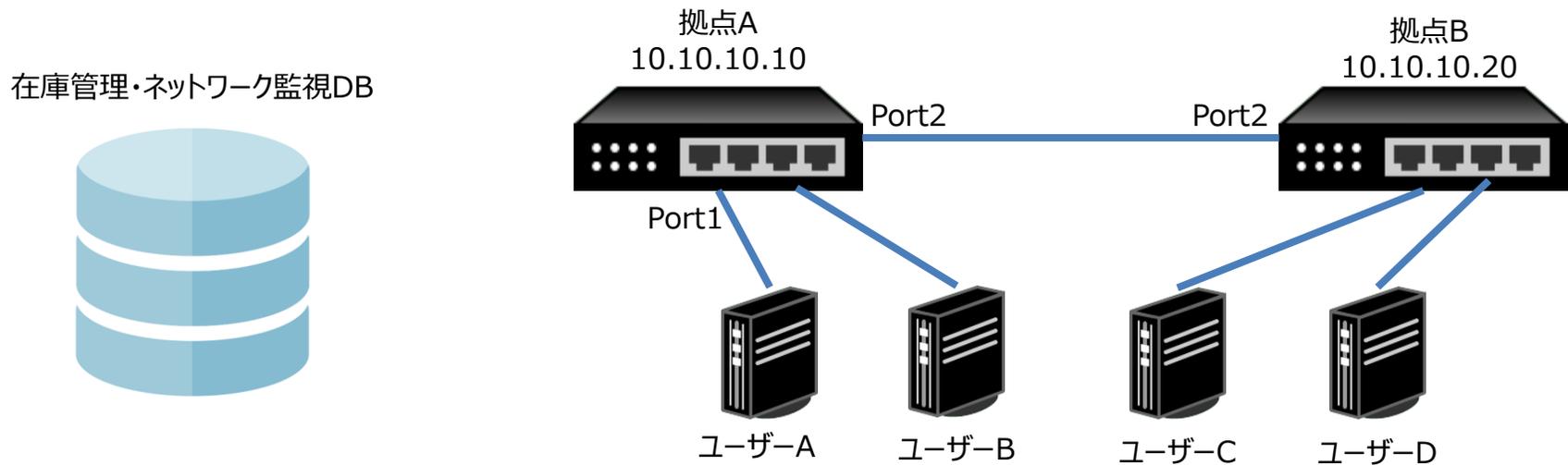
④復旧検知ができれば、メールで一斉に通知

ネットワーク監視用サーバ
(SNMPマネージャ)



10.10.10.2のポート6が復旧したよ！！
設置場所は***
対向は***だよ！

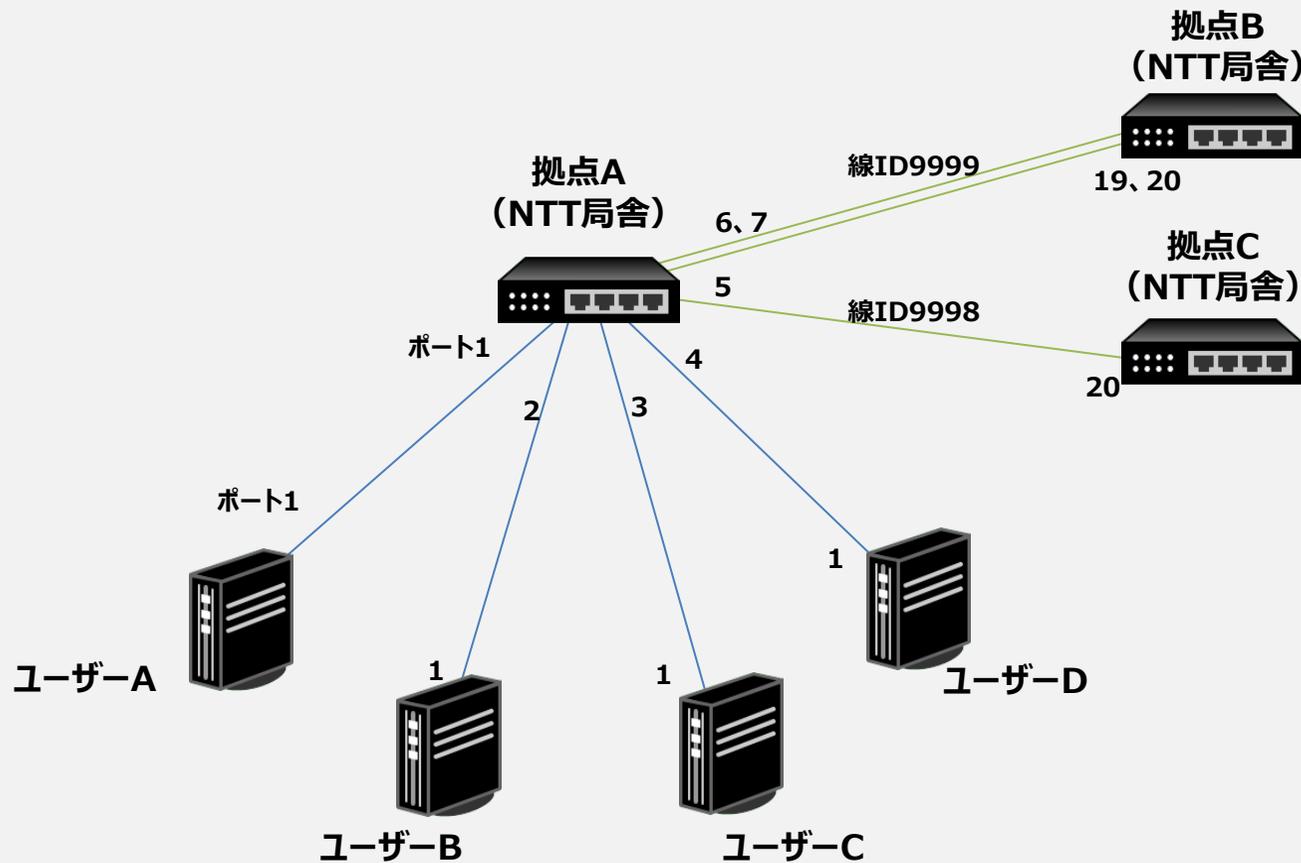
4. 簡易的なネットワーク図を作ってみる



DBに持っている機器・ポート同士の繋がりを利用して、自動でネットワーク図を生成する

目標のグラフ

- 弊社ネットワーク構成を拠点単位でグラフ化（ポート付）
中継線のIDも入れたい（後々）



●ライブラリの選定

* 選定の基準 *

- ・ドキュメントが**豊富**なこと
- ・短時間で実装したかったので、**学習コスト少なめ**がよい
- ・グラフ描写 + **拡張性**があること
 - 線上にポート番号などのテキストを記載したい
 - グラフのレイアウトを自由に決めたい

(見やすいツリー構造での表示が希望)



色々あります。グラフ描写ライブラリ。

ネットワークグラフ描画ライブラリ7個まとめ

<https://qiita.com/SuyamaDaichi/items/c47dc74cfefd92516e28>

名前	情報量	実装のしやすさ	拡張性	動作が軽い か	Neo4j との使用 例	備考
D3.js	◎(バージョン注意)	○ (jqueryライク)	◎	○?	△(中国圏)	リファレンスは多めだが元々グラフDB向けではない
sigma.js	○(日本語少)	△	?	◎(WebGL)	△(英語圏)	SVG,Canvas,WebGL対応、Neo4j使用例有
vis.js	○	◎	△	○	○(英語圏)	グニャグニャ動く
VivaGraphJS	△(Githubのみ)	?	△	◎	×	双方向矢印有(?)
Cytoscape.js	○(日本語少)	△	○	○	○(拡張機能で対応)	開発者に日本人がいる
cola.js	△	○	◎	○(ノード数100以下推奨)	○	D3.js、Cytoscape.jsと組み合わせる
KeyLines	◎	○	△	◎	◎	HTTPS対応。有償

色々あります。グラフ描写ライブラリ。

ネットワークグラフ描画ライブラリ7個まとめ

<https://qiita.com/SuyamaDaichi/items/c47dc74cfefd92516e28>

名前	情報量	実装のしやすさ	拡張性	動作が軽い か	Neo4j との使用 例	備考
D3.js	◎(バージョン注意)	○(jqueryライク)	◎	○?	△(中国圏)	リファレンスは多めだが元々グラフDB向けではない
sigma.js	○(日本語少)	△	?	◎(WebGL)	△(英語圏)	SVG,Canvas,WebGL対応、Neo4j使用例有
vis.js	○	◎	△	○	○(英語圏)	グニャグニャ動く
Vivagraph.js	△(Githubのみ)	?	△	◎	×	双方向矢印有(?)
Cytoscape.js	○(日本語少)	△	○	○	○(拡張機能で対応)	開発者に日本人がいる
cola.js	△	○	◎	○(ノード数100以下推奨)	○	D3.js、Cytoscape.jsと組み合わせる
KeyLines	◎	○	△	◎	◎	HTTPS対応。有償

▲ 描画ライブラリとして有名、ドキュメント豊富で拡張性（大）だが、その分学習コスト多め

▲ 拡張性低め

- ・ツリーレイアウトがなく自身で作成する必要あり
- ・エッジ（点と点を繋ぐ線）上にテキストを挿入できない
- ・拡張レイアウトできない（点の位置を一部のみ自由に変更できない）などなど

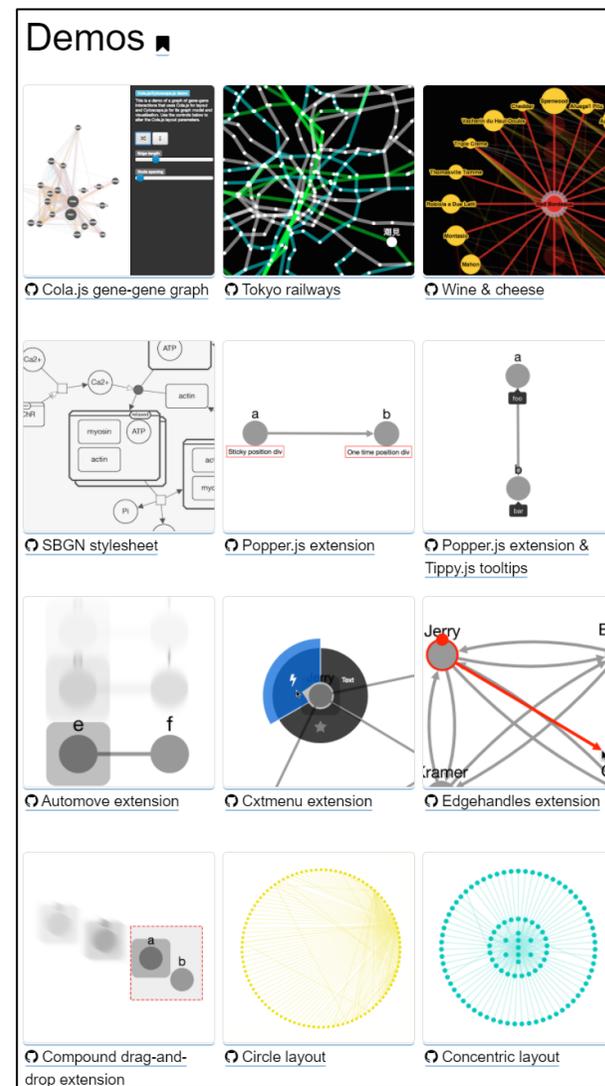
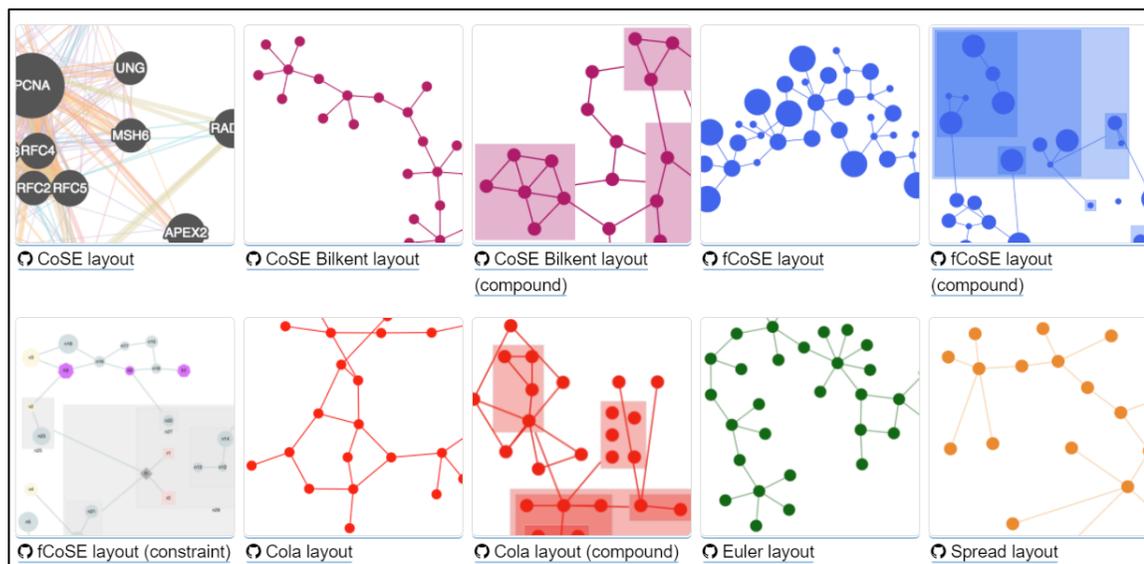
これいいかも？

▲ ドキュメント少ない

▲ 有償

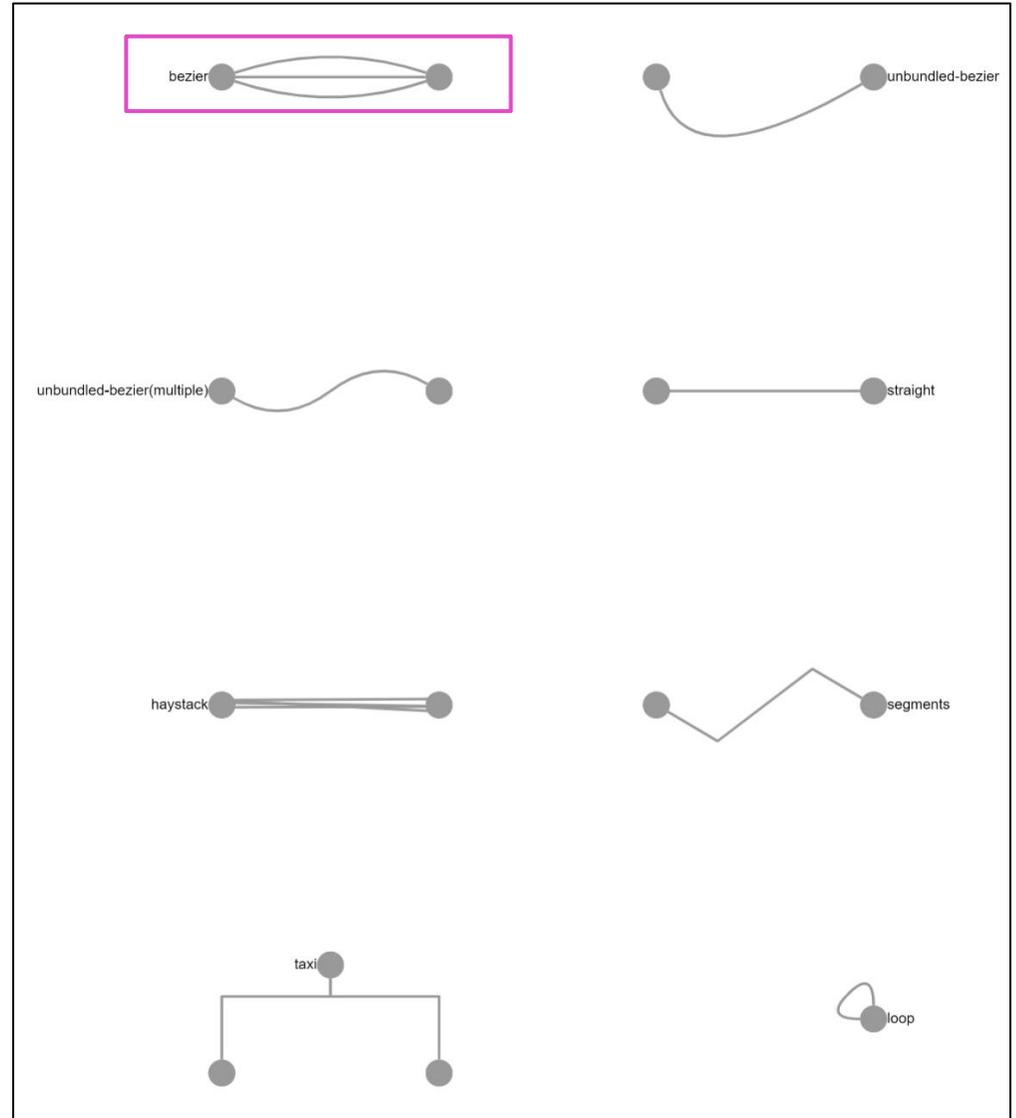
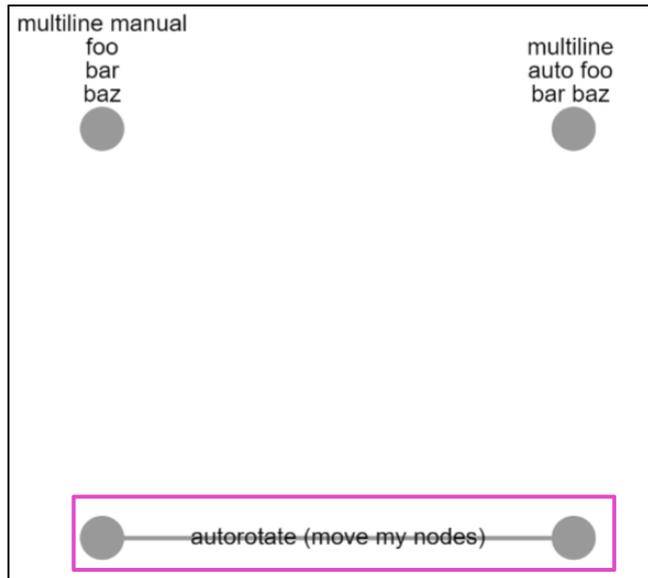
Cytoscape.js ?

- オープンソースのグラフ構造可視化
JavaScript ライブラリ
- レイアウトが無数にある
(ユーザーによる拡張レイアウトも多数)
<https://js.cytoscape.org/#demos>
- グラフライブラリの中では軽量に動作する方 (らしい)
- 英語のみだが公式ドキュメントは豊富



Cytoscape.js ?

- 点同士を繋ぐ線の表現が豊富
(基本は8つ、LAG接続も表示可能)
- 線上に文字列を表示可能

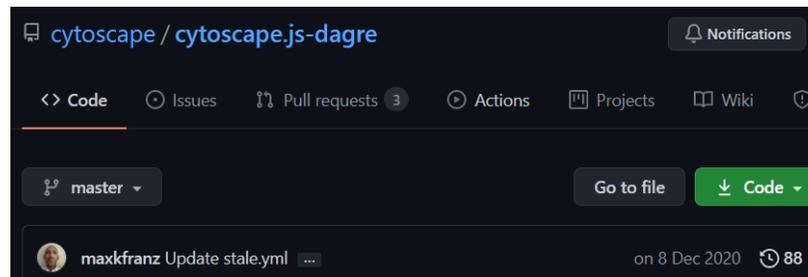
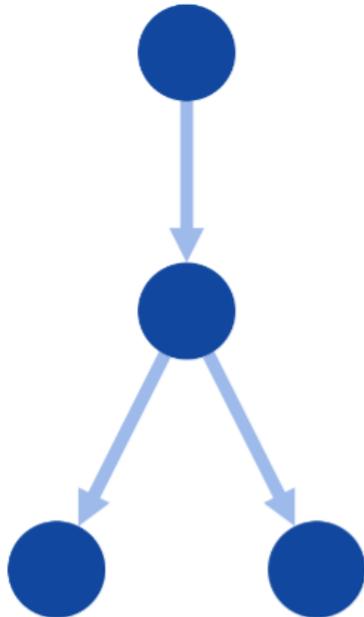


Cytoscape.js ?

- ツリーグラフは作れるのか？

→ デフォルトのCytoscape.jsにはツリーレイアウトが何故かない
githubに公開されているcytoscape-dagreレイアウトをcdnjsから利用

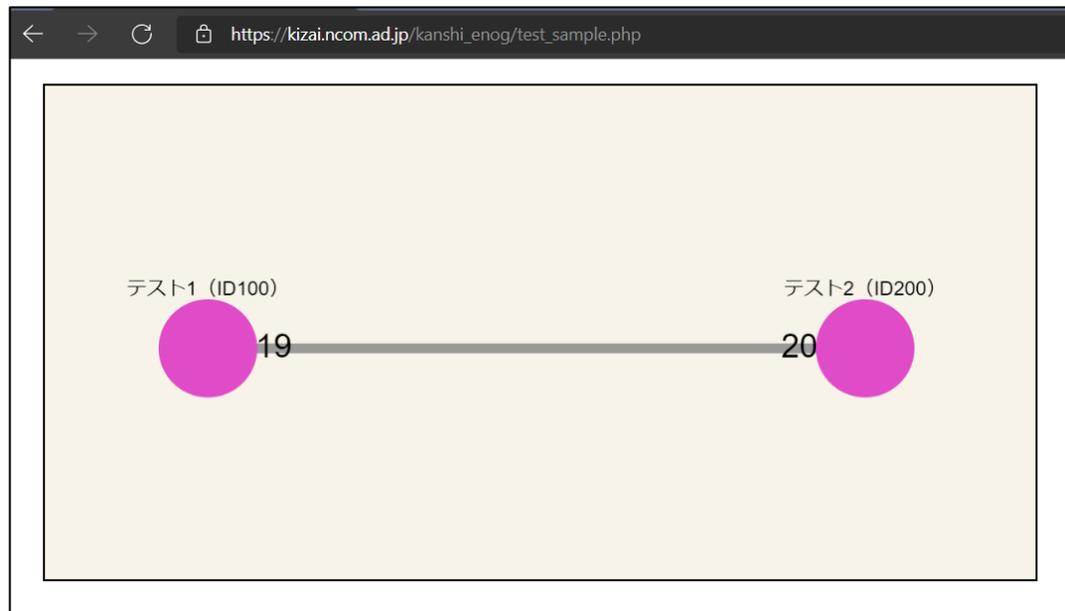
cytoscape-dagre demo



Cytoscape.js ?

●入力する情報

```
{//ノード・エッジ情報
  nodes:[
    {data: {id: '100', name: 'テスト1'}},
    {data: {id: '200', name: 'テスト2'}}
  ],
  edges:[
    {{data: {source: '100', target: '200',
      source_port: '19', target_port: '20'}}
    }
  ],
}
```



テスト1の19とテスト2の20は繋がっている

● グラフ表示までの手順

1. 在庫管理・ネットワーク監視DBから拠点単位で入力データ（JSON）を書き出す

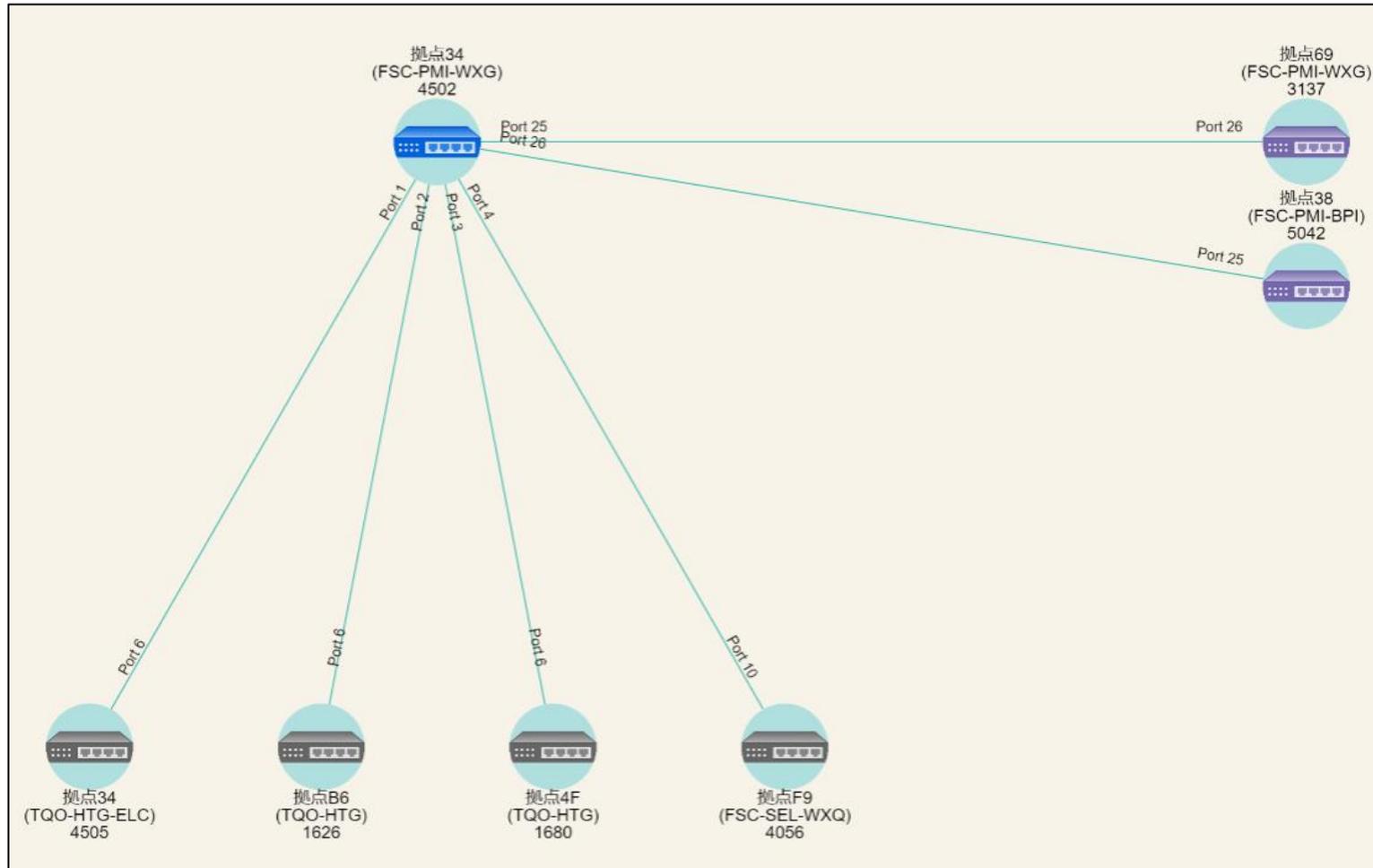


```
nodes:[
  {data: {id: '9999', name: '機器9'}},
  {data: {id: '1111', name: '機器1' }},
  {data: {id: '2222', name: '機器2' }},
  {data: {id: '3333', name: '機器3' }},
  {data: {id: '4444', name: '機器4' }}
],
edges:[
  {data: {source: '9999', target: '1111', source_port: '1', target_port: '6'}},
  {data: {source: '9999', target: '2222', source_port: '2', target_port: '6'}},
  {data: {source: '9999', target: '3333', source_port: '3', target_port: '6'}},
  {data: {source: '9999', target: '4444', source_port: '4', target_port: '6'}}
],
```

2. グラフ描写用Javascriptへ入力データを渡して表示

● 完成した簡易的なネットワーク図

* 拠点34の場合

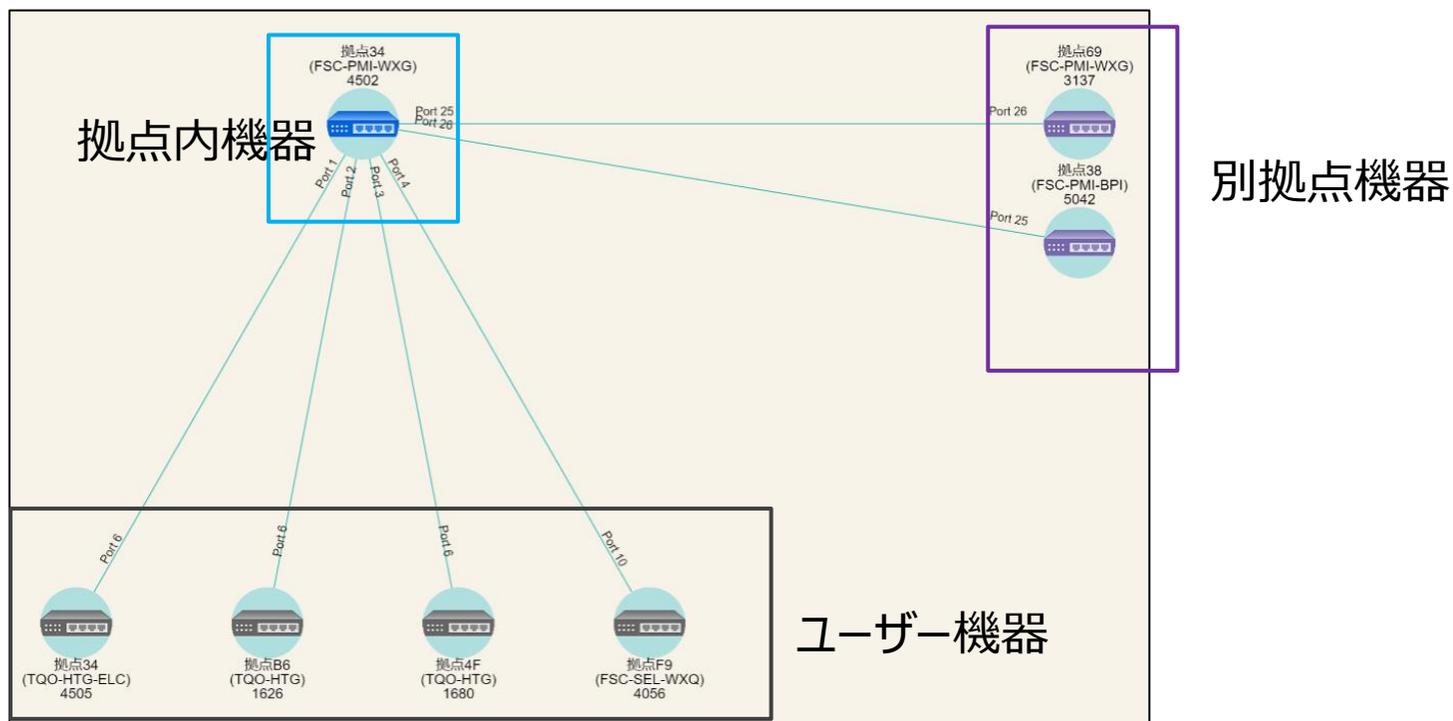


おまけ：レイアウト調整①

- ツリー表示

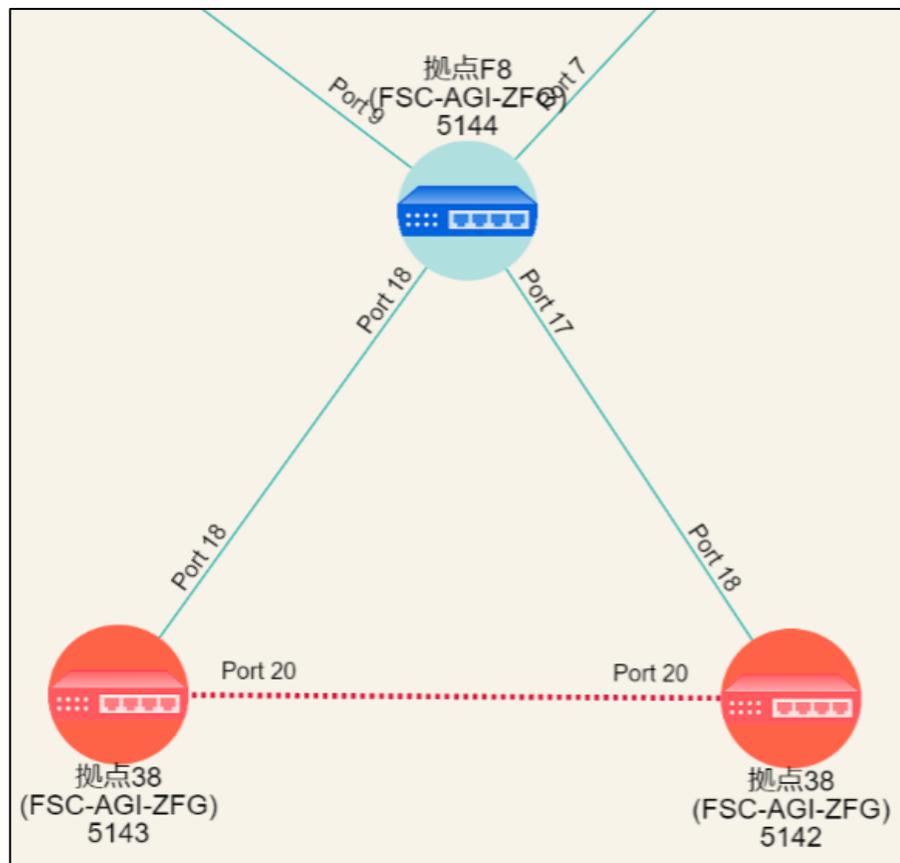
→ 拠点から別拠点へ行く機器は右側に表示

拠点内機器：青、別拠点機器：紫に色分け、また別拠点機器を右側に表示するよう変更



おまけ：レイアウト調整②

●エラー表示



監視でポートエラー or Pingエラーを検知
→ エラーノードを赤く表示

ポートエラー + 対向ポートもエラー
ポートエラー + 対向機器Pingエラー
→ 機器同士を結ぶエッジを点滅

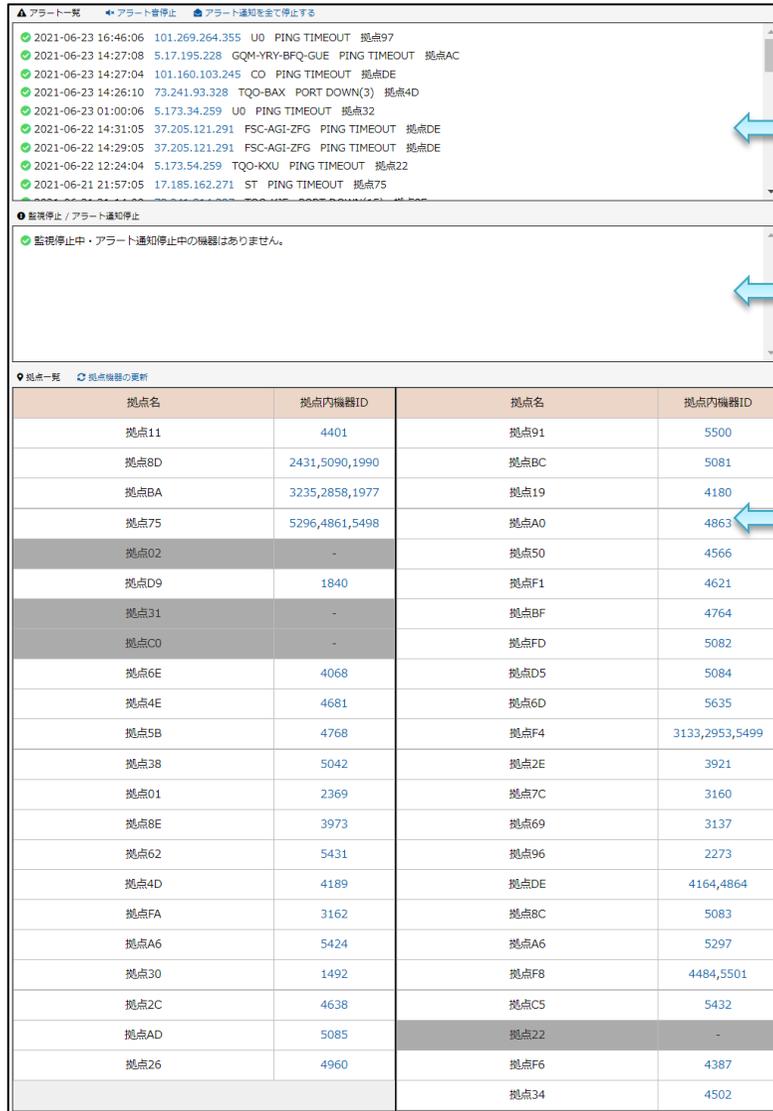
← 例：拠点38にあるFSC-AGI-ZFGの
ポート20がエラーの場合

5. 監視情報を操作する画面を作ってみる

● やること

- ・エラー情報を可視化する
- ・監視のON/OFFが行えるように
- ・エラー時の通知の有無の変更ができるように

● トップページ



The screenshot displays the homepage interface with three main sections:

- アラート一覧 (Alert List):** A scrollable list of alerts with columns for timestamp, IP address, protocol, error type, and point name. Examples include "PING TIMEOUT" for points 97, AC, DE, 4D, 32, OE, 22, and 75.
- 監視停止 / アラート通知停止 (Monitoring Stop / Alert Notification Stop):** A section indicating that no devices are currently in a monitoring stop or alert notification stop state.
- 拠点一覧 (Point List):** A table listing various points with their names and internal IDs.

拠点名	拠点内機器ID	拠点名	拠点内機器ID
拠点11	4401	拠点91	5500
拠点8D	2431,5090,1990	拠点BC	5081
拠点BA	3235,2858,1977	拠点19	4180
拠点75	5296,4861,5498	拠点A0	4863
拠点02	-	拠点50	4566
拠点D9	1840	拠点F1	4621
拠点31	-	拠点BF	4764
拠点C0	-	拠点FD	5082
拠点6E	4068	拠点D5	5084
拠点4E	4681	拠点6D	5635
拠点5B	4768	拠点F4	3133,2953,5499
拠点38	5042	拠点2E	3921
拠点01	2369	拠点7C	3160
拠点8E	3973	拠点69	3137
拠点62	5431	拠点96	2273
拠点4D	4189	拠点DE	4164,4864
拠点FA	3162	拠点8C	5083
拠点A6	5424	拠点A6	5297
拠点30	1492	拠点F8	4484,5501
拠点2C	4638	拠点C5	5432
拠点AD	5085	拠点22	-
拠点26	4960	拠点F6	4387
		拠点34	4502

● エラーがあった場合
エラー検知情報を表示

● 監視停止中・アラート停止中の機器を表示

● 各拠点をリスト表示
管理番号をクリックすることで拠点別の
詳細画面へ遷移

拠点別の詳細画面

- 現状のグラフをツリー表示
 - 拠点内機器、ユーザー機器、別拠点機器
 - エラー機器は赤く表示

- 操作
 - 監視のON・OFF
 - アラート時の通知のON・OFFを設定する

The image displays a network diagram on the left and a configuration interface on the right. The network diagram shows four nodes: 拠点11 (FSC-AGI-ZFG) 4401, 拠点91 (VLO-FPS) 5500, 拠点A0 (FSC-AGI-TIO) 4863, and 拠点01 (TQO-HTG-ELC) 4392. Connections are shown between 拠点11 and 拠点91 (Port 19 to Port 9), 拠点11 and 拠点A0 (Port 20 to Port 27), and 拠点11 and 拠点01 (Port 17 to Port 6).

The configuration interface on the right shows the '監視' (Monitoring) settings for a specific port. A red arrow points to the '監視' (Monitoring) toggle switch, which is currently turned on. Below the interface is a table showing the monitoring configuration for various ports.

ポート	監視	アラート通知	状態	対向機器設置場所	対向機器型番	対向機器ポート	対向機器IPアドレス
1				-	-	-	-
2				-	-	-	-
3				-	-	-	-
4				-	-	-	-
5				-	-	-	-
6				-	-	-	-
7				-	-	-	-
8				-	-	-	-
9				-	-	-	-
10				-	-	-	-
11				-	-	-	-
12				-	-	-	-
13				-	-	-	-
14				-	-	-	-
15				-	-	-	-
16				-	-	-	-
17	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	拠点01	TQO-HTG-ELC	6	17.185.81.272
18				-	-	-	-
19	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	拠点91	VLO-FPS	9	101.103.162.355
20	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	拠点A0	FSC-AGI-TIO	27	19.187.77.273

目次

1. はじめに
2. 開発スタート
- 3. デモ**
4. 今後の課題

目次

1. はじめに
2. 開発スタート
3. デモ
4. 今後の課題

課題

- ・まだポート監視ができていない機器への対応
- ・トラフィックの監視も組み込みたい
- ・機器ごとのログ取得をしていきたい



以上、ありがとうございました。