

RAN 環境構築

環境構築 (RAN) 環境構築

OmniTouch 環境構築

目次

1. 環境構築
2. RAN 環境構築
3. 環境構築
4. Web UI 環境構築
5. Grafana 環境構築
6. 環境構築
7. 環境構築
8. 環境構築
9. 環境構築

目次

RAN 環境構築 Nokia AirScale 環境構築 3GPP LTE 5G 環境構築 RAN 環境構築

目次

- 環境構築 - 環境構築
- 環境構築 - 環境構築
- 環境構築 - 環境構築
- Web 環境構築 - 環境構築 Web UI 環境構築
- Grafana 環境構築 - 環境構築

項目	項目	項目
RAN 設定	設定	設定
Web UI 設定	設定	https://<server>;9443
MySQL 設定	設定	設定
InfluxDB	設定	http://<server>;8086
Grafana	設定	http://<server>;3000
TCE NSNTI 設定	設定	TCP 設定 49151
TCE TZSP 設定	設定 Wireshark	UDP 設定 37008

The screenshot displays the Nokia Monitor Center (NMC) interface, showing a detailed view of a 5G network configuration. The interface is divided into several sections:

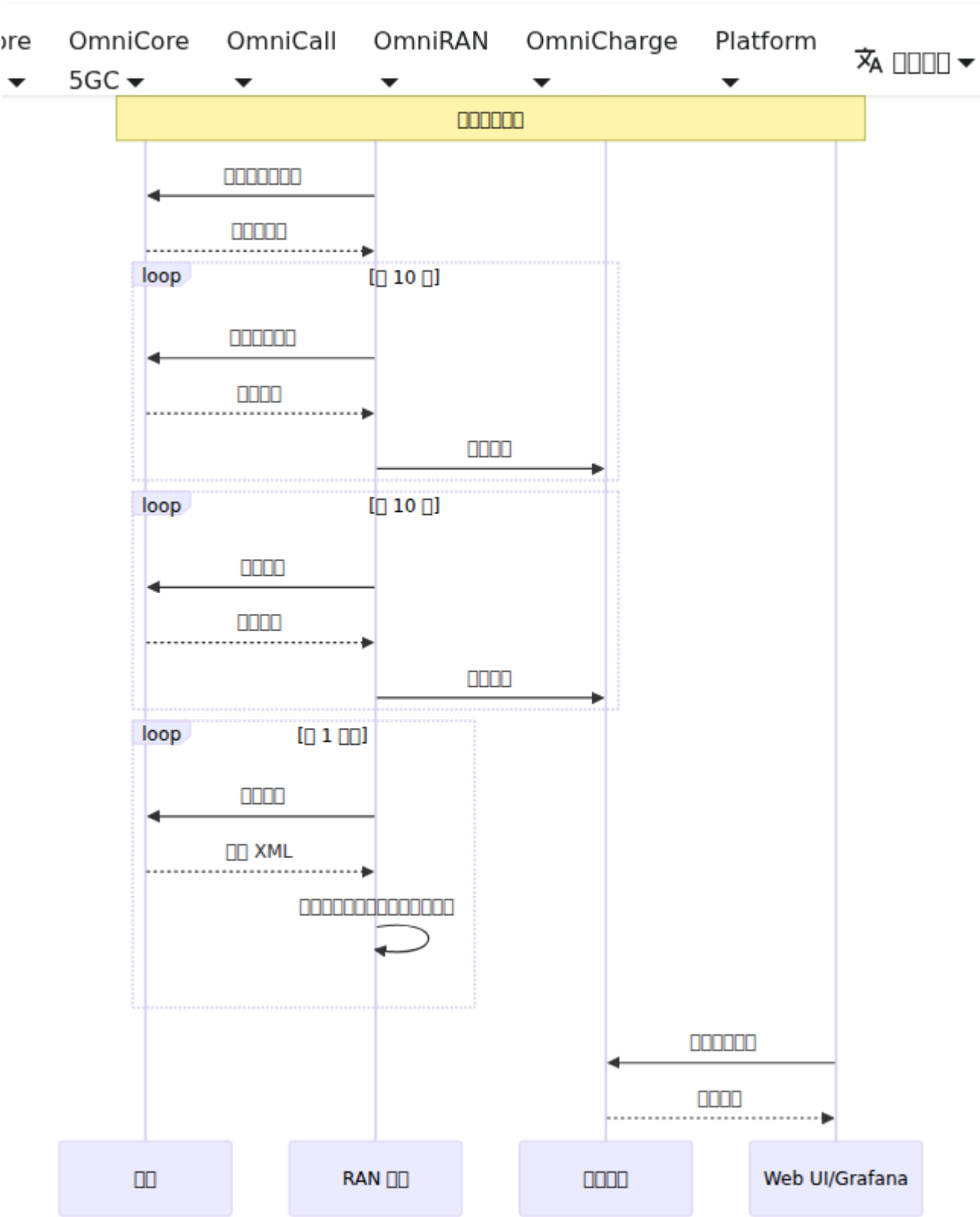
- 5G Connections - LNMME-0**: Shows 5G Name, IP, Administrative State (Enabled), and 5G Link Status (Available).
- 5G Connections - LNMME-1**: Shows 5G Name, IP, Administrative State (Enabled), and 5G Link Status (Available).
- 5G Connections - LNMME-2**: Shows 5G Name, IP, Administrative State (Enabled), and 5G Link Status (Available).
- 5G Connections - LNMME-3**: Shows 5G Name, IP, Administrative State (Enabled), and 5G Link Status (Available).
- Cells - LNCEL-1**: Shows Operational State (Enabled) and Cell Name (-B2-CellI01).
- Cells - LNCEL-2**: Shows Operational State (Enabled) and Cell Name (-B2-CellI01).
- Cells - LNCEL-3**: Shows Operational State (Enabled) and Cell Name (-B2-CellI01).
- Cells - LNCEL-4**: Shows Operational State (Enabled) and Cell Name (HAY04-B12-CellI01).
- Cells - LNCEL-5**: Shows Operational State (Enabled) and Cell Name (-B12-CellI02).
- Cells - LNCEL-6**: Shows Operational State (Enabled) and Cell Name (-B12-CellI01).
- Performance Monitoring**: Shows Last Reported (12:00:00) and Average PDCP Throughput (5.01 Mb/s).
- Operation Map**: Shows a map of the network area.

RAN 部署計画

RAN 環境構築手順

1. 環境構築 - 必要 Nokia 製品をインストール
2. ネットワーク - 10 台の KPI を監視
3. テスト - 各種テストを実行
4. データ - データを収集・分析
5. 監視 - 24時間監視システムを構築
6. 報告 - 必要 Web UI を Grafana で構築

□□□



□□□□□

□□□□□

- 0000000000000000
- 00000000/000
- 000000PRB 00000
- 00000000
- 0000
- 000000

000

- 000000000000000000000000
- 0000000000
- 0000000000
- 0000000000

000

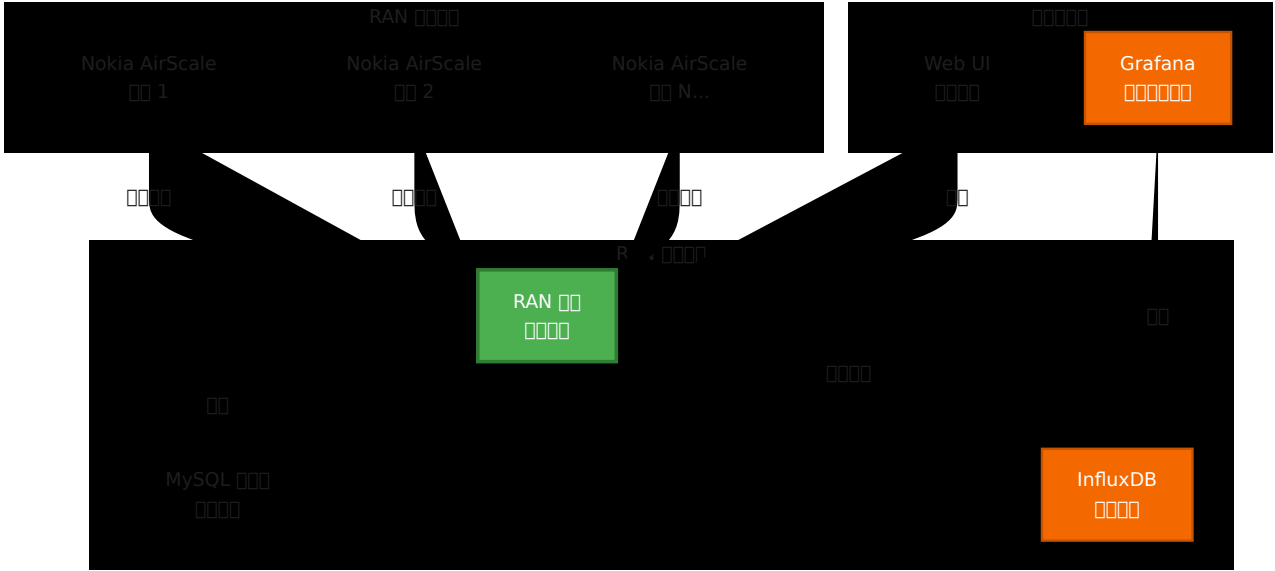
- 000 XML 0000000000000000
- 000000000000
- 0000000000
- 00000000 10 000

0000000000000000 0000000

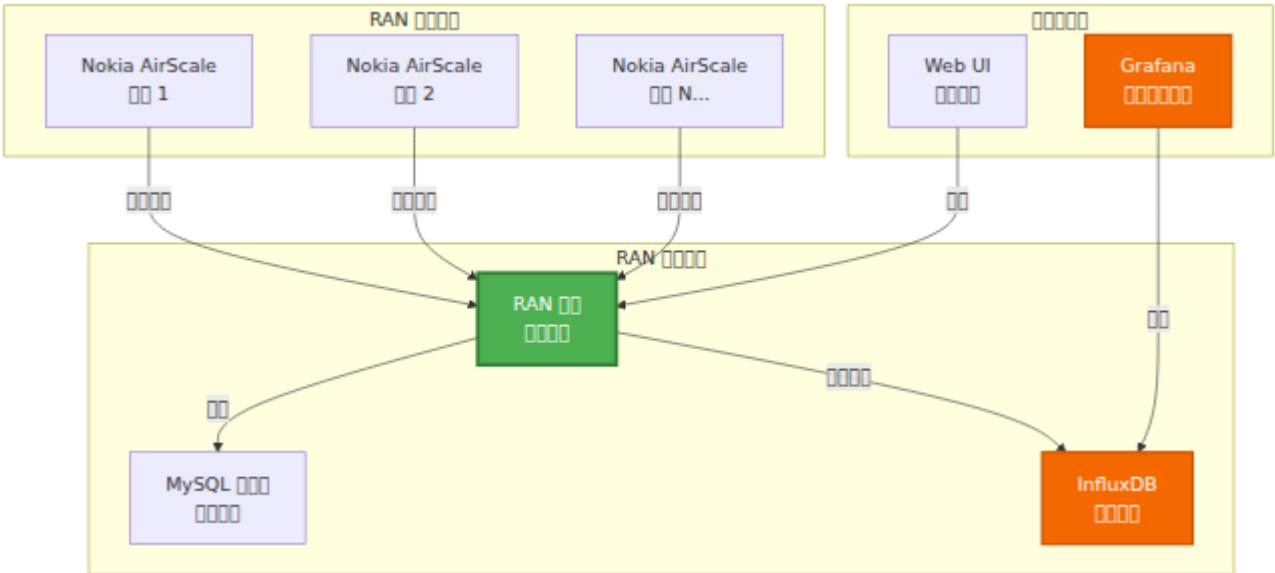
0000000000000000 Nokia 0000000

□□□□

□□□□□□



□□□□



□□□□□□□□□□□□ □□□□□□□□

Network TCE (TCE)

RAN TCE is a tool for analyzing LTE/5G network traffic and performance.

Network TCE

Network TCE is a tool for analyzing LTE/5G network traffic and performance.

- **S1-AP** - eNodeB to EPC interface
- **RRC** - Radio Resource Control
- **NAS** - Non-Access Stratum
- **PD** - PDN Protocol

Network

Network

- Network
- Network
- Network RSRP, RSRQ, SINR
- Network

Network

- Network
- Network
- Network
- Network

Network

- PCI
- Network
- Network
- Network

Network Wireshark TCE MDT

Web UI

RAN Web UI

<https://<ran-monitor-ip>:9443>

-
-
-
-

5

-
-
-
-

□□□□

□□□□□□□□□□

- □□□□□□□□□□□□□□ = □□□□□ = □□□□□ = □□□□□ = □□□□□ = □□□□
- □□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□ □□□□□□□□

□□□□

□□□□□□□□

1. 00 00000000
2. 00 0000 → 0000 ID
3. 00 000000 ID
4. 00 000000
5. 00 000000

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

[illegible]

Web UI - 100%

□□□□

□□□□□□□□□□□□

- □□□□□□□□□□□□
- □□□□□□□□
- □□/□□□□□□
- □□□□□□□□□□
- □□□□□□□□

□□□□□□□□□□ □□□□□□

📄📄📄📄

📄📄📄 InfluxDB 📄📄📄📄

- 📄📄📄📄📄📄📄📄📄
- 📄📄📄📄📄📄📄
- 📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄
- 📄📄📄📄📄📄📄📄📄

📄📄📄📄📄📄📄📄📄 📄📄📄📄📄📄

InfluxDB 簡介

時間序列數據庫? ? ? ?

- 數據庫
- 時間序列數據
- 查詢
- 數據庫引擎
- 5 分鐘上手

查詢

- $SELECT + WHERE = SQL$
- $SELECT + WHERE = InfluxDB$
- $SELECT = InfluxDB$

Web UI

Web UI

Web UI -

Grafana

Web UI Grafana

Grafana

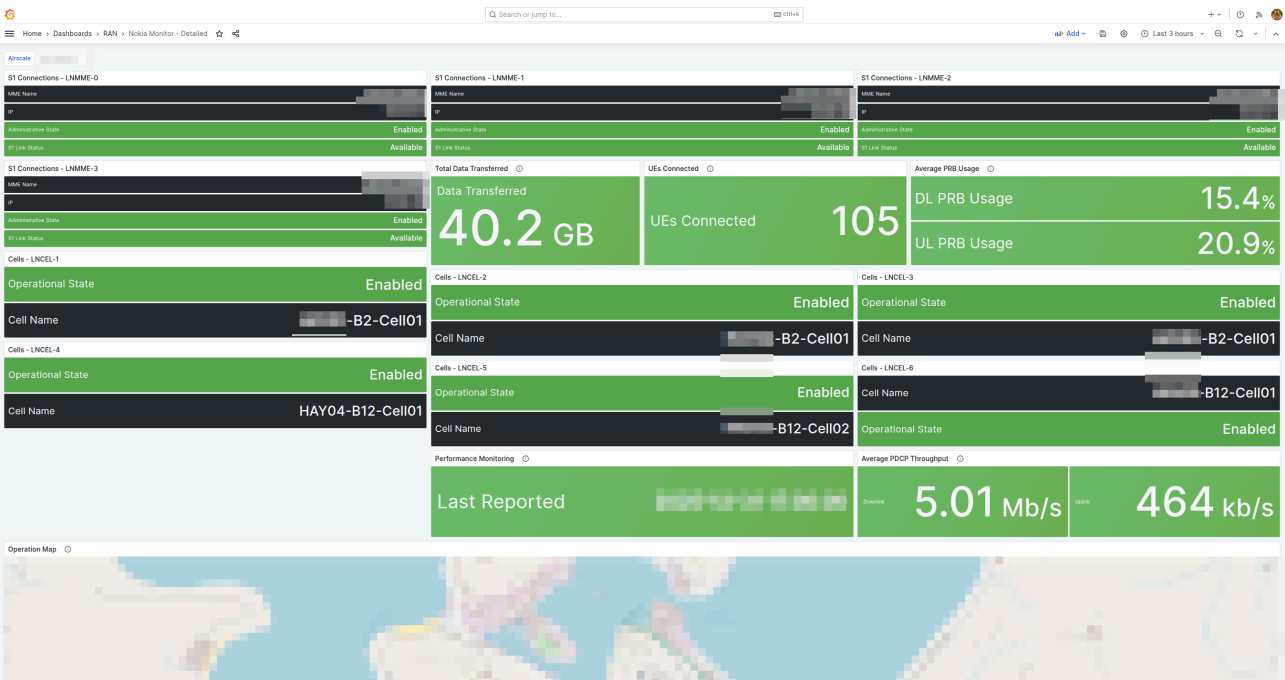
Grafana

-
- KPI
-
-

- 5G SLA 設定
- 5G SLA 監視

Web UI 画面

- 5G SLA 設定
- 5G SLA 監視
- 5G SLA 設定
- 5G SLA 監視
- 5G SLA 設定



5G SLA Grafana 5G SLA 監視

5G SLA 監視

5G SLA 監視

- 5G SLA 監視
- 5G SLA 監視
- 5G SLA 監視
- 5G SLA 監視
- 5G SLA 監視

NOC 項目

- 監視
- 設定
- 障害発生 24 時間
- 対応
- 報告

監視項目

- 稼働率
- 送信電力SINR、RSRP 値
- 送信電力RLC 値、RRC 値
- 温度
- 湿度

Nokia AirScale 項目

- PRB 使用率/総数
- 送信電力PDCP 値
- UE 数
- 温度
- 湿度
- RSSI 値
- RRC 値
- VSWR 値
- 電圧

監視項目

Grafana 監視 - 監視項目

Nokia 監視 - 監視項目

部署

部署

部署需要**5-10** 分钟

1. 部署 Web UI 界面
2. 部署数据库
3. 部署后端服务
4. 部署前端服务
5. 部署中间件

部署需要部署 Web UI 界面 - 数据库

部署

1. 部署数据库
2. 部署后端服务
3. 部署前端服务
4. 部署中间件
5. 部署中间件

部署需要部署数据库 中间件

部署

部署

1. 部署数据库
2. 部署 eNodeB 中间件
3. 部署 `config/runtime.exe`
4. 部署 RAN 中间件
5. 部署中间件
6. 部署 InfluxDB

部署

1. 安装依赖包
2. 在 `config/runtime.exs` 中配置
3. 安装数据库
4. 设置 RAN 值
5. 安装 Grafana 插件

安装依赖包

1. 安装依赖包
2. 在 `config/runtime.exs` 中配置
3. 设置 RAN 值
4. 安装数据库

安装数据库

安装 Grafana 插件 - 安装 Grafana

安装 Grafana

安装 Grafana 插件

1. 00 00000000 - 00000000
2. 0000 00
3. 00 000 → 0000 ID
4. 0000 **ID** 00 → 00000
5. 0000000000
6. 00 000000
7. 00 00000 15-30 00
8. 00 000000000000001 0000

[illegible][illegible]

--

RAN □□□□□□□□□□□□□□

📄📄📄📄📄📄**NOC**📄📄📄📄📄📄

📄📄	📄📄	📄📄📄📄
Web UI 📄📄	📄📄📄📄📄📄📄	📄📄📄📄📄📄📄
📄📄📄📄📄	📄📄📄📄📄	📄📄📄📄📄📄📄📄📄📄
📄📄📄📄📄	📄📄📄📄📄📄📄	📄📄📄📄📄📄📄📄📄📄
📄📄📄📄📄	📄📄📄📄📄	📄📄📄📄📄📄📄📄
📄📄📄📄📄	📄📄📄📄📄	📄📄📄📄📄📄📄
📄📄📄📄📄	📄📄📄📄📄	📄📄📄📄📄📄📄📄
📄📄📄📄📄📄	📄📄📄📄📄	📄📄📄📄📄📄📄📄

📄📄📄📄📄📄

📄📄	📄📄	📄📄📄
Grafana 📄📄📄	📄📄📄📄📄📄📄	📄📄📄📄📄📄📄
Nokia 📄📄📄📄	📄📄📄📄📄📄	📄📄📄📄📄📄📄
AirScale 📄📄📄	📄📄📄📄📄📄	📄📄📄📄📄📄📄
DHCP 📄📄📄	📄📄 DHCP 📄📄📄📄	📄📄 DHCP 📄📄 43 📄📄📄📄
TCE MDT 📄📄📄📄📄	MDT 📄📄📄📄 Wireshark 📄📄	📄📄📄📄📄📄📄📄📄📄
API 📄📄📄	REST API 📄📄	📄📄📄📄📄📄📄

部署環境

OS	DB	サービス
Ubuntu 20.04	MySQL	MySQL

インストール

必要な RAN 環境

1. Web UI 環境を構築
2. DB 環境を構築
3. DB 環境を構築
4. Web UI 環境を構築

インストール

1. Grafana をインストール
2. Nokia をインストール
3. DB をインストール

インストール

インストール

サービス	URL	ポート
Web UI 環境	https://<server>:9443	9443
Grafana	http://<server>:3000	3000
InfluxDB	http://<server>:8086	8086

目錄

目錄	目錄
config/runtime.exs	配置運行時環境
priv/cert/	HTTPS Web UI 的 SSL 證書
priv/external/nokia/	外部數據庫
priv/airscale_configs/	配置數據庫 XML 數據

目錄

目錄

- RAN 數據庫
- 數據庫配置
- 數據庫配置 30 天
- 數據庫 MySQL 數據

目錄

- 10 數據庫
- 10 數據庫 + 數據 Webhook 數據
- 1 數據庫
- 數據庫 InfluxDB 數據

目錄

- 數據庫 720 數據 30 天
- 數據庫
- 數據庫
- 數據 Web UI 數據

目錄 目錄

環境構築

前提条件

1. Web UI → 実行
2. 実行環境のインストール
3. 実行環境
4. 実行環境

実行環境

1. Web UI → 実行 → 実行環境
2. 実行環境のインストール
3. 実行環境
4. 実行環境のインストール
5. 実行環境
6. 実行環境

実行環境

1. 実行環境
 2. 実行 `config/runtime.exe`
 3. 実行環境 `airscales` 実行
 4. 実行 RAN 実行
 5. 実行環境のインストール
-

📁

📁📁📁📁📁

📁	📁
📁📁📁📁	📁📁📁📁📁📁
📁📁📁📁	📁📁📁📁📁📁
📁📁📁📁	📁📁📁📁📁📁
InfluxDB 📁📁	📁📁📁📁

📁📁📁📁📁

📁📁📁📁

1. 📁📁📁📁 → 📁📁
2. 📁📁📁📁📁 ping <device-ip>
3. 📁 config/runtime.exe 📁📁
4. 📁📁📁📁📁📁📁

Grafana 📁📁📁📁

1. 📁📁📁📁📁📁📁📁📁
2. 📁 InfluxDB 📁📁📁📁📁📁📁
3. 📁 InfluxDB 📁
4. 📁 Grafana 📁📁📁

Web UI 📁📁📁

1. 📁📁 9443 📁📁📁
2. 📁📁📁📁📁📁 HTTPS 📁
3. 📁 SSL 📁📁📁📁
4. 📁📁📁📁📁📁 Web UI 📁📁

無線LANの導入と運用 無線LANの導入

無線LAN

無線LANの導入

無線LANの導入

- 無線LANの導入
- 無線LANの導入
- 無線LANの導入/IPv6
- 無線LANの導入
- RAN 無線LANの導入

無線LAN

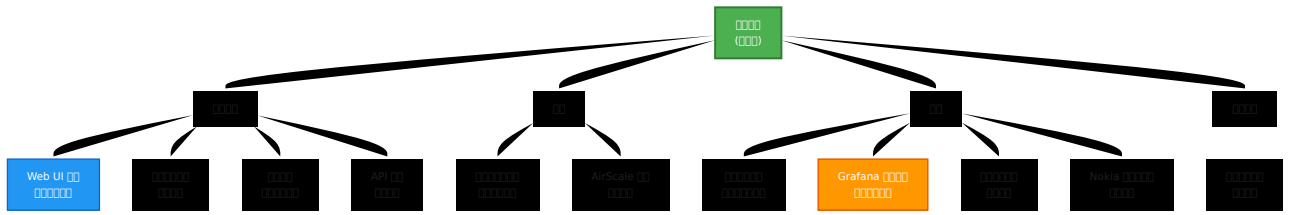
無線 RAN 無線LAN

- Omnitouch 無線LAN
- 無線LANの導入
- 無線LANの導入
- 無線LANの導入

無線LAN

1. 無線 LAN の導入
 2. 無線LANの導入
 3. 無線LANの導入
 4. 無線LANの導入
 5. 無線LANの導入
-

□□□□



Nokia AirScale

RAN □□□□□□□□

11

1. 00
2. 0000
3. 00 WebLM 00
4. 000000
5. 0000000
6. 00
7. 0000

11

RAN  RAN

PMCADM

--	--	--	--	--	--	--	--

Nokia AirScale	Omnitouch
----------------	-----------

- 
- 
- 
- 

OmniTouch OmniTouch

Overview

This document provides information about the

- **Counter** - Counter information
- **Configuration** - Configuration information
- **RAN** - RAN IP information
- **Scale** - Nokia AirScale information

Table

Item	Parameter	Value
RAN IP	RAN IP	10.179.2.139
RAN Port	Port (Port: 9076)	9076
Scale	Scale	60

WebLM

1. Web Element Manager

1. Counter
2. Configuration

http://<base-station-ip>/

https://<base-station-ip>/

□□ **2**□□□□□□□□

1. **Configuration**
2. **Configuration Management**
3. **Parameter Editor**

--	--	--	--	--	--

1000 OAM TLS

□□□□□□ (□□□) □□

1. ☐ **Current BTS configuration**
2. ☐ **CURRENT_BTS_CONF-1**
3. ☐ **MRBTS-X** (☐ X ID)
4. ☐ **oamTls** ☐
5. ☐ forced ☐ off

□□	□□□	□□□
oamTls	forced	off

이제 이 RAN 구성을 배포할 때 forced 옵션을 TLS 구성

이제 2단계로 PMCADM 구성

이제 PMCADM (이제) 구성

1. 이제 **Current BTS configuration**
2. 이제 **CURRENT_BTS_CONF-1**
3. 이제 **MRBTS-X** (이제 X 구성 ID)
4. 이제 **MNL-1** (이제)
5. 이제 **MNLENT-1** (이제)
6. 이제 **PMCADM-1** (이제)

이제 PMCADM-1 구성

이제 3단계로 구성

이제 **Structure 1** 구성

이제

項目	値	説明
TLS 有効/無効	有効	RSA (2048bit TLS)
接続先 IP アドレス	RAN 接続 IP アドレス	RAN 接続 IP (例: 10.179.2.139)
ポート番号	RAN ポート番号	9076 (既定)
タイムアウト	接続タイムアウト	60s (60秒)
SSL/TLS	有効/無効	false (無効)

設定

項目	値	説明
SDL 有効/無効	有効	1
SDL 値	接続先 IP アドレス	12345678
接続先 IP アドレス	接続先 IP アドレス 0	false (無効)

4. 接続先 IP アドレス

接続先 IP アドレス

- 接続先 IP アドレス - 接続先 IP アドレス
- 接続先 IP アドレス
 - 接続先 IP アドレス **Create Plan** 有効
 - 接続先 IP アドレス
 - 接続先 IP アドレス ID
- 接続先 IP アドレス
 - 接続先 IP アドレス **Validate Plan** 有効

- 認證 ID
- 認證碼
- 認證時間戳

4. 認證

- 按 **Activate Plan** 按
- 認證 ID
- 認證碼
- 認證時間戳

XML 格式

以下 XML 格式 PMCADM 認證信息

```
<managedObject class="com.nokia.srbts.mnl:PMCADM" distName="MRBTS-132/MNL-1/MNLENT-1/PMCADM-1" version="MNL25R1_2420_110" operation="create">
  <p name="act3gppXmlEnrichment">false</p>
  <p name="reportingIntervalPm">5min</p>
  <p name="sdlMaxUploadFileNumber">1</p>
  <p name="sdlPrimaryDestIp">10.179.2.139</p>
  <list name="rTPmCollEntity">
    <item>
      <p name="certTypeForTlsAuth">RSA</p>
      <p name="rTPmCollEntityHost">10.179.2.139</p>
      <p name="rTPmCollEntityPortNum">9076</p>
      <p name="rTPmCollInterval">60s</p>
      <p name="tlsEnabled">false</p>
    </item>
  </list>
</managedObject>
```

XML 參數說明

- `rTPmCollEntityHost` - 認證 RAN 的 IP 地址
- `rTPmCollEntityPortNum` - 端口 9076 (webhook 端口)
- `rTPmCollInterval` - 時間 (60s)
- `tlsEnabled` - 是否啟用 TLS false

- `sdlPrimaryDestIp` - RAN IP

10.179.2.139 RAN IP ID MRBTS-132

LNBTS

LNBTS (LTE NodeB) PM

actL1PM - 1

	actL1PM
	NOKLTE:LNBTS
	(true/false)
	false
	RTWP (M8005C306-339)

1 () RTWP () RTWP

- M8005C306-309: AVG_RTWP_RX_ANT_1 AVG_RTWP_RX_ANT_4
- M8005C310-313: AVG_RTWP_RX_ANT_5 AVG_RTWP_RX_ANT_8
- M8005C314: MAX_RTWP
- M8005C317-325: UL_IOT_PUSCH
- M8005C326-335: AVG_SIR

XML

```
<managedObject class="NOKLTE:LNBT" distName="MRBT-XXX/LNB-XXX" operation="update">
  <p name="actL1P">true</p>
</managedObject>
```

WebLM

- 1. Configuration > Configuration Management > Parameter Editor
- 2. MRBT-X > LNB-X
- 3. actL1P
- 4. true
- 5.

actPCMDReport -

	actPCMDReport
	NOKLTE:LNBT
	(true/false)
	false

(PCMD)

XML

```
<managedObject class="NOKLTE:LNBT" distName="MRBT-XXX/LNB-XXX" operation="update">
  <p name="actPCMDReport">true</p>
</managedObject>
```

PMCADM-1

PMCADM ()

-
-
-
-

PMCADM

reportingIntervalPm - PM

	reportingIntervalPm
	com.nokia.srbts.mnl:PMCADM
	5min, 15min, 30min, 60min
	5min 15min

- 5min** -
- 15min** -
- 60min** -

XML


```
<managedObject class="com.nokia.srbts.mnl:PMCADM" distName="MRBTS-XXXX/MNL-1/MNLENT-1/PMCADM-1" operation="update">
  <p name="reportingIntervalPm">5min</p>
</managedObject>
```

WebLM

- Configuration > Configuration Management > Parameter Editor
- MRBTS-X > MNL-1 > MNLENT-1 > PMCADM-1
- reportingIntervalPm
- 5min
-

certTypeForTlsAuth - TLS

- (RSA, DSA, ECDSA)
- TLS
- RSA
- tlsEnabled = true

rTpmCollEntityHost -

- IP (IPv4 IPv6)
- IP
-
- 10.179.2.139
-

rTpmCollEntityPortNum -

- (1-65535)
- TCP

- 9076
- RAN

rTpmCollInterval -

- ()
-
- 15s, 30s, 60s, 300s, 900s, 1800s
- 60s
- 60s 15s

tlsEnabled - TLS

- (true/false)
-
- false
-

sdIMaxUploadFileNumber - SDL

-
-
- 1
-

sdINonce - SDL

- (8)
- SDL
- 12345678
-

suppressZeroValueCount -

- (true/false)
-
- false

- `is_connected` false 状態にリセットされる



接続が成功すると RAN モニタリング

2. RAN モニタリング Web UI

1. RAN モニタリング Web UI: `http://<ran-monitor-ip>:4000/`
2. **BTS Status** を確認
3. 接続が成功すると
4. 接続が成功すると **Connected** (緑色) — 接続が成功すると PM モニタリング
5. 接続が成功すると **Waiting for PM Data** (黄色) (接続が成功すると 15 秒間 PM モニタリング)

接続が成功すると

- **Connected** (緑色)
- 接続が成功すると

- 0000 0000 IP 00
- 0000 00000000

00000000 **Pending**

000000000000000000000000

00000000

- 0000 ID 0000000000
- RAN 00000000000000
- 00000000

00000000000000

00000000 RAN 0000

配置項目

- PMCADM の IP アドレス
- ポート番号
- デフォルトのポート番号 8080
- RAN の IP アドレス

インストール

InfluxDB のインストール

- RAN の Web UI から **InfluxDB Status**
- インストールを確認
- Performance Metrics** を確認
- Last Update** を確認

□□□□

- **Performance Metrics**□ □□□□□
- **Configuration**□ □□□□
- **Alarms**□ □□□□□□□□□□ 0

□□□□□□

1. □□□ **Data Retention Policy** □□
2. □□□□□□
3. □□ **Performance Metrics**□**Configuration** □ **Alarms** □□

□□□□

□□□□□□ **RAN** □□□

□□□ □□□□□□ BTS □□□□

□□□□□

1. □□□□□□□

```
ping <base-station-ip>
```

2. □□ **RAN** □□□□□

- □□□□□□□□ `config/runtime.exs`
- □□ IP □□□□□□□□
- □□□□□□□

3. 检查 RAN 配置

- 检查 **Live Logs** 配置
- 检查配置
- 检查配置

4. 检查配置

- 检查 WebLM 和 PMCADM-1 配置
- 检查 RAN 和 IP 配置
- 检查配置 9076

检查“Pending”配置

检查配置“Pending”

检查配置

1. 检查配置

- 检查 RAN 配置 ID 配置
- 检查配置

2. 检查配置

- 检查 runtime.exs 配置
- 检查/配置

3. 检查配置

- 检查 30-60 配置
- 检查配置

检查

检查 “enetunreach”配置

检查

1. 設定

- 設定 RAN 設定
- 設定
- 設定 VLAN/subnet 設定

2. 設定

- 設定 9076 設定 (設定)
- 設定 8080 設定 (設定 SOAP API 設定)
- 設定 ACL 設定
- 設定 RAN 設定 iptables 設定

3. 設定 RAN 設定

```
# 設定 SOAP API 設定 webhook 設定  
netstat -tlnp | grep -E '8080|9076'
```

設定 RTWP 設定 1 設定

設定 設定 InfluxDB 設定 RTWP 設定 (M8005C306-339)

設定 LNBTS 設定 `actL1PM` 設定 `false` 設定 1 設定

設定

1. 設定

- 設定
- 設定 LNBTS 設定 `actL1PM`
- 設定 `false` 設定

2. 設定 1 設定 PM

- 設定 WebLM
- 設定 MRBTS-X > LNBTS-X
- 設定 `actL1PM` 設定 `true`
- 設定

3. 設定項目

- 収集間隔 (単位 60 秒)
- InfluxDB 接続 ID M8005C306-309 設定
- RTWP 設定項目

設定項目

- `actPCMDReport` - 収集項目 (収集項目)

InfluxDB 接続

接続 設定項目

設定項目

1. 接続項目

- 接続 ID PMCADM-1 rTpmCollInterval 設定
- 収集間隔
- InfluxDB 接続

2. InfluxDB 接続

- InfluxDB 接続
- “Connected” 接続項目
- 接続項目

3. RAN 接続項目

- InfluxDB 接続
- 接続項目
- API 接続項目

TLS/接続

接続 TLS 接続項目

📌

1. 📌

- 📌
- 📌 RAN 📌 CA 📌

2. 📌 **TLS**

- 📌 tlsEnabled 📌 false
- 📌
- 📌 TLS

3. 📌

- 📌
- 📌
- 📌

📌

📌

- 📌 - 📌 RAN 📌
- 📌 - RAN 📌
- **Nokia Counter Reference** - 📌
- **Grafana** 📌 - 📌
- **API** 📌 - 📌 REST API 📌
- 📌 - 📌

📌

- **config/runtime.exs** - RAN 📌

□□

□□□□□□□□□□

1. □□ RAN □□□□□□□□
2. □□□□□□□□□□ Nokia □□□□
3. □□□□□□□□□□
4. □□□□□□□□□□

無線LANの基礎

無線LANの概要

無線LANの歴史

無線LANの種類

1. IEEE 802.11a
 2. IEEE 802.11b
 3. IEEE 802.11g
 4. IEEE 802.11n
 5. IEEE 802.11ac
 6. IEEE 802.11ad
 7. IEEE 802.11ah
 8. IEEE 802.11ax
-

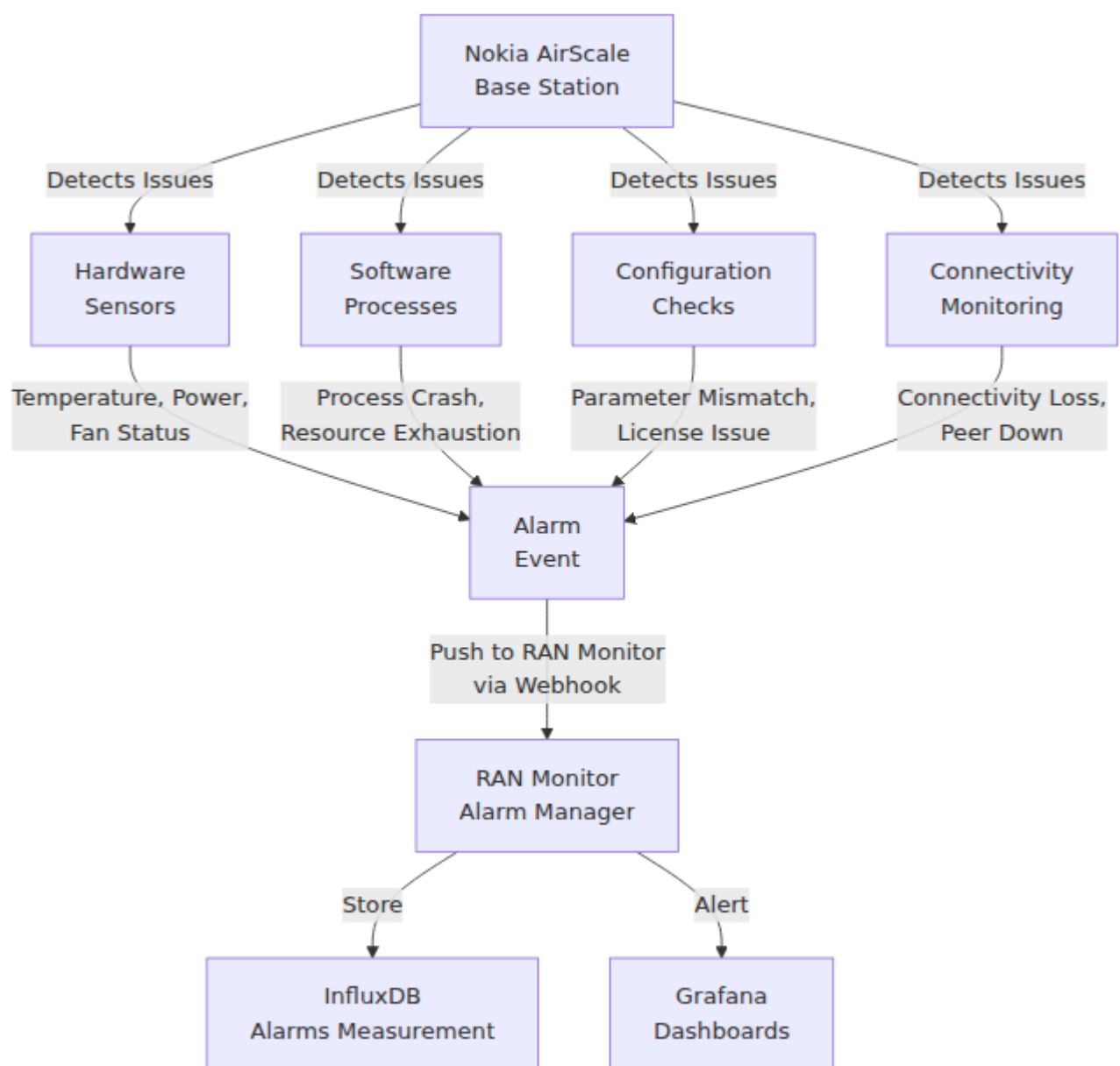
無線LANの規格

無線LANの規格“IEEE 802.11”は、IEEE（Institute of Electrical and Electronics Engineers）が制定する。AirScale は、無線LANの規格IEEE 802.11a/b/g/n/ac/ah/axを対応するRAN Monitor を提供している。

無線LANの規格

4G

□□□□



□□□□□□

□□□□□□

項目	値	単位
識別 ID	a1b2c3d4-e5f6-...	文字列
状態	正常	文字列
コメント	"テスト"	文字列
タグ	"S1 タグ"	文字列
位置情報	/BSC-1/BTS-23/Cell-A	位置情報DN
時刻	2025-12-10 14:23:45	時刻
ページ	1 / 1	ページ

□□□□□□

□□□□

□□□

□□□□

□□□
(□□□)

□□□□

□□□□
(□□□□□□)

□□□□

□□□
(□□□□□)

□□□□

□□□
(□□□□)

□□□
□□

ログ

```
14:23:45 UTC - 実行
└─> 実行環境確認
└─> 実行環境"S1 実行"実行

14:23:47 UTC - 実行 RAN Monitor
└─> NE3S webhook 実行
└─> 実行 InfluxDB
└─> 実行

14:23:50 UTC - 実行
└─> Grafana 実行
└─> 実行 NOC 実行 Slack 実行
└─> 実行 PagerDuty 実行

14:24:15 UTC - 実行
└─> NOC 実行
└─> 実行

14:28:35 UTC - 実行
└─> 実行
└─> 実行
└─> RAN Monitor 実行"実行"

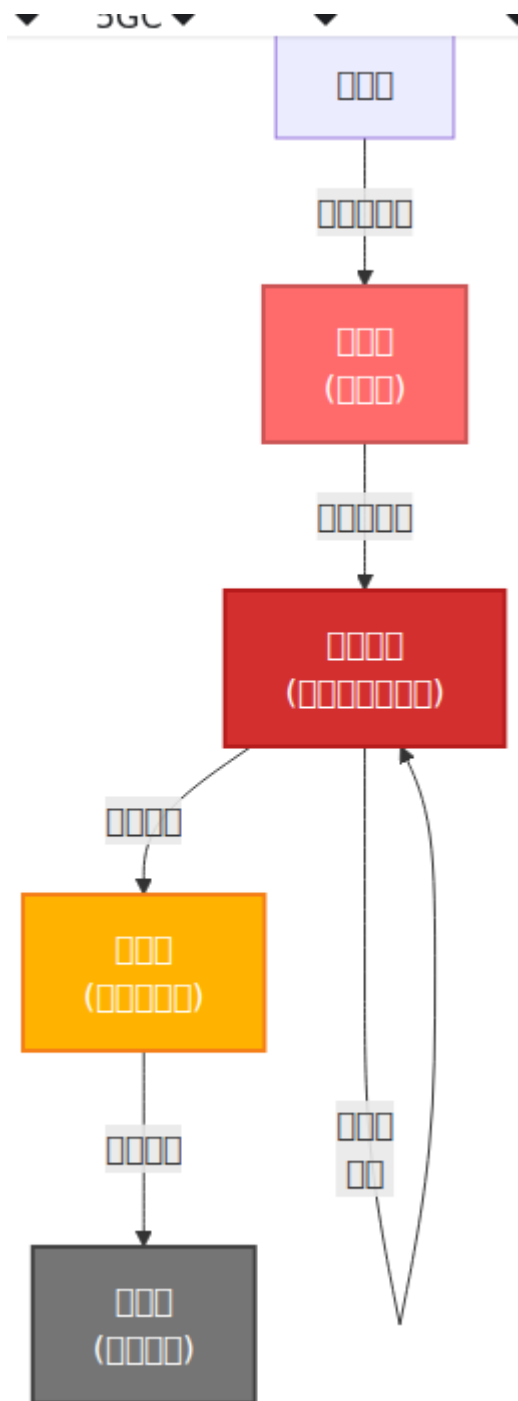
14:28:36 UTC - 実行
└─> 実行5 実行 51 実行
└─> 実行 SLA 実行
└─> 30 実行
```

実行

RAN Monitor 実行

実行

実行 実行



000

- 0000000000000000
- 00000000000000
- 00000000S1 000
- 00000000
- 00000000

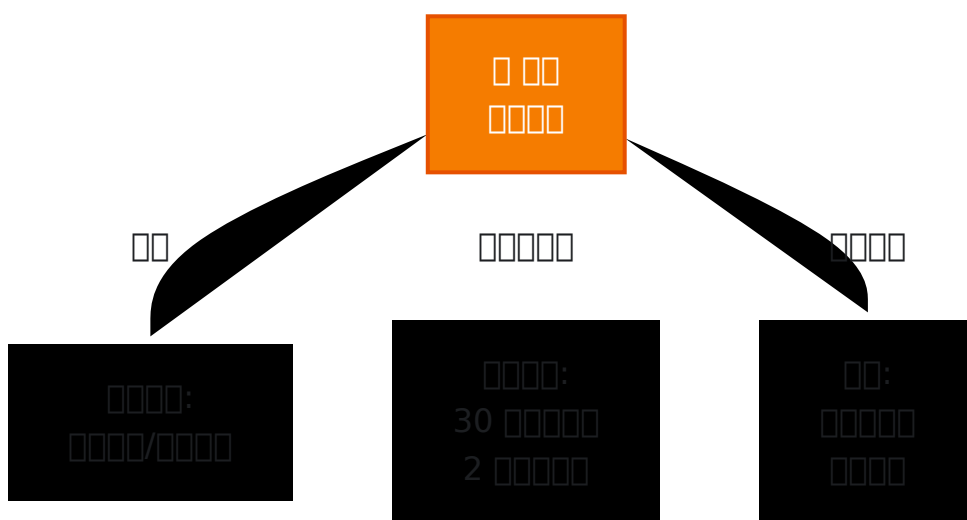
000

- 時間切れの可能性
- 時間切れの可能性
- 時間切れ
- 時間 SLA 時間切れの可能性

時間 SLA < 15 分

時間切れ

時間 時間切れの可能性



時間

- 時間切れ < 95% 時間 15 分
- 時間切れ < 95%
- DL/UL 時間切れ > 95% 時間切れ
- RLC 時間 > 5%
- 時間切れ
- 時間切れ E1/T1 時間切れ

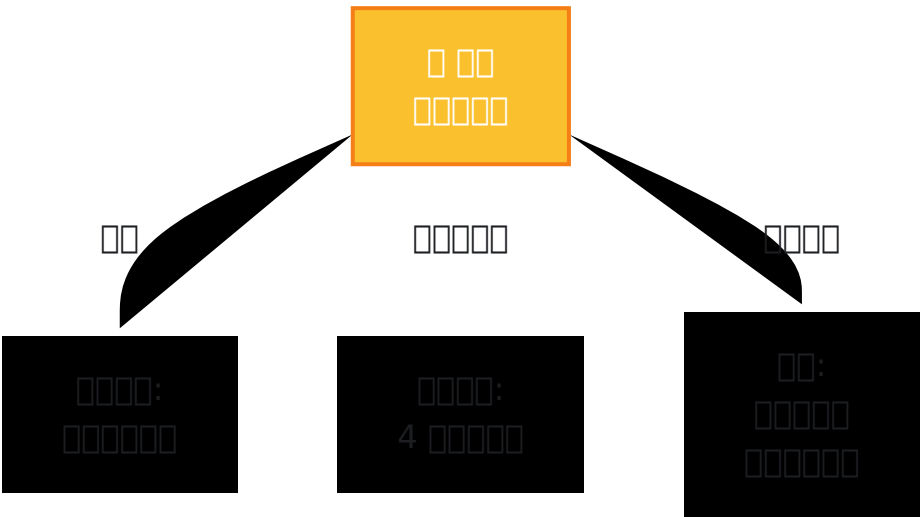
時間

- 時間 NOC 時間 + 時間切れ
- 時間切れ
- 時間 30 時間切れ
- 時間切れ/時間切れ

目標 **SLA** < 30 分

目標

目標 達成率



目標

- 達成率 95-98%
- SLA
- 目標
- 達成率
- SLA
- 目標 < 5 分

目標

- 達成率
- SLA
- 目標
- 達成率

目標 **SLA** 目標

項目別

項目 評価項目

項目

- 評価項目 > 98% 評価項目
- 項目/評価項目評価項目
- 評価項目 60-70%
- 評価項目評価項目
- 評価項目評価項目

項目

- 評価項目
- 評価項目
- 評価項目

項目

項目 評価項目

項目

- 評価項目
 - 評価項目MTTR
 - 項目 SLA 項目
 - 評価項目
-

□□□□

□□□□

□□□□□□

□□□□

- S1 □□□□ → MME/SGW □□□□
- □□□□□□ → IP □□□□
- USIM □□□□□ → HSS □□□□
- NTP □□□□ → □□□□□□□□

□□□□□□□□□□□□

□□□

1. □□□□□□□□□□
2. □□□□□□□□□□□□
3. □□□□□□□□□□
4. □□□□□□□□□□

□□□□□□□

□□□□□□□□

□□□□

- □□□□ → □□□□□□
- □□□□ → UPS/PSU □□
- □□□□ → □□□□□□
- □□□□□□ → □□□□□□
- □□□□ → □□□□□□

□□□□□□□□□□□□

□□□

1. □□□□□□□□□□□□
2. □□□□□□
3. □□□□□□□□
4. □□□□□ CPU □□□

環境構築

前提条件

必要な環境

- Python 3.7以上 → Python 3.7 OOM
- CPU 1コア以上 → 1コア
- メモリ 2GB以上 → 2GB以上
- ディスク 10GB以上 → 10GB

インストール手順

手順

1. Python 3.7インストール
2. Python 3.7インストール
3. CPU/メモリ/ディスク
4. ディスク

環境構築

前提条件

必要な環境

- Python 3.7以上 → Python 3.7/10GB
- DL 1コア以上 → 1コア
- メモリ 2GB以上 → 2GB以上
- ディスク 10GB以上 → RF ディスク

インストール手順

手順

1. Python 3.7インストール
2. Python 3.7インストール
3. RF ディスク
4. ディスク

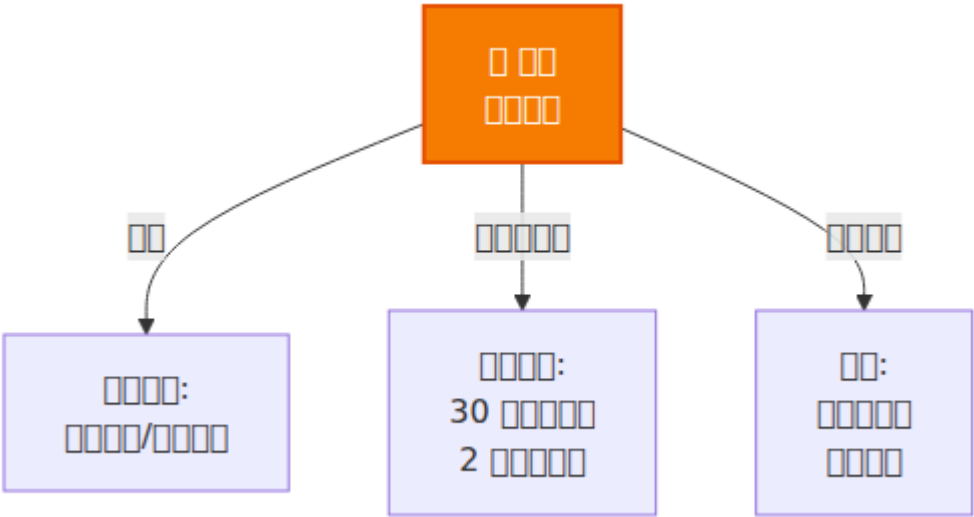
□□□□

-
-
- □□□□ → □□□□
 - □□□□ → □□□□
 - □□□□□□ → □□□□
 - □□□□□□ → □□□□□□□□

-
-
1. □□□□□□
 2. □□□□□□□□
 3. □□□□□□□□□□
 4. □□□□□□□□

□□□□□□□□

□□□□□□



□□□□□□□□□□□□

□□□□□□□□□□□□

設定項目

- ユーザー ID 設定
- パスワード設定
- デバイス名 DN設定/日/時刻
- ログレベル設定
- ログ出力先設定
- ログ保存期間設定

確認

- RAN Monitor Web UI → 確認
- Grafana → 確認
- InfluxDB → 確認

動作確認

動作確認項目

確認

- ユーザー AirScale 設定
- デバイス名設定
- RAN Monitor 確認
- ログ出力先設定

動作確認

動作確認項目

- “DL 送信率”確認
 - └ ユーザー DL 送信率 > 95%
 - └ デバイス名設定
 - └ デバイス名設定
 - └ デバイス名設定
 - └ デバイス名設定

□□□

- Grafana → □□□□□□□
- Web UI → □□□□□□□ → □□❓❓□
- InfluxDB □□□□□□□□□

□□□□□□□□□□□□

□□□□□□□□□□□□□□

□□□□

└ □□□□□□□ 4 □□□
└ □□□□□□□ 24 □□□
└ □□□□□□□□□ 7 □□
└ □□□□□□□ 7 □□
└ □□□□□□□ 7 □□
└ □□□□□□□□□□□□

□□□

- RAN Monitor → □□□□
- □□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□

□□□□□□□□□□□□

□□□□□□□□□□□□□

□□□□□□“□□□□□□”□□

```

□□□□□□□
├
├─ □□□□□□□□
│   ├── □ → □□□□□□□□□□□□
│   └── □ → □□
├
├─ □□□□□□□□□□□□
│   ├── □□□□ → □□□□/□□□□
│   └── □□□□ → □□
├
├─ □□□□□□□□□□
│   ├── □ → □□□□□□□□□□□□
│   └── □ → □□
├
├─ □□□□□□□□□□□□
│   ├── □ → □□□□□□□□□□□□
│   │   └── → □□□□□□□□□□□□
│   └── □ → □□□□□□□□□□
│       └── → □□□□□□□□□□□□ DSP □□
├
└─ □□□□□□□□□□□□
    ├── → □□□□□
    ├── → □□□□□□
    └── → □□□□□

```

□□□□□□□□□□□□

□□□□□□□□□□□□□□

□□□□□□□□

項目	単位	説明
項目	単位/項目	説明
項目	項目	説明
項目	単位/項目	説明
項目	単位/項目	説明
項目	項目	項目 BGP 説明

項目

項目

項目

☐ RAN Monitor 項目

☐ 項目

☐ 項目/項目

☐ 項目

☐ 項目

☐ 項目

☐ 項目 > 30 項目

項目

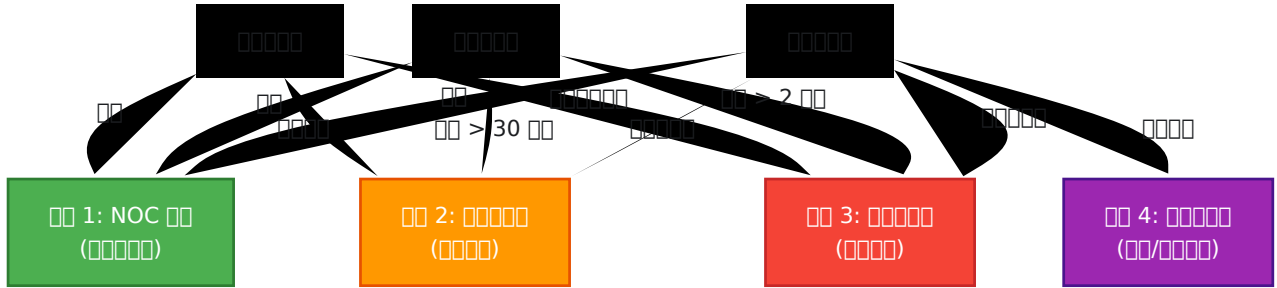
項目

項目

- 項目
- 項目
- 項目 SLA 項目
- 項目
- 項目

--	--	--	--

--	--	--	--



□ □ □ □ □ □

□□□□□□□□□□□□□□ 2□

- 15 分钟
- 30 分钟
- NOC 分钟
- 分钟/天
- > 1 天

□□□□□□□□□□□□ 3□

- 2 1
- 30
-
-
-

□□□□□□ **?** **?** **?** □□□□ **4** □□□□□□□□

- 1000000PSU1CMON 00
- 0000000000000000
- 000/0000
- 0000000000000000
- 000000000000

0000

0000 2 0000

00000 - [000] - [00] - [00]

00000[2025-12-10 14:30 UTC]

00000[15 00]

000[SITE_A_BS1]

000[000000]

000

- 00 A1 00000
- 000000000000
- 00 ping 00

0000000

- 000000
- 00000000
- 0000000
- 00000

000

- 00000000%
- 00000
- 0000000

00000

- 000000000
- 00000000000

0000000

- 0000
- 000000000
- 0000000000

00000[000000]

障害発生時の対応

SLA 確認

障害発生時の SLA 確認

障害発生

└ 障害発生14:23:45 ← 障害発生
└ 障害発生14:23:47 2 分 ← 障害発生
└ 障害発生14:23:50 5 分 ← 障害発生
└ 障害発生14:24:15 30 分 ← 障害発生
└ 障害発生14:24:15 → 14:28:00 3.75 時間
└ 障害発生14:28:00 → 14:28:35 35 分 + 障害発生
└ 障害発生14:28:36 ← 障害発生
└ 障害発生5 分 51 分

SLA 確認

└ 障害発生2 分
└ 障害発生30 分
└ 障害発生5 分 51 分
└ SLA 障害発生< 15 分以内

対応

障害発生時の対応

障害発生

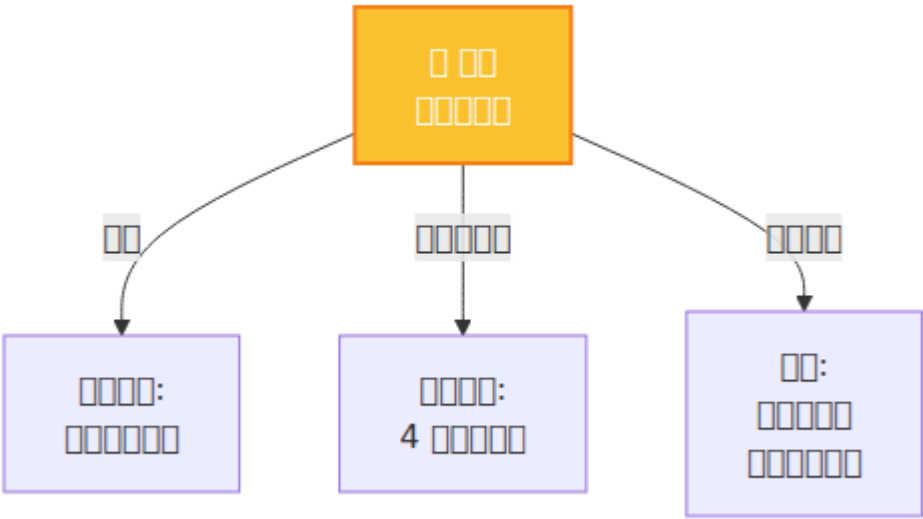
- 障害発生時の対応
- 障害発生時の対応
- 障害発生時の対応
- 障害発生時の対応
- MTTR 障害発生時の対応

対応

- Grafana → 障害発生
- 障害発生

- 時間/時間 MTTR

時間時間



時間時間

時間時間

1. 時間時間

- 時間時間時間時間時間時間
- 時間時間時間時間時間時間時間
- 時間時間
- 時間時間時間

2. 時間時間

- 時間時間時間時間
- 時間時間時間時間時間
- 時間時間時間時間時間
- 時間時間時間時間時間

3. 時間時間

- 
- 
- 
- 

4. □□□□

- 
- 
- 
- 

--	--	--	--	--	--

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

111

□□□[□□□□□]
□□□[□]
MTTR□[5-15 □□]
SLA □□□[30 □□□□□]

-
- □□□“□□□□□”
 - □□□□□□□
 - □□□□□□□□ 0%

- < 5 □□□□
1. □□□□□□ ping□
 2. □□□□□□□□□
 3. □□□□□□□□□□□□□□□

-
- 1□□□□□□□□
- ping □□□ping 192.168.1.100
 - □□□□□□ → □□□□□□

- 2□□□□□□
- □□□□□□□□□□
 - □□□□□□ LED
 - □□□□□□□

- 3□□□□□□
- □□□□□□□□□□
 - □□ 60 □□□□
 - □□□□□□□□□

- 4□□□□□□□□
- □□□□□ 2
 - □□□□□□□
 - □□□□□

-
- □□ > 15 □□ → □□□ [□□□□□□]
 - □□ > 30 □□ → □□□ [□□□□□□]
 - □□□□□□ → □□□□□□□□

-
- □□□□□□□□□□

- □□□□□□□□
- □□□□□□□□

API 開発ガイド

このガイドは RAN の REST API について説明します。

このガイドは RAN の REST API について説明します。

目次

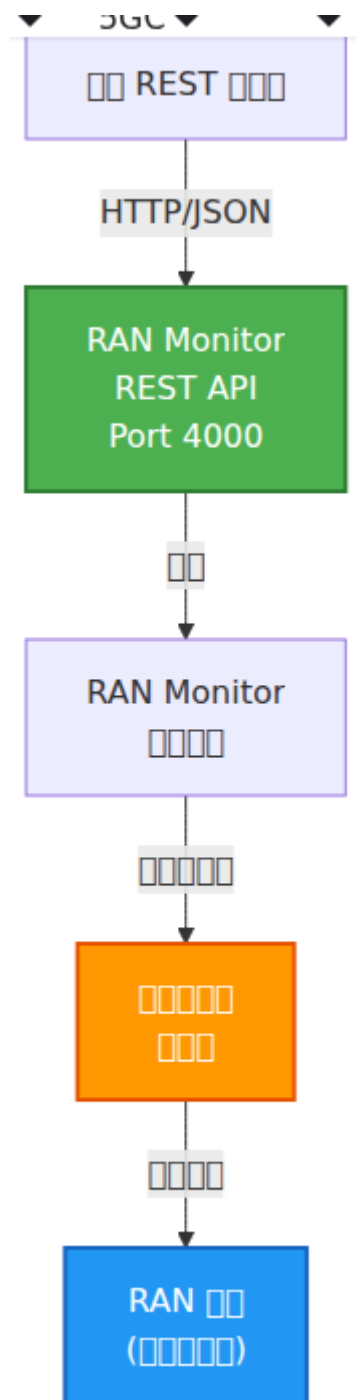
1. 概要
2. API 仕様
3. 認証
4. エラー
5. データ
6. ログ
7. テスト
8. トラブルシューティング
9. API 変更履歴

概要

RAN Monitor は REST API を提供し、RAN の状態を監視するための API を提供します。

- 概要 - 監視対象の RAN
- 監視 - 監視対象の RAN
- テスト - 監視対象の RAN
- データ - 監視対象の RAN
- ログ - 監視対象の RAN

API



API

- **RESTful** - HTTP (GET, POST, PUT, DELETE)
- **JSON** - JSON
- - API
- -

- **API** - **API**
- **API** - **API**

API

API

API RAN Monitor API RAN Monitor API

API

1. API 
2. API
3. API
4. API

API

API RAN Monitor API

- **API**
 - API **Authorization: Bearer <api-key>**
 - OAuth2
 - API/VPN
- **API**
 - API
 - API
- **API**
 - API
 - API

API

REST API

RAN Monitor API

```
/api/v1/
├── /devices                                #
│   ├── GET                               #
│   ├── POST                              #
│   └── /:id                              #
│       ├── GET                           #
│       ├── PUT                           #
│       ├── DELETE                         #
│       ├── /sessions                     #
│       ├── /config                       #
│       ├── /alarms                       #
│       └── /metrics                       #
├── /operations                            #
│   ├── /get                              #
│   ├── /set                              #
│   └── /search                            #
└── /health                               #
    ├── /status                           #
    └── /devices/:id/ping                 #
```

URL

```
http://<ran-monitor-ip>:4000/api/v1
```

□□□□

□□□□

□□□□□

GET /api/v1/devices

□□□

```
{
  "data": [
    {
      "name": "SITE_A_BS1",
      "address": "192.168.1.100:8080",
      "registration_status": "registered",
      "session_status": "active",
      "bts_status": "connected",
      "vendor": "Nokia",
      "agent_type": "AirScale",
      "software_release": "BSC-2250.5.0",
      "hardware_version": "2.0",
      "agent_unique_id": "airscale-001",
      "manager_vendor": "Nokia",
      "manager_release": "NE3S-5.0",
      "license_required": false,
      "retention_hours": 720,
      "last_connected_at": "2025-12-10T14:30:00Z",
      "disconnected_at": null,
      "last_error_code": null,
      "last_error_string": null,
      "data": {
        "pm_points": 45320,
        "pm_last_write": "2025-12-10T14:15:00Z",
        "config_points": 1200,
        "config_last_write": "2025-12-10T14:30:00Z",
        "alarm_points": 15
      },
      "pm_warning": null
    }
  ],
  "meta": {
    "total": 50,
    "page": 1,
    "per_page": 25,
    "total_pages": 2
  },
  "status_counts": {
    "total": 50,
    "connected": 45,
    "waiting_for_pm": 2,
    "pending": 1,
    "registering": 0,
  }
}
```

```
    "disconnected": 2
  }
}
```

□□□□□□

```
GET /api/v1/devices/:id
```

□□□

```
{
  "device": {
    "id": "nokia_bs1",
    "name": "SITE_A_BS1",
    "vendor": "Nokia",
    "address": "192.168.1.100",
    "registration_status": "registered",
    "registration_key": "base64_encoded_key",
    "session_id": "nonuniquesession",
    "session_expiry": "2025-12-11T14:30:00Z",
    "device_info": {
      "type": "AirScale",
      "software_release": "5.0.0",
      "hardware_version": "2.0",
      "agent_unique_id": "airscale-001"
    }
  }
}
```

□□□□

```
PUT /api/v1/devices/:id/register
```

```
Content-Type: application/json
```

```
{
  "address": "192.168.1.100:8080",
  "web_username": "admin",
  "web_password": "password",
  "webhook_url": "http://manager.example.com:9076/webhook",
  "private_key_path": "/etc/certs/private.key",
  "public_key_path": "/etc/certs/public.key"
}
```

응답

```
{
  "result": "Success",
  "registration_key": "base64_encoded_nonce",
  "device_id": "nokia_bs1",
  "message": "완료되었습니다"
}
```

상태코드

- 200 - 성공
- 400 - 잘못된 요청
- 409 - 이미 존재함
- 500 - 서버 오류

요청

```
DELETE /api/v1/devices/:id
```

응답

```
{
  "result": "Success",
  "message": "成功",
  "device_id": "nokia_bs1"
}
```

成功

成功

```
PUT /api/v1/devices/:id/sessions
Content-Type: application/json
```

```
{
  "session_id": "session_unique_identifier"
}
```

成功

```
{
  "result": "Success",
  "session_id": "session_unique_identifier",
  "session_timeout": 86400,
  "expires_at": "2025-12-11T14:30:00Z"
}
```

成功

- 成功24 分
- 成功
- 20 成功

成功

```
GET /api/v1/devices/:id/sessions
```

{} {}

```
{
  "session": {
    "active": true,
    "session_id": "session_unique_identifier",
    "expires_at": "2025-12-11T14:30:00Z",
    "time_remaining_seconds": 82400,
    "last_activity": "2025-12-10T14:30:00Z"
  }
}
```

{} {} {} {} {} {}

POST /api/v1/devices/:id/sessions/keep-alive

{} {}

```
{
  "result": "Success",
  "new_expiry": "2025-12-11T14:30:00Z"
}
```

{} {} {} {}

{} {} {} {}

{} {} {} {} {} {} {} {} {} {}

```
PUT /api/v1/devices/:id/config/upload
Content-Type: application/json
```

```
{
  "filter": {
    "uploadType": "configuration",
    "objects": [
      {
        "sdn": "/BSC-1/BTS-23/*",
        "depth": 100
      }
    ],
    "objectClass": ""
  }
}
```

□□□

```
{
  "result": "Success",
  "configuration": {
    "timestamp": "2025-12-10T14:30:00Z",
    "device_id": "nokia_bs1",
    "parameters": {
      "/BSC-1/BTS-23": {
        "BtsBasics": {
          "BtsName": "CELL_A",
          "BtsType": "MACRO",
          "EnvironmentalSpecifications": {
            "TemperatureRange": "Industrial"
          }
        },
        "CarrierAggregation": {
          "CarrierAggregationCapability": true,
          "MaxUECarriers": 5
        }
      }
    }
  }
}
```

[illegible]

Content-Type: application/json

111

111

111

- INVALID_FILE - config_file 0000000000

- `TRANSPORT_ERROR` - 网络传输错误
- `SESSION_ERROR` - 网络 NE3S 错误
- 网络 SOAP 接口调用失败

请求示例

```
PUT /api/v1/devices/:id/config/set
Content-Type: application/json

{
  "parameter_path": "/BSC-1/BTS-23/BtsBasics/BtsName",
  "value": "NEW_CELL_NAME",
  "value_type": "string"
}
```

响应

```
{
  "result": "Success",
  "parameter": "/BSC-1/BTS-23/BtsBasics/BtsName",
  "old_value": "CELL_A",
  "new_value": "NEW_CELL_NAME",
  "applied_at": "2025-12-10T14:30:45Z"
}
```

返回结果

項目	単位	値	説明
BtsName	文字列	"SITE_A_Cell_1"	基地局名
MaxUEsServed	整数	256	最大接続可能なUE数
CellTXPower	dBm	40 (dBm)	送信電力
EnableCarrierAgg	ブール値	true	キャリアアグリゲーションの有効化
HandoverHysteresis	dB	3 (dB)	ハンドオーバーヒステリシス

APIリクエスト

```
GET /api/v1/devices/:id/config/history?limit=10&days=7
```

レスポンス

```
{
  "history": [
    {
      "timestamp": "2025-12-10T14:30:45Z",
      "change_type": "parameter_modified",
      "parameter": "/BSC-1/BTS-23/BtsBasics/BtsName",
      "old_value": "CELL_A",
      "new_value": "NEW_CELL_NAME",
      "reason": "設定変更"
    }
  ]
}
```

結論

確認事項

今後の対応

```
PUT /api/v1/devices/:id/metrics/upload
Content-Type: application/json
```

```
{
  "filter": {
    "uploadType": "measurement",
    "objects": [
      {
        "sdn": "*",
        "depth": 100
      }
    ]
  }
}
```

□□□

```
{
  "result": "Success",
  "metrics": {
    "timestamp": "2025-12-10T14:30:00Z",
    "measurement_interval": 300,
    "counters": [
      {
        "id": "M1C1",
        "name": "DL □□□□□",
        "value": 125.4,
        "unit": "Mbps",
        "cell_dn": "/BSC-1/BTS-23/Cell-1"
      },
      {
        "id": "M1C2",
        "name": "UL □□□□□",
        "value": 89.2,
        "unit": "Mbps",
        "cell_dn": "/BSC-1/BTS-23/Cell-1"
      }
    ]
  }
}
```

□□□□□□

```
PUT /api/v1/devices/:id/alarms/upload
Content-Type: application/json
```

```
{
  "filter": {
    "uploadType": "active_faults"
  }
}
```

□□□

```
{
  "result": "Success",
  "alarms": [
    {
      "alarm_id": "a1b2c3d4",
      "severity": "Critical",
      "probable_cause": "□□□□□",
      "specific_problem": "□□□□",
      "affected_dn": "/BSC-1/BTS-23/Cell-1",
      "event_time": "2025-12-10T14:15:30Z",
      "description": "□□ 1 □□□□□□□□□□"
    }
  ]
}
```

□□□□□□

```
PUT /api/v1/devices/:id/topology/upload
Content-Type: application/json
```

```
{
  "filter": {
    "uploadType": "topology",
    "objects": [
      {
        "sdn": "*",
        "depth": 100
      }
    ]
  }
}
```

□□□

```
{
  "result": "Success",
  "topology": {
    "device_dn": "/BSC-1",
    "managed_elements": [
      {
        "name": "BTS-23",
        "type": "□□□□□□",
        "dn": "/BSC-1/BTS-23",
        "cells": [
          {
            "name": "Cell-1",
            "type": "EUTRANCell",
            "physical_cell_id": 100,
            "frequency": 2110
          }
        ]
      }
    ]
  }
}
```

□□□□

Ping □□

```
PUT /api/v1/devices/:id/ping
```

□□□

```
{
  "result": "Success",
  "device_id": "nokia_bs1",
  "latency_ms": 45,
  "status": "□□"
}
```

□□□□□□

```
GET /api/system-metrics
```

□□□□□□□□□□PM □□□□□□□□ InfluxDB □□□□□□□□□□□□

□□□

```

{
  "data": {
    "batch_writer": {
      "queue_depth": 120,
      "queue_max": 50000,
      "flush_count": 8452,
      "points_written": 1250000,
      "bytes_written": 890000000
    },
    "pm_spool": {
      "pending_files": 3,
      "files_processed": 1420,
      "files_failed": 0,
      "points_written": 980000,
      "points_filtered": 4200000,
      "bytes_read": 650000000,
      "last_file_ms": 245,
      "started_at": "2025-12-10T08:00:00Z"
    },
    "influxdb_stats": [
      {
        "name": "2045_Booker",
        "total": 12500,
        "last_write": "2025-12-10T14:15:00Z",
        "performance_metrics": 11000,
        "configuration": 1200,
        "alarms": 300
      }
    ],
    "system": {
      "memory_total": 524288000,
      "memory_processes": 312000000,
      "memory_binary": 890000000,
      "memory_ets": 45000000,
      "process_count": 1250,
      "run_queue": 0
    }
  }
}

```

□□□□□□

GET /api/v1/health/status

{} {}

```
{
  "status": "healthy",
  "devices": {
    "total": 50,
    "registered": 48,
    "active_sessions": 45,
    "unreachable": 2
  },
  "database": {
    "influxdb": "connected"
  },
  "timestamp": "2025-12-10T14:30:00Z"
}
```

{} {} {} {}

{} {} {} {} {} {}

Nokia eNodeB {} {} {} {} {} {} {} {}


```

/SystemFunctions
├── /BSC-1 (□□□□□)
│   ├── /BTS-23 (□□□□□□)
│   │   ├── BtsBasics (□□□□□□□□)
│   │   ├── /Cell-1
│   │   │   ├── CellCommonData
│   │   │   ├── CellAdvanced
│   │   │   └── CarrierAggregation
│   │   └── /Cell-2
│   │       └── ...
│   └── /Connectivity
│       ├── S1Interface
│       ├── X2Interface
│       └── NetworkConfiguration

```

□□□□□□□

□□/□□□□

```

{
  "parameter_path": "/BSC-1/BTS-23/Cell-1/CellCommonData/AdminState",
  "value": "UNLOCKED",
  "value_type": "enum"
}

```

□□□□□ LOCKED, UNLOCKED, SHUTTING_DOWN

□□□□□□□

```

{
  "parameter_path": "/BSC-1/BTS-23/Cell-1/CellAdvanced/CellTXPower",
  "value": "35",
  "value_type": "integer"
}

```

□□□0-46 dBm□□□□□□□

000000

```
{  
  "parameter_path": "/BSC-1/BTS-23/Cell-  
1/CellAdvanced/HandoverHysteresis",  
  "value": "3",  
  "value_type": "integer"  
}
```

000dB0000000-8 dB

0000000000

```
{  
  "parameter_path": "/BSC-1/BTS-23/MaxUEsServed",  
  "value": "256",  
  "value_type": "integer"  
}
```

00000000

□□□□□

□□□ **1**□□□□□

□□□□□□

1. □□□□
POST /api/devices

2. □□□□
PUT
/api/devices/:id/register

3. □□□□
PUT
/api/devices/:id/sessions

4. □□□□
PUT
/api/devices/:id/config/upload

5. □□□□
PUT
/api/devices/:id/metrics/upload

□□□□□
□□□

□□□

1. 登録

```
curl -X POST http://localhost:4000/api/v1/devices \
-H "Content-Type: application/json" \
-d '{
  "id": "site_a_bs1",
  "name": "SITE_A_BS1",
  "vendor": "Nokia",
  "address": "192.168.1.100:8080",
  "credentials": {
    "username": "admin",
    "password": "password"
  }
}'
```

2. 登録

```
curl -X PUT
http://localhost:4000/api/v1/devices/site_a_bs1/register \
-H "Content-Type: application/json" \
-d '{
  "webhook_url": "http://manager.example.com:9076/webhook"
}'
```

3. 登録

```
curl -X PUT
http://localhost:4000/api/v1/devices/site_a_bs1/sessions \
-H "Content-Type: application/json" \
-d '{"session_id": "session_001"}'
```

4. 登録

```
curl -X PUT
http://localhost:4000/api/v1/devices/site_a_bs1/config/upload \
-H "Content-Type: application/json" \
-d '{
  "filter": {
    "uploadType": "configuration",
    "objects": [{"sdn": "*", "depth": 100}]
  }
}'
```

000 2000000

```
# 1. 00000
curl -X PUT
http://localhost:4000/api/v1/devices/site_a_bs1/config/upload \
-H "Content-Type: application/json" \
-d '{
  "filter": {
    "uploadType": "configuration",
    "objects": [{"sdn": "/BSC-1/BTS-23/Cell-1", "depth": 10}]
  }
}' | jq '.configuration.parameters["/BSC-1/BTS-23/Cell-1"]'
```

```
# 2. 0000
curl -X PUT
http://localhost:4000/api/v1/devices/site_a_bs1/config/set \
-H "Content-Type: application/json" \
-d '{
  "parameter_path": "/BSC-1/BTS-23/Cell-
1/CellAdvanced/CellTXPower",
  "value": "38",
  "value_type": "integer"
}'
```

```
# 3. 0000
curl -X PUT
http://localhost:4000/api/v1/devices/site_a_bs1/config/upload \
-H "Content-Type: application/json" \
-d '{
  "filter": {
    "uploadType": "configuration",
    "objects": [{"sdn": "/BSC-1/BTS-23/Cell-1/CellAdvanced",
"depth": 5}]
  }
}' | jq '.configuration.parameters["/BSC-1/BTS-23/Cell-
1/CellAdvanced/CellTXPower"]'
```

000 3000000

```
# 00000000000000
#!/bin/bash

DEVICE="site_a_bs1"
INTERVAL=300 # 5 00

while true; do
    # 0000
    METRICS=$(curl -s -X PUT
http://localhost:4000/api/v1/devices/$DEVICE/metrics/upload \
-H "Content-Type: application/json" \
-d '{
    "filter": {
        "uploadType": "measurement",
        "objects": [{"sdn": "*", "depth": 100}]
    }
}')

    # 000000
    DL=$(echo $METRICS | jq '.metrics.counters[] |
select(.id=="M1C1") | .value')
    CELLS=$(echo $METRICS | jq '.metrics.counters | length')

    echo "$(date): DL=$DL Mbps, Cells=$CELLS"

    # 0000
    ALARMS=$(curl -s -X PUT
http://localhost:4000/api/v1/devices/$DEVICE/alarms/upload \
-H "Content-Type: application/json" \
-d '{
    "filter": {
        "uploadType": "active_faults"
    }
}' | jq '.alarms | length')

    if [ "$ALARMS" -gt 0 ]; then
        echo "0000$ALARMS 000000"
    fi
done
```

```
sleep $INTERVAL
done
```

□□□□

HTTP □□□

□□	□□	□□
200	□□	□□□□□
201	□□□	□□□□□
400	□□□□	□□□ JSON □□□
401	□□□	□□/□□□ API □□
404	□□□	□□□□□
409	□□	□□□□□
500	□□□□□	□□□□□□□
503	□□□	□□□□

□□□□□□

```
{
  "error": {
    "code": "DEVICE_NOT_FOUND",
    "message": "□□ 'site_a_bs1' □□□",
    "details": {
      "device_id": "site_a_bs1",
      "timestamp": "2025-12-10T14:30:00Z"
    }
  }
}
```

□□□□

□□□□□□

```
{
  "error": {
    "code": "NOT_REGISTERED",
    "message": "□□□□□□□□□□",
    "solution": "□□□□ PUT /api/devices/:id/register"
  }
}
```

□□□□□

```
{
  "error": {
    "code": "SESSION_EXPIRED",
    "message": "□□□□□□",
    "solution": "□□ PUT /api/devices/:id/sessions □□□□"
  }
}
```

□□□□□□□□

```
{
  "error": {
    "code": "INVALID_PARAMETER",
    "message": "□□□□□□□□",
    "details": {
      "parameter": "/BSC-1/BTS-23/Cell-
1/CellAdvanced/CellTXPower",
      "value": "99",
      "valid_range": "0-46 dBm"
    }
  }
}
```

API 接口

Python 实现

```
import requests
import json

class RanMonitorClient:
    def __init__(self, base_url="http://localhost:4000/api/v1"):
        self.base_url = base_url
        self.session = requests.Session()

    def register_device(self, device_id, address, username, password):
        """注册设备"""
        url = f"{self.base_url}/devices/{device_id}/register"
        payload = {
            "address": address,
            "web_username": username,
            "web_password": password,
            "webhook_url": "http://manager:9076/webhook"
        }
        response = self.session.put(url, json=payload)
        return response.json()

    def get_config(self, device_id, sdn="*", depth=100):
        """获取配置"""
        url = f"{self.base_url}/devices/{device_id}/config/upload"
        payload = {
            "filter": {
                "uploadType": "configuration",
                "objects": [{"sdn": sdn, "depth": depth}]
            }
        }
        response = self.session.put(url, json=payload)
        return response.json()

    def set_config(self, device_id, parameter_path, value, value_type="string"):
        """设置配置"""
        url = f"{self.base_url}/devices/{device_id}/config/set"
        payload = {
```

```

        "parameter_path": parameter_path,
        "value": value,
        "value_type": value_type
    }
    response = self.session.put(url, json=payload)
    return response.json()

def get_metrics(self, device_id):
    """获取设备指标"""
    url = f"
{self.base_url}/devices/{device_id}/metrics/upload"
    payload = {
        "filter": {
            "uploadType": "measurement",
            "objects": [{"sdn": "*", "depth": 100}]
        }
    }
    response = self.session.put(url, json=payload)
    return response.json()

# 创建客户端
client = RanMonitorClient()

# 注册设备
result = client.register_device(
    device_id="site_a_bs1",
    address="192.168.1.100:8080",
    username="admin",
    password="password"
)
print(f"设备注册结果: {result}")

# 获取配置
config = client.get_config("site_a_bs1")
print(f"设备配置: {json.dumps(config, indent=2)}")

# 设置配置
update = client.set_config(
    "site_a_bs1",
    "/BSC-1/BTS-23/Cell-1/CellAdvanced/CellTXPower",
    "38",
    "integer"
)

```

```
)  
print(f"□□□{update}")
```

cURL □□

□□□□□

```
curl -X PUT  
http://localhost:4000/api/v1/devices/site_a_bs1/register \  
-H "Content-Type: application/json" \  
-d '{  
  "address": "192.168.1.100:8080",  
  "web_username": "admin",  
  "web_password": "password",  
  "webhook_url": "http://manager:9076/webhook"  
}'
```

□□□□□□□

```
curl -X GET http://localhost:4000/api/v1/devices/site_a_bs1
```

□□□□□

```
curl -X PUT  
http://localhost:4000/api/v1/devices/site_a_bs1/config/upload \  
-H "Content-Type: application/json" \  
-d '{  
  "filter": {  
    "uploadType": "configuration",  
    "objects": [{"sdn": "/BSC-1/*", "depth": 50}]  
  }  
' | jq '.'
```

環境構築

AirScale 環境構築

概要

環境構築は AirScale を InfluxDB と XML を利用して行う。

環境

- OS - Ubuntu 16.04 LTS
 - 言語 - Python 2.7.12
 - ライブラリ - requests, influxdb, xmltodict
 - ストレージ - 100 MB ~ 690 MB
 - Web 画面 - 環境構築
 - API - 環境構築
 - InfluxDB 画面 - 環境構築
 - API - 環境構築
-

環境構築

環境構築

環境構築 AirScale 環境構築

- API 1 環境構築
- 環境構築 環境構築
- 環境構築 環境構築
- API 環境構築

概要

概要 RAN Monitor 構成

```
priv/airscale_configs/
├─ <airscale-name>/
│   ├─ current.xml # 現在
│   └─ ONS-Lab-Airscale_config_20251230_143522.xml # 2025.12.30 14:35
30 秒間隔 2:35
│   ├─ ONS-Lab-Airscale_config_20251229_120000.xml # 2025.12.29 12:00
│   └─ ONS-Lab-Airscale_config_20251228_093045.xml # 2025.12.28 09:30
└─ ... # ...
100 MB 制限
```

ファイル名 <AirScaleName>_config_YYYYMMDD_HHMMSS.xml

概要 AirScale 構成

構成

- 現在ファイル `current.xml`
- 構成ファイル
- 構成ファイル 100 MB 制限
- 構成ファイル Web UI から `current.xml` に更新
- 構成ファイル
- 構成ファイル

構成

構成

構成ファイル 構成

- 構成ファイル 100 MB 制限
- 構成ファイル

- 000000 000000000000
- 000000 0000 `current.xml` 0000000
- 0000 00000 145KB 000 ~690 00000000000000

0000

00000000000

1. 00000 - 0000 `<AirScale>_config_YYYYMMDD_HHMMSS.xml`
2. 0000 - `current.xml` 0000000
3. 0000 - 000000000000000
4. 00000 - 00000 > 100 MB00000000000000
5. 0000 - 0000000000

--	--	--	--

□□□□□□ 95 MB □655 □□□□□□ 145KB □

```
|— ONS-Lab-Airscale_config_20240101_100000.xml  <- 145KB
```

— ONS-Lab-Airscale_config_20240102_100000.xml (145KB)

└─ ... (653 000)

```
└─ ONS-Lab-Airscale_config_20251230_100000.xml  <- 145KB
```

2025-12-31 10:00:00 145KB

111

1. [0000NS-Lab-Airscale_config_20251231_100000.xml](#) (145KB)

2. $95 \text{ MB} + 145 \text{ KB} = 95.14 \text{ MB}$ 100 MB

3.

4. 656 95.14 MB

— — —

□□□□□□□□□□□□□□□□ 500KB□

111

1. [ONS-Lab-Airscale_config_20251231_150000.xml](#) (500KB)

2. $95.14 \text{ MB} + 500\text{KB} = 95.64 \text{ MB}$

3.

4. 657 95.64 MB

— — —

□ □ □ □ □ □ □ □ □ □ □

99.8 MB 685

□□□□200KB

1.

2. 100 MB

3. < 100 MB

4. “ 3 435 KB”

5. 682 99.6 MB

目標

確認事項

- 監視対象の容量 100 MB
- 監視対象のパス
- 監視対象のファイル
- 監視対象のディレクトリ
- 監視対象のファイル名

手順

監視対象の設定

```
# 監視対象の設定
du -sh priv/airscale_configs/
# 215M 3 70 MB

# 監視対象のパス
du -sh priv/airscale_configs/*/
# 95M      priv/airscale_configs/ons-lab-airscale/
# 68M      priv/airscale_configs/sector-1/
# 52M      priv/airscale_configs/sector-2/

# 監視対象のファイル名
find priv/airscale_configs -maxdepth 1 -type d -exec du -sm {} \;
| awk '$1 > 90 {print $2 " is at " $1 "MB (approaching 100MB limit)}'
```

確認結果

確認結果

Web UI 確認 Nokia → 確認

URL `https://<ran-monitor-ip>:9443/nokia/config-archive`

////////////////////////////////////

□□□□

□□□□□□□□□□□□

1. □□□□

- □□□□ - □□□□□□□□ AirScale □□
- □□□□ - □□□□□□□□□□□□
- □□ - □□□□□□□□□□
- □□□□□ - □□□□□□□□□□

2. □□□□

□□□□□□□□□□□□

項目	説明
時刻	UTC時刻
設定ファイル	config_20251230_143522.xml
サイズ	KB または MB
バージョン	“2.x.x” または “3.x.x”
備考	

時刻は UTC 形式で表示されます。

設定ファイルは `current.xml` 形式で表示されます。

3. 確認

確認事項

- 時刻 - 正確に表示されているか
- 設定ファイル - 正しいファイル名で表示されているか
- サイズ - 正しい単位で表示されているか

確認完了

確認結果

時刻は UTC 形式で表示されています。

設定

- 時刻を確認
- 設定を確認
- サイズを確認

4. 如何 如何 如何

5. 如何如何如何 `<AirScaleName>_config_YYYYMMDD_HHMMSS.xml` 如何如何如何如何

如何

- 如何如何
- 如何如何如何如何
- 如何如何如何如何如何
- 如何如何如何如何如何

如何如何

如何 如何如何如何如何如何如何如何

如何

1. 如何如何如何如何
2. 如何如何
3. 如何如何如何如何
4. 如何 如何 如何
5. 如何如何如何如何如何
6. 如何如何如何如何

如何如何

- 如何 `current.xml` 如何如何如何如何
- 如何如何如何如何 `???`
- 如何如何如何如何如何如何如何如何

如何如何

如何 如何如何如何如何如何如何如何

如何如何

1. 如何如何如何如何如何
2. 如何 XML 如何如何如何 `xmldiff` `Beyond Compare` `WinMerge`

3. 環境構築

環境構築

```
# 環境構築
wget https://<server>:9443/download/config/ONS-Lab-Airscale/ONS-Lab-Airscale_config_20251230_143522.xml
wget https://<server>:9443/download/config/ONS-Lab-Airscale/ONS-Lab-Airscale_config_20251229_120000.xml

# diff 実行
diff ONS-Lab-Airscale_config_20251229_120000.xml ONS-Lab-Airscale_config_20251230_143522.xml

# xml diff 実行
xmldiff ONS-Lab-Airscale_config_20251229_120000.xml ONS-Lab-Airscale_config_20251230_143522.xml
```

環境構築

環境構築

環境構築

環境構築

1. 環境構築
2. 環境構築
3. 環境構築
4. 環境構築
5. 環境構築
6. 環境構築

環境構築

ファイル

- ONS-Lab-Airscale_config_20251230_143522.xml (143KB) - ファイル
- ONS-Lab-Airscale_config_20251228_091045.xml (142KB) - 2 ファイル
- ONS-Lab-Airscale_config_20251225_180000.xml (142KB) - 5 ファイル

ファイル

- ファイル 142KB ファイル 143KB 12 30
- ファイル 12 28 ファイル 12 30 ファイル
- ファイル

ファイル

ファイル ファイル

ファイル

1. ファイル
2. ファイル
3. ファイル Web UI
4. ファイル → ファイル ID
5. ファイル → ファイル
6. ファイル
7. ファイル
8. ファイル

ファイル

- ✓ ファイル
- ✓ ファイル
- ✓ ファイル
- ✓ ファイル
- ✓ ファイル

ファイル

ファイル ファイル

項目

1. 項目名
 2. 項目番号
 3. 項目説明
 4. 項目単位
 5. 項目種類
 6. 項目属性
-

項目

項目名

項目説明

項目

1. 項目/項目
2. XML 項目
3. 項目
4. 項目

項目

- 項目
- 項目
- 項目
- 項目

項目

<!-- □□□□□□ -->

<!-- 10000000 -->

<parameter>

<name>cellId</name>

```
<value>1</value>
```

</parameter>

<!-- 200000 -->

```
<parameter><name>cellId</name><value>1</value></parameter>
```

□□□□□□ ~145 KB □□□□□□□□ AirScale □□□

□□□□□

項目	項目名	項目名 145KB	項目名
10	100 MB	~690	1 GB
50	100 MB	~690	5 GB
100	100 MB	~690	10 GB
500	100 MB	~690	50 GB
1000	100 MB	~690	100 GB

5/5

- 100 MB
- ~690 145KB
- current.xml 145KB
- 100 MB
-

配置

- 配置存储目录
- 配置存储大小
- 配置存储格式
- 配置存储策略

编译

编译选项

- 配置存储 **100 MB** 大小
- 配置存储格式
- 配置 `current.xml` 文件
- 配置存储策略
- 配置存储目录

编译命令

配置 ConfigStorage 选项

```
# 在 lib/ran_monitor/nokia/airscale/config_storage.ex 中
# 配置存储大小
@max_storage_bytes 100 * 1024 * 1024 # 100 MB

# 配置存储格式
@max_storage_bytes 50 * 1024 * 1024 # 50 MB 约 345 个文件
@max_storage_bytes 200 * 1024 * 1024 # 200 MB 约 1380 个文件

# 配置存储策略
def cleanup_old_versions(airscale_name, max_size_bytes \
@max_storage_bytes)
```

编译命令

```
mix compile
# 编译 RAN Monitor 选项
```

配置

每隔 1 秒 3,600,000 毫秒

配置项名称

lib/ran_monitor/nokia/airscale/manager.ex

```
defp schedule_get_airscale_config do
  # 1 秒 3,600,000 ms
  Process.send_after(self(), :get_airscale_config, 3_600_000)
end
```

配置项

- 30 秒 1_800_000
- 1 秒 3_600_000 毫秒
- 2 秒 7_200_000
- 4 秒 14_400_000
- 24 秒 86_400_000

配置项名称 RAN Monitor

配置

配置项名称

配置

- 配置项名称“0 秒”
- 配置项名称

配置项

1. 配置项

- 000000
- 000000“000”00
- 00000000000000

2. 000000

- 0000000000
- 00000000“00”
- 00000000

3. 0000000

- 00000000000000
- 00000000

```
Kernel.send(Process.whereis(RanMonitor.Nokia.Airscale.Manager),
: get_airscale_config)
```

4. 0000000

- 00 `priv/airscale_configs/` 00000000
- 00 RAN Monitor 00000000
- 000000000000000000

0000 **404** 00

000

- 000000“00000000”
- 000000 404 00

000000

1. 00000000

- 00000000000000000000
- 00 `priv/airscale_configs/` 00000000

2. 000000

- 配置文件的权限
- 配置文件的名称

3. 配置

- 配置 Web 服务的配置文件
- 配置文件的名称

配置

```
# 配置文件的权限
ls -la priv/airscale_configs/

# 配置文件的名称
ls -la priv/airscale_configs/<device-name>/

# 配置文件的权限
chmod 755 priv/airscale_configs/
chmod 644 priv/airscale_configs/*/*.xml
```

配置

配置

- 配置 `current.xml` 文件
- 配置文件的名称 1

配置

1. 配置

- 配置文件的名称
- 配置文件的名称“配置文件的名称”

2. 配置

- 配置文件的名称
- 配置文件的名称
- 配置文件的名称

3. 環境構築

- 環境構築
- 環境構築

環境

```
# 環境構築
grep "process_configuration" <log-file>

# 環境構築
# IEx 環境構築
Kernel.send(Process.whereis(RanMonitor.Nokia.Airscale.Manager),
: get_airscale_config)
```

環境構築

環境構築

環境構築

環境構築

```
#!/bin/bash
# backup-configs.sh - 备份配置文件

BACKUP_DIR="/backup/ran-monitor/configs"
CONFIG_DIR="priv/airscale_configs"
DATE=$(date +%Y%m%d)

# 创建备份目录
mkdir -p "$BACKUP_DIR/$DATE"

# 同步文件
rsync -av "$CONFIG_DIR/" "$BACKUP_DIR/$DATE/"

# 清理 30 天前的备份
find "$BACKUP_DIR" -type d -mtime +30 -exec rm -rf {} +

echo "备份完成: $BACKUP_DIR/$DATE"
```

配置 **cron** 任务

```
0 2 * * * /path/to/backup-configs.sh
```

总结

备份策略与工具

备份流程

1. 确定备份范围和频率
2. 选择合适的备份工具
3. 配置备份任务和清理策略

常见问题

```
# config_changes.log

2025-12-30 14:35:22 - ONS-Lab-Airscale
#####John Smith
##### 1 #####
#####txPower 40dBm 43dBm
#####
#####2025-12-30 14:40:00
#####

2025-12-28 09:10:45 - ONS-Lab-Airscale
#####Jane Doe
#####
##### 101112
#####
#####2025-12-28 09:15:00
#####
```

#####

#####

#####

```
# #####
du -sh priv/airscale_configs/

# #####
du -sh priv/airscale_configs/*/

# #####
find priv/airscale_configs/ -name "*.xml" | \
  sed 's|/[^/]*\.xml||' | uniq -c
```

- #####
- #####500MB#####
 - #####
 - ##### 10% #####

□□□□□□□□

□□□□□□□□□□□□□□□□

□□□□□□

□□□□□□□

- □□□□□□□□ `current.xml`
- □□□□□□□□“□□”□□□□□□□□□□

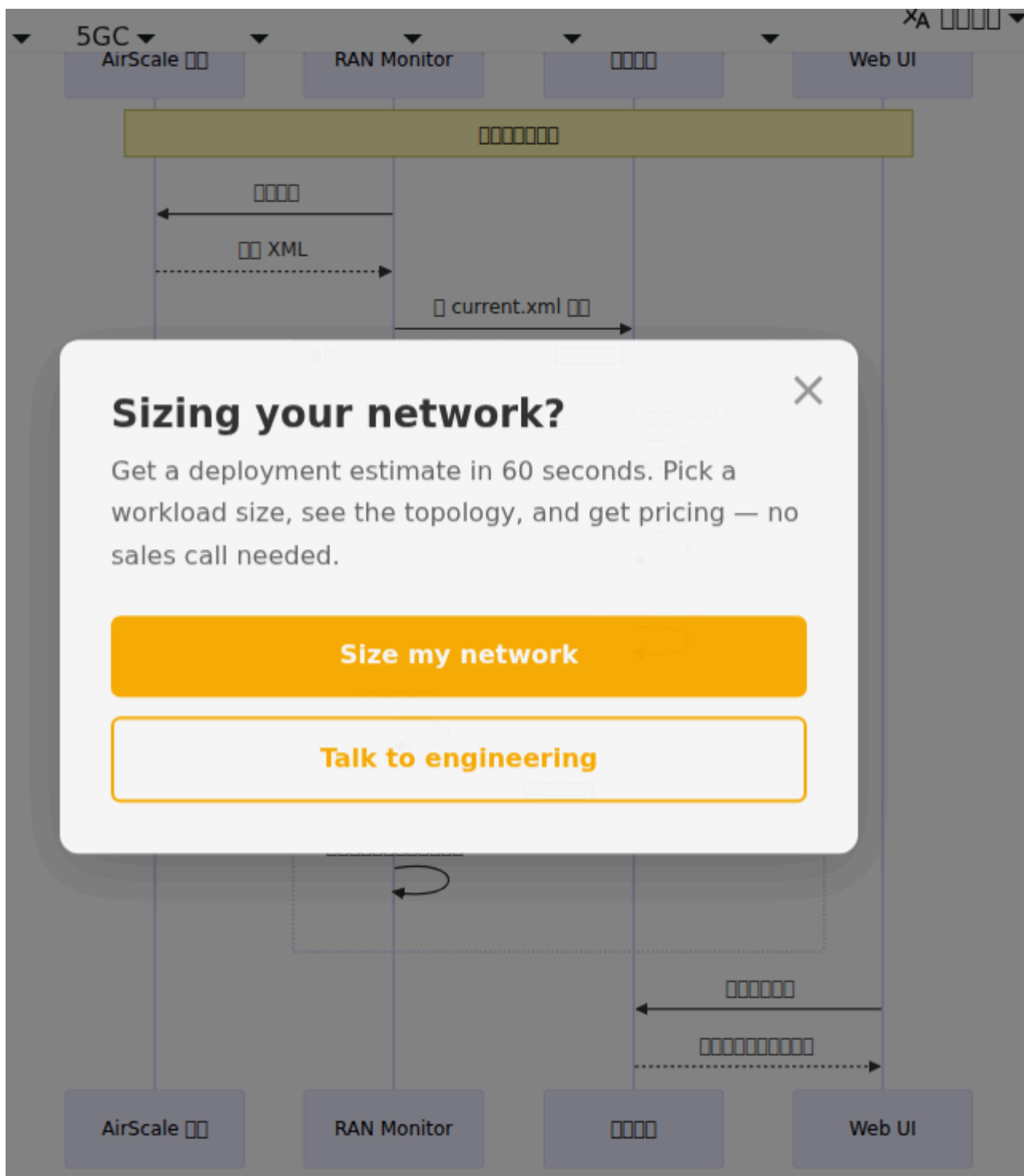
□□□□□□□□□

- □□□□□□□□□
- □□ → □□ → □□□□□□□
- □□□□□□□□□□□□□□□□

□□□□□□

- □□□□□□□□□□
- □□□□□□□□□□
- □□□□ → □□□□□□□□□□

□□□



API □□

□□□□□□□□□□ Web UI □□□□□□□□□□ HTTP □□□□□□□□

□□□□□

```
curl -k "https://<server>:9443/download/config/<airscale-  
name>/current.xml" \  
-o current_config.xml
```

□□□□□

```
curl -k "https://<server>:9443/download/config/<airscale-  
name>/ONS-Lab-Airscale_config_20251230_143522.xml" \  
-o ONS-Lab-Airscale_config_20251230_143522.xml
```

URL AirScale

□ □ □ □ □

□□□□□ **IEx** □□□□□

```
# 列出所有版本
RanMonitor.Nokia.Airscale.ConfigStorage.list_config_versions("ONS-
Lab-Airscale")

# 获取当前配置
{:ok, xml} =
RanMonitor.Nokia.Airscale.ConfigStorage.get_current_config("ONS-
Lab-Airscale")

# 版本数量
RanMonitor.Nokia.Airscale.ConfigStorage.count_versions("ONS-Lab-
Airscale")
# 655

# 最大存储空间
RanMonitor.Nokia.Airscale.ConfigStorage.max_storage_bytes()
# 104857600 100 MB

# 存储空间使用量
RanMonitor.Nokia.Airscale.ConfigStorage.get_storage_usage("ONS-
Lab-Airscale")
# 99614055

# 存储空间统计
RanMonitor.Nokia.Airscale.ConfigStorage.get_storage_stats("ONS-
Lab-Airscale")
# %{:version_count: 655, total_size_bytes: 99614055, ...}

# 清理 50 MB 旧版本
RanMonitor.Nokia.Airscale.ConfigStorage.cleanup_old_versions("ONS-
Lab-Airscale", 50 * 1024 * 1024)
# {:ok, 345, 50000000} - 345 个版本 50MB

# 清理 100 MB 旧版本
RanMonitor.Nokia.Airscale.ConfigStorage.cleanup_old_versions("ONS-
Lab-Airscale")
# {:ok, 0, 0} - 没有旧版本可清理
```



- **Web UI** □□ - □□□□□□□□
- **AirScale** □□□□ - □□□□
- □□□□□□ - □□□□□□
- □□□□□□□□ - □□□□

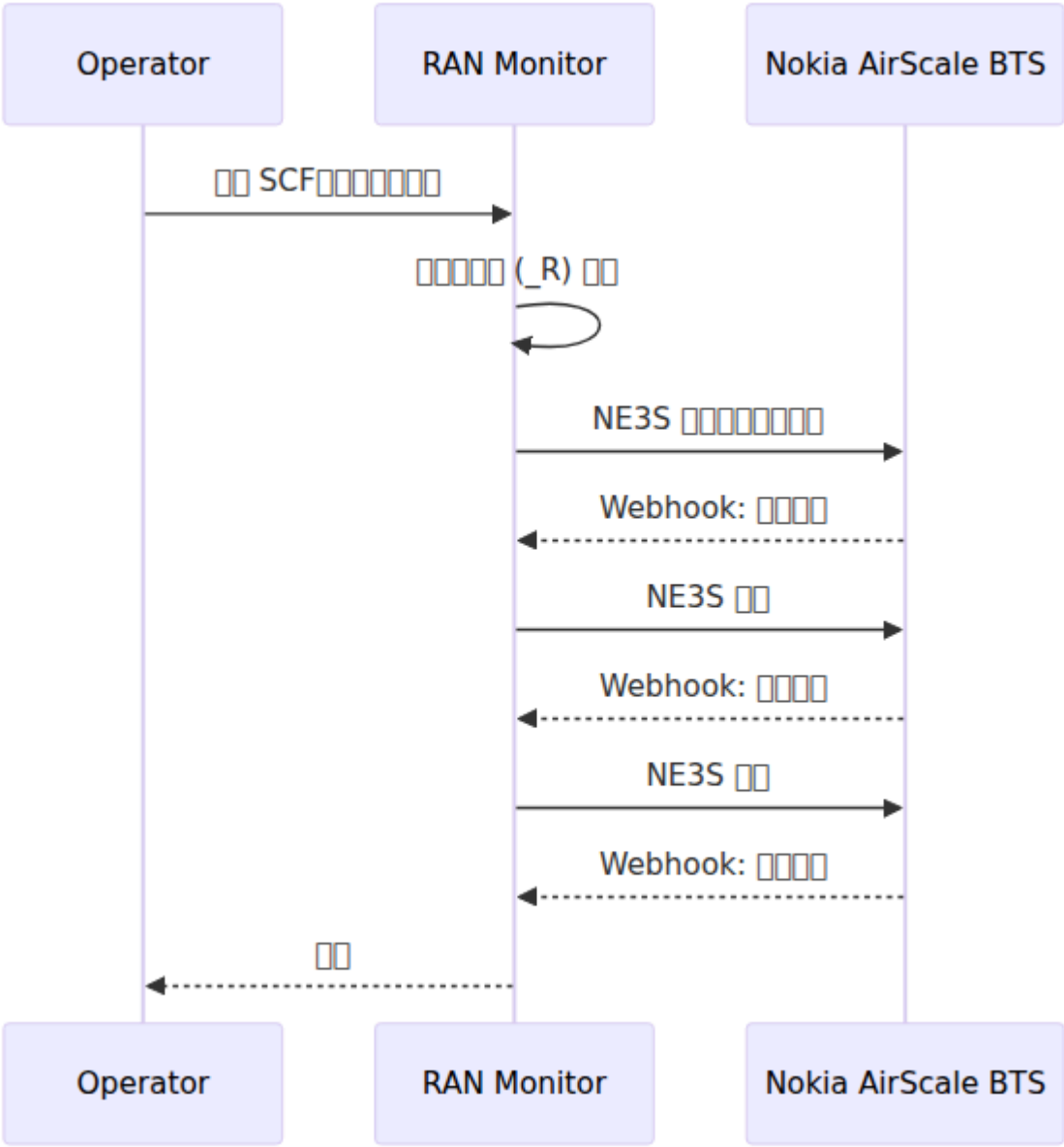


☐☐ RAN Monitor ☐ Nokia AirScale ☐☐☐☐☐☐☐



RAN Monitor 0000 NE3S 00000000 Nokia AirScale 000000000000 RAML 2.1 XML 0
00 SCF0000000000000000 BTS 0000000000000000000000000000000000 RAN
Monitor 0000000000

BTS **SCF**



SCF XML

RAML 2.1 XML

```
<?xml version="1.0" encoding="UTF-8"?>
<raml xmlns="raml21.xsd" version="2.1">
  <cmData type="actual" scope="all" domain="ALL">
    <header>
      <log dateTime="2026-03-21T00:00:00+00:00" action="created"></log>
    </header>

    <!--   -->

  </cmData>
</raml>
```

<managedObject>

```
<managedObject class="NOKLTE:LNBTs" distName="MRBTs-256/LNBTs-256"
operation="update">
  <p name="enbName">MY-SITE-NAME</p>
</managedObject>
```

class	Nokia NOKLTE:LNCEL com.nokia.srbts.tnl:IPIF
distName	—
operation	create SCF update

distName

MRBTS-{bts_id}

MRBTS-{bts_id}/LNBTS-{bts_id}

MRBTS-{bts_id}/LNBTS-{bts_id}/LNCEL-{cell_id}

MRBTS-{bts_id}/GNBTS-{gsm_id}

MRBTS-{bts_id}/GNBTS-{gsm_id}/GNCEL-{cell_id}

MRBTS-{bts_id}/MNL-1/MNLENT-1/PMCADM-1

MRBTS-{bts_id}/EQM-1/APEQM-1

BTS

LTE eNodeB

LTE

GSM BTS

GSM

PM

#

SCF

SCF

SCF BTS 100-200 `operation="create"`

SCF /IP BTS 3-5

SCF `operation="update"` `<p>`

PM

□□□□

	□□ SCF	□□□□
□□□□	operation="create"	operation="update"
□□□□□	□□□□□□□□	□□□□□□□
□□□□□	□□□□□□□□□□	□□□□□□□
□□□□	□□□□	□□□□□□□
□□□□	□□□□	□□□□□

□□□□□□□□□□

□□ **BTS** □□□□□

□□□□□□□□□□□□□□□□ BTS □□□□□□□□□□ □□ **3-5** □□□

□□	□□□
□□ / IP	TNL □ ETHIF □ ETHLK □ IPIF □ IPADDRESSV4 □ IPRT □ IPRTV6 □ VLANIF □ L2SWI □ BRGPRT □ IBRGPR □ IPNO □ ETHSVC □ TNLSVC □ IPSECC □ FIREWALL
□□ / □□	MRBTS □ SMOD □ BBMOD
□□□ □	MPLANENW □ CLOCK □ SYNC □ NTP
□□ / □□	SECADM □ CERTH □ CERTHENT □ CMP □ CMPECDSA □ CMPFH

□□□□□□

□□ **eNB** □□

□□□□□□□□

```
<?xml version="1.0" encoding="UTF-8"?>
<raml xmlns="raml21.xsd" version="2.1">
<cmData type="actual" scope="all" domain="ALL">
<header>
<log dateTime="2026-03-21T00:00:00+00:00" action="created"></log>
</header>
<managedObject class="NOKLTE:LNBTs" distName="MRBTS-256/LNBTs-256"
operation="update">
<p name="enbName">NICK-LAB-AIRSCALE</p>
</managedObject>
</cmData>
</raml>
```

□□□□□□

□□□□□□□□

```
<?xml version="1.0" encoding="UTF-8"?>
<raml xmlns="raml21.xsd" version="2.1">
<cmData type="actual" scope="all" domain="ALL">
<header>
<log dateTime="2026-03-21T00:00:00+00:00" action="created"></log>
</header>
<managedObject class="NOKLTE:LNCEL" distName="MRBTS-256/LNBTs-
256/LNCEL-256" operation="update">
<p name="cellName">TRI-B5-1</p>
</managedObject>
</cmData>
</raml>
```

物理小区 ID

物理小区 ID 设置

```
<?xml version="1.0" encoding="UTF-8"?>
<raml xmlns="raml21.xsd" version="2.1">
  <cmData type="actual" scope="all" domain="ALL">
    <header>
      <log dateTime="2026-03-21T00:00:00+00:00" action="created"></log>
    </header>
    <managedObject class="NOKLTE:LNCEL_FDD" distName="MRBTS-256/LNBTS-256/LNCEL-256/LNCEL_FDD-256" operation="update">
      <p name="physCellId">120</p>
    </managedObject>
  </cmData>
</raml>
```

EARFCN 设置

EARFCN 设置

```
<?xml version="1.0" encoding="UTF-8"?>
<raml xmlns="raml21.xsd" version="2.1">
  <cmData type="actual" scope="all" domain="ALL">
    <header>
      <log dateTime="2026-03-21T00:00:00+00:00" action="created"></log>
    </header>
    <managedObject class="NOKLTE:LNCEL_FDD" distName="MRBTS-256/LNBTS-256/LNCEL-256/LNCEL_FDD-256" operation="update">
      <p name="earfcnDL">1400</p>
      <p name="dlChBw">10 MHz</p>
    </managedObject>
  </cmData>
</raml>
```

PM 设置

PM 设置

```
<?xml version="1.0" encoding="UTF-8"?>
<raml xmlns="raml21.xsd" version="2.1">
<cmData type="actual" scope="all" domain="ALL">
<header>
<log dateTime="2026-03-21T00:00:00+00:00" action="created"></log>
</header>
<managedObject class="com.nokia.srbts.mnl:PMCADM" distName="MRBTS-
256/MNL-1/MNLENT-1/PMCADM-1" operation="update">
<p name="rTPmCollInterval">60s</p>
</managedObject>
</cmData>
</raml>
```

SCF

SCF eNB

```
<?xml version="1.0" encoding="UTF-8"?>
<raml xmlns="raml21.xsd" version="2.1">
<cmData type="actual" scope="all" domain="ALL">
<header>
<log dateTime="2026-03-21T00:00:00+00:00" action="created"></log>
</header>
<managedObject class="NOKLTE:LNBTs" distName="MRBTS-256/LNBTs-256"
operation="update">
<p name="enbName">NICK-LAB-AIRSCALE</p>
</managedObject>
<managedObject class="NOKLTE:LNCEL" distName="MRBTS-256/LNBTs-
256/LNCEL-256" operation="update">
<p name="cellName">TRI-B5-1</p>
<p name="pMax">331</p>
</managedObject>
</cmData>
</raml>
```

IP

IP BTS 3-5

```
<?xml version="1.0" encoding="UTF-8"?>
<raml xmlns="raml21.xsd" version="2.1">
  <cmData type="actual" scope="all" domain="ALL">
    <header>
      <log dateTime="2026-03-21T00:00:00+00:00" action="created"></log>
    </header>
    <managedObject class="com.nokia.srbts.tnl:IPADDRESSV4"
      distName="MRBTS-256/TNL-1/IPNO-1/IPIF-1/IPADDRESSV4-1"
      operation="update">
      <p name="localIpAddr">10.7.15.66</p>
    </managedObject>
  </cmData>
</raml>
```

Web UI

1. BTS
- 2.
3. SCF XML
- 4.
5. RAN Monitor
- 6.

API

```
PUT /api/ne3s/function/provision
Content-Type: application/json
```

```
{
  "airscale_name": "ons-Lab-Airscale",
  "config_file": "/path/to/partial-config.xml"
}
```

`config_file` RAN Monitor SCF XML

```
{
  "status": "success",
  "operation": "provision",
  "plan_id": "plan_1774041772721652742",
  "data": "<activateResponse>...</activateResponse>"
}
```

```
{
  "status": "error",
  "code": "PROVISION_FAILED",
  "message": "validate failed: Validation error details..."
}
```

API

<code>airscale_name</code>			-	RAN Monitor
<code>config_file</code>			-	RAN Monitor SCF XML

- Web UI "API" API
-
- `operation="update"` `operation="create"` SCF

4. SCF BTS
5. `distName` **BTS** MRBTS ID LNBTS ID ID `distName` BTS
6. EARFCN PCI

Duplicated object MRBTS-256/EQM_R-1

SCF _R BTS

RAN Monitor _R RAN Monitor BTS

Operation could not execute because of operation conflict, RequestOperationType = {activate}, OngoingOperationType = {validate}

BTS

RAN Monitor NetAct WebLM BTS

BTS

BTS ping 3-5

/IP

BTS 3-5

安裝說明

簡介

RAN Monitor 是一個用於監控 4G/LTE 網路效能的工具。它透過 InfluxDB 儲存數據，並提供多種圖表來展示網路效能。

系統需求

硬體需求

- IP 位址: `https://localhost:9443`
- 網路連線: 網路
- 支援 eNodeB 設備

軟體需求

- 支援 eNodeB
- 支援“網路”設備
- 支援設備
- 支援設備

安裝步驟

- 安裝 InfluxDB 資料庫
- 安裝 eNodeB 設備

□□

□□□□□□□□□□ *eNodeB* □□□□□□□□□□

□ □□

□□□□□□

- □□□□□□: 720 □□□30 □□
- □□□: □ `config/config.exs` □□□
- □□: □□□□□□□□□□□□□□ *eNodeB*

□□ **eNodeB** □□□

- □□□□: □□□ *eNodeB* □□□□□□□□
- □□□□□□: □□□□ `airscales` □□
- □□: □□□□

📌📌📌

- 📌: 📌📌📌📌📌
- 📌📌📌: `RanMonitor.Data.RetentionCleanupWorker`
- 📌 **eNodeB**: 📌📌📌📌📌
- 📌📌: 📌📌📌📌📌📌📌📌📌📌

📌📌📌📌

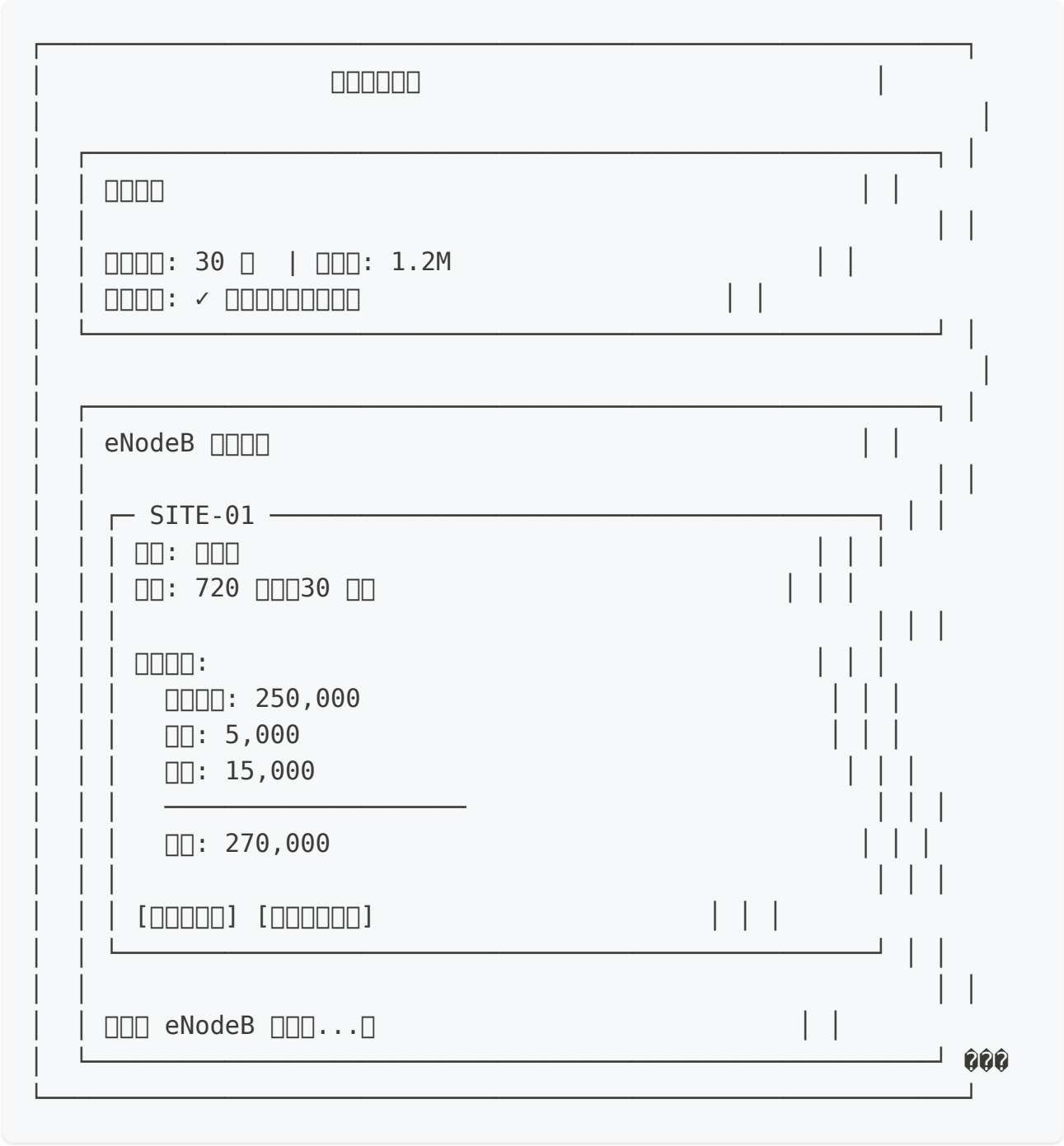
- 📌📌📌: 📌📌📌📌📌📌📌📌
 - 📌📌📌
 - 📌📌
 - 📌📌
 - 📌📌: 📌📌📌 eNodeB 📌📌📌
 - 📌📌: 📌📌📌📌📌📌
-

□ □□□□

□□□□□

□□□□□□□□□□□□□□□□ *eNodeB* □□□□□□□□□□

□□□□□



無線LAN

- 無線LAN (✓): eNodeB 無線LAN
- 無線LAN (X): eNodeB 無線LAN
- 無線LAN: 無線LAN eNodeB

配置

項目	値	説明
監視対象	監視対象	監視対象のリスト
監視周期	監視周期	監視周期 eNodeB 監視周期 ⚠ 監視周期
単位	単位	監視周期の単位

設定

監視対象

config/config.exs:

```
config :ran_monitor,
  ecto_repos: [RanMonitor.Repo],
  generators: [context_app: :ran_monitor],
  data_retention_hours: 720 # 30 監視周期
```


配置参数

参数	单位	默认值	说明
1 分钟	1	0.04	每 1 分钟采样一次
1 天	24	1	每天采样一次
7 天	168	7	每周采样一次
14 天	336	14	每两周采样一次
30 天	720	30	每月采样一次
90 天	2160	90	每季度采样一次
180 天	4320	180	每半年采样一次
1 年	8760	365	每年采样一次

环境变量

环境变量配置

```
export DATA_RETENTION_HOURS=1440 # 60 天
mix phx.server
```

1. 数据库

数据库设计

1. 数据库
 - ├ 基站 → InfluxDB
 - ├ 基站 → InfluxDB
 - └ 基站 → InfluxDB
2. 数据库设计
 - ├ RetentionCleanupWorker 类
 - ├ 基站 eNodeB
 - | ├ 基站 eNodeB 基站
 - | ├ 基站
 - | └ 基站
 - └ 基站
3. 数据库设计
 - ├ 基站 UI 基站
 - ├ 基站
 - ├ 基站
 - └ 基站/基站
4. 基站
 - └ 基站 UI 基站

数据库

基站 eNodeB 数据库

```
effective_retention = case airscales.retention_hours do
  nil -> Config.data_retention_hours() # 基站720h
  hours -> hours # 基站 eNodeB 基站
end
```

基站

- 基站720 基站30 基站

- eNodeB "SITE-01" 168 7
- eNodeB "SITE-02" nil → 720

`expiration = now - (retention_hours * 3600)`

```
30
├ now: 2025-12-11 10:00:00
├ 720 30
├ now: 2025-11-11 10:00:00
└ expiration < now
```

```
[RetentionCleanupWorker]
[RetentionCleanupWorker] SITE-01 : 720h
[RetentionCleanupWorker] SITE-01 15,000
[RetentionCleanupWorker] 5 0 75,000
```

- 1.
2. eNodeB
3. “”

-

- 安装 InfluxDB 数据库
- 安装数据库驱动

1 2 3 4 5

6 7 8 9 10

11 12 13 14 15

```
alias RanMonitor.Data.RetentionPolicy
alias RanMonitor.Database.Nokia

# 1 eNodeB 2
airscale = Nokia.get_airscale!(1)
hours = RetentionPolicy.get_retention_hours(airscale)
# => 720 3

# 4 eNodeB 5
counts = RetentionPolicy.get_record_counts("SITE-01")
# => %{"PerformanceMetrics" => 250000, "Configuration" => 5000,
"Alarms" => 15000}

# 6
total = RetentionPolicy.get_total_record_count("SITE-01")
# => 270000

# 7
{:ok, deleted_count} = RetentionPolicy.delete_old_records("SITE-
01", 720)
# => {:ok, 50000} (8 50k 9)

# 10 eNodeB 11
{:ok, deleted_count} = RetentionPolicy.clear_all_records("SITE-
01")
# => {:ok, 270000} (12 270k 13 14 15)
```

编译选项

lib/ran_monitor/data/retention_cleanup_worker.ex:

```
# 1 3600000ms 30 1800000ms  
@cleanup_interval_ms 1800000 # 30
```

编译选项

```
mix compile
```

编译选项

编译选项

```
SELECT name, retention_hours FROM airscales;
```

编译选项

```
UPDATE airscales  
SET retention_hours = 168  
WHERE name = 'SITE-01';
```

编译选项

编译选项

编译选项 (< 7)

- 编译选项
- 编译选项
- 编译选项

7-30

- 1
- 2
- 3

> 30

- 1
- 2
- 3

	7-14	
	30-60	
	90	
	180-365	

- 1000 $\approx$ 1-5 KB
- 100 $\approx$ 1-5 GB
- $\times$ =

```
# 查看 InfluxDB 桶列表
influx bucket list

# 检查数据目录
df -h /path/to/influxdb/data
```

安装 InfluxDB

简介

- 实时监控和存储时间序列数据
- 支持多种数据源和查询语言
- 易于集成和扩展

安装

- 系统要求: 64位 Linux 或 macOS
- 依赖项: 需要安装 Go 语言环境
- 安装步骤: 见下文

配置

1. 设置 InfluxDB 用户和权限
2. 配置数据源和查询语言
3. 设置日志和监控
4. 配置网络和安全
5. 启动 InfluxDB 服务

1. 背景

1.1 需求背景

背景

- 网络性能优化
- 网络故障排查

背景

1. 网络性能优化: `ps aux | grep mix`
2. 网络故障排查: `RetentionCleanupWorker` 启动失败
 - 网络故障: `[RetentionCleanupWorker] Starting`
3. 网络故障排查: InfluxDB 启动失败
 - 网络故障: `InfluxDB 启动失败: https://localhost:9443/nokia/influx`
4. 网络故障排查: 配置文件中配置了 `data_retention_hours`

1.2 需求背景

背景

- 网络性能优化
- 网络故障排查

背景

1. 网络故障排查: InfluxDB 启动失败
 - 网络故障: `InfluxDB 启动失败`
2. 网络故障排查: 配置文件中配置了 `data_retention_hours`
3. 网络故障排查: 配置文件中配置了 `[RetentionPolicy]`
4. 网络故障排查: eNodeB 启动失败
 - 网络故障: `BTS 启动失败`

環境構築

前提

- Docker
- Kubernetes

手順

1. Docker
2. 5-10 分
3. `@cleanup_interval_ms` 1 分
4. eNodeB

環境構築

前提

- InfluxDB UI
- “InfluxDB”

手順

1. “InfluxDB”
2. InfluxDB
 - InfluxDB
3. InfluxDB
 - InfluxDB
4. eNodeB
 - InfluxDB

環境構築

- InfluxDB - InfluxDB

- **Web UI**  -  
 -  - 
 -  - 
 - **Grafana**  - 
-

- : <https://localhost:9443/nokia/retention>
 - **BTS** : <https://localhost:9443/nokia/status>
 - **InfluxDB** : <https://localhost:9443/nokia/influx>
 - : <https://localhost:9443/nokia/logs>
-

eNodeB

   eNodeB         

             [@cleanup_interval_ms](#) 

   eNodeB         

時間系列データベース

時間系列データベースとは、時間軸に沿ってデータを保存・管理・分析するためのデータベースです。

特徴

時間系列データベースの特徴は、

- 大量のデータを高速に読み書きできる
- 時間軸に沿ってデータを保存・管理・分析できる
- 複雑なクエリを高速に実行できる

用途

時間系列データベースは、IoT、金融、製造業、物流、エネルギー、インフラ、ヘルスケア、マーケティング、セキュリティ、監視、インフラ管理、インフラ監視、インフラ監視、InfluxDB CLI、API

時間系列データベース InfluxDB の特徴

時間系列データベース InfluxDB の特徴は、

インストール

時間系列データベース InfluxDB のインストール方法は、

1 数据库

数据库

文件	说明
lib/ran_monitor/database/nokia/airscale.ex	定义 retention_hours 函数
lib/ran_monitor/config/config.ex	定义 data_retention_hours() 函数
config/config.exs	配置数据库连接
lib/ran_monitor/application.ex	定义数据库连接

数据库

文件
lib/ran_monitor/data/retention_policy.ex
lib/ran_monitor/data/retention_cleanup_worker.ex
lib/ran_monitor/web/live/retention_policy_live.ex
priv/repo/migrations/20251211065257_add_retention_hours_to_airscales.

数据库

RetentionPolicy 函数

- get_retention_hours(airscale) - 获取保留时间
- get_record_counts(airscale_name) - 获取记录数量

- `get_total_record_count(airscale_name)` - 取得
- `delete_old_records(name, hours)` - 削除
- `clear_all_records(name)` - 削除

RetentionCleanupWorker GenServer

- `start_link(opts)` - 起動
 - `init(:ok)` - 初期化
 - `handle_info(:cleanup, state)` - 処理
-

1. 起動

2. 初期化

3. 処理

```
mix ecto.migrate
```

4. 起動

```
mix phx.server
```

5. 処理

- 起動 `https://localhost:9443/nokia/retention`
- 削除

□□□□

1. □□□□□□

- □□□□□□□□720 □□□
- □□□□ eNodeB □□□□□

2. □□□□□□□

- □ `config/config.exs` □□□□□□□
- □□□ UI □□□□ eNodeB

3. □□□□□

- □□□□□□ `[RetentionCleanupWorker]` □□
 - □□□□□□□□□□□
-

1. 背景

2. 目的

1. 背景として、[RetentionPolicy] と [RetentionCleanupWorker] の関係
2. 背景として、RetentionPolicy の役割
3. 背景として、RetentionCleanupWorker の役割
4. 背景として、RetentionPolicy と RetentionCleanupWorker の関係

3. 概要

4. 詳細

- UI 画面の表示
- 背景として eNodeB の役割
- 背景として RetentionPolicy の役割
- 背景として RetentionCleanupWorker の役割
- 背景として RetentionPolicy と RetentionCleanupWorker の関係

2. 背景

3. 目的

- InfluxDB v2.x: 背景として、InfluxDB v2.x の役割
- 背景として: InfluxDB v2.x の役割
- 背景として: InfluxDB v2.x の役割
- 背景として: InfluxDB v2.x の役割

4. 概要

- InfluxDB の役割
- Elixir GenServer の役割
- Phoenix LiveView の役割

Nokia AirScale DHCP



DHCP



- 1.
- 2.
- 3.
4. DHCP
- 5.
- 6.
- 7.
8. 43



Nokia AirScale DHCP RAN Monitor DHCP
DHCP IP DHCP 43 RAN Monitor

DHCP DHCP RAN Monitor IP
NE3S



Nokia AirScale Omnitouch

- DHCP
- DHCP

- 配置 DHCP 服务器

配置 **Omnitouch** 服务器 **Omnitouch** 服务器 **DHCP** 配置

配置

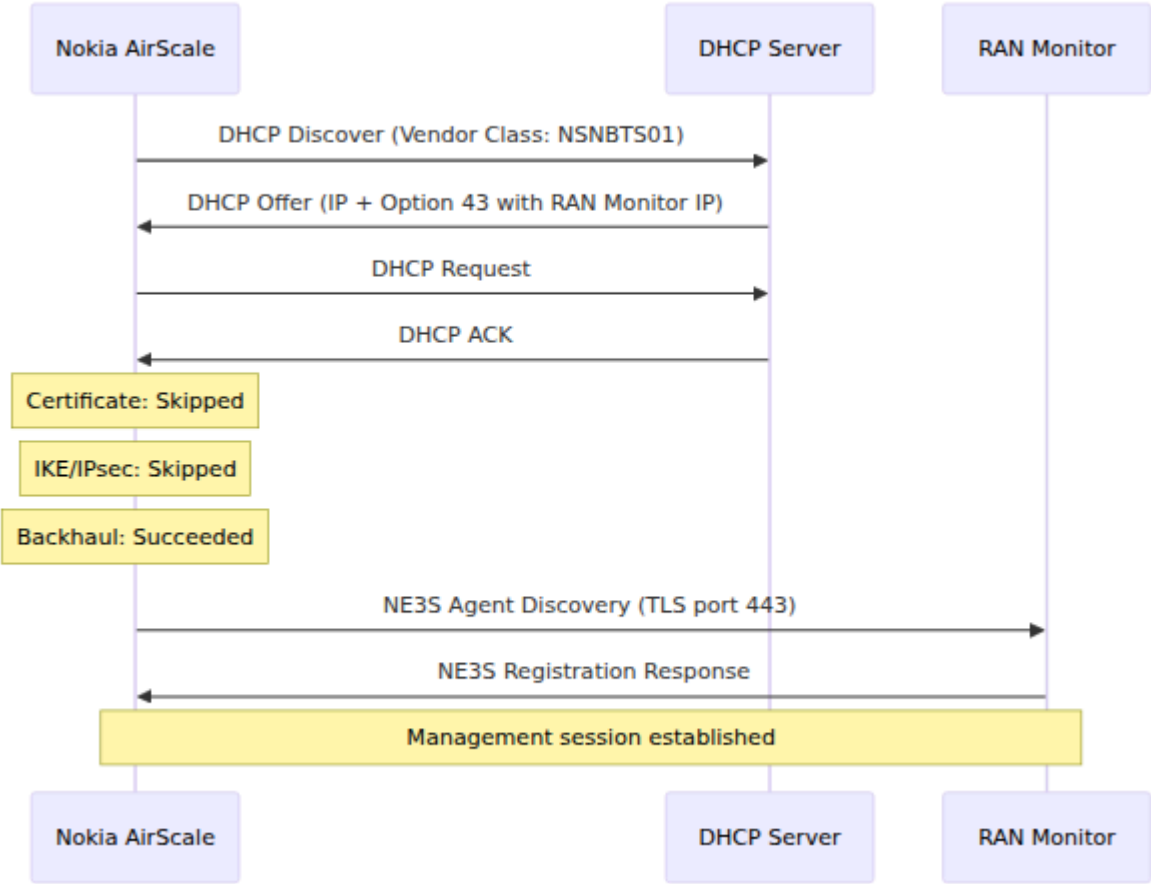
配置 DHCP 服务器

- **DHCP** 配置 — 配置 DHCP 服务器 MikroTik/ISC DHCP/dnsmasq 配置
- **RAN Monitor** — 配置 RAN Monitor 配置
- 配置 — 配置 OAM 配置 DHCP 服务器 2 配置

配置

配置	配置	配置
RAN Monitor IP	RAN Monitor 配置 IP 配置	10.179.2.139
配置	配置	10.7.15.64/26
配置 IP	配置	10.7.15.65
DNS 配置	DNS 配置 IP	10.7.15.65

□□□□



□□□□□□

□□□□□□ DHCP □□□□□□□□□□

□□	□□	□□
DHCP	□□	□□ IP □□□ RAN Monitor □□
□□	□□	□□□□□□□□□□□□□□
IKE	□□	□□□ IPsec □□□□□□□□
□□	□□	□□□□□□
□□	□□	□ RAN Monitor □□ NE3S □□□□

DHCP オプション

オプション 43

Nokia AirScale ネットワーク DHCP オプション 43 オプション RAN Monitor IP アドレスを DHCP オプション 43 の DHCP オプション

オプション 43 の

オプション RAN Monitor IP アドレスを DHCP オプション

```
#!/bin/bash
# generate-nokia-option43.sh
# Usage: ./generate-nokia-option43.sh <ran-monitor-ip>
IP="${1:?Usage: $0 <ran-monitor-ip>}"
IFS='.' read -r a b c d <<< "$IP"
H=$(printf "%02x%02x%02x%02x" $a $b $c $d)
S=$(echo -n "$IP" | xxd -p)
L=$(printf "%02x" ${#IP})
echo "0x0104${H}0104${H}08${L}${S}0904706b69780b0101"
```

例として 10.179.2.139

```
$ ./generate-nokia-option43.sh 10.179.2.139
0x01040ab3028b04040ab3028b080c31302e3137392e322e3133390904706b69780b0101
```

オプション DHCP オプション value

MikroTik RouterOS 7

```
/ip dhcp-server option
add code=43 name=nokia-oms
value=0x01040ab3028b04040ab3028b080c31302e3137392e322e3133390904706b6

/ip dhcp-server option sets
add name=nokia-set options=nokia-oms

/ip dhcp-server network
set [find address=10.7.15.64/26] dhcp-option-set=nokia-set dns-
server=10.7.15.65 gateway=10.7.15.65
```

Table

Field	Description
code=43	DHCP Option 43 Vendor Specific
value=0x...	Vendor RAN Monitor IP address 43 bytes
dhcp-option-set=nokia-set	Vendor Specific DHCP Option

dnsmasq

```
dhcp-
option=43,01:04:0a:b3:02:8b:04:04:0a:b3:02:8b:08:0c:31:30:2e:31:37:39:2e:32:2e:
```

ISC DHCP

```
option vendor-encapsulated-options
01:04:0a:b3:02:8b:04:04:0a:b3:02:8b:08:0c:31:30:2e:31:37:39:2e:32:2e:
```

--	--	--	--

--	--	--	--	--	--

□□□□□□ □□□□ **DHCP** □□□□ Web Element Manager □□

1.  Web Element Manager  <https://<base-station-ip>/>
2.    
3.     **DHCP**
4.      

00000000 00000000 00 0000 43 00000000 DHCP 000

□ □ □ □ □

SystemModuleEIF3

OAM の DHCP 送信元アドレスは VLAN ID — の送信元アドレスと
DHCP 宛先アドレスは VLAN ID

11

DHCP ☐

DHCP Mikrotik

```
/ip dhcp-server lease print where server=<dhcp-server-name>
```

MAC 78:F9:B4 Nokia Solutions and Networks

□ □ □ □

Web Element Manager

```
Dhcp offer Accepted
Identification Server Ip1: 10.179.2.139
Certificate Skipped
Ike Skipped
BTS backhaul connection Succeeded
Started agent discovery with identification IP: 10.179.2.139
```

RAN Monitor

□ RAN Monitor Web UI □□

1. **BTS**
2.
 - — PM
 - **PM** — PM
 - —
 - — NE3S
 - —

--	--	--	--

□□□□ DHCP

0000 00000000 DHCP Discover000000 Offer0

□ □ □ □ □ □ □ □

1. **43** **DHCP** — `dhcp-option-set`
2. “**DHCP**” — Web Element Manager
3. **RAN Monitor IP** — RAN Monitor

“ ”

□□□ DHCP □□□□□□□□□□ RAN Monitor□

□□□

- RAN Monitor □□□□□□□□□□□□□□□□
 - □□□ RAN Monitor □□□ 443 □□□□□□
 - □□□□□□ RAN Monitor □□□□□□
-

□□□□□

- **Nokia AirScale** □□□□ — □□□□□□□□□□
- □□□□□□□□ — RAN Monitor □□□□
- □□□□ — □□□□□□□□

概要

概要 **AirScale** のインストール

インストール手順

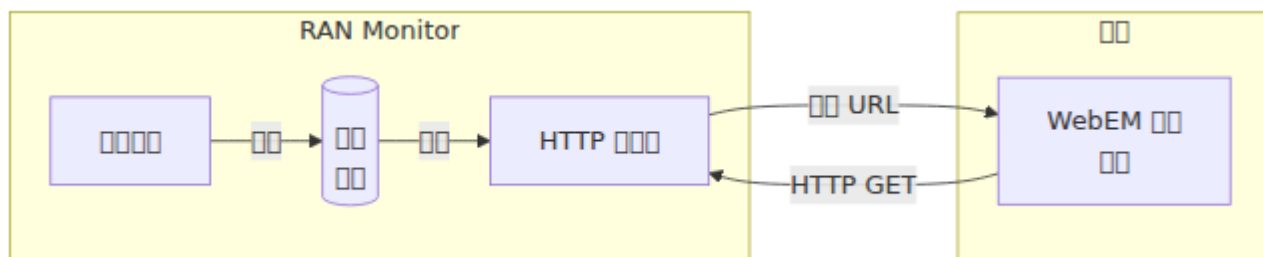
前提条件

インストール前に、RAN Monitor のインストールと HTTP のインストールが完了していることを確認してください。WebEM のインストールは RAN Monitor のインストール後に実行してください。

インストール

- インストール - AirScale .zip をダウンロード
- **HTTP** インストール - HTTP URL を入力
- **MD5** チェック - MD5 チェックを実行
- **URL** 入力 - WebEM のインストール URL を入力
- インストール - インストールを実行

インストール手順



インストール

URL: `https://<ran-monitor-ip>:9443/nokia/firmware`

root: / > cd

../../../../../../../../../../../../

root

root

root RAN Monitor root

```
/var/firmware/nokia/
├─ SBTS25R1_ENB_0000_001046_000000_release_BTSSM_downloadable_A54.zip
├─ SBTS25R1_ENB_0000_001046_000000_release_BTSSM_downloadable_A54.zip
├─
SBTS25R1_ENB_0000_001046_000000_release_BTSSM_downloadable_A54.zip.me
├─ SBTS24R1_ENB_0000_001093_000000_release_BTSSM_downloadable_A52.zip
├─ SBTS24R1_ENB_0000_001093_000000_release_BTSSM_downloadable_A52.zip
├─
SBTS24R1_ENB_0000_001093_000000_release_BTSSM_downloadable_A52.zip.me
└─ ...
```

root

- `.zip` - zip
- `.md5` - MD5
- `.metadata.json` - metadata

root

config/runtime.exs

```
config :ran_monitor,
  firmware_storage_path: "/data/firmware/nokia"
```

項目	項目名	項目値	項目説明
項目名	firmware_storage_path	/var/firmware/nokia	ファームウェアの保存パス

項目名

項目名: /var/firmware/nokia

Web UI

項目名

項目名

項目名	項目値
項目名	項目値
項目名	項目値
項目名	項目値

項目名

項目名 AirScale 項目名

- 項目名 .zip 項目名
- 項目名
- 項目 項目 項目

項目名

ダウンロードした `.zip` ファイルを開いて

約 15 GB

確認

- 約 15.37 MB/s
- 約 15.37 MB/s
- 約 15.37 MB/s
- 約 15.37 MB/s

ダウンロードしたファイルを開いて

SBTS<release>_ENB_<build>_<variant>_release_BTSSM_downloadable_<suffi

SBTS25R1_ENB_0000_001046_000000_release_BTSSM_downloadable_A54.zip

約 15.37 MB/s

約 15.37 MB/s

項目	説明
概要	プロジェクト概要
部品	部品リスト (例: SBTS25R1)
仕様	仕様書 (例: 仕様書)
図面	図面 (例: 図面)
材料	材料 (例: 材料)
MD5	MD5 ハッシュ (例: MD5)
リンク	リンク (例: リンク)
URL	URL (例: URL)

リンク

リンク先: [リンク先](#)

リンク先: [リンク先](#)

項目	内容
名称	無線通信機
型番	無線通信機型番SBTS25R1
製造番号	無線通信機製造番号
機種	無線機
機能	無線機 RAN Monitor 機能
バージョン	無線機バージョン番号FTM_SBTS25R1_2025.05.27_0097
PS 機能	無線機機能
プラットフォーム	Linux プラットフォーム
LTE 機能	LTE 無線機機能FL19A
WCDMA 機能	WCDMA 無線機機能WBTSFP20C
無線機機能	無線機機能ASCE、ASCF、FCTJ 機能
MD5 機能	無線機機能
無線機 URL	無線機 URL

無線通信機機能無線通信機機能無線通信機機能

--	--	--	--

[illegible]

□□ **URL** □

□□□□□□□□□□□□□□□□ URL□

```
http://<ran-monitor-ip>:4000/firmware/nokia/<filename>.zip
```

□ URL □□□ WebEM □“□□□□□”URL □□□□

MD5 ☐ ☐ ☐

1111

MD5 .md5

□ □ □ □ □

XXXXXXXXXXXX .md5 XXX

```

/var/firmware/nokia/
├── SBTS25R1_ENB_0000_001046_000000_release_BTSSM_downloadable_A54.zip
└── SBTS25R1_ENB_0000_001046_000000_release_BTSSM_downloadable_A54.zip.mc

```

.md5

a1b2c3d4e5f6789012345678abcdef01

11

MD5

Linux/macOS:

```
md5sum
SBTS25R1_ENB_0000_001046_000000_release_BTSSM_downloadable_A54.zip
```

Windows PowerShell

```
Get-FileHash -Algorithm MD5  
SBTS25R1 ENB 0000 001046 000000 release BTSSM downloadable A54.zip
```

□ □ □ □

- 4G RAN Monitor
- 3G RAN Monitor 4G RAN Monitor
- 4G WebEM 4G

□□□□



1. RAN Monitor

Download the RAN Monitor

- `https://<ran-monitor-ip>:9443/nokia/firmware`
- `.zip` file
- MD5 checksum

2. URL

Download the **URL** file

```
http://<ran-monitor-ip>:4000/firmware/nokia/<filename>.zip
```

3. WebEM

WebEM

- >
- BTS

4.

WebEM “”

- 1.
2. URL **URL**
- 3.

WebEM RAN Monitor

5.

- 1.
2. **SW**
- 3.

4. □□ □□ □□□□□□□□

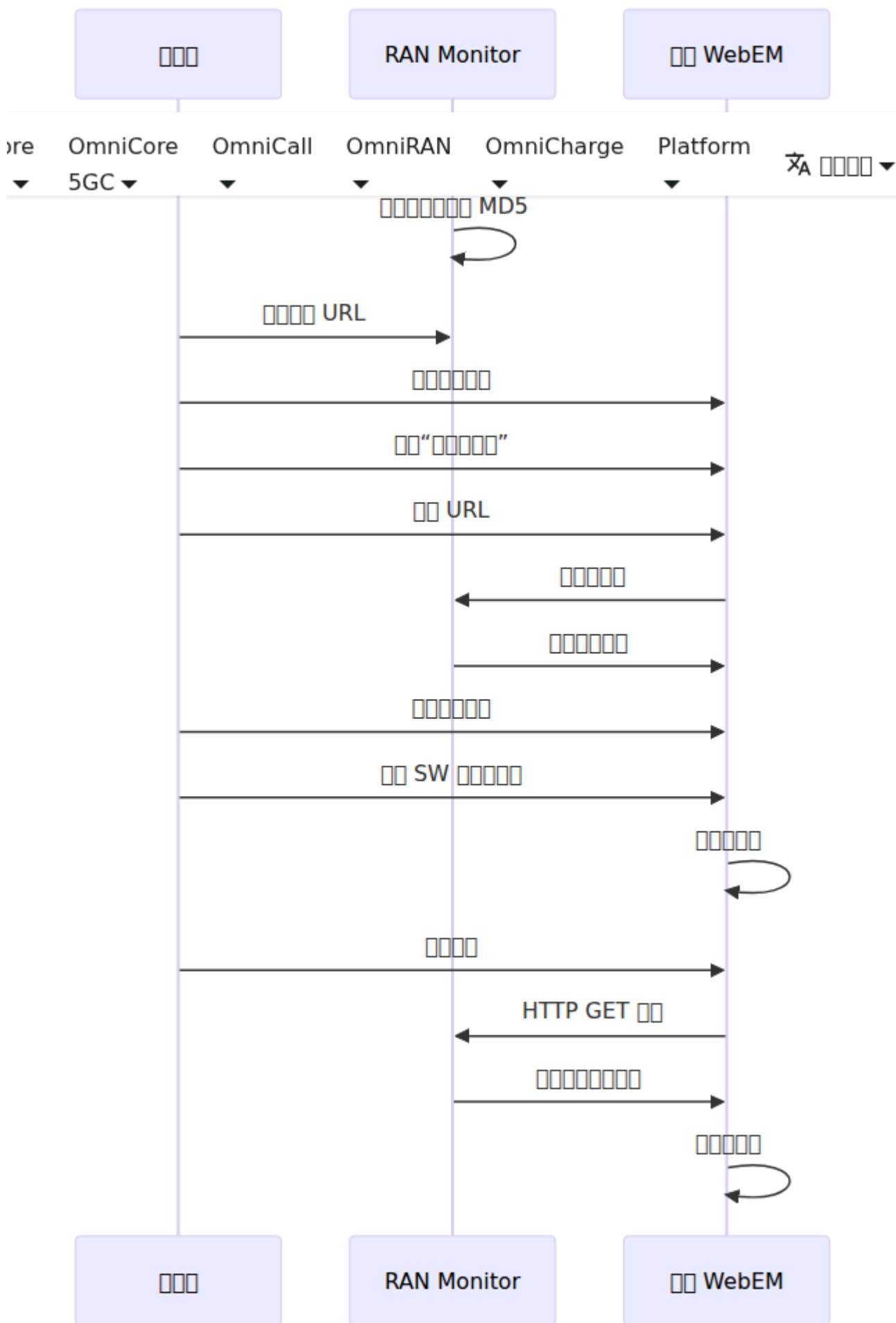
--	--	--	--	--

--	--	--	--	--	--

1. 透過 WebEM 查詢 訊
2. 利用 RAN Monitor 查詢 訊
3. 利用 KPI 查詢 訊

☐ RAN Monitor ☐ WebEM

□□□□



XML

XML URL

`sw_update.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<raml xmlns="raml21.xsd" version="2.1">
  <cmData type="plan" scope="changes">
    <managedObject class="com.nokia.srbts.btsswm:BTSSWM" distName="MRBTS-
operation="update">
      <p name="swPkgUrl">http://{ran-monitor-
ip}:4000/firmware/nokia/SBTS25R1_ENB_0000_001046_000000_release_BTSSM
      <p name="swPkgActivate">true</p>
    </managedObject>
  </cmData>
</raml>
```

Parameters

- `{site_id}` - MRBTS ID 256 2057
- `{ran-monitor-ip}` - RAN Monitor IP

Steps

1. RAN Monitor
- 2.
3. `sw_update.xml`
- 4.
- 5.

API

XML MRBTS ID RAN Monitor API

環境変数

項目	説明
swPkgUrl	ソフトウェアパッケージの HTTP URL
swPkgActivate	ソフトウェアパッケージをアクティブにするかどうか (true: true, false: false)

インストール

RAN Monitor をインストールするためのコマンドを実行します。

インストール

インストールコマンドを実行します。

項目	説明
IP 指定	インストールする IP 指定
インストール	インストールする AirScale の名前
インストール	インストールする

インストール

インストールする IP 指定を指定するコマンドを実行します。

インストールする AirScale の名前を指定するコマンドを実行します。

API 簡介

端點介紹

REST API 端點介紹

GET /api/firmware/nokia

返回

```
{
  "firmware": [
    {
      "filename": "SBTS25R1_ENB_0000_001046_000000_release_BTSSM_dowr",
      "size": 1073741824,
      "size_formatted": "1.0 GB",
      "modified": "2025-01-15T10:30:00Z",
      "download_url": "http://ran-monitor:4000/firmware/nokia/SBTS25R1_ENB_0000_001046_000000_release_E",
      "metadata": {
        "release": "SBTS25R1",
        "build": "0000_001046_000000",
        "variant": "release",
        "suffix": "A54",
        "type": "BTS 5G"
      }
    }
  ],
  "stats": {
    "file_count": 3,
    "total_size": 3221225472,
    "total_size_formatted": "3.0 GB",
    "storage_path": "/var/firmware/nokia"
  }
}
```

端點介紹

GET /firmware/nokia/<filename>

Content-Type: application/zip

URL

URL

```
config :ran_monitor,
  firmware_base_url: "http://ran-monitor.example.com:4000"
```

firmware_base_url			http://localhost:4000	URL URL RAN Monitor /IP

Web UI URL API URL HTTP

- 15 GB
- .zip
-

- 刪除檔案

刪除

1. 刪除 15 GB
2. 刪除 .zip
3. 刪除 `df -h /var/firmware/nokia`
4. 刪除 `ls -la /var/firmware`

刪除

WebEM “刪除”刪除

刪除

- RAN Monitor 刪除
- HTTP 4000
- URL IP 刪除

刪除

1. ping RAN Monitor
2. 4000
3. URL IP/刪除
4. URL

MD5 “...”

MD5 “...”

刪除

- 刪除
- .md5 刪除

刪除

1. 刪除

-
2. □□ □□ □□□□□□
3. □□□□□□□□□□□□□□

□ □ □ □ □ □ □

□□□ □□□□□□□□“□□□”□□

□□□□□

1.
2.
3.

📶📶📶 RAN Monitor

RAN Monitor 📶📶📶📶📶📶

📶📶📶📶📶📶📶 RAN Monitor

📶

1. 📶
 2. 📶📶
 3. 📶📶📶
 4. 📶
 5. 📶📶
-

📶

📶📶📶📶 RAN Monitor 📶📶📶📶📶📶📶📶

📶📶📶📶📶

📶📶📶📶📶

- ✓ 📶📶📶📶📶MySQL📶InfluxDB📶
- ✓ 📶📶📶📶📶📶 RAN Monitor
- ✓ 📶 RAN Monitor 📶📶
- ✓ 📶📶📶📶 Nokia AirScale 📶
- ✓ 📶📶📶📶 InfluxDB
- ✓ 📶 Web UI 📶📶

📶📶📶 📶📶📶📶 30-60 📶

概要

RAN Monitor のインストールと設定

前提条件

MySQL のインストール

- MySQL 5.7+ または MariaDB 10.3+
- RAN Monitor のインストール
- CREATE、SELECT、INSERT、UPDATE、DELETE の権限
- RAN Monitor のインストール
- データベースの作成

InfluxDB のインストール

- InfluxDB 1.8+ または 2.0+
- RAN Monitor のインストール
- Bucket/データベースの作成
- API のインストール (InfluxDB 2.x)
- データベースの作成

RAN Monitor のインストール

- Linux (Ubuntu 20.04+、CentOS 8+) のインストール
- RAM 4GB以上 8GB
- CPU 2 コア以上 4+ GHz
- ディスク容量 20GB
- MySQL と InfluxDB のインストール

インストール

インストール

- RAN Monitor → Nokia AirScale のインストール 8080
- Nokia のインストール → RAN Monitor のインストール 9076 の Webhook

- RAN Monitor → MySQLポート 3306
- RAN Monitor → InfluxDBポート 8086
- 管理 → RAN Monitor Web UIポート 9443

ポート

- 管理 8080 管理コンソール
- 管理 9076 管理Webhook 管理
- 管理 9443 管理HTTPS Web UI
- 管理 MySQL と InfluxDB

Nokia 管理

管理

- **IP** 管理 - 管理コンソール
- 管理 - 管理コンソール 8080
- 管理 - 管理 WebLM 管理コンソール
- 管理 - 管理ping 管理
- 管理 - 管理

管理

- 管理 - 管理コンソールPEM 管理
- 管理 - 管理コンソールDER 管理
- 管理 Nokia 管理 OpenSSL 管理

Grafana管理

- 管理Grafana 8.0+
- 管理 InfluxDB 管理
- 管理

部署指南

1 部署指南

1.1 MySQL 部署

1 RAN Monitor 部署

```
CREATE DATABASE ran_monitor CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci;
```

创建用户并授予权限

```
CREATE USER 'ran_monitor_user'@'%' IDENTIFIED BY  
'secure_password';  
GRANT CREATE, SELECT, INSERT, UPDATE, DELETE ON ran_monitor.* TO  
'ran_monitor_user'@'%';  
FLUSH PRIVILEGES;
```

通过 MySQL 客户端连接 RAN Monitor 数据库

```
mysql -h <mysql-host> -u ran_monitor_user -p ran_monitor
```

1.2 InfluxDB 部署

1 InfluxDB 1.x 部署

```
influx -execute 'CREATE DATABASE "nokia-monitor"'
```

2 InfluxDB 2.x 部署 bucket

```
influx bucket create -n nokia-monitor -o your-org
```

通过 InfluxDB 2.x API 部署 InfluxDB 2.x

```
influx auth create --org your-org --write-buckets
```

[illegible]

1.3 五五五五五

□□□□□□□□□□□□

```
# ping -c 1 10.7.15.66
ping 10.7.15.66

# telnet 10.7.15.66 8080
telnet 10.7.15.66 8080
```

MySQL & InfluxDB 数据库

```
# Test MySQL
telnet <mysql-host> 3306

# Test InfluxDB
curl http://<influxdb-host>:8086/ping
```

2 RAN Monitor

```
config/runtime.exs
```

2.1 背景

```
config/runtime.exs
```



```
config :ran_monitor, RanMonitor.Repo,  
  username: "ran_monitor_user",  
  password: "secure_password",  
  hostname: "mysql-host",  
  database: "ran_monitor",  
  stacktrace: true,  
  show_sensitive_data_on_connection_error: true,  
  pool_size: 10
```

2.2 InfluxDB 配置

配置 InfluxDB 连接

```
config :ran_monitor, RanMonitor.InfluxDbConnection,  
  auth: [  
    username: "monitor",  
    password: "influx_password" # 使用 InfluxDB 2.x 的 API  
  ],  
  database: "nokia-monitor",  
  host: "influxdb-host"
```

2.3 Web 配置

配置 Web 应用

```

# SOAP/API
config :ran_monitor, RanMonitor.Web.Endpoint,
  http: [ip: {0, 0, 0, 0}, port: 8080],
  check_origin: false,
  secret_key_base: "generate_with_mix_phx_gen_secret",
  server: true

# Web UI HTTPS
config :control_panel, ControlPanelWeb.Endpoint,
  url: [host: "0.0.0.0", port: 9443, scheme: "https"],
  https: [
    ip: {0, 0, 0, 0},
    port: 9443,
    keyfile: "priv/cert/omnitouch.pem",
    certfile: "priv/cert/omnitouch.crt"
  ]

# Webhook
config :ran_monitor, RanMonitor.Web.Nokia.Airscale.Endpoint,
  url: [host: "0.0.0.0"],
  http: [ip: {0, 0, 0, 0}, port: 9076],
  server: true

```

2.4 Nokia

```

config :ran_monitor,
  general: %{
    mcc: "001", # 中国移动
    mnc: "001"  # 中国移动
  },
  nokia: %{
    ne3s: %{
      webhook_url: "http://<ran-monitor-ip>:9076/webhook",
      private_key: Path.join(Application.app_dir(:ran_monitor,
"priv"), "external/nokia/ne.key.pem"),
      public_key: Path.join(Application.app_dir(:ran_monitor,
"priv"), "external/nokia/ne.cert.der"),
      reregister_interval: 30
    },
    airscales: [
      %{
        address: "10.7.15.66",
        name: "Site-A-BS1",
        port: "8080",
        web_username: "admin",
        web_password: "password"
      }
    ]
  }
}

```

2.5 配置 SSL 证书

配置 HTTPS Web UI 需要 SSL 证书

```

# 生成证书/私钥
openssl req -newkey rsa:2048 -nodes -keyout
priv/cert/omnitouch.pem \
  -x509 -days 365 -out priv/cert/omnitouch.crt

```

生成 CA 证书

生成证书链文件

3 安装部署

安装部署 RAN Monitor

3.1 安装部署

安装部署

```
mix ecto.migrate
```

安装部署

3.2 安装 RAN Monitor

安装部署

```
mix phx.server
```

安装部署

```
MIX_ENV=prod mix release  
_build/prod/rel/ran_monitor/bin/ran_monitor start
```

3.3 安装部署

安装部署

```
[info] Running RanMonitor.Web.Endpoint with cowboy  
[info] Running ControlPanelWeb.Endpoint with cowboy  
[info] Running RanMonitor.Web.Nokia.Airscale.Endpoint with cowboy  
[info] Starting RAN Monitor Manager  
[info] Connecting to InfluxDB...  
[info] InfluxDB connection established  
[info] Attempting registration with device: Site-A-BS1  
[info] Successfully registered with Site-A-BS1
```

安装部署

- Web 管理
 - 監視
 - InfluxDB 管理
 - 設定
-

目次

第 4 章 監視

監視

4.1 Web UI 管理

監視

`https://<ran-monitor-ip>:9443`

監視 RAN Monitor 監視

4.2 監視

Web UI 監視

1. 監視 監視 監視
2. 監視監視監視監視監視
3. 監視監視“監視”監視
4. 監視監視“監視”
5. 監視監視監視監視監視監視

監視監視/監視監視

- 監視監視
- 監視監視
- 監視監視監視監視
- 監視監視監視監視

4.3 安装 InfluxDB

Web UI

1. 安装 **InfluxDB** 包
2. 配置
3. 启动
4. 验证“安装”“配置”“启动”

安装 InfluxDB

```
# InfluxDB 1.x
influx -database 'nokia-monitor' -execute 'SELECT COUNT(*) FROM PerformanceMetrics'

# InfluxDB 2.x
influx query 'from(bucket:"nokia-monitor")
  |> range(start: -1h)
  |> filter(fn: (r) => r._measurement == "PerformanceMetrics")
  |> count()'
```

4.4 验证

验证安装

Web UI

1. 安装 验证
2. 验证“安装”
3. 验证

安装 `mix phx.server` 验证

4.5 验证

Web UI

1. 验证
2. 验证
 - 验证

- 部署到边缘设备
 - 部署到服务器
 - 部署到云
-

部署

部署 RAN Monitor 部署到边缘设备

部署

1. 部署

- 在 `config/runtime.exs` 中配置
- 部署到边缘设备
- 部署到云

2. 部署 Grafana 部署

- 部署 Grafana
- 部署 InfluxDB 部署
- 部署
- 部署 Grafana 部署

3. 部署

- 部署
- 部署
- 部署

4. 部署

- 部署 Web UI 部署
- 部署 Grafana 部署
- 部署
- 部署

環境構築

前提条件

- 最新の **Web UI** をインストール
- 最新の **コンテナエンジン** をインストール
- 最新の **コンテナエンジン** をインストール

インストール

- **コンテナエンジン** Web UI
- **コンテナエンジン** をインストール
- **コンテナエンジン** をインストール

設定

- **Grafana** をインストール
- **コンテナエンジン** をインストール
- **コンテナエンジン** Slack/PagerDuty

検証

- **CA** をインストール
- **コンテナエンジン** をインストール
- **config/runtime.exs** をインストール
- **コンテナエンジン** をインストール

環境構築

前提条件

- **コンテナエンジン** をインストール
- **コンテナエンジン** をインストール
- **コンテナエンジン** をインストール
- **コンテナエンジン** をインストール
- **コンテナエンジン** をインストール

環境構築

- 仮想マシンを構築
- 24 時間稼働させる
- ネットワーク
- 監視システム

監視

- Web UI を構築
 - 監視システム
 - Grafana を構築
 - 監視システム
-

運用

監視

- 監視システム - 監視システム
- Web UI 監視 - 監視システム
- 監視システム - 監視システム

運用

- 監視システム - 監視システム
- 監視システム - 監視システム
- AirScale 監視 - 監視システム

監視

監視

- 監視システムping
- 監視システム

- 8080 8080
-

InfluxDB

- InfluxDB
-
- API
- `curl http://<influxdb-host>:8086/ping`

Web UI

- HTTPS 9443
- SSL
- Web
-

- -
- **Web UI** -
- -
- -
- **AirScale** -
- **Grafana** -
- -
- -
- -

Grafana 部署指南

本文档介绍如何在 RAN 环境中部署 Grafana

Grafana 部署指南 KPI 部署指南

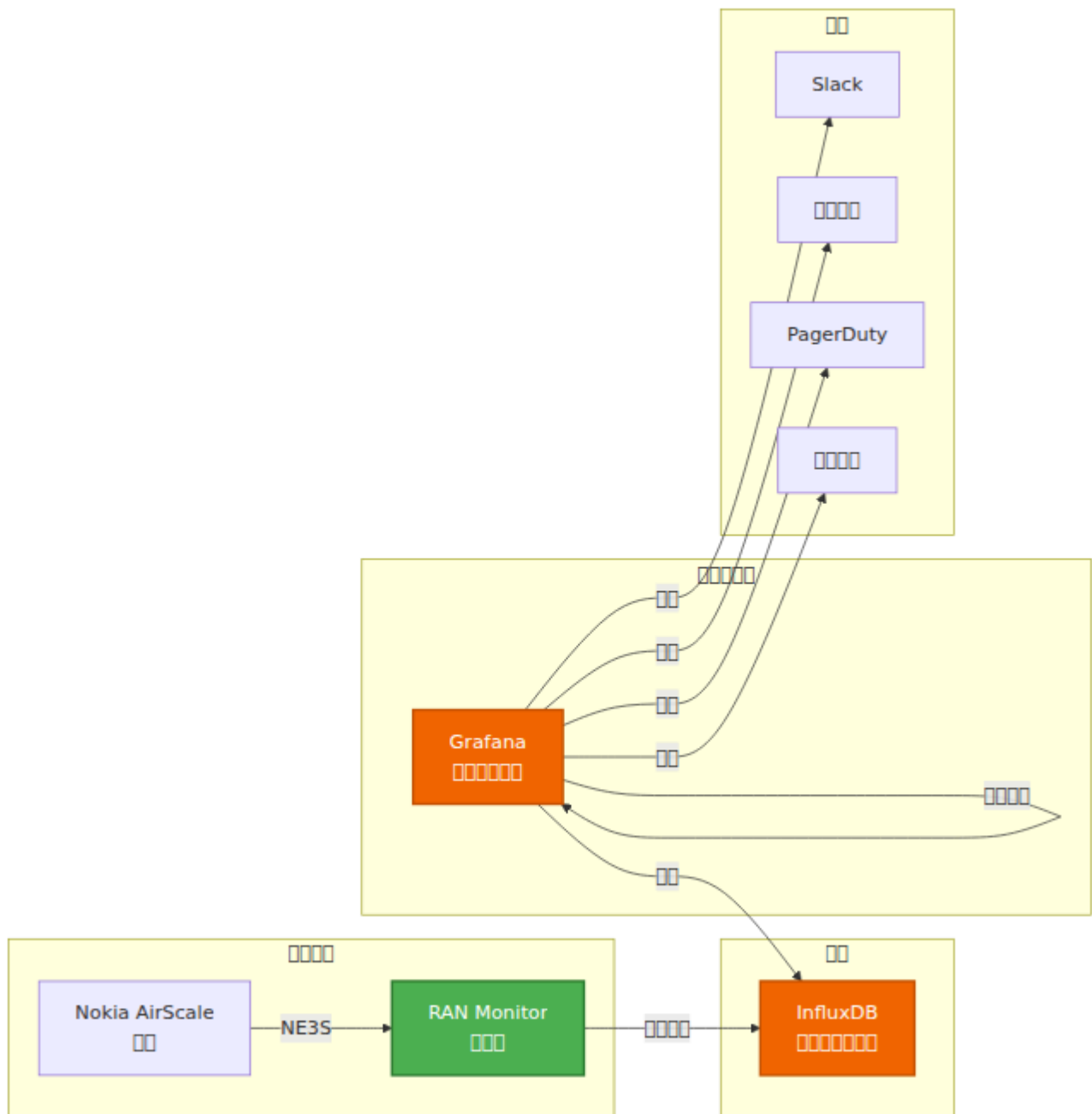
目录

1. 简介
 2. Grafana 与 InfluxDB 部署
 3. 部署环境
 4. 部署步骤
 5. 配置
 6. 验证
 7. 故障排除
 8. 附录
-

简介

Grafana 是一个开源的监控和可视化平台，用于展示来自各种数据源（如 InfluxDB、MySQL、PostgreSQL 等）的实时数据。在 RAN 环境中，Grafana 可以用于监控网络性能、设备状态、业务指标等。

--	--	--	--



Grafana

- 00000 - 0000000000000000
- 0000 - 0000/00/00000000
- 00 - 00000000000000
- 00000 - 0000000000000000000000
- 00 - 0000000000

目標

このドキュメントは、RAN Monitor/NOC (ONS) の KPI を Grafana で可視化する

前提条件

- RAN Monitor/NOC (ONS) のインストールが完了している
- Grafana のインストールが完了している
- Docker
- RAN Monitor/NOC (ONS) の Nokia のドキュメント

ONS の構成

- RAN Monitor/NOC (ONS) のインストール
- Grafana のインストール
- RAN Monitor/NOC (ONS) の KPI の可視化
- RAN Monitor/NOC (ONS) の KPI の可視化
- RAN Monitor/NOC (ONS) の KPI の可視化

このドキュメントは、RAN Monitor/NOC (ONS) の KPI を Grafana で可視化する

Grafana と InfluxDB のインストール

前提条件

前提条件

- InfluxDB 2.0+ と RAN Monitor/NOC (ONS) のインストールが完了している
- Grafana のインストールが完了している
- Grafana と InfluxDB の接続が正常に設定されている

Docker Compose のインストール

```

version: '3.8'
services:
  influxdb:
    image: influxdb:2.7
    environment:
      INFLUXDB_DB: ran_metrics
      INFLUXDB_ADMIN_USER: admin
      INFLUXDB_ADMIN_PASSWORD: change_me
    ports:
      - "8086:8086"
    volumes:
      - influxdb_data:/var/lib/influxdb2

  grafana:
    image: grafana/grafana:latest
    environment:
      GF_SECURITY_ADMIN_PASSWORD: change_me
    ports:
      - "3000:3000"
    depends_on:
      - influxdb
    volumes:
      - grafana_data:/var/lib/grafana
      - ./provisioning:/etc/grafana/provisioning

volumes:
  influxdb_data:
  grafana_data:

```

📦 InfluxDB API 📦

1. 📦 InfluxDB UI📦📦 8086📦
 2. 📦📦 API 📦
 3. 📦📦📦📦📦📦📦📦📦📦📦
 - 📦📦📦📦📦📦 ran_metrics 📦📦📦📦📦
 4. 📦📦📦📦
 5. 📦 Grafana 📦📦📦📦📦📦
-

設定

Grafana と InfluxDB の設定

- 1. 接続設定
 - Grafana → 設定 → 接続
- 2. データソース設定
 - 接続名“InfluxDB”
- 3. データソース

接続名	接続情報	接続先
接続名	RAN Monitor	Grafana データソース
URL	http://influxdb:8086	接続先 Grafana の URL
接続名	接続先	Grafana データソース
接続名	omnitouch	接続 InfluxDB の URL
接続名	API 接続	接続 API 接続
接続名	ran_metrics	RAN Monitor データソース
接続名	10s	接続先

- 4. データソース
 - 接続名“接続”
 - 接続名“接続先”

接続 InfluxDB の接続 URL は RAN Monitor の接続 InfluxDB の接続 URL
接続 接続先 AirScale の接続 接続先

Flux

Grafana Flux InfluxDB

```
from(bucket:"ran_metrics")
  |> range(start: -7d, stop: now())
  |> filter(fn: (r) => r._measurement == "PerformanceMetrics")
  |> filter(fn: (r) => r.device == "SITE_A_BS1")
  |> group(by: ["_field"])
  |> aggregateWindow(every: 1h, fn: mean)
```

- `from()` -
 - `range()` -
 - `filter()` -
 - `group()` -
 - `aggregateWindow()` -
-

□□□□□□□□

□□□□□□□□

□□□□
(CEO/□□)

□□

□□□□□
(NOC □□)

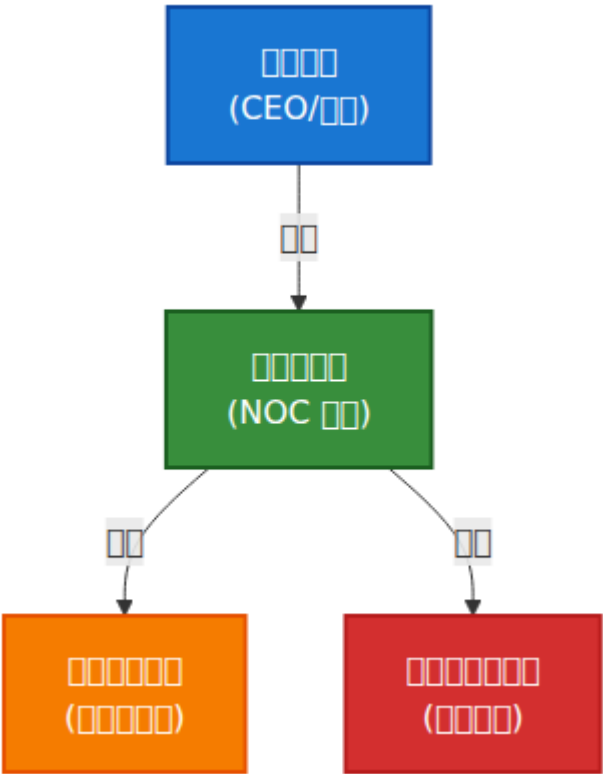
□□

□□

□□□□□□
(□□□□□)

□□□□□□
(□□□□)

組織体制



運用体制

運用体制の概要

運用体制の詳細

運用体制			
運用時間	運用場所	運用体制	運用体制
24時間365日	東京・大阪	24時間365日	24時間365日
48/50 (96%)	3 運用体制	98.5%	
運用体制 [2023] 運用			

運用

- 運用体制 < 10 運用体制
- 運用/運用体制

目標

このセッションでは Nokia のパフォーマンスメトリクスを InfluxDB に保存し、Nokia のパフォーマンスメトリクスを InfluxDB から取得し、それを可視化する。

前提条件

このセッションでは 24 時間

```
from(bucket:"ran_metrics")
  |> range(start: -24h)
  |> filter(fn: (r) => r._measurement == "PerformanceMetrics")
  |> filter(fn: (r) => r._field == "CellAvailability")
  |> group(by: ["device"])
  |> aggregateWindow(every: 1h, fn: mean)
  |> yield(name: "cell_availability")
```

結果

- パフォーマンス SLA
- パフォーマンスメトリクス
- パフォーマンス > 99.5%

このセッションでは 7 時間

```
from(bucket:"ran_metrics")
  |> range(start: -7d)
  |> filter(fn: (r) => r._measurement == "PerformanceMetrics")
  |> filter(fn: (r) => r._field =~ /Throughput.*/)
  |> group(by: ["device", "_field"])
  |> aggregateWindow(every: 10m, fn: mean)
  |> yield(name: "traffic_trend")
```

結果

- パフォーマンス

- 設定する
- 設定する

設定する

```
from(bucket:"ran_metrics")
  |> range(start: -1h)
  |> filter(fn: (r) => r._measurement == "PerformanceMetrics")
  |> filter(fn: (r) => r._field == "DLResourceUtilization")
  |> filter(fn: (r) => r.device == "SITE_A_BS1")
  |> aggregateWindow(every: 10s, fn: last)
  |> yield(name: "dl_resource")
```

設定

- 設定する
- 設定 80% 設定 95% 設定
- 設定する

設定

設定する **24** 設定

```
from(bucket:"ran_metrics")
  |> range(start: -24h)
  |> filter(fn: (r) => r._measurement == "Alarms")
  |> filter(fn: (r) => r.status == "active")
  |> group(by: ["severity"])
  |> count()
  |> yield(name: "alarm_count")
```

設定

- 設定する
- 設定
- 設定する

設定する

```
from(bucket:"ran_metrics")
  |> range(start: -7d)
  |> filter(fn: (r) => r._measurement == "Alarms")
  |> group(by: ["severity"])
  |> aggregateWindow(every: 1h, fn: count)
  |> yield(name: "alarm_rate")
```

📄

- 📄📄📄📄📄📄📄📄
- 📄📄📄📄📄📄
- 📄📄📄📄📄

📄📄📄📄📄

```
from(bucket:"ran_metrics")
  |> range(start: -7d)
  |> filter(fn: (r) => r._measurement == "Alarms")
  |> group(by: ["alarm_description"])
  |> count()
  |> sort(columns: ["_value"], desc: true)
  |> limit(n: 10)
  |> yield(name: "top_alarms")
```

📄

- 📄📄📄📄📄📄📄📄
- 📄📄📄📄📄📄
- 📄📄📄📄📄📄

📄📄📄📄

📄📄📄📄📄📄📄📄

```

from(bucket:"ran_metrics")
  |> range(start: -30d)
  |> filter(fn: (r) => r._measurement == "PerformanceMetrics")
  |> filter(fn: (r) => r._field == "CellAvailability")
  |> filter(fn: (r) => r.device == "SITE_A_BS1")
  |> aggregateWindow(every: 1h, fn: mean)
  |> statefulWindow(every: 1h, period: 24h)
  |> map(fn: (r) => ({r with _value: float(v: r._value)}))
  |> reduce(fn: (r, acc) => ({
    x: acc.x + [float(v: r._time)],
    y: acc.y + [r._value]
  })),
  initial: {x: [], y: []})
  |> yield(name: "availability_forecast")

```

□□□

- □□ SLA □□□□□□□□
- □□□□□□
- □□□□

□□□□□□□□□□□□

```

from(bucket:"ran_metrics")
  |> range(start: -7d)
  |> filter(fn: (r) => r._measurement == "PerformanceMetrics")
  |> filter(fn: (r) => r._field =~ /HandoverSuccess|Traffic/)
  |> group(by: ["device", "_field"])
  |> aggregateWindow(every: 1h, fn: mean)
  |> pivot(rowKey: ["_time"], columnKey: ["_field"], valueColumn:
    "_value")
  |> map(fn: (r) => ({r with correlation: float(v:
    r.HandoverSuccess) * float(v: r.Traffic)}))
  |> yield(name: "ho_traffic_correlation")

```

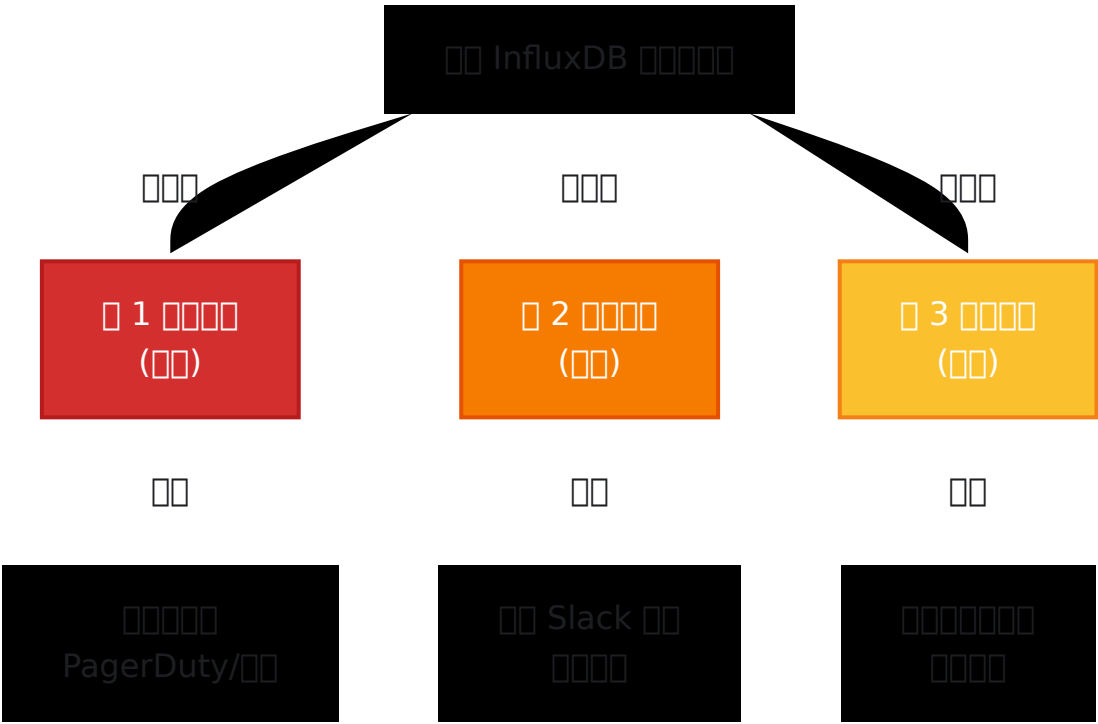
□□□

- □□□□□□□□□□□□□□
- □□□□□□□□

- 監視ツール

監視ツール

監視ツール



Grafana

1

1. 監視ツール
2. 監視ツール
3. 監視 → 監視
4. 監視 → 監視 → 監視

2

1


```
CellAvailability < 95%
    15
    1
    15
```

- 95% SLA 99.5%
- 15
-

2

```
count(active_alarms) > 10
    5
    2
    5
```

- 10
- 5
-

3

```
DLResourceUtilization > 90%
    30
    5
    30
```

-
- 30
- 5

■ 3 監視範囲を決定

1. 監視対象を決定
2. 監視範囲を決定 (Slack, PagerDuty など)
3. 監視範囲を決定

監視範囲を決定

```
rule {  
  name: ".AlertRuleName"  
  severity: ".Severity"  
  labels: {  
    device: ".Labels.device"  
  }  
  evalMatches: {  
    startsAt: ".StartsAt"  
  }  
  ruleUrl: ".RuleUrl"  
}
```

監視範囲

■ 1 監視範囲を決定

- 監視対象を決定 (SLA, 監視範囲)
- 監視範囲 1-5 監視
- 監視範囲 + Slack + PagerDuty
- 監視範囲 監視範囲
- **SLA** < 15 監視範囲

■ 2 監視範囲を決定

- 監視対象を決定 (監視範囲)
- 監視範囲 15-30 監視
- 監視範囲 Slack + 監視範囲 + PagerDuty
- 監視範囲 NOC 監視 + 監視範囲
- **SLA** < 30 監視範囲

■ 3 監視範囲を決定

- 監視対象を決定 (監視範囲)
- 監視範囲 1+ 監視

- 接続 Slack + 接続
- 接続 接続 / 接続
- **SLA** 接続

接続

Slack 接続

1. 接続 Slack 接続
2. 接続 URL
3. 接続 Grafana 接続 → 接続
4. 接続“Slack”接続
5. 接続 URL
6. 接続

Slack 接続

接続 接続 - SITE_A_BS1_Cell1
 接続45 接続
 接続~2000 接続
 接続 2:15

[接続] [接続] [接続]

PagerDuty 接続

1. 接続 PagerDuty 接続
2. 接続 Grafana 接続 → 接続
3. 接続“PagerDuty”接続
4. 接続
5. 接続
 - 接続 → 接続
 - 接続 → 接続
 - 接続 → 接続

接続

1. Grafana SMTP
2. →
3. “”
- 4.
5. CSV

1

5

4

- /
-
- %
- Gbps

2.

- **7**
- **7**
-

3.

-
- **7**

4.

- 
- 

□□□□□□

PRB

2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520 2521 2522 2523 2524 2525 2526 2527 2528 2529 2530 2531 2532 2533 2534 2535 2536 2537 2538 2539 2540 2541 2542 2543 2544 2545 2546 2547 2548 2549 2550 2551 2552 2553 2554 2555 2556 2557 2558 2559 2560 2561 2562 2563 2564 2565 2566 2567 2568 2569 2570 2571 2572 2573 2574 2575 2576 2577 2578 2579 2580 2581 2582 2583 2584 2585 2586 2587 2588 2589 2590 2591 2592 2593 2594 2595 2596 2597 2598 2599 2600 2601 2602 2603 2604 2605 2606 2607 2608 2609 2610 2611 2612 2613 2614 2615 2616 2617 2618 2619 2620 2621 2622 2623 2624 2625 2626 2627 2628 2629 2630 2631 2632 2633 2634 2635 2636 2637 2638 2639 2640 2641 2642 2643 2644 2645 2646 2647 2648 2649 2650 2651 2652 2653 2654 2655 2656 2657 2658 2659 2660 2661 2662 2663 2664 2665 2666 2667 2668 2669 2670 2671 2672 2673 2674 2675 2676 2677 2678 2679 2680 2681 2682 2683 2684 2685 2686 2687 2688 2689 2690 2691 2692 2693 2694 2695 2696 2697 2698 2699 2700 2701 2702 2703 2704 2705 2706 2707 2708 2709 2710 2711 2712 2713 2714 2715 2716 2717 2718 2719 2720 2721 2722 2723 2724 2725 2726 2727 2728 2729 2730 2731 2732 2733 2734 2735 2736 2737 2738 2739 2740 2741 2742 2743 2744 2745 2746 2747 2748 2749 2750 2751 2752 2753 2754 2755 2756 2757 2758 2759 2760 2761 2762 2763 2764 2765 2766 2767 2768 2769 2770 2771 2772 2773 2774 2775 2776 2777 2778 2779 2780 2781 2782 2783 2784 2785 2786 2787 2788 2789 2790 2791 2792 2793 2794 2795 2796 2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812 2813 2814 2815 2816 2817 2818 2819 2820 2821 2822 2823 2824 2825 2826 2827 2828 2829 2830 2831 2832 2833 2834 2835 2836 2837 2838 2839 2840 2841 2842

□□□ □□□□□□□□

□□□□ 10 □

111

1. □□□□□□□□

- ☐ □□□□□□□□□□□□□□
- ☐ □□□□□□□□□□□□

2. □□□□□□□□

- □□□□ %□□□□□
- □□□□ %□□□□□

- CPU %

3.

- / 24
- 24
-

4.

-
-

5.

- IP/
-
-

6.

- /
-
-

4G

3

1

1.

-
-
-

2.

- SINR
- RSRP
-

3.

- RLC 階層
- RRC 階層
- 無線電波

4. 無線電波

- 無線電波の伝播
- 無線電波の伝送

5. 無線電波の伝送

- 無線電波の伝送
- 無線電波の伝送
- 無線電波の伝送

無線電波 4 階層

無線電波の伝送

無線電波 5 階層

無線電波の伝送

無線電波

1. 無線電波の伝送

- 無線電波の伝送
- 無線電波の伝送/無線電波

2. 無線電波の伝送

- 無線電波の伝送
- 無線電波の伝送

3. 無線電波の伝送

- 無線電波の伝送
- 無線電波の伝送
- 無線電波の伝送

4. 測試項目

- 共 4 測試項目
- 測試
- 測試

5. 測試項目

- 共 30 測試項目
- 測試
- 測試

6. 測試項目

- 測試
- 測試
- 測試/測試

測試 5-Nokia AirScale 測試

測試 RF 測試項目

測試 30

測試 Nokia 測試 KPI

測試 Nokia 測試 AirScale 測試 Nokia 測試

測試 1測試項目

測試 PRB測試項目

```
// PRB
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["recordType"] == "performanceMetric")
  |> filter(fn: (r) => r["basebandName"] == "${Airscale}")
  |> filter(fn: (r) => r["metricCounter"] == "M8011C37")
  |> filter(fn: (r) => r._field == "counterValue")
  |> group()
  |> mean()
  |> map(fn: (r) => ({ r with _value: r._value / 10.0})) // 
  |> rename(columns: {_value: "PRB "})

// PRB
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["recordType"] == "performanceMetric")
  |> filter(fn: (r) => r["basebandName"] == "${Airscale}")
  |> filter(fn: (r) => r["metricCounter"] == "M8011C24")
  |> filter(fn: (r) => r._field == "counterValue")
  |> group()
  |> mean()
  |> map(fn: (r) => ({ r with _value: r._value / 10.0})) // 
  |> rename(columns: {_value: "PRB "})
```

PRB PRB

- PRB0-70%
- PRB70-85%
- PRB85-100%

PRB 2PRB

PRB PDCP PRB

```
// 平均値
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["recordType"] == "performanceMetric")
  |> filter(fn: (r) => r["basebandName"] == "${Airscale}")
  |> filter(fn: (r) => r["metricCounter"] == "M8012C26")
  |> filter(fn: (r) => r._field == "counterValue")
  |> map(fn: (r) => ({ r with _value: r._value / 1000.0})) // 単位
Mbps
  |> rename(columns: {_value: "平均 Mbps"})
```

```
// 平均値
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["recordType"] == "performanceMetric")
  |> filter(fn: (r) => r["basebandName"] == "${Airscale}")
  |> filter(fn: (r) => r["metricCounter"] == "M8012C23")
  |> filter(fn: (r) => r._field == "counterValue")
  |> map(fn: (r) => ({ r with _value: r._value / 1000.0})) // 単位
Mbps
  |> rename(columns: {_value: "平均 Mbps"})
```

まとめ

- **M8012C26** - PDCP 送信データレート/秒
- **M8012C23** - PDCP 受信データレート/秒

3GPP UE 仕様

仕様

```
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["recordType"] == "performanceMetric")
  |> filter(fn: (r) => r["basebandName"] == "${Airscale}")
  |> filter(fn: (r) => r["metricCounter"] == "M8018C1")
  |> filter(fn: (r) => r._field == "counterValue")
  |> rename(columns: {_value: "平均 UE"})
```

まとめ

- **M8018C1** - eNB UE

4

```
import "strings"

from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["recordType"] == "performanceMetric")
  |> filter(fn: (r) => r["basebandName"] == "${Airscale}")
  |> filter(fn: (r) => r["metricCounter"] == "M8020C3" or
                      r["metricCounter"] == "M8020C6" or
                      r["metricCounter"] == "M8020C4")
  |> filter(fn: (r) => r._field == "counterValue")
  |> filter(fn: (r) => strings.containsStr(v: r["DN"], substr:
"${CellKey}"))
  |> group()
  |> pivot(rowKey:["_time"], columnKey: ["metricCounter"],
valueColumn: "_value")
  |> map(fn: (r) => ({
    _time: r._time,
    "": 100.0 * r.M8020C3 / (r.M8020C6 - r.M8020C4)
  })))
```

- **M8020C3** -
- **M8020C6** -
- **M8020C4** -

< 99%

5 **PRB**

```
import "strings"

// 1 PRB 1000
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["recordType"] == "performanceMetric")
  |> filter(fn: (r) => r["basebandName"] == "${Airscale}")
  |> filter(fn: (r) => r["metricCounter"] == "M8011C24")
  |> filter(fn: (r) => r._field == "counterValue")
  |> filter(fn: (r) => strings.containsStr(v: r["DN"], substr:
"${CellKey}"))
  |> group()
  |> map(fn: (r) => ({ r with _value: r._value / 10.0}))
  |> rename(columns: {"_value": "1 PRB 1000"})

// 1000 PRB 1000
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["recordType"] == "performanceMetric")
  |> filter(fn: (r) => r["basebandName"] == "${Airscale}")
  |> filter(fn: (r) => r["metricCounter"] == "M8011C37")
  |> filter(fn: (r) => r._field == "counterValue")
  |> filter(fn: (r) => strings.containsStr(v: r["DN"], substr:
"${CellKey}"))
  |> group()
  |> map(fn: (r) => ({ r with _value: r._value / 10.0 }))
  |> rename(columns: {"_value": "1000 PRB 1000"})
```

6

PDCP □□□□□□□□

```

import "strings"

// PDCP トラフィック
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["recordType"] == "performanceMetric")
  |> filter(fn: (r) => r["basebandName"] == "${Airscale}")
  |> filter(fn: (r) => r["metricCounter"] == "M8012C26")
  |> filter(fn: (r) => r._field == "counterValue")
  |> filter(fn: (r) => strings.containsStr(v: r["DN"], substr:
"${CellKey}"))
  |> group()
  |> rename(columns: {"_value": "PDCP トラフィック"})

// PDCP トラフィック
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["recordType"] == "performanceMetric")
  |> filter(fn: (r) => r["basebandName"] == "${Airscale}")
  |> filter(fn: (r) => r["metricCounter"] == "M8012C23")
  |> filter(fn: (r) => r._field == "counterValue")
  |> group()
  |> rename(columns: {"_value": "PDCP トラフィック"})

```

7RSSI

```

import "strings"

// PUCCH RSSI
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["recordType"] == "performanceMetric")
  |> filter(fn: (r) => r["basebandName"] == "${Airscale}")
  |> filter(fn: (r) => r["metricCounter"] == "M8005C0")
  |> filter(fn: (r) => r._field == "counterValue")
  |> filter(fn: (r) => strings.containsStr(v: r["DN"], substr:
"${CellKey}"))
  |> group()
  |> rename(columns: {"_value": "PUCCH RSSI"})

// PUCCH RSSI
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["recordType"] == "performanceMetric")
  |> filter(fn: (r) => r["basebandName"] == "${Airscale}")
  |> filter(fn: (r) => r["metricCounter"] == "M8005C2")
  |> filter(fn: (r) => r._field == "counterValue")
  |> filter(fn: (r) => strings.containsStr(v: r["DN"], substr:
"${CellKey}"))
  |> group()
  |> rename(columns: {"_value": "PUCCH RSSI"})

// PUCCH RSSI
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["recordType"] == "performanceMetric")
  |> filter(fn: (r) => r["basebandName"] == "${Airscale}")
  |> filter(fn: (r) => r["metricCounter"] == "M8005C1")
  |> filter(fn: (r) => r._field == "counterValue")
  |> filter(fn: (r) => strings.containsStr(v: r["DN"], substr:
"${CellKey}"))
  |> group()
  |> rename(columns: {"_value": "PUCCH RSSI"})

```

📊📊📊📊

- **M8005C0** - PUCCH PUCCH RSSI[dBm]
- **M8005C1** - PUCCH PUCCH RSSI[dBm]

- **M8005C2** - PUCCH RSSI [dBm]

8

PDCP SDU

```
import "strings"

from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["recordType"] == "performanceMetric")
  |> filter(fn: (r) => r["basebandName"] == "${Airscale}")
  |> filter(fn: (r) => r["metricCounter"] == "M8001C2")
  |> filter(fn: (r) => r._field == "counterValue")
  |> filter(fn: (r) => strings.containsStr(v: r["DN"], substr:
"${CellKey}"))
  |> group()
  |> rename(columns: {"_value": ""})
```

- **M8001C2** - PDCP SDU DTCH

9 RRC


```

import "strings"

from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["recordType"] == "performanceMetric")
  |> filter(fn: (r) => r["basebandName"] == "${Airscale}")
  |> filter(fn: (r) => r["metricCounter"] == "M8013C5" or
                      r["metricCounter"] == "M8013C17" or
                      r["metricCounter"] == "M8013C18" or
                      r["metricCounter"] == "M8013C19" or
                      r["metricCounter"] == "M8013C34" or
                      r["metricCounter"] == "M8013C31" or
                      r["metricCounter"] == "M8013C21" or
                      r["metricCounter"] == "M8013C93" or
                      r["metricCounter"] == "M8013C91")
  |> filter(fn: (r) => r._field == "counterValue")
  |> filter(fn: (r) => strings.containsStr(v: r["DN"], substr:
"${CellKey}"))
  |> group()
  |> pivot(rowKey:["_time"], columnKey: ["metricCounter"],
valueColumn: "_value")
  |> map(fn: (r) => ({
    _time: r._time,
    "RSSI": 100.0 * r.M8013C5 / (r.M8013C17 + r.M8013C18 +
r.M8013C19 + r.M8013C34 + r.M8013C31 + r.M8013C21 + r.M8013C93 +
r.M8013C91)
  })))

```

📶📶📶📶📶

- **M8013C5** - 📶📶📶📶📶
- **M8013C17-M8013C93** - 📶📶📶📶📶

📶📶📶 📶📶📶 < 95%

📶 **10**📶📶 **VSWR**📶📶📶📶📶📶📶

📶📶📶📶📶📶📶📶

```
import "strings"

from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["recordType"] == "performanceMetric")
  |> filter(fn: (r) => r["basebandName"] == "${Airscale}")
  |> filter(fn: (r) => r["metricCounter"] == "M40001C0")
  |> filter(fn: (r) => r._field == "counterValue")
  |> filter(fn: (r) => strings.containsStr(v: r["DN"], substr:
"${RadioKey}"))
  |> map(fn: (r) => ({
    r with
    "DN": strings.split(v: r["DN"], t: "/")[5],
    "VSWR": r._value / 10.0
  }))
  |> group()
  |> pivot(rowKey: ["_time"], columnKey: ["DN"], valueColumn:
"VSWR")
```

📊📊📊📊

- **M40001C0** - 📊📊📊📊 VSWR📊0.1 📊📊

📊📊📊📊 VSWR > 2.0

📊 **11**📊📊📊📊📊📊📊

📊📊📊📊📊

```
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["recordType"] == "performanceMetric")
  |> filter(fn: (r) => r["basebandName"] == "${Airscale}")
  |> filter(fn: (r) => r["metricCounter"] == "M40002C2")
  |> filter(fn: (r) => r._field == "counterValue")
  |> group()
  |> map(fn: (r) => ({ r with _value: r._value / 100000.0 }))
  |> rename(columns: {"_value": "📊"})
```

📊📊📊📊📊

- **M40002C2** - 100000

Grafana

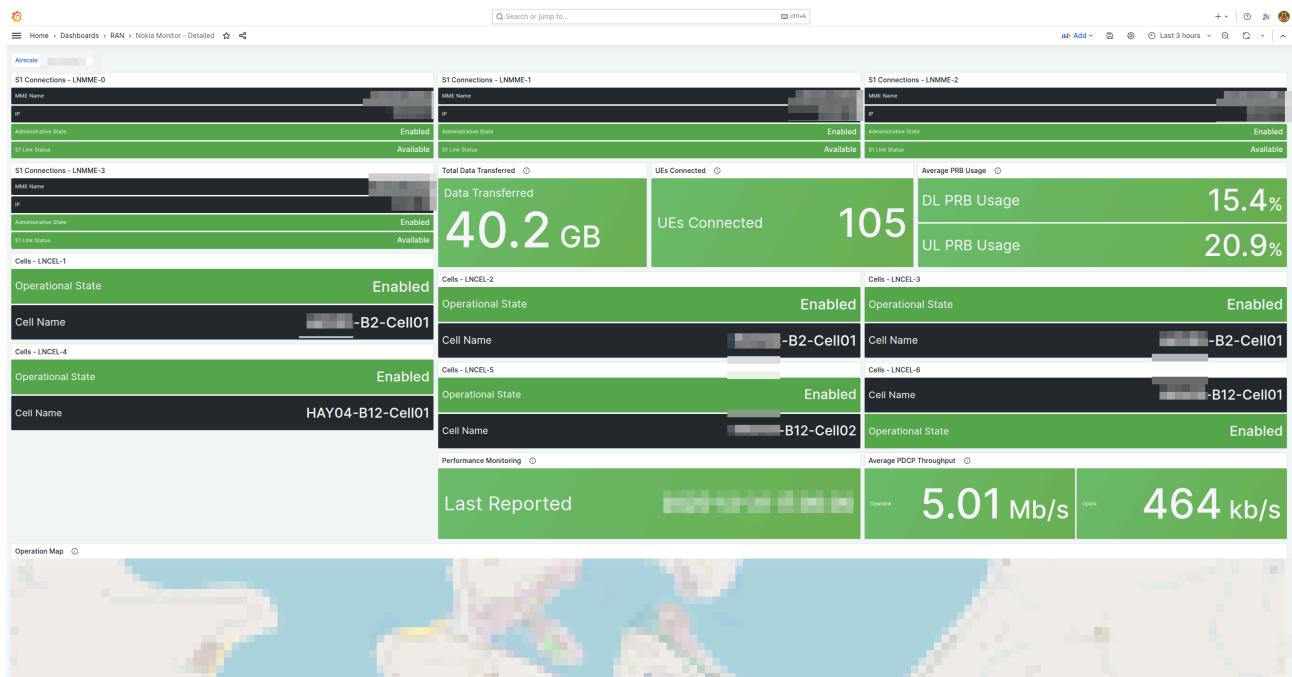
- **\${Airscale}** -
- **\${CellKey}** -
- **\${RadioKey}** - VSWR

1. **PRB** - DL UL PRB > 85% 5
2. - < 99% 10
3. - RRC < 95% 5
4. **VSWR** - VSWR > 2.0 15
5. - > 20%

1. - PRB
2. - RSSI
3. - VSWR
4. - SLA

Nokia

-



Nokia Monitor S1 PRB

PRB

LNCCEL LTE PRB TTI PDCP

RSSI RRC

VSWR RMOD-2 ANT1-1ANT1-2ANT1-3ANT1-4

--	--	--	--

--	--	--	--	--	--	--	--

111

- □□□□□□□□□□“□□□□”
- □□□□□□□□□□

111

1. `pip install influxdb`
2. `pip install influxdb-client`
3. `pip install influxdb-python`
4. `pip install influxdb2-python`

□□□□□

- InfluxDB UI 查看数据
- 数据保留时间 24 小时

- 配置 RAN Monitor 参数
- 配置 RAN Monitor 参数

配置 RAN Monitor

配置

- 配置 RAN Monitor > 5
- 配置 RAN Monitor

配置

1. 配置 RAN Monitor > 8
2. 配置 RAN Monitor/配置 RAN Monitor
3. InfluxDB 配置
4. 配置

配置

- 配置 RAN Monitor
- 配置 RAN Monitor 24 小时 vs. 1 小时
- 配置 InfluxDB 配置
- 配置 InfluxDB CPU/配置
- 配置 RAN Monitor

配置 RAN Monitor

配置

- 配置 RAN Monitor
- 配置 RAN Monitor
- 配置 RAN Monitor

配置

1. 環境構築
2. データ収集
3. データ処理
4. データ可視化

環境構築

- Dockerコンテナ環境 → 構築
- Prometheus → 導入
- Grafana URL/ポート
- Grafana 接続設定
- データソース設定

データ収集

ソース

- ログファイル
- データベース
- API

ターゲット

1. データベース
2. データベース
3. データベース/ファイル
4. データベース/ファイル

データ処理

- データベース InfluxDB 接続
- aggregateWindow 関数/関数
- InfluxDB データベース
- データベース

無線ネットワーク

概要

- 無線ネットワークの種類
- 無線ネットワークの構成
- 無線ネットワークの性能

無線ネットワーク

- 無線ネットワークの種類
- 無線ネットワークの構成
- 無線ネットワークの性能

無線ネットワーク

概要

- InfluxDB の概要
- 無線ネットワーク eNodeB の概要
- 無線ネットワークの性能

無線ネットワーク

- 無線ネットワークの種類
- 無線ネットワーク AirScale の概要
- 無線ネットワーク eNodeB の概要

無線ネットワーク

- Nokia の無線ネットワーク - 無線ネットワークの種類
- 無線ネットワークの種類 - 無線ネットワークの種類
- 無線ネットワークの種類 - 無線ネットワークの種類
- AirScale の無線ネットワーク - 無線ネットワークの種類

無線LAN

無線 LAN 接続方法

無線 LAN 接続方法

無線 LAN

- 無線 LAN
 - 無線 LAN
 - 無線 LAN
 - 無線 LAN
 - 無線 LAN
 - 無線 LAN
 - 無線 LAN
 - 無線 LAN
 - 無線 LAN
 - 無線 LAN
-

無線 LAN

無線 LAN 接続方法 RAN 無線 LAN 接続方法 NOC 無線 LAN 接続方法

無線 LAN

- 無線 LAN 接続方法
- 無線 LAN 接続方法
- 無線 LAN 接続方法
- 無線 LAN 接続方法

無線 LAN 接続方法

目標

確認 Nokia AirScale のインストールが完了している

ステップ 1: 確認

確認   インストールが完了している

```
# 確認
ping <base-station-ip>

# インストールが完了している
telnet <base-station-ip> 8080
```

確認 ping 確認 telnet 確認

確認

- 確認
- インストールが完了している
- インストールが完了している

ステップ 2: 確認

確認

項目	値	説明
IP 番号	10.7.15.67	管理用 IP 番号
ポート	8080	管理用ポート Nokia AirScale の 8080
サイト名	Site-B-Tower-1	管理用サイト名
ユーザ名	admin	管理用ユーザ名
パスワード	password123	管理用パスワード

管理用ユーザ名

- 管理用ユーザ名
- 管理用パスワード
- 管理用サイト名
- 管理用 IP 番号

3. 管理用ユーザ名

管理用ユーザ名

1. 管理用 Web UI: `https://<ran-monitor-ip>:9443`
2. 管理用ユーザ名 **eNodeBs**
3. 管理用 IP 番号 ID
4. 管理用パスワード ID

管理用ユーザ名 RAN ユーザ

4. 管理用ユーザ名

管理用 `config/runtime.exe` ユーザ名 `airscales` ユーザ名

```

config :ran_monitor,
  nokia: %{
    ne3s: %{
      # ... ne3s ...
    },
    airscales: [
      # 
      %{
        address: "10.7.15.66",
        name: "Site-A-BS1",
        port: "8080",
        web_username: "admin",
        web_password: "password1"
      },

      # 
      %{
        address: "10.7.15.67",           # IP 
        name: "Site-B-Tower-1",         # 
        port: "8080",                  # 
        web_username: "admin",          # WebLM 
        web_password: "password123"    # WebLM 
      }
    ]
  }
}

```

Elxir -

5

```
elixir -c config/runtime.exs
```

-

-  { 
- 
- 

6 RAN

□□□□□□□□□□□□

```
# 编译并安装
# 按 Ctrl+C 退出编译
mix phx.server

# 启动并查看日志
systemctl restart ran_monitor

# 重启服务
/path/to/ran_monitor/bin/ran_monitor restart
```

00 **7**00000000

□ □ □ □ □ □ □ □ □ □ □ □ □ □

1.

```
[info] 000000000000Site-B-Tower-1
[info] 0000 Site-B-Tower-1
```

2. Web UI

- 0000 00 00
- 00000000000000
- 0000 "000"0000
- 000000 "000"

3. □□□□□□□□

- □□□□

- 資料庫連線測試
- 新增 "測試" 資料庫

4. 新增 InfluxDB 資料庫

- 新增 InfluxDB 資料庫
- 資料庫名稱
- 資料庫類型

第 8 步 測試資料庫

測試資料庫連線是否成功

1. 新增 測試 資料庫
2. 測試資料庫連線
3. 新增 "測試" 資料庫
4. 測試成功

測試成功! 資料庫連線成功

第 9 步 新增 Grafana 資料庫

新增 Grafana 資料庫

1. 新增 測試 資料庫
2. 測試資料庫連線
3. 測試成功

測試成功! Grafana 資料庫連線成功

測試成功

測試成功! 資料庫連線成功

00 1 000000000000

000000000000000000

00 A00000

- 000000000000
- 0000000000000000

00 B00000

- 000000000000
- 00 InfluxDB 00
- 000 - 000000

00 2 000000000000

00000000000000

00 config/runtime.exe 00 airscales 0000000000000000

```
airscales: [  
  {%  
    address: "10.7.15.66",  
    name: "Site-A-BS1",  
    port: "8080",  
    web_username: "admin",  
    web_password: "password1"  
  },  
  
  # 0000 - 00000000  
  # {%  
  #   address: "10.7.15.67",  
  #   name: "Site-B-Tower-1",  
  #   port: "8080",  
  #   web_username: "admin",  
  #   web_password: "password123"  
  # }  
]
```


第 3 步 部署 RAN 节点

部署 RAN 节点

1. 部署 Web UI 界面 部署 部署
2. 部署 RAN 节点
3. 部署 RAN 节点 部署
4. 部署 RAN 节点

部署 RAN 节点

第 4 步 部署 RAN 节点

部署 RAN 节点

```
systemctl restart ran_monitor  
#  
mix phx.server
```

第 5 步 部署 RAN 节点

部署 RAN 节点

1. 部署 RAN 节点
 - 部署 RAN 节点
 - 部署 RAN 节点
2. 部署 RAN 节点
 - 部署 RAN 节点
 - 部署 RAN 节点
3. 部署 InfluxDB
 - 部署 InfluxDB
 - 部署 InfluxDB

第 6 章 Grafana 部署

1 Grafana 部署环境

1. 部署环境
2. 部署环境
3. 部署环境

部署环境

部署环境 RAN 环境

第 1 步 部署环境

部署环境

1. 部署环境
2. 部署环境
3. 部署环境

第 2 步 部署环境

在 `config/runtime.exs` 文件中

```
airscales: [  
  %{  
    address: "10.7.15.66",  
    name: "Site-A-BS1",  
    port: "8080",  
    web_username: "admin",  
    web_password: "new_password_here" # 密码  
  }  
]
```

ステップ 3 RAN モニタリング

確認作業

```
systemctl restart ran_monitor
```

ステップ 4 ログの確認

確認

1. ログの確認

```
[info] 2024-01-15 10:00:00 Site-A-BS1  
[info] 2024-01-15 10:00:01 Site-A-BS1
```

2. Web UI

- ログレベル "info"を確認
- ログを確認
- "2024-01-15"を確認

確認

- ログを確認
- ログを確認
- ログを確認
- ログを確認

確認完了

RAN モニタリングが正常に完了しました。

概要

RAN 監視システム概要

項目	単位	値	説明
監視対象	10 分	1	監視対象範囲/監視対象
監視周期	10 分	1	監視周期
監視時間	60 分	1	監視時間
監視回数	30 分	1	監視回数

監視対象

監視対象範囲

- 監視対象範囲
- 監視対象範囲
- 監視対象 SLA 値
- 監視対象範囲

監視対象範囲

- 監視対象範囲
- 監視 InfluxDB 監視対象
- 監視対象範囲
- 監視対象範囲

監視 1 監視対象

監視対象範囲 runtime.exs 監視対象範囲監視対象範囲

監視対象範囲監視対象範囲

- 監視対象 `lib/ran_monitor/nokia/airscale/manager.ex`

- `lib/ran_monitor/nokia/airscale/manager.ex`
- `lib/ran_monitor/nokia/airscale/manager.ex`

OmniTouch 環境構築

2. 環境構築

環境構築

環境構築

- $\text{環境構築} \times \text{環境構築} \times \text{環境構築} = \text{環境構築}$
- $\text{環境構築} = \text{環境構築}$
- 環境構築環境構築

環境構築

- $\text{環境構築} \times \text{環境構築} = \text{環境構築}$
- $\text{環境構築} 10 \text{ 環境構築} = \text{環境構築} 8,640 \text{ 環境構築}$
- 環境構築 InfluxDB 環境構築

環境構築

- $\text{環境構築} = \text{RAN 環境構築 CPU 環境構築}$
- 環境構築環境構築

3. 環境構築

環境構築

1. 環境構築環境構築
2. 環境構築
 - RAN 環境構築 CPU 環境構築
 - 環境構築
 - InfluxDB 環境構築 I/O 環境構築
3. 環境構築
 - 環境構築 InfluxDB 環境構築

- 無線ネットワーク

4. ネットワーク

- ネットワークの種類
- ネットワークの構成

無線ネットワーク

無線 LAN ネットワーク

無線 LAN

無線 LAN ネットワークの Web UI - 無線 LAN

無線 LAN

1. 無線 LAN ネットワーク
2. 無線 LAN
3. 無線 LAN - 無線 LAN ID
4. 無線 LAN ID ネットワーク
5. 無線 LAN ネットワーク
6. 無線 LAN

無線 LAN

無線 LAN

- 無線 LAN ネットワーク
- 無線 LAN
- 無線 LAN

無線 LAN

- 無線 LAN ネットワーク
- 無線 LAN
- 無線 LAN

環境構築

- 仮想マシンをインストール
- ネットワーク設定
- 必要なパッケージをインストール

インストール

- 依存関係を満たす
- 実行 ID を取得
- 実行権限を付与
- サービスを起動

確認

- 実行確認
- ログを確認
- 実行回数 15-30 回
- 実行結果を確認

デモ実行

実行環境 RAN 環境構築

実行方法

実行コマンド

1. Web UI 実行

```
https://<ran-monitor-ip>:9443
```

2. ログ確認

- ログファイルを確認

-
- 

3. □□□□□□

- 
- 
- 

4.

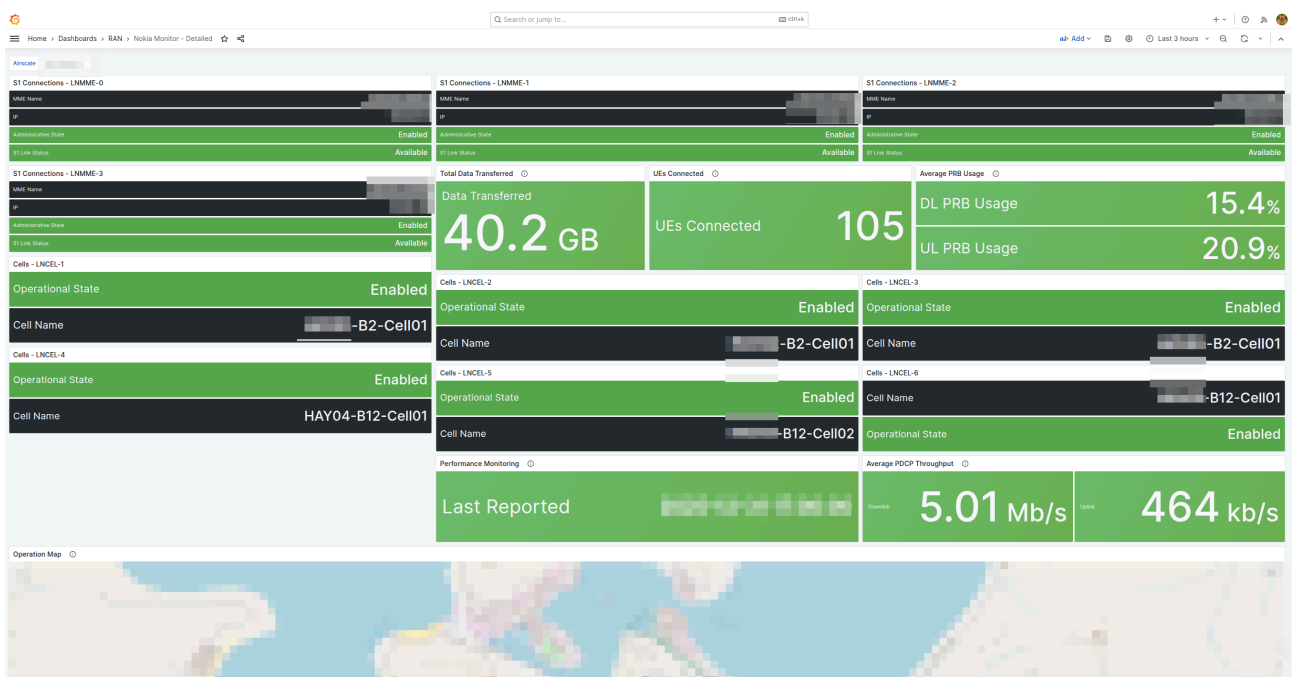
- InfluxDB
- 
-

5. □□□□□□

- □□□□□□□□
- □□ "□□" □□
- □□□□□□□□

Web UI 00 - Web UI 000000

Grafana



LMNME S1 UEs PRB

1.

- Grafana
-
-

2.

- InfluxDB
- MySQL
-

3.

-
-
-

4.

- RAN CPU
-
-

5.

-
-
-

30-45

環境構築

前提条件

OS: Ubuntu 20.04 LTS

バージョン

インストール

- Git
- Python 3.8

確認

- Git
- ファイル `config/config.exs` を確認

実行

- コマンド "start" を実行
- 確認

結果

- インフラ InfluxDB を確認
- 確認
- 確認

確認

確認

- 確認
- ファイル "start"
- XML を確認

4. 导出数据到CSV

使用 **InfluxDB** 的

```
# 导出数据到CSV
influx -database 'nokia-monitor' -execute '
  SELECT * FROM PerformanceMetrics
  WHERE basebandName=''Site-A-BS1'''
  AND time > now() - 7d
' -format csv > export.csv
```

使用 **Grafana** 查看

- 配置数据源
- 创建仪表盘
- 添加 "Influx" - "CSV" 数据源

部署与配置

部署

部署步骤

1. 克隆仓库

```
# 克隆仓库
cp config/runtime.exs backups/runtime.exs.$(date +%Y%m%d)

# 打包配置
tar -czf backups/config-$(date +%Y%m%d).tar.gz config/
```

2. 安装依赖

- 安装 Web UI 依赖
- 安装数据库依赖
- 安装监控工具/脚本

3. MySQL 备份

```
# 备份MySQL数据库
mysqldump -u ran_monitor_user -p ran_monitor >
backups/ran_monitor-$(date +%Y%m%d).sql
```

4. InfluxDB 备份

```
# InfluxDB 1.x 备份
influxd backup -portable -database nokia-monitor
/backups/influx-$(date +%Y%m%d)

# InfluxDB 2.x 备份
influx backup /backups/influx-$(date +%Y%m%d)
```

5. SSL 备份

```
cp priv/cert/* backups/certificates-$(date +%Y%m%d)/
```

备份策略

MySQL

- MySQL 数据库备份
- 备份策略

InfluxDB

- InfluxDB 数据库备份
- 备份策略

SSL 证书

- 证书备份
- 证书策略
- 证书管理

步骤四

停止 RAN 服务

1. 停止 RAN 服务

```
systemctl stop ran_monitor
```

2. 备份配置文件

```
cp backups/runtime.exs.20251230 config/runtime.exs
```

3. 启动服务

```
elixir -c config/runtime.exs
```

4. 启动 RAN 服务

```
systemctl start ran_monitor
```

5. 验证服务状态

恢复 RAN 服务

1. 停止 RAN 服务

```
systemctl stop ran_monitor
```

2. 恢复 MySQL 数据

```
mysql -u ran_monitor_user -p ran_monitor < backups/ran_monitor-20251230.sql
```

3. 启动 RAN 服务

```
systemctl start ran_monitor
```

4. 設定ファイル

5. 設定ファイル

6. 設定ファイル InfluxDB

設定ファイル

1. 設定ファイル RAN

2. 設定ファイル

3. MySQL 設定

4. InfluxDB 設定

5. SSL 設定

6. RAN 設定

7. 設定ファイル

8. 設定ファイル

設定

設定

設定

1. 設定

- 設定ファイル
- 設定ファイル
- 設定ファイル

2. 設定

- 設定ファイル
- 設定ファイル

- 0000000000

3. 0000

- 0000000000
- 000000000000
- 0000000000

4. 0000

- 00000000
- 00000000
- 000000000000

0000

1. 000000

- 00000000
- 00000000
- 00000000

2. 0000

- 000000
- 000000
- 0000000000 SSL 00

3. 0000

- 00000000
- 0000000000
- 000000000000

000

1. SSL 0000

- 00000000 SSL 00

- 部署環境構築

2. 構築

- 環境構築
- 環境構築
- API 呼び出し

3. 構築

- RAN 構築
- 構築
- 構築

構築

構築

1. 構築

- 構築
- Grafana 構築

2. 構築

- 構築
- 構築
- 構築

3. 構築

- 構築
- 構築
- 構築

4. 構築

- 構築
- 構築

- 環境構築

5. 環境構築

- 環境構築
- 環境構築
- 環境構築
- 環境構築

6. 環境構築

- 環境構築
- 環境構築
- 環境構築

環境構築

- 環境構築 - 環境構築
- **Web UI** 環境 - 環境構築
- 環境構築 - 環境構築
- **AirScale** 環境 - 環境構築
- 環境構築 - 環境構築
- **Grafana** 環境構築 - 環境構築
- 環境構築 - 環境構築
- 環境構築 - 環境構築
- 環境構築 - 環境構築
- 環境構築 - 環境構築

部署 LTE 网络

部署 AirScale/FlexiRadio 网络

部署

1. 部署
 2. 部署
 3. 部署
 4. 部署
 5. 部署
 6. 部署
 7. 部署
 8. 部署
 9. 部署
 10. 部署 QoS
 11. 部署
 12. 部署 Grafana
 13. 部署
-

部署

部署 LTE 网络 AirScale/FlexiRadio 网络

部署

部署

- 部署
- 部署

- 
- 
- 

111

00000000000000000000000000000000 LTE 000000000000

規格項目 (M8xxx)

項目	項目	規格値	説明
M8000	S1 状態	33	ネットワーク状態S1 状態/状態UE 状態
M8001	状態	336	PDPC 状態RACH状態MCS 状態
M8004	X2 状態	4	X2 状態eNB 状態
M8005	状態	237	RSSI、SINR、状態AMC
M8006	EPS 状態	54	状態/状態/状態
M8007	状態	14	DRB 状態
M8008	RRC 状態	14	状態
M8009	状態	8	HO 状態
M8010	CQI 状態	27	状態
M8011	状態	55	PRB 状態
M8012	状態	121	PDPC 状態、状態、状態
M8013	状態	21	RRC 状態/状態
M8014	eNB 状態	14	状態 X2 状態
M8015	eNB 状態	13	状態
M8016	CS 状態	18	状態
M8017	状態 HO	10	状態 RAT、3G/2G
M8018	eNB 状態	8	状態 UE 状態

項目	単位	測定範囲	測定対象
M8019	NACC	4	送信電力
M8020	dBm	7	受信電力
M8021	dBm HO	17	ハンドオーバー電力
M8022	X2 単位	2	X2 接続電力
M8023	単位	36	PDCCP SDU 送信電力

測定項目 (M5xxx)

項目	単位	測定範囲	測定対象
M5112	IP 単位	112	IP 接続電力
M5113	単位 RX	21	受信電力

測定項目 (M4xxxx)

項目	単位	測定範囲	測定対象
M40001	単位	単位	VSWR測定RF 単位
M40002	単位	単位	単位

測定項目

測定項目

M<measurement><type>C<counter>

例 M8011C24

- **M** - 測定項目“電圧”
- **8011** - 測定項目番号
- **C** - 測定項目単位“V”
- **24** - 測定項目単位 PRB 単位

例

測定項目番号測定項目単位

- 1 - 電圧
- 2 - 電圧
- 3 - 電圧
- 4 - 電圧
- 5 - 電圧
- 6 - 電圧
- 7 - 電圧

例

例

例

- M8011Cxx - PRB 単位
- M8018Cxx - UE 単位
- M8012Cxx - 単位

例

- M8001Cxx - 単位
- M8005Cxx - RSSI/SINR

□□□□

- 0-1000 0.0% 100.0%
- 10
- 453 = 45.3% PRB

□□□□□□□□

- < 50% - 非常低
- 50-70% - 低
- 70-85% - 中
- > 85% - 高/非常高

Grafana

```
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["metricCounter"] == "M8011C37")
  |> filter(fn: (r) => r._field == "counterValue")
  |> map(fn: (r) => ({ r with _value: r._value / 10.0})) // 00000
```

--	--	--	--	--	--

M8012 - □□□□□□

PDCP

類別	項目	說明	單位	備註	參考
M8012C23	PDCP 吞吐量 測試	測試 PDCP 吞吐量	kbit/s	測試 結果	測試 UE 測試 PDCP SDU 量
M8012C26	PDCP 吞吐量 測試	測試 PDCP 吞吐量	kbit/s	測試 結果	測試 UE 測試 PDCP SDU 量

測試 PDCP 吞吐量

- **PDCP** 吞吐量測試測試
- 測試吞吐量測試kbit/s測試測試
- 測試吞吐量測試測試測試
- 測試測試

測試測試

- 1 測試測試測試測試
- 測試測試測試測試
- 測試 VoLTE 測試測試

測試測試

測試**M8012C26**測試

- < 10 Mbps - 測試/測試
- 10-50 Mbps - 測試
- 50-100 Mbps - 測試/測試測試
- > 100 Mbps - 測試測試測試測試測試

測試**M8012C23**測試

- < 5 Mbps - 測試
- 5-20 Mbps - 測試
- 20-40 Mbps - 測試
- > 40 Mbps - 測試

Grafana 仪表盘

```
from(bucket: "nokia-monitor")
  |> range(start: -1h)
  |> filter(fn: (r) => r["metricCounter"] == "M8012C26")
  |> filter(fn: (r) => r._field == "counterValue")
  |> map(fn: (r) => ({ r with _value: r._value / 1000.0})) // 单位
Mbps
```

仪表盘配置

M8018 - eNB 配置

配置项名称配置项值UE配置

配置项名称	配置项值	配置项值	配置项值	配置项值	配置项值
M8018C1	配置 eNB 配置 UE 配置	配置 eNodeB 配置 UE 配置	配置	配置	1 配置

配置 UE 配置

- 配置 UE = 配置 SRB 配置 DRB 配置
- 配置 1 配置
- 配置

配置

UE		
0-50		
50-100		
100-150		
> 150		

-
- AirScale 150-250 UE
- UE

M8020 -

M8020C3					~10
M8020C4					~10
M8020C6					~10

% = 100.0 × M8020C3 / (M8020C6 - M8020C4)

指標

- **M8020C3** - 平均応答時間
- **M8020C6** - 最大応答時間
- **M8020C4** - 最大応答時間

指標

指標	単位	説明
> 99.9%	%	平均応答時間 SLA
99.0-99.9%	%	最大応答時間
95.0-99.0%	%	最大応答時間
< 95.0%	%	最大応答時間 - 最大応答時間

Grafana 指標

```
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["metricCounter"] == "M8020C3" or
    r["metricCounter"] == "M8020C6" or
    r["metricCounter"] == "M8020C4")
  |> pivot(rowKey:["_time"], columnKey: ["metricCounter"],
valueColumn: "_value")
  |> map(fn: (r) => ({
    _time: r._time,
    "指標": 100.0 * r.M8020C3 / (r.M8020C6 - r.M8020C4)
  })))
```


M8005C0	PUCCH RSSI	RSSI	dBm	
M8005C1	PUCCH RSSI	RSSI	dBm	
M8005C2	PUCCH RSSI	RSSI	dBm	

- RSSI** =
- PUCCH** =
- dBm
- UE RSSI

RSSI		
> -70 dBm		
-70 ~ -85 dBm		
-85 ~ -100 dBm		
-100 ~ -110 dBm		
< -110 dBm		

000

- 0000 - 0 RSSI 00000000
- 0000000 - 000 RSSI 00
- **RF** 00 - 00000000

00000000

M8013 - 0000000

0000000 RRC00000000000000000000000000000000

000	00	00	00	00
M8013C5	00000000	000 RRC 0000	00	0
M8013C17	00000000 MO-S	0000 - 0000000	00	0
M8013C18	00000000 MT	0000 - 0000	00	0
M8013C19	00000000 MO-D	0000 - 0000000	00	0
M8013C21	0000000000	00000000	00	0
M8013C31	000000000000	00000000	00	0
M8013C34	000000000000	00000000	00	0
M8013C91	00000000 MO-V	0000 - 0000000	00	0
M8013C93	00000000 MT-00	0000 - MT 00	00	0

00000000

$$\square\square\square\square \% = 100.0 \times M8013C5 / (M8013C17 + M8013C18 + M8013C19 + M8013C34 + M8013C31 + M8013C21 + M8013C93 + M8013C91)$$

5/5

- **M8013C5** - 5년 RRC 5년
- 5년 - 5년

5/5

- **MO-S (M8013C17)** - 1000000000000000
- **MT (M8013C18)** - 0000000/000
- **MO-D (M8013C19)** - 00000000000000
- **MO-V (M8013C91)** - 0000000VoLTE 000
- **00 (M8013C21)** - 0000091101120

□□□□□

□□□	□□	□□
> 99%	□□	□□□□
95-99%	□□	□□□□□□
90-95%	□□	□□□□
< 90%	□□	□□□□ - □□□□□□

□ □ □ □ □ □ □

- 
- 
- 
- 
- 

QoS

M8001 -

M8001C2	PDCP SDU	eNB PDCP SDU	ms	

- **PDCP SDU** =
- =
- **DL DTCH** =
- =

< 10 ms		VoLTE
10-20 ms		
20-50 ms		
> 50 ms		

-
-
-
-

S1 介面

M8000 - S1 介面

本圖表顯示 eNodeB 與 MME 之間的 S1 介面功能。S1 介面功能如下：

功能	描述	功能	描述	功能
M8000C0	eNodeB 向 MME 註冊	MME 向 eNodeB 註冊	eNodeB 向 MME 註冊	註冊
M8000C1	eNodeB 向 MME 註冊	MME 向 eNodeB 註冊	eNodeB 向 MME 註冊	註冊
M8000C2	eNodeB 向 MME 註冊	MME 向 eNodeB 註冊	eNodeB 向 MME 註冊	註冊
M8000C3	eNodeB 向 MME 註冊	MME 向 eNodeB 註冊	eNodeB 向 MME 註冊	註冊
M8000C6	S1 介面	S1 介面	S1 介面	註冊
M8000C7	S1 介面	S1 介面	S1 介面	註冊
M8000C11	S1 介面	S1 介面	S1 介面	註冊
M8000C12	UE 向 S1 介面	UE 向 S1 介面	UE 向 S1 介面	註冊
M8000C29	NAS 向 S1 介面	NAS 向 S1 介面	NAS 向 S1 介面	註冊
M8000C30	NAS 向 S1 介面	NAS 向 S1 介面	NAS 向 S1 介面	註冊

S1 介面

- **S1** 介面 eNodeB 與 EPC 之間的介面
- 本圖表顯示 eNodeB 與 MME 之間的 S1 介面功能
- **S1** 介面 eNodeB 與 MME 之間的介面
- **NAS** 介面 eNodeB 與 MME 之間的介面

本圖表

$$S1 \text{ } = 100 \times M8000C7 / M8000C6$$

$$\text{ } = 100 \times M8000C1 / M8000C0$$

- S1 > 99%
- > 95%

EPS

M8006 - EPS

E-UTRAN E-RAB EPS

M8006C0	EPS			
M8006C1	EPS			
M8006C2-C5				

EPS

- = UE
- =
- = QoS VoLTE

-
-

- QoS

M8009 -

M8014 - eNB

M8015 - eNB

M8021 -

- UE

eNB		M8015
eNB	X2	M8014
		M8021
	LTE→3G/2G	M8017

Modell	Typ	Spezifikation
M8009	Standard	Standard-Modell
M8014	2x X2 HO	Standard-Modell
M8015	HO	Standard-Modell
M8021	HO	Standard-Modell

Modell

$$\text{Modell} = 100 \times (\text{Modell}) / (\text{Modell})$$

Modell

- eNB HO Modell > 99%
- eNB HO Modell > 98%
- HO Modell > 95%

Modell

- Modell
- Modell
- Modell UE Modell
- Modell

Modell

M8010 - CQI

Modell UE CQI Modell

Modell CQI

- **CQI** = UE → eNodeB 報告値
- 範囲 CQI 0~15
- 範囲 報告値 MCS 値
- 範囲 報告値

CQI 範囲

CQI 範囲	範囲	範囲	範囲
0-3	範囲	< 1 Mbps	QPSK
4-6	範囲	1-5 Mbps	QPSK
7-9	範囲	5-15 Mbps	16-QAM
10-12	範囲	15-40 Mbps	64-QAM
13-15	範囲	40-150 Mbps	64-QAM

M8010 範囲

- M8010C0 - M8010C15 範囲 0-15 CQI 範囲

範囲

$$\text{CQI} = \frac{\sum(\text{CQI_level} \times \text{M8010C}[\text{level}])}{\sum(\text{M8010C}[\text{level}])}$$

範囲 CQI 範囲

- 範囲 CQI 10-15 範囲
- 範囲 CQI 7-9 範囲
- 範囲 CQI 0-6 範囲

CS 測試項目

M8016 - CS 測試

測試項目測試項目CSFB測試項目 LTE UE 測試項目 2G/3G 測試

CSFB

- 測試項目 VoLTE 測試項目
- 測試 UE 測試 LTE → 測試項目 → 測試項目 2G/3G → 測試 LTE
- 測試項目 測試項目 LTE 測試

測試項目

測試項目	測試項目	測試項目
M8016C0	測試項目 CSFB 測試	測試項目 CSFB
M8016C1	測試項目 CSFB 測試	測試項目 CSFB

CSFB 測試

- 測試 UE 測試 LTE 測試項目 2G/3G測試項目
- 測試 測試 LTE 測試 2G/3G 測試項目

測試項目

- CSFB 測試項目 = 測試項目 / 測試 CSFB 測試
- 測試項目 > 98%

測試項目

M8023 - PDCP SDU 測試

測試項目測試項目測試項目測試項目

無線技術

- 無線通信技術の進化
- 4G/LTEから5Gへの移行
- 無線通信の応用

無線技術

- 無線通信技術の進化
- 4G/LTEから5Gへの移行
- 無線通信の応用
- QoS 保証

無線技術 M8012 無線通信技術

- M8012 無線通信技術 kbit/s
- M8023 無線通信技術

X2 無線通信

M8004 - X2 無線通信

M8022 - X2 無線通信

無線通信技術 eNodeB から X2 無線通信 eNB 無線通信技術

M8004 - X2 無線通信

- 無線通信技術 eNodeB 無線通信技術
- 無線通信 X2 無線通信

M8022 - X2 無線通信

- M8022C0 X2 無線通信
- M8022C1 X2 無線通信

M8019 - 無線通信規格NACC

無線 NACC 規格は LTE と GSM 規格を統合

無線 規格は UE と GSM 規格を統合

無線規格

M5112 - IP 無線規格

M5113 - 無線 RX 規格

無線規格

M5112 - IP 無線規格112 無線規格

- 無線規格
- 無線規格
- 無線規格
- 無線規格

M5113 - 無線 RX規格21 無線規格

- 無線規格
- 無線規格
- 無線規格CRC 無線規格

無線

- 無線規格
 - 無線規格
 - 無線規格
 - 無線規格
-

□□□□□□□□□□

M40001 - □□□□□□

□◀▶□	□□	□□	□□	□□	□□
M40001C0	□□□□□□ VSWR	□□□□□	0.1	□□	□□ 10

□□ VSWR□

- VSWR = □□□□□
- □□□□□□□□
- □□□□□□□□□□□□/□□□□
- □□□□ = □□

VSWR □□□

VSWR	□□	□□
1.0-1.5	□□	□□□□
1.5-2.0	□□	□□□
2.0-3.0	□□	□□
> 3.0	□□	□□/□□□□ - □□□□□□

□ VSWR □□□□□□

- □□□□□□□□
- □□□□□□□
- □□□□
- □□□□
- □□□□□

Grafana □□□□□

```
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["metricCounter"] == "M40001C0")
  |> filter(fn: (r) => r._field == "counterValue")
  |> map(fn: (r) => ({ r with "VSWR": r._value / 10.0}))
```

M40002 - 〇〇〇〇

〇〇〇	〇〇	〇〇	〇〇	〇〇	〇〇
M40002C2	〇〇	〇〇〇〇	100000 〇〇	〇〇	〇〇 100000

〇〇〇〇〇

- 〇〇〇〇〇〇〇〇〇
- 〇 OPEX 〇〇~~〇〇〇~~〇〇〇〇〇〇〇
- 〇〇〇〇〇/〇〇〇〇〇〇〇〇〇〇〇

Grafana 〇〇〇〇〇

```
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["metricCounter"] == "M40002C2")
  |> filter(fn: (r) => r._field == "counterValue")
  |> map(fn: (r) => ({ r with "Power": r._value / 100000.0}))
```

〇 Grafana 〇〇〇〇〇〇〇

〇〇〇〇〇〇〇〇〇

1. 〇〇〇〇〇〇〇〇〇

〇〇 M8011C24 〇 M8011C37 〇〇〇〇/〇〇 PRB 〇〇〇

```
// 00 PRB 00
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["metricCounter"] == "M8011C24")
  |> map(fn: (r) => ({ r with _value: r._value / 10.0}))
  |> rename(columns: {"_value": "00 PRB %"})

// 00 PRB 00
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["metricCounter"] == "M8011C37")
  |> map(fn: (r) => ({ r with _value: r._value / 10.0}))
  |> rename(columns: {"_value": "00 PRB %"})
```

2. 000000

0000000000

```
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["metricCounter"] == "M8012C23" or
r["metricCounter"] == "M8012C26")
  |> map(fn: (r) => ({ r with _value: r._value / 1000.0})) // 000
Mbps
  |> pivot(rowKey: ["_time"], columnKey: ["metricCounter"],
valueColumn: "_value")
  |> map(fn: (r) => ({
    _time: r._time,
    "00 Mbps": r.M8012C23,
    "00 Mbps": r.M8012C26
  })))
```

3. 000000

000000000000

```

from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["metricCounter"] =~ /M8020C(3|4|6)/)
  |> pivot(rowKey: ["_time"], columnKey: ["metricCounter"],
valueColumn: "_value")
  |> map(fn: (r) => ({
    _time: r._time,
    "RSSI %": 100.0 * r.M8020C3 / (r.M8020C6 - r.M8020C4)
  })))

```

4. 4G LTE

4G RRC 4G

```

from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["metricCounter"] =~
/M8013C(5|17|18|19|21|31|34|91|93)/)
  |> pivot(rowKey: ["_time"], columnKey: ["metricCounter"],
valueColumn: "_value")
  |> map(fn: (r) => ({
    _time: r._time,
    "4G %": 100.0 * r.M8013C5 / (r.M8013C17 + r.M8013C18 +
r.M8013C19 + r.M8013C34 + r.M8013C31 + r.M8013C21 + r.M8013C93 +
r.M8013C91)
  })))

```

4G 4G

4G 4G

- 4G - 4G PRB
- 4G - 4G
- 4G - 4G UE
- 4G - 4G RSSI

4G 4G

- 4G

- 網路連線品質
- 網路連線穩定性

網路品質

- PRB 使用率 $< 70\%$ 70-85% $> 85\%$
- 網路品質 $> 99\%$ 95-99% $< 95\%$
- 網路品質 $> 99\%$ 95-99% $< 95\%$

網路品質

- 網路品質監控/網路品質
- 網路品質監控/網路品質
- 網路品質監控/網路品質

網路品質

- 網路品質 - RAN Monitor 網路品質
- Grafana 網路品質 - Grafana 網路品質
- 網路品質 - RAN Monitor
- AirScale 網路品質 - 網路品質
- 網路品質 - InfluxDB 網路品質

□□□□□

□□□□□□□

□□□	□□	□□	□□
M8011C24	□□ PRB □    □□	□□□□	0.1%
M8011C37	□□ PRB □□□□	□□□□	0.1%
M8012C23	PDCP □□□□□	□□□□	kbit/s
M8012C26	PDCP □□□□□	□□□□	kbit/s
M8018C1	□□□□ UE	□□□□	□□
M8020C3	□□□□□□	□□□□□	□□
M8020C6	□□□□□	□□□□□	□□
M8013C5	□□□□	□□□□□	□□
M8005C0	RSSI □□□	□□□□	dBm
M8005C2	RSSI □□□	□□□□	dBm
M8001C2	PDCP □□□□	□□□□	ms
M40001C0	VSWR	□□□□	0.1 □□

□□□

- □□□ FlexiRadio LTE □□□□
- □□□ AirScale □□□□□□□
- 3GPP LTE □□□□□□

□□□□□□□□

□□□□□□ KPI □□□□□□□ 1,186 □□□□ LTE □□□□□□□□□□

=====
LTE
=====

M8000

M8000C0					
M8000C1					
M8000C2					
M8000C3					
M8000C4					
M8000C5					
M8000C6		S1			
M8000C7		S1			
M8000C8					
M8000C9		MME			
M8000C11		S1			
M8000C12		UE			
M8000C13		eNB			
M8000C14		MME			
M8000C15		eNB			
M8000C16		MME			
M8000C23		UE			
M8000C24					
M8000C25		UE			
M8000C29		NAS			
M8000C30		NAS			
M8000C31					
M8000C32		IMS			
M8000C33		IMS			
M8000C34		IMS			
M8000C35					
M8000C36					
M8000C37		X2 IP			
M8000C38		X2 IP			
M8000C39		WRITE-REPLACE			
M8000C40		WRITE-REPLACE			
M8000C41		KILL			
M8000C42		KILL			

M8001

M8001C0		□□ PDCP SDU □□□□	
M8001C1		□□ PDCP SDU □□□□	
M8001C2		□□ PDCP SDU □□□□	□□
M8001C3		□□ PDCP SDU □□□□	
M8001C4		□□ PDCP SDU □□□□	
M8001C5		□□ PDCP SDU □□□□	□□
M8001C6		□□ RACH □□□□	□
M8001C7		□□ RACH □□□□	□
M8001C8		RACH □□□□	□
M8001C9		□ PCH □□□□ TB □□	
M8001C10		□ BCH □□□□ TB □□	
M8001C11		□ DL-SCH □□□□ TB □□	
M8001C12		□ DL-SCH □□ HARQ □□	
M8001C13		□□□□□□ UL-SCH TB □□	□
M8001C14		□□□ UL-SCH TB □□□□	□
M8001C15		□□□ UL-SCH TB □□□□	
M8001C16		□□ MCS0 □ PUSCH □□	
M8001C17		□□ MCS1 □ PUSCH □□	
M8001C18		□□ MCS2 □ PUSCH □□	
M8001C19		□□ MCS3 □ PUSCH □□	
M8001C20		□□ MCS4 □ PUSCH □□	
M8001C21		□□ MCS5 □ PUSCH □□	
M8001C22		□□ MCS6 □ PUSCH □□	
M8001C23		□□ MCS7 □ PUSCH □□	
M8001C24		□□ MCS8 □ PUSCH □□	
M8001C25		□□ MCS9 □ PUSCH □□	
M8001C26		□□ MCS10 □ PUSCH □□	
M8001C27		□□ MCS11 □ PUSCH □□	
M8001C28		□□ MCS12 □ PUSCH □□	
M8001C29		□□ MCS13 □ PUSCH □□	
M8001C30		□□ MCS14 □ PUSCH □□	
M8001C31		□□ MCS15 □ PUSCH □□	
M8001C32		□□ MCS16 □ PUSCH □□	
M8001C33		□□ MCS17 □ PUSCH □□	
M8001C34		□□ MCS18 □ PUSCH □□	
M8001C35		□□ MCS19 □ PUSCH □□	
M8001C36		□□ MCS20 □ PUSCH □□	
M8001C37		□□ MCS21 □ PUSCH □□	
M8001C38		□□ MCS22 □ PUSCH □□	
M8001C39		□□ MCS23 □ PUSCH □□	
M8001C40		□□ MCS24 □ PUSCH □□	
M8001C41		□□ MCS25 □ PUSCH □□	
M8001C42		□□ MCS26 □ PUSCH □□	

M8001C43		□□	MCS27	□	PUSCH	□□	
M8001C44		□□	MCS28	□	PUSCH	□□	
M8001C45		□□	MCS29	□	PUSCH	□□	
M8001C46		□□	MCS30	□	PUSCH	□□	
M8001C47		□□	MCS31	□	PUSCH	□□	
M8001C48		□□	MCS32	□	PUSCH	□□	
M8001C49		□□	MCS33	□	PUSCH	□□	
M8001C50		□□	MCS34	□	PUSCH	□□	
M8001C51		□□	MCS35	□	PUSCH	□□	
M8001C52		□□	MCS36	□	PUSCH	□□	
M8001C53		□□	MCS37	□	PUSCH	□□	
M8001C54		□□	MCS38	□	PUSCH	□□	
M8001C55		□□	MCS39	□	PUSCH	□□	
M8001C56		□□	MCS40	□	PUSCH	□□	
M8001C57		□□	MCS41	□	PUSCH	□□	
M8001C58		□□	MCS42	□	PUSCH	□□	
M8001C59		□□	MCS43	□	PUSCH	□□	
M8001C60		□□	MCS44	□	PUSCH	□□	
M8001C61		□□	MCS45	□	PUSCH	□□	
M8001C62		□□	MCS46	□	PUSCH	□□	
M8001C63		□□	MCS47	□	PUSCH	□□	
M8001C64		□□	MCS48	□	PUSCH	□□	
M8001C65		□□	MCS49	□	PUSCH	□□	
M8001C66		□□	MCS50	□	PUSCH	□□	
M8001C67		□□	MCS51	□	PUSCH	□□	
M8001C68		□□	MCS52	□	PUSCH	□□	
M8001C69		□□	MCS53	□	PUSCH	□□	
M8001C70		□□	MCS54	□	PUSCH	□□	
M8001C71		□□	MCS55	□	PUSCH	□□	
M8001C72		□□	MCS56	□	PUSCH	□□	
M8001C73		□□	MCS57	□	PUSCH	□□	
M8001C74		□□	MCS58	□	PUSCH	□□	
M8001C75		□□	MCS59	□	PUSCH	□□	
M8001C76		□□	MCS60	□	PUSCH	□□	
M8001C77		□□	MCS61	□	PUSCH	□□	
M8001C78		□□	MCS62	□	PUSCH	□□	
M8001C79		□□	MCS63	□	PUSCH	□□	
M8001C80		□□	MCS64	□	PUSCH	□□	
M8001C81		□□	MCS65	□	PUSCH	□□	
M8001C82		□□	MCS66	□	PUSCH	□□	
M8001C83		□□	MCS67	□	PUSCH	□□	
M8001C84		□□	MCS68	□	PUSCH	□□	
M8001C85		□□	MCS69	□	PUSCH	□□	
M8001C86		□□	MCS70	□	PUSCH	□□	

M8001C87		□□	MCS71	□	PUSCH	□□	
M8001C88		□□	MCS72	□	PUSCH	□□	
M8001C89		□□	MCS73	□	PUSCH	□□	
M8001C90		□□	MCS74	□	PUSCH	□□	
M8001C91		□□	MCS75	□	PUSCH	□□	
M8001C92		□□	MCS76	□	PUSCH	□□	
M8001C93		□□	MCS77	□	PUSCH	□□	
M8001C94		□□	MCS78	□	PUSCH	□□	
M8001C95		□□	MCS79	□	PUSCH	□□	
M8001C96		□□	MCS80	□	PUSCH	□□	
M8001C97		□□	MCS81	□	PUSCH	□□	
M8001C98		□□	MCS82	□	PUSCH	□□	
M8001C99		□□	MCS83	□	PUSCH	□□	
M8001C100		□□	MCS84	□	PUSCH	□□	
M8001C101		□□	MCS85	□	PUSCH	□□	
M8001C102		□□	MCS86	□	PUSCH	□□	
M8001C103		□□	MCS87	□	PUSCH	□□	
M8001C104		□□	MCS88	□	PUSCH	□□	
M8001C105		□□	MCS89	□	PUSCH	□□	
M8001C106		□□	MCS90	□	PUSCH	□□	
M8001C107		□□	MCS91	□	PUSCH	□□	
M8001C108		□□	MCS92	□	PUSCH	□□	
M8001C109		□□	MCS93	□	PUSCH	□□	
M8001C110		□□	MCS94	□	PUSCH	□□	
M8001C111		□□	MCS95	□	PUSCH	□□	
M8001C112		□□	MCS96	□	PUSCH	□□	
M8001C113		□□	MCS97	□	PUSCH	□□	
M8001C114		□□	MCS98	□	PUSCH	□□	
M8001C115		□□	MCS99	□	PUSCH	□□	
M8001C116		□□	MCS100	□	PUSCH	□□	
M8001C117		□□	MCS101	□	PUSCH	□□	
M8001C118		□□	MCS102	□	PUSCH	□□	
M8001C119		□□	MCS103	□	PUSCH	□□	
M8001C120		□□	MCS104	□	PUSCH	□□	
M8001C121		□□	MCS105	□	PUSCH	□□	
M8001C122		□□	MCS106	□	PUSCH	□□	
M8001C123		□□	MCS107	□	PUSCH	□□	
M8001C124		□□	MCS108	□	PUSCH	□□	
M8001C125		□□	MCS109	□	PUSCH	□□	
M8001C126		□□	MCS110	□	PUSCH	□□	
M8001C127		□□	MCS111	□	PUSCH	□□	
M8001C128		□□	MCS112	□	PUSCH	□□	
M8001C129		□□	MCS113	□	PUSCH	□□	
M8001C130		□□	MCS114	□	PUSCH	□□	

M8001C131		□□	MCS115	□	PUSCH	□□	
M8001C132		□□	MCS116	□	PUSCH	□□	
M8001C133		□□	MCS117	□	PUSCH	□□	
M8001C134		□□	MCS118	□	PUSCH	□□	
M8001C135		□□	MCS119	□	PUSCH	□□	
M8001C136		□□	MCS120	□	PUSCH	□□	
M8001C137		□□	MCS121	□	PUSCH	□□	
M8001C138		□□	MCS122	□	PUSCH	□□	
M8001C139		□□	MCS123	□	PUSCH	□□	
M8001C140		□□	MCS124	□	PUSCH	□□	
M8001C141		□□	MCS125	□	PUSCH	□□	
M8001C142		□□	MCS126	□	PUSCH	□□	
M8001C143		□□	MCS127	□	PUSCH	□□	
M8001C144		□□	MCS128	□	PUSCH	□□	
M8001C145		□□	MCS129	□	PUSCH	□□	
M8001C146		□□	MCS130	□	PUSCH	□□	
M8001C147		□□	MCS131	□	PUSCH	□□	
M8001C148		□□	MCS132	□	PUSCH	□□	
M8001C149		□□	MCS133	□	PUSCH	□□	
M8001C150		□□	MCS134	□	PUSCH	□□	
M8001C151		□□	MCS135	□	PUSCH	□□	
M8001C152		□□	MCS136	□	PUSCH	□□	
M8001C153		□□	MCS137	□	PUSCH	□□	
M8001C154		□□	MCS138	□	PUSCH	□□	
M8001C155		□□	MCS139	□	PUSCH	□□	
M8001C156		□□	MCS140	□	PUSCH	□□	
M8001C157		□□	MCS141	□	PUSCH	□□	
M8001C158		□□	MCS142	□	PUSCH	□□	
M8001C159		□□	MCS143	□	PUSCH	□□	
M8001C160		□□	MCS144	□	PUSCH	□□	
M8001C161		□□	MCS145	□	PUSCH	□□	
M8001C162		□□	MCS146	□	PUSCH	□□	
M8001C163		□□	MCS147	□	PUSCH	□□	
M8001C164		□□	MCS148	□	PUSCH	□□	
M8001C165		□□	MCS149	□	PUSCH	□□	
M8001C166		□□	MCS150	□	PUSCH	□□	
M8001C167		□□	MCS151	□	PUSCH	□□	
M8001C168		□□	MCS152	□	PUSCH	□□	
M8001C169		□□	MCS153	□	PUSCH	□□	
M8001C170		□□	MCS154	□	PUSCH	□□	
M8001C171		□□	MCS155	□	PUSCH	□□	
M8001C172		□□	MCS156	□	PUSCH	□□	
M8001C173		□□	MCS157	□	PUSCH	□□	
M8001C174		□□	MCS158	□	PUSCH	□□	

M8001C175		□□	MCS159	□	PUSCH	□□	
M8001C176		□□	MCS160	□	PUSCH	□□	
M8001C177		□□	MCS161	□	PUSCH	□□	
M8001C178		□□	MCS162	□	PUSCH	□□	
M8001C179		□□	MCS163	□	PUSCH	□□	
M8001C180		□□	MCS164	□	PUSCH	□□	
M8001C181		□□	MCS165	□	PUSCH	□□	
M8001C182		□□	MCS166	□	PUSCH	□□	
M8001C183		□□	MCS167	□	PUSCH	□□	
M8001C184		□□	MCS168	□	PUSCH	□□	
M8001C185		□□	MCS169	□	PUSCH	□□	
M8001C186		□□	MCS170	□	PUSCH	□□	
M8001C187		□□	MCS171	□	PUSCH	□□	
M8001C188		□□	MCS172	□	PUSCH	□□	
M8001C189		□□	MCS173	□	PUSCH	□□	
M8001C190		□□	MCS174	□	PUSCH	□□	
M8001C191		□□	MCS175	□	PUSCH	□□	
M8001C192		□□	MCS176	□	PUSCH	□□	
M8001C193		□□	MCS177	□	PUSCH	□□	
M8001C194		□□	MCS178	□	PUSCH	□□	
M8001C195		□□	MCS179	□	PUSCH	□□	
M8001C196		□□	MCS180	□	PUSCH	□□	
M8001C199		RRC	□□□	UE	□□□		

11

□ □

--	--	--	--

□ □ □ □

項目	概要
開発者 PM の役割	データベース InfluxDB の構築
開発者	データベース InfluxDB 22,000+ の構築

--	--	--	--	--

PM

種別	識別子	数値	説明
LTE	M8xxx	~5,900	LTE L1/L2/L3 間のERABとRRC間の
WCDMA	M5xxx	~885	3G WCDMA 間のMAC とCQIとHSDPA
5G-NR	M55xxx	~14,500	5G NR 間のMIMO間の
5G-UE	M51xxx	~500	5G UE間の
5G-UE	M40xxx	~250	5G UE間の

データ

データは `priv/pm_counters.csv` に保存されています。

- 識別子
- 数値
- 説明
- RRC 間の/
- **PRB**間の
- 間の/間の
- **RRC**とRRC 間の
- **ERAB**とE-RAB 間の
- **PDCH**とPDCH 間の
- 間の
- 間の

環境構築

前提条件

1. “Docker”環境が構築されていること
2. Docker Desktopがインストールされていること
3. Docker Desktopが起動していること
4. Docker Desktopのデフォルト設定が適用されていること

インストール

1. “Docker PM”をインストールする
2. Docker PMの起動を確認する

確認

確認方法

- Docker PMのIDを確認する
- Docker PMのバージョンを確認する (LTE/5G-NR)

設定

設定ファイルは `priv/pm_counters.csv` に保存されている

実行

“Docker PM”を実行する

1. 設定ファイルは `priv/pm_filters.etf`
2. Docker PMの起動を確認する
3. Docker PMのバージョンを確認する

InfluxDB にデータを保存する/取得する

概要

サービス概要

料金 22,000+ 円/年

項目	内容
容量	約 100-500 GB/年
データベース	InfluxDB
接続方法	API
セキュリティ	SSL/TLS

特徴

- 高可用性
- 柔軟な拡張性
- コスト効率

お問い合わせ

お問い合わせ先

“お問い合わせ”ボタンをクリックしてください。

- ID: M8012C23
- RSRP

PM Counter

PM Counter	Radio	Measurement
M8012C23	LTE	PM Counter
M8012C26	LTE	PM Counter
M8001C2	LTE	PDCP SDU Counter
M8011C24	LTE	PRB Counter
M8011C37	LTE	PRB Counter
M8013C17	LTE	RRC Counter
M8020C3	LTE	Counter
M40001C0	5G	Counter

PM Counter

pm_counters.csv

PM Counter

```
# PM Counter, radio, counter
M8012C23, radio, PM Counter
M8012C26, radio, PM Counter
M8001C2, radio, PDCP SDU Counter
...
```

PM Counter `priv/pm_counters.csv`

pm_metrics.csv

PM Metrics Data

```
# PM_Code,Type,Status
M8000C6,LTE,S1_SETUP_ATT
M8000C7,LTE,S1_SETUP_SUCC
...
```

File: `priv/pm_metrics.csv`

Overview

PM Metrics

- PM Metrics are categorized by PM Code and Type.
- eNodeB PM Metrics are stored in InfluxDB.
- PM Metrics ID is used for tracking.

Configuration

- PM Metrics are collected from eNodeB.
- eNodeB PM Metrics are collected every 15 seconds.
- PM Metrics are stored in `[PmFilterStore]`.
- PM Metrics are processed by the PM Filter.

Usage

- PM Metrics are stored in `priv/pm_metrics.csv`.
 - PM Metrics are processed by the PM Filter.
 - PM Metrics are stored in UTF-8 format.
-

□□□□

- □□□□□ - PM □□□□□□
 - Grafana □□ - □□ PM □□□□□□
 - InfluxDB □□ - □□ PM □□
-

□□□

- **PM** □□□□ <https://localhost:9443/nokia/pm-filters>
- □□□□ <https://localhost:9443/nokia/retention>
- **InfluxDB** □□□ <https://localhost:9443/nokia/influx>

RAN 部署环境搭建

配置 `config/runtime.exs`

配置

1. 配置
 2. 数据库
 3. Web 配置
 4. 部署环境
 5. 部署环境
 6. InfluxDB 配置
 7. 部署环境
-

配置

`config/runtime.exs` 配置 RAN 部署环境搭建

配置

- 部署环境MySQL
- Web 部署环境
- 部署环境
- InfluxDB 部署环境
- 部署环境
- 部署环境

配置

`config/runtime.exs`

環境構築

まず RAN モニタリング **Omnitouch** をインストールします。

- インストール
- データベースの作成
- サービスの起動
- ログの確認

次に **Omnitouch** の監視対象として **Omnitouch** を設定します。

監視対象として **Grafana** を設定します。

確認

MySQL/MariaDB

```
config :ran_monitor, RanMonitor.Repo,  
  username: "omnitouch",  
  password: "omnitouch2024",  
  hostname: "localhost",  
  database: "ran_monitor",  
  stacktrace: true,  
  show_sensitive_data_on_connection_error: true,  
  pool_size: 10
```

MySQL のインストールと設定

設定

username (ユーザ名)

- ユーザ名
- パスワード "omnitouch"
- 権限 CREATE, SELECT, INSERT, UPDATE, DELETE

- 密碼 密碼密碼密碼密碼密碼密碼密碼

password (密碼)

- 密碼密碼密碼密碼
- 密碼 "omnitouch2024"
- 密碼 密碼密碼密碼密碼密碼密碼
- 密碼 密碼密碼密碼密碼

hostname (主機)

- 密碼密碼密碼
- 密碼 "localhost"
- 密碼
 - "localhost" - 密碼密碼密碼???
 - "127.0.0.1" - TCP 密碼密碼
 - "10.179.2.135" - 密碼密碼 IP
 - "db.example.com" - 密碼密碼

database (資料庫)

- 密碼密碼密碼
- 密碼 "ran_monitor"
- 密碼 密碼密碼 RAN 密碼密碼
- 密碼 CREATE DATABASE ran_monitor CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

stacktrace (堆疊)

- 密碼密碼密碼密碼
- 密碼 true
- 密碼 true - 密碼
- 密碼 false - 密碼

show_sensitive_data_on_connection_error (顯示)

- 密碼密碼密碼密碼

- `pool_size` `true`
- `pool_size` `true` - `pool_size` `10`
- `pool_size` `false` - `pool_size` `20`

pool_size (pool)

- `pool_size` `10`
- `pool_size` `20`
 - 1-5 `pool_size` `5`
 - 6-20 `pool_size` `10`
 - 21-50 `pool_size` `15`
 - 50+ `pool_size` `20`
- `pool_size` `20` + 5 `pool_size` Web UI

Web

RAN `pool_size` Web `pool_size`

SOAP/API

```
config :ran_monitor, RanMonitor.Web.Endpoint,
  http: [ip: {0, 0, 0, 0}, port: 8080],
  check_origin: false,
  secret_key_base:
    "v5t0S1/QRonjw0ky7adGGfkBbrJmiJyXhpesJy/jvSZhqLZkREV+rlo1/pR8lkbu",
  server: true
```

`pool_size` `20` NE3S `pool_size` SOAP `pool_size`

ip (pool)

- `pool_size` `10`
- `pool_size` `{0, 0, 0, 0}` `pool_size` `20`
- `pool_size` `20`

- `{0, 0, 0, 0}` - 0.0.0.0
- `{127, 0, 0, 1}` - localhost
- `{10, 179, 2, 135}` - 10.179.2.135 IP address

port (int)

- TCP port
- default `8080`
- port number must be between 1 and 65535
- port number must be unique IP

check_origin (bool)

- check WebSocket/HTTP origin
- default `false`
- if you use SOAP API, set `false` to use Web UI

secret_key_base (string)

- secret key
- length 64 characters
- example `mix phx.gen.secret`
- secret key must be unique
- secret key must be 64 characters

server (bool)

- server mode
- default `true`
- if you use runtime.exs, set `true`

Web UI

```
# 設定 HTTPS ポート 9443
https_port =
String.to_integer(System.get_env("CONTROL_PANEL_HTTPS_PORT") ||
"9443")

config :control_panel, ControlPanelWeb.Endpoint,
  url: [host: "0.0.0.0", port: https_port, scheme: "https"],
  https: [
    ip: {0, 0, 0, 0},
    port: https_port,
    keyfile: "priv/cert/omnitouch.pem",
    certfile: "priv/cert/omnitouch.crt"
  ]
```

Web UI を HTTPS 化

手順

- **CONTROL_PANEL_HTTPS_PORT** - HTTPS ポート 9443
 - 環境変数として HTTPS ポート
 - `export CONTROL_PANEL_HTTPS_PORT=8443`

url (設定)

- URL 設定
- **host:** `"0.0.0.0"` - ホスト名
- **port:** `https_port` `CONTROL_PANEL_HTTPS_PORT` の値
- **scheme:** `"https"` - HTTPS かどうか

https (設定)

- HTTPS 設定
- **ip:** `{0, 0, 0, 0}` - IP アドレス
- **port:** `https_port` url のポート
- **keyfile:** SSL キーファイル
- **certfile:** SSL 証明書ファイル

SSL 証明書

- 証明書 SSL/TLS 用
- 証明書生成
- 証明書 CA 証明書
- 証明書

```
openssl req -newkey rsa:2048 -nodes -keyout omnitouch.pem -x509
-days 365 -out omnitouch.crt
```

AirScale Webhook

```
config :ran_monitor, RanMonitor.Web.Nokia.Airscale.Endpoint,
  url: [host: "0.0.0.0"],
  http: [ip: {0, 0, 0, 0}, port: 9076],
  server: true
```

AirScale 証明書

port ()

- 9076
- PMCADM rTpmCollEntityPortNum
- WebLM

証明書

```
config :logger,
  level: :info

config :logger, :console,
  format: "$time $metadata[$level] $message\n",
  metadata: [:request_id]
```

level

level (level)

- 消息级别
- 消息:info
- 消息
 - :debug - 调试信息
 - :info - 一般信息
 - :warning - 警告
 - :error - 错误

format

- 消息:debug - 调试信息
- 消息:info - 一般信息
- 消息:debug 消息
- 消息:warning - 警告

format

format (format)

- 消息格式
- 消息"\$time \$metadata[\$level] \$message\n"
- 消息
 - \$time - 时间
 - \$metadata - 元数据
 - \$level - 消息级别info,error
 - \$message - 消息

metadata (metadata)

- 消息格式
- 消息[:request_id]
- request_id: 消息的 HTTP 请求 ID

□□□□□

□□□□ RAN □□□□□□□□□□□□

```
config :ran_monitor,
  general: %{
    mcc: "505",
    mnc: "57"
  },
  nokia: %{
    ne3s: %{
      webhook_url: "http://10.5.198.200:9076/webhook",
      private_key: Path.join(Application.app_dir(:ran_monitor,
"priv"), "external/nokia/ne.key.pem"),
      public_key: Path.join(Application.app_dir(:ran_monitor,
"priv"), "external/nokia/ne.cert.der"),
      reregister_interval: 30
    },
    airscales: [
      %{
        address: "10.7.15.67",
        name: "ONS-Lab-Airscale",
        port: "8080",
        web_username: "Nemuadmin",
        web_password: "nemuuser"
      }
    ]
  }
}
```

□□□□

mcc (□□□)

- □□□□□
- □□□□ "505"
- □□□ □□ 3GPP □□□□□
- □□□ 3 □□□

- 参考 ITU-T E.212

mnc (整数)

- 整数値
- 範囲 "57"
- 10進法で指定
- 範囲 2 ~ 3 桁

NE3S 設定 NE3S 設定

webhook_url (文字列)

- 指定する URL
- 範囲 "http://10.5.198.200:9076/webhook"
- 範囲 http://<ran-monitor-ip>:<port>/webhook
- **IP** 範囲 指定 RAN 範囲 IP 範囲
- 範囲 範囲 RanMonitor.Web.Nokia.Airscale.Endpoint 範囲9076範囲
- 範囲 範囲 /webhook

private_key (文字列 - 文字列)

- 指定する文字列
- 範囲 priv/external/nokia/ne.key.pem
- 範囲 PEM 範囲
- 範囲 範囲
- 範囲 範囲 OpenSSL 範囲

public_key (文字列 - 文字列)

- 指定する文字列
- 範囲 priv/external/nokia/ne.cert.der
- 範囲 DER 範囲
- 範囲 範囲
- 範囲 範囲 private_key 範囲

reregister_interval (整数)

- □□□□□□□□
- □□□□**30**
- □□□ □□□□□□□□□□□□□□□□
- □□□ 30-300 □
- □□□ 30 □□□□□□□□

AirScale

airscales (airscales)

- 空気圧 AirScale 空気圧
- 空気圧センサー

--	--	--	--	--	--	--	--	--

address ()

- 配置 IP 地址
- 配置 "10.7.15.66"
- 配置 配置 IPv4 地址
- 配置 配置 RAN 配置
- 配置 ping 10.7.15.66 配置

name (□□□)

- 環境構築
- 環境構築 "ONS-Lab-Airscale"
- 環境構築 Web UI環境構築 InfluxDB 環境構築
- 環境構築 環境構築環境構築環境構築環境構築
- 環境構築
 - "NYC-Site-A-BS1"
 - "LAX-Tower-Main"
 - "TestLab-Airscale-01"

port (□□□)

- □□□□□□□□

- 端口 "8080"
- 端口 8080 AirScale 端口 8080
- 端口 8080
- 端口 8080

web_username (端口)

- WebLM 端口
- 端口 "Nemuadmin"
- 端口 API 端口
- 端口 8080/8080
- 端口 8080

web_password (端口)

- WebLM 端口
- 端口 "nemuuser"
- 端口 8080/8080
- 端口 8080

端口

端口 aircscales 端口

```

airscales: [
  %{
    address: "10.7.15.66",
    name: "ONS-Lab-Airscale",
    port: "8080",
    web_username: "Nemuadmin",
    web_password: "nemuuser"
  },
  %{
    address: "10.7.15.67",
    name: "Site-A-Tower-1",
    port: "8080",
    web_username: "admin",
    web_password: "password123"
  },
  %{
    address: "192.168.100.50",
    name: "Site-B-Indoor",
    port: "8080",
    web_username: "admin",
    web_password: "different_password"
  }
]

```

InfluxDB

```
config :ran_monitor, RanMonitor.InfluxDbConnection,
  auth: [
    username: "monitor",
    password: "sideunderTexasgalaxyview_61"
  ],
  database: "nokia-monitor",
  host: "10.179.2.135"
```

0000 0000 InfluxDB 00000000000000000000000000000000

auth

auth (auth)

- InfluxDB 认证
- **username:** InfluxDB 用户名 "monitor"
- **password:** InfluxDB 密码 "sideunderTexasgalaxyview_61"
- 注意 2.x 版本 API 变更

database (数据库)

- InfluxDB 数据库/租户
- 名称 "nokia-monitor"
- **InfluxDB 1.x** 支持
- **InfluxDB 2.x** 支持
- 用于 RAN 数据库

```
# InfluxDB 1.x
influx -execute 'CREATE DATABASE "nokia-monitor"'

# InfluxDB 2.x
influx bucket create -n nokia-monitor -o your-org
```

host (主机)

- InfluxDB 主机
- 地址 "10.179.2.135"
- 端口 IP 地址
- 默认 8086 端口
- 配置
 - "localhost" - 本地 RAN 数据库
 - "10.179.2.135" - InfluxDB 主机
 - "influxdb.example.com" - 主机

InfluxDB 配置

配置

- RAN 8086 InfluxDB
- `curl http://10.179.2.135:8086/ping`

- Web UI
- 30 720
-

- InfluxDB 60
 -
 - InfluxDB
-

部署指南

环境准备

1. 安装依赖

```
# 安装数据库驱动
password: "omnitech2024"

# 配置数据库连接
password: System.getenv("DB_PASSWORD") || "default_password"
```

2. 配置权限

```
chmod 600 config/runtime.exs
chown ran_monitor:ran_monitor config/runtime.exs
```

3. 部署操作

- 将 runtime.exs 文件移动到 `.gitignore`
- 创建数据库连接池
- 启动应用

配置

1. 数据库配置

- 数据库名称
- 数据库连接池大小 `pool_size`
- 数据库连接超时时间 2 秒

2. 日志配置

- 日志级别
- 60 秒轮转一次
- 保留最近 15 个日志文件

3. InfluxDB 設定

- 環境変数設定
- InfluxDB 接続
- InfluxDB 接続

環境変数

1. 環境変数

- IP 設定
- RAN 接続
- 接続
- 接続

2. 環境変数

- `debug` 環境変数
- `info` 環境変数
- 環境変数

3. RAN 設定

- 接続
- 接続
- InfluxDB 接続
- 接続

環境

1. 環境

- `runtime.exe`
- 接続
- 接続
- RAN 接続

2. 設定

```
# 1. 設定 runtime.exs
vim config/runtime.exs

# 2. 設定 Elixir 設定
elixir -c config/runtime.exs

# 3. サービス再起動
systemctl restart ran_monitor
```

3. 検証

- サービス再起動後CPU使用率が低下
- サービス再起動後メモリ使用率が低下
- 50+ のサービスが InfluxDB に接続
- MySQL と InfluxDB の接続が正常

まとめ

サービス再起動後CPU使用率が低下

```

import Config

#
=====
#   配置
#
=====

config :ran_monitor, RanMonitor.Repo,
  username: System.get_env("DB_USERNAME") || "ran_monitor_user",
  password: System.get_env("DB_PASSWORD") || "change_this_password",
  hostname: System.get_env("DB_HOST") || "localhost",
  database: "ran_monitor",
  stacktrace: false, # 是否显示堆栈跟踪
  show_sensitive_data_on_connection_error: false, # 是否在连接错误时显示敏感数据
  pool_size: 15 # 6 个进程 * 2 + 3 个进程

#
=====
# Web 配置
#
=====

config :ran_monitor, RanMonitor.Web.Endpoint,
  http: [ip: {0, 0, 0, 0}, port: 8080],
  check_origin: false,
  secret_key_base: System.get_env("SECRET_KEY_BASE") ||
"generate_with_mix_phx_gen_secret",
  server: true

config :control_panel, ControlPanelWeb.Endpoint,
  url: [host: "0.0.0.0", port: 9443, scheme: "https"],
  https: [
    ip: {0, 0, 0, 0},
    port: 9443,
    keyfile: "priv/cert/server.key",
    certfile: "priv/cert/server.crt"
  ]

config :ran_monitor, RanMonitor.Web.Nokia.Airscale.Endpoint,
  url: [host: "0.0.0.0"],
  http: [ip: {0, 0, 0, 0}, port: 9076],
  server: true

```



```

#
=====

# 配置
#
=====

config :logger,
  level: :info # 配置

config :logger, :console,
  format: "$time $metadata[$level] $message\n",
  metadata: [:request_id]

#
=====

# 配置
#
=====

config :ran_monitor,
  general: %{
    mcc: "001",
    mnc: "001"
  },
  nokia: %{
    ne3s: %{
      webhook_url: "http://10.179.2.135:9076/webhook",
      private_key: Path.join(Application.app_dir(:ran_monitor, "priv'
"external/nokia/ne.key.pem"),
      public_key: Path.join(Application.app_dir(:ran_monitor, "priv'
"external/nokia/ne.cert.der"),
      reregister_interval: 30
    },
    airtowers: [
      # 配置 A - 配置
      %{
        address: "10.7.15.66",
        name: "Site-A-Main-Tower",
        port: "8080",
        web_username: "admin",
        web_password: System.get_env("BS_SITE_A_PASSWORD") || "passwo
      },

```

```
# 00 A - 000
%{
  address: "10.7.15.67",
  name: "Site-A-Backup-Tower",
  port: "8080",
  web_username: "admin",
  web_password: System.getenv("BS_SITE_A_PASSWORD") || "password",
},

# 00 B - 00
%{
  address: "10.7.16.10",
  name: "Site-B-Indoor-DAS",
  port: "8080",
  web_username: "admin",
  web_password: System.getenv("BS_SITE_B_PASSWORD") || "password",
},

# 00 C - 00
%{
  address: "192.168.100.50",
  name: "Site-C-Rooftop",
  port: "8080",
  web_username: "admin",
  web_password: System.getenv("BS_SITE_C_PASSWORD") || "password",
},

# 000 - 0000
%{
  address: "10.5.198.100",
  name: "Lab-Test-Airscale-01",
  port: "8080",
  web_username: "Nemuadmin",
  web_password: "nemuuser"
},

# 000 - 00
%{
  address: "10.5.198.101",
  name: "Lab-Dev-Airscale-02",
  port: "8080",
  web_username: "Nemuadmin",
  web_password: "nemuuser"
}
```

```

    ]
  }

#
=====
# InfluxDB 配置
#
=====

config :ran_monitor, RanMonitor.InfluxDbConnection,
  auth: [
    username: System.get_env("INFLUX_USERNAME") || "monitor",
    password: System.get_env("INFLUX_PASSWORD") || "change_this_password",
  ],
  database: "nokia-monitor",
  host: System.get_env("INFLUX_HOST") || "10.179.2.135"

```

部署

- [InfluxDB](#) - 数据库
- [AirScale](#) [配置](#) - 配置
- [Prometheus](#) - 监控系统
- [Grafana](#) [配置](#) - 监控系统
- [API](#) [配置](#) - REST API 配置
- [部署](#) - 部署

MDT 配置 TCE

配置 (TCE)

RAN Monitor 配置 LTE/5G 配置

TCE

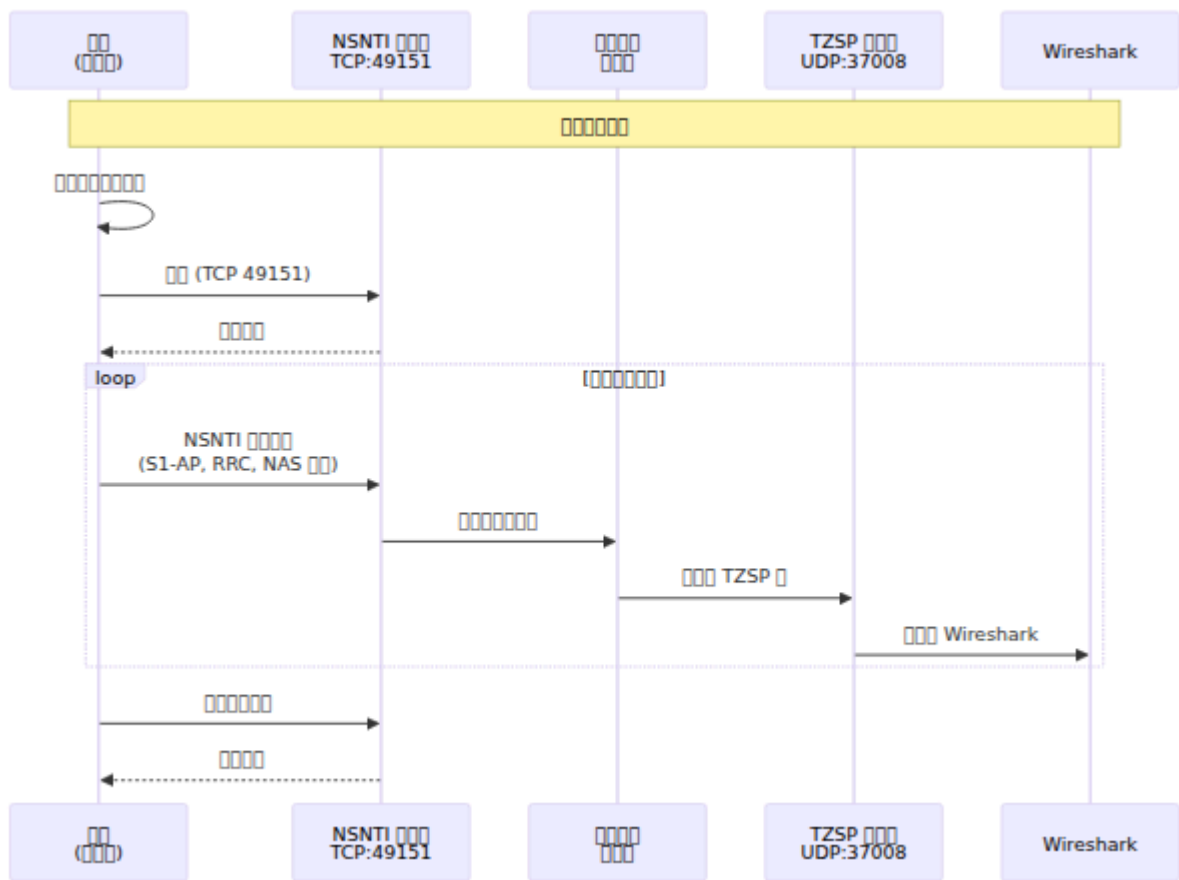
AirScale 配置

- **S1-AP** - eNodeB 到 EPC 配置
- **RRC** - 配置
- **NAS** - 配置
- - PDCP 配置

TCE

配置	配置	配置	配置
NSNTI 配置	TCP	49151	配置
TZSP 配置	UDP	37008	配置 Wireshark 配置
配置	ASN.1	-	配置 S1-AP 到 RRC 配置

□□□□



~~~~~NSNTI ~~~~ (49151)~TZSP ~~~~~

~~~~~

1. ~ TCE ~~~~

```
ss -tlnp | grep 49151
# ~~~: LISTEN 0.0.0.0:49151
```

2. ~~~~~

- ~~~~ IP ~~~ RAN Monitor ~~~
- ~~~~~ 49151
- ~~~~~ (S1-AP, RRC, NAS)
- ~~~~~

3. ~ Wireshark~

~~~~~

- ~~~ TZSP ~~~~~
- ~~~~~ `udp port 37008`

~~~~~

RAN Monitor ~~~~ UDP ~~~~~ RRC ~~~~ Wireshark ~“~~~~”~~~~~
~~~~~

### ~~ 1~~~ Wireshark GUI

- ~~ ~ ~ ~> ~~~...
- ~~ + ~~~~~
- iii. ~~~~~

| 項目        | 値     | 項目 | 項目   | 説明                |
|-----------|-------|----|------|-------------------|
| udp.port  | 36412 | 項目 | (項目) | SCTP              |
| sctp.port | 36412 | 項目 | (項目) | S1AP              |
| udp.port  | 37000 | 項目 | (項目) | TZSP              |
| udp.port  | 37001 | 項目 | (項目) | LTE RRC (DL-CCCH) |
| udp.port  | 37002 | 項目 | (項目) | LTE RRC (DL-DCCH) |
| udp.port  | 37003 | 項目 | (項目) | LTE RRC (BCCH)    |
| udp.port  | 37004 | 項目 | (項目) | LTE RRC (PCCH)    |
| udp.port  | 37008 | 項目 | (項目) | TZSP              |
| udp.port  | 37011 | 項目 | (項目) | LTE RRC (UL-CCCH) |
| udp.port  | 37012 | 項目 | (項目) | LTE RRC (UL-DCCH) |
| udp.port  | 38000 | 項目 | (項目) | MAC-LTE           |
| udp.port  | 38001 | 項目 | (項目) | MAC-LTE (DL)      |
| udp.port  | 38002 | 項目 | (項目) | MAC-LTE (BCH)     |
| udp.port  | 38003 | 項目 | (項目) | MAC-LTE (PCH)     |
| udp.port  | 38011 | 項目 | (項目) | MAC-LTE (UL)      |
| udp.port  | 38012 | 項目 | (項目) | MAC-LTE (RACH)    |

項目 2項目 decode\_as\_entries 項目

~~~~~ ~/.config/wireshark/decode\_as\_entries (Linux/Mac) ~  
%APPDATA%\Wireshark\decode_as_entries (Windows)~

```
# RAN Monitor TZSP ~~~~  
decode_as_entry: udp.port,36412,(none),SCTP  
decode_as_entry: sctp.port,36412,(none),S1AP  
decode_as_entry: udp.port,37000,(none),TZSP  
decode_as_entry: udp.port,37001,(none),LTE RRC  
decode_as_entry: udp.port,37002,(none),LTE RRC  
decode_as_entry: udp.port,37003,(none),LTE RRC  
decode_as_entry: udp.port,37004,(none),LTE RRC  
decode_as_entry: udp.port,37008,(none),TZSP  
decode_as_entry: udp.port,37011,(none),LTE RRC  
decode_as_entry: udp.port,37012,(none),LTE RRC  
decode_as_entry: udp.port,38000,(none),MAC-LTE  
decode_as_entry: udp.port,38001,(none),MAC-LTE  
decode_as_entry: udp.port,38002,(none),MAC-LTE  
decode_as_entry: udp.port,38003,(none),MAC-LTE  
decode_as_entry: udp.port,38011,(none),MAC-LTE  
decode_as_entry: udp.port,38012,(none),MAC-LTE
```

~~~~~



번호	프로토콜	채널/신호	기능
<b>36412</b>	S1AP	-	S1AP 프로토콜 (eNodeB ↔ EPC)
<b>37000</b>	RRC	신호	RRC 프로토콜
<b>37001</b>	RRC	DL-CCCH	다운링크 공통 채널
<b>37002</b>	RRC	DL-DCCH	다운링크 전용 채널
<b>37003</b>	RRC	BCCH-DL-SCH	방송 채널 (다운링크)
<b>37004</b>	RRC	PCCH	Paging 채널
<b>37008</b>	TZSP	-	TZSP 프로토콜
<b>37011</b>	RRC	UL-CCCH	업링크 공통 채널 (RRC)
<b>37012</b>	RRC	UL-DCCH	업링크 전용 채널 (RRC)
<b>38000</b>	MAC-LTE	신호	MAC 프로토콜
<b>38001</b>	MAC-LTE	신호	MAC 프로토콜
<b>38002</b>	MAC-LTE	BCH	방송 채널
<b>38003</b>	MAC-LTE	PCH	Paging 채널
<b>38011</b>	MAC-LTE	신호	MAC 프로토콜
<b>38012</b>	MAC-LTE	RACH	랜덤 액세스 채널

이 표는 LTE 시스템의 주요 프로토콜과 채널을 요약한 것입니다.

```
# 0000 TZSP 000
tzsp

# 000000
slap || rrc || mac-lte

# 00000 RRC 00
udp.port == 37011 || udp.port == 37012

# 00000 RRC 00
udp.port == 37001 || udp.port == 37002

# 00 RRC 0000
rrc.rrcConnectionRequest || rrc.rrcConnectionSetup

# 000000
slap.HandoverRequired || slap.HandoverCommand
```

00

000000

- 000000 RF 00
- 000000
- 000000 (RSRP, RSRQ, SINR)
- 000000

000000

- 0000000000
- 0000000
- 0000
- 0000000

000000

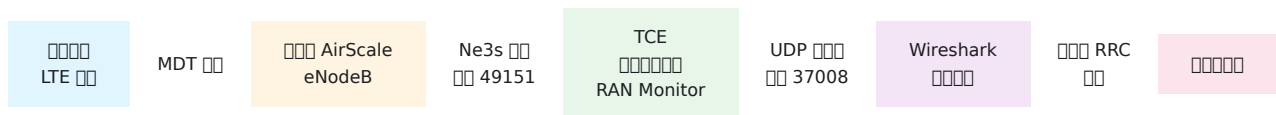
- PCI 00000
- 0000000

- 配置
- 部署

## 部署

配置 (MDT) 配置 (RSRP, RSRQ, 配置) 配置  
 Omnitouch RAN Monitor Web UI 配置 AirScale 配置 MDT 配置  
 Wireshark 配置

## 部署



TCE Omnitouch RAN Monitor 配置 Ne3s 配置  
 Wireshark 配置

## 部署

### 部署

Airscale 配置 配置 配置

配置 ONS

### 部署

- TCE Omnitouch RAN Monitor
- Wireshark 3.0+
- Wireshark TCE Lua
- AirScale 配置

# MDT

RAN Monitor TCE Wireshark

## 1

RAN Monitor Web UI TCE

1. Web UI: `https://<ran-monitor-ip>:9443`
- 2.
- 3.
- 4.
- 5.
6. TCE
  - IP: `<RAN Monitor IP>`
  - : `49151`
- 7.
- 8.
- 9.

AirScale ONS

## 2 AirScale MDT

MDT

- MDT MDT
- PLMN
- 5000ms
- RSRP RSRQ
- 

ONS

## 3 環境構築

ここでは AirScale を利用して MDT を TCE-TCE で観測する環境を構築する。

## Wireshark で MDT を観測

### Wireshark のインストール

1. Wireshark をインストール
2. Linux 上では `lo`、macOS 上では `lo0`、Windows 上では `Loopback` を選択
3. `udp port 37008` をフィルタリング
4. 観測開始

Wireshark で `S1AP` (InitialUEMessage, Attach request) / `LTE RRC` (`RRCConnectionReject`, `RRCConnectionReestablishment`) の TCE を観測する。

### MDT の観測

観測結果の表示方法

```
# 0000 RRC 0000
lte-rrc.measurementReport

# 000000 RRC 00
udp.dstport >= 37011 && udp.dstport <= 37012

# 00 RSRP 00 (< -100 dBm)
lte-rrc.rsrpResult < 40

# 00 RSRQ 00 (< -12 dB)
lte-rrc.rsrqResult < 22
```

0000

MDT 000 **RRC MeasurementReport** 000000000000

- 000000: 000000 RSRP 0 RSRQ
- 000000: 000000 RSRP 0 RSRQ
- 00 **ID**: 0000000000 ID
- **GPS** 00: 000000000000

0 Wireshark 000 RRC 000000000000

```
000000 (RRC)
└─ UL-DCCH-Message
    └─ message: measurementReport
        └─ MeasurementReport
            └─ measResults
                └─ measResultServCell (0000 RSRP/RSRQ)
                    └─ measResultNeighCells (00000)
```

00000000

00000000

1. 00 → 00000000 → 00 **CSV**
2. 0000: `lte-rrc.rsrpResult`, `lte-rrc.rsrqResult`, `lte-rrc.physCellId`

### 3. ExcelPython 00000000

0000

0000: 00 RSRP/RSRQ 0000

```
lte-rrc.rsrpResult < 40 || lte-rrc.rsrqResult < 22
```

0000: 000000000000

```
lte-rrc.MeasResultListEUTRA
```

0000: 000 RSRP 000 RSRQ 0000

```
lte-rrc.rsrpResult > 50 && lte-rrc.rsrqResult < 20
```

0000

**Wireshark** 000000

- 00 TCE 000000 `ps aux | grep beam`
- 00 Wireshark 00000000 `udp port 37008` 0000
- 00 AirScale 000000000000
- 00000000 TCE IP/000000

000000

- 0000000000 (MDT + 000000)
- 000000000000
- 000000000 MDT (LTE Release 10+)

000000000000 **AirScale** 00000000 **ONS**

--	--	--	--	--	--	--

- ☐ `MDT` `lte-rrc`
- ☐ `AirScale` `TCE IP (RAN Monitor IP)` `49151`
- ☐ `RAN Monitor` `TCE`
- ☐ `MDT` `lte-rrc`
- ☐ `Wireshark` `udp port 37008`
- ☐ `lte-rrc.measurementReport`
- ☐ `lte-rrc`

11

- **Omnitouch 0000 (ONS):** 00 AirScale 000000000000



# 目次

## RAN 概要

概要

---

## 目次

1. 概要
  2. 概要
  3. 概要
  4. Web UI 概要
  5. 概要
  6. 概要
  7. 概要
  8. 概要
  9. 概要
- 

## 目次

概要 RAN 概要

## 目次

### 1. 概要

- 概要
- 概要
- 概要

### 2. 概要

- 部署環境
- Web UI 環境
- 監視環境
- 通知環境

3. 監視

- 監視対象
- 監視手段
- 監視ツール
- 監視ルール

4. 通知

- 通知対象
- 通知手段
- 通知ツール
- 通知ルール

まとめ

環境構築

- RAN 監視環境構築 (ps aux | grep ran\_monitor)
- データベース (MySQL, InfluxDB)
- 監視ツール
- 通知ツール

監視環境

通知環境

まとめ

- Web UI 環境

- 確認する
- 確認する
- 確認する

確認

## 1. 確認

```
# 確認
ping <device-ip>

# 確認
telnet <device-ip> 8080
```

確認 ping 確認 telnet 確認  
確認確認 確認 - 確認確認確認確認

## 2. 確認

Web UI → 確認 → 確認 → 確認

- IP 確認
- 確認確認確認 8080
- 確認

config/runtime.exs 確認

```
%{
  address: "10.7.15.66", # 確認 IP
  name: "Site-A-BS1",
  port: "8080",          # 確認
  web_username: "admin", # 確認
  web_password: "password" # 確認
}
```

## 3. 確認

Web UI → 確認 → 確認

□□□

- [error] Authentication failed → □□□□□
- [error] Connection refused → □□/□□□□□
- [error] Timeout → □□□□□□
- [error] Certificate error → □□□□□□/□□□□□

□□□□□

□□□□□

1. □□□□□□□□□□□□□□
2. □□ RAN □□□□□□□□□□□□□□
3. □□□□□□□□□
  - RAN □□□ → □□□□ 8080
  - □□ → RAN □□□□□ 9076□webhooks□
4. □ RAN □□□□□□□□□□□

□□□□□□□

1. □□□□□□□□ WebLM □□□□□□□□
2. □□ config/runtime.exs □□□□□
3. □□ RAN □□□□
4. □□□□□□□□□□□□□□

□□/□□□□□□□

1. □□□□□□□□□□□□□□
2. □□□□□□□□□□□□□□
3. □□□□□□□□□□ telnet <device-ip> 8080
4. □□□□□□□□□□□□□□

□□□□□□□/□□□□□□□

1. □□□□□□□□□□□□□□
  - priv/external/nokia/ne.key.pem
  - priv/external/nokia/ne.cert.der
2. □□□□□□□□□□□□□□

3. 配置 Nokia 设备

4. 配置 Nokia 设备

---

## 配置 Nokia 设备

配置

- 配置 Nokia 设备
- 配置“Nokia”设备
- Web UI 配置设备   设备/设备
- 配置设备

配置

### 1. 配置设备

Web UI → 设备 → 配置 → 配置设备

- 配置设备
- 配置设备
- 配置设备

### 2. 配置设备

在 `config/runtime.exs` 中

```
nokia: %{
  ne3s: %{
    reregister_interval: 30 # 设备 30-60 秒
  }
}
```

### 3. 配置设备

- 配置设备
- 配置设备 `ping <device-ip> -c 100`
- 配置设备

#### 4. 設定

```
# RAN 設定
date
```

```
# 設定
# 設定
```

設定

設定

1. `reregister_interval` 30
2. RAN
- 3.

設定

- 1.
- 2.
3. /
- 4.

設定

1. RAN NTP
- 2.
- 3.

設定

- 1.
- 2.
- 3.

## ❓❓ 部署環境構築

概要

- 既存 Web UI 環境に“監視”環境を追加
- 既存 InfluxDB 環境を拡張
- Grafana 環境を構築
- InfluxDB 環境を拡張

環境構築

### 1. 環境構築

Web UI → 監視

- 監視環境構築
- 監視環境構築
- 監視環境構築

### 2. 既存 InfluxDB 拡張

Web UI → InfluxDB 拡張

- 監視環境構築
- RAN 監視環境 InfluxDB

環境構築

```
# 既存 RAN 監視環境  
curl http://<influxdb-host>:8086/ping
```

### 3. 環境構築

概要

- [error] InfluxDB write failed → 監視環境構築
- [error] Failed to collect metrics → 監視環境構築
- [info] Metrics collected: 0 → 監視環境構築

## 4. InfluxDB

InfluxDB

```
# InfluxDB 1.x
influx -database 'nokia-monitor' -execute '
    SELECT COUNT(*) FROM PerformanceMetrics
    WHERE basebandName=''Site-A-BS1''
    AND time > now() - 1h
'

# InfluxDB 2.x
influx query 'from(bucket:"nokia-monitor")
    |> range(start: -1h)
    |> filter(fn: (r) => r.basebandName == "Site-A-BS1")
    |> filter(fn: (r) => r._measurement == "PerformanceMetrics")
    |> count()'
```

### InfluxDB

1. InfluxDB
2. `config/runtime.exs`
  - 
  - 8086
  - /
  - /API
3. RAN
4. 8086
5. RAN

### InfluxDB

1. /
2. InfluxDB
3. API
4. `config/runtime.exs`



## 5. RAN 環境

### InfluxDB 環境構築

1. ディスク容量 `df -h`
2. ディスクフォーマット
3. ディスクパーティション
4. RAN 環境構築

### 環境構築

1. ディスクパーティション
2. ディスクフォーマット webhook URL 設定
3. ディスクパーティション
4. RAN 環境 webhook 環境構築 9076

## 環境構築

### 環境構築

### 環境

- Grafana 環境構築
- ディスクパーティション
- InfluxDB 環境構築

### 環境構築

### 1. ディスクパーティション

### 環境構築

```
# ディスクパーティション  
journalctl -u ran_monitor --since "2025-12-29" --until "2025-12-30"
```

## 2. 環境構築

Web UI → ① → ② → ③“インストール”

- インストール完了
- インストール

## 3. ③ InfluxDB ④

④InfluxDB ⑤

- ④ InfluxDB ⑤
- インストール/アップグレード

⑥

## RAN ⑦

- インストール完了
- インストール
- インストール

⑧

- インストール
- インストール
- インストール完了
- インストール

## InfluxDB ⑨

- インストール完了
- ⑧ RAN インストール完了
- ④ InfluxDB ⑤
- インストール

⑩

- ⑧ RAN インストール完了

- 設定ファイルの作成
  - InfluxDB のインストール
  - 設定ファイルの修正/確認
- 

# Web UI 構築

## Web UI のインストール

前提

- 接続先 `https://<ran-monitor-ip>:9443`
- 接続先
- SSL 証明書

確認

### 1. Web UI のインストール

インストール

```
[info] Running ControlPanelWeb.Endpoint with cowboy
```

確認

```
ps aux | grep control_panel  
netstat -tulpn | grep 9443
```

### 2. 接続

接続

```
telnet <ran-monitor-ip> 9443
```

■ RAN 接続確認

```
curl -k https://localhost:9443
```

### 3. 確認

```
# 確認
sudo iptables -L -n | grep 9443

# 
sudo firewall-cmd --list-ports
```

確認

確認

#### 1. 確認

```
sudo firewall-cmd --add-port=9443/tcp --permanent
sudo firewall-cmd --reload
```

#### 2. 確認

### Web UI 確認

1. 確認 `config/runtime.exs` Web 確認
2. 確認 SSL 確認
3. 確認
4. 確認 RAN 確認

### SSL 確認

#### 1. 確認

```
ls -l priv/cert/omnitouch.pem
ls -l priv/cert/omnitouch.crt
```

#### 2. 確認

```
openssl x509 -in priv/cert/omnitouch.crt -text -noout
```

3. 確認します。

4. RAN 確認

確認

1. `config/runtime.exs` 確認

2. 確認

3. `CONTROL_PANEL_HTTPS_PORT` 確認

---

## Web UI 確認

確認

- Web UI 確認
- 確認
- 確認

確認

1. 確認

`config/runtime.exs` 確認

```
airscales: [  
  # 確認  
]
```

2. 確認

MySQL 確認

```
mysql -u ran_monitor_user -p ran_monitor -e "SELECT * FROM  
airscales;"
```

3. 確認

確認

確認

確認

1. `config/runtime.exs` 確認
2. RAN 確認
3. Web UI 確認

確認

1. MySQL 確認
2. `config/runtime.exs` 確認
3. 確認
4. RAN 確認

確認

MySQL 確認

確認

- 確認
- Web UI 確認
- “確認”

確認

1. MySQL 確認

```
systemctl status mysql
#
systemctl status mariadb
```

2. 確認

RAN 確認

```
mysql -h <mysql-host> -u <username> -p <database>
```

### 3. 配置

在 `config/runtime.exs` 中

```
config :ran_monitor, RanMonitor.Repo,  
  username: "ran_monitor_user",  
  password: "password",  
  hostname: "localhost",  
  database: "ran_monitor",  
  pool_size: 10
```

配置完成

## MySQL 配置

### 1. 安装 MySQL 服务

```
systemctl start mysql
```

### 2. 配置 MySQL 服务

### 3. RAN 配置 MySQL 服务

配置完成

### 1. 配置 MySQL 服务

### 2. 配置 MySQL 服务

### 3. 配置 MySQL 服务 `config/runtime.exs`

### 4. 配置 RAN 服务

配置完成

### 1. 配置 MySQL 服务

### 2. 配置 MySQL 服务 3306

### 3. 配置 MySQL 服务

配置完成

1. MySQL max\_connections
  2. pool\_size
  3. RAN
- 

## MySQL

### CPU

- RAN CPU RAM
- 
- 
- 

#### 1.

```
# CPU
top -p $(pgrep -f ran_monitor)

#
ps aux | grep ran_monitor
```

#### 2.

- =
- 

#### 3.



- 100 = 100 CPU/1000
- 1000 10 1000000

#### 4. 1000000000

config/runtime.exs

```
pool_size: 10 # 1000000
```

1000000

1000000000000

1. 1000000000
2. 1000000000 CPU/RAM
3. 10000000000
4. 1000000000000

1000000000

1. 1000000 pool\_size
2. 1000000000 2 1000 + 5 1000 Web UI
3. 100 RAN 1000
4. 1000000000

1000000

1. 1000000000
2. 1000000000000000000
3. 1000000000 RAN 1000
4. 1000000000000

#### InfluxDB 1000000

1. 100 InfluxDB 1000000
2. 100 InfluxDB 1000000
3. 1000000 InfluxDB 1000
4. 1000000000000

# Web UI

- Web UI
- 
- 

## 1.

RAN

```
top
free -h
df -h
```

## 2.

```
# MySQL
mysql -u root -p -e "SHOW VARIABLES LIKE 'slow_query_log%';"
```

## 3.

- 1.
- 2.
- 3.

環境構築

1. MySQL のインストール
2. MySQL の起動と停止
3. MySQL の接続

基本操作

1. データの取得
2. データの挿入
3. データの更新

応用

データベースのバックアップと復元

まとめ

- データベースの重要性
- データベースの設計
- データベースの最適化

おわりに

1. データベースの役割

データベースは、データを整理し、検索、更新、削除を行うためのシステムです。

2. Webhook の仕組み

Webhook は、特定のイベントが発生したときに、外部のシステムに通知を送信するための仕組みです。

```
netstat -tulpn | grep 9076
```

おわりに

```
tcp 0 0.0.0.0:9076 0.0.0.0:* LISTEN
```

### 3. Webhook

webhook URL RAN

```
http://<ran-monitor-ip>:9076/webhook
```

### 4.

webhook

### 5. InfluxDB

```
influx -database 'nokia-monitor' -execute '
SELECT COUNT(*) FROM Alarms WHERE time > now() - 1h
'
```

### Webhook

1. `config/runtime.exs` webhook
2. 9076
3. RAN
- 4.

- 1.
2. webhook URL
- 3.

1. 設定 RAN モニタ 9076
2. 設定
3. 設定 `telnet <ran-monitor-ip> 9076`

## InfluxDB 設定

1. InfluxDB 設定
2. 設定
3. InfluxDB 設定
4. 設定

## Web UI 設定

1. 設定
2. 設定
3. 設定
4. 設定

systemd 設定

```
journalctl -u ran_monitor -f
```

mix 設定

- 設定

- 設定 - 設定

- 10 - 10000000
- 10 - 10000
- 10 - 10000000
- 10 - 10000000

10000000

- 100000000 "Site-A-BS1"
- "error" 1 "failed"
- "InfluxDB" 1 "MySQL"
- "registration" 1 "session"

## InfluxDB 10

10000000

```
influx -database 'nokia-monitor' -execute '
SELECT * FROM PerformanceMetrics
WHERE basebandName=''Site-A-BS1''
AND time > now() - 5m
LIMIT 10
'
```

10000000

```
influx -database 'nokia-monitor' -execute '
SELECT COUNT(*) FROM PerformanceMetrics
GROUP BY basebandName
'
```

100000

```
influx -database 'nokia-monitor' -execute '
SELECT * FROM Alarms
WHERE time > now() - 1h
'
```

# MySQL 操作

数据库操作

```
SELECT name, address, port, registration_status  
FROM airscales;
```

数据库操作

```
mysql -u ran_monitor_user -p ran_monitor -e "SHOW PROCESSLIST;"
```

网络操作

网络操作

```
# 网络  
ping <device-ip>  
  
# 网络  
telnet <device-ip> 8080  
nc -zv <device-ip> 8080  
  
# 网络  
traceroute <device-ip>
```

网络操作

```
# 网络  
sudo iptables -L -n -v  
  
# 网络  
sudo iptables -L -n | grep 8080
```

---

# 目錄

## 前言

## 第一章 緒論

### 1. 背景

- 背景說明
- 研究動機
- 研究目標

### 2. 研究範圍

- 研究範圍說明
- 研究限制
- 研究貢獻

### 3. 研究方法

- RAN 研究方法
- 數據收集
- 數據分析MySQLInfluxDB
- 結果驗證

### 4. 結論

- 研究結論
- 研究建議
- 未來研究方向

### 5. 參考文獻

- 參考文獻1
- 參考文獻2
- 參考文獻3



## 目標

- **Web UI** 構築 - 監視画面
- 監視対象 - サーバ
- 監視項目 - CPU使用率
- **Grafana** 構築 - 監視画面
- 監視対象 - サーバ
- 監視項目 - CPU使用率
- 監視画面 - 監視画面

## 前提条件

### 環境

- Web UI 構築環境
- 監視対象 `journalctl -u ran_monitor`
- 監視画面

### 監視項目

- CPU使用率
- メモリ使用率
- ディスク使用率
- ネットワーク使用率

### 監視画面

- 監視対象 Web UI 構築
- 監視項目
- InfluxDB 構築
- 監視画面

## 結論

### 監視画面

1. 安裝環境與軟體
  2. 安裝硬體
  3. 安裝 Omnitouch 安裝與設定
  4. 安裝軟體
    - 安裝環境與軟體
    - 安裝硬體
    - 安裝軟體與設定
    - 安裝軟體
- 

## 安裝環境

- 安裝環境 - 安裝環境
- **Web UI** 安裝 - 安裝環境
- 安裝環境 - 安裝環境
- 安裝環境 - 安裝環境
- 安裝環境 - 安裝環境
- **Grafana** 安裝 - 安裝環境
- 安裝環境 - 安裝環境

# Web UI 00

RAN 000000 - 000000

00 RAN 0000 Web 000000000000

---

## 00

1. 00
2. 00 Web UI
3. 0000
4. 0000
5. 000000
6. 0000
7. 0000
8. 0000 eNodeB 00
9. 000000
10. 00000000
11. InfluxDB 0000
12. 000000
13. PM 000000
14. 000000
15. 000000
16. Web UI 0000

## 0000

- 000000 - 00000000
-



RAN 管理プラットフォーム Web 管理プラットフォーム管理プラットフォームWeb UI 管理プラットフォーム管理プラットフォーム管理プラットフォーム

## Web UI 管理プラットフォーム Grafana

### Web UI 管理プラットフォーム

- 管理プラットフォーム
- 管理プラットフォーム
- 管理プラットフォーム
- 管理プラットフォーム
- 管理プラットフォーム

### Grafana 管理プラットフォーム

- 管理プラットフォーム
- 管理プラットフォーム KPI 管理プラットフォーム
- 管理プラットフォーム
- 管理プラットフォーム
- 管理プラットフォーム

管理プラットフォーム Grafana 管理プラットフォーム Grafana 管理プラットフォーム

## 管理プラットフォーム Web UI

管理プラットフォーム HTTPS 管理プラットフォーム

**URL:** `https://<ran-monitor-ip>:9443`

管理プラットフォーム: 9443管理プラットフォーム `CONTROL_PANEL_HTTPS_PORT` 管理プラットフォーム

### SSL 管理プラットフォーム

- 管理プラットフォーム管理プラットフォーム

- 証明書 CA 証明書
- 証明書 config/runtime.exs

証明書証明書証明書

証明書 証明書 5 証明書証明書証明書

証明書

証明書 RAN 証明書証明書

証明書

証明書

- 証明書
- 証明書証明書

証明書

- 証明書/証明書
- 証明書
- 証明書

証明書

- 証明書? ? ? 証明書
- 証明書証明書証明書証明書
- 証明書証明書

証明書

- 証明書
- 証明書
- 証明書

11

- 5 00000
- 00000000000000 = 00000 = 000
- 0000000000
- 000000

--	--	--	--

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

**URL:** `https://<ran-monitor-ip>:9443/nokia/enodeb`

BTS

□ □ □ □

□□□□□□□□□□

項目	内容
項目	項目項目項目
項目	項目項目項目項目項目項目項目項目 <b>PM</b> 項目項目
項目	項目項目項目項目項目項目項目項目
項目	項目項目項目項目項目項目

項目項目

項目	内容
項目	項目項目項目
項目	項目項目項目項目
項目	項目 <b>IP</b> 項目項目
項目	項目項目“項目”項目項目“項目”項目項目
項目	項目項目項目

項目

項目	単位	説明
PM	分	PM の実行時間
PM の PM	分	PM の実行時間 eNodeB PM の実行時間 15 分
PM	分	PM の実行時間
PM	分	NE3S の実行時間
PM	分	PM の実行時間

項目

項目

項目	単位
Ping	分
PM	分
PM	分
PM	分

項目

項目



項目	内容
患者 ID	00000000
患者 ID	00000000
患者名	000000000000COMA
患者名	000000000000

## 患者情報

- 00000000
- 000000 IP 0000
- 000000

## 患者情報

00000000000000000000

## 患者情報??

- 0000000000
- 00000
- 00000000
- 00000000

## 患者情報

- 0000000
- 0000000
- 0000000000
- 0000000000

- 〇〇〇〇〇〇〇〇〇〇〇

〇〇〇〇

- 〇〇〇〇〇〇〇〇〇〇
- 〇〇〇〇〇〇〇〇〇〇
- 〇〇〇〇〇〇〇
- 〇〇〇〇〇〇〇

〇〇〇〇

- 〇〇〇〇〇〇〇〇〇〇〇〇
- 〇〇〇〇〇〇〇〇〇〇
- 〇〇〇〇〇〇
- 〇〇〇〇〇〇〇〇

〇〇〇〇

- 〇〇〇〇〇〇
- 〇〇〇〇〇〇〇〇〇
- 〇〇〇〇〇〇
- 〇〇〇〇〇〇〇

〇〇〇〇〇

〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇

〇〇〇〇

〇〇〇〇〇〇〇

- 〇〇〇〇〇〇〇 - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- 〇〇〇〇〇〇〇 - 〇〇〇〇〇〇〇〇〇〇〇〇
- 〇〇〇〇〇〇〇 - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇

- 00000000 - 00000000
- 00000000 - 000000000000

00000000

- 0000
- 00000
- 00000000DN - Distinguished Name
- 00000000000000000000

00

00000

- 00000000
- 00 = 0000
- 00 = 0000
- 00 = 0000
- 00 = 00
- 00 = 000

000000

- 000000000000
- 00000000
- 000000

00000

- 000000000000000000
- 0000000000
- 00000000

0000000000000000 0000000



- □□□□□□□□□□□□□□□□

□□□□□ **3** □□□

[illegible]

1. **பெயர்** - **பெயர்** **ID**
2. **பெயர்** - **பெயர்** **ID** **பெயர்** **ID**
3. **பெயர்** - **பெயர்** **ID** **பெயர்** **ID**

XXXXXXXXXX ID XXXXXXXXXXXXXXXXXXXX

--	--	--	--	--	--

□□□□□

1.
2.
3.
4.
5.

□□□□□□□□□□□□□□ AirScale □□□□□

□□□□ **eNodeBs** □□

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

11

□□□□ eNodeBs □□□□□□

- 
- 
- 

-  

ID

- 
- ID

- 
- 

- 
- 

- 
- 

- 
- 

- 
- 

-

- 確認事項

## 確認事項

確認事項を確認してください

1. 確認事項 **ID**
2. 確認事項 **config/runtime.exs**

```
airscales: [  
  %{  
    address: "10.7.15.66",  
    name: "Site-A-BS1",  
    port: "8080",  
    web_username: "admin",  
    web_password: "password"  
  }  
]
```

3. 確認 **RAN** 確認事項

確認事項を確認してください

## 確認

- 確認事項 確認事項
- 確認事項 確認事項
- 確認事項 確認事項
- 確認事項 確認事項

## 確認事項

確認事項を確認してください

□ □ □ □

□ □ □ □ □ □ □ □

- 11 - 1111111
- 11 - 1111111111
- 11 - 1111
- 11 - 1111
- 11 - 1111
- 11 - 111111
- 11 - 111111
- 11 - 1111111

□ □ □    □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

11

□ □ □ □ □ □

- ১০০০০০০০০০০০০
- ১০০০০০০০ 500 ০০০০
- ১০০০০০০

111

- 00/00 - 00000000000000
- 00 - 0000000000
- 0000 - 0000000000000000

□ □ □ □ □

- 00 - 00/00/0000
- 000 - 0000
- 00 - 0000
- 00 - 0000
- 00 - 0000
- 00 - 0000



## 目標

### 基礎知識

- ネットワークの種類
- ネットワークの IP 設定
- ネットワークの接続

### 応用知識

- ネットワークのトラブルシューティング
- ネットワークのセキュリティ
- ネットワークの最適化

### まとめ

- ネットワークの種類
- ネットワークの IP 設定
- ネットワークの接続
- ネットワークのトラブルシューティング

### 参考文献

- ネットワークの種類
- ネットワークの IP 設定
- ネットワークの接続

## 目標

- ネットワークの種類
  - ネットワークの IP 設定
    - ネットワークの種類
    - ネットワークの IP 設定
    - ネットワークの接続
  - ネットワークの接続
  - ネットワークのトラブルシューティング
-

# 環境構築

環境構築 InfluxDB 環境構築

## 環境構築

環境構築

- 環境構築/環境構築
- `config/config.exe` 環境構築
- 環境720 環境30 環境

環境構築

- 環境構築/環境構築
- 環境構築

環境構築

- 環境構築/環境構築
- 環境構築

## 環境構築

環境構築

環境構築

- 環境構築
- 環境構築/環境構築
- 環境構築

環境構築

- 環境構築 - 環境 PM 環境構築
- 環境 - 環境構築
- 環境 - 環境構築
- 環境 - 環境構築

## 目標

- 監視対象 - 監視対象の範囲を明確にする
- 監視対象 - 監視対象の範囲を明確にする
- 監視対象 - 監視対象の範囲を明確にする

## 監視対象

1. 監視対象 - 監視対象の範囲を明確にする
2. 監視対象 - 監視対象の範囲を明確にする
3. 監視対象 - 監視対象の範囲を明確にする
4. 監視対象 - 監視対象の範囲を明確にする

## 監視対象

- **720** 監視**30** 分 - 監視対象
- **2160** 監視**90** 分 - 監視対象
- **4320** 監視**180** 分 - 監視対象
- **8760** 監視**365** 分 - 監視対象

## 監視

- 監視対象の範囲を明確にする
- 監視対象の範囲を明確にする
- 監視対象の範囲を明確にする
- 監視 InfluxDB 監視対象

監視対象の範囲を明確にする

監視対象の範囲を明確にする

---

# InfluxDB 監視

監視対象 InfluxDB 監視対象

**URL:** https://<ran-monitor-ip>:9443/nokia/influx

*InfluxDB*

	InfluxDB
InfluxDB	2.x



項目	単位
入力	入力データ InfluxDB データ
出力	出力データ
処理	処理データ
結果	結果 InfluxDB データ
速度	速度データKB/s 1 MB/s
時間	時間データ
位置	位置データ

入力データ 出力データ 処理データ 結果データ 速度データ 時間データ 位置データ

入力

項目	単位
入力	入力データ
出力	出力データ
処理	処理データ

環境構築

項目	内容
DB	InfluxDB インストール
DB	InfluxDB インポートスクリプト8086
API	InfluxDB クエリ
DB	バックアップ
DB	バックアップスクリプト3分間隔で実行

確認

環境構築完了確認

- InfluxDB インポート - インポートスクリプト
- インポート - インポートスクリプト
- インポート - インポートスクリプト
- インポート - インポートスクリプト

確認

完了 30 分間隔で実行

項目

項目	説明
項目 + 項目名	項目名
項目 + 項目	項目名 PM 項目
項目	項目 InfluxDB 項目名
項目名/項目	InfluxDB 項目名
項目	項目 - 項目名
項目	項目 - 項目名

項目

- 項目 InfluxDB 項目名
- 項目名
- 項目名
- 項目名
- 項目名

項目名

項目 InfluxDB 項目名

**URL:** `https://<ran-monitor-ip>:9443/nokia/metrics`



How to connect InfluxDB with Airscale

# InfluxDB

???	
	InfluxDB
	InfluxDB

# Airscale InfluxDB

InfluxDB

項目	説明
Airscale	監視器
監視器	監視 PM 監視器
監視	監視監視監視
監視	監視監視監視
監視	監視監視監視監視
監視器	監視監視監視監視

監視監視監視監視監視監視

監視器

Erlang VM 監視器

項目	説明
監視	監視 Erlang VM 監視
監視器	Erlang 監視監視
監視器	監視監視監視監視XMLJSON 監視
監視器	監視監視
監視器	監視 Erlang 監視
監視器	監視 CPU 監視監視監視0 監視

PM

PM 5

PM

- PM
- PM
- PM
- PM
- PM

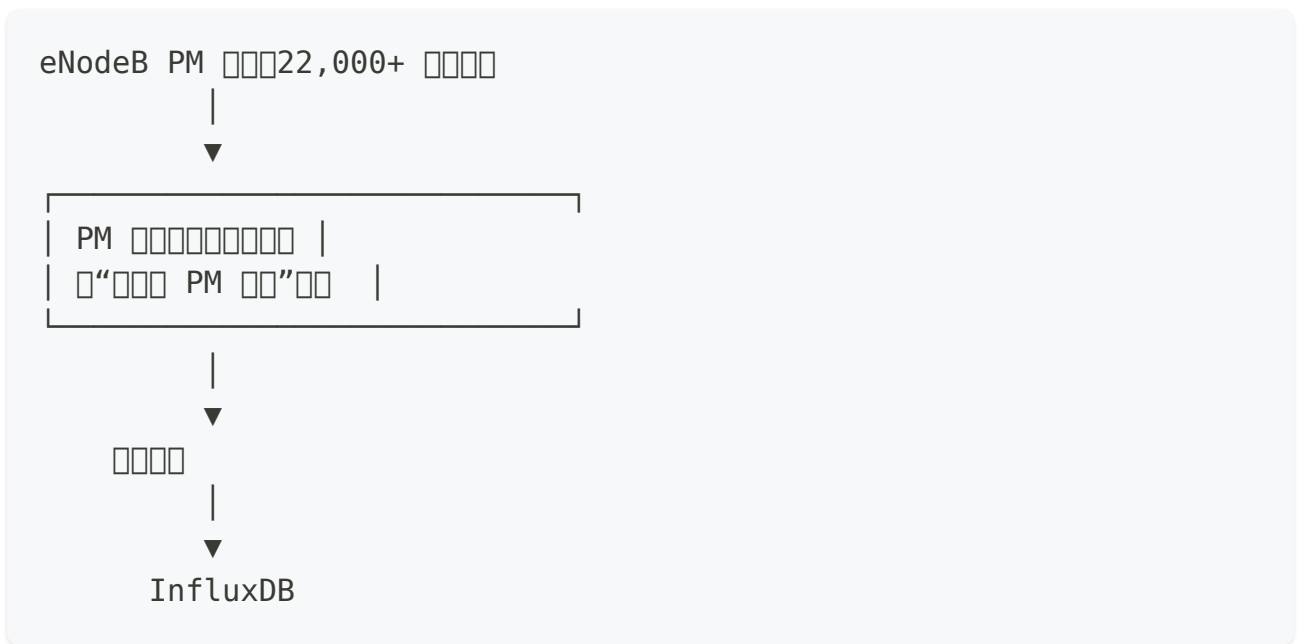
# PM

PM InfluxDB PM AirScale 22,000 PM PM

URL: <https://<ran-monitor-ip>:9443/nokia/pm-filters>

PM

**PM** □□□□□□□□



"  PM  "  PM

□ □ □ □

項目	内容
PM 担当者	InfluxDB のインストール
インストール	22,000+ のインストール
インストール	config_keys.csv

□□□ **PM** □□□□

□□□□□□□□□□□□□□ InfluxDB□

1111

- 電話番号 ID 番号
- 電話番号LTE/5G-NR 番号
- 電話番号 CSV 番号

資料

- 環境省環境政策評価部
- 環境省環境政策評価部
- 環境省環境政策評価部 `pm_counters.csv` 環境省

環境省環境政策評価部

環境省環境政策評価部 PM 環境省環境政策評価部

資料

- 環境 ID 環境省環境政策評価部
- 環境省環境政策評価部

資料

- 環境省環境政策評価部
- 環境省環境政策評価部 200 年

環境省環境政策評価部

資料	環境省	資料
<b>LTE</b>	M8xxx	LTE L1/L2/L3 資料
<b>WCDMA</b>	M5xxx	3G WCDMA 資料
<b>5G-NR</b>	M55xxx	5G NR 資料
<b>5G-資料</b>	M51xxx	5G 資料
<b>5G-資料</b>	M40xxx	5G 資料

資料

資料

- 配置する `priv/pm_filters.etf`
- 設定を確認する
- 設定を確認する

### 確認

InfluxDB 設定を確認する

- **PM** 設定する PM 設定を確認する
- 設定を確認する/設定を確認する

設定を確認する **PM** 設定を確認する

### 設定

設定する AirScale 設定を確認する

**URL:** `https://<ran-monitor-ip>:9443/nokia/firmware`

### 設定

設定を確認する RAN 設定する WebEM “設定”設定を確認する HTTP URL 設定を確認する

### 設定

項目	設定
項目	設定する AirScale .zip 設定
項目 <b>URL</b>	設定する WebEM 設定を確認する HTTP URL
<b>MD5</b> 項目	設定を確認する
項目	設定を確認する

## □□□□□□

1. □□□□□□ RAN □□
2. □□ □□ **URL** □□□□ URL
3. □ WebEM □□□□□□ □□□□
4. □□ URL □□□ □□□□□
5. □□□□□□□□

□□□□□□□□□□□□□□□□□□□□ □□□□□□

---

## □□□□□□

□□□□□□□□□□□□□□□□

**URL:** `https://<ran-monitor-ip>:9443/nokia/data`

□□□□□□□□ ETS □□□□□□□□ InfluxDB □□□□□□□□

# ETS 環境保護

環境保護の重要性

項目	内容
環境保護の重要性	環境保護の重要性
環境保護の重要性	環境保護の重要性
環境保護の重要性	環境保護の重要性
InfluxDB のインストール	InfluxDB のインストール

環境保護 ETS の重要性

## 環境保護

環境保護の重要性

- 環境保護の重要性
- 環境保護の重要性

環境保護の重要性 /tmp

## InfluxDB のインストール

InfluxDB のインストール

- 環境保護PM の重要性
- 環境保護
- 環境保護

環境保護の重要性 “環境保護” InfluxDB の重要性

環境保護の重要性 InfluxDB の重要性



□□□□□

□□□□□□□□□□□□□□□□□□ `config/runtime.exs` □□□□□□ ETS □□

□□

- □□□□□□□□□□
- □□□□□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□□□

## Web UI □□□□□

□□ Web UI □□□□□□□□□□□□

□□□□□□□

□□□□ □□□□□□□□□□□□□□

□□□□

1. □□□□□□□
2. □□□□□□□□□□□□□□
3. □□□□□□□□□□□□
4. □□□□□□□□/□□□□□□
5. □□□□□□□□□□
6. □□□□□□□□□□□□

□□□□ < 5 □□

□□□□□

□□□□ □□□□□□□□

□□□□

1. 註冊帳號
2. 選擇註冊類型
3. 填寫基本資料
4. 驗證手機號碼
5. 設定密碼
6. 完成註冊
7. 登入帳號
8. 開始使用

註冊流程圖表如下：

註冊步驟

第一步：選擇註冊類型

第二步：

1. 選擇註冊類型
2. 填寫基本資料
3. 驗證手機號碼
4. 設定密碼 ID
5. 登入 ID 密碼
6. 完成註冊
7. 登入帳號
8. 開始使用 15-30 天
9. 開始使用

註冊注意事項

- 註冊時請注意
- 註冊時請注意
- 註冊時請注意
- 註冊時請注意

## 環境構築

OS: Ubuntu 18.04 LTS

前提

1. 仮想マシン eNodeBs をインストール
2. 仮想マシン ID と IP を取得
3. 仮想マシン `config/runtime.exs` を編集
4. 仮想マシン RAN をインストール
5. 仮想マシンを起動
6. 仮想マシンを停止
7. 仮想マシンを InfluxDB に接続
8. 仮想マシンを Grafana に接続
9. 仮想マシン Grafana を起動

仮想マシンを起動するコマンド

## 仮想マシンを起動

OS: Ubuntu 18.04 LTS

前提

1. 仮想マシンをインストール
2. 仮想マシン/仮想マシンをインストール
3. 仮想マシンをインストール
4. 仮想マシンをインストール
5. 仮想マシンを ping する
6. 仮想マシンをインストール
7. 仮想マシンをインストール
8. 仮想マシンをインストール
9. 仮想マシンをインストール
10. 仮想マシン

仮想マシンを起動するコマンド

---

## □□□□

- □□□□ - □□□□□□
- □□□□□□ - □□□□□□
- □□□□□□ - □□□□□□
- □□□□□□ - □□□□
- **Grafana** □□□□ - □□□□□□
- □□□□□□□□ - □□□□□□□□
- □□□□□□ - □□□□□□□□