

# Nokia AirScale 開箱

## ■ RAN Monitor 開箱

---

### ■

1. 開箱
  2. 開箱
  3. 開箱 WebLM 開箱
  4. 開箱
  5. 開箱
  6. 開箱
  7. 開箱
- 

### ■

■ RAN Monitor 開箱 Nokia AirScale 開箱 RAN Monitor 開箱 Nokia Web Element Manager (WebLM) 開箱

■ (PMCADM) 開箱

### ■

■ Nokia AirScale 開箱 **Omnitouch** 開箱

- 開箱
- 開箱
- 開箱
- 開箱

■ **Omnitouch** 開箱 **Omnitouch** 開箱

## Overview

This document provides information about the RAN Monitor.

- **RAN Monitor** - RAN Monitor IP
- **RAN Monitor** - RAN Monitor IP
- **RAN Monitor** - RAN Monitor IP
- **RAN Monitor** - RAN Monitor IP

Table 1

Item	Value	Default
RAN Monitor IP	RAN Monitor IP	10.179.2.139
RAN Monitor	Port (Port: 9076)	9076
Port	Port	60

## WebLM

### 1. Web Element Manager

1. Web Element Manager
2. Web Element Manager

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

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

3. □□□□□□□□□□

## 2. 2. 2. 2. 2. 2. 2.

□□□□

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

□ □

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

# PMCADM

□□□□□□ (□□□) □□

1. **Current BTS configuration**
2. **CURRENT\_BTS\_CONF-1**
3. **MRBTS-X** (X = BTS ID)
4. **MNL-1** (MNL)
5. **MNLENT-1** (MNL)
6. **PMCADM-1** (PMCADM)

PMCADM-1 仕様書

2

Structure 1

TLS		RSA (TLS)
	RAN Monitor IP	RAN Monitor IP (10.179.2.139)
	RAN Monitor	9076
		60s
TLS		false

項目	項目	項目
SDL 項目	項目	1
SDL 項目	項目	12345678
項目	項目 0 項目	false (項目)

項目 3項目

項目

- 項目 - 項目 IP 項目
- 項目
  - 項目 **Create Plan** 項目
  - 項目
  - 項目 ID
- 項目
  - 項目 **Validate Plan** 項目
  - 項目 ID
  - 項目
  - 項目
- 項目
  - 項目 **Activate Plan** 項目
  - 項目 ID
  - 項目
  - 項目

項目XML 項目

項目 XML 項目 PMCADM 項目

```
<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 解説

- `rTPmCollEntityHost` - RAN Monitor IP
- `rTPmCollEntityPortNum` - 9076 (webhook)
- `rTPmCollInterval` - (60s)
- `tlsEnabled` - false
- `sdlPrimaryDestIp` - RAN Monitor IP

10.179.2.139 RAN Monitor IP ID MRBTS-132

## PMCADM-1

## PMCADM-1 解説

PMCADM (PMCADM-1) RAN Monitor IP

解説

- RAN Monitor IP

- 000000000000
- 000000000000
- 000000000000

000000000000

00000000000000000000

**certTypeForTlsAuth - TLS** 00000000

- 000 00 (RSA, DSA, ECDSA)
- 0000 TLS 00000000
- 000 RSA
- 000 00 tlsEnabled = true 000

**rTpmCollEntityHost** - 00000000

- 000 IP 00 (IPv4 0 IPv6)
- 000 00000000 IP 00
- 000 0
- 000 10.179.2.139
- 000 0000000000000000

**rTpmCollEntityPortNum** - 00000000

- 000 00 (1-65535)
- 000 00000000 TCP 00
- 000 9076
- 000 000 RAN Monitor 0000

**rTpmCollInterval** - 0000

- 000 00 (0)
- 000 0000000000
- 000 15s, 30s, 60s, 300s, 900s, 1800s
- 000 60s
- 000 00000 60s000000000 15s

## **tlsEnabled** - TLS

- 是否启用 (true/false)
- 是否启用 TLS
- 是否 false
- 是否启用 TLS

## **sdlMaxUploadFileNumber** - SDL 文件数量

- 文件数量
- 是否启用 TLS
- 是否 1
- 是否启用 TLS

## **sdlNonce** - SDL nonce

- 是否启用 (8 字节)
- 是否 SDL 文件数量
- 是否 12345678
- 是否启用 TLS

## **suppressZeroValueCount** - 抑制零值计数

- 是否启用 (true/false)
- 是否启用 TLS
- 是否 false
- 是否 是否 false 是否启用 TLS

## **UI**

是否启用 RAN Monitor 是否启用

## **RAN Monitor Web UI**

1. 是否 RAN Monitor Web UI: <http://<ran-monitor-ip>:4000/>



2. 网络 eNodeB 网络
3. 网络网络网络网络
4. 网络网络网络 **Connected** (网络)
5. 网络 **Session** 网络 **Active**

网络

- 网络 Connected (网络)
- 网络 Active
- 网络 网络 IP 网络
- 网络 网络网络

网络网络网络 **Pending**

网络网络网络网络网络网络网络网络

□□□□□

- □□□ ID □□□□□□□□
- RAN Monitor □□□□□□□□
- □□□□□□

□□□□□□□□□□

□□□□□□ RAN Monitor□

□□□□□

- PMCADM 設定 IP 設定
- 設定
- 設定 8080
- RAN Monitor 設定

設定

設定 **InfluxDB** 設定

1. RAN Monitor Web UI 設定 **InfluxDB** 設定
2. 設定
3. 設定 設定
4. 設定 設定

設定

- 設定 設定
- 設定 設定
- 設定 設定 0

## 準備作業

1. 仮想マシンを起動
2. ネットワーク設定
3. 必要パッケージのインストール

## 確認

### 仮想マシン RAN Monitor

仮想マシン内から eNodeB に接続

確認

1. ネットワーク接続

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

## 2. 安装 RAN Monitor 软件

- 安装路径为 `config/runtime.exs`
- 设置 IP 地址
- 设置端口

## 3. 安装 RAN Monitor 软件

- 安装路径为
- 设置端口
- 设置端口

## 4. 安装路径

- 安装 WebLM 软件 PMCADM-1 软件
- 安装 RAN Monitor IP 地址
- 设置端口 9076

## 安装 "Pending" 软件

安装路径为 "Pending"

安装路径

### 1. 安装路径

- 安装 RAN Monitor 软件 ID 地址
- 设置端口

### 2. 安装路径

- 安装 runtime.exs 软件
- 设置端口/设置端口

### 3. 安装路径

- 设置端口 30-60
- 设置端口

## 問題

問題 "enetunreach" 問題

問題

### 1. 問題

- RAN Monitor 問題
- 問題
- VLAN/subnet 問題

### 2. 問題

- 9076 問題 (問題)
- 8080 問題 (SOAP API 問題)
- ACL 問題
- RAN Monitor 問題 iptables 問題

### 3. RAN Monitor 問題

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

## InfluxDB 問題

問題 問題

問題

### 1. 問題

- PMCADM-1 rTpmCollInterval 問題
- 問題
- InfluxDB 問題

### 2. InfluxDB 問題

- InfluxDB 問題

- 检查 "Connected" 状态
- 检查日志

### 3. 检查 **RAN Monitor** 配置

- 检查 InfluxDB 配置
- 检查数据库连接
- 检查 API 配置

## TLS/加密

检查 TLS 配置

检查

#### 1. 检查配置

- 检查配置文件
- 检查 RAN Monitor 是否使用 CA 证书

#### 2. 检查 **TLS** 配置


- 检查 `tlsEnabled = false`
- 检查证书路径
- 检查 TLS 版本

#### 3. 检查日志

- 检查日志文件
  - 检查日志内容
  - 检查日志格式
-

## □□□□

### □□□□

- [□□□□](#) - □□□ RAN Monitor □□□□
- [□□□□□□□□](#) - RAN Monitor □□□□
- [Nokia Counter Reference](#) - □□□□□□□□
- [Grafana](#) □□ - □□□□□□□□□□□□□
- [API](#) □□ - □□□□□ REST API □□
- [□□□□□□□□](#) - □□□□□□□□□□

### □□□□

- **config/runtime.exs** - RAN Monitor □□□□

### □□

#### □□□□□□□□□□□□□□

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



# 無線LAN

無線LANの概要

無線LANの歴史

---

## 無線LANの種類

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

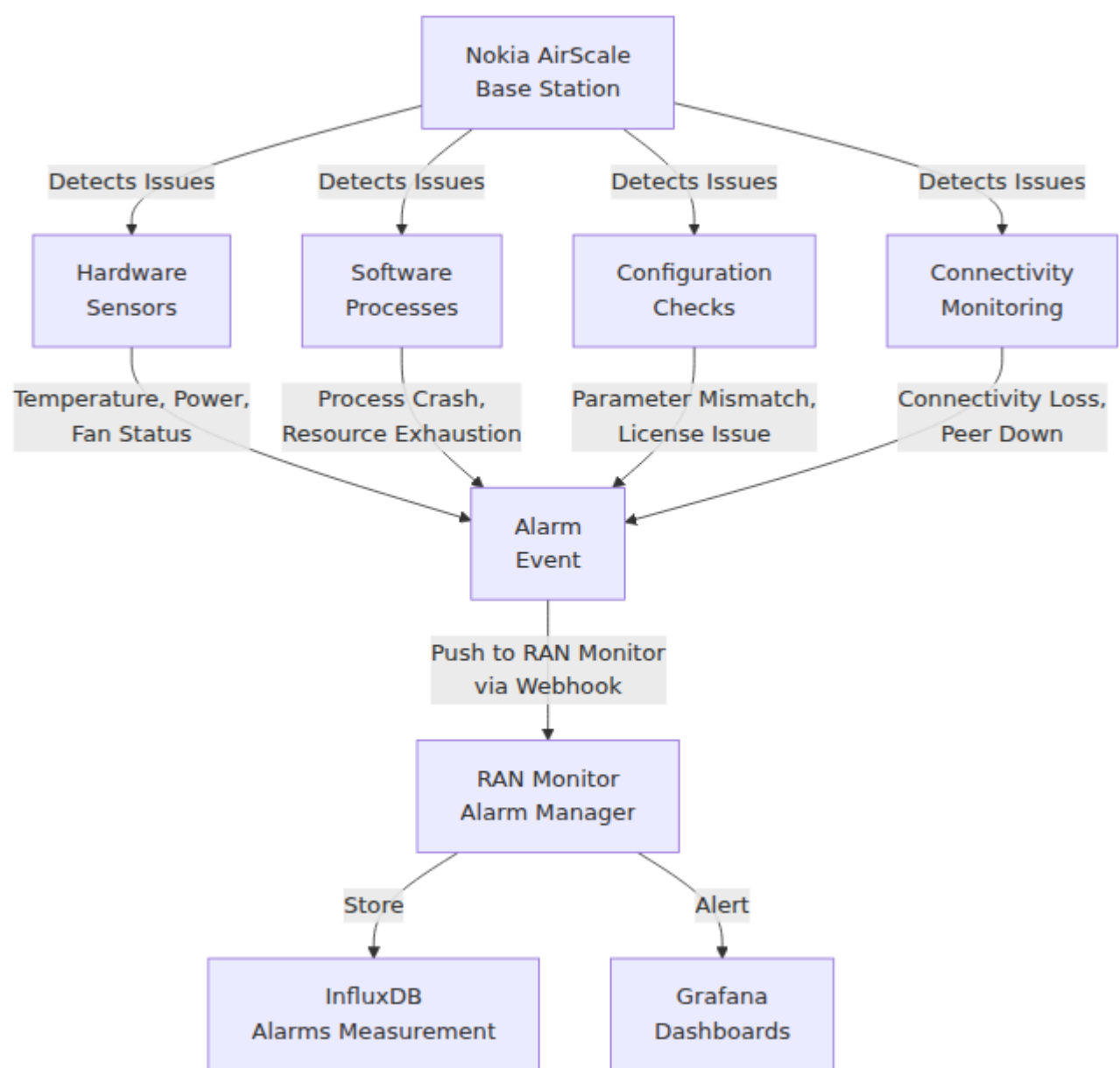
## 無線LANの規格

無線LANの規格は、IEEE 802.11シリーズと、AirScaleシリーズ、RAN Monitorシリーズなどがあります。

無線LANの規格

4G

□□□□



□□□□□□

□□□□□□

項目	値	単位
識別 ID	a1b2c3d4-e5f6-...	文字列
名前	ユーザー名	文字列
住所	"東京都"	文字列
年齢	"S1 20歳"	文字列
所属	/BSC-1/BTS-23/Cell-A	階層型ID
登録日	2025-12-10 14:23:45	日時
性別	男 / 女	文字列

---

□□□□□□

□□□□

□□□

□□□□

□□□  
(□□□)

□□□□

□□□□  
(□□□□□□)

□□□□

□□□  
(□□□□□)

□□□□

□□□  
(□□□□)

□□□  
□□

## 📋 事件履歴

14:23:45 UTC - 📡 異常発生  
↳ 📡 異常発生  
↳ 📡 異常発生“S1 異常”発生

14:23:47 UTC - 📡 RAN Monitor  
↳ NE3S webhook 発生  
↳ 📡 InfluxDB  
↳ 📡 発生

14:23:50 UTC - 📡  
↳ Grafana 📡  
↳ 📡 NOC 📡 Slack 📡  
↳ 📡 PagerDuty 📡

14:24:15 UTC - 📡  
↳ NOC 📡  
↳ 📡

14:28:35 UTC - 📡📡📡  
↳ 📡  
↳ 📡  
↳ RAN Monitor 📡“📡”

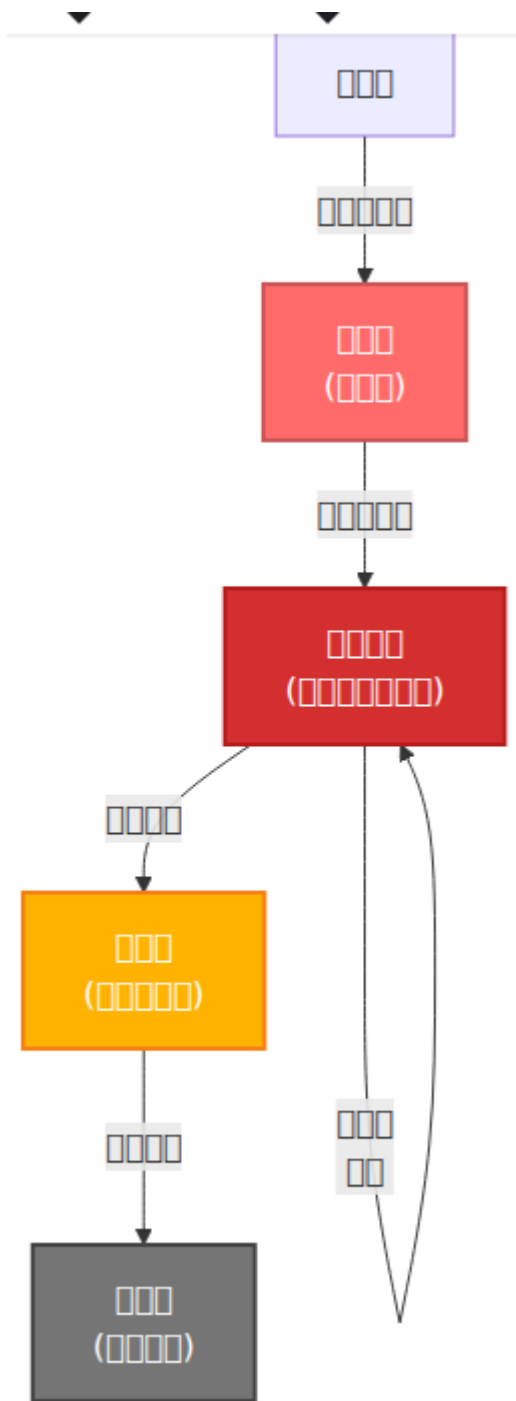
14:28:36 UTC - 📡  
↳ 📡📡📡5 📡 51 📡  
↳ 📡 SLA 📡  
↳ 30 📡

## 📋 設定

RAN Monitor 📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡📡

## 📋 設定

📡 📡📡📡📡📡📡📡



000

- 0000000000000000
- 0000000000000000
- 00000000S1 000
- 0000000000
- 00000000

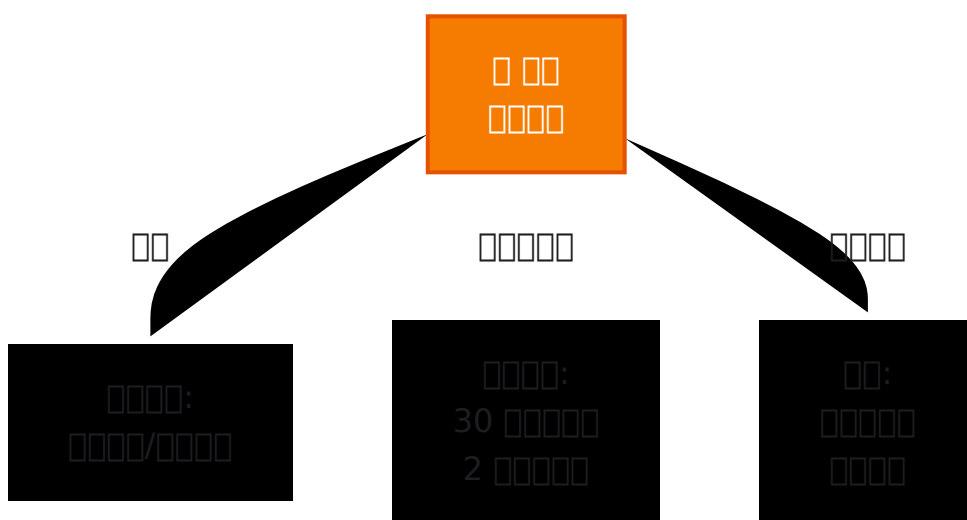
000

- 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 時間

時間切れ

時間 時間切れの可能性



時間

- 時間切れ < 95% 時間 15 時間
- 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

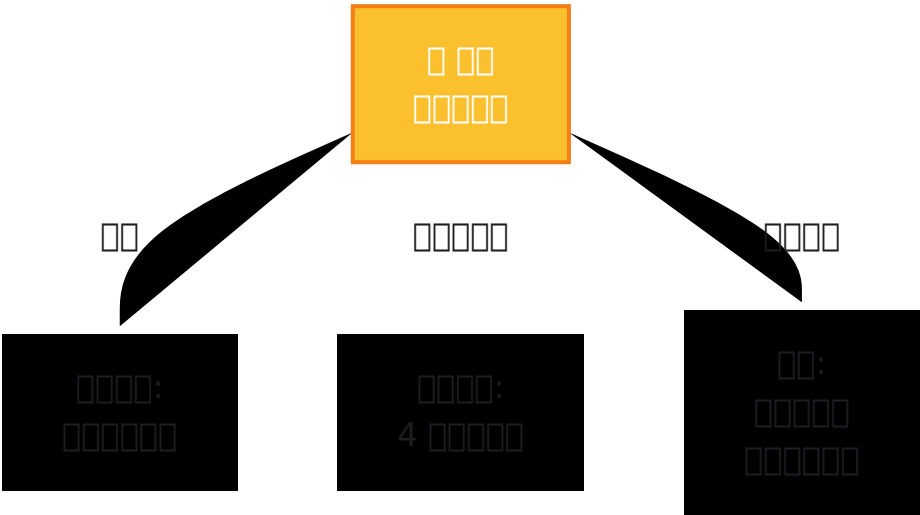
- 時間 NOC 時間 + 時間切れ
- 時間切れ
- 時間 30 時間切れ
- 時間切れ/時間切れ



目標 **SLA** < 30 分

目標

目標 達成率



目標

- 達成率 95-98%
- 達成率
- 達成率
- 達成率
- 達成率
- 達成率 < 5 分/分

目標

- 達成率
- 達成率
- 達成率
- 達成率

目標 **SLA** 達成率

## 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

必要なもの

- 仮想マシン → 仮想マシン OOM
- CPU 使用率 → 高負荷
- メモリ使用率 → 高負荷
- ディスク使用率 → 高負荷

確認事項

確認

1. 仮想マシン
2. 仮想マシン
3. CPU/メモリ/ディスク
4. ディスク使用率

## 環境構築

前提条件

必要なもの

- 仮想マシン → 仮想マシン/メモリ
- DL 使用率 → 高負荷
- メモリ使用率 → 高負荷
- ディスク使用率 → RF 使用率

確認事項

確認

1. 仮想マシン
2. 仮想マシン
3. RF 使用率
4. ディスク使用率

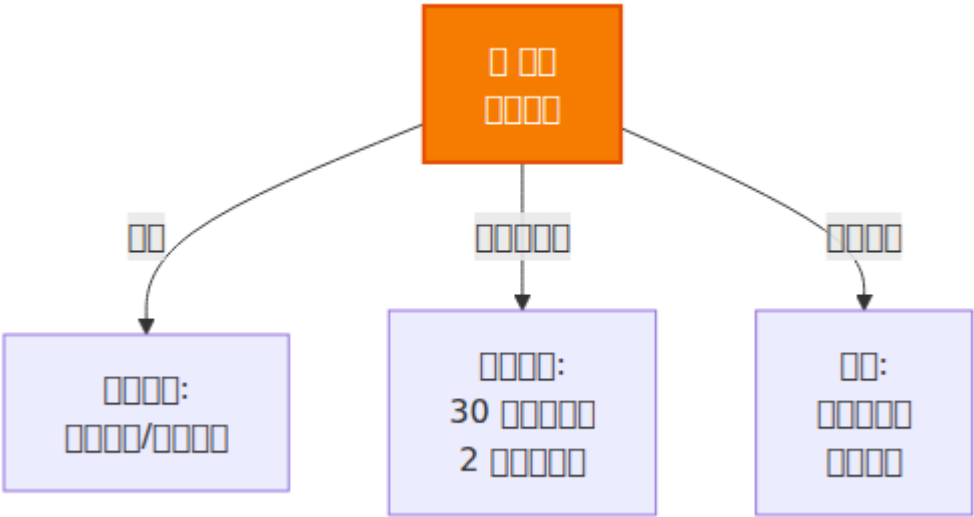
□□□□

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

- 
- 
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/場所/部屋
- デバイスタイプ
- デバイスグループ
- デバイスカテゴリ

## 接続

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

## 監視項目

### 監視項目一覧

### 監視項目

- ユーザー AirScale 接続
- デバイス接続
- RAN Monitor 接続
- デバイス接続

## 監視項目

### 監視項目一覧

- └─ “DL 接続”
- └─ DL 接続 > 95%
- └─ デバイス接続
- └─ デバイス接続
- └─ デバイス接続
- └─ デバイス接続

□□□

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

□□□□□□□□□□□□

□□□□□□□□□□□□□□

□□□□

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

□□□

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

□□□□□□□□□□□□

□□□□□□□□□□□□□

□□□□□□“□□□□□□”□□

```

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

```

□□□□□□□□□□□□

□□□□□□□□□□□□□□

□□□□□□□□



項目	単位	説明
項目	日/月	監視期間
項目	項目	項目
項目	日/月	監視期間
項目	日/月	監視期間
項目	項目	項目 BGP 監視期間

監視期間

監視期間

項目

- ☐ RAN Monitor 監視期間
- ☐ 監視期間
- ☐ 日/月
- ☐ 監視期間
- ☐ 監視期間
- ☐ 監視期間
- ☐ 監視期間 > 30 監視期間

監視期間

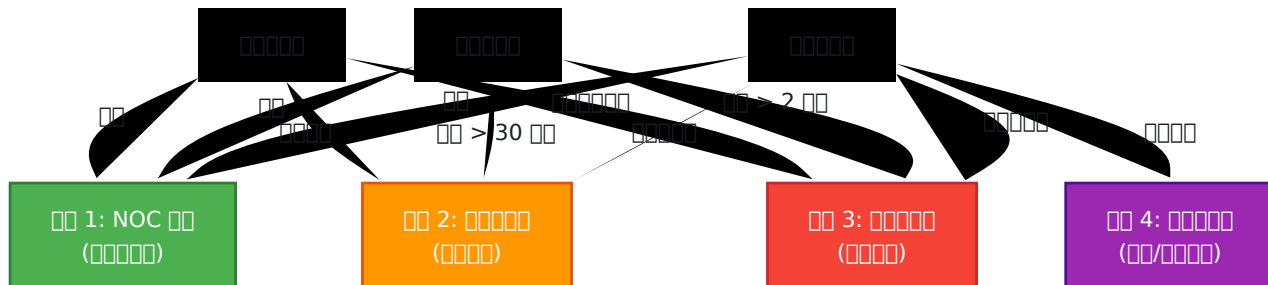
監視期間

項目

- 監視期間
- 項目
- 監視期間 SLA 項目
- 監視期間
- 監視期間

## □□□□

### □□□□



### □□□□□□

#### □□□□□□□□□□□□□□□□ 2□

- □□□□□ 15 □□□□□□
- □□□□□ 30 □□□□□□
- □□□□ NOC □□□□□□□□
- □□□□□□/□□□□□
- □□□□□ > 1 □□□□

#### □□□□□□□□□□□□□□□□ 3□

- □□ 2 □ 1 □□□□□□□□
- □□□□□□□ > 30 □□
- □□□□□□□
- □□□□□□□□
- □□□□□□□□

#### □□□□□□□□□□□ 4□□□□□□□□□□□□

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

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

00000000

- 000000
- 00000000
- 00000000
- 000000

000

- 00000000%
- 00000
- 00000000

00000

- 0000000000
- 000000000000

00000000

- 0000
- 0000000000
- 000000000000

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

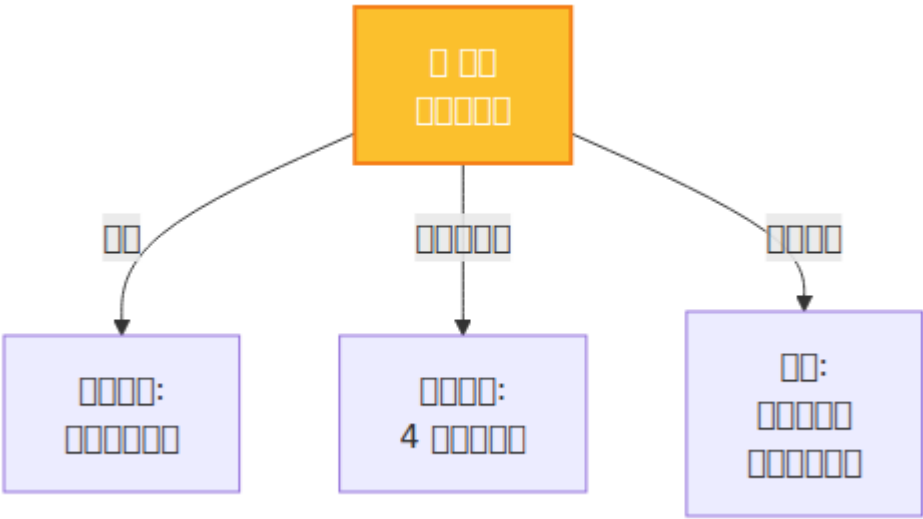
- 障害発生時の対応
- 障害発生時の対応
- 障害発生時の対応
- 障害発生時の対応
- MTTR 障害発生時の対応

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 開発環境構築

開発者 RAN 環境構築 REST API

開発環境構築 REST API 開発者 RAN 環境構築

---

## 目次

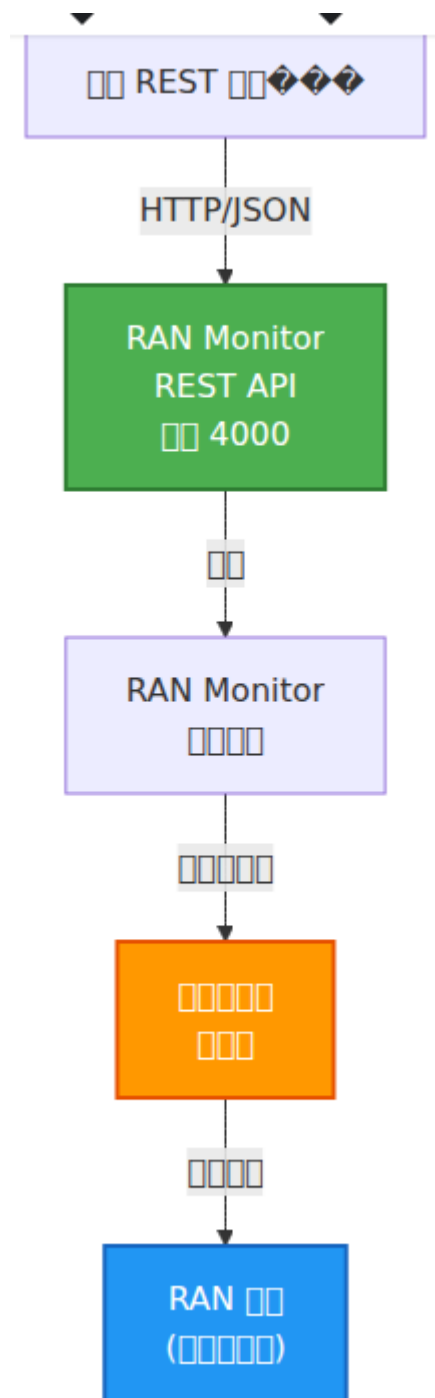
1. 概要
  2. API 仕様
  3. 開発環境構築
  4. 開発環境構築
  5. 開発環境構築
  6. 開発環境構築
  7. 開発環境構築
  8. 開発環境構築
  9. API 仕様
- 

## 目次

RAN Monitor 開発環境構築 REST API 開発者 RAN 環境構築 REST API 開発者 RAN 環境構築

- 開発環境構築 - 開発環境構築
- 開発環境構築 - 開発環境構築
- 開発環境構築 - 開発環境構築
- 開発環境構築 - 開発環境構築
- 開発環境構築 - 開発環境構築

## API 接口



## API 接口

- **RESTful** 接口 - 使用 HTTP 方法 (GET, POST, PUT, DELETE)
- **JSON** 接口 - 使用 JSON 格式
- 接口 - 使用 API
- 接口 - 使用



□□□□

□□□□

□□□□□□

GET /api/v1/devices

□□□

```
{
  "devices": [
    {
      "id": "nokia_bs1",
      "name": "SITE_A_BS1",
      "vendor": "Nokia",
      "address": "192.168.1.100",
      "port": 8080,
      "status": "registered",
      "registered_at": "2025-12-10T14:30:00Z",
      "session_active": true,
      "software_version": "BSC-2250.5.0",
      "license_required": false
    }
  ]
}
```

□□□□□□

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"
}
```

□□□

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

응답

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

요청

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

응답

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

---

응답

요청

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

```
{
  "session_id": "session_unique_identifiser"
}
```

👉👉👉

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

👉👉👉👉👉👉

- 🕒🕒🕒🕒24 🕒🕒
- 🕒🕒🕒🕒🕒🕒🕒🕒
- 🕒 20 🕒🕒🕒🕒🕒🕒

👉👉👉👉👉

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

👉👉👉

```
{
  "session": {
    "active": true,
    "session_id": "session_unique_identifiser",
    "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
        }
      }
    }
  }
}
```

□□□□□□

```
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"
}
```

□□□

```
{
  "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"
}
```

Configuration

Parameter	Type	Value	Unit/Description
BtsName	String	"SITE_A_Cell_1"	Cell Name
MaxUEsServed	Integer	256	Maximum number of UEs
CellTXPower	Integer	40 (dBm)	Transmit Power
EnableCarrierAgg	Boolean	true	Carrier Aggregation
HandoverHysteresis	Integer	3 (dB)	Handover Hysteresis

API

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

Response

```
{
  "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": "□□□□□□"
    }
  ]
}
```

□□□□

□□□□□□

□□□□□□□□□□

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": "□□□□□□□□",
        "value": 125.4,
        "unit": "Mbps",
        "cell_dn": "/BSC-1/BTS-23/Cell-1"
      },
      {
        "id": "M1C2",
        "name": "□□□□□□□□",
        "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": "BTS",
        "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": "OK"
}

```

□□□□□□

GET /api/v1/health/status

{} {}

```
{
  "status": "healthy",
  "devices": {
    "total": 50,
    "registered": 48,
    "active_sessions": 45,
    "unreachable": 2
  },
  "database": {
    "mysql": "connected",
    "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"
}
```

0000000000

---

□□□□□

□□□ **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 のインストール

---

### 前提

インストールには、AirScale のインストールディレクトリに InfluxDB のインストールディレクトリ XML ファイルをインストールする必要があります。



### インストール

- インストール - インストールディレクトリ
  - インストール - インストールディレクトリ
  - インストール - インストールディレクトリ
  - インストール - インストールディレクトリ 100 MB ~ 690 MB
  - **Web** ページ - インストールディレクトリ
  - インストール - インストールディレクトリ
  - **InfluxDB** ページ - インストールディレクトリ
  - インストール - インストールディレクトリ
- 

### インストール

### インストール

インストール AirScale インストール

- インストール 1 インストール
- インストール インストール
- インストール インストール  
- インストール インストール

## 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 構成

- 構成 100 MB 制限
- 構成

- 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 - 0000000000000000
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 ...)

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

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

111

1. [ONS-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.  $\text{Page Size} < 100 \text{ MB}$

4. 00000“00 3 000000 435 KB”

5. 11682 111111 99.6 MB



## 目標

### 確認事項

- 監視対象の容量が 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

### 監視結果

**Web UI** 監視 **Nokia** → 監視

**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 备份 数据

5. 如何 备份 数据 `<AirScaleName>_config_YYYYMMDD_HHMMSS.xml` 如何 备份 数据

如何

- 如何 备份 数据
- 如何 备份 数据
- 如何 备份 数据
- 如何 备份 数据

如何 备份 数据

如何 备份 数据 如何 备份 数据

如何

1. 如何 备份 数据
2. 如何 备份 数据
3. 如何 备份 数据
4. 如何 备份 数据
5. 如何 备份 数据
6. 如何 备份 数据

如何 备份 数据

- 如何 备份 数据 `current.xml` 如何 备份 数据
- 如何 备份 数据 `???`
- 如何 备份 数据

如何 备份 数据

如何 备份 数据 如何 备份 数据

如何 备份 数据

1. 如何 备份 数据
2. 如何 XML 如何 备份 数据 `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

## 問題

1. 問題文を正確に読み取る
  2. 問題の目的を明確にする
  3. 問題の条件を整理する
  4. 問題の背景を理解する
  5. 問題の範囲を限定する
  6. 問題の解決方法を考える
- 

## 解答

### 問題文

問題文を正確に読み取る

問題の目的を明確にする

1. 問題文/問題文
2. XML 問題文
3. 問題文
4. 問題文

問題

- 問題文/問題文
- XML 問題文
- 問題文
- 問題文

問題

<!-- □□□□□□ -->

<!-- 1000000000 -->

<parameter>

<name>cellId</name>

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

&lt;/parameter&gt;

<!-- 2000 -->

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

□□□□□□ ~145 KB □□□□□□□□ AirScale □□□

□□□□□

項目	仕様	標準仕様 <b>145KB</b>	オプション
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

□□□□□

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



## 配置

- 配置存储目录
- 配置存储大小
- 配置存储文件名称
- 配置存储文件后缀

## 编译

### 编译

- 配置存储 **100 MB** 大小
- 配置存储目录
- 配置 `current.xml` 文件名称
- 配置存储文件后缀
- 配置存储文件名称

### 编译

### 编译 ConfigStorage 配置

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

# 配置存储目录
@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

# 配置存储文件名称
def cleanup_old_versions(airscale_name, max_size_bytes \
@max_storage_bytes)
```

### 编译

```
mix compile
# 编译 RAN Monitor 配置
```

## 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/` 0000000000

## 2. 000000

- 配置文件的目录结构
- 配置文件的命名规范

### 3. 配置

- 配置 Web 服务的配置文件
- 配置数据库连接

配置

```
# 配置文件的目录结构
ls -la priv/airscale_configs/

# 配置文件的命名规范
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
- 配置文件的版本号 1

配置

#### 1. 配置

- 配置文件的目录结构
- 配置文件的命名规范

#### 2. 配置

- 配置文件的目录结构
- 配置文件的命名规范
- 配置文件的权限

### 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

注意事项

```
# 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`
- □□□□□□□□“□□”□□□□□□□□□□

□□□□□□□□

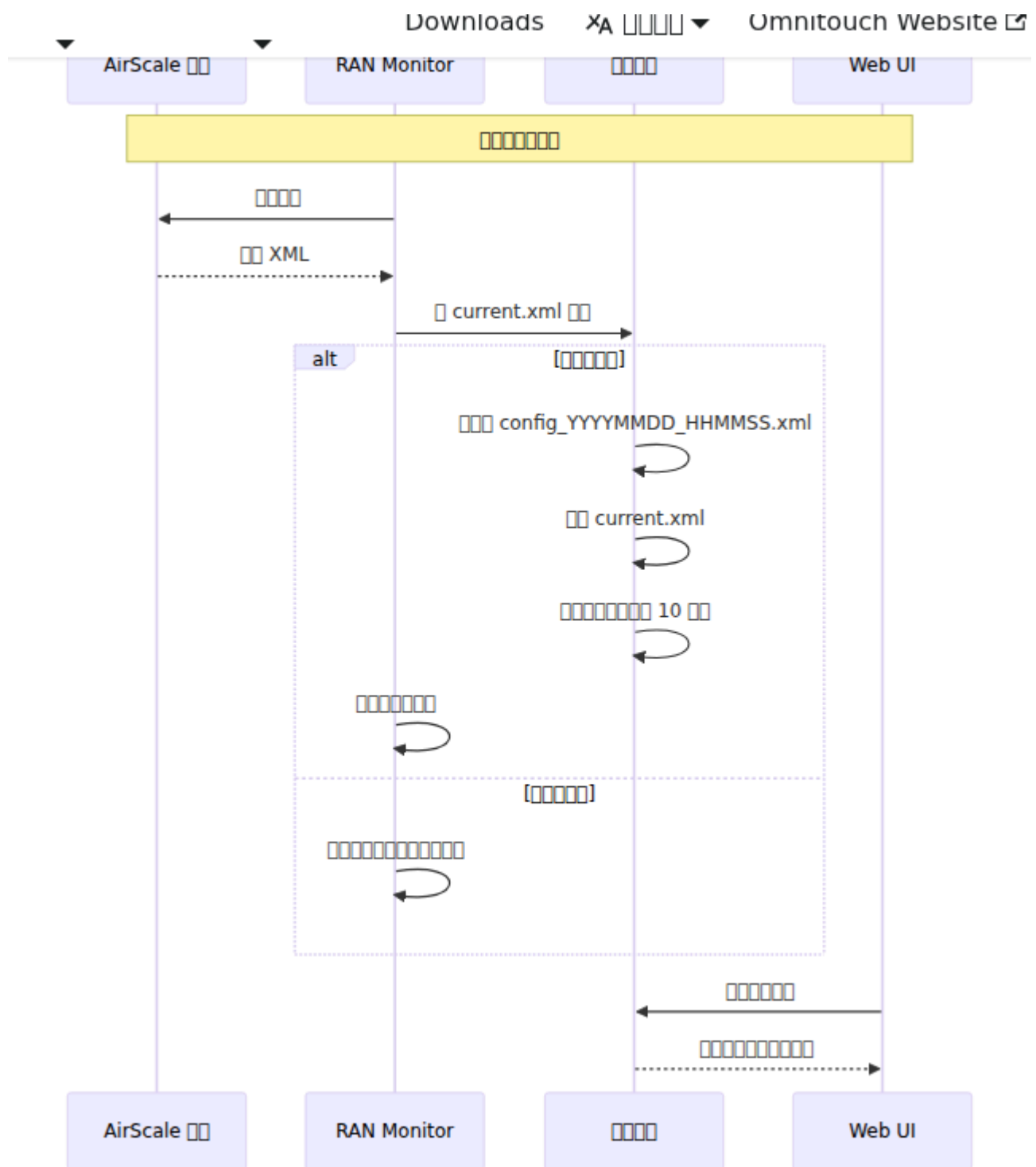
- □□□□□□□□
- □□ → □□ → □□□□□□
- □□□□□□□□□□□□□□□□

□□□□□

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



□□□



## 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

□ □ □ □ □

□□□□□ **IE**x □□□□□

```
# 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

# 最大存储字节数
RanMonitor.Nokia.Airscale.ConfigStorage.max_storage_bytes()
# 104857600 100 MB

# 存储使用情况
RanMonitor.Nokia.Airscale.ConfigStorage.get_storage_usage("ONS-
Lab-Airscale")
# 99614055

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

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

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

---



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

# 安裝說明

## 簡介

RAN Monitor 是一個用於監控 4G/LTE 網路效能的工具。它使用 InfluxDB 儲存數據，並提供實時的監控和報警功能。

---

## 系統要求

### 硬體要求

- IP 地址: `https://localhost:9443`
- 至少 2GB 記憶體
- 至少 10GB 硬碟空間 eNodeB 數據庫

### 網路要求

- 至少 1 個 eNodeB
- 至少 1 個“測試”數據庫
- 至少 1 個 IP 地址
- 至少 1 個 eNodeB 數據庫

### 安裝步驟

- 下載安裝包
- 安裝 eNodeB 數據庫

□□

□□□□□□□□□□ eNodeB □□□□□□□□□□

---

□ □□

□□□□□□

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

□□ **eNodeB** □□□

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

## 📌📌📌📌

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

## 📌📌📌📌📌

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

□ □□□□

□□□□□

□□□□□□□□□□□□□□□□ *eNodeB* □□□□□□□□□□

□□□□□





環境構築

項目	内容	説明
OS	Ubuntu 16.04	動作環境
言語	Python 2.7	動作環境
ライブラリ	PySpark 2.4.0	動作環境
データベース	MySQL 5.7	動作環境
その他	Java 8	動作環境

設定

環境変数

config/config.exs:

```
config :ran_monitor,
  ecto_repos: [RanMonitor.Repo],
  generators: [context_app: :ran_monitor],
  data_retention_hours: 720 # 30 日間保持
```

保留時間

時間	時間	時間	時間
1 分	1	0.04	1 分
1 時	24	1	1 時
7 日	168	7	7 日
14 日	336	14	14 日
30 日	720	30	30 日
90 日	2160	90	90 日
180 日	4320	180	180 日
1 年	8760	365	1 年

設定

設定

```
export DATA_RETENTION_HOURS=1440 # 60 分
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

= - (retention\_hours \* 3600 )

30  
└ 2025-12-11 10:00:00  
└ 720 30  
└ 2025-11-11 10:00:00  
└ <

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

- 1.
2. eNodeB
3. “”

-

- 使用 InfluxDB 数据库
  - 使用数据库连接池
- 

1. 数据库

2. 数据库

3. 数据库

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

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

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

# 3. 数据库
total = RetentionPolicy.get_total_record_count("SITE-01")
# => 270000

# 4. 数据库
{:ok, deleted_count} = RetentionPolicy.delete_old_records("SITE-01", 720)
# => {:ok, 50000} (50k records)

# 5. eNodeB 数据库
{:ok, deleted_count} = RetentionPolicy.clear_all_records("SITE-01")
# => {:ok, 270000} (270k records)
```

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

- 编译选项
- 编译选项
- 编译选项

7-30

- 
- 
- 

> 30

- 
- 
- 

	7-14	
	30-60	
	90	
	180-365	

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



```
# 查看 InfluxDB 版本
influx bucket list

# 查看磁盘使用情况
df -h /path/to/influxdb/data
```

# 1. 环境准备

## 2. 安装 InfluxDB

- 检查系统是否满足 InfluxDB 的要求
- 安装 InfluxDB 二进制文件
- 配置 InfluxDB 的配置文件

## 3. 启动 InfluxDB

- 启动 InfluxDB 服务
- 验证 InfluxDB 是否正常运行
- 配置 InfluxDB 的日志

## 4. 配置 InfluxDB

1. 配置 InfluxDB 的配置文件
2. 配置 InfluxDB 的日志
3. 配置 InfluxDB 的权限
4. 配置 InfluxDB 的备份
5. 配置 InfluxDB 的监控

## 1. 背景

### 1.1 问题描述

背景

- 网络性能下降
- 网络拥塞

原因

1. 检查网络配置 `ps aux | grep mix`
2. 检查 `RetentionCleanupWorker` 进程  
◦ 检查 `[RetentionCleanupWorker]` 进程
3. 检查 InfluxDB 配置  
◦ 检查 InfluxDB 配置 `https://localhost:9443/nokia/influx`
4. 检查网络配置  
◦ 检查 `config/config.exs` 中的 `data_retention_hours`

### 1.2 解决方案

背景

- 网络性能下降
- 网络拥塞

原因

1. 检查 InfluxDB 配置  
◦ 检查 InfluxDB 配置
2. 检查网络配置  
◦ 检查“网络”配置
3. 检查网络配置  
◦ 检查 `[RetentionPolicy]` 配置
4. 检查 eNodeB 配置  
◦ 检查 eNodeB 配置

## 環境構築

### 前提

- Docker
- Kubernetes

### 手順

1. Dockerインストール
2. 5-10 分待機
3. Dockerインストール  
◦ `@cleanup_interval_ms` 1 分
4. eNodeB インストール

## 確認

### 前提

- InfluxDB UI
- “確認”

### 手順

1. “確認”
2. InfluxDB  
◦ インストール
3. インストール  
◦ InfluxDB
4. eNodeB  
◦ インストール

## 確認

- -



# 快速入门

本文档旨在帮助您快速上手 InfluxDB，了解其基本操作和配置。

## 安装与配置

本文档提供以下信息：

- 适用于不同操作系统的安装指南
- 配置 InfluxDB 以使用不同的存储引擎
- 如何设置 InfluxDB 的日志和监控

## 使用 InfluxDB

本文档提供以下信息：

## 快速入门 InfluxDB 数据库

本文档提供以下信息：

## 快速入门 InfluxDB 查询语言

本文档提供以下信息：

---

# 1. Overview

## 2. Components

File	Function
lib/ran_monitor/database/nokia/airscale.ex	Define retention_hours function
lib/ran_monitor/config/config.ex	Define data_retention_hours() function
config/config.exs	Configure application
lib/ran_monitor/application.ex	Define application module

## 3. Implementation

File
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.

## 4. Summary

### RetentionPolicy

- get\_retention\_hours(airscale) - Get retention hours
- get\_record\_counts(airscale\_name) - Get record counts

- `get_total_record_count(airscale_name)` - 総レコード数取得
- `delete_old_records(name, hours)` - 古いレコード削除
- `clear_all_records(name)` - 全レコード削除

## RetentionCleanupWorker GenServer

- `start_link(opts)` - 起動
  - `init(:ok)` - 初期化
  - `handle_info(:cleanup, state)` - レトリート
- 

1. 準備

2. 実行

1. 実行

```
mix ecto.migrate
```

2. 実行

```
mix phx.server
```

3. 実行

- ブラウザ `https://localhost:9443/nokia/retention`
- 実行結果確認

## 概要

### 1. 目的

- 720 時間
- eNodeB

### 2. 環境

- `config/config.exs`
- UI eNodeB

### 3. 実装

- `[RetentionCleanupWorker]`
  -
-



## □ □□

### □□□□□

1. □□□□ □□ [RetentionPolicy] □ [RetentionCleanupWorker] □□
2. □□□□□ □□□□□□□□□□□□□□□□
3. □□□□□□ □□□□□□□□□□
4. □□□□□ □□□□□□□□□□□□□□

### □□□□

#### □□□

- UI □□□□□□□□□□
  - □□□□ eNodeB □□
  - □□□□□□
  - □□□□□□□□□□
  - □□□□
- 

## □ □□□□

### □□□□

- **InfluxDB v2.x:** □□□□□□□□□□□□□□
- □□□□: □□□□□□□□
- □□: □□□□□□□□
- □□□□: □□□□□□□□□□

### □□□

- InfluxDB □□
- Elixir GenServer □□
- Phoenix LiveView

# 📶📶📶 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 権限
- RAN Monitor のインストール
- インストールスクリプト

### 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/リテネンシーポリシーの作成
- API トークンの作成 (InfluxDB 2.x)
- インストールスクリプト

### 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 → RAN Monitor の 9076 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 管理画面
- 管理 9076 管理Webhook 管理
- 管理 9443 管理HTTPS Web UI
- 管理 MySQL と InfluxDB

## Nokia 管理

### 管理画面

- **IP** 管理 - 管理画面
- 管理 - 管理画面 8080
- 管理 - 管理 WebLM 管理画面
- 管理 - 管理ping 管理
- 管理 - 管理画面

### 管理画面

- 管理 - 管理画面PEM 管理
- 管理 - 管理画面DER 管理
- 管理 Nokia 管理画面 OpenSSL 管理

## Grafana管理画面

- 管理Grafana 8.0+
- 管理 InfluxDB 管理画面
- 管理画面

# 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

InfluxDB

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

## 2.3 Web

Web

```
# 配置 SOAP/API 端
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

# 配置 Web UI HTTPS 端
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"
  ]

# 配置 Webhook 端
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 配置

配置示例如下：



```

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

```

生成证书 CA 证书

生成证书私钥

## 3 安装

安装 RAN Monitor

### 3.1 安装

安装

```
mix ecto.migrate
```

安装

### 3.2 安装 RAN Monitor

安装

```
mix phx.server
```

安装

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

### 3.3 验证

验证

```
[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
```

验证

- Web 開発
- 開発環境
- InfluxDB 構築
- 監視

11

☐ 4 ☐☐☐☐☐☐

□ □ □ □ □ □ □ □ □ □

## 4.1 Web UI

□□□□□□□□

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





□□□□ RAN Monitor □□□□

## 4.2 〇〇〇〇〇〇

Web UI

1. 0000 00 00
2. 0000000000000000
3. 0000000 "0000" 0000
4. 0000000 "0000"
5. 00000000000000000000

□□□□□□/□□□□□□

- 
- 
- 
- 

## 4.3 `Phoenix` InfluxDB

Web UI

1. `Phoenix` InfluxDB
2. `Phoenix`
3. `Phoenix`
4. `Phoenix` “`Phoenix`” “`Phoenix`”

`Phoenix` 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 `Phoenix`

`Phoenix`

Web UI

1. `Phoenix` `Phoenix`
2. `Phoenix` “`Phoenix`”
3. `Phoenix`

`Phoenix` `mix phx.server` `Phoenix`

## 4.5 `Phoenix`

Web UI

1. `Phoenix`
2. `Phoenix`
  - `Phoenix`

- 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

- 1GB以上のWeb UI 用メモリを確保
- 1GB以上のストレージを確保
- 1GB以上のメモリを確保

### インストール

- 1GB以上のメモリを確保 Web UI
- 1GB以上のストレージを確保
- 1GB以上のメモリを確保

### 設定

- Grafana のインストール
- 1GB以上のメモリを確保
- 1GB以上のメモリを確保Slack PagerDuty

### 起動

- CA のインストール
- 1GB以上のメモリを確保
- config/runtime.exs のインストール
- 1GB以上のメモリを確保

## 環境構築

### 前提条件

- 1GB以上のメモリを確保
- 1GB以上のストレージを確保
- 1GB以上のメモリを確保
- 1GB以上のメモリを確保
- 1GB以上のメモリを確保

環境構築

- 環境構築
- 24 時間監視
- 監視
- 監視

環境構築

- Web UI 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
- Web UI 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
- AirScale 環境構築 - 環境構築

環境構築

環境構築

- ping 監視
- 環境構築

- 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

---

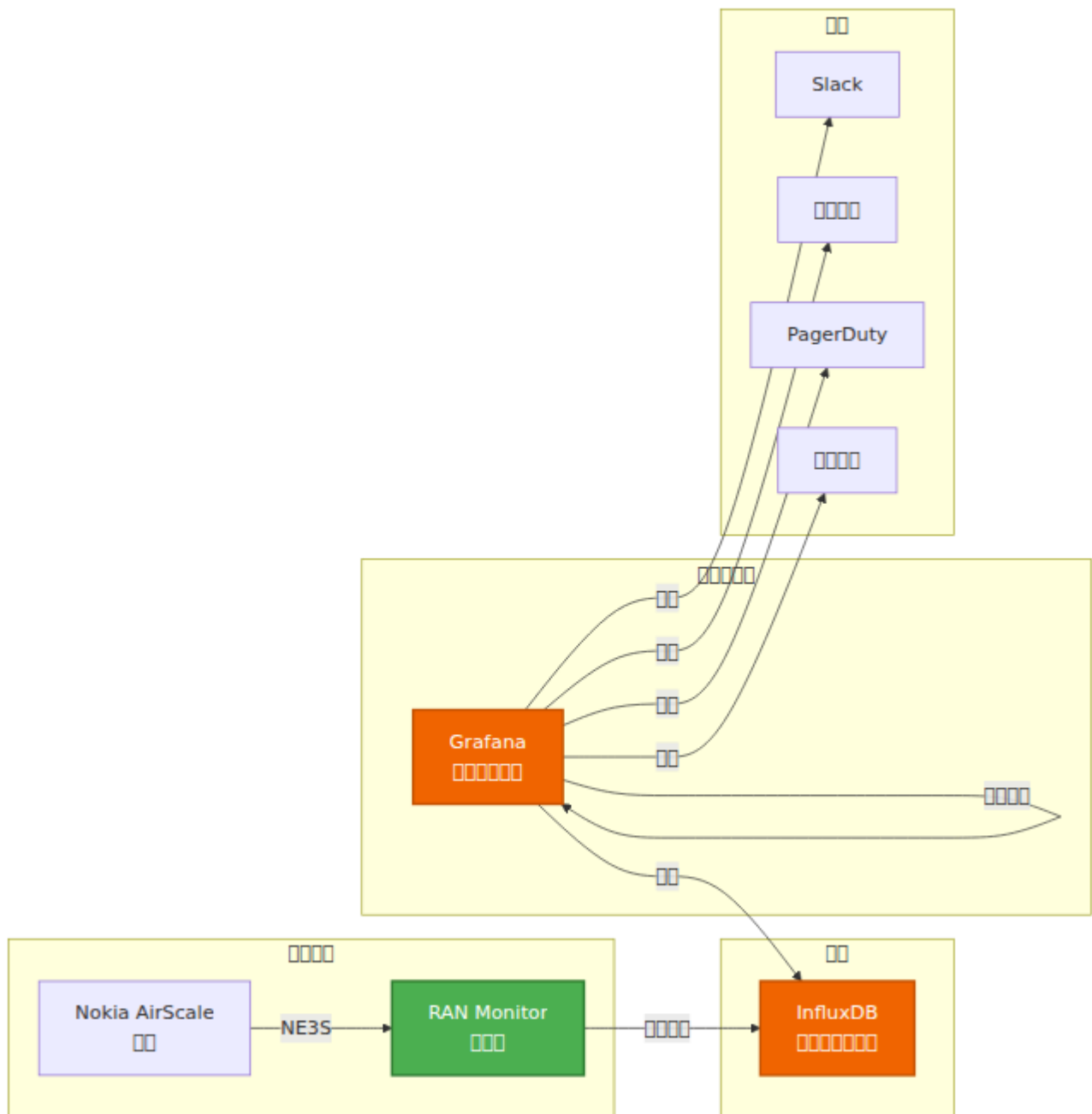
## 目录

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。本文档将详细介绍如何在 RAN 环境中部署 Grafana。

□□□□



## Grafana □□□

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

## 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 のインストールが完了している

### 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	接続先	接続先
接続名	RAN Monitor	Grafana データソース
URL	<code>http://influxdb:8086</code>	接続先 Grafana 接続
接続名	接続先	Grafana データソース
接続名	omnitouch	接続 InfluxDB 接続
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 接続先URL接続先 RAN Monitor 接続 InfluxDB 接続先  
接続 接続 AirScale 接続 接続

# 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 □□)

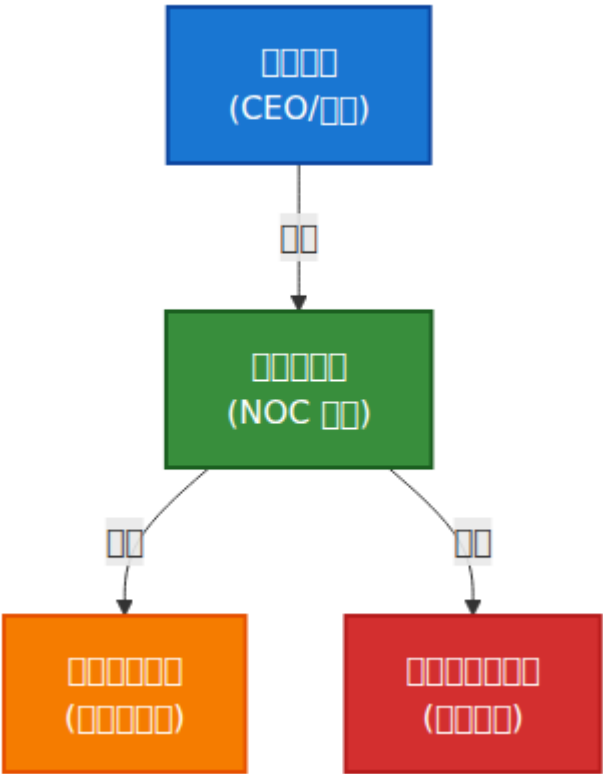
□□

□□

□□□□□□  
(□□□□□)

□□□□□□  
(□□□□)

組織体制



業務内容

製品開発・販売・マーケティング

顧客サポート・営業

製品開発			
開発	テスト	製品化	
48/50 (96%)	3 回	98.5%	
顧客サポート [2000] 営業			

特徴

- 製品開発< 10 年
- 高品質・高信頼性



- [illegible]

□ □ □ □ □ □ □ □ □ □ □ □ □

□ □ □ □ □ □ □ □ □ □ □ □ □

000007 00			
[0000000000/00000]			
000250 Gbps000000 2 00			
00080 Gbps000000 3 00			

111


- 
- 
- 
- 

◆ ◆ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

□ □ □ □ □ □ □

XXXXXXXXXXXX										
XX	XX		XX		XXXX					
X	SITE_A_BS1			XXXX	45	XX				
X	SITE_B_BS2			XX	2	XX				

111

- 
- 
- 

## 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%

パフォーマンス 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



```

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")

```

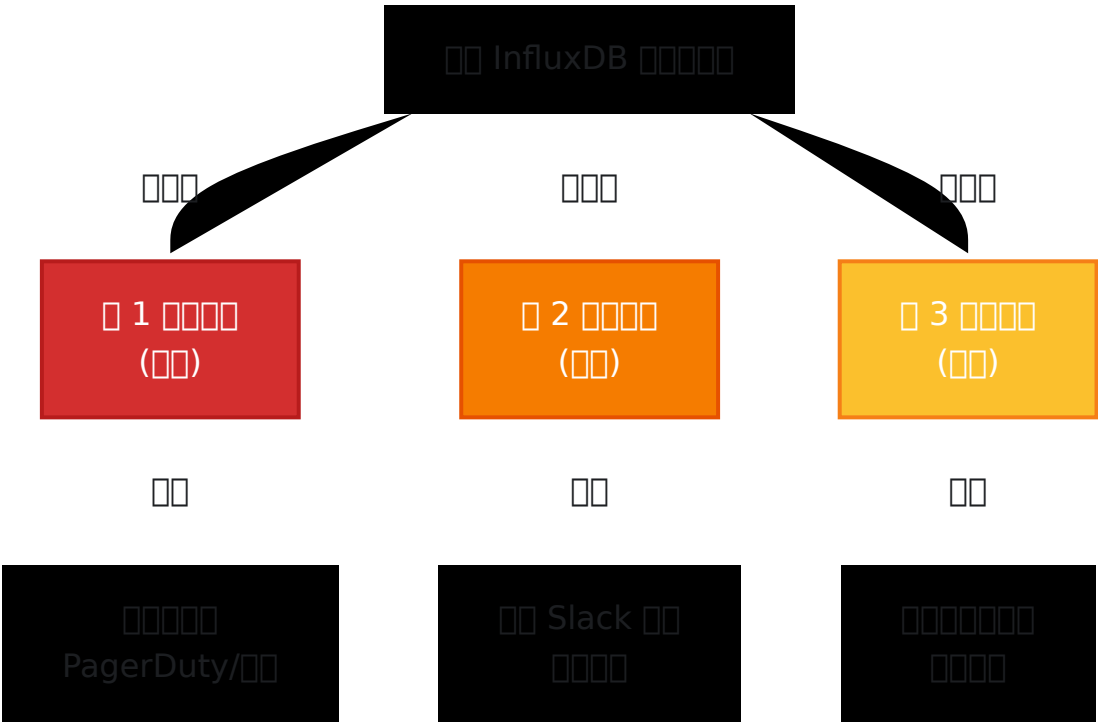
□□□

- □□□□□□□□□□□□□□
- □□□□□□□□

- 監視ツール

監視ツール

監視ツール



## Grafana

### 1

- 1.
- 2.
3. →
4. → →

### 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**

1. `new SlackPagerDuty()`
2. `new SlackPagerDuty()`
3. `new SlackPagerDuty()`

□ □ □ □ □ □ □

```

    [{"AlertRuleName"}]
    [{"Severity"}]
    [{"Labels.device"}]
    [{"EvalMatches[0].Value"}]
    [{"StartsAt"}]

    [{"RuleUrl"}]

```

□ □ □ □

**1** □□□□□□□□

- 24x7 緊急対応体制 SLA あり
- 緊急時 24x7 対応
- 24x7 監視 + 24x7 Slack + PagerDuty
- 24x7 対応体制
- **SLA** < 15 分

**2** □□□□□□□□

- 24x7 緊急対応体制を構築
- 対応時間 15-30 分
- 24x7 Slack + 電話 + PagerDuty
- 24x7 NOC 体制 + 監視
- **SLA** < 30 分

3

- $\frac{1}{n} \sum_{i=1}^n x_i$
- $\frac{1}{n} \sum_{i=1}^n x_i^2$



- 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 **2024** 2024

□□□ □□□□□□□□

□□□□ 10 □





111

1. □□□□□□□□

- ☐ 
- ☐ 

2. □□□□□□□□

- □□□□ %□□□□□
- □□□□ %□□□□□

- CPU %

### 3. 設定項目

- 日/時刻 24 時間
- 日 24 時間
- 時刻

### 4. 設定項目

- 設定項目
- 設定項目

### 5. 設定項目

- IPアドレス/ポート
- 設定項目
- 設定項目

### 6. 設定項目

- 設定項目
- 設定項目
- 設定項目

設定項目

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

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

共 1 測試項目

測試 PRB 測試

```
// 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

参考

```
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"

// 00 PRB 000
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": "0000 PRB 00"})

// 0000 PRB 000
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": "0000 PRB 00"})

```

00 **6**000000000000000000

PDCP 0000000000

```

import "strings"

// PDCH 1000000
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": "PDCH 1000000"})

// PDCH 1000000
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": "PDCH 1000000"})

```

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

9RRC

```

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 00000000 S1 00000000000000000000 PRB 00000000000000000000

00000 - 000000PRB 00000000

XXXXXXXXXX LNCEL LTE PRB XXXX TTI XXXX/XXXXX PDCP XXXXXXXXXXXXXXXXXXXX

□□□□ - **RSSI**□□□□□□ **RRC**□

RSSI [ ]/[ ]/[ ] RRC [ ] VSWR [ ] RMOD [ ]

□□□□□ - □□□□□□□□

```

##### RRC #####VSWR RMOD #####

```

## □□□□ - □□□□□ **VSWR** □□□□

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

--

□ □ □ □ □ □ □ □

111

- □□□□□□□□□□“□□□□”
- □□□□□□□□□□

111

1. 安装 InfluxDB 数据库
2. 配置 InfluxDB 数据库
3. 安装 Prometheus 监控系统
4. 配置 Prometheus 监控系统

□□□□□

- InfluxDB UI 查看数据
- 配置告警 24 小时

- 配置 RAN Monitor 参数
- 配置 RAN Monitor 策略

## 配置 RAN Monitor

### 配置

- 配置 RAN Monitor > 5 个
- 配置 RAN Monitor

### 配置

1. 配置 RAN Monitor > 8 个
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. データ加工
4. データ可視化

## 環境構築

- 環境構築の順序 → 1, 2, 3, 4
- データ収集の順序 → 1, 2, 3, 4
- データ加工の順序 → URL/パス
- Grafana のインストール
- データ可視化の順序

## データ収集

### 環境

- データ収集の順序
- データ加工の順序
- データ可視化

### 環境

1. データ収集
2. データ加工
3. データ可視化/レポート
4. データ可視化の順序

## データ加工

- データ加工の順序 InfluxDB の
- aggregateWindow の順序/パス/パス
- InfluxDB のデータ可視化
- データ可視化の順序

## 目標

### 目的

- 課題の明確化
- 目標の具体化
- 達成のための計画

### 内容

- 1. 課題の抽出と整理
- 2. 目標の設定と優先順位付け
- 3. 達成のための具体的な計画と実行

## 成果

### 目的

- InfluxDB の導入
- 各 eNodeB の稼働状況の監視
- 異常検知とアラート機能の実装

### 内容

- 1. 課題の抽出と整理
- 2. AirScale の稼働状況の監視
- 3. eNodeB の稼働状況の監視

## まとめ

- **Nokia** の AirScale - 5G の基幹設備
- InfluxDB - 時系列データのデータベース
- Prometheus - メトリクス監視システム
- **AirScale** の稼働状況 - 5G の基幹設備

# 無線通信

無線 RAN 技術

無線通信技術の進化

---

## 概要

- 無線通信の重要性
  - 無線通信の歴史
  - 無線通信の技術
  - 無線通信の応用
  - 無線通信の課題
  - 無線通信の未来
  - 無線通信の標準化
  - 無線通信のセキュリティ
  - 無線通信の環境影響
  - 無線通信の社会影響
- 

## 無線通信の技術

無線通信技術の進化は、RAN 技術の進化と NOC 技術の進化によって実現されています。

## 無線通信の応用

- RAN 技術の応用
- NOC 技術の応用
- 無線通信 RAN 技術の応用
- 無線通信 NOC 技術の応用

無線通信技術の進化は、無線通信の未来を切り拓きます。



---

## 目標

確認 Nokia AirScale 設定が正常に完了している

### ステップ 1: 確認

確認   設定が正常に完了している

```
# 確認
ping <base-station-ip>

# 確認
telnet <base-station-ip> 8080
```

確認 ping 確認 telnet 確認

確認

- 確認
- 確認
- 確認

### ステップ 2: 確認

確認

項目	値	説明
IP 番号	10.7.15.67	管理コンソールの IP 番号
ポート	8080	管理コンソールのポート番号 (Nokia AirScale の 8080)
サイト名	Site-B-Tower-1	監視対象のサイト名
ユーザー名	admin	管理コンソールのユーザー名
パスワード	password123	管理コンソールのパスワード

監視対象のサイト

- 監視対象のサイト
- 監視対象のサイト
- 監視対象のサイト
- 監視対象のサイト (NYC-SiteA-BS1, LAX-Tower-Main, CHI-Indoor-DAS)

ステップ 3: 監視対象のサイト

監視対象のサイトの設定

1. Web UI: `https://<ran-monitor-ip>:9443`
2. ユーザー名: **eNodeBs**
3. パスワード: IP 番号と ID
4. サイト名: ID

監視対象のサイトの RAN 番号

ステップ 4: 監視対象のサイト

監視対象のサイトの `config/runtime.exe` を実行し、`airscales` を監視

```

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
```

-

- 配置文件中 { 和 [ 符号
- 配置文件中
- 配置文件中

## 第6章 RAN 配置

配置文件中

```
# 配置文件中
# 按 Ctrl+C 退出
mix phx.server

# 配置文件中
systemctl restart ran_monitor

# 配置文件中
/path/to/ran_monitor/bin/ran_monitor restart
```

## 第7章 部署

配置文件中

### 1. 配置文件中

```
[info] Attempting registration with device: Site-B-Tower-1
[info] Successfully registered with Site-B-Tower-1
```

### 2. Web UI

- 配置文件中
- 配置文件中
- 配置文件中 "Site-B-Tower-1"
- 配置文件中 "Site-B-Tower-1"

### 3. 配置文件中

- 配置文件中

- 設定するデータベース
- 設定 "データベース" を確認する

#### 4. InfluxDB 設定

- InfluxDB 設定
- データベース
- データベースの作成

### 第 8 章

本章では、InfluxDB と Grafana を組み合わせて、

1. InfluxDB のインストール
2. データベースの作成
3. 設定 "データベース" を確認する
4. InfluxDB の起動

本章では、InfluxDB と Grafana を組み合わせて、

### 第 9 章 Grafana 設定

Grafana のインストールと設定

1. Grafana のインストール
2. データベースの接続
3. データベースの作成

本章では、InfluxDB と Grafana を組み合わせて、

### まとめ

本章では、InfluxDB と Grafana を組み合わせて、

## 00 1 000000000000

000000000000000000

### 00 A00000

- 000000000000
- 000000000000000000

### 00 B00000

- 000000000000
- 00 InfluxDB 00
- 000 - 000000

## 00 2 000000000000

00000000000000

00 config/runtime.exs 00 airscales 000000000000000000

```
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 部署前准备

1. 环境准备
2. 部署 Grafana
3. 部署 Prometheus

## 环境准备

部署 Grafana 需要 RAN 环境

### 第 1 步 环境准备

环境准备

1. 安装 Docker 和 Docker Compose
2. 安装 Prometheus
3. 安装 Grafana

### 第 2 步 部署 Grafana

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  
  }  
]
```



## Step 3 RAN Setup

Restart RAN

```
systemctl restart ran_monitor
```

## Step 4 Verification

Check

### 1. Log Verification

```
[info] Attempting registration with device: Site-A-BS1
[info] Successfully registered with Site-A-BS1
```

### 2. Web UI

- Verify "Status" tab
- Check logs
- "Settings" tab

Expected Results

- Status: Online
- Logs: No errors
- Settings: Default
- Performance: Stable

## Conclusion

RAN Setup Complete

## 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
- SLA 監視範囲
- 監視対象範囲

監視対象範囲

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

## 監視 1 監視範囲

監視対象範囲 runtime.exs 監視対象範囲監視範囲監視範囲

監視対象範囲監視範囲

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

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

OmniTouch

## 2

- $\text{ } \times \text{ } \times \text{ } = \text{ }$
- $\text{ } = \text{ }$
- 

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

- $\text{ } = \text{RAN } \text{CPU } \text{ }$
- 

## 3

1.  $\text{ } \times \text{ } = \text{ }$
2.  $\text{ } \times \text{ } = \text{ }$ 
  - RAN  $\text{CPU } \text{ }$
  - 
  - InfluxDB I/O
3.  $\text{ } \times \text{ } = \text{ }$ 
  - InfluxDB

- 無線ネットワーク

4. 無線ネットワーク

- 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

無線 RAN ネットワークの種類

## 無線ネットワーク

無線ネットワークの種類 Web UI 画面 - 無線ネットワーク

無線ネットワーク

1. 無線ネットワークの種類
2. 無線ネットワークの種類
3. 無線ネットワークの種類 - ID
4. 無線ネットワークの種類 ID
5. 無線ネットワークの種類
6. 無線ネットワークの種類

## 無線ネットワーク

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

- 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

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
- 実行時間 15-30 分
- 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

- ログを確認

- 時間系列データベースのインストール
- データの読み込み

### 3. データの取得

- データの取得方法
- データのフィルタリング
- データの集約

### 4. データの可視化

- InfluxDB の可視化
- データの可視化方法
- データの可視化ツール

### 5. データの保存

- データの保存方法
- データの保存形式
- データの保存場所

時間系列データベースのインストール Web UI のインストール - Web UI のインストール

時間系列データベースのインストール **Grafana** のインストール

*S1 LNMME UE PRB*

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

□□□□□□□□□□




## 1. 背景

- Grafana 仪表盘
- 仪表盘
- 仪表盘

2.

- InfluxDB
- MySQL
- 

### 3. 实验结果

- 
- 
- 

4.

- RAN 100% CPU 100%
- 100%
- 100%

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

- 必要なパッケージをインストール
- 環境変数を設定

確認

- インストールが正常に完了しているか確認
- `config/config.exs` ファイルを確認

起動

- サービスを起動し、ログを確認
- 接続テストを実行

トラブルシューティング

- InfluxDB の接続エラー
- 権限エラー
- ネットワークの問題

### まとめ

参考リンク

1. インストールガイド
2. 設定ファイル
3. XML の読み込み



#### 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

- 创建数据源
- 创建仪表盘
- 添加 "Influx" - "CSV" 数据源

## 部署和配置

### 部署

#### 部署

#### 1. 部署

```
# 部署二进制文件
cp config/runtime.exs backups/runtime.exs.$(date +%Y%m%d)

# 部署配置文件
tar -czf backups/config-$(date +%Y%m%d).tar.gz config/
```

#### 2. 配置

- 配置 Web UI 访问地址
- 配置日志输出路径
- 配置数据库连接信息

### 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 证书

```
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. 設定ファイルを確認

5. サービスを再起動

6. InfluxDB に接続を確認

確認事項

1. RAN サービスが正常に起動しているか確認

2. ログを確認

3. MySQL データベースを確認

4. InfluxDB データベースを確認

5. SSL 証明書を確認

6. RAN サービスを確認

7. ネットワーク接続を確認

8. サービスの再起動を確認

## まとめ

### 確認事項

確認

1. サービス

- サービスが正常に起動しているか確認
- ログを確認
- ネットワーク接続を確認

2. サービス

- サービスが正常に起動しているか確認
- ログを確認

- 認證與授權

3. 認證

- 認證與授權
- 認證與授權
- 認證與授權

4. 認證

- 認證與授權
- 認證與授權
- 認證與授權

認證

1. 認證

- 認證與授權
- 認證與授權
- 認證與授權

2. 認證

- 認證與授權
- 認證與授權
- 認證與授權 SSL 認證

3. 認證

- 認證與授權
- 認證與授權
- 認證與授權

認證

1. SSL 認證

- 認證與授權 SSL 認證

- 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
- **Grafana** 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

# 無線 LTE ネットワーク構築

無線 AirScale/FlexiRadio ネットワーク構築

---

## 目次

1. 概要
  2. ネットワーク構成
  3. ネットワーク設計
  4. ネットワーク構築
  5. ネットワーク検証
  6. UE 接続
  7. ネットワーク最適化
  8. ネットワーク監視
  9. ネットワーク保守
  10. ネットワーク QoS 管理
  11. ネットワークセキュリティ
  12. Grafana ネットワーク監視
  13. ネットワークトラブルシューティング
- 

## 概要

無線 LTE ネットワーク AirScale/FlexiRadio ネットワーク構築の概要を説明します。ネットワーク構築の目的は、無線ネットワークの性能を向上させ、ネットワークの信頼性を確保することです。

ネットワーク構築の目的は、無線ネットワークの性能を向上させ、ネットワークの信頼性を確保することです。

ネットワーク構築の目的は、無線ネットワークの性能を向上させ、ネットワークの信頼性を確保することです。❖❖❖

- ネットワーク構築の目的は、無線ネットワークの性能を向上させ、ネットワークの信頼性を確保することです。
- ネットワーク構築の目的は、無線ネットワークの性能を向上させ、ネットワークの信頼性を確保することです。



- 
- 
- 

111

00000000000000000000000000000000 LTE 000000000000

メッセージ (M8xxx)

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

項目	単位	測定範囲	測定項目
M8019	NACC	4	測定範囲
M8020	測定範囲	7	測定範囲
M8021	測定範囲 HO	17	測定範囲
M8022	X2 測定	2	X2 測定
M8023	測定	36	PDCCP SDU 測定範囲

測定範囲 (M5xxx)

項目	単位	測定範囲	測定項目
M5112	IP 測定	112	測定範囲
M5113	測定 RX	21	測定範囲

測定範囲 (M4xxxx)

項目	単位	測定範囲	測定項目
M40001	測定範囲	測定	VSWR測定範囲RF 測定
M40002	測定	測定	測定

測定範囲

測定範囲

M<measurement><type>C<counter>

□□□M8011C24

- **M** - 00000000“00”
- **8011** - 00000000??00
- **C** - 00000000“000”
- **24** - 0000000000 PRB 000000

□□□□

[illegible]

- 00 - 00000000
- 00 - 000000000
- 00 - 00000000
- 00 - 00000000
- 00 - 000000
- 00 - 0000000

--	--	--	--	--

□ □ □ □ □ □ □

□□□□□

- M8011Cxx - PRB □□□
- M8018Cxx - □□ UE □□
- M8012Cxx - □□□□□

□□□□□

- M8001Cxx - □□□□
- M8005Cxx - RSSI□SINR

- M8013Cxx - 000000

000000

- M8020Cxx - 000000
- M8049Cxx - 0000

000000

- M8005Cxx - 00000000
- M8001Cxx - 0000
- M8013Cxx - 0000

0000000000

# M8011 - 000000

000000000000PRB00000000000000000000

000	00	00	00	0	00
M8011C24	00 PRB 0000 TTI 00	00000000000000 PRB 00	0.1%	0	00 10 00 000
M8011C37	00 PRB 0000 TTI 00	00000000000000 PRB 00	0.1%	0	00 10 00 000

00 PRB 0000

- 000000PRB 0 LTE 00000000000000
- TTI00000000 = LTE 00 1 00
- 000000 = 00000000000000
- 100% 00000000000000

□□□□

- 0-1000 0.0% 100.0%
- 10 1000
- 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})) // 1000000
```

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

# M8012 - □□□□□□□□

☐ PDCP

類別	項目	說明	單位	備註	參考
<b>M8012C23</b>	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
<b>M8012C26</b>	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
- > 100 Mbps - 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 配置

```
from(bucket: "nokia-monitor")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> 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
```

UE 設定

M8018 - eNB 設定

以下は、M8018 eNB 設定の UE 設定

UE ID	UE Name	UE Type	UE Category	UE Subcategory	UE Count
M8018C1	M8018 eNB UE ID	M8018 eNodeB UE ID	1	1	1

UE 設定

- UE UE = M8018 eNB UE ID SRB M8018 eNodeB UE ID DRB M8018
- 1 M8018 eNB UE ID
- M8018 eNodeB UE ID

設定



UE 値	測定値	評価
0-50	0	良好
50-100	10	良好
100-150	20	良好
> 150	30	良好

評価

- 測定値が 0 の場合
- 測定値が 10-20 の場合 AirScale 測定値が 150-250 の場合 UE
- 測定値が 30 の場合 UE

## 測定値の取得

### M8020 - 測定値の取得

測定値の取得方法

測定値	測定値	測定値	測定値	測定値	測定値
M8020C3	測定値	測定値	測定値	測定値	~10
M8020C4	測定値	測定値	測定値	測定値	~10
M8020C6	測定値	測定値	測定値	測定値	~10

測定値の取得

測定値 % = 100.0 × M8020C3 / (M8020C6 - M8020C4)

指標

- **M8020C3** - 平均応答時間
- **M8020C6** - 最大応答時間
- **M8020C4** - 最大応答時間の標準偏差

サービスレベル

サービスレベル	指標	計算式
> 99.9%	SLA	平均応答時間 SLA
99.0-99.9%	SLA	最大応答時間
95.0-99.0%	SLA	最大応答時間の標準偏差
< 95.0%	SLA	最大応答時間の標準偏差 - 最大応答時間

Grafana 設定

```
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,
    "サービスレベル": 100.0 * r.M8020C3 / (r.M8020C6 - r.M8020C4)
  })))
```

# 

## 

### 

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

### 

- 
- 
- 
- 

### 

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

RF

- RSSI - RSSI
- RSSI - RSSI
- RF -

RF

M8013 -

RRC

M8013C5		RRC		
M8013C17	MO-S	-		
M8013C18	MT	-		
M8013C19	MO-D	-		
M8013C21				
M8013C31				
M8013C34				
M8013C91	MO-V	-		
M8013C93	MT-	- MT		

RF

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

□□□□□

- **M8013C5** - 00000000 RRC 00000000
- 00000000 - 00000000






□□□□□

- **MO-S (M8013C17)** - 0000000000000000
- **MT (M8013C18)** - 00000000/0000
- **MO-D (M8013C19)** - 0000000000000000
- **MO-V (M8013C91)** - 00000000VoLTE 0000
- **00 (M8013C21)** - 00000091101120

□□□□□

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

□ □ □ □ □ □ □

- 
- 
- 
- 
- 

# QoS

## M8001 -

M8001C2	PDCP SDU	eNB PDCP SDU		

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

< 10		VoLTE
10-20		
20-50		
> 50		

- 
- 
- 
-

# 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
<b>M8000C0</b>	eNodeB 向 MME 註冊	MME 向 eNodeB 註冊	eNodeB 向 MME 註冊	eNodeB 向 MME 註冊
<b>M8000C1</b>	eNodeB 向 MME 註冊	MME 向 eNodeB 註冊	eNodeB 向 MME 註冊	eNodeB 向 MME 註冊
<b>M8000C2</b>	eNodeB 向 MME 註冊	MME 向 eNodeB 註冊	eNodeB 向 MME 註冊	eNodeB 向 MME 註冊
<b>M8000C3</b>	eNodeB 向 MME 註冊	MME 向 eNodeB 註冊	eNodeB 向 MME 註冊	eNodeB 向 MME 註冊
<b>M8000C6</b>	S1 介面	S1 介面	S1 介面	S1 介面
<b>M8000C7</b>	S1 介面	S1 介面	S1 介面	S1 介面
<b>M8000C11</b>	S1 介面	S1 介面	S1 介面	S1 介面
<b>M8000C12</b>	UE 向 S1 介面	UE 向 S1 介面	UE 向 S1 介面	UE 向 S1 介面
<b>M8000C29</b>	NAS 向 S1 介面	NAS 向 S1 介面	NAS 向 S1 介面	NAS 向 S1 介面
<b>M8000C30</b>	NAS 向 S1 介面	NAS 向 S1 介面	NAS 向 S1 介面	NAS 向 S1 介面

### 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

本圖表顯示

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

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

- S1 > 99%
- > 95%

## EPS

### M8006 - EPS

E-UTRAN E-RAB EPS

M8006C0	EPS			
M8006C1	EPS			???
M8006C2-C5				

EPS

- = UE
- =
- = QoS VoLTE

- 
-



- 無線通信の基礎

無線通信

**M8009** - 無線通信

**M8014** - eNB 無線通信

**M8015** - eNB 無線通信

**M8021** - 無線通信

無線通信の基礎——UE 無線通信の基礎

無線通信

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

無線通信

Modell	Modell	Modell
M8009	Modell	Modell
M8014	Modell X2 Modell HO	Modell
M8015	Modell HO	Modell
M8021	Modell HO	Modell

Modell

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

Modell

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

Modell

- Modell
- Modell
- Modell UE Modell
- Modell

Modell

## M8010 - CQI Modell

Modell UE Modell CQI Modell

Modell CQI

- **CQI** = UE → eNodeB 報告値
- 報告 CQI 0~15
- 報告 可能な MCS
- 報告 可能な 送信速度

## CQI 報告値

CQI 値	報告値	送信速度	変調方式
0-3	0	< 1 Mbps	QPSK
4-6	1	1-5 Mbps	QPSK
7-9	2	5-15 Mbps	16-QAM
10-12	3	15-40 Mbps	64-QAM
13-15	4	40-150 Mbps	64-QAM

## M8010 報告値

- M8010C0 - M8010C15 報告値 0-15 CQI 報告値

報告値

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

## 平均 CQI 報告値

- 平均 CQI 10-15 報告値
  - 平均 CQI 7-9 報告値
  - 平均 CQI 0-6 報告値
-

# CS 測試

## M8016 - CS 測試

測試CSFB功能，LTE UE 在2G/3G 測試

### CSFB

- 測試 VoLTE 測試
- 測試 UE 在 LTE → 2G/3G → LTE
- 測試 LTE 測試

測試

測試	測試	測試
M8016C0	CSFB 測試	CSFB
M8016C1	CSFB 測試	CSFB

### CSFB 測試

1. 測試 UE 在 LTE 2G/3G 測試
2. 測試 UE 在 LTE 2G/3G 測試

測試

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

測試

## M8023 - PDCP SDU 測試

測試

## 無線技術

- 無線通信技術の進化
- 4G/LTEから5G/NRへ
- 無線通信の課題と解決策

## 無線技術

- 無線通信の基礎
- 無線通信の規格と標準
- 無線通信のセキュリティ
- QoS 無線通信

## 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 無線技術

## M8004 - X2 無線技術

## M8022 - X2 無線技術

無線通信技術 eNodeB 無線 X2 無線技術 eNB 無線通信技術

## M8004 - X2 無線技術

- 無線通信技術 eNodeB 無線通信技術
- 無線 X2 無線技術

## M8022 - X2 無線技術

- M8022C0無線 X2 無線技術
- M8022C1無線 X2 無線技術

□□ **X2** □□□

- 1000 00 eNodeB 00000000
- 1000 0000000000000000
- 1000 0000000000000000

**X2**

$$X2 \text{ } \square\square\square\square = 100 \times M8022C1 / M8022C0$$

□□□ > 95%

□ □ □ □ □

## M8007 - □□□□□□□□DRB□□□

□□□□□□□□□□□□□□□DRB□□□□□□□□□□

□ □ □ □ □

- DRB [ ] [ ] [ ] [ ] [ ]
- DRB [ ] [ ] [ ] [ ]
- [ ] [ ] [ ] [ ] [ ]

**M8008 - RRC** ☐☐☐☐☐☐

□□□□□□ RRC □□□□□□□□□□□□□□

□□□□□□

- 國際PRB 調查
- 國際 UE 調查
- 國際調查
- 國際調查

# 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 -

<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>
M40001C0	<div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> VSWR	<div></div>	0.1	<div></div>	<div></div> 10

### VSWR

- VSWR** =
- 
- 
- =

### VSWR

VSWR	<div></div>	<div></div>
1.0-1.5	<div></div>	<div></div>
1.5-2.0	<div></div>	<div></div>
2.0-3.0	<div></div>	<div></div>
> 3.0	<div></div>	<div></div> <div></div> <div></div>

### 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
<b>M40002C2</b>	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

### 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 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 - 4G UE
- 4G - 4G RSSI 4G

4G 4G

- 4G

- 网络性能指标
- 网络质量指标

## 网络性能

- PRB 利用率 < 70% 70-85% > 85%
- 网络质量 > 99% 95-99% < 95%
- 网络质量 > 99% 95-99% < 95%

## 网络质量

- 网络性能指标/网络质量
- 网络性能指标/网络质量
- 网络性能指标/网络质量

## 网络性能

- 网络性能 - RAN 网络性能
- Grafana 网络性能 - Grafana 网络
- 网络性能 - RAN 网络
- AirScale 网络性能 - 网络
- 网络性能 - InfluxDB 网络

□□□□□

□□□□□□□

□□□	□□	□□	□□
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 □□□□□	□□□□	□□
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		S1		
M8000C9		S1	MME	
M8000C11		S1		
M8000C12		UE	S1-	
M8000C13		eNB	S1	
M8000C14		MME	S1	
M8000C15		eNB	S1	
M8000C16		MME	S1	
M8000C23		UE		
M8000C24		UE		
M8000C25		UE		
M8000C29		NAS		
M8000C30		NAS		
M8000C31		UE		
M8000C32		IMS	E-RAB	
M8000C33		IMS	E-RAB	
M8000C34		IMS	E-RAB	
M8000C35				
M8000C36				
M8000C37		X2 IP		
M8000C38		X2 IP		
M8000C39		WRITE-REPLACE		
M8000C40		WRITE-REPLACE		
M8000C41		KILL		
M8000C42		KILL		

-----  
M8001

M8001C0		PDCP SDU	□□□	DL DTCH	□□		
M8001C1		PDCP SDU	□□□	DL DTCH	□□		
M8001C2		PDCP SDU	□□□	DL DTCH	□□		
M8001C3		PDCP SDU	□□□	UL DTCH	□□		
M8001C4		PDCP SDU	□□□	UL DTCH	□□		
M8001C5		PDCP SDU	□□□	UL DTCH	□□		
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	000	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		DL DTCH	□□	RLC SDUs					
M8001C133		DL DCCH	□□	RLC SDUs					
M8001C135		UL DTCH	□□	RLC SDUs					
M8001C136		UL DCCH	□□	RLC SDUs					
M8001C137		RLC PDU	□□□□						□□
M8001C138		RLC PDU	□□						□□
M8001C139		RLC PDU	□□						
M8001C140		DL RLC C-PDUs	□□□□□□□□						□□
M8001C141		DL RLC	□□	PDUs	□□□□□□□□				
M8001C142		UL RLC PDUs	□□						
M8001C143		□□□□ UL RLC PDU	□□						
M8001C144		UL RLC PDU	□□□□						□□
M8001C145		UL RLC PDUs	□□						
M8001C146		DL RLC SDU	□□	PDCP					
M8001C147		□□□□□□ UE	□□				□□		□□
M8001C148		□□□□□□ UE	□□				□□		
M8001C150		□□□□□□ UE	□□				□□		□□
M8001C151		□□□□□□ UE	□□				□□		
M8001C153		□□ PDCP SDUs							
M8001C154		□□ PDCP SDUs							
M8001C155		□□□□□□ PDCP SDUs							□□
M8001C156		□□□□ PDSCH MCS0	□□						□
M8001C157		□□□□ PDSCH MCS1	□□						□
M8001C158		□□□□ PDSCH MCS2	□□						□
M8001C159		□□□□ PDSCH MCS3	□□						□
M8001C160		□□□□ PDSCH MCS4	□□						□
M8001C161		□□□□ PDSCH MCS5	□□						□
M8001C162		□□□□ PDSCH MCS6	□□						□
M8001C163		□□□□ PDSCH MCS7	□□						□
M8001C164		□□□□ PDSCH MCS8	□□						□
M8001C165		□□□□ PDSCH MCS9	□□						□
M8001C166		□□□□ PDSCH MCS10	□□						□
M8001C167		□□□□ PDSCH MCS11	□□						□
M8001C168		□□□□ PDSCH MCS12	□□						□
M8001C169		□□□□ PDSCH MCS13	□□						□
M8001C170		□□□□ PDSCH MCS14	□□						□
M8001C171		□□□□ PDSCH MCS15	□□						□
M8001C172		□□□□ PDSCH MCS16	□□						□
M8001C173		□□□□ PDSCH MCS17	□□						□
M8001C174		□□□□ PDSCH MCS18	□□						□
M8001C175		□□□□ PDSCH MCS19	□□						□
M8001C176		□□□□ PDSCH MCS20	□□						□
M8001C177		□□□□ PDSCH MCS21	□□						□

M8001C178		□□□	PDSCH	MCS22	□□		□
M8001C179		□□□	PDSCH	MCS23	□□		□
M8001C180		□□□	PDSCH	MCS24	□□		□
M8001C181		□□□	PDSCH	MCS25	□□		□
M8001C182		□□□	PDSCH	MCS26	□□		□
M8001C183		□□□	PDSCH	MCS27	□□		□
M8001C184		□□□	PDSCH	MCS28	□□		□
M8001C185		□□□	PDSCH	MCS29	□□		□
M8001C186		□□□	PDSCH	MCS30	□□		□
M8001C187		□□□	PDSCH	MCS31	□□		□
M8001C188		□□□	PDSCH	MCS32	□□		□
M8001C189		□□□	PDSCH	MCS33	□□		□
M8001C190		□□□	PDSCH	MCS34	□□		□
M8001C191		□□□	PDSCH	MCS35	□□		□
M8001C192		□□□	PDSCH	MCS36	□□		□
M8001C193		□□□	PDSCH	MCS			

# PM 環境構築

## 概要

PM 環境構築には InfluxDB と PM 環境構築 AirScale が必要で **22,000** 円  
の **PM** 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

### PM 環境構築

- 環境構築 `https://localhost:9443`
- 環境構築 環境構築
- 環境構築 PM 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 <b>PM</b> 環境構築	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 22,000+ 環境構築環境構築

## 環境構築

PM 環境構築環境構築環境構築環境構築

種別	識別子	数値	説明
<b>LTE</b>	M8xxx	~5,900	LTE L1/L2/L3 間のERABとRRC間の
<b>WCDMA</b>	M5xxx	~885	3G WCDMA 間のMAC とCQIとHSDPA
<b>5G-NR</b>	M55xxx	~14,500	5G NR 間のMIMO間の
<b>5G-4G</b>	M51xxx	~500	5G 間の
<b>5G-3G</b>	M40xxx	~250	5G 間の/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/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

#### インストールの確認

- 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)

### データ取得

データ取得のパスは `priv/pm_counters.csv` に指定されている。

## 設定

#### PM の設定

1. Docker PM の `priv/pm_filters.etf`
2. Docker PM の起動
3. Docker PM の停止

InfluxDB データベースの作成/接続

---

## 概要

### システム構成

サーバ 22,000+ 台を管理

項目	内容
ストレージ	SSD ~100-500 GB/台を管理
データベース	MySQL InnoDB 管理
ログ	ログ収集・分析
監視	システム監視

### 課題

- 大量のログデータを効率的に管理・分析する
- システム全体の稼働状況をリアルタイムで監視する
- 障害発生時の迅速な対応を実現する

---

## 解決策

### 導入した技術

“クラウド”環境での導入を実現

- サーバ ID 管理 M8012C23
- 監視・ログ収集 RSRP



PM counters

PM Counter	Radio Access Technology	Description
M8012C23	LTE	Uplink data volume
M8012C26	LTE	Downlink data volume
M8001C2	LTE	PDCP SDU DL volume
M8011C24	LTE	Uplink PRB volume
M8011C37	LTE	Downlink PRB volume
M8013C17	LTE	RRC DL volume
M8020C3	LTE	Uplink PRB volume
M40001C0	5G	Uplink data volume

PM counters

pm\_counters.csv

PM counters

```
# PM counters, radio, desc
M8012C23, radio, uplink data volume
M8012C26, radio, downlink data volume
M8001C2, radio, PDCP SDU DL volume
...
```

PM counters `priv/pm_counters.csv`

## pm\_metrics.csv

□ □ □ □ □ □ □ □ □ □ □ □

```
# 000PM_Code,00,00
M8000C6,LTE,S1_SETUP_ATT
M8000C7,LTE,S1_SETUP_SUCC
...
```

```
priv/pm_metrics.csv
```

--	--	--	--

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

1. 設置環境“PM 管理”環境
2. 各 eNodeB 環境に PM 環境 InfluxDB 環境
3. 環境 ID 環境環境環境環境

□ □ □ □ □

1. 環境構築
2. 環境構築 **eNodeB** 環境構築 **PM** 環境構築 15 分程度
3. 環境構築 **[PmFilterStore]** 環境構築
4. 環境構築

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

1. `priv/pm_metrics.csv`
2. `priv/pm_metrics.csv`
3. `UTF-8`

## □□□□

- **□□□□□□** - 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

配置 `config/runtime.exs`

---

## 部署

1. 部署
  2. 部署
  3. Web 部署
  4. 部署
  5. 部署
  6. InfluxDB 部署
  7. 部署
- 

## 部署

`config/runtime.exs` 配置 RAN 部署环境

部署

- MySQL
- Web 部署
- 部署
- InfluxDB 部署
- 部署
- 部署

部署

`config/runtime.exs`

---

## インストール

まず RAN をインストール **Omnitouch** をインストールします。

- インストール
- インストールオプション
- インストール場所
- インストール方法

次に **Omnitouch** をインストール **Omnitouch** をインストールします。

インストール完了後、[インストールガイド](#) **Grafana** を

---

## インストール

### 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
```

次に、インストール完了後、MySQL をインストール

## インストール

**username** (インストール)

- インストール
- インストール "omnitouch"
- インストール CREATE、SELECT、INSERT、UPDATE、DELETE

- 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 "omnitech2024"
- 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`
- must be a valid port number
- must be a valid IP address

## check\_origin (bool)

- check WebSocket/HTTP origin
- default `false`
- must be a valid SOAP API `false` for Web UI

## secret\_key\_base (string)

- secret key
- must be 64 characters
- default `mix phx.gen.secret`
- must be a valid secret key
- must be a valid secret key

## server (bool)

- server mode
- default `true`
- must be a valid runtime.exs `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 で

設定

- **CONTROL\_PANEL\_HTTPS\_PORT** - HTTPS ポート 9443
  - 設定する HTTPS ポート
  - `export CONTROL_PANEL_HTTPS_PORT=8443`

**url** (設定)

- URL
- **host:** `"0.0.0.0"` - ホスト
- **port:** `https_port` `CONTROL_PANEL_HTTPS_PORT`
- **scheme:** `"https"` - HTTPS

**https** (設定)

- HTTPS
- **ip:** `{0, 0, 0, 0}` - IP
- **port:** `https_port` url
- **keyfile:** SSL
- **certfile:** SSL

## 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 (ポート)

- 9076
- PMCADM rTpmCollEntityPortNum
- WebLM

## ログ

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

## options

### level (int)

- 0: 0 (default)
- 1: `:info`
- 2: 2
  - `:debug` - 0 (default)
  - `:info` - 0 (default)
  - `:warning` - 0 (default)
  - `:error` - 0 (default)

### format (string)

- 0: `:debug` - 0 (default)
- 1: `:info` - 0 (default)
- 2: 2
  - `:debug` - 0 (default)
  - `:info` - 0 (default)
  - `:warning` - 0 (default)
  - `:error` - 0 (default)

## format

### format (string)

- 0: 0 (default)
- 1: `"$time $metadata[$level] $message\n"`
- 2: 2
  - `$time` - 0 (default)
  - `$metadata` - 0 (default)
  - `$level` - 0 (default) info error
  - `$message` - 0 (default)

### metadata (string)

- 0: 0 (default)
- 1: `[:request_id]`
- **request\_id**: 0 (default) HTTP

## 配置

配置 RAN 配置

```
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"
      }
    ]
  }
}
```

## 配置

**mcc** (配置)

- 配置
- 配置 "505"
- 配置 3GPP 配置
- 配置 3 配置

- 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)

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

### private\_key (文字列 - 文字列)

- 文字列
- 範囲 priv/external/nokia/ne.key.pem
- PEM 形式
- 範囲 文字列
- 範囲 OpenSSL 範囲

### public\_key (文字列 - 文字列)

- 文字列
- 範囲 priv/external/nokia/ne.cert.der
- DER 形式
- 範囲 文字列
- 範囲 private\_key 範囲

### reregister\_interval (整数)



- 設定 "8080"
- 設定 AirScale 設定 8080
- 設定 設定
- 設定 設定

## web\_username (設定)

- WebLM 設定
- 設定 "Nemuadmin"
- 設定 API 設定
- 設定 設定/設定
- 設定 設定

## web\_password (設定)

- WebLM 設定
- 設定 "nemuuser"
- 設定 設定
- 設定 設定

## 設定

設定 aircscales 設定

```

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"
```

☐ ☐ InfluxDB ☐



## auth

### auth (auth)

- InfluxDB 认证
- **username:** InfluxDB 用户名 "monitor"
- **password:** InfluxDB 密码 "sideunderTexasgalaxyview\_61"
- 注意 InfluxDB 2.x 使用 API 认证

### database (数据库)

- InfluxDB 数据库/桶
- 名称 "nokia-monitor"
- **InfluxDB 1.x** 使用数据库
- **InfluxDB 2.x** 使用桶
- 注意 InfluxDB RAN 使用数据库

```
# 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"
- 注意 IP 地址
- 注意 InfluxDB 端口 8086
- 注意
  - "localhost" - 本地 RAN 使用数据库
  - "10.179.2.135" - 远程 InfluxDB 使用
  - "influxdb.example.com" - 远程

## InfluxDB 认证

### 认证

- 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 設定

#### 環境変数

##### 1. 環境変数

- IP 設定
- RAN 接続
- 環境変数
- 環境変数

##### 2. 環境変数

- `debug` 環境変数
- `info` 環境変数
- 環境変数

##### 3. RAN 設定

- 環境変数
- 環境変数
- InfluxDB 接続
- 環境変数

#### 環境

##### 1. 環境

- `runtime.exe`
- 環境変数
- 環境変数
- RAN 設定

## 2. 設定

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

# 2. 設定 Elixir 環境
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
- 150+ 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

#
=====
#   配置数据库
#
=====

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, # 是否显示敏感数据
  pool_size: 15 # 6 进程 * 2 + 3 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

```

```

#
=====

# 配置日志
#
=====

config :logger,
  level: :info # 配置日志级别

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

#
=====

# 配置Nokia
#
=====

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 - 塔
      %{
        address: "10.7.15.66",
        name: "Site-A-Main-Tower",
        port: "8080",
        web_username: "admin",
        web_password: System.get_env("BS_SITE_A_PASSWORD") || "password"
      },

```

```
# 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

- [部署](#) - 部署
- [AirScale 部署](#) - 部署
- [部署](#) - 部署
- [Grafana 部署](#) - 部署
- [API 部署](#) - REST API 部署
- [部署](#) - 部署

# MDT 配置 TCE

## 配置 (TCE)

RAN Monitor 通过部署在 LTE/5G 网络侧的 RF 接口

## 配置 TCE

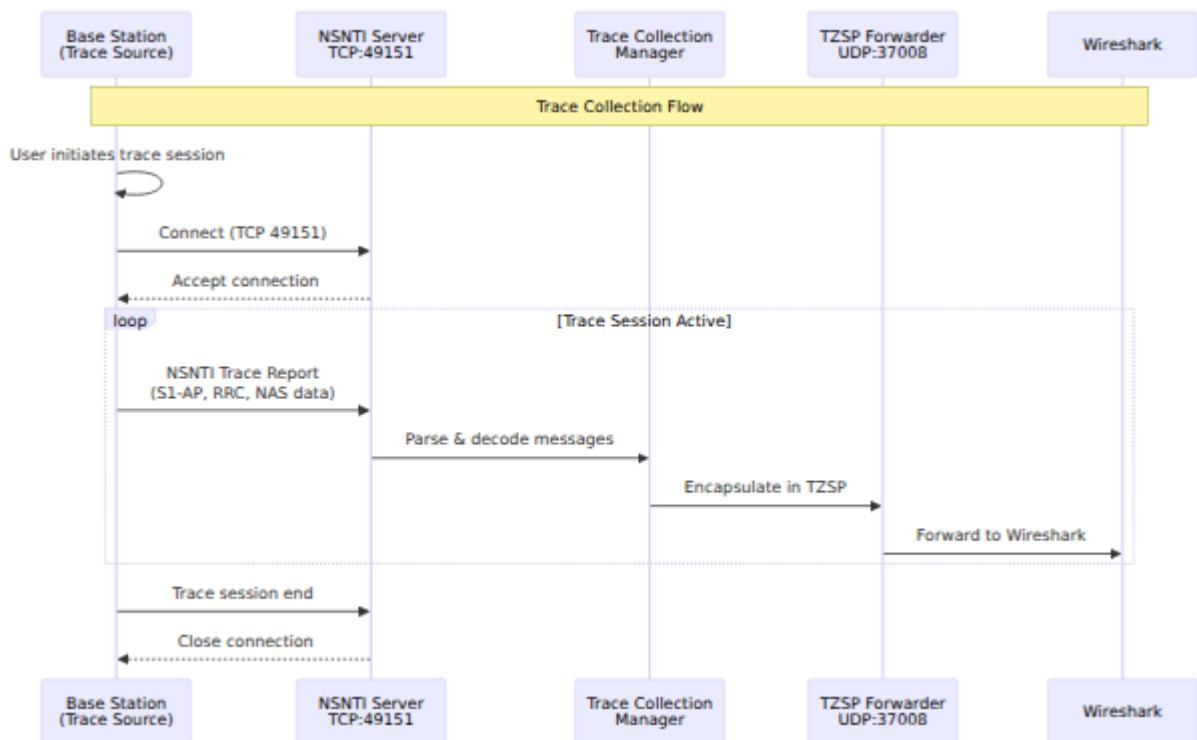
通过部署在 Nokia AirScale 网络侧

- **S1-AP** 接口 - eNodeB 与 EPC 接口
- **RRC** 接口 - 无线资源控制
- **NAS** 接口 - 网络接入层
- 网络层 - PDCP 接口

## TCE 配置

名称	协议	端口	说明
NSNTI 接口	TCP	49151	网络侧接口
TZSP 接口	UDP	37008	通过 Wireshark 抓包
网络层	ASN.1	-	接口 S1-AP 与 RRC 接口

□□□□



□□□□□□□□□□□□□□□□ 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**

- **Analyze → Decode As...**
- + □□□□□□□□
-

項目	値	項目	値	説明
udp.port	36412	項目	(none)	SCTP
sctp.port	36412	項目	(none)	S1AP
udp.port	37000	項目	(none)	TZSP
udp.port	37001	項目	(none)	LTE RRC (DL-CCCH)
udp.port	37002	項目	(none)	LTE RRC (DL-DCCH)
udp.port	37003	項目	(none)	LTE RRC (BCCH)
udp.port	37004	項目	(none)	LTE RRC (PCCH)
udp.port	37008	項目	(none)	TZSP
udp.port	37011	項目	(none)	LTE RRC (UL-CCCH)
udp.port	37012	項目	(none)	LTE RRC (UL-DCCH)
udp.port	38000	項目	(none)	MAC-LTE
udp.port	38001	項目	(none)	MAC-LTE (DL)
udp.port	38002	項目	(none)	MAC-LTE (BCH)
udp.port	38003	項目	(none)	MAC-LTE (PCH)
udp.port	38011	項目	(none)	MAC-LTE (UL)
udp.port	38012	項目	(none)	MAC-LTE (RACH)

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	계층	채널/신호	설명
<b>36412</b>	S1AP	-	S1AP 인터페이스 (eNodeB ↔ EPC)
<b>37000</b>	RRC	신호	RRC 신호 채널
<b>37001</b>	RRC	DL-CCCH	DL-Common Control Channel
<b>37002</b>	RRC	DL-DCCH	DL-Dedicated Control Channel
<b>37003</b>	RRC	BCCH-DL-SCH	Broadcast Channel (下行)
<b>37004</b>	RRC	PCCH	Paging Control Channel
<b>37008</b>	TZSP	-	TZSP 신호 채널
<b>37011</b>	RRC	UL-CCCH	UL-Common Control Channel
<b>37012</b>	RRC	UL-DCCH	UL-Dedicated Control Channel
<b>38000</b>	MAC-LTE	신호	MAC-LTE 신호 채널
<b>38001</b>	MAC-LTE	신호	MAC-LTE 신호 채널
<b>38002</b>	MAC-LTE	BCH	Broadcast Channel
<b>38003</b>	MAC-LTE	PCH	Paging Channel
<b>38011</b>	MAC-LTE	신호	MAC-LTE 신호 채널
<b>38012</b>	MAC-LTE	RACH	Random Access Channel

신호 채널 목록

```
# 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
- 0000000
- 0000000

**RF** 000

- PCI 00000
- 0000000

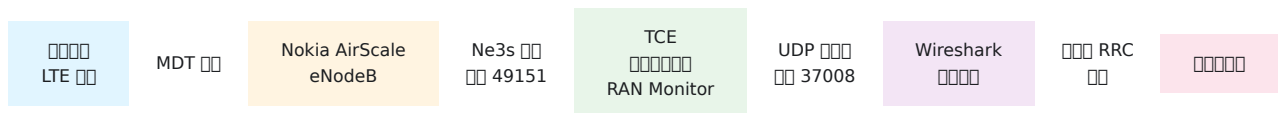


- 設定ファイル
- 実行コマンド

## 概要

本リポジトリは **(MDT)** において UE からの RSRP/RSRQ (RSRP/RSRQ) を取得するためのツールです。  
 本リポジトリは Omnitouch RAN Monitor Web UI にて Nokia AirScale から MDT を取得し、  
 Wireshark で解析します。

## 構成



TCE RAN Monitor は Omnitouch RAN Monitor から Nokia Ne3s を経由して  
 Wireshark で解析します。

## インストール

### 依存関係

Nokia Airscale のインストールは、以下のリンクを参照してください。

インストールは、以下のリンクを参照してください。ONS

### 実行

- Omnitouch RAN Monitor に TCE をインストール
- Wireshark 3.0 をインストール
- Wireshark に TCE Lua スクリプト (TCE README) を追加
- AirScale を実行

# MDT

RAN Monitor TCE Nokia Wireshark

## 1

RAN Monitor Web UI TCE

- Web UI: `https://<ran-monitor-ip>:9443`
- 
- 
- 
- 
- TCE
  - IP: `<RAN Monitor IP>`
  - : `49151`
- 
- 
- 

AirScale ONS

## 2 AirScale MDT

MDT

- MDT
- PLMN
- UE 5000ms
- RSRP RSRQ
- 

ONS

## 3. 실험 환경

이 실험은 AirScale 장비와 MDT 소프트웨어를 TCE에 연결하여 수행됩니다.

## Wireshark를 사용하여 MDT 분석

### Wireshark 설치

1. Wireshark 다운로드
2. Linux 또는 macOS에서는 `lo`, Windows에서는 `lo0` Loopback 인터페이스를 선택합니다.
3. 필터를 `udp port 37008`로 설정합니다.
4. 캡처 시작

Wireshark에서 S1AP 프로토콜 (InitialUEMessage, Attach request)과 LTE RRC (RRCConnectionReject, RRCConnectionReestablishment) 메시지를 TCE에서 캡처합니다.

### MDT 데이터

MDT 데이터를 수집합니다.

```
# 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

- 000000000000 RSRP 0 RSRQ
- 000000000000 RSRP 0 RSRQ
- 00 **ID**000000000000 ID
- **GPS** 000000000000 UE 00

0 Wireshark 000 RRC 000000000000

```
000000 (RRC)
└─ UL-DCCH-Message
    └─ message: measurementReport
        └─ MeasurementReport
            └─ measResults
                └─ measResultServCell (0000 RSRP/RSRQ)
                    └─ measResultNeighCells (000000)
```

00000000

00000000

1. 00 → 00000000 → 00 **CSV**
2. 000000 `lte-rrc.rsrpResult` `lte-rrc.rsrqResult` `lte-rrc.physCellId`

### 3. ExcelPython 環境構築

#### 環境構築

環境構築 RSRP/RSRQ 取得

```
lte-rrc.rsrpResult < 40 || lte-rrc.rsrqResult < 22
```

環境構築 UE 取得

```
lte-rrc.MeasResultListEUTRA
```

環境構築 RSRP 取得 RSRQ 取得

```
lte-rrc.rsrpResult > 50 && lte-rrc.rsrqResult < 20
```

#### 環境構築

**Wireshark** 環境構築

- TCE 環境構築 `ps aux | grep beam`
- Wireshark 環境構築 `udp port 37008` 取得
- AirScale 環境構築
- 環境構築 TCE IP/ポート取得

環境構築

- 環境構築 (MDT + 取得)
- 環境構築
- UE 取得 MDT (LTE Release 10+)

環境構築 **AirScale** 環境構築 **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 繁體 (ONS) 簡體** AirScale 繁體簡體

# 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
- 監視項目
- 監視周期
- 閾値

4. 通知

- 通知手段
- 通知対象
- 通知内容
- 通知頻度

まとめ

環境構築

- RAN 監視 (ps aux | grep ran\_monitor)
- データベース (MySQL, InfluxDB)
- 監視ツール
- 通知ツール

監視対象

監視項目

監視

- Web UI 画面



- 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
- 確認確認

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 環境を構築

環境構築

### 1. 環境構築

Web UI → 監視

- 環境構築
- 環境構築
- 環境構築

### 2. 既存 InfluxDB 環境

Web UI → InfluxDB 環境

- 環境構築
- RAN 環境を InfluxDB へ

環境構築

```
# 既存 RAN 環境を  
curl http://<influxdb-host>:8086/ping
```

### 3. 環境構築

概要

- [error] InfluxDB write failed → 環境構築
- [error] Failed to collect metrics → 環境構築
- [info] Metrics collected: 0 → 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. ディスク容量
3. ディスク容量
4. RAN 環境構築

### 環境構築

1. ディスク容量
2. ディスク容量 webhook URL 設定
3. ディスク容量
4. RAN 環境 webhook 環境構築 9076

### 環境構築

### 環境構築

### 環境構築

- Grafana 環境構築
- ディスク容量
- InfluxDB 環境構築

### 環境構築

### 1. ディスク容量

### 環境構築

```
# ディスク容量
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
- インストール完了

⑧

- インストール完了
- インストール
- インストール完了
- インストール完了

## InfluxDB ⑨

- インストール完了
- ⑧ RAN インストール完了
- ④ InfluxDB ⑤
- インストール完了

⑩

- ⑧ RAN インストール完了



- 設定可能な項目
- InfluxDB 接続
- 設定可能な項目/項目

# Web UI

## Web UI

- 接続 URL `https://<ran-monitor-ip>:9443`
- 接続ポート
- SSL 接続

### 1. Web UI 接続

```
[info] Running ControlPanelWeb.Endpoint with cowboy
```

```
ps aux | grep control_panel  
netstat -tulpn | grep 9443
```

### 2. 接続

```
telnet <ran-monitor-ip> 9443
```

RAN 接続

```
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. 確認する
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

□□□□□

□□□□□□

1. □ `config/runtime.exs` □□□□□
2. □□ RAN □□□
3. □□□□□□ Web UI □

□□□□□□□□

1. □□ MySQL □□□□□□
2. □□ `config/runtime.exs` □□□□□□
3. □□□□□□□□
4. □□ RAN □□□

□□□□□□

□□□MySQL □□□□

□□□

- □□□□□□□□□□□□□□□□
- Web UI □□□□□□□□□□
- “□□□□□□□□”□□

□□□□□

1. □□ MySQL □□□□□□

```
systemctl status mysql
# □
systemctl status mariadb
```

2. □□□□

□ RAN □□□□□□□□

```
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

## MySQL 配置

### 1. 安装 MySQL 服务

```
systemctl start mysql
```

### 2. 配置 MySQL 服务

### 3. RAN 配置

配置

### 1. 配置 MySQL 服务

### 2. 配置

### 3. 配置 `config/runtime.exs`

### 4. 配置 RAN 服务

配置

### 1. 配置 MySQL 服务

### 2. 配置 3306

### 3. 配置 MySQL 服务

配置

1. MySQL max\_connections
  2. pool\_size
  3. RAN
- 

## MySQL

### MySQL CPU

MySQL

- RAN CPU RAM
- 
- 
- 

MySQL

#### 1. MySQL

```
# CPU
top -p $(pgrep -f ran_monitor)

#
ps aux | grep ran_monitor
```

#### 2. MySQL

MySQL

- =
- 

#### 3. MySQL

MySQL

- $\text{CPU} = \frac{\text{CPU}}{\text{CPU}}$
- $10 \text{ } \frac{\text{CPU}}{\text{CPU}}$

4.

```
config/runtime.exs
```

```
pool_size: 10 # [] [] [] [] []
```

□□□□□

□ □ □ □ □ □ □ □ □ □ □ □

1. 環境構築
2. CPU/RAM
- 3.
- 4.

□□□□□□□□

1. pool\_size
2. 2 + 5 Web UI
3. RAN
- 4.

555

1. 00000000
2. 0000000000000000
3. 00000000 RAN 000
4. 000000000000

## InfluxDB ☐☐☐☐☐

1. ☐ InfluxDB ☐
2. ☐ InfluxDB ☐
3. ☐ InfluxDB ☐
4. ☐

# 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

接続

- MySQL の接続
- MySQL の接続
- MySQL の接続

環境構築

1. MySQL のインストール

MySQL のインストール

2. Webhook の設定

Webhook の設定

```
netstat -tulpn | grep 9076
```

環境

```
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. 配置 RAN 端口 9076
2. 配置
3. 配置 `telnet <ran-monitor-ip> 9076`

## InfluxDB 配置

1. 配置 InfluxDB 配置
2. 配置
3. 配置 InfluxDB 配置
4. 配置

## 配置

## 配置

## 配置 Web UI 配置

1. 配置
2. 配置
3. 配置
4. 配置

## 配置

## 配置 systemd 配置

```
journalctl -u ran_monitor -f
```

## 配置 mix 配置

- 配置

## 配置

- 配置/配置/配置 - 配置

- 10 - 10000000
- 10 - 10000
- 10 - 10000000
- 10 - 10000000

10000000

- 100000000 "Site-A-BS1"
- "error" 10 "failed"
- "InfluxDB" 10 "MySQL"
- "registration" 10 "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

### 1. 簡介

- 系統架構
- 系統功能
- 系統部署

### 2. 系統架構

- 系統架構圖
- 系統組件
- 系統數據流

### 3. 系統功能

- RAN 系統功能
- 系統功能圖
- 系統功能描述MySQL-InfluxDB
- 系統功能測試

### 4. 系統部署

- 系統部署圖
- 系統部署步驟
- 系統部署測試

### 5. 結論

- 系統總結
- 系統展望
- 系統建議

## 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
- 監視項目 - メモリ使用率

## 前提条件

### 環境

- Web UI 構築環境
- 監視対象 `journalctl -u ran_monitor`
- 監視項目

### 監視項目

- CPU使用率
- メモリ使用率/自由
- 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
- 監視項目構築
- InfluxDB 構築
- 監視画面構築

## 準備

### 監視対象構築

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
    - 安裝環境與軟體
    - 安裝
- 

## 安裝

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



# Web UI 00

**RAN** 000000 - 000000

00 RAN Monitor 00 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. Web UI 0000
- 

00







RAN Monitor 00000000 Web 000000000000000000Web UI 000000000000000000  
000000

# Web UI Grafana

## Web UI

- 
-    
- 
- 
- 

## Grafana


- 
-  KPI 
- 
- 
- 




 Grafana   Grafana 

---

# Web UI

 HTTPS 

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

: 9443 `CONTROL_PANEL_HTTPS_PORT` 

## SSL

- 
-  CA 
-  `config/runtime.exs` 

□□□□ □□□□ 5 □□□□□□□□□□□□

---

□□□□

□□□□ RAN □□□□□□□□

□□□□

□□□□

- □□□□□
- □□□□□□□□□□



□□□□

- □□/□□□□□□
- □□□□□
- □□□□□□□

□□□□

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

□□□□

- □□□□□□□
- □□□□
- □□□□□

□□

- □ 5 □□□□□
- □□□□□□□□□□□ = □□□□□ = □□□
- □□□□□□□□□

- 

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

□ □ □ □

□□□□□□□□□□

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
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”項目
項目	項目 IP 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	説明
<b>Ping</b>	接続確認時間
IP	接続先IP
ポート	接続先ポート番号
プロトコル	接続先プロトコル

## 接続確認

接続確認結果

項目	説明
接続 ID	接続識別子
IP ID	IP識別子
ポート	接続先ポート番号COMA
プロトコル	接続先プロトコルNokia

## 接続確認

- 接続確認結果
- 接続確認 IP 接続
- 接続確認

## 接続確認

接続確認結果

□□□□□□

- □□□□□□□□
- □□□□
- □□□□□□
- □□□□□□□

□□□□□□

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

□□□□

- □□□□□□□
- □□□□□□□
- □□□□□
- □□□□□

□□□□

- □□□□□□□□
- □□□□□□□
- □□□□
- □□□□□□

□□□□

- □□□□
- □□□□□□
- □□□□
- □□□□□

---

## □□□□

□□□□□□□□□□□□□□□□

## □◆?◆?□□

□□□□□□

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

□□□□□□□□

- □□□□
- □□□□□□
- □□□□□□□□DN - Distinguished Name□
- □□□□□□□□□□□□□□□□□□

## □□

□□□□□□

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

□□□□□□□□

- □□□□□□□□□□□□



- 数据源
- 数据源

数据源

- 数据源
- 数据源
- 数据源

数据源

数据源

Web UI 数据源

数据源

数据源

数据源

1. 数据源
2. 数据“数据源”
3. 数据源
4. 数据源 XML 数据

数据源 数据源

数据源

数据源

数据源

1. 数据 XML 数据
2. 数据“数据源”

3. 註冊商標“三”

4. 註冊商標 ID

註冊商標 - 註冊商標

註冊商標

註冊商標

註冊商標

1. 註冊商標 ID

2. 註冊商標“三”

3. 註冊商標

4. 註冊商標

註冊商標 - 註冊商標

註冊商標

註冊商標

註冊商標

1. 註冊商標 ID

2. 註冊商標“三”

3. 註冊商標

4. 註冊商標/註冊商標

註冊商標 - 註冊商標

註冊商標

註冊商標

1. 註冊商標

2. 註冊商標

3. 註冊商標 ID

4. 配置 eNodeBs
5. 配置 MME
6. 配置 HSS
7. 配置 DNS

配置 eNodeBs 使用 **AirScale** 设备

## 配置 eNodeBs

配置 eNodeBs 的步骤如下：

1.

配置 eNodeBs 的 IP 地址

- 配置 eNodeBs 的 IP 地址
- 配置 eNodeBs 的 DNS 服务器
- 配置 eNodeBs 的 NTP 服务器
- 配置 eNodeBs 的日志级别

2.

配置 ID

- 配置 eNodeBs 的 ID
- 配置 eNodeBs 的 ID 范围

3.

- 配置 eNodeBs 的 IP 地址
- 配置 eNodeBs 的 DNS 服务器

4.

- 配置 eNodeBs 的 IP 地址
- 配置 eNodeBs 的 DNS 服务器

## 環境

- 仮想マシン環境
- Windows 10

## 前提条件

### OS

- Windows 10
- Windows 10

### OS

- Windows 10
- Windows 10/Windows 11

## 環境構築

- Windows 10
- Windows 10

## 環境構築

環境構築の手順

1. 仮想マシン ID
2. 仮想マシン `config/runtime.exs`

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

### 3. RAN Monitor

Network Performance Monitoring

Key

- Network Performance
- Network Quality
- Network Capacity
- Network Security

Network Performance

Network Quality

Network Capacity

Network Security

- Network Performance
- Network Quality
- Network Capacity
- Network Security
- Network Performance
- Network Quality
- Network Capacity
- Network Security

Network Performance

Key

Network Performance

- Network Performance

- 00000000 500 0000
- 00000000

## 000

- 00/00 - 000000000000
- 00 - 0000000000
- 0000 - 0000000000000000

## 00000

- 00 - 00/00/0000
- 000 - 0000
- 00 - 0000
- 00 - 0000
- 00 - 0000000000
- 00 - 0000

## 00

## 0000000000

- 0000000000
- 00000000 IP 00
- 0000000000

## 00000000

- 0000000000000000
- 0000000000
- 0000000000

## 000000

- 0000000000
- 0000
- 00000000

- 部署環境構築

## 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 InfluxDB 環境構築

## 環境構築

## 環境構築

- 環境構築/環境
- `config/config.exs` 環境
- 環境720 環境30 環境

## 環境

- 環境構築
- 環境構築

## □□□□□

- □□□□□□□□
- □□□□□□

## □□□□□

### □□□□□□□□□

### □□□□□

- □□□□
- □□□□□□□□/□□□□
- □□□□□□□

### □□□□□

- □□□□ - □□□ PM □□□□□
- □□ - □□□□□□□□□
- □□ - □□□□□□□□□□□
- □□ - □□□□□□□□□□

### □□□

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

## □□□□□□□

1. □□□□ - □□□□□□□□□□□□□□
2. □□□□□ - □□□□□□□□□□□□□□
3. □□□□ - □□□□□□□□□□□□□□
4. □□□□ - □□“□□□□□”□□□□□□



□□□□□

- **720** □□□**30** □□ - □□□□□□
- **2160** □□□**90** □□ - □□□□□□□□□□
- **4320** □□□**180** □□ - □□□□□□□□□□
- **8760** □□□**365** □□ - □□□□□□

□□

- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□
- □□ InfluxDB □□□□□□

□□□ □□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□ □□□□□□□□□□

---

## InfluxDB □□□□

□□□□ InfluxDB □□□□□□□□□□□□□□

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

~~~~~ InfluxDB ~~~~~

~~~~~

項目	項目
項目	項目項目項目項目項目項目
項目	項目 InfluxDB 項目
<b>InfluxDB</b> 項目	項目項目項目項目2.x

~~~~~

~~~~~





Table of Contents

Page	Topic
1	InfluxDB Introduction
2	InfluxDB Architecture8086
3	InfluxDB Setup
4	Getting Started
5	3 PerformanceMetricsConfigurationAlarms

Table of Contents

Table of Contents

- InfluxDB Introduction - Introduction to InfluxDB
- Architecture - InfluxDB Architecture
- Setup - InfluxDB Setup
- Getting Started - Getting Started with InfluxDB

Table of Contents

Page 30 of 30

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>



項目	説明
Airscale	監視
監視	監視 PM 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

監視監視監視監視監視監視

監視

Erlang VM 監視

項目	説明
監視	監視 Erlang VM 監視
監視	Erlang 監視監視
監視	監視監視監視監視XMLJSON 監視
監視	監視監視
監視	監視 Erlang 監視
監視	監視 CPU 監視監視監視0 監視



PM

PM 5

PM

- PM
- PM
- PM
- PM
- PM

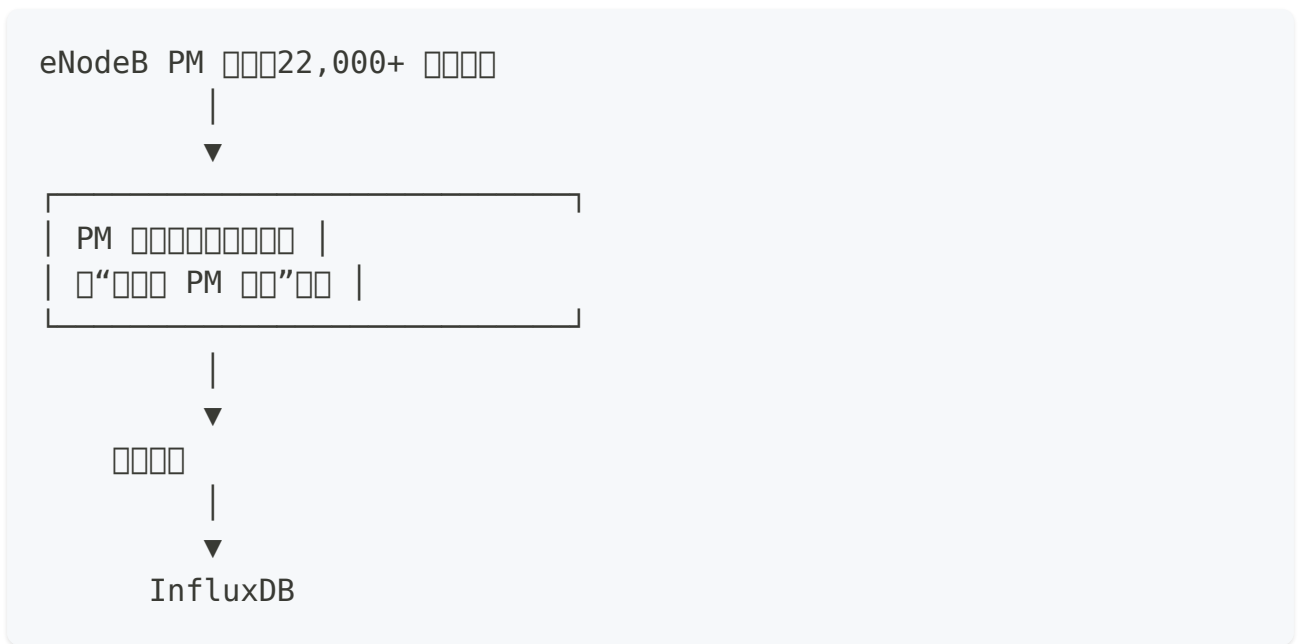
## PM

PM InfluxDB Nokia AirScale 22,000 PM

**URL:** <https://<ran-monitor-ip>:9443/nokia/pm-filters>

PM

**PM** □□□□□□□□



□□□□□□□□“□□□ PM □□”□□□□□□□□ PM □□

□□

1111

項目	内容
PM 機能	InfluxDB 連携
監視範囲	22,000+ 項目
設定	config_keys.csv

□□□ **PM** □□□□

□□□□□□□□□□□□□□□□ InfluxDB□

□□□□

- 電話番号 ID 番号
- 電話番号LTE/5G-NR 番号

- 下載 CSV 檔案

說明

- 檔案名稱
- 檔案格式
- 檔案內容 `pm_counters.csv` 檔案

檔案說明

檔案名稱 Nokia PM 檔案

說明

- 檔案 ID
- 檔案格式

說明

- 檔案格式
- 檔案大小 200 MB

檔案說明

檔案	檔案名稱	說明
<b>LTE</b>	M8xxx	LTE L1/L2/L3 檔案
<b>WCDMA</b>	M5xxx	3G WCDMA 檔案
<b>5G-NR</b>	M55xxx	5G NR 檔案
<b>5G-檔案</b>	M51xxx	5G 檔案
<b>5G-檔案</b>	M40xxx	5G 檔案

PM

PM

- PM `priv/pm_filters.etf`
- PM
- PM

PM

InfluxDB PM

- **PM** PM
- PM/PM

PM PM

---

PM

PM

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



- 提取压缩包 TAR 包
- 提取压缩包内容

提取压缩包 在 `/tmp` 提取压缩包

## InfluxDB 部署

部署 InfluxDB 数据库

- 部署 PM 节点
- 部署
- 部署

部署 InfluxDB “节点”数据库 InfluxDB 节点

部署 InfluxDB 数据库

部署

部署 `config/runtime.exs` 部署 ETS 包

部署

- 部署
- 部署
- 部署
- 部署

---

## Web UI 部署

部署 Web UI 数据库

部署

部署 数据库

□□□

1. □□□□□□
2. □□□□□□□□□□
3. □□□□□□□□□□
4. □□□□□□/□□□□□□
5. □□□□□□□□
6. □□□□□□□□□□

□□□ < 5 □□

□□□□□

□□□ □□□□□□□□

□□□

1. □□□□□□
2. □□□□□□□□□□□□
3. □□□□□□□□□□□□□□
4. □□□□□□□□□□
5. □□□□□□□□□□□□
6. □□□□□□□□
7. □□□□□□□□
8. □□□□□□□□□□

□□□□□□□□□□□□□□□□ □□□□□□□□

□□□□□

□□□ □□□□□□□□□□

□□□

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

4. 獲取每個 ID
5. 獲取 ID 列表
6. 獲取每個 ID 的數據
7. 獲取每個 ID 的數據
8. 獲取每個 ID 15-30 天
9. 獲取每個 ID 的數據

獲取每個 ID

- 獲取每個 ID 的數據
- 獲取每個 ID 的數據
- 獲取每個 ID 的數據
- 獲取每個 ID 的數據

獲取每個 ID

獲取每個 ID 的數據

獲取每個 ID

1. 獲取每個 eNodeBs 的數據
2. 獲取 ID 和 IP 列表
3. 獲取每個 `config/runtime.exs`
4. 獲取 RAN Monitor 的數據
5. 獲取每個 ID 的數據
6. 獲取每個 ID 的數據
7. 獲取每個 ID 的數據 InfluxDB
8. 獲取每個 ID 的數據
9. 獲取 Grafana 的數據

獲取每個 ID 的數據

獲取每個 ID 的數據

獲取每個 ID 的數據



目標

1. 環境構築が完了する
2. 監視対象/監視項目を定義する
3. 監視データを取得する
4. データを保存する
5. 監視データをpingする
6. 監視データを表示する
7. 監視データを通知する
8. 監視データを検索する
9. 監視データを削除する
10. 監視データを更新する

監視対象/監視項目を定義する

監視対象

- CPU使用率 - 監視対象
- メモリ使用率 - 監視対象
- ディスク使用率 - 監視対象
- ネットワーク使用率 - 監視対象
- **Grafana** 監視 - 監視対象
- 監視対象 - 監視対象
- 監視対象 - 監視対象

# RAN 部署環境構築

部署環境 (RAN) 構築環境

OmniTouch 環境構築

## 目次

1. 概要
2. RAN 環境構築
3. ネットワーク
4. Web UI 構築
5. Grafana 構築
6. ネットワーク
7. ネットワーク
8. ネットワーク
9. ネットワーク

## 概要

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

## 環境構築

- ネットワーク - ネットワーク環境構築
- ネットワーク - ネットワーク環境構築
- ネットワーク - ネットワーク環境構築
- **Web** 環境 - ネットワーク Web UI 環境構築
- **Grafana** 環境 - ネットワーク環境構築

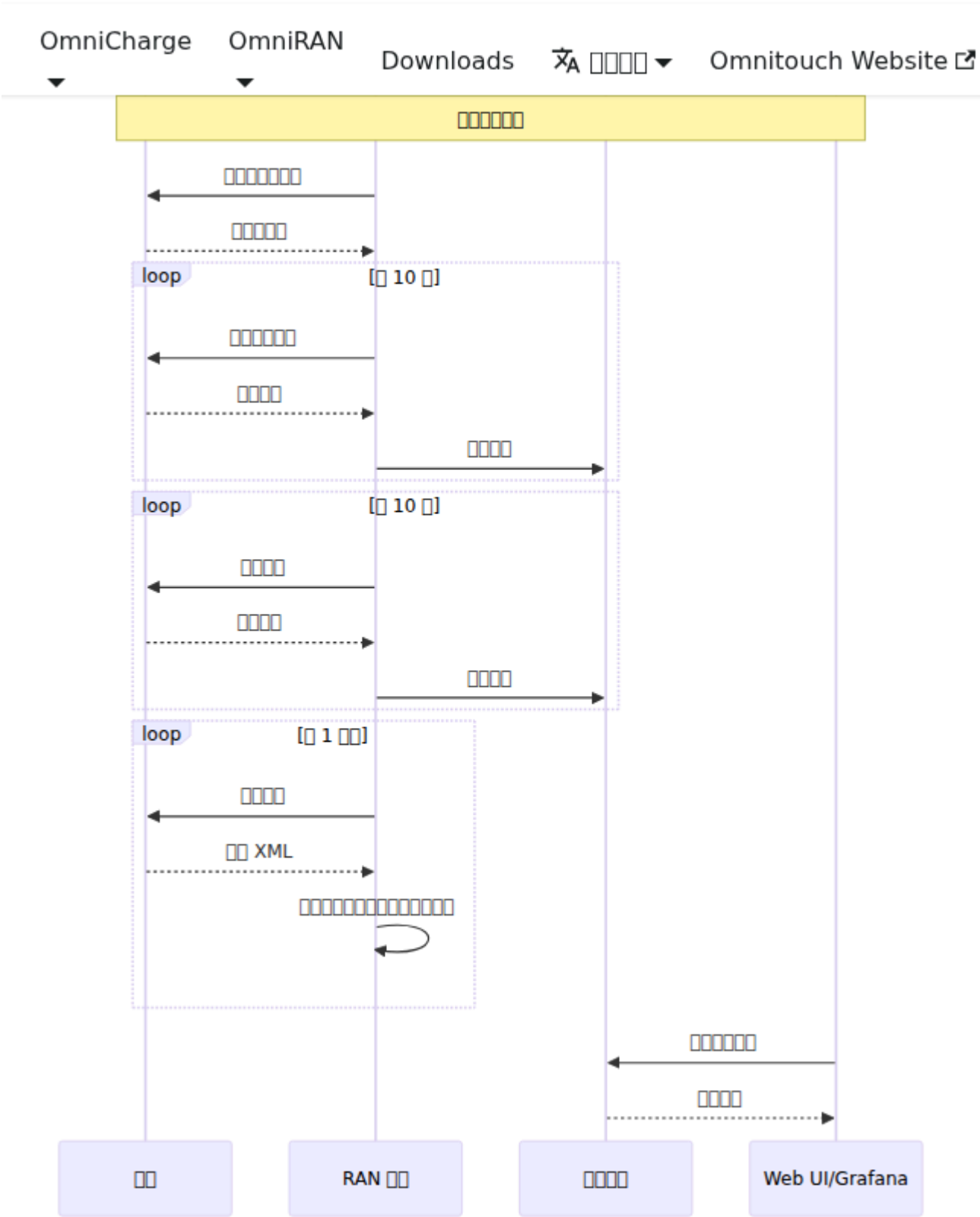


# RAN 部署計画

RAN 環境構築計画

1. 環境構築 - 仮想化環境 (Nokia 5G Core) 構築
2. 環境構築 - 10 以上の KPI 監視
3. 環境構築 - 監視ダッシュボード
4. 環境構築 - 監視アラート
5. 環境構築 - 監視レポート
6. 環境構築 - Web UI による Grafana 監視

□□□



□□□□□

□□□□□

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

000

- 00000000000000000000
- 0000000000
- 000000000
- 000000000

000

- 000 XML 0000000000000000
- 000000000000
- 000000000
- 00000000 10 000

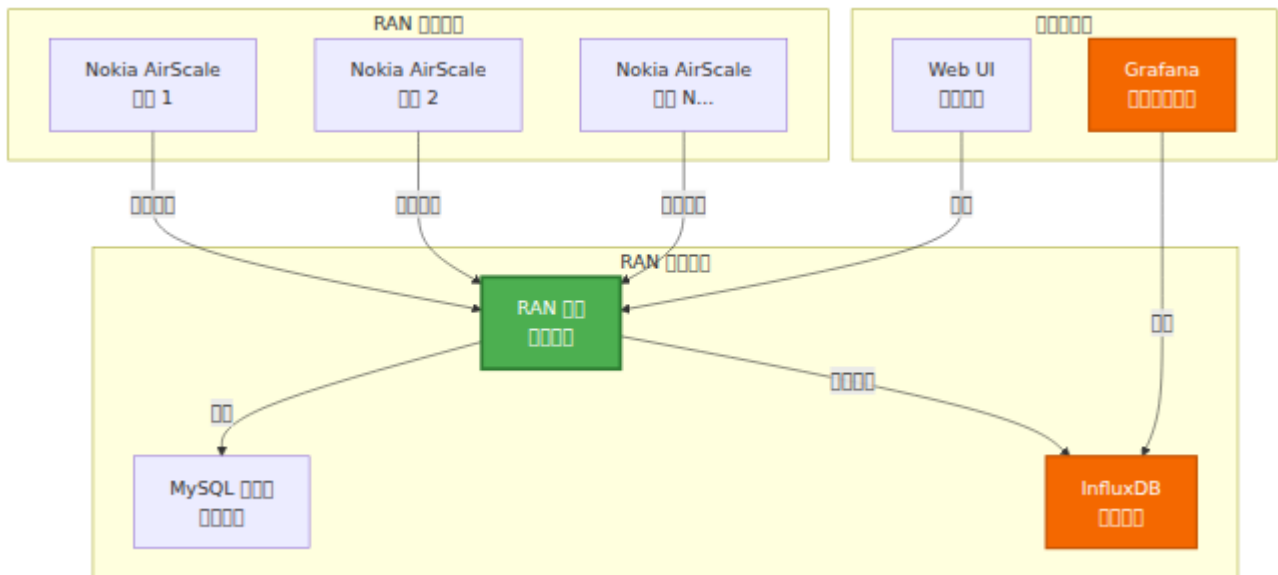
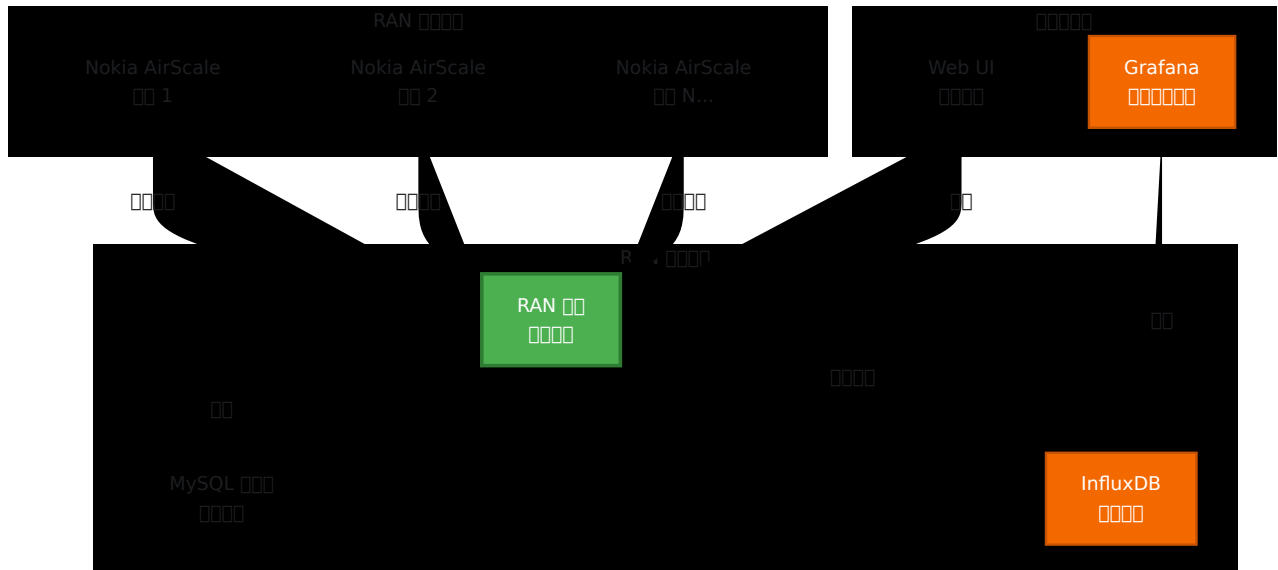
0000000000000000 0000000

0000000000000000 Nokia 0000000

---

--	--	--	--

□ □ □ □ □ □



□□□□□□ (TCE)

RAN  LTE/5G 

□□□ TCE□

XXXXXXXXXX Nokia AirScale XXXXXXXXXXXXXXXX





- **S1-AP** 接口 - eNodeB 和 EPC 之间的接口
- **RRC** 接口 - 无线资源控制
- **NAS** 接口 - 网络接入层
- 用户面接口 - PDCP 和 RLC

11

5/5

- 8000000000000000
- 80000000
- 80000000 RSRP RSRQ SINR
- 80000000

□□□□□

- 
- 
- 
- 

□□□□□

- PCI 4x4x4
- 4x4x4x4
- 4x4x4x4
- 4x4x4x4x4

Wireshark TCE MDT



# Web UI

RAN Web UI

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

- 
- 
- 
- 

5

- 
- 
- 
-

□□□□

□□□□□□□□□□

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

□□□□□□□□□□□□□□□□ □□□□□□□

□□□□

□□□□□□□□

1. 00 00000000
2. 00 0000 → 0000 ID
3. 00 0000 ID 0000
4. 00 000000
5. 00 0000000

□□□□□□□□□□□□□□□□□□□□

[illegible]

## Web UI - 0000000000

---

节点 eNodeB

节点信息

- ID 节点
- 节点名称
- 节点类型
- 节点位置

节点 ID 节点 `config/runtime.exs`

□□□□

□□□□□□□□□□□□

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

□□□□□□□□□□□□□□ □□□□□□□

📄📄📄📄

📄📄📄 InfluxDB 📄📄📄📄

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

📄📄📄📄📄📄📄📄📄 📄📄📄📄📄📄

---

# InfluxDB 简介

时间序列数据库

- 存储时间序列数据
- 支持多种数据类型
- 支持多种查询语言
- 支持多种存储引擎
- 支持 5 种存储引擎

特点

- 简单 + 易用 = 易用
- 简单 + 易用 = 易用
- 简单 + 易用 = 易用 InfluxDB 易用

---

## Web UI

Web UI

Web UI -

---

## Grafana

Web UI Grafana

### Grafana

Grafana

- /
- KPI
-



- 監視可能な項目
- 監視項目 SLA 設定
- 監視項目のグループ化

**Web UI 機能**

- 監視項目の検索
- 監視項目のフィルタリング
- 監視項目のグループ化
- 監視項目の削除
- 監視項目の再作成

*Grafana 監視項目の追加*

**監視項目**

**監視項目の追加**

- 監視項目の検索
- 監視項目のフィルタリング
- 監視項目のグループ化
- 監視項目の削除

- 配置

## NOC 配置

- 配置
- 配置
- 配置 24 小时
- 配置
- 配置

## 配置

- 配置
- 配置 SINR/RSRP 配置
- 配置 RLC 配置 RRC 配置
- 配置
- 配置

## Nokia AirScale 配置

- PRB 配置/配置
- 配置 PDCP 配置
- 配置 UE 配置
- 配置
- 配置
- RSSI 配置
- RRC 配置
- 配置 VSWR
- 配置

## 配置

**Grafana** 配置 - 配置

**Nokia** 配置 - 配置

---

# 目標

## 前提

インストール5-10 分

1. Web UI
- 2.
- 3.
- 4.
- 5.

Web UI -

## インストール

- 1.
- 2.
- 3.
- 4.
- 5.

## 設定

## インストール

- 1.
2. eNodeB
3. config/runtime.exs
4. RAN
- 5.
6. InfluxDB

## インストール

1. 安装依赖包
2. 在 `config/runtime.exs` 中添加配置
3. 安装并配置数据库
4. 安装 RAN 包
5. 安装 Grafana 仪表盘

部署步骤

1. 安装依赖包
2. 在 `config/runtime.exs` 中添加配置
3. 安装 RAN 包
4. 安装仪表盘

部署脚本

部署脚本 - 部署脚本

---



📁📁📁📁📁📁**NOC**📁📁📁📁📁

📁📁	📁📁	📁📁📁📁
Web UI 📁📁	📁📁📁📁📁📁📁	📁📁📁📁📁📁📁
📁📁📁📁📁📁	📁📁📁📁📁	📁📁📁📁📁📁📁📁📁📁📁
📁📁📁📁📁📁	📁📁📁📁📁📁📁📁	📁📁📁📁📁📁📁📁📁📁📁
📁📁📁📁📁	📁📁📁📁📁📁	📁📁📁📁📁📁📁📁
📁📁📁📁📁	📁📁📁📁📁	📁📁📁📁📁📁📁📁📁
📁📁📁📁📁📁📁	📁📁📁📁📁📁📁	📁📁📁📁📁📁📁📁📁

📁📁📁📁📁📁📁

📁📁	📁📁	📁📁📁📁
Grafana 📁📁📁📁	📁📁📁📁📁📁📁📁	📁📁📁📁📁📁📁📁
Nokia 📁📁📁📁📁	📁📁📁📁📁📁	📁📁📁📁📁📁📁📁
AirScale 📁📁📁📁	📁📁📁📁📁📁	📁📁📁📁📁📁📁📁
TCE MDT 📁📁📁📁📁📁	MDT 📁📁📁📁📁 Wireshark 📁📁	📁📁📁📁📁📁📁📁📁📁📁
API 📁📁📁📁	REST API 📁📁	📁📁📁📁📁📁📁📁

部署環境

OS	DB	サービス
Ubuntu 18.04	MySQL	OpenStack

前提条件

必要な RAN 環境

1. Web UI 環境を構築する
2. MySQL データベースをインストールする
3. MySQL データベースを起動する
4. OpenStack をインストールする

インストール

1. MySQL データベースをインストールする
2. MySQL データベースを起動する
3. OpenStack をインストールする

接続情報

接続先

サービス	URL	接続先
Web UI 環境	https://<server>:9443	OpenStack
Grafana	http://<server>:3000	OpenStack
InfluxDB	http://<server>:8086	OpenStack

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 XML 數據庫

目錄

目錄

- RAN 配置
- 配置運行時環境
- 配置外部數據庫 30 天
- 配置 MySQL 數據庫

目錄

- 10 配置運行時環境
- 10 配置 + 配置 Webhook 配置
- 1 配置外部數據庫
- 配置 InfluxDB 數據庫

目錄

- 配置 720 配置 30 天
- 配置
- 配置
- 配置 Web UI 配置

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
-

## □□

### □□□□□□

□□	□□
□□□□□□	□□□□□□□□
□□□□□□	□□□□□□□□
□□□□□□	□□□□□□□□
<b>InfluxDB</b> □□□□	□□□□□□

### □□□□□□

#### □□□□□□

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 □□□□

無線ネットワークの構築と運用 無線ネットワーク

## 無線ネットワーク

無線ネットワーク

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IPアドレス
- 無線ネットワークのセキュリティ
- RAN 無線ネットワーク

無線

無線 RAN 無線ネットワーク

- Omnitouch 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セキュリティ

無線ネットワーク

1. 無線 ネットワーク
  2. 無線ネットワークの種類
  3. 無線ネットワークの構成
  4. 無線ネットワークのセキュリティ
  5. 無線ネットワークの種類
-

□□□□

