Nokia AirScale □置指南

为 **RAN Monitor** 集成□置基站

目录

1. □述
2. 先决条件
3. 访问 WebLM 界面
4. 配置性能监控
5. 配置参数参考
6. 验证
7. 故障排除

概述

为了使 RAN Monitor 能够从 Nokia AirScale 基站收集性能指标、警报和配置数据，您必须配置基站以向 RAN Monitor 系统报告数据。这是通过 Nokia Web Element Manager (WebLM) 界面完成的。

本指南介绍了配置性能测量公共管理 (PMCADM) 管理对象的过程，该对象控制基站如何将性能数据发送到外部系统。

谁应该使用本指南

重要：所有 Nokia AirScale 基站配置均由 **Omnitouch** 执行，作为初始部署和持续支持的一部分。本指南适用于：

- 高级用户，希望了解基站配置
- 自我管理的部署，客户自行配置基站
- 故障排除 和理解当前配置
- 在自我管理环境中额外的基站入网

如果您是 **Omnitouch** 管理的客户，请联系 **Omnitouch** 支持以获取基站□置和入网。

有关收集的指标的理解，请参见 Nokia Counter Reference。有关系统配置，请参见 Runtime Configuration Guide。

先决条件

在配置基站之前，请确保您具备：

- 网络访问 - 连接到基站管理界面
- 管理员凭据 - 具有配置权限的用户名和密码
- **RAN Monitor** 详细信息 - RAN Monitor 监听的 IP 地址和端口
- 支持的软件 - 兼容的 Nokia AirScale 基站软件版本

所需信息：

参数	值	示例
RAN Monitor IP 地址	RAN Monitor 运行的 IP	10.179.2.139
RAN Monitor 端口	收集端口 (默认: 9076)	9076
收集间隔	发送指标的频率	60 秒

访问 **WebLM** 界面

步骤 **1**: 打开 **Web Element Manager**

1. 打开一个网页浏览器
2. 导航到基站管理界面：

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

或

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

3. 使用您的管理员凭据登录

步骤 2：导航到□置管理

登录后：

1. 点击顶部菜单栏中的 **Configuration**
2. 从下拉菜单中选择 **Configuration Management**
3. 点击 **Parameter Editor** 标签

您现在应该在左侧面板中看到配置树，在主窗口中看到参数编辑器。

□置性能监控

步骤 1：禁用 OAM TLS

重要：在配置性能监控之前，您必须禁用基站上的 OAM TLS 强制。默认情况下，`oamTls` 管理对象设置为 `forced`，这会阻止与外部管理系统的未加密通信。

在左侧导航面板（对象树）中：

1. 展开 **Current BTS configuration**
2. 展开 **CURRENT_BTS_CONF-1**
3. 展开 **MRBTS-X** (其中 X 是您的基站 ID)
4. 找到 **oamTls** 管理对象
5. 将值从 `forced` 更改为 `off`

参数	默认值	所需值
oamTls	forced	off

注意：在 RAN Monitor 建立与基站的通信之前，必须更改此设置。如果保持默认的 `forced` 设置，基站将拒绝非 TLS 连接。

步骤 2：找到 PMCADM 管理对象

在左侧导航面板（对象树）中：

1. 展开 **Current BTS configuration**
2. 展开 **CURRENT_BTS_CONF-1**
3. 展开 **MRBTS-X**（其中 X 是您的基站 ID）
4. 展开 **MNL-1**（管理链路）
5. 展开 **MNLENT-1**（管理链路实体）
6. 点击 **PMCADM-1**（性能测量公共管理）

参数编辑器将显示 PMCADM-1 配置参数。

步骤 3：□置实时性能监控

向下滚动到 **Structure 1** 部分，该部分包含实时性能监控收集实体设置。配置以下参数：

所需参数：

参数	描述	推荐值
TLS 认证的证书类型	安全证书类型	RSA (如果启用 TLS)
实时性能监控收集实体主机	RAN Monitor 的 IP 地址	您的 RAN Monitor IP (例如, 10.179.2.139)
实时性能监控收集实体端口号	RAN Monitor 监听的端口	9076 (默认)
实时性能监控收集间隔	发送指标的频率	60s (根据需求调整)
启用 **TLS**	使用加密连接	false (用于初始设置)

可选参数:

参数	描述	默认值
SDL 最大上传文件数量	最大并发上传	1
SDL Nonce	安全的唯一标识符	12345678
启用零值计数器抑制	抑制值为 0 的计数器	false (建议保留所有数据)

步骤 4: 保存并激活□置

配置所有参数后:

1. 审核您的更改 - 验证所有 IP 地址和端口是否正确
2. 创建□置计划:
    - 点击顶部的 **Create Plan** 按钮
    - 系统将验证您的更改
    - 注意提供的计划 ID
3. 验证计划:

- 点击 **Validate Plan** 按钮
- 输入计划 ID
- 等待验证完成
- 解决任何验证错误

4. 激活□置:

- 点击 **Activate Plan** 按钮
- 输入计划 ID
- 确认激活
- 基站将应用新配置

替代方案: **XML** □置

对于高级用户或自动化部署，可以使用 XML 应用 PMCADM 配置。这是与上述手动配置相对应的配置片段:

```
<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 中的关键参数:

- `rTPmCollEntityHost` - 设置为您的 RAN Monitor IP 地址
- `rTPmCollEntityPortNum` - 设置为 9076 (默认 webhook 端口)
- `rTPmCollInterval` - 收集间隔 (推荐 60s)

- `tlsEnabled` - 初始设置时设置为 false
- `sdlPrimaryDestIp` - 设置为您的 RAN Monitor IP 地址

注意：将 `10.179.2.139` 替换为您的实际 RAN Monitor IP 地址，并根据您的基站 ID 调整 `MRBTS-132`。

□置参数参考

LNBTS 性能测量参数

LNBTS (LTE NodeB) 管理对象包含控制收集哪些性能测量组的参数。必须启用这些参数以收集某些 PM 计数器类别。

actL1PM - 激活层 1 性能测量

设置	值
参数	actL1PM
位置	NOKLTE:LNBTS 管理对象
类型	布尔值 (true/false)
默认	false
所需	RTWP 计数器 (M8005C306-339)

描述：启用层 1 (物理层) 性能测量，包括每个天线的 RTWP (接收总宽带功率)。禁用时，基站将不报告 RTWP、干扰和其他低级无线测量。

受影响的计数器：

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

通过 **XML** 配置:

```
<managedObject class="NOKLTE:LNBTS" distName="MRBTS-XXXX/LNBTS-
XXXX" operation="update">
  <p name="actL1PM">true</p>
</managedObject>
```

通过 **WebLM** 配置:

1. 导航到 Configuration > Configuration Management > Parameter Editor
2. 展开 MRBTS-X > LNBTS-X
3. 找到 `actL1PM` 参数
4. 将值设置为 `true`
5. 创建、验证并激活配置计划

actPCMDReport - 激活每通话测量数据报告

设置	值
参数	actPCMDReport
位置	NOKLTE:LNBTS 管理对象
类型	布尔值 (true/false)
默认	false
所需	每通话测量数据

描述：启用每通话测量数据 (PCMD) 报告，提供单个通话会话的详细测量。

通过 **XML** 配置:

```
<managedObject class="NOKLTE:LNBTS" distName="MRBTS-XXXX/LNBTS-
XXXX" operation="update">
  <p name="actPCMDReport">true</p>
</managedObject>
```

PMCADM-1 对象概述

PMCADM（性能测量公共管理）管理对象控制如何从基站收集和报告性能数据。

主要职责：

- 配置实时性能监控目标
- 设置指标报告的收集间隔
- 控制数据格式和传输参数
- 管理数据传输的安全设置

PMCADM 关键参数

reportingIntervalPm - PM 报告间隔

设置	值
参数	reportingIntervalPm
位置	com.nokia.srbts.mnl:PMCADM 管理对象
类型	枚举
选项	5min, 15min, 30min, 60min
推荐	5min 或 15min 用于实时监控

描述：控制基站上传性能测量数据的频率。较低的值提供更细粒度的数据，但会增加网络流量和存储需求。

对数据收集的影响：

- **5min** - 最适合实时监控和故障排除
- **15min** - 在细粒度和资源使用之间取得良好平衡
- **60min** - 仅适合长期趋势分析

通过 **XML** 置：

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

通过 **WebLM** □置:

1. 导航到 Configuration > Configuration Management > Parameter Editor
2. 展开 MRBTS-X > MNL-1 > MNLENT-1 > PMCADM-1
3. 找到 `reportingIntervalPm` 参数
4. 将值设置为所需的间隔（例如，`5min`）
5. 创建、验证并激活配置计划

实时性能监控收集实体

此子结构定义基站发送实时性能指标的位置和方式。

certTypeForTlsAuth - TLS 认证的证书类型

- 类型：枚举 (RSA, DSA, ECDSA)
- 目的：指定启用 TLS 时的证书类型
- 默认： RSA
- 使用：仅在 tlsEnabled = true 时相关

rTpmCollEntityHost - 收集实体主机

- 类型： IP 地址 (IPv4 或 IPv6)
- 目的�� 性能指标的目标 IP 地址
- 必需：是
- 示例： 10.179.2.139
- 备注：必须能够从基站管理网络访问

rTpmCollEntityPortNum - 收集实体端口号

- 类型：整数 (1-65535)
- 目的：收集系统监听的 TCP 端口

默认： 9076
备注：必须与 RAN Monitor 配置匹配

rTpmCollInterval - 收集间隔

类型：时间 (秒)
目的：性能数据传输的频率
选项： 15s, 30s, 60s, 300s, 900s, 1800s
默认： 60s
推荐：标准监控为 60s，详细故障排除为 15s

tlsEnabled - 启用 TLS

类型：布尔值 (true/false)
目的：加密传输中的性能数据
默认： false
备注：如果启用，双方需要有效的证书

sdlMaxUploadFileNumber - SDL 最大上传文件数量

类型：整数
目的：最大并发文件上传数量
默认： 1
备注：在高流量环境中增加

sdlNonce - SDL Nonce

类型：字符串 (8 位数字)
目的： SDL 协议安全的唯一标识符
默认： 12345678
备注：可以出于安全目的更改

suppressZeroValueCount - 抑制零值计数器

类型：布尔值 (true/false)
目的：从报告中省略值为零的计数器
默认： false

推荐：保持为 false 以维护完整的数据以进行趋势分析

验证

激活配置后，验证基站是否成功向 RAN Monitor 发送数据。

检查 RAN Monitor Web UI

1. 打开 RAN Monitor Web UI: http://<ran-monitor-ip>:4000/
2. 导航到eNodeB Status页面
3. 在设备列表中找到您的基站
4. 验证状态显示为 Connected (绿色指示)
5. 检查Session显示为Active

预期状态：

状态： Connected (绿色)
会话： Active
地址：与基站 IP 匹配

操作：所有按钮启用

如果状态显示为 Pending：

设备正在尝试注册，但尚未完成身份验证。

可能的原因：

管理器 ID 和注册密钥不匹配
RAN Monitor 未配置为接受此设备
网络连接问题

如果状态显示连接错误：

设备无法到达 RAN Monitor。

可能的原因：

PMCADM 配置中的 IP 地址不正确
网络路由问题
防火墙阻止端口 8080
RAN Monitor 服务未运行

检查数据收集

查看 InfluxDB 状态：

1. 在 RAN Monitor Web UI 中，导航到InfluxDB Status
2. 验证数据点是否在增加
3. 检查 Performance Metrics 计数是否在增长
4. 确认 Last Update 时间戳是最近的

预期指标：

Performance Metrics：计数定期增加
Configuration：存在数据点
Alarms：如果没有活动故障，可能为 0

检查数据保留：

1. 导航到Data Retention Policy页面
2. 找到您的基站
3. 验证Performance Metrics、Configuration和Alarms计数

故障排除

基站未出现在 **RAN Monitor** 中

症状：设备未显示在 eNodeB 状态页面

解决方案：

1. 验证网络连接：

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

2. 检查 **RAN Monitor** □置：

   - 确保设备已添加到 `config/runtime.exs`
   - 验证 IP 地址与基站匹配
   - 确认凭据正确

3. 查看 **RAN Monitor** 日志:
    - 导航到 **Live Logs** 页面
    - 过滤错误消息
    - 查找来自基站的连接尝试
4. 验证基站□置:
    - 重新检查 WebLM 中的 PMCADM-1 设置
    - 确认 RAN Monitor IP 地址正确
    - 确保指定了端口 9076

设备显示**“Pending”**状态

症状：设备出现但状态为黄色“Pending”

解决方案:

1. 检查管理器注册:
    - 验证 RAN Monitor 中的管理器 ID 是否与基站预期匹配
    - 确认注册密钥是否正确配置
2. 查看身份验证:
    - 检查 runtime.exs 中的凭据
    - 确保用户名/密码与基站设置匹配
3. 等待注册周期:
    - 注册可能需要 30-60 秒
    - 等待后刷新页面

连接错误

症状： “网络错误: enetunreach”或类似错误

解决方案:

1. 验证网络路径:
    - 测试基站到 RAN Monitor 的连接
    - 检查路由表
    - 验证 VLAN/subnet 是否正确配置
2. 检查防火墙规则:
    - 确保端口 9076 是开放的 (用于实时性能数据)
    - 确保端口 8080 是开放的 (用于 SOAP API 通信)
    - 验证没有 ACL 阻止流量
    - 检查 RAN Monitor 服务器上的 iptables 规则
3. 验证 **RAN Monitor** 是否在监听:

```
# □查 SOAP API 和 webhook 端点
netstat -tlnp | grep -E '8080|9076'
```

RTWP 或层 1 计数器缺失

症状: 基站正在报告数据，但 InfluxDB 中缺少 RTWP 计数器 (M8005C306-339)

原因: LNBTS 配置中的 `actL1PM` 参数设置为 `false`，这会禁用层 1 性能测量。

解决方案:

1. 检查当前设置:
    - 导出基站配置
    - 在 LNBTS 部分搜索 `actL1PM`
    - 如果值为 `false`，这就是原因
2. 启用层 **1 PM**:
    - 访问基站上的 WebLM
    - 导航到 MRBTS-X > LNBTS-X
    - 将 `actL1PM` 设置为 `true`
    - 创建、验证并激活配置计划

3. 激活后验证：

等待下一个收集间隔 (通常为 60 秒)
检查 InfluxDB 中的 M8005C306-309 计数器 RTWP 数据现在应该出现

相关参数：

actPCMDReport - 每通话测量数据 (可能也需要启用)

InfluxDB 中没有指标

症状：设备已连接，但数据库中没有数据

解决方案：

1. 验证收集间隔：

检查 PMCADM-1 rTpmCollInterval 设置
至少等待一个完整的间隔周期
刷新 InfluxDB 状态页面

2. 检查 InfluxDB 连接：

导航到 InfluxDB 状态页面
验证“已连接”指示器为绿色
确认桶名称正确

3. 查看 RAN Monitor 日志：

查找 InfluxDB 写入错误
检查数据解析问题
验证 API 令牌具有写入权限

TLS/证书问题

症状：启用 TLS 时连接失败

解决方案:

1. 验证证书是否已安装:
    - 检查基站是否具有有效证书
    - 确保 RAN Monitor 具有相应的 CA 证书
2. 先尝试不使用 **TLS**:
    - 将 tlsEnabled 设置为 false
    - 验证基本连接是否正常
    - 确认功能后重新启用 TLS
3. 检查证书有效性:
    - 验证证书是否未过期
    - 确认证书主题名称匹配
    - 检查证书链是否完整

其他资源

相关文档

- 操作指南 - 完整的 RAN Monitor 操作文档
- 运行时配置指南 - RAN Monitor 配置参考
- Nokia Counter Reference - 性能计数器定义
- Grafana 集成 - 使用收集的指标构建仪表板
- API 端点 - 设备管理的 REST API 参考
- 数据保留政策 - 管理存储的性能数据

□置文件

- **config/runtime.exs** - RAN Monitor 设备配置

支持

对于本指南未涵盖的问题：

1. 查看 RAN Monitor 应用日志
2. 检查适用于您软件版本的 Nokia 基站文档
3. 验证网络基础设施配置
4. 与网络运营团队咨询

警报管理与升级

故障处理、严重性级别和操作响应

[illegible]

目录

1. [illegible]述
2. 警报生命周期
3. 严重性级别
4. 警报类别
5. 调查与故障排除
6. 升级程序
7. 解决方案跟踪
8. 最佳实践

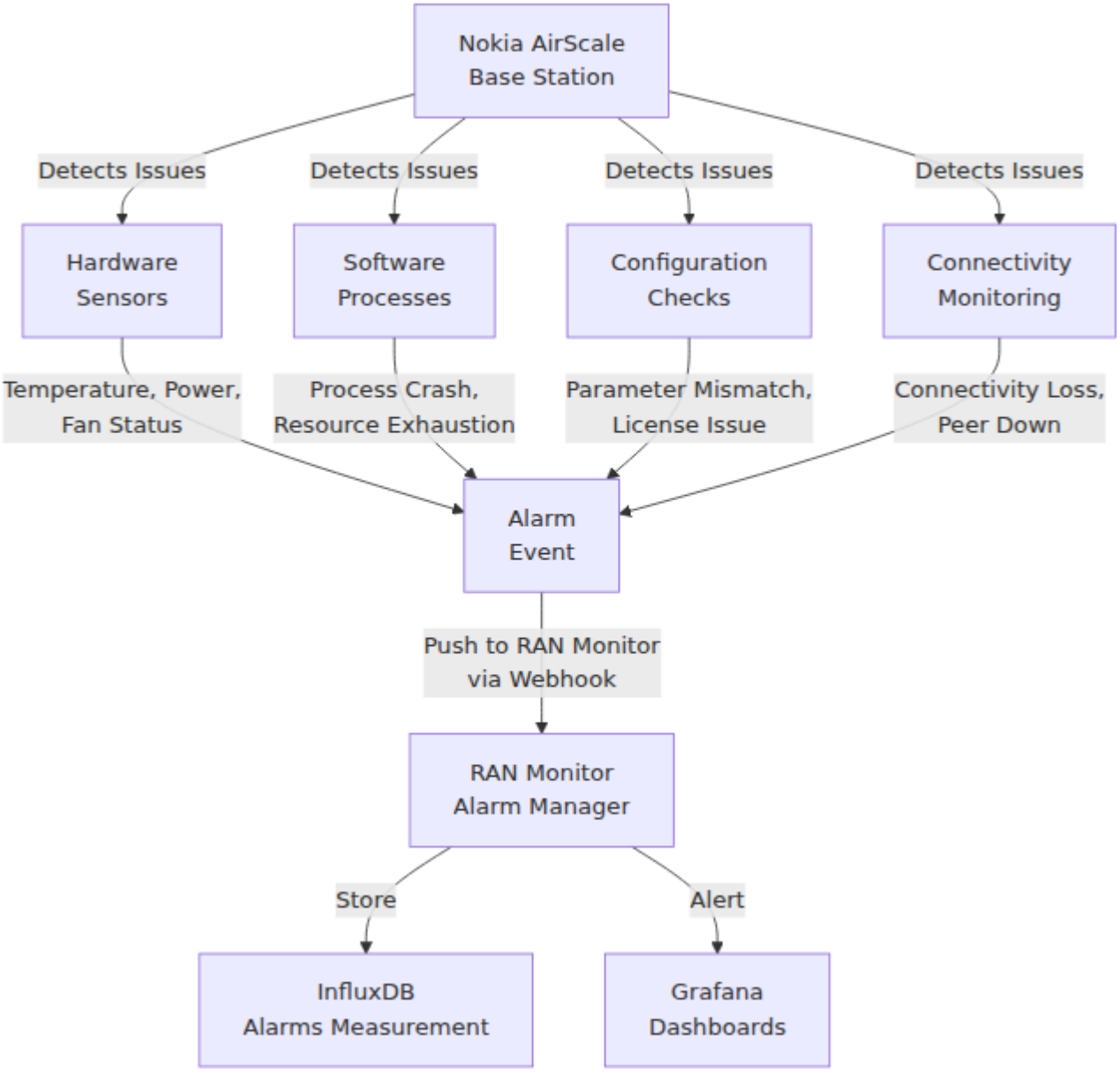
概述

警报（也称为“故障”）表示在诺基亚 AirScale 基站上检测到的问题或异常。RAN Monitor 持续监控活动警报，并跟踪其从生成到解决的生命周期。

示例警报仪表板：

□□□□ 4G □□□□□□□□□□□□□□□□□□□□□□/□□□□□□□□□□□□□□□/□□□□□□□□□□□□□□□□□□□□

□□□□


Nokia AirScale
Base Station
Detects Issues
Detects Issues
Detects Issues
Detects Issues
Hardware
Sensors
Software
Processes
Configuration
Checks
Connectivity
Monitoring
Temperature, Power,
Fan Status
Process Crash,
Resource Exhaustion
Parameter Mismatch,
License Issue
Connectivity Loss,
Peer Down
Alarm
Event
Push to RAN Monitor
via Webhook
RAN Monitor
Alarm Manager
Store
Alert
InfluxDB
Alarms Measurement
Grafana
Dashboards

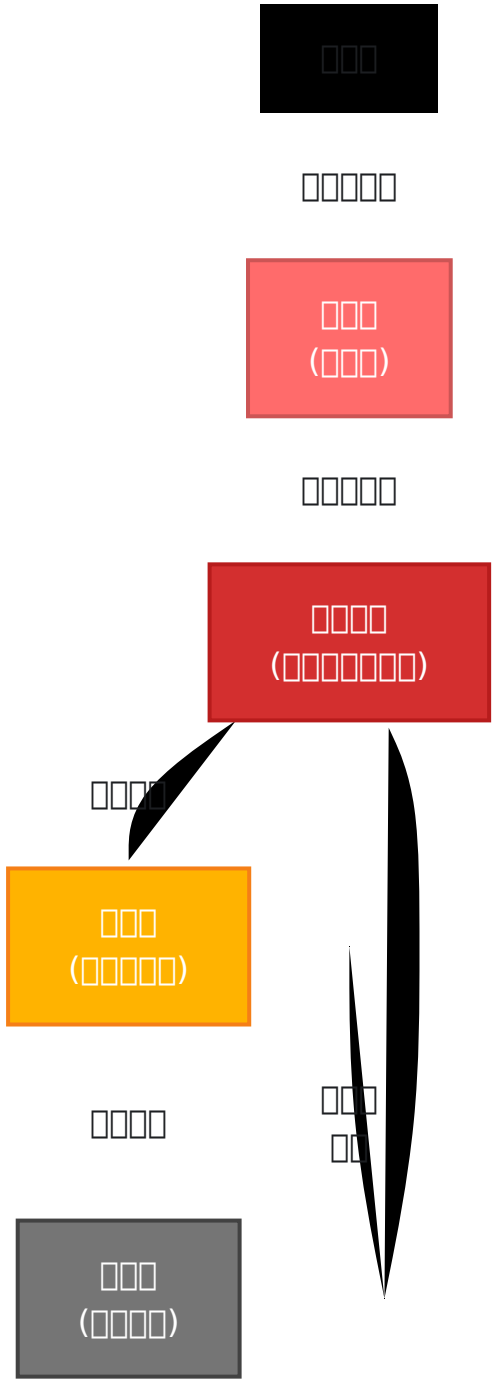
□□□□□□

□□□□□□□

属性	示例	目的
警报 **ID**	a1b2c3d4-e5f6-...	唯一标识符
严重性	严重、主要、次要	优先级级别
可能原因	"小区不可用"	根本原因类别
具体问题	"S1 连接丢失"	详细问题
受影响系统	/BSC-1/BTS-23/Cell-A	受影响的内容（DN）
事件时间	2025-12-10 14:23:45	检测时间
状态	活动 / 已清除	当前状态

□□□□□□□

□□□□



示例警报时间线

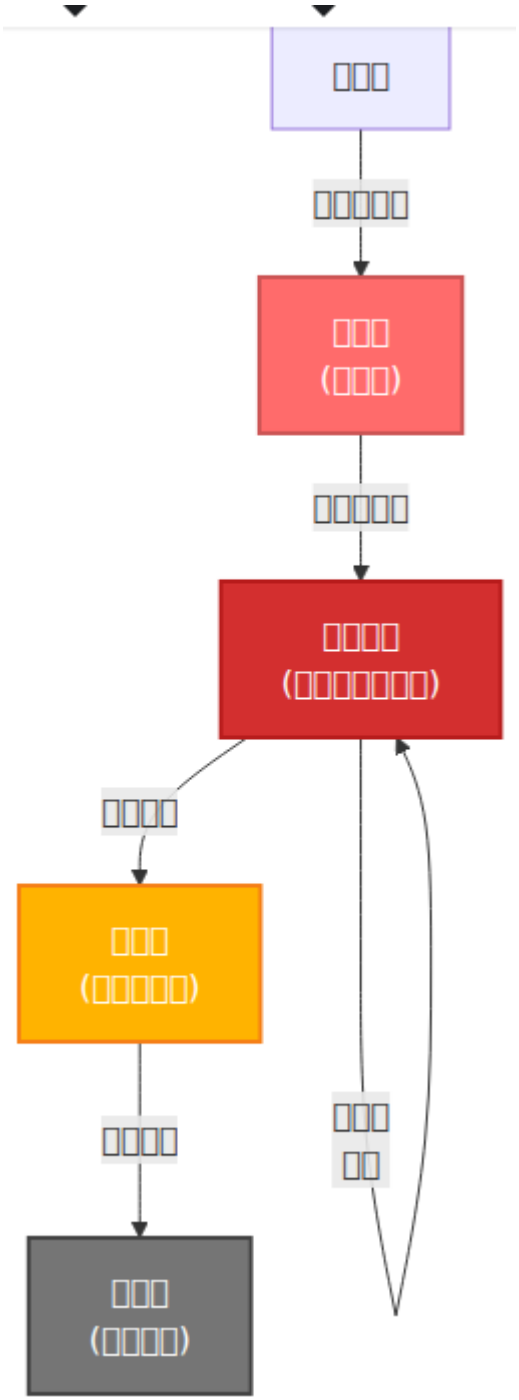
```
14:23:45 UTC - 问题发生
              └─> 基站□测到连接丢失
              └─> 生成警报:“S1 连接丢失”(严重)

14:23:47 UTC - 警报推送到 RAN Monitor
              └─> NE3S webhook 通知已接收
              └─> 存储在 InfluxDB
              └─> 警报规则触发

14:23:50 UTC - 发送通知
              └─> Grafana 警报触发
              └─> 向 NOC 团队发送 Slack 消息
              └─> 创建 PagerDuty 事件

14:24:15 UTC - 操作员确认
              └─> NOC 团队标记为已确认
              └─> 开始持续时间跟踪

14:28:35 UTC - 问题���动解决
              └─> 连接恢复
              └─> 基站清除警报
              └─> RAN Monitor 记录“已清除”

14:28:36 UTC - 警报关闭
              └─> 持续时间记录: 5 分钟 51 秒
              └─> 用于 SLA 报告的跟踪
              └─> 30 天后归档
```

严重性级别

RAN Monitor 跟踪五个严重性级别，每个级别具有不同的操作影响和升级要求:

严重严重性

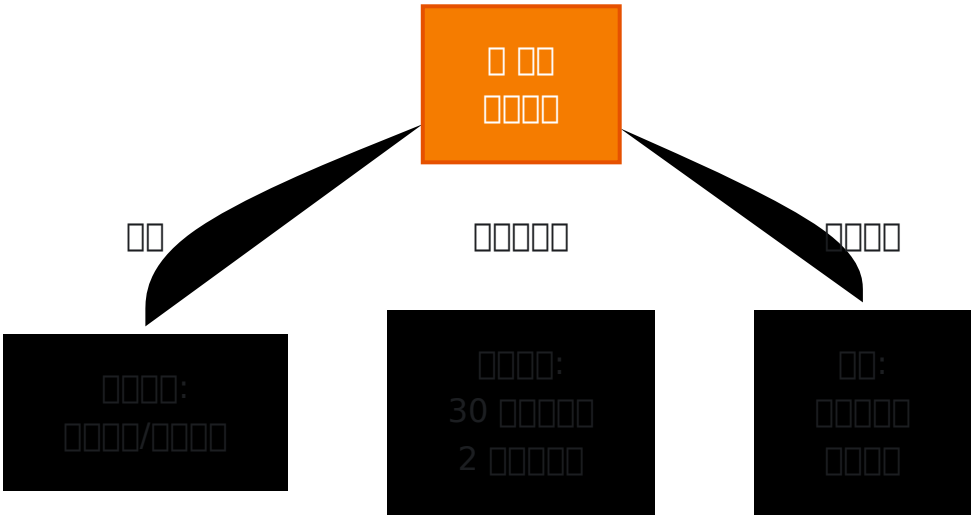
定义: 影响服务，需要立即采取行动



示例：

- 设备完全无法访问（连接丧失）
- 所有小区故障（基带故障）
- 控制面接口故障（S1 丢失）
- 完全数据转发故障
- 基站未响应管理

升级：



- 立即通知值班工程师（电话）
- 在事件管理系统中创建事件
- 更新状态页面
- 如果 SLA 受到影响，通知管理层

响应 **SLA**: < 15 分钟

主要严重性

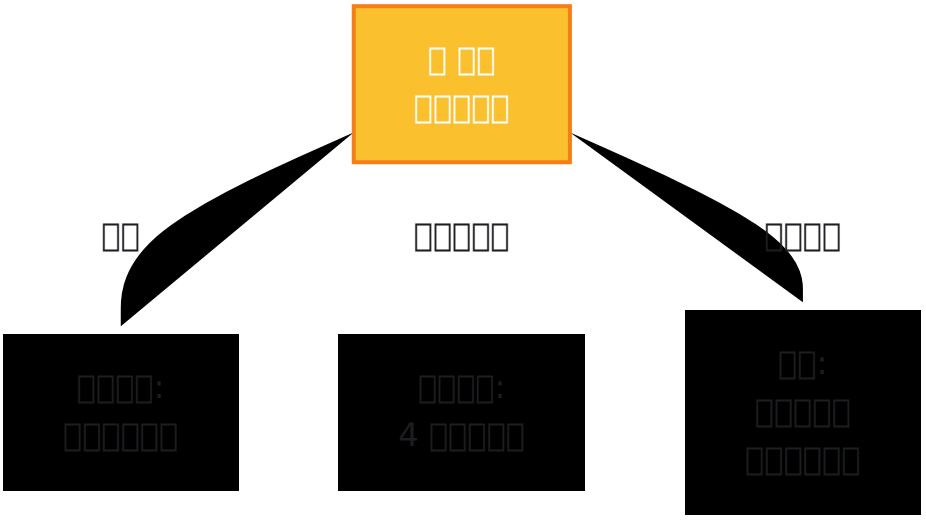
定义：性能下降，需要紧急调查

示例：

- 小区可用性 < 95% 超过 15 分钟
- 切换成功率 < 95%
- DL/UL 资源被阻塞（> 95% 利用率持续）
- RLC 重传率 > 5%
- 多个小区显示质量差
- 链路降级（E1/T1 错误增加）

升级：

- 通知 NOC 团队 + 高级工程师
- 在事件管理中创建事件
- 如果 30 分钟后仍未解决，则通知工程团队
- 检查是否对其他小区/站点造成级联问题



响应 **SLA**：< 30 分钟调查

次要严重性

定义：降级，跟踪和调查

示例：

- 小区可用性 95-98%（趋势下降）
- 放大器高温警告
- 许可证容量接近限制
- 配置一致性问题
- 管理接口性能缓慢
- 间歇性警报（< 5 次/小时）

升级：

- 在仪表板上记录以提高意识
- 分配给工程团队进行调查
- 如有需要，安排在下一个维护窗口进行处理
- 创建趋势分析的工单

响应 **SLA**：当天审核

警告严重性

定义：信息性，监控趋势

示例：

小区可用性 > 98% 但趋势下降
温度/电源在正常范围内但升高
资源利用率在 60-70%
配置不匹配（非关键参数）
新故障类型的首次出现

升级：

仅在仪表板上可见
无自动通知
按节奏手动审核

已清除

定义：先前活动的警报现在已解决

目的：

记录问题已解决
跟踪平均修复时间（MTTR）
启用 SLA 合规报告
识别重复问题

警报类别

连接警报

类别：外部连接

可能原因：

- S1 连接丢失 → MME/SGW 无法访问
- 回程链路故障 → IP 传输故障
- USIM 接口错误 → HSS 连接问题
- NTP 同步丢失 → 时间同步网络问题

影响：服务中断，呼叫设置失败

调查：

1. 检查设备之间的网络连接
2. 验证防火墙规则允许所需协议
3. 检查对等设备状态和错误
4. 查看网络接口统计信息

硬件与环境警报

类别：物理基础设施

可能原因：

- 高温警告 → 冷却系统降级
- 电源降级 → UPS/PSU 问题
- 风扇故障 → 冷却风扇故障
- 磁盘空间不足 → 存储接近限制
- 内存耗尽 → 进程内存泄漏

影响：潜在级联故障，数据丢失

调查：

1. 通过管理接口检查硬件状态
2. 查看温度趋势
3. 检查冷却系统操作
4. 监控内存和 CPU 利用率

软件与进程警报

类别：应用层

可能原因：

- 进程崩溃 → 软件错误或 OOM
- 高 CPU 利用率 → 性能瓶颈
- 队列过载 → 消息处理积压
- 许可证违规 → 超出容量

影响：服务降级或中断

调查：

1. 检查进程状态
2. 查看日志中的错误消息
3. 监控 CPU/内存/队列深度
4. 验证许可证状态

无线资源警报

类别：空中接口

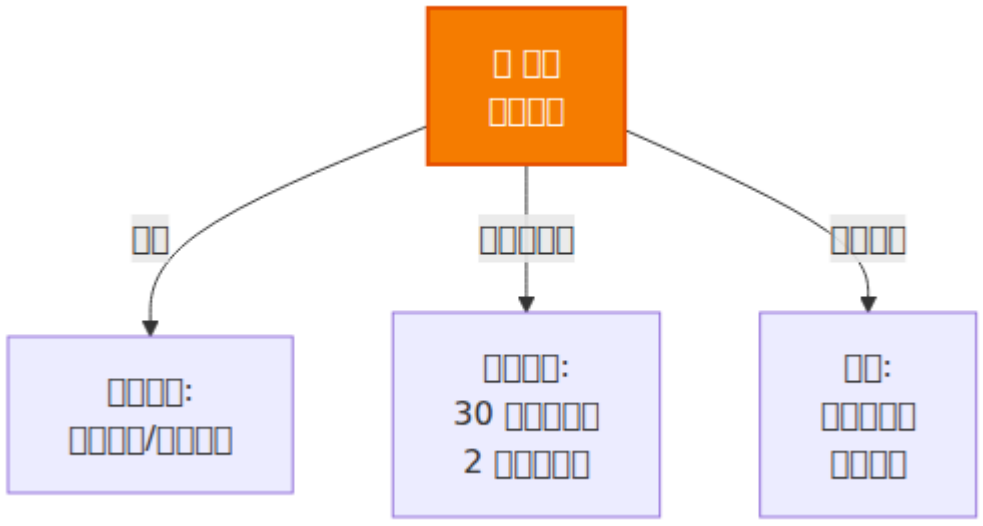
可能原因：

- 小区不可用 → 无无线覆盖/电源
- DL 资源被阻塞 → 容量耗尽
- 切换失败率高 → 邻居配置问题
- 小区质量差 → RF 干扰或路径损失

影响：用户体验降级

调查：

1. 检查小区物理参数
2. 查看邻居小区配置
3. 分析 RF 质量指标
4. 检查天线对准



□置警报

类别：系统状态

可能原因：

- 参数不匹配 → 配置不一致
- 许可证过期 → 许可问题
- 配置校验和错误 → 损坏或冲突
- 功能未获得许可 → 功能使用超出许可证

影响：功能不可用或降级

调查：

1. 查看配置更改
2. 比较当前与预期配置
3. □查许可证文件和过期情况
4. 验证软件版本兼容性

调查与故障排除

调查工作流程

第一步：查看警报详细信息

当警报触发时，首先收集信息：

要收集的内容：

- 警报 ID 和唯一标识符
- 严重性和可能原因
- 受影响的系统 DN（设备/小区/组件）
- 事件时间（发生时间）
- 持续时间（活动时间）
- 完整的警报描述和文本

工具：

- RAN Monitor Web UI → 警报页面
- Grafana → 活动警报表
- InfluxDB → 查询原始警报记录

第二步：研究可能原因

每种警报类型都有已知原因和调查方法：

文档知识：

- 诺基亚 AirScale 故障排除指南
- 以前的工单历史（类似问题）
- RAN Monitor 记录的运行手册
- 团队专业知识（主题专家）

第三步：检查相关指标

交叉关联警报与性能指标以了解影响：

```
示例："DL 资源被阻塞"警报
├─ □查 DL 资源利用率（应 > 95%）
├─ □查流量吞吐量（趋势上升？）
├─ □查呼叫设置成功率（掉线？）
├─ □查切换成功率（受影响？）
└─ □查小区可用性（故障？）
```

工具：

- Grafana → 设备特定仪表板
- Web UI → 设备详细信息页面 → 指标��分
- InfluxDB 直接查询以进行关联

第四步：与最近更改相关联

许多问题是由最近的修改引起的：

```
时间线：
├─ 配置更改（过去 4 小时）
├─ 软件升级（过去 24 小时）
├─ 功能参数调整（过去 7 天）
├─ 维护活动（过去 7 天）
├─ 网络更改（过去 7 天）
└─ 第三方更改（外部网络）
```

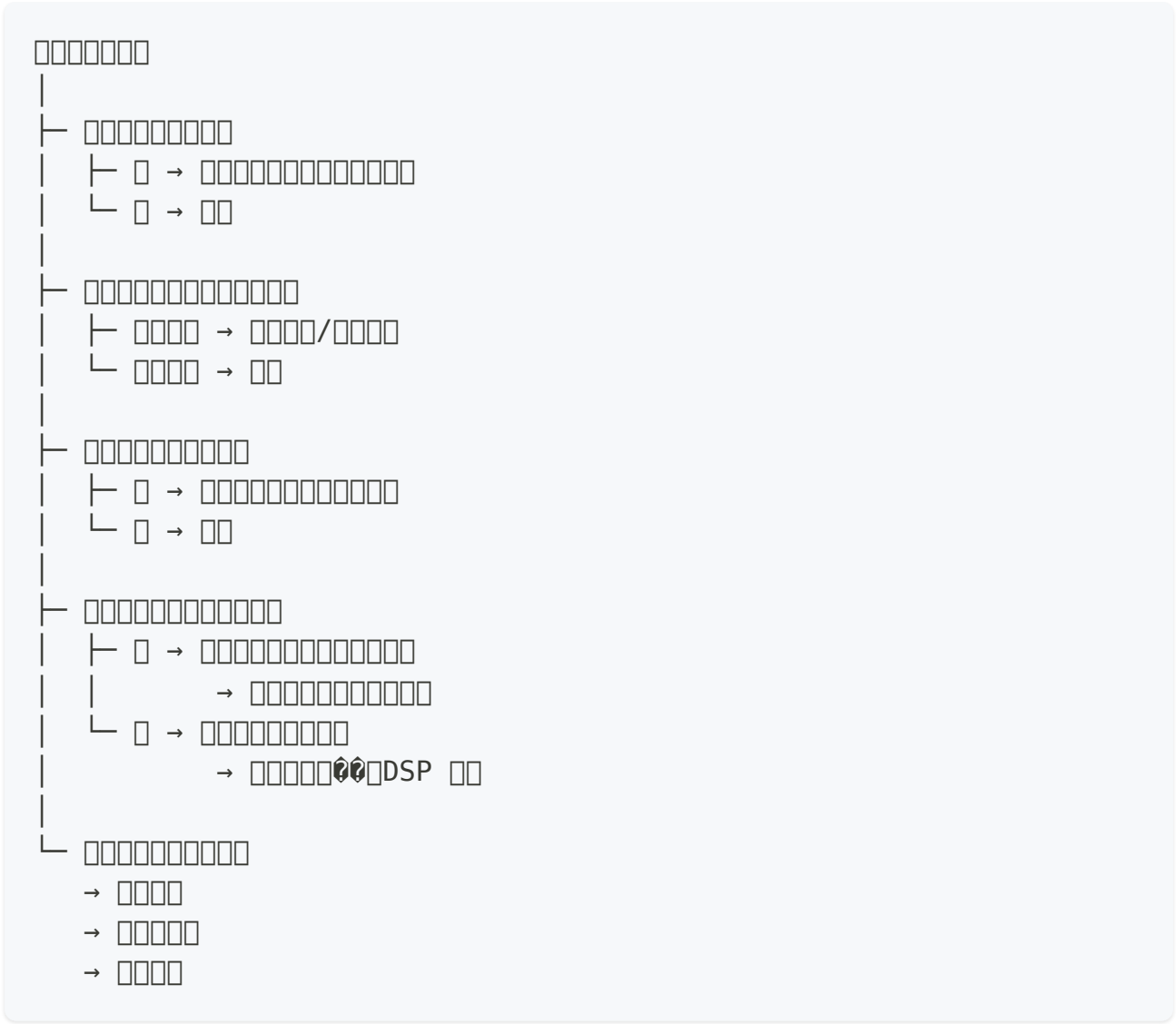
工具：

- RAN Monitor → 配置历史
- 变更管理系统
- 事件历史（以前的类似问题）
- 跨团队通知日志

第五步：诊断根本原因

根据调查，确定根本原因：

决策树示例：“小区不可用”警报



```
小区不可用警报
│
├─ 设备是否响应管理?
│   ├─ 否 → □查设备连接，如有需要重启
│   └─ 是 → 继续
│
├─ 所有小区故障还是特定小区?
│   ├─ 所有小区 → □查基带/电源硬件
│   └─ 特定小区 → 继续
│
├─ 小区是否在发射功率?
│   ├─ 否 → □查功率放大器、天线连接
│   └─ 是 → 继续
│
├─ 邻居小区是否报告该小区?
│   ├─ 是 → 其他设备将该小区视为不可用
│   │          → □查天线对准、线缆连接
│   └─ 否 → 小区因内部原因故障
│              → □查基带模□□,DSP 状态
│
└─ □查日志中的错误消息
    → 软件崩溃
    → 许可证违规
    → 参数错误
```

第六步：实施解决方案

一旦确定根本原因，实施修复：

解决方案类型：

类型	方法	示例
立即	重启/重启	设备重启以清除挂起的进程
□置	调整参数	更改切换阈值
硬件	更换/修复	更换故障的电源供应
软件	升级/修补	安装软件错误修复
网络	修复连接	恢复 BGP 路由，修复防火墙

第七步：验证解决方案

实施修复后，进行验证：

验证清单：

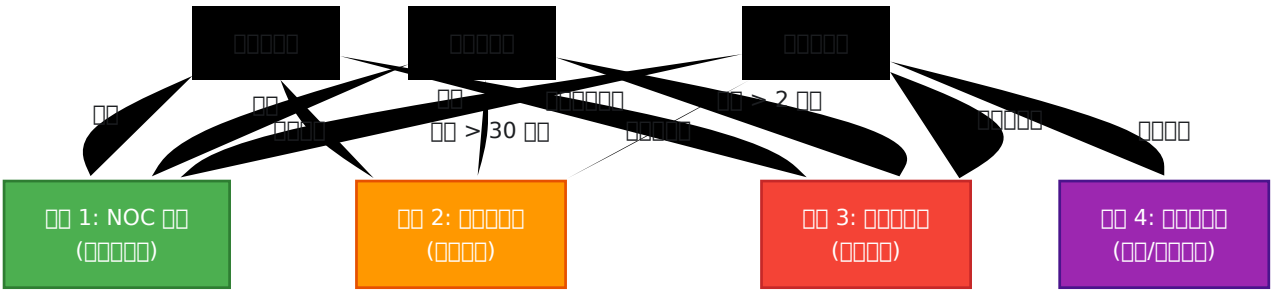
- □ RAN Monitor 中的警报已清除
- □ 相关指标已恢复正常
- □ 无二次/级联警报
- □ 性能指标恢复到基线
- □ 客户报告（如适用）已解决
- □ 系统稳定（> 30 分钟观察）

第八步：记录学习

记录发现以供将来参考：

记录：

- 根本原因和促成因素
- 解决步骤
- 花费的时间（用于 SLA 跟踪）
- 采取的预防措施
- 与团队共享的知识



升级程序

升级阶梯

升级触发条件

如果满足以下条件，则升级到级别 **2**：

- 严重警报在 15 分钟后未清除
- 主要警报在 30 分钟后未清除
- 问题超出 NOC 团队的专业知识
- 需要设备重启/重大更改
- 同时影响 > 1 个站点

如果满足以下条件，则升级到级别 **3**：

- 级别 2 在 1 小时后无法诊断
- 严重问题持续 > 30 分钟
- 怀疑硬件故障
- 检测到级联问题
- 需要供应商参与

如果确认硬件故障（级别 **4**），请联系供应商：

- 硬件故障确认（PSU、CMON 等）
- 怀疑软件错误（不可恢复崩溃）
- 许可证/激活问题
- 问题记录在供应商已知问题中
- 多个升级级别无法解决

升级沟通

升级到级别 **2** 的模板：

主题：升级 - [严重性] - [设备] - [问题]

警报时间：[2025-12-10 14:30 UTC]
持续时间：[15 分钟]
设备：[SITE_A_BS1]
问题：[小区不可用]

症状：
- 小区 A1 未响应管理
- 所有小区报告小区不可用
- 设备 ping 成功

已执行的调查：
- 验证设备连接
- □查基带模块状态
- 电源状态：正常
- 温度：正常

指标：
- 小区可用性：0%
- 小区无流量
- 控制面：已连接

初步分析：
- 可能是基带模块故障
- 或功率放大器硬件问题

建议的下一步：
- 硬件诊断
- 如果可用，模块更换
- 设备重启作为最后手段

升级链接：[仪表板链接]

解决方案跟踪

SLA 跟踪

跟踪解决时间以确保 SLA 合规：

```
警报时间线：
├─ 事件时间：14:23:45 ← 问题发生时间
├─ □测时间：14:23:47（2 秒）← 管理□测到
├─ 警报时间：14:23:50（5 秒）← 操作通知
├─ 确认时间：14:24:15（30 秒）← 操作员确认
├─ 调查：14:24:15 → 14:28:00（3.75 分钟）
├─ 解决：14:28:00 → 14:28:35（35 秒修复 + 验证）
└─ 清除时间：14:28:36 ← 警报已清除
     └─ 总持续时间：5 分钟 51 秒

SLA 指标：
├─ □测延迟：2 秒
├─ 警报到确认：30 秒
├─ 解决时间：5 分钟 51 秒
└─ SLA 状态：通过（< 15 分钟目标）
```

趋势分析

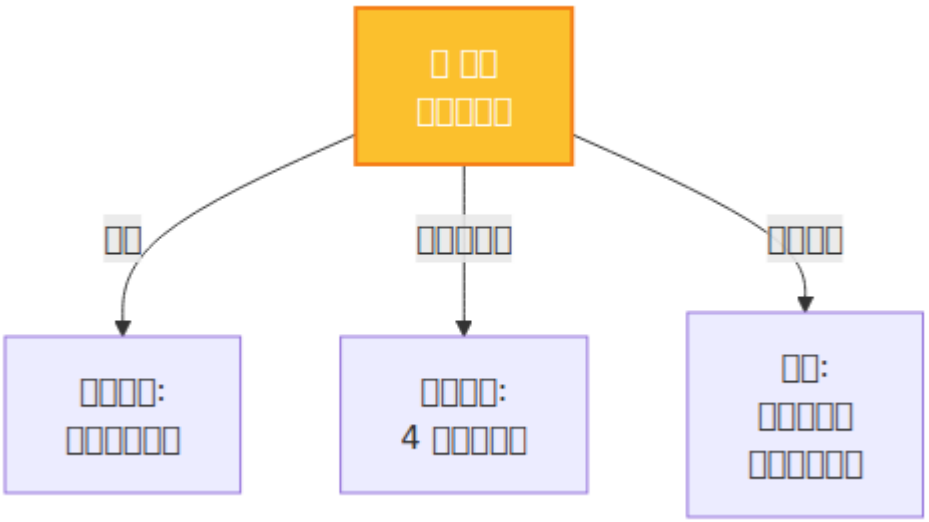
跟踪警报数据中的模式：

要问的问题：

- 我们是否看到相同的警报重复出现？
- 警报率是在增加还是减少？
- 警报是否在特定时间聚集？
- 是否同时影响多个站点？
- MTTR 是否随着时间的推移而改善？

工具：

- Grafana → 警报趋势仪表板
- 顶级警报报告（每周）



- 按设备/类型跟踪 MTTR

防止重复问题

最佳实践

操作卓越

1. 防止警报疲劳

- 设置有意义的阈值（不要过于敏感）
- 使用持续时间窗口（而不是单个峰值）
- 聚合相关警报
- 抑制已知的误报

2. 快速响应

- 保持运行手册最新
- 培训团队处理常见问题
- 对于例行重置使用自动化
- 随时准备好升级联系人

3. 高质量文档

记录每个事件
与团队共享经验
根据事件更新运行手册
交叉培训团队成员

4. 主动监控

在严重之前监控警告
容量规划的趋势分析
定期健康检查
建立性能基线

运行手册开发

每个频繁发生的警报都应有一个运行手册：

模板：

警报：[小区不可用]
概率：[高]
MTTR：[5-15 分钟]
SLA 目标：[30 分钟内解决]

症状：
- 警报：“小区不可用”
- 用户：无法连接
- 指标：小区可用性 0%

快速诊断（< 5 分钟）：
1. 设备是否响应 ping？
2. 其他小区是否正常？
3. 基带是否在运行（查日志）？

解决步骤：
步骤 1：设备连接查
   - ping 设备：ping 192.168.1.100
   - 如果没有响应 → 查网络连接

步骤 2：硬件状态
   - 查功率放大器状态
   - 查基带模块 LED
   - 查天线连接

步骤 3：重启小区
   - 通过管理接口重启小区
   - 等待 60 秒以启动
   - 验证指标恢复正常

步骤 4：如果仍然故障
   - 升级到级别 2
   - 准备设备重启
   - 通知客户

升级：
- 如果 > 15 分钟 → 升级到 [工程师姓名]
- 如果 > 30 分钟 → 升级到 [高级工程师]
- 如果硬件故障 → 联系诺基亚支持

预防：
- 定期基带固件更新

- □□□□□□□□□□
- □□□□□□□□□□

API 端点与□置管理

用于管理 **RAN** 设备□置和操作的 **REST API**

□□□□□□□□□□□□□□□□□□□□□□□□ API □□□ RAN □□□□□

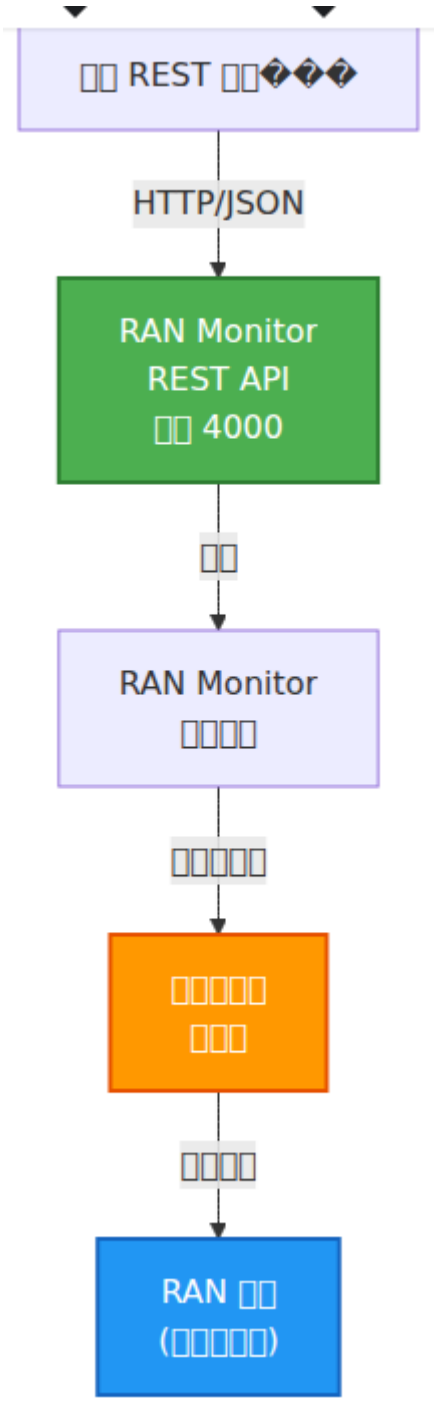
目录

1. □述
2. API 架构
3. 身份验证与访问
4. 端点参考
5. 配置管理
6. 数据检索操作
7. 常见工作流
8. 错误处理
9. API 示例

概述

RAN Monitor 提供了一个全面的 REST API，用于管理 RAN 设备配置和查询操作数据。该 API 提供了一个供应商无关的接口，抽象了底层设备通信协议。该 API 使得：

- 设备管理 - 注册、注销和监控设备
- □置查询 - 检索设备参数和系统状态
- 数据收集 - 拉取性能指标、警报和拓扑
- 会话控制 - 管理与设备的通信会话
- 网络操作 - 自动化常规管理任务



API 架构

API 特性

- **RESTful** 设计 - 标准 HTTP 方法 (GET, POST, PUT, DELETE)
- **JSON** 格式 - 所有请求和响应均为 JSON
- 供应商抽象 - 跨不同设备供应商的统一 API
- 有状态操作 - 维护设备会话和状态

- 错误处理 - 详细的错误消息和状态码
- 异步处理 - 非阻塞请求以处理长时间操作

身份验证与访问

设备注册

在进行任何操作之前，设备必须在 RAN Monitor 中注册。注册建立了 RAN Monitor 与设备之间的连接，使用每个供应商的原生身份验证机制。

注册过程：

1. 设备凭据（用户名/密码或 API 密钥）被安全存储
2. 初始连接测试验证设备是否可达
3. 设备注册并准备进行���作
4. 开始持续健康监测

API 访问控制

目前，RAN Monitor API 在管理网络内可访问。对于生产部署，请考虑：

- 身份验证方法：
  - 头部中的 API 密钥：`Authorization: Bearer <api-key>`
  - OAuth2 与身份提供者集成
  - 基于网络的访问控制（防火墙/VPN）
- 速率限制：
  - 每个客户端的限制以防止滥用
  - 每个设备的操作频率限制
- 审计日志：
  - 所有 API 调用都记录时间戳和用户/客户端信息
  - 配置更改记录前后值

端点参考

设备管理

列出所有设备

```
GET /api/v1/devices
```

响应:

```
{
  "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
    }
  ]
}
```

获取设备详情

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

响应:

```
{
  "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": "设备注册成功"
}
```

状态码:

- `200` - 注册成功
- `400` - 无效参数或设备错误
- `409` - 设备已注册
- `500` - 内部错误

注销设备

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

响应:

```
{
  "result": "Success",
  "message": "设备已注销",
  "device_id": "nokia_bs1"
}
```

会话管理

开始会话

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

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

响应:

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

会话生命周期:

- 默认超时: 24 小时
- 超时前需要保持活动
- 每 20 小时自动刷新

检查会话状态

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

响应:

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

保持会话活动

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

响应:

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

□置管理

查询□置

检索设备配置参数:

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

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

响应:

```
{
  "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"
}
```

常见□置参数:

参数	类型	示例	目的
BtsName	字符串	"SITE_A_Cell_1"	小区/基站标识符
MaxUEsServed	整数	256	最大同时用户数
CellTXPower	整数	40 (dBm)	发射功率级别
EnableCarrierAgg	布尔	true	支持载波聚合
HandoverHysteresis	整数	3 (dB)	切换���敏度

获取□置历史

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

响应:

```
{
  "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": "手动配置更新"
    }
  ]
}
```

数据检索

获取性能指标

检索性能计数器数据:

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

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

响应:

```
{
  "result": "Success",
  "metrics": {
    "timestamp": "2025-12-10T14:30:00Z",
    "measurement_interval": 300,
    "counters": [
      {
        "id": "M1C1",
        "name": "下行小区吞吐量",
        "value": 125.4,
        "unit": "Mbps",
        "cell_dn": "/BSC-1/BTS-23/Cell-1"
      },
      {
        "id": "M1C2",
        "name": "上行小区吞吐量",
        "value": 89.2,
        "unit": "Mbps",
        "cell_dn": "/BSC-1/BTS-23/Cell-1"
      }
    ]
  }
}
```

获取活动警报

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

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

响应:

```
{
  "result": "Success",
  "alarms": [
    {
      "alarm_id": "a1b2c3d4",
      "severity": "Critical",
      "probable_cause": "小区不可用",
      "specific_problem": "电源故障",
      "affected_dn": "/BSC-1/BTS-23/Cell-1",
      "event_time": "2025-12-10T14:15:30Z",
      "description": "小区 1 因电源故障而不可用"
    }
  ]
}
```

获取设备拓扑

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

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

响应:

```
{
  "result": "Success",
  "topology": {
    "device_dn": "/BSC-1",
    "managed_elements": [
      {
        "name": "BTS-23",
        "type": "基站收发信机",
        "dn": "/BSC-1/BTS-23",
        "cells": [
          {
            "name": "Cell-1",
            "type": "EUtranCell",
            "physical_cell_id": 100,
            "frequency": 2110
          }
        ]
      }
    ]
  }
}
```

健康检查

Ping 设备

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

响应:


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


获取系统健康

```
GET /api/v1/health/status
```

响应:

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

□置管理

□置数据模型

Nokia eNodeB 配置以层次结构组织:

```
/SystemFunctions
├── /BSC-1 (基站控制器)
│   └── /BTS-23 (基站收发信机)
│       ├── BtsBasics (名称、类型、位置)
│       ├── /Cell-1
│       │   ├── CellCommonData
│       │   ├── CellAdvanced
│       │   └── CarrierAggregation
│       └── /Cell-2
│           └── ...
└── /Connectivity
    ├── S1Interface
    ├── X2Interface
    └── NetworkConfiguration
```

常见□置任务

启用/禁用小区

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

可能的值：`LOCKED`，`UNLOCKED`，`SHUTTING_DOWN`

调整小区功率

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

范围：0-46 dBm（取决于设备）

□置切换滞后

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

单位: dB, 典型范围: 0-8 dB

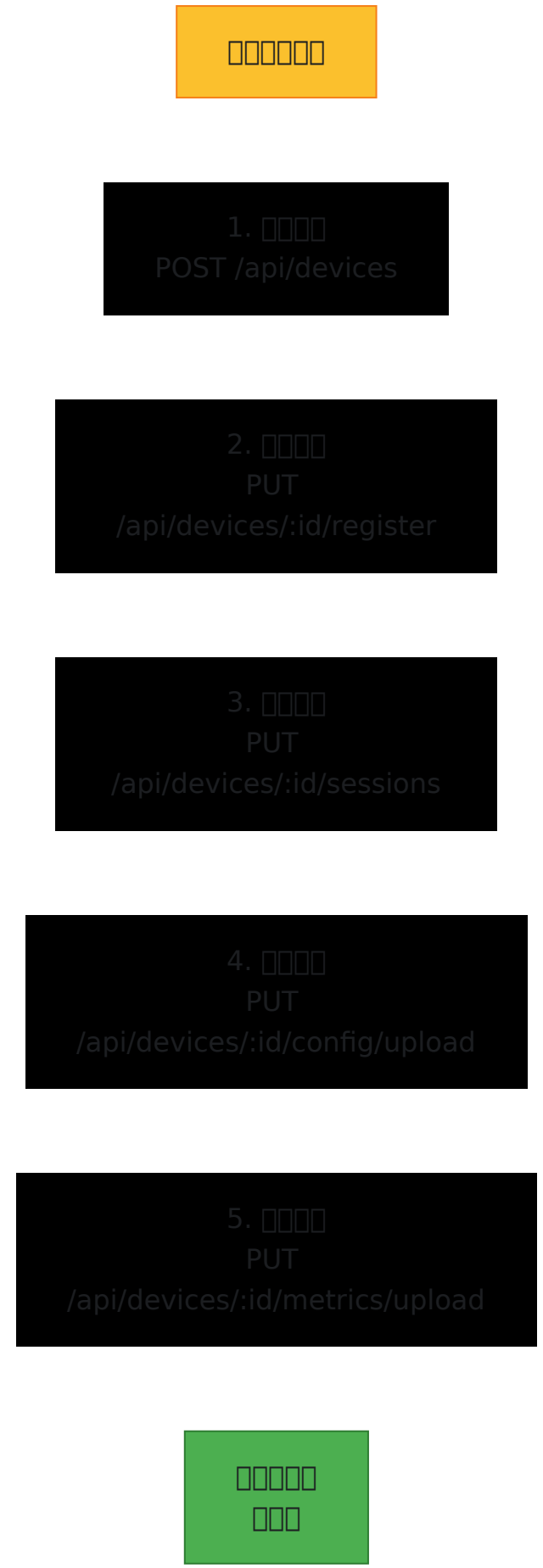
设置最大连接用户数

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

取决于设备的限制

常见工作流

工作流 1：设备入驻



开始：新设备
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
成功：设备
可操作

示例：

```
# 1. 创建设备条目
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. 与设备注册
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. 开始会话
curl -X PUT
http://localhost:4000/api/v1/devices/site_a_bs1/sessions \
  -H "Content-Type: application/json" \
  -d '{"session_id": "session_001"}'

# 4. 获取配置
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}]
    }
  }'
```

工作流 **2**: □置更新

```
# 1. 查询当前值
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. 修改参数
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. 验证更改
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"]'
```

工作流 3：性能监控

```
# 持续监控循环（示例脚本）
#!/bin/bash

DEVICE="site_a_bs1"
INTERVAL=300  # 5 分钟

while true; do
  # 拉取指标
  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}]
      }
    }')

  # 提取关键指标
  DL=$(echo $METRICS | jq '.metrics.counters[] |
select(.id=="M1C1") | .value')
  CELLS=$(echo $METRICS | jq '.metrics.counters | length')

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

  # 查警报
  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 "警告: $ALARMS 个活动警报"
  fi
```

```
  sleep $INTERVAL
done
```

错误处理

HTTP 状态码

代码	意义	示例
`200`	成功	配置已检索
`201`	创建	设备已注册
`400`	错误请求	无效的 JSON 或参数
`401`	未授权	缺少/无效的 API 密钥
`404`	未找到	设备不存在
`409`	冲突	设备已注册
`500`	服务器错误	数据库连接失败
`503`	不可用	维护模式

错误响应格式

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

常见错误

设备未注册:

```
{
  "error": {
    "code": "NOT_REGISTERED",
    "message": "设备必须在操作之前注册",
    "solution": "首先调用 PUT /api/devices/:id/register"
  }
}
```

会话过期:

```
{
  "error": {
    "code": "SESSION_EXPIRED",
    "message": "设备会话已过期",
    "solution": "调用 PUT /api/devices/:id/sessions 以开始新会话"
  }
}
```

□置参数无效:

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

API □□

Python □□□□□

```
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):
        """□□□□□"""
        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):
        """□□□□□□"""
        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"):
        """□□□□□□"""
        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):
        """索性能指标"""
        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()

# 例用法
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"注册: {result}")

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

# 更新参数
update = client.set_config(
    "site_a_bs1",
    "/BSC-1/BTS-23/Cell-1/CellAdvanced/CellTXPower",
    "38",
    "integer"
```

```
)
print(f"更新: {update}")
```

cURL 示例

注册设备:

```
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"
  }'
```

获取设备状态:

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

查询□置:

```
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 '.'
```

置归档指南

AirScale 置的自动版本控制和历史跟踪

概述

配置归档系统自动跟踪和版本化所有 AirScale 基站配置文件。配置快照不再存储在 InfluxDB 中，而是作为带时间戳的 XML 文件保存在服务器上，提供完整的配置更改审计跟踪。

主要特性

- 自动版本控制 - 仅在配置更改时创建新版本
- 每小时轮询 - 每小时检查一次配置更改（可配置）
- 变更检测 - 智能比较检测实际更改，忽略空白
- 基于大小的限制 - 每个设备最大存储 100 MB（保持 ~690 个版本）
- **Web** 界面 - 浏览、下载和管理配置版本
- 快速访问 - 基于文件的存储以实现即时检索
- 零 **InfluxDB** 负载 - 配置不再存储在时间序列数据库中
- 自动清理 - 当达到大小限制时删除旧版本

工作原理

轮询计划

从每个注册的 AirScale 基站轮询配置：

- 间隔：每 1 小时（默认）
- 首次轮询：应用启动时立即进行
- 变更检测：与先前版本的内容进行��较
- 存储：仅在更改或首次时保存

存储位置

配置存储在 RAN Monitor 服务器文件系统中：

```
priv/airscale_configs/
└── <airscale-name>/
    ├── current.xml                                          # 最新配置
    ├── ONS-Lab-Airscale_config_20251230_143522.xml    # 版本来自 12 月
30 日，下午 2:35
    ├── ONS-Lab-Airscale_config_20251229_120000.xml    # 先前版本
    ├── ONS-Lab-Airscale_config_20251228_093045.xml    # 更旧版本
    └── ...                                                  # 版本保留直到
100 MB 限制
```

文件名格式：`<AirScaleName>_config_YYYYMMDD_HHMMSS.xml`

目录命名：AirScale 名称经过清理（特殊字符替换为下划线，小写）

版本管理

- 最新版本：始终可用作 `current.xml`
- 历史版本：带时间戳的文件显示配置更改的时间
- 自动清理：当达到 100 MB 大小限制时删除最旧版本
- 手动清理：通过 Web UI 删除特定版本（`current.xml` 除外）
- 存储保护：基于大小的限制防止无限制磁盘使用
- 灵活保留：如果文件较小，则保留更多版本，如果文件较大，则保留较少版本

存储保护

基于大小的存储限制

为防止无限制的磁盘使用，系统使用 基于大小的限制 而不是版本计数：

- 最大大小：每个设备 100 MB（可配置）
- 自动清理：当超过大小限制时删除最旧版本

清理时机：每次保存新配置版本时
受保护文件：始终保留 current.xml 和至少一个版本
灵活性：每个版本约 145KB 时保留 ~690 个版本，如果文件较小则更多

工作原理

当保存新配置版本时：

1. 保存新版本 - 配置写入 <AirScale>_config_YYYYMMDD_HHMMSS.xml
2. 更新当前 - current.xml 更新为最新配置
3. 计算大小 - 系统计算所有版本文件的总大小
4. 清理旧版本 - 如果总大小 > 100 MB，删除最旧版本直到低于限制
5. 记录活动 - 删除记录并释放空间

示例场景

初始状态：使用 95 MB（655 个版本，每个 145KB）

```
├── ONS-Lab-Airscale_config_20240101_100000.xml  <- 最旧（145KB）
├── ONS-Lab-Airscale_config_20240102_100000.xml  (145KB)
├── ... (还有 653 个版本)
└── ONS-Lab-Airscale_config_20251230_100000.xml  <- 最新（145KB）
```

在 2025-12-31 10:00:00 □测到新配置（145KB）

操作：

1. 保存：ONS-Lab-Airscale_config_20251231_100000.xml (145KB)
2. 现在总大小：95 MB + 145KB = 95.14 MB（仍在 100 MB 限制内）
3. 不需要删除
4. 最终：656 个版本，使用 95.14 MB

稍后：大配置更改（添加新功能，文件现在为 500KB）

操作：

1. 保存：ONS-Lab-Airscale_config_20251231_150000.xml (500KB)
2. 现在总大小：95.14 MB + 500KB = 95.64 MB（仍在限制内）
3. 不需要删除
4. 最终：657 个版本，使用 95.64 MB

经过许多更改后：接近限制

当前状态：99.8 MB（685 个版本）

新配置：200KB

1. 保存新版本
2. 总大小将为：100 MB（超过限制）
3. 删除最旧版本直到总大小 < 100 MB
4. 删除记录：“删除 3 个版本，释放 435 KB”
5. 最终：682 个版本，使用 99.6 MB

存储保证

自动清理确保：

- 有界存储：每个设备限制为最大 100 MB
- 无意外：存储不会无限增长
- 生产安全：无需手动干预
- 灵活历史：小配置保留更多版本，大配置保留较少版本
- 始终可用：始终保留至少一个版本

监控存储

检查所有设备的存储使用情况：

```
# 总存储使用情况
du -sh priv/airscale_configs/
# 示例：215M（3 个设备平均 70 MB）

# 每个设备的存储
du -sh priv/airscale_configs/*/
# 示例：
# 95M     priv/airscale_configs/ons-lab-airscale/
# 68M     priv/airscale_configs/sector-1/
# 52M     priv/airscale_configs/sector-2/

# 查是否有设备接近限制
find priv/airscale_configs -maxdepth 1 -type d -exec du -sm {} \;
| awk '$1 > 90 {print $2 " is at " $1 "MB (approaching 100MB
limit)"}'
```

使用置归档页面

访问置归档

Web UI：导航到 **Nokia → **置归档

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

界面概述

配置归档页面有三个主要部分：

1. 基站选择器

- 网格视图 - 显示所有注册的 AirScale 设备
- 版本计数 - 每个设备存储的配置版本数量
- 选择 - 点击设备查看其配置历史
- 视觉指示器 - 选定设备以蓝色高亮显示

2. 版本历史表

显示所选基站的所有配置版本：

列	描述
时间戳	配置保存的时间（UTC）
文件名	版本文件名（例如，`config_20251230_143522.xml`）
大小	文件大小（KB 或 MB）
年龄	版本创建的时间（例如，“2小时前”，“3天前”）
操作	下载或删除按钮

排序：最新版本首先出现（按时间戳降序）

当前□置：`current.xml` 文件无法被删除（安全措施）

3. 存储信息

摘要面板显示：

- 总版本 - 存储的配置版本数量
- 总大小 - 所有版本的总大小
- 存储路径 - 服务器上的文件系统位置

常见操作

下载□置

目的：检索特定配置版本以进行审查、备份或比较

步骤：

1. 导航到配置归档页面
2. 选择所需的基站
3. 在表中找到您想要的版本

4. 点击 下载 按钮
5. 文件下载格式为：`<AirScaleName>_config_YYYYMMDD_HHMMSS.xml`（与存储的文件名匹配）

用例：

- 创建离线备份
- 比较时间戳之间的配置
- 回滚到先前已知的良好配置
- 分析随时间变化的配置漂移

删除旧版本

目的：删除过时的配置版本以释放存储空间

步骤：

1. 导航到配置归档页面
2. 选择基站
3. 找到要删除的版本
4. 点击 删除 按钮
5. 在弹出对话框中确认删除
6. 版本将被永久删除

重要说明：

- 无法删除 `current.xml`（最新版本受保护）
- 删除是立即的，无法撤销
- 手动删除不会影响自动清理设置

比较配置

目的：确定两个配置版本之间的变化

手动比较：

1. 下载您要比较的两个版本
2. 使用 XML 差异工具（例如，`xmldiff`，`Beyond Compare`，`WinMerge`）

3. 审查差异以了解更改

使用命令行的示例：

```
# 下载两个版本
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

# 使用 diff 进行比较
diff ONS-Lab-Airscale_config_20251229_120000.xml ONS-Lab-Airscale_config_20251230_143522.xml

# 或使用 xmldiff 以获得更清晰的输出
xmldiff ONS-Lab-Airscale_config_20251229_120000.xml ONS-Lab-Airscale_config_20251230_143522.xml
```

□置管理工作流程

审计跟踪调查

场景：需要确定配置何时更改

步骤：

1. 打开配置归档
2. 选择受影响的基站
3. 审查版本时间戳
4. 下载相关版本
5. 比较以确定确切的更改
6. 与性能问题或警报相关联

示例：

版本时间线：
- ONS-Lab-Airscale_config_20251230_143522.xml (143KB) - 最近
- ONS-Lab-Airscale_config_20251228_091045.xml (142KB) - 2 天前
- ONS-Lab-Airscale_config_20251225_180000.xml (142KB) - 5 天前

分析：
- 大小从 142KB 增加到 143KB 在 12 月 30 日
- 比较 12 月 28 日与 12 月 30 日以找出添加了什么
- 查时间是否与警报峰值相关

置回滚

场景：最近的配置更改导致问题，需要恢复先前版本

步骤：

1. 确定已知的良好配置版本
2. 从配置归档下载该版本
3. 导航到配置管理页面（Web UI）
4. 上传下载的配置 → 接收计划 ID
5. 验证计划 → 检查错误
6. 如果验证成功，激活计划
7. 监控设备的稳定性
8. 验证新配置在下次轮询后出现在归档中

安全检查清单：

- ✓ 下载了正确的先前版本
- ✓ 在激活前验证了计划
- ✓ 与运营团队协调
- ✓ 在维护窗口期间安排
- ✓ 监控工具准备好进行验证

基线置管理

场景：维护“黄金配置”以实现标准化

最佳实践：

1. 创建并验证基线配置
2. 应用到参考设备
3. 在下次轮询后从配置归档下载
4. 作为模板外部存储
5. 用于部署新设备
6. 定期审查和更新基线

技术细节

变更检测算法

系统使用智能内容比较来避免误报：

规范化过程：

1. 删除前导/尾随空白
2. 在 XML 标签之间折叠空白
3. 规范化内部空白
4. 比较结果规范化的内容

好处：

格式更改不会触发新版本
仅实际配置更改会创建版本
减少存储需求
提供有意义的变更历史

示例：

```
<!-- 这些被视为相同 -->

<!-- 版本 1（带额外空白） -->
<parameter>
    <name>cellId</name>
    <value>1</value>
</parameter>

<!-- 版本 2（紧凑） -->
<parameter><name>cellId</name><value>1</value></parameter>
```

存储要求

典型□置大小：~145 KB 每个版本（基于实际 AirScale 配置）

容量规划：

设备数量	每个设备最大大小	保留的版本（每个 **145KB**）	最大总存储
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

增长特性：

- 每个设备的最大存储：100 MB（可配置）
- 典型保留版本：~690（每个 145KB）
- 如果□置从未更改：最小增长（仅 current.xml 为 145KB）
- 如果□置频繁更改：增长在 100 MB 限制处停止
- 自适应：小配置保留更多版本，大配置保留较少版本

自动保护：

- 当达到大小限制时删除旧版本
- 无需手动干预
- 每个设备的存储使用严格限制
- 始终保留至少一个版本

保留政策

默认设置：

- 每个设备最大 **100 MB** 存储
- 超过限制时自动删除最旧版本
- 始终保留 `current.xml`（免于清理）
- 每次保存新版本时进行清理
- 始终保留至少一个版本文件

自定义保留：

存储限制在 ConfigStorage 模块中配置：

```
# 在 lib/ran_monitor/nokia/airscale/config_storage.ex 中
# 文件顶部的模块属性
@max_storage_bytes 100 * 1024 * 1024  # 默认 100 MB

# 更改为不同的限制：
@max_storage_bytes 50 * 1024 * 1024   # 50 MB（保留 ~345 个版本）
@max_storage_bytes 200 * 1024 * 1024  # 200 MB（保留 ~1380 个版本）

# 清理函数使用此默认值
def cleanup_old_versions(airscale_name, max_size_bytes \
@max_storage_bytes)
```

修改后，重新编译：

```
mix compile
# 重启 RAN Monitor 以应用更改
```

轮询□置

默认间隔： 1 小时（3,600,000 毫秒）

要更改轮询间隔：

编辑 `lib/ran_monitor/nokia/airscale/manager.ex`：

```
defp schedule_get_airscale_config do
  # 每 1 小时拉取配置（3,600,000 ms）
  Process.send_after(self(), :get_airscale_config, 3_600_000)
end
```

常见间隔：

- 30 分钟：`1_800_000`
- 1 小时：`3_600_000`（默认）
- 2 小时：`7_200_000`
- 4 小时：`14_400_000`
- 24 小时：`86_400_000`

更改后，重新编译并重启 RAN Monitor。

故障排除

没有□置版本显示

症状：

- 配置归档页面显示“0 个版本”
- 选定设备显示空表

可能原因：

1. 设备未注册

- 检查基站页面
- 验证设备显示“已注册”状态
- 查看应用日志以获取注册错误

2. 会话未激活

   - 检查设备详细视图
   - 确保会话状态为“活动”
   - 查看会话时间戳

3. 置尚未轮询

   - 等待第一次每小时轮询周期
   - 或手动触发：`Kernel.send(Process.whereis(RanMonitor.Nokia.Airscale.Manager), :get_airscale_config)`

4. 存储路径问题

   - 检查 `priv/airscale_configs/` 目录是否存在
   - 验证 RAN Monitor 是否具有写权限
   - 查看应用日志以获取文件系统错误

下载返回 **404** 错误

症状：

- 点击下载显示“找不到配置文件”
- 浏览器显示 404 错误

可能原因：

1. 文件路径不匹

   - 目录名称经过清理（小写，特殊字符替换）
   - 验证 `priv/airscale_configs/` 中的实际目录名称

2. 文件已删除

- 检查文件是否已从文件系统手动删除
- 刷新配置归档页面以更新列表

3. 权限问题

- 验证 Web 服务器进程是否可以读取配置文件
- 检查配置目录的文件权限

解决方案：

```
# 查目录是否存在
ls -la priv/airscale_configs/

# 验证文件是否存在
ls -la priv/airscale_configs/<device-name>/

# 如果需要，修复权限
chmod 755 priv/airscale_configs/
chmod 644 priv/airscale_configs/*/*.xml
```

置未更新

症状：

- 仅存在 `current.xml`，没有新版本
- 即使更改后版本计数仍保持为 1

可能原因：

1. 置未更改

- 系统仅在内容更改时创建版本
- 查看日志：“配置未更改，未建新版本”

2. 轮询未运行

- 检查应用日志以获取轮询消息
- 验证管理器进程是否正在运行
- 检查配置检索期间是否有错误

3. 变更检测过于严格
    - 仅空白更改被忽略（按设计）
    - 验证实际参数值是否更改

验证：

```
# 查日志以获取配置轮询信息
grep "process_configuration" <log-file>

# 手动触发配置拉取
# 在 IEx 控制台中：
Kernel.send(Process.whereis(RanMonitor.Nokia.Airscale.Manager),
:get_airscale_config)
```

最佳实践

定期备份

建议：创建关键配置的外部备份

自动备份脚本示例：

```
#!/bin/bash
# backup-configs.sh - 每日配置备份到外部存储

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

# 创建带日期的备份目录
mkdir -p "$BACKUP_DIR/$DATE"

# 复制所有配置
rsync -av "$CONFIG_DIR/" "$BACKUP_DIR/$DATE/"

# 保留最近 30 天
find "$BACKUP_DIR" -type d -mtime +30 -exec rm -rf {} +

echo "备份完成: $BACKUP_DIR/$DATE"
```

使用 **cron** 安排:

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

变更文档

最佳实践：记录配置更改的原因

建议流程:

1. 在进行配置更改之前，���录原因
2. 在配置旁边创建变更日志文件
3. 包括：日期、设备、变更、理由、批准者

示例变更日志:

```
# config_changes.log

2025-12-30 14:35:22 - ONS-Lab-Airscale
更改者: John Smith
原因: 提高小区功率以改善 1 区覆盖
参数: txPower 从 40dBm 更改为 43dBm
已验证: 是
激活时间: 2025-12-30 14:40:00
结果: 覆盖改善，未观察到降级

2025-12-28 09:10:45 - ONS-Lab-Airscale
更改者: Jane Doe
原因: 在新站点部署后更新邻居小区列表
参数: 添加邻居小区 10，11，12
已验证: 是
激活时间: 2025-12-28 09:15:00
结果: 切换正常工作
```

存储监控

建议：定期监控磁盘使用情况

检查存储使用情况：

```
# 配置归档的总大小
du -sh priv/airscale_configs/

# 每个设备的大小
du -sh priv/airscale_configs/*/

# 每个设备的版本数量
find priv/airscale_configs/ -name "*.xml" | \
  sed 's|/[^/]*\.xml||' | uniq -c
```

设置警报：

- 如果总大小超过阈值（例如，500MB）则警报
- 如果任何设备的版本计数异常高则警报
- 如果磁盘空间低于 10% 可用则警报

与□置管理的集成

配置归档系统与配置管理页面协同工作：

工作流程集成

下载当前□置：

- 使用配置归档获取 `current.xml`
- 或使用配置管理的“下载”按钮（触发立即拉取）

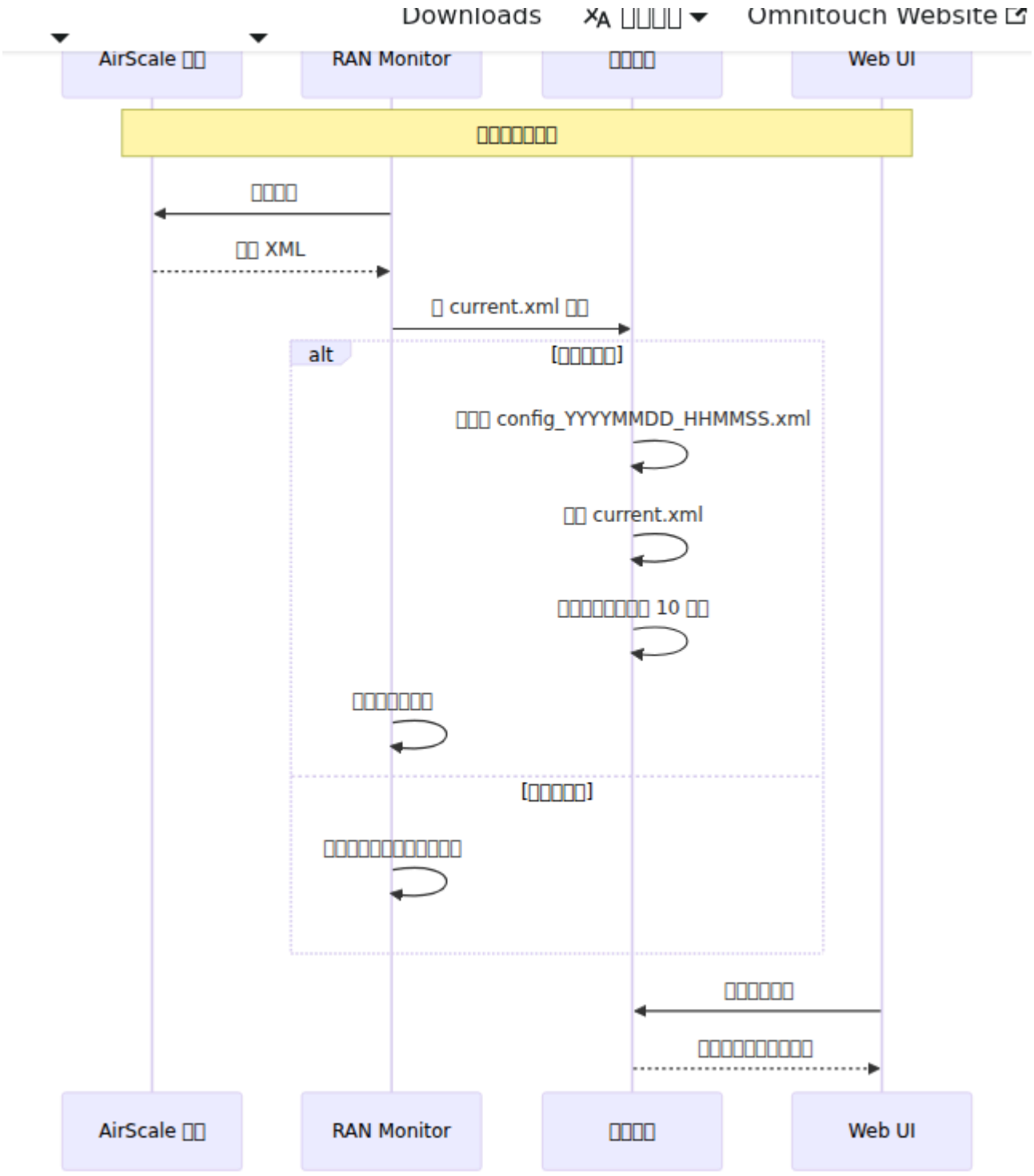
上传修改后的□置：

- 使用配置管理页面
- 上传 → 验证 → 激活工作流程
- 新版本在下次轮询后出现在归档中

回滚过程：

- 从归档下载先前版本
- 通过配置管理上传
- 按照验证 → 激活工作流程进行

数据流



API 访问

虽然配置归档主要通过 Web UI 访问，但也可以通过直接 HTTP 请求下载配置。

下载端点

当前□置:

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

特定版本:

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

注意: URL 中的 AirScale 名称必须与清理后的目录名称匹配（小写，特殊字符用下划线替换）

程序化访问

列出版本（从 **IEx** 控制台）:

```
# 获取设备的所有版本
RanMonitor.Nokia.Airscale.ConfigStorage.list_config_versions("ONS-Lab-Airscale")

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

# 查版本计数
RanMonitor.Nokia.Airscale.ConfigStorage.count_versions("ONS-Lab-Airscale")
# 返回: 655

# 获取最大存储大小设置
RanMonitor.Nokia.Airscale.ConfigStorage.max_storage_bytes()
# 返回: 104857600 (100 MB 以字节为单位)

# 获取当前存储使用情况
RanMonitor.Nokia.Airscale.ConfigStorage.get_storage_usage("ONS-Lab-Airscale")
# 返回: 99614055 (字节)

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

# 手动清理 (保持在 50 MB 以下)
RanMonitor.Nokia.Airscale.ConfigStorage.cleanup_old_versions("ONS-Lab-Airscale", 50 * 1024 * 1024)
# 返回: {:ok, 345, 50000000} - 删除 345 个版本, 释放 50MB

# 使用默认值清理 (100 MB)
RanMonitor.Nokia.Airscale.ConfigStorage.cleanup_old_versions("ONS-Lab-Airscale")
# 返回: {:ok, 0, 0} - 如果在限制内, 则不需要清理
```

另见

- **Web UI** 指南 - 完整控制面板参考
- **AirScale** 置指南 - 基站设置
- 常见操作指南 - 日常管理任务
- 数据保留政策指南 - 存储管理

数据保留政策指南

概述

RAN Monitor 应用程序现在包括一个全面的 数据保留政策 系统，允许您管理性能指标、配置数据和警报记录在 InfluxDB 中存储的时间。本指南涵盖了您需要了解的有关管理数据保留的所有内容。

快速开始

访问保留政策仪表板

1. 导航到 控制面板：`https://localhost:9443`
2. 在导航菜单中点击 数据保留
3. 查看并管理所有配置的 eNodeB 的保留设置

设置自定义保留期限

1. 在列表中找到 eNodeB
2. 更新“保留期限”字段（以小时为单位）
3. 设置会立即保存
4. 如果留空，则回退到全局默认值

清理旧数据

1. 点击 清理旧数据 按钮以删除超过保留期限的记录
2. 或点击 清除所有数据 以删除该 eNodeB 的所有记录（请谨慎使用！）

截图

□□□□□□□□□□□ eNodeB □□□□□□□□□□

□ 特性

全局保留设置

- 默认保留期限：720 小时（30 天）
- 可□置：在 `config/config.exs` 中更改
- 回退：应用于没有自定义设置的所有 eNodeB

每个 **eNodeB** 的保留

- 覆盖全局：为特定 eNodeB 设置自定义保留
- 数据库存储：持久化在 `airscales` 表中
- 实时：立即生效

自动清理

定期: 每小时自动运行
后台工作者: RanMonitor.Data.RetentionCleanupWorker
每个 eNodeB: 尊重各自的保留设置
记录: 所有清理操作都有日志记录以供审计

数据可见性

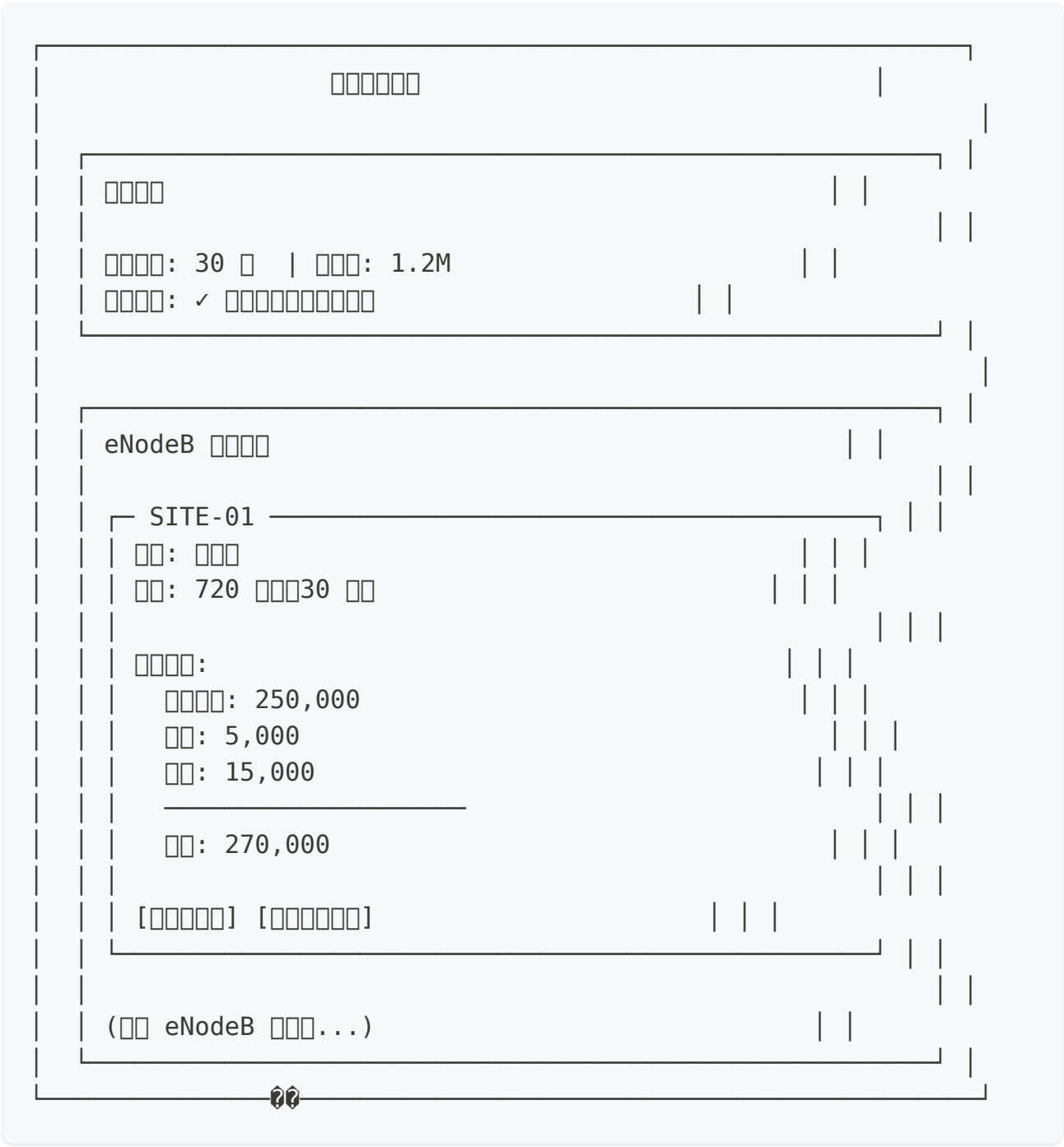
记录计数: 查看每种测量类型的记录数量：
性能指标
配置
警报
总摘要: 查看每个 eNodeB 的总记录
实时: 在页面刷新时更新

□ □□□□

□□□□□

□□□□□□□□□□□□□□□□□ eNodeB □□□□□□□□□□

□□□□□



□□□□□

- □□ **(✓)**: eNodeB □□□□□□□□□□□□
- □□ **(✗)**: eNodeB □□□□□□□□
- □□: □□□□□□□□ eNodeB □□□

操作按钮

按钮	操作	效果
清理旧数据	删除旧记录	删除超过保留期限的记录
清除所有数据	完全清除	删除该 eNodeB 的所有记录（⚠ 请谨慎使用！）
刷新	更新显示	重新获取记录计数和设置

⚙ □置

全局保留□置

编辑 `config/config.exs`：

```
config :ran_monitor,
  ecto_repos: [RanMonitor.Repo],
  generators: [context_app: :ran_monitor],
  data_retention_hours: 720  # 30 天，根据需要调整
```

支持的时间值

周期	小时	天	推荐用途
1 小时	1	0.04	仅用于测试
1 天	24	1	短期指标
7 天	168	7	每周报告
14 天	336	14	双周报告
30 天	720	30	月度报告（默认）
90 天	2160	90	长期趋势
180 天	4320	180	半年报告
1 年	8760	365	年度报告

环境变量

可在运行时覆盖：

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

工作原理

数据保留流程

```
1. 数据插入
   ├─ 性能指标 → InfluxDB
   ├─ 配置数据 → InfluxDB
   └─ 警报 → InfluxDB

2. 自动清理（每小时）
   ├─ RetentionCleanupWorker 触发
   ├─ 对每个 eNodeB:
   │  ├─ 获取有效保留（每个 eNodeB 或全局）
   │  ├─ 计算截止时间戳
   │  └─ 删除超过截止时间的记录
   └─ 记录结果

3. 手动清理（按需）
   ├─ 用户在 UI 中点击按钮
   ├─ 应用保留政策
   ├─ 记录立即删除
   └─ 显示成功/错误通知

4. 监控
   └─ 在 UI 中显示记录计数
```

保留逻辑

每个 **eNodeB** 的有效保留:

```
effective_retention = case airscale.retention_hours do
  nil -> Config.data_retention_hours()      # 使用全局（720h）
  hours -> hours                             # 使用每个 eNodeB 的自定义值
end
```

示例:

- 全局默认: 720 小时（30 天）

- eNodeB "SITE-01" 自定义: 168 小时 (7 天)
- eNodeB "SITE-02" 自定义: nil → 使用全局 720 小时

清理过程

```
截止时间戳 = 现在 - (retention_hours * 3600 秒)

以 30 天保留为例:
├─ 当前: 2025-12-11 10:00:00
├─ 保留: 720 小时 (30 天)
├─ 截止: 2025-11-11 10:00:00
└─ 删除所有时间戳 < 截止的记录
```

监控与日志记录

日志条目

系统记录所有保留活动。查找:

```
[RetentionCleanupWorker] 启动保留清理工作者
[RetentionCleanupWorker] 正在清理 SITE-01 的数据 (保留: 720h)
[RetentionCleanupWorker] 删除了 SITE-01 的 15,000 条记录
[RetentionCleanupWorker] 清理周期完成: 5 成功, 0 失败, 75,000 总共删除
```

监控记录计数

实时可见性:

1. 打开数据保留仪表板
2. 查看每个 eNodeB 每种测量的当前记录计数
3. 点击“刷新”以更新计数

历史追踪:

- 检查应用日志以获取清理摘要

- 随时间监控 InfluxDB 磁盘使用情况
- 根据记录计数增长设置警报

高级用法

编程访问

在代码中使用保留政策服务：

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

# 获取 eNodeB 的有效保留
airscale = Nokia.get_airscale!(1)
hours = RetentionPolicy.get_retention_hours(airscale)
# => 720（或自定义值如果设置）

# 获取 eNodeB 的记录计数
counts = RetentionPolicy.get_record_counts("SITE-01")
# => %{"PerformanceMetrics" => 250000, "Configuration" => 5000,
"Alarms" => 15000}

# 获取总记录
total = RetentionPolicy.get_total_record_count("SITE-01")
# => 270000

# 手动删除旧记录
{:ok, deleted_count} = RetentionPolicy.delete_old_records("SITE-
01", 720)
# => {:ok, 50000}  (删除了 50k 条记录)

# 清除 eNodeB 的所有记录
{:ok, deleted_count} = RetentionPolicy.clear_all_records("SITE-
01")
# => {:ok, 270000}  (所有 270k 条记录已删除)
```

调整清理间隔

编辑 `lib/ran_monitor/data/retention_cleanup_worker.ex`:

```
# 从 1 小时（3600000ms）更改为 30 分钟（1800000ms）
@cleanup_interval_ms 1800000  # 30 分钟
```

然后重新编译:

```
mix compile
```

数据库级查询

直接查看保留设置:

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

通过数据库更新保留:

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

最佳实践

保留期限选择

短期（**< 7** 天）

- 用于：测试、暂存环境
- 不推荐用于：生产
- 风险：可能删除重要的历史数据

标准（**7-30** 天）

- 用于：具有典型存储的生产部署
- 最适合：大多数用例
- 平衡：良好的历史记录与可管理的存储

长期（**> 30** 天）

- 用于：趋势分析、合规要求
- 成本：更高的存储需求
- 好处：扩展的历史数据

按用例推荐

用例	保留	理由
每日报告	7-14 天	每周审查周期
每周报告	30-60 天	月度摘要
每月报告	90 天	季度分析
趋势分析	180-365 天	长期模式
合规	按需	法律/监管

存储考虑

估算存储需求：

- 1000 条记录 ≈ 1-5 KB（取决于测量类型）
- 100 万条记录 ≈ 1-5 GB
- 保留期限 × 收集速率 = 总存储

使用以下命令监控增长：

```
# □查 InfluxDB 存储桶大小
influx bucket list

# 或□查磁盘使用情况
df -h /path/to/influxdb/data
```

□ 安全与合规

数据隐私

- 默认情况下不加密 存储
- 网络访问 通过 InfluxDB 安全控制
- 访问日志 可在应用日志中查看

合规

- 审计跟踪：所有清理操作都有时间戳记录
- 数据完整性：软删除，应用层没有硬删除
- 保留证明：日志显示保留/删除的内容

建议

1. 为生产启用 **InfluxDB** 身份验证
2. 定期监控清理日志
3. 仔细设置保留 以平衡合规与存储
4. 在批量操作前备份 关键数据
5. 首先在暂存环境中测试��留政策

故障排除

问题：清理未运行

症状：

- 超过保留期限的记录仍然存在
- 没有清理日志条目

解决方案：

1. 检查应用是否在运行：`ps aux | grep mix`
2. 验证 `RetentionCleanupWorker` 是否已启动：
   - 检查日志中的 `[RetentionCleanupWorker] 启动`
3. 检查 InfluxDB 连接：
   - 访问 InfluxDB 状态页面：`https://localhost:9443/nokia/influx`
4. 验证保留设置是否已配置：
   - 检查 `config/config.exs` 中的 `data_retention_hours`

问题：手动清理失败

症状：

- 点击“清理旧数据”时出现错误消息
- 记录未删除

解决方案：

1. 检查 InfluxDB 是否可访问：
   - 在仪表板中测试连接
2. 验证记录计数是否准确：
   - 点击“刷新”以更新
3. 检查应用日志中的错误：
   - 查找 `[RetentionPolicy]` 错误条目
4. 验证 eNodeB 是否已注册：
   - 检查 eNodeB 状态页面

问题：清理后内存使用量高

症状：

- 清理后应用变慢
- 内存使用量激增

解决方案：

1. 这在大规模删除时是正常的
2. 允许 5-10 分钟内存正常化
3. 考虑减少清理频率：
    - 更改 `@cleanup_interval_ms`（默认 1 小时）
4. 或减少受影响 eNodeB 的保留期限

问题：记录计数不正确

症状：

- 记录计数与 InfluxDB UI 不匹配
- “清理旧数据”显示不同数字

解决方案：

1. 点击“刷新”以强制更新
2. 检查 InfluxDB 查询：
    - 可能需要时间才能反映最近的删除
3. 等待一分钟再试：
    - InfluxDB 可能正在处理删除操作
4. 检查 eNodeB 名称是否完全匹配：
    - 区分大小写的比较

相关文档

- 操作指南 - 完整的操作述

- **Web UI** 指南 - 控制面板参考和功能
- 入门指南 - 快速入门指南
- 常见操作指南 - 日常管理任务
- **Grafana** 集成指南 - 分析和仪表板

访问点

- 数据保留仪表板：`https://localhost:9443/nokia/retention`
- **eNodeB** 状态：`https://localhost:9443/nokia/status`
- **InfluxDB** 状态：`https://localhost:9443/nokia/influx`
- 实时日志：`https://localhost:9443/nokia/logs`

常见问题

问：清理会删除活动数据吗？

答：不。只有超过保留期限的记录会被删除。当前正在收集的数据不会受到影响。

问：我可以为不同的 **eNodeB** 设置不同的保留吗？

答：可以！每个 eNodeB 都可以有自己的保留设置。如果未设置，则使用全局默认值。

问：自动清理多久运行一次？

答：默认每小时运行一次。如果需要，可以在工作者中调整 `@cleanup_interval_ms`。

问：如果我清除所有数据会发生什么？

答：所有记录（性能指标、配置、警报）将被永久删除。这无法恢复。

问：清理会影响数据收集吗？

答：不。清理和数据收集是独立的。新数据将在旧数据被删除时继续写入。

问：清理需要多长时间？

答：取决于记录计数：
小（< 100k）：< 1 秒
中（100k-1M）：1-10 秒
大（> 1M）：10-60+ 秒

问：我可以手动删除特定记录吗？
答：不能通过 UI。仅提供完整清理或完全清除。要进行细粒度删除，请直接使用 InfluxDB CLI 或 API。

问：如果 InfluxDB 不可用怎么办？

答：清理将静默失败并在下小时重试。数据收集继续不受影响。

问：清理会影响性能吗？

答：清理期间会有轻微影响（几秒到几分钟，具体取决于数据大小）。选择每小时间隔以最小化影响。

实施细节

修改的文件

文件	更改
`lib/ran_monitor/database/nokia/airscale.ex`	添加 `retention_hours` 字段
`lib/ran_monitor/config/config.ex`	添加 `data_retention_hours()` 获取器
`config/config.exs`	添加全局保留配置和页面路由
`lib/ran_monitor/application.ex`	将清理工作者添加到监督树

创建的文件

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

关键函数

RetentionPolicy 模块:

- `get_retention_hours(airscale)` - 获取有效保留
- `get_record_counts(airscale_name)` - 获取记录计数

- `get_total_record_count(airscale_name)` - 总计
- `delete_old_records(name, hours)` - 清理旧记录
- `clear_all_records(name)` - 完全清除

RetentionCleanupWorker GenServer:

- `start_link(opts)` - 启动清理工作者
- `init(:ok)` - 初始化工作者
- `handle_info(:cleanup, state)` - 运行清理周期

入门

设置（一次性）

1. 运行迁移:

   ```
   mix ecto.migrate
   ```

2. 重启应用程序:

   ```
   mix phx.server
   ```

3. 验证安装:

   - 导航到 `https://localhost:9443/nokia/retention`
   - 应该看到数据保留仪表板

首次使用

1. 检查当前设置：
    - 查看全局保留（默认：720 小时）
    - 查看每个 eNodeB 的保留设置
2. 如有需要，进行自定义：
    - 在 `config/config.exs` 中更新全局保留
    - 或通过 UI 设置每个 eNodeB
3. 监控清理：
    - 观察日志中的 `[RetentionCleanupWorker]` 条目
    - 验证记录计数随时间减少

支持

需要帮助?

1. 检查日志：查找 `[RetentionPolicy]` 或 `[RetentionCleanupWorker]` 条目
2. 查看本指南：大多数问题在故障排除部分中都有覆盖
3. 检查其他文档：参考上面的相关文档链接
4. 验证设置：确保迁移已运行且工作者已启动

报告问题

包括：

- UI 或日志中的错误消息
- 受影响的 eNodeB 名称
- 当前保留设置
- 删除前后的记录计数
- 重现步骤

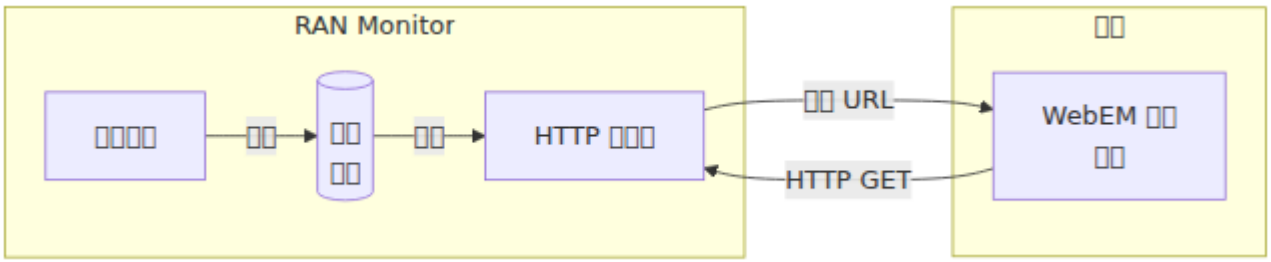
学习资源

相关概念

- **InfluxDB v2.x**：带有保留政策的时间序列数据库
- 保留政策：数据保留的时间
- 清理：自动删除旧数据
- 测量类型：性能指标、配置、警报

外部资源

- InfluxDB 文档
- Elixir GenServer 指南
- Phoenix LiveView



固件管理

诺基亚 **AirScale** 软件包仓库

□□□□□□□□□□□□□□□□

概述

固件管理页面提供了一个集中式的诺基亚 AirScale 基站软件包仓库。上传到 RAN Monitor 的固件文件通过 HTTP 提供，使基站能够使用 WebEM 的“远程服务器”选项直接从 RAN Monitor 下载软件更新。

主要功能

- 上传固件包 - 存储诺基亚 AirScale .zip 软件包
- **HTTP** 文件服务 - 基站通过标准 HTTP URL 下载固件
- **MD5** 校验和 - 自动计算校验和以验证完整性
- **URL** 复制 - 一键复制 WebEM 配置的下载 URL
- 存储统计 - 监控固件仓库磁盘使用情况

工作流程概述

访问页面

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

导航：控制面板侧边栏 > 固件

□□□□□□□□□□□□□□□□□□□□□□

存储□置

默认存储路径

固件文件存储在 RAN Monitor 服务器文件系统中：

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

对于每个固件包，创建以下文件：

- `.zip` - 固件包
- `.md5` - 缓存的 MD5 校验和
- `.metadata.json` - 从包中提取的缓存元数据

自定义存储路径

在 `config/runtime.exs` 中配置备用存储位置：

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

参数	类型	必需	默认	��述
`firmware_storage_path`	字符串	否	`/var/firmware/nokia`	固件文件存储的目录路径。如果目录不存在，将自动创建。

回退行为

如果无法创建配置的存储路径（例如，权限被拒绝），系统将回退到应用程序目录中的 `priv/firmware/nokia`。

Web UI 参考

存储概述

顶部部分显示仓库统计信息：

指标	描述
固件文件	存储的固件包总数
总大小	所有固件文件的总大小
存储路径	当前固件存储的文件系统位置

上传部分

上传诺基亚 AirScale 固件包：

1. 点击上传区域或拖放 `.zip` 文件
2. 验证文件名显示及大小信息
3. 点击 上传固件 以存储文件

□□□□□□□□□□□□□□□□□□□□

支持的文件：仅限 `.zip` 文件（诺基亚软件包）

最大文件大小： 15 GB

上传进度显示：

- 当前上传速度（例如，15.37 MB/s）
- 完成百分比
- 传输字节 / 总大小
- 剩余时间估计（ETA）

文件名格式：诺基亚固件包遵循命名约定：

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

示例： `SBTS25R1_ENB_0000_001046_000000_release_BTSSM_downloadable_A54.zip`

固件列表

该表显示所有上传的固件包：

列	描述
文件名	完整的固件包文件名
版本	提取的发布版本（例如，SBTS25R1）
构建日期	从固件元数据中提取的构建时间戳
信息	查看详细固件元数据的按钮
大小	人类可读格式的文件大小
MD5	MD5 校验和（点击复制）
下载次数	固件被下载的次数
操作	复制 URL、下载、删除按钮

固件元数据

点击任何固件行的 信息 按钮以查看从包中提取的详细元数据：

字段	描述
文件名	完整的包文件名
版本	基站软件发布（例如，SBTS25R1）
构建日期	固件构建时的时间戳
大小	文件大小
文件修改	文件上传到 RAN Monitor 的时间
软件版本	完整的软件版本字符串（例如，FTM_SBTS25R1_2025.05.27_0097）
PS 版本	平台软件发��版本
内核版本	Linux 内核版本
LTE 版本	LTE 软件发布（例如，FL19A）
WCDMA 版本	WCDMA 软件发布（例如，WBTSFP20C）
支持的硬件单元	兼容硬件模块列表（ASCE、ASCF、FCTJ 等）
MD5 校验和	文件完整性校验和
下载 **URL**	基站下载的完整 URL

元数据在上传时自动从固件包中提取并缓存以便快速访问。

操作按钮

按钮	描述
信息	在模态框中查看详细固件元数据
复制 **URL**	将固件下载 URL 复制到剪贴板
下载	直接下载固件文件
删除	从存储中移除固件文件

下载 **URL** 行

在每个固件条目下，显示完整的下载 URL:

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

此 URL 输入到 WebEM 的“远程服务器”URL 字段中。

MD5 校验和

自动计算

在列出固件文件时，MD5 校验和会自动计算。校验和缓存为与固件包同名的 `.md5` 文件，以便快速后续查找。

校验和文件

对于每个固件包，创建相应的 `.md5` 文件:

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

`.md5` 文件包含小写十六进制校验和：

```
a1b2c3d4e5f6789012345678abcdef01
```

验证

使用 MD5 校验和验证下载后的固件完整性：

Linux/macOS:

```
md5sum
SBTS25R1_ENB_0000_001046_000000_release_BTSSM_downloadable_A54.zip
```

Windows（PowerShell）：

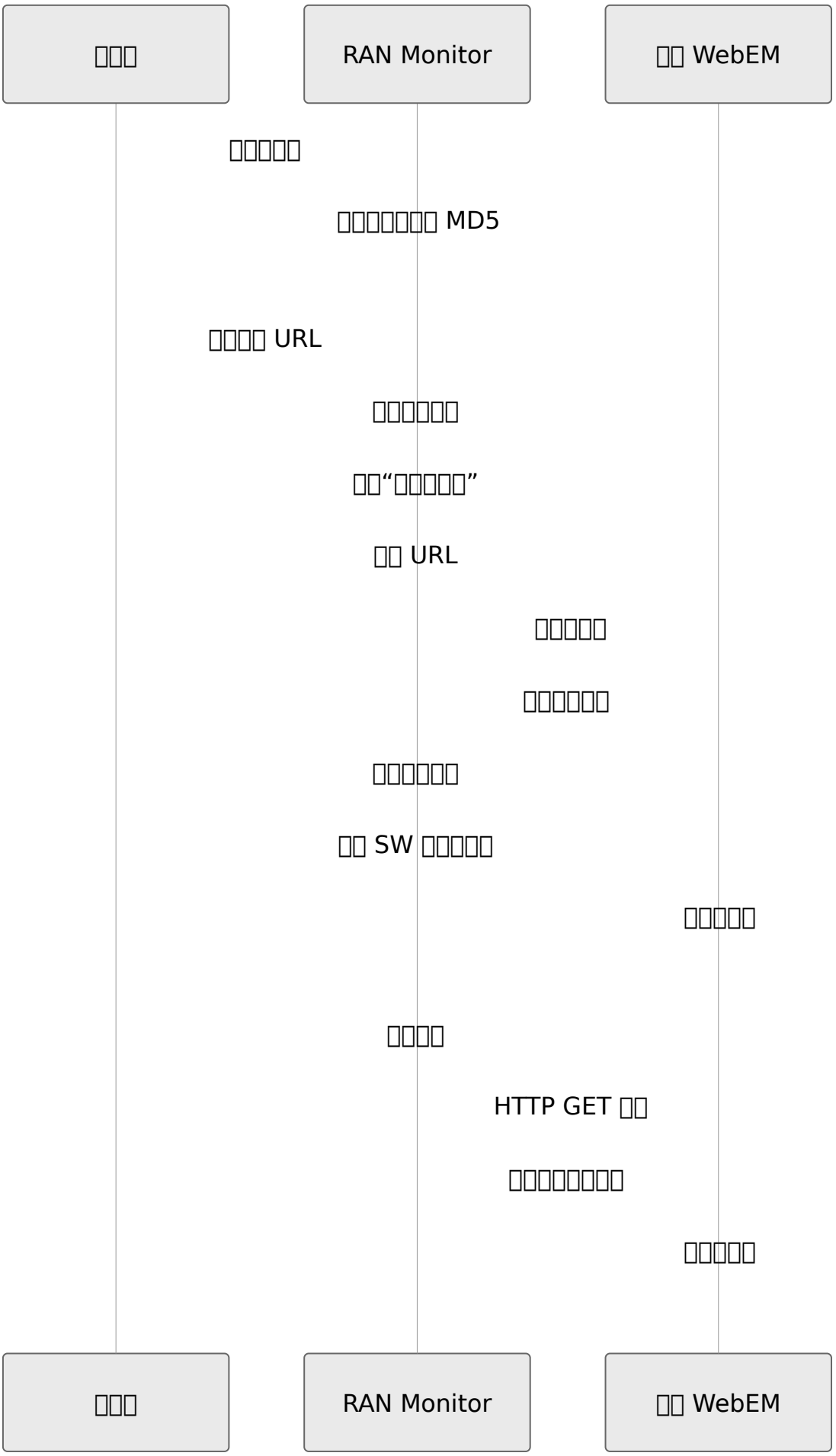
```
Get-FileHash -Algorithm MD5
SBTS25R1_ENB_0000_001046_000000_release_BTSSM_downloadable_A54.zip
```

基站软件更新工作流程

前提条件

- 固件包已上传到 RAN Monitor
- 基站与 RAN Monitor 之间的网络连接
- 访问基站 WebEM 界面

□□□□



操作员
RAN Monitor
基站 WebEM
上传固件包
存储文件，计算 MD5
复制下载 URL
打开软件管理
选择“远程服务器”
粘贴 URL
连接并刷新
返回文件列表
选择固件文件
开始 SW 重启预检查
验证兼容性
开始下载
HTTP GET 固件
流式传输固件文件
安装并激活
操作员
RAN Monitor
基站 WebEM

详细步骤

1. 上传固件到 RAN Monitor

导航到固件管理页面并上传软件包：

- 访问 `https://<ran-monitor-ip>:9443/nokia/firmware`
- 上传 `.zip` 固件包
- 验证文件在固件列表中显示并具有正确的 MD5

2. 复制下载 URL

点击固件包旁的 复制 **URL** 。URL 格式为：

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

3. 访问 WebEM 软件管理

打开基站的 WebEM 界面：

- 导航到 软件管理 > 软件更新
- 查看当前 BTS 软件版本（活动和被动）

4. 配置远程服务器

在 WebEM 的“从远程服务器下载和激活软件”部分：

1. 选择 远程服务器
2. 将复制的 URL 粘贴到 **URL** 字段
3. 点击 连接并刷新

WebEM 连接到 RAN Monitor 并检索文件列表。

5. 选择固件并开始更新

1. 从 软件文件 下拉菜单中选择固件文件
2. 可选地运行 开始 **SW** 重启预检查 以验证兼容性
3. 如果希望立即激活，请勾选 下载后激活软件

4. 点击 开始 以开始软件更新

更新后验证

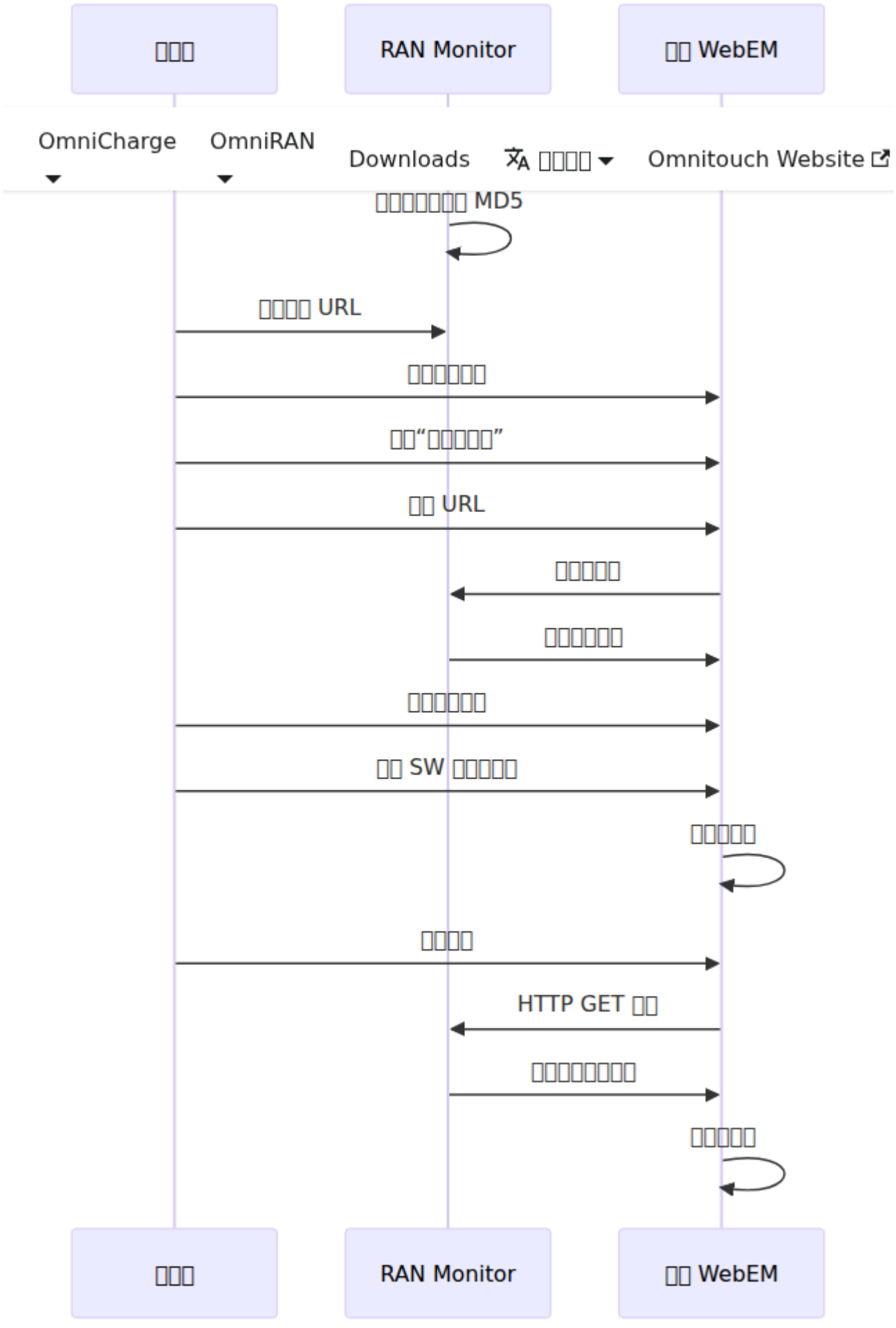
更新完成后:

1. 验证新软件版本在 WebEM 中显示为 活动
2. 检查 RAN Monitor 是否有来自基站的新警报
3. 通过 KPI 确认小区服务恢复

通过□置推送进行远程���件更新

您可以通过 RAN Monitor 的配置管理功能上传配置片段，远程触发软件更新，而无需手动访问每个基站的 WebEM。

□□□□


□□□
RAN Monitor
□□ WebEM
OmniCharge
OmniRAN
Downloads
□□□□
Omnitouch Website
□□□□□□□□ MD5
□□□□ URL
□□□□□□□
□□“□□□□□”
□□ URL
□□□□□
□□□□□□□
□□□□□□□
□□ SW □□□□□
□□□□□
□□□□
HTTP GET □□
□□□□□□□□□
□□□□□
□□□
RAN Monitor
□□ WebEM

□置片段

创建一个包含软件下载参数的 XML 文件。当配置被激活时，基站将从指定的 URL 获取固件。

示例：`sw_update.xml`

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

替换：

- `{site_id}` - 目标基站的 MRBTS ID（例如，`256`，`2057`）
- `{ran-monitor-ip}` - 从基站可访问的 RAN Monitor 的 IP 地址��主机名

推送□置

1. 导航到 RAN Monitor 中基站的详细页面
2. 点击 □置操作 以打开配置操作
3. 上传 `sw_update.xml` 片段
4. 点击 验证 以验证配置
5. 点击 激活 以应用配置

基站将自动下载并安装固件。

批量更新

要更新多个站点，为每个站点创建一个单独的 XML 片段（使用正确的 MRBTS ID），并通过 RAN Monitor 依次推送，或使用 API 自动化该过程。

参数参考

参数	描述
`swPkgUrl`	要下载的固件包的 HTTP URL
`swPkgActivate`	设置为 `true` 以在下载后自动激活

下载跟踪

RAN Monitor 跟踪固件下载，以帮助识别哪些基站已检索软件包。

跟踪信息

对于每次下载，记录以下信息：

字段	描述
IP 地址	下载者的远程 IP 地址
主机名	解析的主机名或匹配的 AirScale 名称
时间戳	下载的日期和时间

查看下载历史

固件列表中的 下载 列显示总下载次数。将鼠标悬停在计数上以查看最近的下载详细信息，包括 IP 地址和时间戳。

下载跟踪数据持久保存到磁盘，并在应用程序重启后仍然有效。

API 访问

列出固件文件

通过 REST API 检索固件库存:

端点: `GET /api/firmware/nokia`

响应:


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


下载固件文件

端点: `GET /firmware/nokia/<filename>`

响应：二进制文件下载，`Content-Type: application/zip`

□置参考

下载 URL 基础

配置固件下载的外部可访问基础 URL：

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

参数	类型	必需	默认	描述
`firmware_base_url`	字符串	否	`http://localhost:4000`	生成下载 URL 时使用的基础 URL。将其设置为 RAN Monitor 服务器的外部可访问主机名/IP。

此设置会影响在 Web UI 中显示的 URL 和 API 返回的 URL。实际文件服务使用应用程序配置的 HTTP 端口。

故障排除

上传失败

症状：文件上传显示错误或从未完成

可能原因：

- 文件超过 15 GB 最大大小
- 文件不是 `.zip` 文件
- 存储路径磁盘空间不足

- 存储目录权限被拒绝

解决方案：

1. 验证文件大小在 15 GB 以下
2. 确认文件扩展名为 `.zip`
3. 检查可用磁盘空间：`df -h /var/firmware/nokia`
4. 验证目录权限：`ls -la /var/firmware`

基站无法下载

症状： WebEM “连接并刷新”失败或下载超时

可能原因：

- 基站与 RAN Monitor 之间的网络连接问题
- 防火墙阻止 HTTP 端口 4000
- URL 不正确（错误的 IP 或主机名）

解决方案：

1. 验证网络路径：从基站网络 ping RAN Monitor
2. 检查防火墙允许端口 4000 入站
3. 确认 URL 使用正确的 IP/主机名，基站可访问
4. 在同一网络段的浏览器中测试 URL

MD5 显示“计算中...”

症状： MD5 列显示“计算中...”且从未更新

可能原因：

- 非常大的文件���在处理
- 文件系统权限问题，阻止 `.md5` 文件创建

解决方案：

1. 等待计算完成（大文件需要更长时间）

2. 点击 刷新 以重新加载页面
3. 检查存储目录权限，允许文件创建

文件已存在错误

症状：上传失败，显示“已存在”错误

解决方案：

1. 如果打算替换，请删除现有文件
2. 如果需要两个版本，请重命名新文件
3. 验证您没有意外重新上传相同的文件

开始使用 RAN Monitor

RAN Monitor 部署和□置快速入门指南

□□□□□□□□□□□□□□□ RAN Monitor

目录

1. □述
2. 先决条件
3. 初始设置过程
4. 验证
5. 后续步骤

概述

本指南将引导您完成 RAN Monitor 的初始部署，从基础设施准备到首次基站连接。

您将完成的任务

通过本指南，您将：

- ✓ 准备所需的基础设施（MySQL, InfluxDB）
- ✓ 使用您的环境详细信息配置 RAN Monitor
- ✓ 启动 RAN Monitor 应用程序
- ✓ 连接您的第一个 Nokia AirScale 基站
- ✓ 验证指标已流入 InfluxDB
- ✓ 访问 Web UI 仪表板

预计时间：第一次设置需要 30-60 分钟

先决条件

在部署 RAN Monitor 之前，请确保您具备以下条件：

基础设施要求

MySQL 数据库服务器

版本：MySQL 5.7+ 或 MariaDB 10.3+
访问：来自 RAN Monitor 服务器的网络连接权限：CREATE、SELECT、INSERT、UPDATE、DELETE 权限
数据库：为 RAN Monitor 创建的空数据库
推荐：专用数据库实例或模式

InfluxDB 时间序列数据库

版本：InfluxDB 1.8+ 或 2.0+
访问：来自 RAN Monitor 服务器的网络连接
Bucket/数据库：已创建并准备好用于存储指标
API 令牌：具有写入权限的令牌（InfluxDB 2.x）
存储：足够的磁盘空间以满足您的保留策略

RAN Monitor 服务器

操作系统：Linux（Ubuntu 20.04+，CentOS 8+ 或类似） RAM：最低 4GB，推荐 8GB
CPU：最低 2 核心，推荐 4+ 核心
磁盘：应用程序和日志最低 20GB
网络：与基站、MySQL 和 InfluxDB 的连接

网络要求

网络连接

RAN Monitor → Nokia AirScale 基站（端口 8080）
Nokia 基站 → RAN Monitor（端口 9076 用于 Webhook）

RAN Monitor → MySQL（端口 3306）
RAN Monitor → InfluxDB（端口 8086）
操作员 → RAN Monitor Web UI（端口 9443）

防火墙规则

允许端口 8080 的入站（基站通信）
允许端口 9076 的入站（Webhook 接收器）
允许端口 9443 的入站（HTTPS Web UI）
允许出站到 MySQL 和 InfluxDB

Nokia 基站要求

对于每个基站：

IP 地址 - 基站可访问的网络地址
端口 - 管理接口端口（通常为 8080）
凭据 - 用于 WebLM 身份验证的用户名和密码
网络路由 - 验证连接性（ping 应成功）
管理接口 - 已启用并可访问

管理者身份验证密钥

私钥 - 用于管理者身份验证（PEM 格式）
公钥证书 - 管理者身份证书（DER 格式）
由 Nokia 提供或使用 OpenSSL 生成

Grafana（可选但推荐）

版本：Grafana 8.0+
访问：与 InfluxDB 的网络连接
目的：分析仪表板和警报

初始设置过程

第 1 步：准备基础设施

1.1 设置 MySQL 数据库

为 RAN Monitor 创建数据库：

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

创建具有适当权限的专用用户：

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

验证来自 RAN Monitor 服务器的连接性：

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

1.2 部署 InfluxDB

对于 InfluxDB 1.x，创建数据库：

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

对于 InfluxDB 2.x，创建一个 bucket：

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

创建一个具有写入权限的 API 令牌（InfluxDB 2.x）：

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

保存令牌以供配置使用。

1.3 验证网络路由

确保与所有基站的网络连接：

```
# 测试与每个基站的连接
ping 10.7.15.66

# 验证管理端口是否可访问
telnet 10.7.15.66 8080
```

验证 MySQL 和 InfluxDB 是否可访问：

```
# 测试 MySQL 连接性
telnet <mysql-host> 3306

# 测试 InfluxDB 连接性
curl http://<influxdb-host>:8086/ping
```

第 2 步：□置 RAN Monitor

所有配置都在 `config/runtime.exs` 文件中管理。

2.1 数据库□置

编辑 `config/runtime.exs` 并配置 MySQL 连接：

```
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 □□□

```
# 基站的主要 SOAP/API 端点
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

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

# 基站通知的 Webhook 端点
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

配置您的网络标识符和基站：

```
config :ran_monitor,
  general: %{
    mcc: "001",  # 您的移动国家代码
    mnc: "001"   # 您的移动网络代码
  },
  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 生成 SSL 证书（如有需要）

对于 HTTPS Web UI，生成 SSL 证书：

```
# 自签名证书用于实验/测试
openssl req -newkey rsa:2048 -nodes -keyout
priv/cert/omnitouch.pem \
  -x509 -days 365 -out priv/cert/omnitouch.crt
```

对于生产环境，请使用 CA 签名的证书。

有关详细的配置选项，请参见 运行时配置指南。

第 3 步：启动系统

配置完成后，启动 RAN Monitor。

3.1 运行数据库迁移

初始化数据库架构：

```
mix ecto.migrate
```

这将创建所有必要的表以管理会话状态。

3.2 启动 RAN Monitor

启动应用程序：

```
mix phx.server
```

或用于生产部署：

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

3.3 监控启动日志

查看日志以确认成功启动：

```
[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
```

查找：

- Web 端点已启动
- 数据库连接已建立
- InfluxDB 连接已确认
- 基站注册尝试

验证

第 4 步：验证操作

检查系统是否正常运行。

4.1 访问 Web UI 仪表板

打开浏览器并导航到：

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

您应该看到 RAN Monitor 控制面板。

4.2 验证基站状态

在 Web UI 中：

1. 导航到 基站 页面
2. 验证您的基站是否出现在列表中
3. 状态应显示为“已关联”（绿色）
4. 注册状态应为“已注册”
5. 会话信息应显示活动会话及到期时间

如果状态为红色/失败，请检查：

- 与基站的网络连接
- 凭据是否正确
- 基站管理接口是否可访问
- 应用日志中的错误消息

4.3 确认指标已流入 InfluxDB

在 Web UI 中:

1. 导航到 **InfluxDB** 状态 页面
2. 连接状态应为绿色
3. 测量计数应在增加
4. 验证“性能指标”、“配置”和“警报”计数

或者，直接查询 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 审查启动日志

检查应用日志以查找任何错误:

在 Web UI 中:

1. 导航到 应用日志 页面
2. 过滤“错误”级别
3. 验证没有关键错误

或者，如果通过 `mix phx.server` 运行，请检查控制台输出。

4.5 检查设备详细信息

在 Web UI 中:

1. 从基站页面单击您的基站
2. 验证:
    - 注册详细信息已填充

- 会话具有有效的到期时间
- 最近的指标显示数据
- 配置状态显示参数

后续步骤

现在 RAN Monitor 正在运行，以下是推荐的后续步骤：

立即行动

1. 添加更多基站
   - 在 `config/runtime.exs` 中添加其他设备
   - 重新启动应用程序以获取更改
   - 请参见 常见操作指南
2. 设置 **Grafana** 仪表板
   - 如果尚未部署，请安装 Grafana
   - 配置 InfluxDB 数据源
   - 导入或创建仪表板
   - 请参见 Grafana 集成指南
3. □置数据保留
   - 设置适当的保留期限
   - 如有需要，按设备配置保留
   - 请参见 数据保留政策指南
4. □置警报和通知
   - 在 Web UI 中查看活动警报
   - 设置 Grafana 警报规则
   - 配置通知渠道
   - 请参见 警报管理指南

操作准备

文档审查：

阅读 Web UI 指南 以了解日常操作
审查 常见操作指南 以了解例行任务
学习 故障排除指南 以解决问题

团队培训：

与操作团队一起浏览 Web UI
练习常见工作流程（每日健康检查、警报调查）
审查关键警报的升级程序

监控设置：

在 Grafana 中创建操作仪表板
为关键指标设置警报规则
配置通知渠道（Slack、电子邮件、PagerDuty）

安全加固：

用 CA 签名的证书替换自签名证书
将凭据移至环境变量
限制 config/runtime.exs 的文件权限
配置防火墙规则

生产部署

生产前：

首先在暂存环境中测试
验证所有基站成功连接
确认指标准确
测试警报通知
记录任何自定义配置

生产启动：

在维护窗口期间部署
在前 24 小时内密切监控
准备回滚计划
保持支持联系可用

持续操作：

通过 Web UI 进行每日健康检查
每周审查警报趋势
每月使用 Grafana 进行容量规划
定期备份配置

获取帮助

故障排除资源

故障排除指南 - 常见问题和解决方案
Web UI 指南 - 控制面板参考
应用日志页面 - 实时系统日志

文档

操作指南 - 完整的操作参考
运行时配置指南 - 配置详细信息
AirScale 配置 - 基站设置

常见首次问题

基站未注册：

验证网络连接（ping）
检查凭据是否正确

- 确认端口 8080 是否可访问
- 查看应用日志中的错误

InfluxDB 连接失败:

- 验证 InfluxDB 是否正在运行
- 检查主机和端口配置
- 确认 API 令牌具有写入权限
- 测试连接性: `curl http://<influxdb-host>:8086/ping`

Web UI 无法访问:

- 验证 HTTPS 端口 9443 是否开放
- 检查 SSL 证书是否存在
- 确认日志中已启动 Web 端点
- 尝试先从本地机器访问

相关文档

- 操作指南 - 完整的操作□述
- **Web UI** 指南 - 控制面板用户指南
- 常见操作指南 - 日常任务
- 运行时□置指南 - 配置参考
- **AirScale** □置 - 基站设置
- **Grafana** 集成指南 - 分析和仪表板
- 警报管理指南 - 警报处理
- 数据保留政策指南 - 数据管理
- 故障排除指南 - 问题解决

Grafana 集成与分析指南

构建用于 **RAN** 监控的操作仪表板和警报

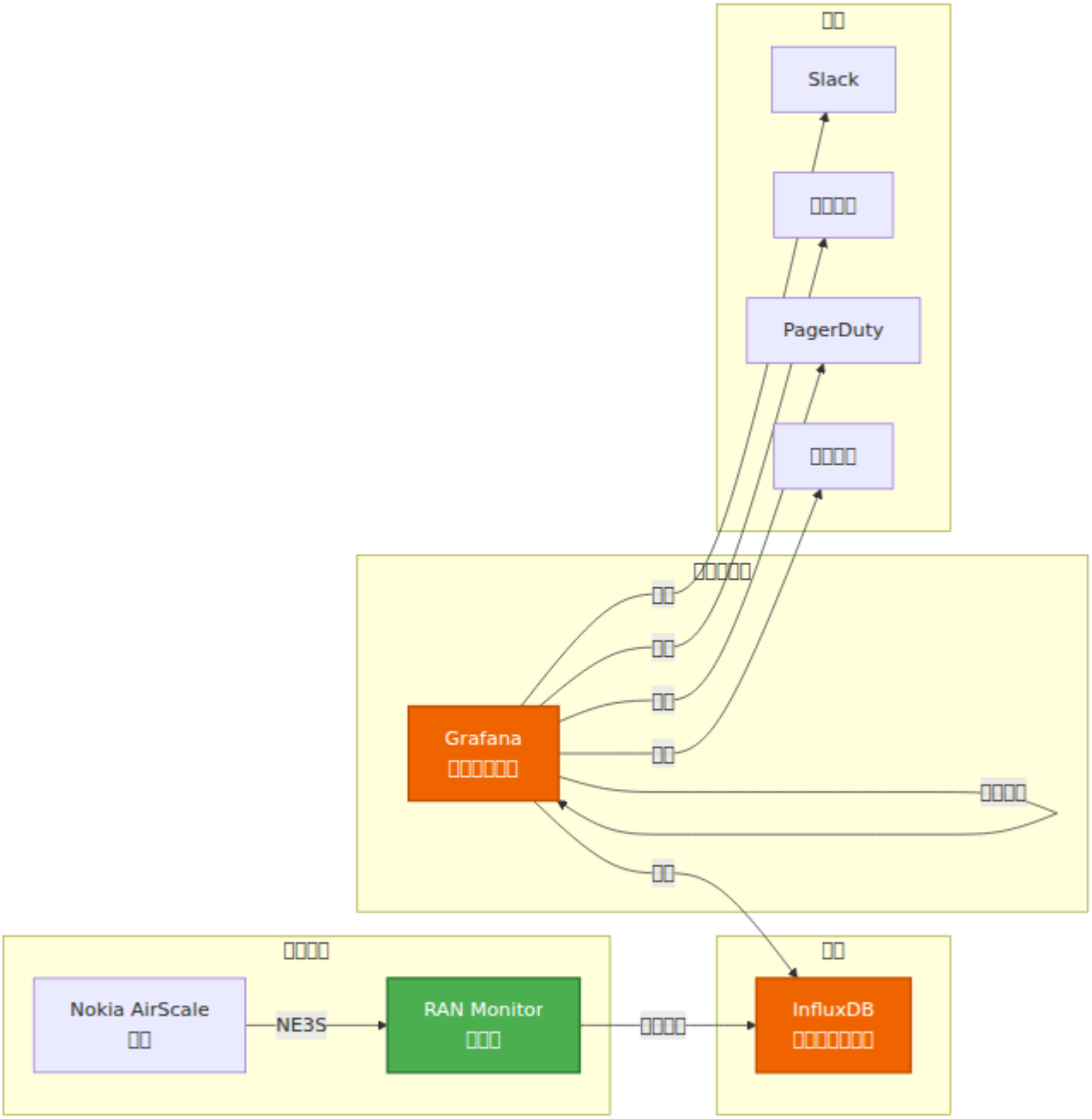
Grafana □□□□□□□□□ KPI □□□□□□□□

目录

1. □述
2. Grafana 和 InfluxDB 设置
3. 数据源配置
4. 仪表板设计模式
5. 查询示例
6. 警报规则与升级
7. 操作仪表板
8. 故障排除

概述

Grafana 是一个可视化和警报平台，将 RAN 监控收集的指标转化为网络运营团队可操作的洞察。



监控架构

Grafana 的好处

- 实时可见性 - 显示当前网络状态的实时仪表板
- 历史分析 - 持续几天/几周/几个月的趋势分析
- 警报 - 在问题影响用户之前主动通知
- 自定义视图 - 针对不同角色（高管、运营、工程）的仪表板
- 报告 - 快照导出和定期报告

仪表板定制

重要：本指南中描述的仪表板和可视化是 示例和模板。运营/NOC (ONS) 团队 将根据其特定的操作要求、KPI 和监控工作流程设计和构建 Grafana 仪表板。

本指南提供：

查询示例和可供构建的模式
仪表板组织的最佳实践
警报配置模板
计数器参考映射（请参见 Nokia 计数器参考）

ONS 团队应定制：

面板布局和可视化
警报阈值和升级策略
其数据量的保留策略（请参见 数据保留政策）
基于监控需求的聚合窗口
通知渠道和路由

有关运行时配置选项和数据收集设置，请参阅 运行时配置指南。

Grafana 和 InfluxDB 设置

安装

先决条件：

InfluxDB 2.0+，为 RAN Monitor 创建了存储桶
具有读取权限的 InfluxDB API 令牌
Grafana 和 InfluxDB 之间的网络连接

Docker Compose 示例：

```yaml
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:
```

创建 **InfluxDB API** 令牌

1. 打开 InfluxDB UI（端口 8086）
2. 导航到 API 令牌
3. 创建具有以下权限的新令牌��
   - 读取：存储桶，`ran_metrics`（你的存储桶）
4. 复制令牌值
5. 在 Grafana 数据源配置中使用

数据源□置

在 **Grafana** 中添加 **InfluxDB** 作为数据源

1. 访问数据源
    - Grafana → 配置 → 数据源
2. 创建新数据源
    - 点击“添加数据源”
    - 选择“InfluxDB”
3. □置连接

设置	值	备注
名称	RAN Monitor	Grafana 中的显示名称
URL	`http://influxdb:8086`	必须能够从 Grafana 访问
访问	服务器（默认）	Grafana 后端访问数据库
组织	omnitouch	你的 InfluxDB 组织
令牌	（API 令牌）	来自 API 令牌创建
默认存储桶	ran_metrics	RAN Monitor 写入的位置
最小时间间隔	10s	匹配轮询间隔

4. 测试连接
    - 点击“测试”按钮
    - 应显示“数据源正常工作”

注意： InfluxDB 连接设置（URL、组织、存储桶名称）必须与 RAN Monitor 配置匹配。有关 InfluxDB 设置的详细信息，请参见 运行时配置指南 和 AirScale 配置 以获取基站注册信息。

Flux 查询语言

Grafana 使用 Flux 查询 InfluxDB。基本语法:

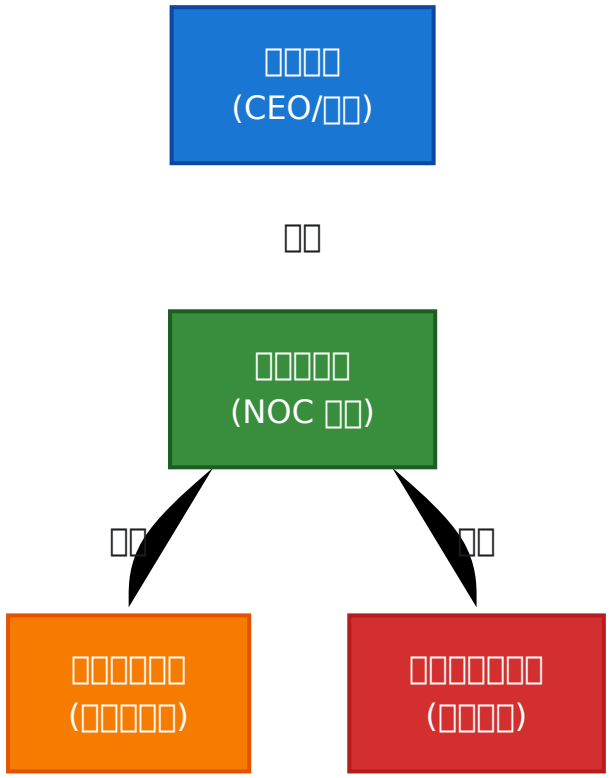
```
from(bucket:"ran_metrics")
  |> range(start: -7d, stop: now())
  |> filter(fn: (r) => r._measurement == "PerformanceMetrics")
  |> filter(fn: (r) => r.device == "SITE_A_BS1")
  |> group(by: ["_field"])
  |> aggregateWindow(every: 1h, fn: mean)
```

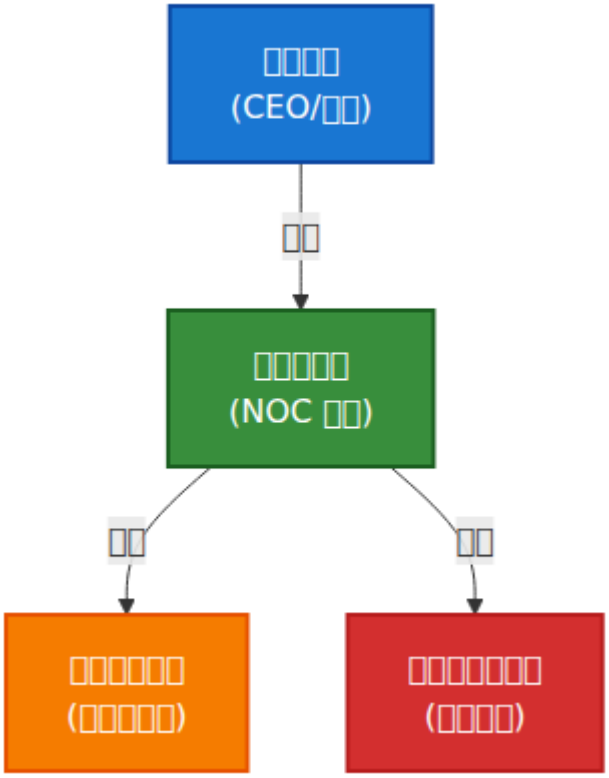
关键概念:

- `from()` - 选择存储桶
- `range()` - 时间窗口
- `filter()` - 选择数据
- `group()` - 组织结果
- `aggregateWindow()` - 汇总时间段

□□□□□□□□

□□□□□□□□


□□□□
(CEO/□□)
□□
□□□□□□
(NOC □□)
□□
□□
□□□□□□□
(□□□□□□)
□□□□□□□□□
(□□□□□)



面板类型与用例

标准仪表板部分

顶部部分：关键指标（状态指示器）

一目了然地显示当前状态：

```
┌──────────────────────────────────────────────────┐
│ 网络健康快照                                     │
├──────────────┬──────────────┬──────────────┬─────┤
│ 设备正常     │ 活动警报     │ 平均小区可用性     │
│ 48/50 (96%)  │ 3 个关键警报 │ 98.5%              │
├──────────────┴──────────────┴──────────────┴─────┤
│ 最近事件:  [2小时前]  已修复                     │
└──────────────────────────────────────────────────┘
```

目的：

- 快速状态验证（< 10 秒评估）
- 绿色/红色指示器显示即时问题

- 链接到详细仪表板以进行调查

中间部分：趋势（时间序列图）

显示随时间变化的模式和变化：

```
┌──────────────────────────────────────────────────────────┐
│ 流量模式（7 天）                                          │
│ [大型面积图显示每日/每周模式]                             │
│ 峰值：250 Gbps（星期三下午 2 点）                         │
│ 谷值：80 Gbps（星期天凌晨 3 点）                          │
└──────────────────────────────────────────────────────────┘
```

目的：

- 识别容量限制
- 理解流量模式
- 预测高峰时段
- 检测异常

部部分：详细信息与警报（表格）

显示详细信息：

```
┌──────────────────────────────────────────────────────────┐
│ 活动警报（按严重性排序）                                  │
├──────┬──────────────┬──────────────┬──────────────┬──────┤
│ 级别 │ 设备         │ 问题         │ 持续时间     │      │
├──────┼──────────────┼──────────────┼──────────────┼──────┤
│ □    │ SITE_A_BS1   │ 小区故障     │ 45 分钟      │      │
│ □    │ SITE_B_BS2   │ 高温         │ 2 小时       │      │
└──────┴──────────────┴──────────────┴──────────────┴──────┘
```

目的：

- 立即行动项目
- 调查详细信息
- 趋势信息（持续时间、频率）

查询示例

注意：以下查询示例使用 Nokia 特定的性能计数器。有关详细的计数器定义、单位和使用指南，请参见 Nokia 计数器参考。有关配置数据收集间隔和 InfluxDB 设置的信息，请参阅 运行时配置指南。

性能指标查询

按设备的小区可用性（过去 **24** 小时）

```
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")
```

用法：

- 高管仪表板用于 SLA 报告
- 显示每小时平均值的时间序列图
- 目标：> 99.5% 可用性

流量吞吐量趋势（**7** 天）

```
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")
```

用法：

- 容量规划仪表板

- 显示峰值与谷值的面积图
- 确定繁忙时段以便调度

按小区的下行资源利用率

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

用法：

- 实时操作仪表板
- 当达到 80% 时发出警告，95% 时为关键警告
- 快速识别拥堵的小区

警报查询

按严重性排序的活动警报（过去 **24** 小时）

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

用法：

- 状态指示器显示警报计数
- 饼图显示分布
- 点击查看详细警报列表

警报率（每小时警报数）

```
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")
```

用法:

- 趋势图显示警报风暴发生的时间
- 确定高不稳定性时段
- 与配置更改相关联

频繁触发的警报

```
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")
```

用法:

- 识别系���性问题
- 优先考虑工程工作
- 根本原因分析重点

高级分析

小区可用性预测（线性回归）

```
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")
```

用法:

- 预测 SLA 可能被违反的时间
- 主动维护调度
- 容量预测

切换成功率与流量的相关性

```
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")
```

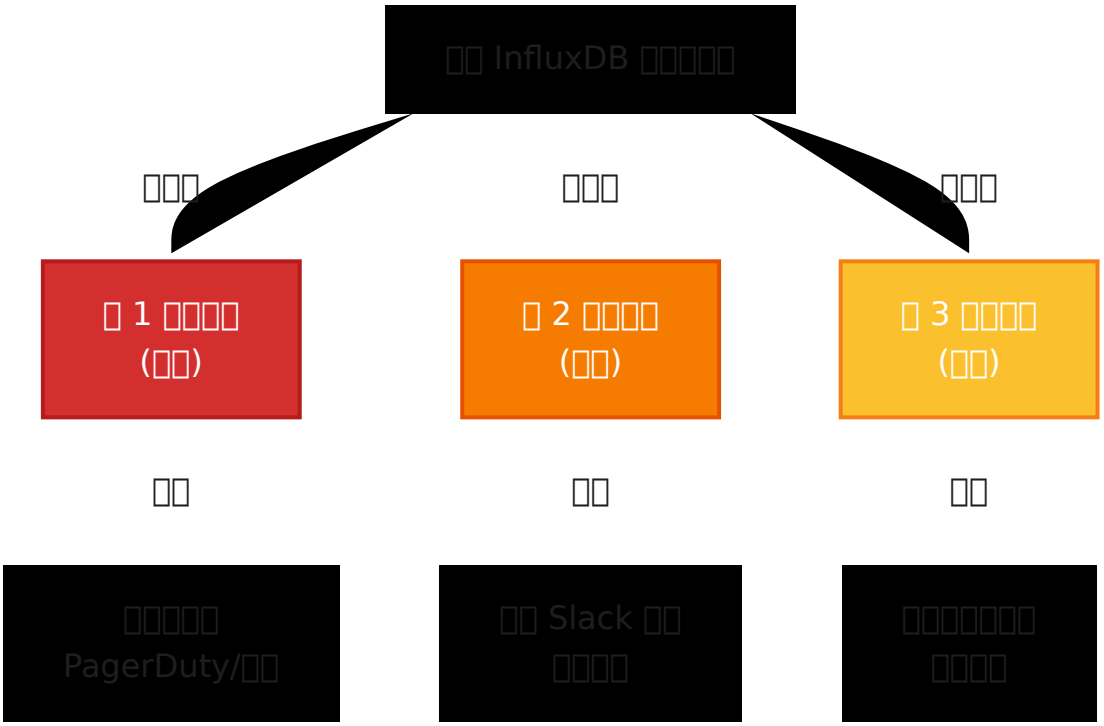
用法:

- 确定切换问题是否与负载相关
- 调整切换滞后阈值

- □□□□□□□

□□□□□□□□

□□□□□□□



□ Grafana □□□□□□□□

□ **1** □□□□□□□□□

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

□ **2** □□□□□□□□□

□□ **1**□□□□□□□□

条件: `CellAvailability < 95%`
持续时间: 15 分钟
评估频率: 每 1 分钟
持续: 过去 15 分钟

理由:

- 在 95% 时触发，以在 SLA 违反（99.5%）之前发出警告
- 15 分钟窗口避免因瞬态而产生的误报
- 每分钟监控以快速响应

示例 **2**: 警报风暴检测

条件: `count(active_alarms) > 10`
持续时间: 5 分钟
评估频率: 每 2 分钟
持续: 过去 5 分钟

理由:

- 10 个以上警报指示系统性问题
- 快速 5 分钟检测以便快速响应
- 频繁检查以捕捉升级

示例 **3**: 下行资源耗尽

条件: `DLResourceUtilization > 90%`
持续时间: 30 分钟
评估频率: 每 5 分钟
持续: 过去 30 分钟

理由:

- 持续高资源使用指示拥堵
- 30 分钟窗口防止因流量激增而产生的误报
- 每 5 分钟监控以捕捉持续拥堵

第 3 步：配置通知渠道

1. 点击“通知渠道”
2. 选择或创建渠道（Slack、电子邮件、PagerDuty 等）
3. 配置消息模板

消息模板示例：

```
警报: {{ .AlertRuleName }}
严重性: {{ .Severity }}
设备: {{ .Labels.device }}
值: {{ .EvalMatches[0].Value }}
持续时间: {{ .StartsAt }}

{{ .RuleUrl }}
```

升级策略

第 1 级警报（关键）：

- 条件：影响服务（设备故障，SLA 违反即将发生）
- 持续时间：立即（1-5 分钟）
- 通知：电话 + 短信 + Slack + PagerDuty
- 负责人：值班工程师
- **SLA**：< 15 分钟内响应

第 2 级警报（主要）：

- 条件：性能下降（质量、可用性趋势下降）
- 持续时间：15-30 分钟
- 通知：Slack + 电子邮件 + PagerDuty
- 负责人：NOC 团队 + 高级工程师
- **SLA**：< 30 分钟内响应

第 3 级警报（次要）：

- 条件：信息性（趋势、接近限制）
- 持续时间：1+ 小时

- 通知：Slack + 仪表板
- 负责人：容量规划 / 工程
- **SLA**：每日审查

通知渠道

Slack 集成

1. 在工作区中创建 Slack 应用
2. 获取网络钩子 URL
3. 在 Grafana 警报 → 通知渠道
4. 添加“Slack”渠道
5. 粘贴网络钩子 URL
6. 测试通知

Slack 消息格式：

 关键：小区故障 - SITE_A_BS1_Cell1
持续时间：45 分钟
影响：~2000 名用户
最后成功数据：下午 2:15

[调查] [确认] [仪表板]

PagerDuty 集成

1. 在 PagerDuty 中创建集成密钥
2. 在 Grafana 警报 → 通知渠道
3. 添加“PagerDuty”渠道
4. 粘贴集成密钥
5. 映射严重性级别：
    - 关键 → 触发事件
    - 主要 → 以较低紧急性触发
    - 次要 → 添加到现有事件

电子邮件集成

1. 在 Grafana 配置中配置 SMTP
2. 在警报 → 通知渠道
3. 添加“电子邮件”渠道
4. 输入收件人地址
5. 可以包含收件人 CSV 以便于分发列表

操作仪表板

仪表板 1：执行健康仪表板

受众：管理层、高管

刷新频率： 5 分钟

目的：高级健康□述

面板：

1. 状态摘要（4 个状态面板）
    - 设备正常 / 总数
    - 活动警报（按严重性着色）
    - 平均小区可用性（%）
    - 当前峰值流量（Gbps）
2. 网络健康（时间序列）
    - 小区可用性趋势（7 天）
    - 警报率趋势（7 天）
    - 流量预测与实际
3. 最近事件（表格）
    - 时间、持续时间、根本原因、状态
    - 最近 7 天，按严重性排序
4. 设备状态网格（热图）

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

□□□□□□□

□□PRB □□□□□□□□□□□□

□□□ 2□NOC □□□□□

□□□□ □□□□□□□□□□
□□□□□□ 10 □
□□□□□ □□□□□□□

□□□□

1. □□□□□□□□□□

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

2. □□□□□□□□□□□

   - □□□□□ %□□□□□□
   - □□□□□ %□□□□□□

设备的 CPU %

3. 流量概述（面积图）

下行/上行吞吐量（过去 24 小时）
当前与 24 小时平均值
峰值小时指示

4. 警报趋势（条形图）

按严重性计数（过去一小时，滚动）
堆叠条形图显示分布

5. 设备状态（快速查看）

设备名称、IP、状态（绿色/红色）
最近指标更新时间戳
链接到设备特定仪表板

6. 最近事件（时间序列）

警报出现/清除
配置更改
会话状态变化

示例仪表板：

示例显示 4G 状态概览，包括地理地图、按严重性级别的警报表和性能统计。

仪表板 3：工程深度分析

受众：网络工程师
刷新频率： 1 分钟
目的：详细技术分析

面板：

1. 流量模式分析（多系列）

下行/上行按站点比较
基线 + 当前叠加
每日时段季节性

2. 小区质量指标（多系列）

SINR 分布（直方图）
RSRP 分布（直方图）
切换成功率趋势

3. 无线性能（时间序列）

- RLC 重传率（按站点）
- RRC 设置成功率
- 呼叫掉线率

4. □置审计（表格）
    - 设备、配置日期、参数更改
    - 突出显示最近的修改
5. 相关性分析（散点图）
    - 下行资源与流量
    - 流量与切换成功
    - 可用性与警报计数

仪表板 4：值班警报仪表板

受众：事件响应人员（值班）

刷新频率： 5 秒

目的：快速事件评估和响应

面板：

1. 警报摘要（大状态）
    - 活动关键警报的计数
    - 背景颜色：绿色（正常）/红色（问题）
2. 关键问题（表格，大文本）
    - 设备、问题、持续时间
    - 自动滚动到最新的优先显示
3. 相关指标（时间序列）
    - 流量、资源利用率
    - 受影响设备的质量指标
    - 根据警报自动填充

4. 最近更改（表格）

最近 4 小时内的配置更改
软件版本
参数修改

5. 类似问题（表格）

最近 30 天内的相同问题类型
解决时间
确定的根本原因

6. 升级路径（文本面板）

下一级升级联系人
维护窗口信息
相关票据/事件编号

仪表板 5：Nokia AirScale 详细性能

受众： RF 工程师、性能分析师
刷新频率： 30 秒
目的：深入了解 Nokia 特定指标和 KPI

该仪表板使用 Nokia 特定的性能计数器提供 AirScale 基站性能的全面可见性。有关详细计数器定义，请参见 Nokia 计数器参考。

面板 1：资源利用率概述（仪表）

显示当前 PRB（物理资源块）使用情况：

```
// 下行 PRB 使用
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}))  // 转换为百分比
  |> rename(columns: {_value: "下行 PRB 使用"})

// 上行 PRB 使用
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}))  // 转换为百分比
  |> rename(columns: {_value: "上行 PRB 使用"})
```

可视化：仪表面板带有阈值：

- 绿色：0-70%
- 黄色：70-85%
- 红色：85-100%

面板 **2**：吞吐量趋势（时间序列）

显示 PDCP 层的下行和上行吞吐量：

```
// 下行吞吐量
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}))  // 转换为
Mbps
  |> rename(columns: {_value: "下行 Mbps"})

// 上行吞吐量
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}))  // 转换为
Mbps
  |> rename(columns: {_value: "上行 Mbps"})
```

使用的计数器:

- **M8012C26** - PDCP 吞吐量下行均值（千比特/秒）
- **M8012C23** - PDCP 吞吐量上行均值（千比特/秒）

面板 **3**: 活动 **UE** 数量（时间序列）

跟踪连接用户的数量:

```
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: "已连接 UE"})
```

使用的计数器:

- **M8018C1** - 每个 eNB 的活动 UE 数量（计数）

面板 **4**：小区可用性（时间序列带警报阈值）

计算并显示小区可用性百分比：

```
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,
      "小区可用性": 100.0 * r.M8020C3 / (r.M8020C6 - r.M8020C4)
    }))
```

使用的计数器：

- **M8020C3** - 小区可用时的样本
- **M8020C6** - 小区可用性分母
- **M8020C4** - 小区计划不可用时的样本

阈值警报：小区可用性 < 99%

面板 **5**：每小区 **PRB** 利用率（多系列时间序列）

显示按单个小区划分的资源使用情况：

```
import "strings"

// 上行 PRB 每小区
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": "平均上行 PRB 使用"})

// 下��� PRB 每小区
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": "平均下行 PRB 使用"})
```

面板 **6**：每小区吞吐量（多系列时间序列）

PDCP 吞吐量按小区划分：

```
import "strings"

// 下行 PDCP 吞吐量每小区
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": "下行 PDCP 吞吐量"})

// 上行 PDCP 吞吐量每小区
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": "上行 PDCP 吞吐量"})
```

面板 **7**：**RSSI**（接收信号强度指示）（多系列时间序列）

显示上行信号强度统计：

```
import "strings"

// □□ 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": "□□ RSSI"})

// □□ 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": "□□ RSSI"})

// □□ 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": "□□ RSSI"})
```

□□□□□□□

- **M8005C0** - PUCCH □□ RSSI□dBm□
- **M8005C1** - PUCCH □□ RSSI□dBm□

- **M8005C2** - PUCCH 平均 RSSI (dBm)

面板 **8**: 延迟 (时间序列)

PDCP SDU 延迟测量:

```
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": "延迟"})
```

使用的计数器:

- **M8001C2** - PDCP SDU 延迟下行 DTCH 均值 (毫秒)

面板 **9**: **RRC** 连接设置成功率 (时间序列带阈值)

计算成功连接设置的百分比:

```
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,
      "设置成功率": 100.0 * r.M8013C5 / (r.M8013C17 + r.M8013C18 +
r.M8013C19 + r.M8013C34 + r.M8013C31 + r.M8013C21 + r.M8013C93 +
r.M8013C91)
    }))
```

使用的计数器：

- **M8013C5** - 信令连接建立完成
- **M8013C17-M8013C93** - 各种连接尝试类型

阈值警报：设置成功率 < 95%

面板 **10**：天线的 **VSWR**（电压驻波比）（时间序列）

对天线系统的硬件健康监控：

```
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")
```

使用的计数器:

- **M40001C0** - 每个天线分支的 VSWR（0.1 比率）

阈值警报: VSWR > 2.0

面板 **11**: 功耗（时间序列）

基站功耗监控:

```
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": "功耗"})
```

使用的计数器:

M40002C2 - 功耗（因子为 100000）

仪表板变量：

该仪表板使用 Grafana 模板变量进行动态过滤：

${Airscale} - 基站选择器（下拉菜单）
${CellKey} - 多小区站点的单元选择器（下拉菜单） ${RadioKey} - VSWR 的无线单元选择器（下拉菜单）

此仪表板的警报规则：

1. 高 PRB 利用率 - 当 DL 或 UL PRB > 85% 持续 5 分钟时触发
2. 低小区可用性 - 当可用性 < 99% 持续 10 分钟时触发
3. 低设置成功率 - 当 RRC 设置成功率 < 95% 持续 5 分钟时触发
4. 高 VSWR - 当任何天线的 VSWR > 2.0 持续 15 分钟时触发
5. 异常功耗 - 当功耗偏离基线 > 20% 时触发

使用此仪表板：

1. 容量分析 - 监控 PRB 利用率以识别接近容量的小区
2. 性能故障排除 - 使用 RSSI、延迟和设置成功率诊断问题
3. 硬件健康 - 跟踪 VSWR 和功耗以进行主动维护
4. 质量保证 - 监控可用性和吞吐量以确保 SLA 合规

请参见 Nokia 计数器参考 以获取完整的计数器定义和使用指南。

示例仪表板 - 详细视图：

Nokia Monitor □□□□□□□ S1 □□□□□□□□□□□□□□□□□□□□ PRB □□□□□□□□□□□□□□□□□

□□□□□ - □□□□□□□PRB □□□□□□□

□□□□□□□□□ LNCEL□□LTE PRB □□□□ TTI□□□/□□□□□ PDCP □□□□□□□□□□□□□□□

□□□□□ - RSSI□□□□□□□ RRC□

RSSI □□□□□/□□/□□□□□□□□□□□□□□□RRC □□□□□□□□□□□ VSWR□RMOD□□□□

□□□□□ - □□□□□□□□

□□□□□□□□ RRC □□□□□□□□□□□□□□□VSWR RMOD □□□□□□□□□□□□□□□□

□□□□□ - □□□□□□ **VSWR** □□□□□

□□□ VSWR RMOD-2 □□□□□□□□□□ANTL-1□ANTL-2□ANTL-3□ANTL-4□□□□□□□□□□□□□□□□□□□□□□

故障排除

面板中没有数据出现

症状：

- 仪表板加载但面板显示“无数据”
- 数据源显示已连接

诊断：

1. □查 InfluxDB 查询是否有效
2. 验证测量名称在 InfluxDB 中是否存在
3. □查时间范围是否包含数据点
4. 验证过滤条件是否匹配数据标签

解决方案：

- 在 InfluxDB UI 中直接测试查询
- 调整时间范围（尝试过去 24 小时）

- 检查标签名称是否与 RAN Monitor 输出匹配
- 启用查询检查器以查看实际结果

仪表板加载缓慢

症状:

- 仪表板加载时间 > 5 秒
- 面板逐个缓慢出现

诊断:

```
1. 面板数量过多 (> 8)
2. 查询过于复杂/数据范围过大
3. InfluxDB 性能问题
4. 网络延迟
```

解决方案:

- 减少面板数量
- 限制时间范围 (24 小时 vs. 1 年)
- 在 InfluxDB 中预聚合数据
- 检查 InfluxDB CPU/内存
- 增加查询超时

警报未触发

症状:

- 警报规则已启用
- 条件应满足
- 未收到通知

诊断:

1. □查警报评估是否正在进行
2. 验证通知渠道是否正常工作
3. □查警报规则条件
4. 查看通知渠道日志

解决方案：

- 手动测试警报规则（铅笔图标 → 测试）
- 检查警报规则状态在警报 → 警报规则中
- 验证通知渠道是否具有正确的 URL/密钥
- 检查 Grafana 日志以获取错误信息
- 使用手动触发重新测试通知渠道

仪表板中的数据不正确

症状：

- 值与预期数字不匹配
- 数据在时间上看起来偏移
- 聚合似乎错误

诊断：

1. □查时区设置
2. 验证聚合函数
3. □查标签/标签过滤器
4. 审查查询以查找数学错误

解决方案：

- 将仪表板时区设置为与 InfluxDB 匹配
- 验证 aggregateWindow 函数（均值/总和/最后）
- 在 InfluxDB 中直接测试过滤器
- 简化查询以隔离问题

数据保留问题

症状：

- 历史数据缺失或不完整
- 查询返回的数据少于预期
- 仪表板显示时间序列中的间隙

解决方案：

- 检查 数据保留政策 中的保留策略设置
- 验证保留期限是否足够满足你的查询时间范围
- 如有必要，调整每个 eNodeB 的保留设置

□置问题

症状：

- InfluxDB 连接错误
- 查询中缺少 eNodeB 数据
- 数据收集间隔不正确

解决方案：

- 查看 运行时配置指南 以获取正确的设置
- 验证 AirScale 配置 是否正确
- 检查 eNodeB 注册状态

相关文档

- **Nokia** 计数器参考 - 完整的性能计数器定义
- 数据保留政策 - 管理数据生命周期和存储
- 运行时□置指南 - 系统配置和调优
- **AirScale** □置 - 基站设置和注册

常见操作指南

日常 **RAN** 监控管理任务

[illegible]

目录

1. [illegible]述
2. 添加新基站
3. 移除基站
4. 更新设备凭据
5. 调整采集间隔
6. 管理设备配置
7. 监控系统健康
8. 数据管理
9. 备份与恢复
10. 系统维护

概述

本指南涵盖了日常操作任务，以管理 RAN 监控的日常操作。这些程序旨在为 NOC 团队、网络管理员和运营人员提供支持。

前提条件

- RAN 监控已安装并运行
- 您可以访问配置文件
- 您可以重启 RAN 监控应用程序
- 您了解您的网络拓扑

有关初始设置，请参见 入门指南。

添加新基站

在部署新的 Nokia AirScale 基站时，请按照以下步骤将其添加到监控中。

步骤 1：验证网络连接

在将设��添加到配置之前，请确保网络连接正常：

```
# 测试基本连接
ping <base-station-ip>

# 验证管理端口是否可访问
telnet <base-station-ip> 8080
```

预期结果：成功的 ping 响应和 telnet 连接

如果失败：

- 检查网络路由
- 验证防火墙规则允许连接
- 确认基站已通电并正常运行

步骤 2：收集设备信息

收集以下信息：

信息	示例	查找位置
IP 地址	10.7.15.67	网络文档或设备标签
端口	8080	通常为 Nokia AirScale 的 8080
设备名称	Site-B-Tower-1	使用站点命名约定
用户名	admin	来自基站配置
密码	password123	来自基站配置

设备命名最佳实践：

- 使用描述性、一致的命名
- 包含站点标识符
- 如果同一站点有多个设备，包含设备类型
- 示例：`NYC-SiteA-BS1`，`LAX-Tower-Main`，`CHI-Indoor-DAS`

步骤 3：检查未□置的设备

在手动添加之前，检查设备是否已尝试连接：

1. 打开 Web UI：`https://<ran-monitor-ip>:9443`
2. 导航到 未□置的 **eNodeBs** 页面
3. 查找您的设备的 IP 地址或代理 ID
4. 如果找到，请记下代理 ID

这有助于验证设备是否可以到达 RAN 监控。

步骤 4：将设备添加到□置

编辑 `config/runtime.exs` 并将新设备添加到 `airscales` 列表中：

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

      # 新设备
      %{
        address: "10.7.15.67",           # 新基站的 IP 地址
        name: "Site-B-Tower-1",           # 描述性名称
        port: "8080",                     # 管理端口
        web_username: "admin",            # WebLM 用户名
        web_password: "password123"       # WebLM 密码
      }
    ]
  }
```

重要：确保 Elixir 语法正确 - 逗号、缩进和映射结构必须正确。

步骤 5：验证□置

在重启之前，验证配置语法：

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

预期输出：无错误

如果有错误：

- 检查是否缺少逗号

- 验证所有打开的花括号 `{` 和方括号 `[` 都已关闭
- 确保字符串正确用
- 检查缩进是否一致

步骤 6：重启 RAN 监控

重启应用程序以加载新配置：

```
# 如果在开发中运行
# 按 Ctrl+C 两次，然后：
mix phx.server

# 如果作为服务运行
systemctl restart ran_monitor

# 如果通过发布运行
/path/to/ran_monitor/bin/ran_monitor restart
```

步骤 7：验证设备注册

重启后，验证设备是否已被监控：

1. 检查应用程序日志：

   ```
   [info] 正在尝试与设备注册：Site-B-Tower-1
   [info] 成功注册 Site-B-Tower-1
   ```

2. 检查 **Web UI**：

   - 导航到 基站 页面
   - 在列表中找到您的新设备
   - 状态应为 "已关联"（绿色）
   - 注册状态应为 "已注册"

3. 检查设备详细信息：

   - 点击设备

- 验证会话信息显示活动会话
- 确认 "最后联系" 时间戳是最近的

4. 检查 **InfluxDB** 状态:

   - 导航到 **InfluxDB** 状态 页面
   - 验证总记录数在增加
   - 随着数据的收集，测量计数应增加

步骤 **8**: 置数据保留（可选）

如果该设备需要与全局默认值不同的保留策略:

1. 导航到 数据保留 页面
2. 在列表中找到您的新设备
3. 更新 "保留期限" 字段（以小时为单位）
4. 系统会自动保存

有关多详细信息，请参见 数据保留政策指南。

步骤 **9**: 添加到 **Grafana** 仪表板

更新 Grafana 仪表板以包含新设备:

1. 编辑仪表板模板变量
2. 将设备名称添加到下拉选择器
3. 如果需要，创建设备特定的仪表板

有关更多详细信息，请参见 Grafana 集成指南。

移除基站

在退役基站时，请按照以下步骤将其从监控中移除。

步骤 1：决定数据处理方式

在移除之前，决定如何处理历史数据：

选项 **A**：保留数据

- 禁用监控但保留历史数据
- 对于已退役但可能返回的设备有用

选项 **B**：移除数据

- 删除设备的所有历史数据
- 释放 InfluxDB 存储
- 不可逆 - 数据无法恢复

步骤 2：禁用设备（保留数据）

要停止监控但保留历史数据：

编辑 `config/runtime.exs` 并在 `airscales` 列表中找到设备。将其注释掉或移除：

```
airscales: [
  %{
    address: "10.7.15.66",
    name: "Site-A-BS1",
    port: "8080",
    web_username: "admin",
    web_password: "password1"
  },

  # 退役设备 - 注释掉以保留数据
  # %{
  #   address: "10.7.15.67",
  #   name: "Site-B-Tower-1",
  #   port: "8080",
  #   web_username: "admin",
  #   web_password: "password123"
  # }
]
```

步骤 3：移除数据（可选）

要删除设备的所有历史数据：

1. 在 Web UI 中导航到 数据保留 页面
2. 在列表中找到设备
3. 点击 清除所有数据 按钮
4. 确认操作

警告：这是永久性的，无法撤销。

步骤 4：重启 RAN 监控

重启应用程序以应用配置更改：

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

步骤 5：验证移除

重启后：

1. 检查基站页面：
    - 设备应不再出现在活动列表中
    - 如果数据被保留，设备可能仍会在历史查询中显示
2. 检查应用程序日志：
    - 没有对已移除设备的注册尝试
    - 没有关于缺少设备的错误
3. 检查 **InfluxDB**：
    - 如果数据被删除，记录计数应降低
    - 设备应不出现在新指标中

步骤 6：更新 Grafana 仪表板

从 Grafana 配置中移除设备：

1. 编辑仪表板模板变量
2. 从下拉列表中移除设备名称
3. 如果存在，删除设备特定的仪表板

更新设备凭据

当基站密码更改时，更新 RAN 监控配置。

步骤 1：记录当前状态

在进行更改之前：

1. 验证设备当前已连接（绿色状态）
2. 记录当前会话信息
3. 截图或记录当前状态以供比较

步骤 2：更新□置

编辑 `config/runtime.exs` 并更新凭据：

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

步骤 3：重启 RAN 监控

应用配置更改：

```
systemctl restart ran_monitor
```

步骤 4：验证重新连接

重启后：

1. 检查应用程序日志：

   ```
   [info] 正在尝试与设备注册：Site-A-BS1
   [info] 成功注册 Site-A-BS1
   ```

2. 检查 **Web UI**：

   - 设备状态应为"已关联"（绿色）
   - 应建立新会话
   - "最后联系"应为最近

如果设备未能连接：

- 验证新凭据是否正确
- 检查密码中的拼写错误
- 确认凭据在基站上直接有效
- 查看应用程序日志以获取身份验证错误

调整采集间隔

更改 RAN 监控从基站收集数据的频率。

理解采集间隔

RAN 监控以不同的间隔收集三种类型的数据：

数据类型	默认间隔	可□置	短间隔的影响
性能指标	10 秒	是	更细粒度的数据，更高的网络/存储使用
报警	10 秒	是	更快的报警检测，更多的查询
配置	60 秒	是	更频繁的配置快照，更多的存储
健康检查	30 秒	是	对连接问题更敏感

何时调整间隔

缩短间隔（更频繁）：

- 解决正在进行的问题
- 高价值的关键基础设施
- 有严格要求的 SLA 监控
- 容量测试和分析

延长间隔（不那么频繁）：

- 减少网络流量
- 减少 InfluxDB 存储使用
- 低优先级的测试环境
- 带宽受限的链路

步骤 1：修改□置

采集间隔在应用程序代码中配置，而不是在 runtime.exs 中。要更改它们，您需要修改源代码并重新编译。

示例位置（可能因版本而异）：

- 性能指标：`lib/ran_monitor/nokia/airscale/manager.ex`

- 报警: `lib/ran_monitor/nokia/airscale/manager.ex`
- 配置: `lib/ran_monitor/nokia/airscale/manager.ex`

注意：联系 Omnitouch 支持以获取有关采集间隔更改的帮助，因为这需要源代码修改。

步骤 2：考虑影响

在更改间隔之前：

###络影响：

- 计算：设备 × 计数器 × 间隔 = 每秒查询数
- 短间隔 = 更多网络流量
- 确保网络能够处理增加的负载

存储影响：

- 计算：每天数据点 × 保留期 = 总存储
- 示例：10 秒间隔 = 每个计数器每天 8,640 次测量
- 确保 InfluxDB 有足够的磁盘空间

系统性能：

- 更频繁的轮询 = RAN 监控的 CPU 使用率更高
- 更改后监控系统资源

步骤 3：更改后监控

调整间隔后：

1. 观察应用程序日志 是否有错误
2. 监控系统资源：
   - RAN 监控服务器的 CPU 使用率
   - 网络带宽利用率
   - InfluxDB 磁盘 I/O 和存储增长
3. 验证数据质量：
   - 检查 InfluxDB 是否达到预期的测量频率

- 验证数据中没有缺口

4. 如有需要，进行调整：
    - 如果系统过载，恢复原状
    - 根据观察到的性能进行微调

管理设备□置

如何通过 RAN 监控安全管理基站配置。

□置工作流程

有关详细的配置管理程序，请参见 Web UI 指南 - 配置管理。

快速参考：

1. 下载 当前配置（备份）
2. 修改 离线配置
3. 上传 新配置 - 获取计划 ID
4. 验证 使用计划 ID 进行配置
5. ���活 如果验证成功
6. 验证 更改已生效

最佳实践

始终先下载：

- 保留当前配置的备份
- 如果需要，可以回滚
- 在更改之前记录配置

激活前验证：

- 从不在未验证的情况下激活
- 验证可以捕获语法错误
- 防止服务中断

适当安排更改：

- 尽可能在维护窗口中使用
- 避免高峰流量时段
- 准备回滚计划

记录更改：

- 记录更改的内容及原因
- 记录计划 ID 以便跟踪
- 记录验证结果
- 更新变更管理系统

更改后监控：

- 观察警报
- 验证指标正常化
- 检查设备稳定性 15-30 分钟
- 如果出现问题，准备回滚

监控系统健康

例行检查以确保 RAN 监控正常运行。

每日健康检查

在每个班次开始时执行这些检查：

1. 访问 **Web UI** 仪表板

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

2. 审查系统状态

- 所有设备是否显示绿色（已关联）状态？

- 是否有任何红色（失败）设备需要调查？
- 报警数量是否合理？

3. 检查报警摘要

- 是否有任何关键报警处于活动状态？
- 报警率是上升还是下降？
- 是否有任何重复报警表明系统性问题？

4. 验证数据收集

- 导航到 InfluxDB 状态页面
- 测���计数是否在增加？
- 最后更新的时间戳是否最近？

5. 审查最近的日志

- 导航到应用程序日志页面
- 过滤 "错误" 级别
- 是否有任何重复错误？

有关详细的健康检查程序，请参见 Web UI 指南 - Web UI 工作流程。

示例：**Grafana** 监控仪表板

*□□□□□□□□□□□ LNMME □ S1 □□□□□□□□□□□□□□□□□□□ UEs□□□ PRB □□□□□□□□□□□□□□□□□□□

数据管理

管理数据保留

有关完整详细信息，请参见 数据保留政策指南。

快速参考：

查看当前保留：

- 导航到数据保留页面
- 检查全局默认和每设备设置

调整保留期限：

- 更新特定设备的保留小时数
- 或在 `config/config.exs` 中修改全局默认值

清理旧数据：

- 手动清理：点击 "清理旧数据" 按钮
- 自动清理每小时运行

存储规划：

- 每周监控 InfluxDB 磁盘使用情况
- 根据可用存储调整保留
- 在保留期与存储容量之间取得平衡

导出数据

导出设备□置：

1. 导航到设备详细信息页面
2. 点击 "下载配置"
3. 将 XML 文件保存到安全位置

4. 用设备名称和日期标记

导出指标（通过 **InfluxDB**）：

```
# 导出特定设备的数据
influx -database 'nokia-monitor' -execute '
  SELECT * FROM PerformanceMetrics
  WHERE basebandName='''Site-A-BS1'''
  AND time > now() - 7d
' -format csv > export.csv
```

通过 **Grafana** 导出：

- 打开仪表板
- 选择时间范围
- 点击 "分享" - "导出" - "CSV"

备份与恢复

定期备份

备份内容：

1. 配置文件

```
# 备份运行时配置
cp config/runtime.exs backups/runtime.exs.$(date +%Y%m%d)

# 备份整个配置目录
tar -czf backups/config-$(date +%Y%m%d).tar.gz config/
```

2. 设备配置

- 通过 Web UI 从所有设备下载配置
- 存储在版本控制或备份位置
- 每周执行或在更改前/后执行

3. MySQL 数据库

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

4. InfluxDB 数据

```
# 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)/
```

备份计划

每日：

- MySQL 数据库（会话状态）
- 如果更改，备份配置文件

每周：

- InfluxDB 数据（或根据保留政策）
- 从所有基站下载设备配置

在更改之前：

- 配置文件
- 设备配置
- 数据库快照

恢复过程

从□置错误恢复：

1. 停止 RAN 监控

```
systemctl stop ran_monitor
```

2. 恢复配置文件

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

3. 验证配置

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

4. 重启 RAN 监控

```
systemctl start ran_monitor
```

5. 验证设备重新连接

从数据库丢失恢复：

1. 停止 RAN 监控

```
systemctl stop ran_monitor
```

2. 恢复 MySQL 数据库

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

3. 重启 RAN 监控

```
systemctl start ran_monitor
```

4. 设备将自动重新注册
5. 新的指标收集开始
6. 历史数据保留在 InfluxDB 中

从完整系统丢失恢复：

1. 在新服务器上重新安装 RAN 监控
2. 恢复配置文件
3. 恢复 MySQL 数据库
4. 恢复 InfluxDB 数据
5. 恢复 SSL 证书
6. 启动 RAN 监控
7. 验证所有设备重新连接
8. 验证历史数据可访问

系统维护

日常维护任务

每月：

1. 审查日志

- 存档旧的应用程序日志
- 检查是否有重复的错误或警告
- 清理日志文件以释放磁盘空间

2. 更新文档

- 记录任何配置更改
- 如果拓扑发生变化，更新网络图

- 审查并更新操作程序

3. 安全更新

- 应用操作系统安全补丁
- 如有需要，更新数据库软件
- 根据安全政策轮换密码

4. 性能审查

- 分析系统资源趋势
- 确定是否有性能下降
- ���有需要，计划容量升级

每季度：

1. 灾难恢复测试

- 测试备份恢复过程
- 验证恢复程序有效
- 更新灾难恢复文档

2. 安全审计

- 审查访问日志
- 验证用户权限
- 如果即将到期，更新 SSL 证书

3. 容量规划

- 审查存储增长趋势
- 预测未来的容量需求
- 如有需要，计划硬件升级

每年：

1. SSL 证书续订

- 更换即将到期的 SSL 证书

在到期前测试新证书

2. 密码轮换

更新所有基站凭据
更新数据库密码
更新 API 令牌

3. 系统升级

计划 RAN 监控版本升级
在暂存环境中测试
安排生产升级

维护窗口

规划维护窗口：

1. 在低流量期间安排：

通常夜间或周末最佳
避免在 Grafana 中识别的繁忙时段

2. 通知利益相关者：

通知运营团队
更新状态页面
设置停机期望

3. 准备回滚计划：

备份当前状态
记录回滚步骤
如果升级，准备好以前的版本

4. 执行维护：

按照文档程序进行
密切监控进度

- 记录任何偏差

5. 验证系统健康：
    - 所有设备重新连接
    - 指标正常流动
    - 日志中没有错误
    - 执行健康检查程序
6. 记录结果：
    - 记录所做的工作
    - 记录遇到的任何问题
    - 如有需要，更新程序

相关文档

- 入门指南 - 初始设置程序
- **Web UI** 指南 - 控制面板用户指南
- 运行时□置指南 - 配置参考
- **AirScale** □置 - 基站设置
- 固件管理 - 软件包库和更新
- **Grafana** 集成指南 - 分析和仪表板
- 报警管理指南 - 报警处理程序
- 数据保留政策指南 - 数据生命周期管理
- 故障排除指南 - 问题解决
- 操作指南 - 完整的操作□述

诺基亚 LTE 性能计数器参考

诺基亚 **AirScale/FlexiRadio** 性能测量计数器的综合指南

目录

1. □述
2. 计数器命名约定
3. 计数器类别
4. 资源利用率计数器
5. 吞吐量计数器
6. UE 活动计数器
7. 小区可用性计数器
8. 无线电质量计数器
9. 连接管理计数器
10. 延迟和 QoS 计数器
11. 硬件和无线电单元计数器
12. 在 Grafana 中使用计数器
13. 相关文档

概述

诺基亚 LTE 基站（AirScale, FlexiRadio）使用结构化计数器系统报告性能数据。每个计数器测量网络性能的特定方面，从资源利用率到无线电质量。

什么是性能计数器?

性能计数器是基站在定期间隔内收集的数值测量。它们提供了对以下内���的可见性：

- 网络的繁忙程度（资源利用率）
- 性能的好坏（吞吐量、质量）

连接的用户数量（负载）
服务是否可用（可用性）
信号质量和无线电条件

测量组

计数器被组织成测量组，每个组涵盖特定的功能领域。完整的诺基亚 LTE 计数器集包括以下测量组：

核心性能测量 (M8xxx)

测量	类别	计数器数量	主要关注点
M8000	S1 接口	33	初始上下文设置、S1 设置/重置、UE 上下文
M8001	小区性能	336	PDCP 延迟、RACH、传输块、MCS 分布
M8004	X2 接口	4	X2 数据量、eNB 之间的流量
M8005	无线电质量	237	RSSI、SINR、无线电条件、AMC
M8006	EPS 承载	54	承载设置/修改/释放
M8007	数据无线电承载	14	DRB 建立和管理
M8008	RRC 连接拒绝	14	连接拒绝原因和统计
M8009	切换准备	8	HO 准备失败
M8010	CQI 分布	27	信道质量指示器统计
M8011	资源利用率	55	PRB 使用、资源分配
M8012	吞吐量	121	PDCP 数据速率、体积统计
M8013	信令连接	21	RRC 连接设置尝试/成功
M8014	eNB 之间的切换	14	基于 X2 的切换程序
M8015	eNB 内部切换	13	内部小区切换
M8016	CS 回退	18	电路交换回退程序
M8017	系统间 HO	10	切换到其他 RAT (3G/2G)
M8018	eNB 负载	8	活跃 UE 计数、负载统计

测量	类别	计数器数量	主要关注点
M8019	NACC	4	网络辅助小区切换
M8020	小区可用性	7	小区状态和可用性采样
M8021	频率间 HO	17	频率间切换程序
M8022	X2 设置	2	X2 接口建立
M8023	数据量	36	PDCP SDU 在空中接口上的体积

网络接口测量 **(M5xxx)**

测量	类别	计数器数量	主要关注点
M5112	IP 接口输入	112	传入数据包统计、接口指标
M5113	以太网 RX	21	接收的以太网数据包统计

硬件测量 **(M4xxxx)**

测量	类别	计数器数量	主要关注点
M40001	无线电硬件	可变	VSWR、天线健康、RF 指标
M40002	功耗	可变	基站功耗

计数器命名约定

诺基亚计数器遵循标准命名格式：

```
M<measurement><type>C<counter>
```

示例：**M8011C24**

- **M** - 固定前缀，表示“测量”
- **8011** - 测量组（小区��源）
- **C** - 固定分隔符，表示“计数器”
- **24** - 组内特定计数器（上行 PRB 利用率均值）

逻辑类型

每个计数器都有一个逻辑类型，决定其数据聚合方式：

- 总和 - 事件的累积计数
- 平均 - 测量周期内的均值
- 最小 - 观察到的最小值
- 最大 - 观察到的最大值
- 当前 - 当前采样值
- 分母 - 用于比率计算

计数器类别

按功能领域划分

容量规划：

- M8011Cxx - PRB 利用率
- M8018Cxx - 活跃 UE 计数
- M8012Cxx - 吞吐量速率

性能监控：

- M8001Cxx - 延迟和延时
- M8005Cxx - RSSI、SINR

- M8013Cxx - 设置成功率

可用性跟踪：

- M8020Cxx - 小区可用性
- M8049Cxx - 连接状态

故障排除：

- M8005Cxx - 无线电质量问题
- M8001Cxx - 排队问题
- M8013Cxx - 设置失败

资源利用率计数器

M8011 - 小区资源测量

这些计数器跟踪物理资源块（PRB）利用率，指示小区无线电资源的繁忙程度。

计数器	名称	描述	单位	类型	缩放
M8011C24	上行 PRB 利用率每 TTI 均值	每个传输时间间隔的平均上行 PRB 使用	0.1%	平均	除以 10 得到百分比
M8011C37	下行 PRB 利用率每 TTI 均值	每个传输时间间隔的平均下行 PRB 使用	0.1%	平均	除以 10 得到百分比

理解 **PRB** 利用率：

- 物理资源块（**PRB**）是 LTE 中无线资源分配的最小单位
- **TTI**（传输时间间隔） = LTE 中的 1 毫秒
- 更高的利用率 = 更繁忙的小区（更多流量）
- 100% 利用率表示小区已达到容量

值范围：

- 0-1000（表示 0.0% 到 100.0%）
- 将计数器值除以 10 得到百分比
- 示例：计数器值 453 = 45.3% PRB 利用率

在容量规划中的使用：

- < 50% - 小区有剩余容量
- 50-70% - 正常负载小区
- 70-85% - 重负载，监控拥塞
- > 85% - 接近容量，考虑增加扇区/载波

Grafana 查询示例：

```
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}))  // 转换为百分比
```

吞吐量计数器

M8012 - 小区吞吐量测量

这些计数器测量 PDCP（分组数据汇聚协议）层的吞吐量，指示实际用户数据传输速率。

计数器	名称	描述	单位	类型	更新触发
M8012C23	PDCP 上行吞吐量均值	平均上行 PDCP 吞吐量	kbit/s	平均	当从 UE 接收到 PDCP SDU 时
M8012C26	PDCP 下行吞吐量均值	平均下行 PDCP 吞吐量	kbit/s	平均	当向 UE 传输 PDCP SDU 时

理解 **PDCP** 吞吐量:

- **PDCP** 层是用户数据包处理的地方
- 吞吐量以千比特每秒（kbit/s）为单位测量
- 表示用户流量的实际数据传输速率
- 每秒更新一次

吞吐量计算:

- 在 1 秒采样间隔内测量平均值
- 考虑小区内所有活跃用户
- 包括 VoLTE 和数据承载

性能基准:

下行（**M8012C26**）:

- < 10 Mbps - 低流量/少用户
- 10-50 Mbps - 中等流量
- 50-100 Mbps - 高流量/多活跃用户
- > 100 Mbps - 峰值流量（取决于小区带宽）

上行（**M8012C23**）:

- < 5 Mbps - 低流量
- 5-20 Mbps - 中等流量
- 20-40 Mbps - 高流量
- > 40 Mbps - 峰值流量

Grafana 查询示例:

```
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}))  // 转换为
Mbps
```

UE 活动计数器

M8018 - eNB 负载测量

这些计数器跟踪连接到基站的活跃用户设备（UE）数量。

计数器	名称	描述	单位	类型	更新间隔
M8018C1	每个 eNB 的活跃 UE 最大值	每个 eNodeB 的最大活跃 UE 数量	整数	最大	1 秒

理解 **UE** 活动:

- 活跃 **UE** = 至少有一个信令无线电承载（SRB）和一个数据无线电承载（DRB）的设备
- 在 1 秒采样周期内观察到的最大值
- 指示峰值并发用户负载

负载水平:

活跃 **UE** 数量	负载水平	建议
0-50	低	正常操作
50-100	中等	监控容量
100-150	高	评估增加容量
> 150	非常高	需要扩展容量

注意:

- 实际容量取决于基站硬件配置
- 诺基亚 AirScale 通常可以支持每个小区 150-250 个并发活跃 UE
- 高 UE 数量可能会影响每用户吞吐量

小区可用性计数器

M8020 - 小区可用性测量

这些计数器通过定期采样小区状态来计算小区可用性百分比。

计数器	名称	描述	单位	类型	更新间隔
M8020C3	小区可用样本	小区可用时的样本计数	整数	总和	~10 秒
M8020C4	小区计划不可用样本	小区计划维护时的样本计数	整数	总和	~10 秒
M8020C6	小区可用性分母	可用性检查样本的总数	整数	分母	~10 秒

计算小区可用性:

```
小区可用性 % = 100.0 × M8020C3 / (M8020C6 - M8020C4)
```

公式解释:

- **M8020C3** - 小区正常运行时的样本
- **M8020C6** - 采集的总样本
- **M8020C4** - 计划停机样本（从计算中排除）

可用性目标:

可用性	等级	状态
> 99.9%	优秀	满足运营商级 SLA
99.0-99.9%	良好	正常操作
95.0-99.0%	一般	调查问题
< 95.0%	差	严重 - 需要立即采取行动

Grafana 查询示例:

```
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,
      "小区可用性": 100.0 * r.M8020C3 / (r.M8020C6 - r.M8020C4)
    }))
```

无线电质量计数器

M8005 - 无线电质量测量

这些计数器测量接收信号强度指示器（RSSI）和信号与干扰噪声比（SINR），提供无线电条件的洞察。

RSSI 测量

计数器	名称	描述	单位	类型
M8005C0	PUCCH 最小 RSSI	物理上行控制信道上的最小 RSSI	dBm	最小
M8005C1	PUCCH 最大 RSSI	物理上行控制信道上的最大 RSSI	dBm	最大
M8005C2	PUCCH 平均 RSSI	物理上行控制信道上的平均 RSSI	dBm	平均

理解 **RSSI**:

- **RSSI** = 接收信号强度指示器（总接收功率）
- **PUCCH** = 物理上行控制信道（承载控制信息）
- 以 dBm（分贝毫瓦）为单位测量
- 当计算与 UE 相关的 RSSI 值时更新

RSSI 值解释:

RSSI 范围	质量	描述
> -70 dBm	优秀	信号非常强
-70 到 -85 dBm	良好	信号强，性能良好
-85 到 -100 dBm	一般	信号足够
-100 到 -110 dBm	较差	信号弱，可能存在问题
< -110 dBm	非常差	信号非常弱，可能存在问题

用例：
覆盖分析 - 低 RSSI 表示覆盖缺口
干扰故障排除 - 意外的 RSSI 模式
RF 规划 - 验证信号强度预测

连接管理计数器

M8013 - 信令连接建立

这些计数器跟踪 RRC（无线资源控制）连接设置尝试和成功，这是网络可访问性的关键指标。

计数器 名称 描述 单位 类型

M8013C5 信令连接建立完成 成功的 RRC 连接设置 整数 总和

M8013C17 信令连接建立尝试 MO-S 连接尝试 - 移动发起信令 整数 总和

M8013C18 信令连接建立尝试 MT 连接尝试 - 移动终止 整数 总和

M8013C19 信令连接建立尝试 MO-D 连接尝试 - 移动发起数据 整数 总和

M8013C21 信令连接建立尝试紧急 紧急呼叫连接尝试 整数 总和

M8013C31 信令连接建立尝试高优先级 高优先级连接尝试 整数 总和

M8013C34 信令连接建立尝试延迟容忍 延迟容忍连接尝试 整数 总和

M8013C91 信令连接建立尝试 MO-V 连接尝试 - 移动发起语音 整数 总和

M8013C93 信令连接建立尝试 MT-接入 连接尝试 - MT 接入 整数 总和

计算设置成功率：

```
设置成功率 % = 100.0 × M8013C5 / (M8013C17 + M8013C18 + M8013C19 + M8013C34 + M8013C31 + M8013C21 + M8013C93 + M8013C91)
```

公式解释：

- **M8013C5** - 成功完成（接收到 RRC 连接设置完成）
- 尝试计数器的总和 - 所有类别的总连接尝试

连接类型：

- **MO-S (M8013C17)** - 移动发起信令（短信、位置更新）
- **MT (M8013C18)** - 移动终止（来电/数据）
- **MO-D (M8013C19)** - 移动发起数据（数据会话）
- **MO-V (M8013C91)** - 移动发起语音（VoLTE 通话）
- 紧急 **(M8013C21)** - 紧急呼叫（911、112）

性能目标：

成功率	等级	状态
> 99%	优秀	正常操作
95-99%	良好	可接受的性能
90-95%	一般	建议调查
< 90%	差	严重问题 - 立即排查

常见失败原因：

- 覆盖问题（信号弱）
- 拥塞（小区达到容量）
- 配置错误
- 硬件问题
- 干扰

延迟和 QoS 计数器

M8001 - 小区性能测量

计数器	名称	描述	单位	类型
M8001C2	PDCP SDU 下行延迟均值	eNB 中下行 PDCP SDU 保留时间的均值	毫秒	平均

理解延迟：

- **PDCP SDU** = 分组数据汇聚协议服务数据单元（用户数据包）
- 延迟 = 数据包在基站中等待传输的时间
- 下行 **DTCH** = 下行专用流量信道（用户数据通道）
- 较低的值 = 更好的响应能力

延迟目标：

延迟	等级	应用影响
< 10 毫秒	优秀	适合 VoLTE、游戏、视频通话
10-20 毫秒	良好	大多数应用可接受
20-50 毫秒	一般	在交互式应用中可察觉
> 50 毫秒	差	影响实时应用

高延迟的原因：

- 排队拥塞（用户过多）
- 调度器配置问题
- 无线电条件差（多次重传）
- 回程延迟

S1 接口计数器

M8000 - S1 接口测量

这些计数器跟踪 eNodeB 和 MME（移动管理实体）之间的 S1 接口，包括上下文设置、S1 连接管理和信令程序。

计数器	名称	描述	单位	类型
M8000C0	初始上下文设置请求	初始上下文设置尝试的数量	整数	总和
M8000C1	初始上下文设置完成	成功的初始上下文设置	整数	总和
M8000C2	设置失败 - 无线网络	由于无线网络问题导致的失败	整数	总和
M8000C3	设置失败 - 传输	由于传输层问题导致的失败	整数	总和
M8000C6	S1 设置请求	S1 接口建立尝试	整数	总和
M8000C7	S1 设置成功	成功的 S1 设置	整数	总和
M8000C11	S1 分页请求	来自 MME 的分页消息	整数	总和
M8000C12	UE 逻辑 S1 连接设置	建立的与 UE 相关的 S1 连接	整数	总和
M8000C29	上行 NAS 传输	发送到 MME 的 NAS 消息	整数	总和
M8000C30	下行 NAS 传输	从 MME 接收的 NAS 消息	整数	总和

理解 **S1** 接口：

- **S1** 接口连接 eNodeB 到 EPC（演进分组核心）
- 初始上下文设置 为新的 UE 连接建立上下文
- **S1** 设置 是 eNodeB 和 MME 之间的初始握手
- **NAS**（非接入层）消息承载更高层的信令

成功率计算：

```
S1 设置成功率 = 100 × M8000C7 / M8000C6
初始上下文成功率 = 100 × M8000C1 / M8000C0
```

性能目标：

- S1 设置成功率：> 99%
- 初始上下文成功率：> 95%

EPS 承载计数器

M8006 - EPS 承载测量

这些计数器跟踪 E-UTRAN 无线接入承载（E-RAB）和 EPS 承载的建立、修改和释放程序。

关键计数器：

计数器	名称	描述	单位	类型
M8006C0	EPS 承载设置尝试	承载建立尝试	整数	总和
M8006C1	EPS 承载设置完成	成功的承载设置	整数	总���
M8006C2-C5	按原因分类的设置失败	按原因分类的失败	整数	总和

理解 **EPS** 承载：

- 承载 = UE 和网络之间用户数据的逻辑通道
- 默认承载 = 始终在线的承载，用于互联网连接
- 专用承载 = 针对特定 QoS 要求（VoLTE、视频流）的额外承载

用例：

- 监控承载设置成功率
- 确定承载失败的原因

- 跟踪优质服务承载的分配

切换计数器

M8009 - 切换准备测量

M8014 - eNB 之间的切换测量

M8015 - eNB 内部切换测量

M8021 - 频率间切换测量

这些测量组跟踪切换程序——将 UE 连接从一个小区转移到另一个小区而不掉线的过程。

切换类型：

类型	描述	测量组
内部 **eNB**	在同一基站之间的切换	M8015
eNB 之间	在不同基站之间的切换（X2）	M8014
频率间	切换到不同载波频率	M8021
系统间	切换到不同技术（LTE→3G/2G）	M8017

关键指标：

计数器组	关注点	关键计数器
M8009	切换准备失败	准备尝试、按原因分类的失败
M8014	基于 X2 的 HO	准备尝试、执行、失败
M8015	内部小区 HO	准备尝试、执行、失败
M8021	频率间 HO	尝试、成功、失败

切换成功率公式：

```
切换成功率 = 100 × (切换执行) / (切换准备尝试)
```

性能目标：

- 内部 eNB HO 成功率：> 99%
- eNB 之间 HO 成功率：> 98%
- 频率间 HO 成功率：> 95%

常见切换失败原因：

- 目标小区拥塞（没有可用资源）
- 目标小区的无线电条件差
- 定时器过期（UE 没有及时响应）
- 测量间隙阻止正确的小区选择

信道质量测量

M8010 - CQI（信道质量指示器）分布

这些计数器跟踪来自 UE 的 CQI 报告的分布，提供无线电信道质量的洞察。

理解 **CQI**：

- **CQI** = UE 向 eNodeB 报告的信道质量指示器
- 范围：CQI 0（最差）到 CQI 15（最好）
- 目的：帮助调度器选择适当的 MCS（调制和编码方案）
- 更新频率：根据信道条件每几毫秒更新一次

CQI 到数据速率映射：

CQI 级别	信道质量	近似最大速率	调制
0-3	非常差	< 1 Mbps	QPSK
4-6	较差	1-5 Mbps	QPSK
7-9	一般	5-15 Mbps	16-QAM
10-12	良好	15-40 Mbps	64-QAM
13-15	优秀	40-150 Mbps	64-QAM

M8010 计数器：

- M8010C0 - M8010C15：每个级别（0-15）的 CQI 报告计数

性能分析：

```
平均 CQI = Σ(CQI_level × M8010C[level]) / Σ(M8010C[level])
```

解释 **CQI** 分布：

- 高 **CQI**（**10-15**）：覆盖良好，高吞吐量潜力
- 中等 **CQI**（**7-9**）：覆盖足够，吞吐量适中
- 低 **CQI**（**0-6**）：覆盖问题，考虑小区优化

CS 回退计数器

M8016 - CS 回退测量

这些计数器跟踪电路交换回退（CSFB）程序，其中 LTE UE 为语音通话回退到 2G/3G 网络。

理解 **CSFB**：

- 目的：在没有 VoLTE 的网络上处理语音通话
- 过程：UE 在 LTE → 语音通话 → 临时转移到 2G/3G → 返回 LTE
- 影响：通话设置延迟，临时丢失 LTE 数据

关键计数器：

计数器	名称	描述
M8016C0	带重定向的 CSFB 尝试	使用重定向方法的 CSFB
M8016C1	带切换的 CSFB 尝试	使用切换方法的 CSFB

CSFB 方法：

1. 重定向：UE 从 LTE 断开，重新选择 2G/3G（更快但短暂服务中断）
2. 切换：从 LTE 到 2G/3G 的适当切换（更慢但无缝）

性能指标：

- CSFB 成功率 = 成功过渡 / 总 CSFB 尝试
- 目标：> 98%

数据量计数器

M8023 - PDCP SDU 数据量测量

这些计数器跟踪通过空中接口传输的用户数据的总量。

关键指标：

总数据量（上行和下行）
按 QCI（服务质量类别标识符）划分的体积
按承载类型划分的体积

用例：

网络容量规划
收入估算（数据使用跟踪）
每用户数据消费分析
QoS 类流量分析

与 M8012（吞吐量）的关系：

M8012：瞬时数据速率（kbit/s）
M8023：累计数据量（字节）

X2 接口计数器

M8004 - X2 数据量测量

M8022 - X2 设置测量

这些计数器跟踪 eNodeB 之间的 X2 接口，用于 eNB 之间的切换和负载平衡。

M8004 - X2 数据量：

测量在切换期间在 eNodeB 之间转发的数据
跟踪进出 X2 流量的体积

M8022 - X2 设置：

M8022C0： X2 设置尝试
M8022C1： X2 设置成功

理解 **X2** 接口：

- 目的：邻近 eNodeB 之间的直接通信
- 功能：切换协调、负载共享、干扰管理
- 好处：减少核心网络负载，加快切换速度

X2 设置成功率：

```
X2 设置成功率 = 100 × M8022C1 / M8022C0
```

目标： > 95%

其他测量组

M8007 - 数据无线电承载（DRB）测量

跟踪用户数据传输的数据���线电承载（DRB）的建立、修改和释放。

关注领域：

- DRB 设置成功率
- DRB 修改程序
- 承载释放统计

M8008 - RRC 连接拒绝测量

跟踪被拒绝的 RRC 连接请求，按拒绝原因分类。

常见拒绝原因：

- 网络拥塞（PRB 不足）
- 达到最大 UE 限制
- 负载平衡重定向
- 移动限制

M8019 - 网络辅助小区切换（NACC）

跟踪 NACC 程序，以优化从 LTE 到 GSM 的小区重新选择。

目的：在重新选择之前向 UE 提供 GSM 邻近小区系统信息，加快过渡。

网络接口测量

M5112 - IP 接口输入测量

M5113 - 以太网 RX 测量

这些测量组跟踪回程接口统计数据。

M5112 - IP 接口输入（112 个计数器）：

传入数据包计数
数据包大小分布
协议特定统计
接口利用率

M5113 - 以太网 RX（21 个计数器）：

接收的以太网帧计数
帧大小分布
错误统计（CRC 错误、帧错误）

用例：

回程容量监控
传输链路健康评估
故障排除连接问题
传输升级的容量规划

硬件和无线电单元计数器

M40001 - 无线电硬件测量

计数器	名称	描述	单位	类型	缩放
M40001C0	每个天线分支的 VSWR	电压驻波比	0.1	平均	除以 10

理解 **VSWR**:

- **VSWR** = 电压驻波比
- 测量天线系统效率
- 指示阻抗不匹配和潜在的电缆/天线问题
- 较低的值 = 更好

VSWR 解释:

VSWR	状态	行动
1.0-1.5	优秀	正常操作
1.5-2.0	良好	可接受
2.0-3.0	一般	调查
> 3.0	差	电缆/天线问题 - 立即排查

高 **VSWR** 的常见原因:

- 损坏的同轴电缆
- 松动的连接器
- 水分侵入
- 天线损坏
- 阻抗不匹配

Grafana 查询示例:

```
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 - 功耗测量

计数器	名称	描述	单位	类型	缩放
M40002C2	功耗	基站功耗	100000 因子	平均	除以 100000

理解功耗:

- 测量基站的总功耗
- 对 OPEX 计算和容量规划有用
- 如果意外高/低，可能指示硬件问题

Grafana 查询示例:

```
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}))
```

在 Grafana 中使用计数器

构建有效的仪表板

1. 资源利用率仪表板

结合 M8011C24 和 M8011C37 显示上行/下行 PRB 使用:

```
// 上行 PRB 使用
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": "上行 PRB %"})

// 下行 PRB 使用
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": "下行 PRB %"})
```

2. 吞吐量仪表板

显示数据传输速率：

```
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}))  // 转换为
Mbps
  |> pivot(rowKey: ["_time"], columnKey: ["metricCounter"],
valueColumn: "_value")
  |> map(fn: (r) => ({
      _time: r._time,
      "上行 Mbps": r.M8012C23,
      "下行 Mbps": r.M8012C26
    }))
```

3. 可用性仪表板

计算并显示小区可用性：

```
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,
      "可用性 %": 100.0 * r.M8020C3 / (r.M8020C6 - r.M8020C4)
    }))
```

4. 连接成功率仪表板

跟踪 RRC 设置性能:

```
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,
      "设置成功 %": 100.0 * r.M8013C5 / (r.M8013C17 + r.M8013C18 +
r.M8013C19 + r.M8013C34 + r.M8013C31 + r.M8013C21 + r.M8013C93 +
r.M8013C91)
    }))
```

仪表板最佳实践

使用适当的可视化类型:

- 时间序列图 - 趋势数据（吞吐量、PRB 使用）
- 仪表 - 当前值（可用性百分比）
- 单一状态面板 - 峰值值（最大活跃 UE）
- 热力图 - 分布数据（RSSI 水平）

设置有意义的阈值:

- 绿色: 正常操作

- 黄色：警告（调查）
- 红色：严重（立即行动）

示例阈值□置：

- PRB 利用率：绿色 < 70%，黄色 70-85%，红色 > 85%
- 可用性：绿色 > 99%，黄色 95-99%，红色 < 95%
- 设置成功：绿色 > 99%，黄色 95-99%，红色 < 95%

分组相关指标：

- 为容量、性能和质量创建单独的仪表板
- 使用模板变量进行站点/小区选择
- 包括详细视图的钻取链接

相关文档

- 操作指南 - RAN 监控操作和工作流程
- Grafana 集成指南 - 设置 Grafana 仪表板
- 运行时配置指南 - 配置 RAN 监控
- AirScale 配置指南 - 基站设置
- 指标参考 - InfluxDB 数据结构

快速参考表

最常用的计数器

计数器	名称	用例	单位
M8011C24	上行 PRB 利用均值	容量规划	0.1%
M8011C37	下行 PRB 利用均值	容量规划	0.1%
M8012C23	PDCP 上行吞吐量	性能监控	kbit/s
M8012C26	PDCP 下行吞吐量	性能监控	kbit/s
M8018C1	最大活跃 UE	负载监控	计数
M8020C3	小区可用样本	可用性跟踪	计数
M8020C6	可用性分母	可用性计算	计数
M8013C5	设置完成	成功率跟踪	计数
M8005C0	RSSI 最小值	覆盖分析	dBm
M8005C2	RSSI 均值	覆盖分析	dBm
M8001C2	PDCP 延迟均值	延迟监控	毫秒
M40001C0	VSWR	硬件健康	0.1 比率

来源:

- 诺基亚 FlexiRadio LTE 性能测量
- 诺基亚 AirScale 性能计数器规格
- 3GPP LTE 性能管理标准

附录：完整计数器列表

以下是从诺基亚 KPI 规格中提取的所有 1,186 个诺基亚 LTE 性能计数器的完整参考列表。

```
================================================================================
诺基亚 LTE 性能计数器完整参考
================================================================================

--------------------------------------------------------------------------------
测量组: M8000
--------------------------------------------------------------------------------
M8000C0          | 初始上下文设置请求                                    | 总和
M8000C1          | 初始上下文设置完成                                  | 总和          |
M8000C2          | 初始上下文设置失败由于                              | 总和          | 整[
M8000C3          | 初始上下文设置失败由于                              | 总和          | 整[
M8000C4          | 初始上下文设置失败由于                              | 总和          | 整[
M8000C5          | 初始上下文设置失败由于                              | 总和          | 整[
M8000C6          | S1 设置请求
M8000C7          | S1 设置完成                                                | [
M8000C8          | S1 设置失败由于定时器过期                           | 总和          | 整[
M8000C9          | S1 设置失败由于 MME 拒绝                              | 总和
M8000C11         | S1 分页请求                                                |
M8000C12         | UE 关联的逻辑 S1-                                   | 总和          |
M8000C13         | 由 eNB 发起的全局 S1 重置                                | 总和
M8000C14         | 由 MME 发起的全��? S1 重置                               | 总和
M8000C15         | 由 eNB 发起的部分 S1 重置                              | 总和
M8000C16         | 由 MME 发起的部分 S1 重置                              | 总和
M8000C23         | UE 上下文修改尝试                                    | 总和
M8000C24         | UE 上下文修改尝试由于                               | 总和          |
M8000C25         | UE 上下文修改失败                                    | 总和
M8000C29         | 上行 NAS 传输的数量                                      | 总和
M8000C30         | 下行 NAS 传输的数量                                    | 总和
M8000C31         | UE 上下文修改尝试由于                               | 总和          |
M8000C32         | IMS 的 E-RAB 设置请求                                      |
M8000C33         | IMS 的 E-RAB 设置完成                                  | 总和
M8000C34         | IMS 的 E-RAB 设置失败                                      |
M8000C35         | 位置报告控制的数量                                  | 总和          | 整数
M8000C36         | 位置报告消息的数量                                    | 总和          | [
M8000C37         | 成功的 X2 IP 地址的数量                                | 总和
M8000C38         | 尝试的 X2 IP 地址的数量                                  | 总和
M8000C39         | WRITE-REPLACE 的数量
M8000C40         | WRITE-REPLACE 的数量
M8000C41         | KILL 请求消息的数量                                    | 总和
M8000C42         | KILL 响应的数量                                            | [

--------------------------------------------------------------------------------
测量组: 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 ��□ 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 数据收集指南

概述

PM 数据收集页面允许您管理存储在 InfluxDB 中的性能指标（PM）计数器。诺基亚 AirScale 基站报告超过 22,000 个独特的 PM 计数器，但在大多数用例中，存储所有计数器既不实际也不必要。
本指南解释了如何根据您的监控需求选择要收集的计数器。

快速开始

访问 PM 数据收集页面
1. 导航到控制面板： https://localhost:9443
2. 在导航菜单中点击 数据过滤器
3. 查看和管理 PM 计数器收集设置

理解界面

该页面分为两个主要部分：

部分 描述
存储的 PM 数据（左侧） 当前正在收集并存储在 InfluxDB 中的计数器可用计数器（右侧） 所有 22,000+ 可添加到您的收集中计数器

计数器类别

PM 计数器按其代码前缀分类，前缀指示技术和功能：

类别	代码前缀	数量	描述
LTE	M8xxx	~5,900	LTE L1/L2/L3 计数器（ERAB、RRC、切换等）
WCDMA	M5xxx	~885	3G WCDMA 计数器（MAC 层、CQI、HSDPA）
5G-NR	M55xxx	~14,500	5G NR 计数器（大规模 MIMO、波束成形等）
5G-移动性	M51xxx	~500	5G 移动性和测量
5G-通用	M40xxx	~250	5G 通用/共享计数器

默认计数器

首次启动时，从 `priv/pm_counters.csv` 加载合理的默认值。这些默认值包括以下基本计数器：

- 能量：功耗监控
- 数据量：流量量度
- 可用性：小区可用性统计
- 可接入性：RRC 连接成功/失败
- **PRB**：物理资源块利用率
- 吞吐量：上行/下行吞吐量指标
- **RRC**：RRC 连接统计
- **ERAB**：E-RAB 建立和释放计数器
- **PDCP**：PDCP 层指标
- 切换：小区间切换统计
- 干扰：上行干扰测量

管理计数器

添加计数器

1. 使用“可用计数器”部分中的 搜索框 或 类别过滤器
2. 点击行以选择计数器（复选框将显示为选中）
3. 使用 全选 选择所有可见计数器
4. 点击 添加所选 将它们移至存储集合

移除计数器

1. 在“存储的 PM 数据”部分，选择要移除的计数器
2. 点击 移除所选 停止收集这些计数器

过滤和搜索

两��部分都支持：

- 文本搜索：按计数器 ID 或描述过滤
- 类别过滤器：仅显示来自特定类别的计数器（LTE、5G-NR 等）

重置为默认值

点击 重置为默认值 从 `priv/pm_counters.csv` 恢复原始计数器列表。这将移除任何自定义添加。

持久性

对 PM 计数器选择的更改将：

1. 持久化到磁盘 在 `priv/pm_filters.etf`
2. 在应用程序重启后仍然有效
3. 立即生效（无需重启）

InfluxDB 批处理写入器会被通知更改，并立即开始/停止收集受影响的计数器。

存储考虑

为什么不收集所有数据?

收集所有 22,000+ 个计数器将导致:

场景	影响
存储	每个站点 ~100-500 GB/月（取决于收集间隔）
写入负载	对 InfluxDB 施加显著写入压力
查询性能	由于数据量大，仪表板查询速度较慢
成本	更高的存储和计算成本

推荐方法

1. 从默认值开始：预配置的计数器涵盖大多数常见监控需求
2. 根据需要添加：在构建新仪表板时，添加您需要的特定计数器
3. 定期审查：移除不再使用的计数器

计数器参考

查找计数器描述

“可用计数器”部分显示诺基亚对每个计数器的官方描述。使用搜索功能按以下方式查找计数器:

- 计数器 **ID**（例如，`M8012C23`）
- 描述关键字（例如，`吞吐量`、`切换`、`RSRP`）

常见计数器示例

计数器	类别	描述
M8012C23	LTE	每个小区的平均上行吞吐量
M8012C26	LTE	每个小区的平均下行吞吐量
M8001C2	LTE	PDCP SDU 延迟 DL 平均值
M8011C24	LTE	上行 PRB 利用率
M8011C37	LTE	下行 PRB 利用率
M8013C17	LTE	RRC 连接用户数
M8020C3	LTE	切换成功
M40001C0	5G	能量消耗

□置文件

pm_counters.csv

首次启动时加载的默认计数器：

```
# 格式：计数器,类别,描述
M8012C23,吞吐量,平均上行吞吐量
M8012C26,吞吐量,平均下行吞吐量
M8001C2,可用性,小区可用性
...
```

位置：`priv/pm_counters.csv`

pm_metrics.csv

所有可用计数器的完整参考：

```
# 格式: PM_Code,类别,描述
M8000C6,LTE,S1_SETUP_ATT
M8000C7,LTE,S1_SETUP_SUCC
...
```

位置：`priv/pm_metrics.csv`

故障排除

计数器未被收集

1. 验证计数器是否在“存储的 PM 数据”（左侧）
2. 检查 eNodeB 是否正在推送 PM 数据（请参见 InfluxDB 状态页面）
3. 验证计数器 ID 是否完全匹配（区分大小写）

更改未生效

1. 过滤更改会立即应用于批处理写入器
2. 新数据仅在 **eNodeB** 下次推送 **PM** 数据后出现（通常每 15 分钟一次）
3. 检查应用程序日志中的 `[PmFilterStore]` 错误
4. 验证磁盘是否可写以进行持久化文件

缺少计数器描述

1. 计数器描述来自 `priv/pm_metrics.csv`
2. 确保该文件存在且格式正确
3. 检查 UTF-8 编码问题

相关文档

数据保留政策 - PM 数据保留多长时间
Grafana 集成 - 使用 PM 数据构建仪表板
InfluxDB 查询 - 查询 PM 数据

访问点

PM 数据收集： https://localhost:9443/nokia/pm-filters
数据保留： https://localhost:9443/nokia/retention InfluxDB 状态： https://localhost:9443/nokia/influx

RAN 监控运行时□置指南

理解 **config/runtime.exs**

目录

1. □述
2. 数据库配置
3. Web 端点
4. 日志记录配置
5. 诺基亚集成
6. InfluxDB 配置
7. 配置最佳实践

概述

`config/runtime.exs` 文件是 RAN 监控的主要配置文件。它在运行时（应用程序启动时）进行评估，允许您配置系统行为的各个方面。

□置内容：

- 数据库连接（MySQL）
- Web 服务器端点和端口
- 诺基亚基站详细信息
- InfluxDB 时间序列数据库
- 日志记录行为
- 安全凭证

文件位置：

```
config/runtime.exs
```

谁应该使用本指南

重要：所有 RAN 监控配置由 **Omnitouch** 执行，作为初始部署和持续支持的一部分。本指南提供给：

- 高级用户，希望了解系统配置
- 自我管理的部署，客户维护自己的配置
- 故障排除 和理解系统如何配置
- 定制部署，具有特定要求

如果您是 **Omnitouch** 管理的客户，请联系 **Omnitouch** 支持以进行任何□置更改。

要了解正在收集的数据，请参见 诺基亚计数器参考。有关仪表板创建，请参见 Grafana 集成。

数据库□置

MySQL/MariaDB 连接

```
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
```

目的：配置用于会话状态管理和操作数据的 MySQL 数据库连接。

参数解释

username (字符串)

- 数据库用户帐户
- 当前值：`"omnitouch"`
- 用法：必须具有 CREATE、SELECT、INSERT、UPDATE、DELETE 权限

- 安全性：考虑使用具有最低所需权限的专用用户

password (字符串)

- 用于身份验证的数据库密码
- 当前值：`"omnitouch2024"`
- 安全性：应在生产环境中存储在环境变量中
- 建议：使用强大且唯一的密码

hostname (字符串)

- 数据库服务器地址
- 当前值：`"localhost"`
- 选项：
  - `"localhost"` - 数据库在同一台机器���
  - `"127.0.0.1"` - TCP 连接到本地机器
  - `"10.179.2.135"` - 远程数据库服务器 IP
  - `"db.example.com"` - 远程数据库主机名

database (字符串)

- 要使用的数据库名称
- 当前值：`"ran_monitor"`
- 注意：数据库必须在启动 RAN 监控之前存在
- 创建：`CREATE DATABASE ran_monitor CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;`

stacktrace (布尔值)

- 在错误消息中包含堆栈跟踪
- 当前值：`true`
- 开发：`true` - 有助于调试
- 生产：`false` - 减少日志噪声

show_sensitive_data_on_connection_error (布尔值)

- 在连接错误消息中显示凭证

- 当前值：`true`
- 开发：`true` - 更容易故障排除
- 生产：`false` - 防止凭证在日志中暴露

pool_size（整数）

- 要维护的数据库连接数量
- 当前值：`10`
- 大小指南：
  - 1-5 个设备：`pool_size: 5`
  - 6-20 个设备：`pool_size: 10`
  - 21-50 个设备：`pool_size: 15`
  - 50+ 个设备：`pool_size: 20`
- 公式：每个基站大约 2 个连接 + 5 个用于 Web UI

Web 端点

RAN 监控运行多个 Web 服务器以满足不同目的。

主要 **SOAP/API** 端点

```
config :ran_monitor, RanMonitor.Web.Endpoint,
  http: [ip: {0, 0, 0, 0}, port: 8080],
  check_origin: false,
  secret_key_base:
"v5tOS1/QRonjwOky7adGGfkBbrJmiJyXhpesJy/jvSZhqLZkREV+rlo1/pR8lkbu",
  server: true
```

目的：基站通信的主要端点（诺基亚 NE3S 协议的 SOAP 接口）。

ip（元组）

- 要绑定的接口
- 当前值：`{0, 0, 0, 0}`（所有接口）
- 选项：

{0, 0, 0, 0} - 监听所有网络接口
{127, 0, 0, 1} - 仅在本地主机上监听
{10, 179, 2, 135} - 在特定 IP 地址上监听

port (整数)

TCP 端口号
当前值：8080
注意：基站必须配置为将数据发送到此端口
防火墙：确保端口对基站 IP 开放

check_origin (布尔值)

验证 WebSocket/HTTP 来源头
当前值：false
解释：对于 SOAP API 设置为 false（不是面向用户的 Web UI）

secret_key_base (字符串)

会话的加密签名密钥
当前值：64 字符随机字符串
生成： mix phx.gen.secret
安全性：保持此密钥的秘密，绝不要提交到公共存储库
影响：更改此项会使所有现有会话失效

server (布尔值)

应用程序启动时启动端点
当前值：true
始终：在 runtime.exs 中应为 true

控制面板 Web UI

```
# 从环境变量获取 HTTPS 端口，默认为 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"
  ]
```

目的：用于基于 Web 的控制面板 UI 的 HTTPS 端点。

环境变量：

- **CONTROL_PANEL_HTTPS_PORT** - HTTPS 端口号（默认：9443）
  - 设置此环境变量以在运行时更改 HTTPS 端口
  - 示例：`export CONTROL_PANEL_HTTPS_PORT=8443`

url (关键字列表)

- 外部 URL 配置
- **host:** `"0.0.0.0"` - 接受来自任何主机的连接
- **port:** 使用 `https_port` 变量（可通过 CONTROL_PANEL_HTTPS_PORT 配置）
- **scheme:** `"https"` - 使用 HTTPS 协议

https (关键字列表)

- HTTPS 服务器配置
- **ip:** `{0, 0, 0, 0}` - 绑定到所有接口
- **port:** 使用 `https_port` 变量（必须与 url 端口匹配）
- **keyfile:** SSL 私钥路径
- **certfile:** SSL 证书路径

SSL 证书文件：

- 必须是有效的 SSL/TLS 证书
- 自签名证书适用于实验室环境
- 生产环境应使用 CA 签名的证书
- 生成自签名证书：

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

诺基亚 **AirScale Webhook** 端点

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

目的：接收来自诺基亚 AirScale 基站的实时性能数据。

port (整数)

- 当前值：`9076`
- 注意：必须与基站 PMCADM 中配置的端口（rTpmCollEntityPortNum）匹配
- 协调：此端口必须与您在诺基亚 WebLM 参数编辑器中配置的端口匹配

日志记录□置

```
config :logger,
  level: :info

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

日志级别

level (原子)

- 控制日志记录的详细程度
- 当前值: `:info`
- 选项:
  - `:debug` - 极其详细，所有细节
  - `:info` - 正常操作，推荐用于生产
  - `:warning` - 仅警告和错误
  - `:error` - 仅错误

何时使用每个级别:

- 开发: `:debug` - 查看所有内部操作
- 生产: `:info` - 在可见性和噪声之间取得平衡
- 故障排除: 暂时设置为 `:debug`，然后恢复
- 安静的生产: `:warning` - 仅在出现问题时发出警报

控制台格式

format (字符串)

- 日志消息的显示方式
- 当前值: `"$time $metadata[$level] $message\n"`
- 变量:
  - `$time` - 时间戳
  - `$metadata` - 上下文信息
  - `$level` - 日志级别（info、error 等）
  - `$message` - 实际日志消息

metadata (原子列表)

- 要包含的附加上下文
- 当前值: `[:request_id]`
- **request_id:** 跟踪系统中的单个 HTTP 请求

诺基亚集成

本节配置 RAN 监控如何与诺基亚基站通信。

```
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"
      }
    ]
  }
```

一般设置

mcc (字符串)

- 移动国家代码
- 当前值: `"505"`
- 用法：标识 3GPP 网络的国家
- 格式： 3 位数字

- 参考：ITU-T E.212

mnc (字符串)

- 移动网络代码
- 当前值：`"57"`
- 用法：标识特定的网络运营商
- 格式：2 或 3 位数字

NE3S ⍰⍰置（诺基亚 NE3S 协议）

webhook_url (字符串)

- 基站发送通知的 URL
- 当前值：`"http://10.5.198.200:9076/webhook"`
- 格式：`http://<ran-monitor-ip>:<port>/webhook`
- **IP** 地址：必须是 RAN 监控运行的 IP 地址
- 端口：必须与 `RanMonitor.Web.Nokia.Airscale.Endpoint` 端口（9076）匹配
- 路径：始终为 `/webhook`

private_key (字符串 - 文件路径)

- 管理员身份验证的私钥
- 当前值：`priv/external/nokia/ne.key.pem`
- 格式：PEM 编码的私钥
- 安全性：保持此文件安全，绝不要分享
- 生成：由诺基亚提供或使用 OpenSSL 生成

public_key (字符串 - 文件路径)

- 管理员身份的公共证书
- 当前值：`priv/external/nokia/ne.cert.der`
- 格式：DER 编码的证书
- 用法：在注册期间发送给基站
- □对：必须与 private_key 对应

reregister_interval (整数)

- 与基站重新注册的频率（秒）
- 当前值：`30`
- 解释：会话会过期，定期重新注册以保持连接
- 范围： 30-300 秒
- 建议： 30 秒以确保可靠监控

AirScale 基站

airscales (映射列表)

- 要监控的诺基亚 AirScale 基站列表
- 当前值：配置了一个基站

每个基站条目需要：

address (字符串)

- 基站的 IP 地址
- 当前值：`"10.7.15.66"`
- 格式：作为字符串的 IPv4 地址
- 网络：必须可以从 RAN 监控服务器访问
- 验证： `ping 10.7.15.66` 应该成功

name (字符串)

- 用于识别的友好名称
- 当前值：`"ONS-Lab-Airscale"`
- 用法：出现在 Web UI、日志和 InfluxDB 标签中
- 建议：使用描述性名称（站点代码、位置等）
- 示例：
  - `"NYC-Site-A-BS1"`
  - `"LAX-Tower-Main"`
  - `"TestLab-Airscale-01"`

port (字符串)

- 基站上的管理接口端口

当前值："8080"
标准：诺基亚 AirScale 通常使用 8080
验证：检查基站文档
注意：值为字符串，而不是整数

web_username (字符串)

WebLM 身份验证的用户名
当前值："Nemuadmin"
用法：用于 API 调用以管理基站
权限：应具有配置读/写访问权限
注意：区分大小写

web_password (字符串)

WebLM 身份验证的密码
当前值："nemuuser"
安全性：应在生产环境中存储在环境变量中
轮换：根据安全策略定期更改

添加多个基站

要监控多个基站，请向 airscales 列表添加额外条目：

```
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 □置

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

目的：配置与 InfluxDB 时间序列数据库的连接，以存储指标、警报和配置数据。

参数解释

auth (关键字列表)

- InfluxDB 的身份验证凭证
- **username:** InfluxDB 用户帐户（`"monitor"`）
- **password:** InfluxDB 密码（`"sideunderTexasgalaxyview_61"`）
- 注意：对于 InfluxDB 2.x，这可能是 API 令牌

database (字符串)

- InfluxDB 中的桶/数据库名称
- 当前值：`"nokia-monitor"`
- **InfluxDB 1.x**：数据库名称
- **InfluxDB 2.x**：桶名称
- 创建：必须在启动 RAN 监控之前创建

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

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

host (字符串)

- InfluxDB 服务器地址
- 当前值：`"10.179.2.135"`
- 格式：IP 地址或主机名
- 端口：默认 InfluxDB 端口（8086）假定
- 示例：
  - `"localhost"` - 与 RAN 监控相同的服务器
  - `"10.179.2.135"` - 远程 InfluxDB 服务器
  - `"influxdb.example.com"` - 主机名

InfluxDB 连接注意事项

网络访问：

RAN 监控必须能够在端口 8086 上访问 InfluxDB 服务器验证：curl http://10.179.2.135:8086/ping

保留策略：

通过 Web UI 数据保留页面设置
默认：30 天（720 小时）
可以根据基站自定义

写入性能：

InfluxDB 每个采集间隔（默认 60 秒）接收写入
每个基站每个间隔生成数百个数据点
定期监控 InfluxDB 磁盘空间

□置最佳实践

安全性

1. 保护敏感数据

```
# 而不是硬编码密码:
password: "omnitouch2024"

# 使用环境变量:
password: System.get_env("DB_PASSWORD") || "default_password"
```

2. 限制文件权限

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

3. 永远不要提交秘密

- 如果 runtime.exs 包含秘密，请使用 `.gitignore`
- 使用环境变量或秘密管理系统
- 定期轮换密码

性能

1. 数据库池大小

- 监控连接使用情况
- 如果看到连接超时错误，请增加 pool_size
- 每个设备在主动轮询期间需要约 2 个连接

2. 采集间隔

- 在数据粒度和系统负载之间取得平衡
- 60 秒的间隔适用于大多数部署
- 较短的间隔（15 秒）用于故障排除

3. InfluxDB 优化

- 使用保留策略管理磁盘使用情况
- 监控 InfluxDB 写入性能
- 对于大型部署，考虑单独的 InfluxDB 服务器

可靠性

1. 网络□置

- 对所有组件使用静态 IP 地址
- 验证 RAN 监控与基站之间的网络路由
- 在添加设备之前测试连接
- 适当地配置防火墙规则

2. 日志记录策略

- 开发：`: debug` 用于详细故障排除
- 生产：`: info` 用于操作可见性
- 关键系统：考虑外部日志聚合

3. 监控 RAN 监控

- 监控监控（元监控）
- 观察数据库连接错误
- 跟踪 InfluxDB 写入成功率
- 对基站断开连接发出警报

维护

1. □置更改

- 在更改之前始终备份 runtime.exs
- 首先在开发中测试配置
- 使用注释记录��改
- 在配置更改后重新启动 RAN 监控

2. 添加基站

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

# 2. 验证 Elixir 语法
elixir -c config/runtime.exs

# 3. 重新启动应用程序
systemctl restart ran_monitor
```

3. 扩展考虑

- 监控资源使用情况（CPU、内存、网络）
- 随着设备数量的增加增加数据库池大小
- 在 50+ 个设备时考虑单独的 InfluxDB 实例
- 监控 MySQL 和 InfluxDB 的磁盘空间

示例：完整□置

以下是应用多个基站和最佳实践的完整示例：

```
import Config

#
============================================================================
# 数据库配置
#
============================================================================

config :ran_monitor, RanMonitor.Repo,
  username: System.get_env("DB_USERNAME") || "ran_monitor_user",
  password: System.get_env("DB_PASSWORD") || "change_this_password",
  hostname: System.get_env("DB_HOST") || "localhost",
  database: "ran_monitor",
  stacktrace: false,  # 生产: 隐藏堆栈跟踪
  show_sensitive_data_on_connection_error: false,  # 生产: 隐藏凭证
  pool_size: 15  # 6 个基站 * 2 + 3 的开销

#
============================================================================
# Web 端点
#
============================================================================

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
```

```
#
================================================================================
# □□□□□□
#
================================================================================

config :logger,
  level: :info  # □□□□

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

#
================================================================================
# □□□□□
#
================================================================================

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
    },
    airscales: [
      # □□ A - □□
      %{
        address: "10.7.15.66",
        name: "Site-A-Main-Tower",
        port: "8080",
        web_username: "admin",
        web_password: System.get_env("BS_SITE_A_PASSWORD") || "passwo
      },
```

```
    # 站点 A - 备用塔
    %{
      address: "10.7.15.67",
      name: "Site-A-Backup-Tower",
      port: "8080",
      web_username: "admin",
      web_password: System.get_env("BS_SITE_A_PASSWORD") || "passwo
    },

    # 站点 B - 室内
    %{
      address: "10.7.16.10",
      name: "Site-B-Indoor-DAS",
      port: "8080",
      web_username: "admin",
      web_password: System.get_env("BS_SITE_B_PASSWORD") || "passwo
    },

    # 站点 C - 屋顶
    %{
      address: "192.168.100.50",
      name: "Site-C-Rooftop",
      port: "8080",
      web_username: "admin",
      web_password: System.get_env("BS_SITE_C_PASSWORD") || "passwo
    },

    # 实验室 - 测试设备
    %{
      address: "10.5.198.100",
      name: "Lab-Test-Airscale-01",
      port: "8080",
      web_username: "Nemuadmin",
      web_password: "nemuuser"
    },

    # 实验室 - 开发
    %{
      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_passw
  ],
  database: "nokia-monitor",
  host: System.get_env("INFLUX_HOST") || "10.179.2.135"
```

相关文档

- 操作指南 - 日常操作
- AirScale 配置指南 - 配置基站
- 诺基亚计数器参考 - 性能计数器定义
- Grafana 集成 - 构建仪表板和警报
- API 端点 - REST API 参考
- 数据保留政策 - 管理数据生命周期

MDT 数据收集与 TCE

跟踪收集实体 (TCE)

RAN Monitor 包含一个集成的跟踪收集实体，用于捕获和分析 LTE/5G 协议消息。这使得详细的故障排除、驱动测试和 RF 优化成为可能。

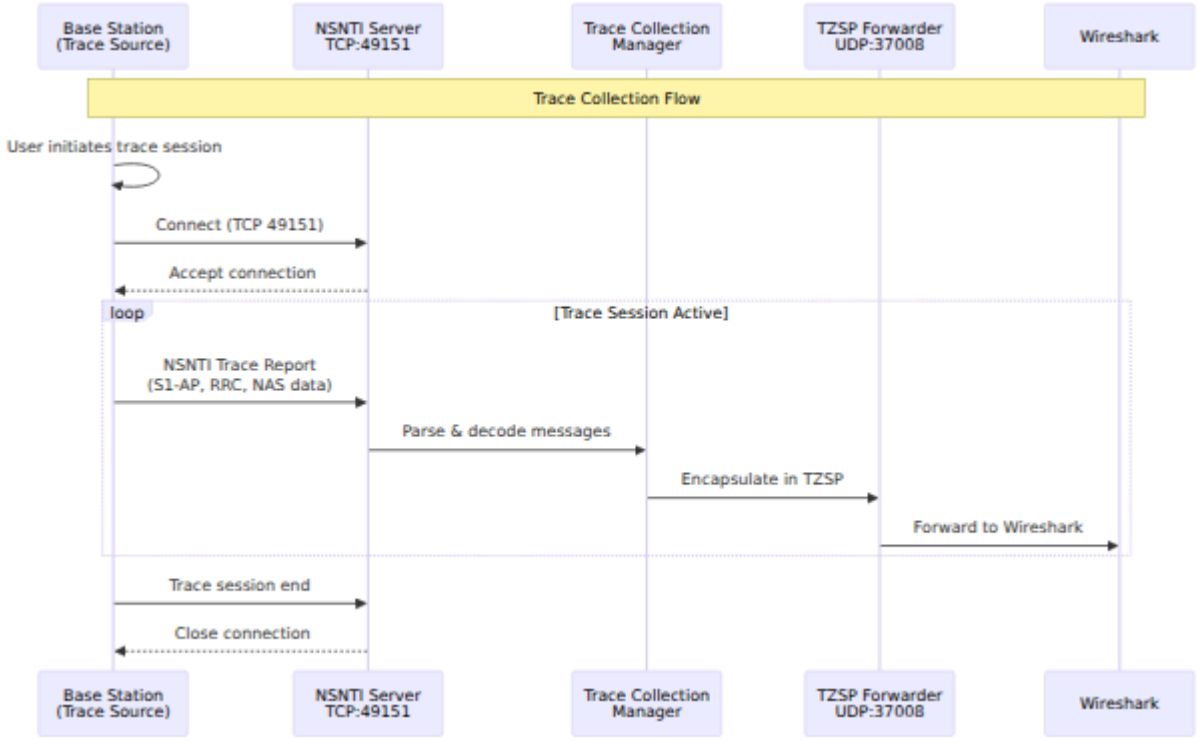
什么是 TCE？

跟踪收集实体接收来自 Nokia AirScale 基站的跟踪数据，包含：

- **S1-AP** 消息 - eNodeB 和 EPC 之间的控制平面信令
- **RRC** 消息 - 无线资源控制信令
- **NAS** 消息 - 非接入层信令
- 用户平面数据 - PDCP 层吞吐量信息

TCE 组件

组件	协议	端口	目的
NSNTI 服务器	TCP	49151	接收来自基站的跟踪消息
TZSP 服务器	UDP	37008	将跟踪转发到 Wireshark 进行实时分析
协议解码器	ASN.1	-	解码 S1-AP 和 RRC 消息



工作原理

□□□□□□□□□□□□□□NSNTI □□□□ (49151)□TZSP □□□□□□□□□

跟踪收集设置

1. 验证 **TCE** 是否正在运行:

```
ss -tlnp | grep 49151
# 应显示: LISTEN 0.0.0.0:49151
```

2. □置基站跟踪:

   - 将跟踪目标 IP 设置为 RAN Monitor 服务器
   - 将跟踪目标端口设置为 49151
   - 根据需要启用跟踪类别 (S1-AP、RRC、NAS)
   - 启动跟踪会话

3. □置 **Wireshark**:

   基本设置:

   - 在接收 TZSP 数据包的接口上启动捕获

- 使用捕获过滤器：`udp port 37008`

协议解码□置：

RAN Monitor 使用特定的 UDP 端口来识别不同的协议类型和 RRC 通道。配置 Wireshark 的“解码为”功能以正确解码这些协议：

方法 **1**：使用 **Wireshark GUI**

i. 转到 **Analyze → Decode As...**
ii. 点击 **+** 按钮以添加新条目
iii. 按如下方式配置每一行：

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

□□ 2□□□ decode_as_entries □□

创建或编辑 `~/.config/wireshark/decode_as_entries` (Linux/Mac) 或 `%APPDATA%\Wireshark\decode_as_entries` (Windows):

```
# RAN Monitor TZSP 端口映射
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
```

端口参考指南:

端口	协议	通道/类型	描述
36412	S1AP	-	标准 S1AP 控制平面 (eNodeB ↔ EPC)
37000	RRC	通用	未知 RRC 通道类型的后备
37001	RRC	DL-CCCH	下行公共控制信道
37002	RRC	DL-DCCH	下行专用控制信道
37003	RRC	BCCH-DL-SCH	广播控制信道 (���统信息)
37004	RRC	PCCH	Paging Control Channel
37008	TZSP	-	主要 TZSP 监听端口
37011	RRC	UL-CCCH	上行公共控制信道 (RRC 连接请求)
37012	RRC	UL-DCCH	上行专用控制信道 (测量报告)
38000	MAC-LTE	通用	未知 MAC 通道类型的后备
38001	MAC-LTE	下行	下行共享信道
38002	MAC-LTE	BCH	广播信道
38003	MAC-LTE	PCH	Paging Channel
38011	MAC-LTE	上行	上行共享信道
38012	MAC-LTE	RACH	随机接入信道

有用的显示过滤器:

```
# 显示所有 TZSP 数据包
tzsp

# 显示特定协议
s1ap || rrc || mac-lte

# 仅显示上行 RRC 消息
udp.port == 37011 || udp.port == 37012

# 仅显示下行 RRC 消息
udp.port == 37001 || udp.port == 37002

# 显示 RRC 连接建立
rrc.rrcConnectionRequest || rrc.rrcConnectionSetup

# 显示切换消息
s1ap.HandoverRequired || s1ap.HandoverCommand
```

用例

驱动测试:

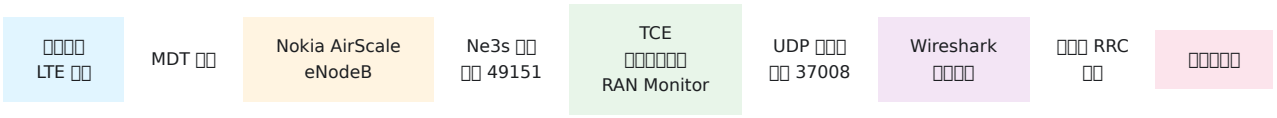
- 捕获最终用户 RF 体验
- 分析切换性能
- 测量信号质量 (RSRP, RSRQ, SINR)
- 识别覆盖盲区

故障排除:

- 调试呼叫设置失败
- 分析切换问题
- 调查掉线呼叫
- 审查移动事件

RF 优化:

- PCI 规划验证
- 邻区关系优化



- 切换参数调整
- 覆盖和容量分析

概述

最小化驱动测试 **(MDT)** 允许您直接从 UE 收集无线测量 (RSRP、RSRQ、覆盖数据)，而无需传统的驱动测试。本指南向您展示如何使用 Omnitouch RAN Monitor Web UI 从 Nokia AirScale 基站捕获 MDT 数据，并在 Wireshark 中查看。

架构

TCE（跟踪收集实体）集成在 Omnitouch RAN Monitor 中，负责将 Nokia 特定的 Ne3s 协议跟踪转换为可在 Wireshark 中查看的标准格式。

先决条件

所需许可证

Nokia Airscale 需要激活功能，包括 每通话测量数据 以收集此数据，并启用和配置这些功能。

如果您需要有关许可证的帮助或对您的特定部署有疑问，请联系 ONS。

系统要求

- Omnitouch RAN Monitor 与 TCE 正在运行
- 在您的机器上安装 Wireshark 3.0 及以上版本
- 在 Wireshark 中安装 TCE Lua 插件 (请参见 TCE README)
- 与 AirScale 的网络连接

设置 MDT 跟踪

内置于 RAN Monitor 的 TCE 将传入的 Nokia 数据转换为标准 Wireshark 可查看的格式。

第 1 步：□置跟踪收集实体

使用 RAN Monitor Web UI 设置基站以将跟踪发送到 TCE：

1. 打开 Web UI：`https://<ran-monitor-ip>:9443`
2. 导航到 基站 页面
3. 点击您想要跟踪的设备
4. 转到 □置管理 部分
5. 下载当前配置（备份）
6. 编辑配置以添加/更新 TCE 设置：
   - 跟踪收集实体 **IP**：`<您的 RAN Monitor IP>`
   - 跟踪收集实体端口：`49151`
7. 上传修改后的配置
8. 验证配置（等待验证完成）
9. 激活配置

有关特定□置参数或 **AirScale** 软件版本的帮助，请联系 **ONS**。

第 2 步：在 AirScale 上□置 MDT

在您的基站上启用 MDT 跟踪。配置选项包括：

- 跟踪类型：即时 MDT（实时）或记录 MDT（空闲模式）
- 区域范围：特定小区、跟踪区域或 PLMN 范围
- 测量间隔：UE 报告的频率（例如，5000ms）
- 测量类型：RSRP、RSRQ 或两者
- 跟踪深度：最小、中等或最大

有关如何为您的特定用例□置这些参数的指导，请联系 **ONS**。

第 3 步：激活跟踪会话

配置完成后，在 AirScale 上激活跟踪会话。基站将开始将 MDT 数据发送到 TCE，后者将其转发到您的监控机器。

在 Wireshark 中查看 MDT 数据

设置 Wireshark 捕获

1. 在您的机器上启动 Wireshark
2. 在 回环接口 上捕获（Linux 上为 `lo`，macOS 上为 `lo0`，Windows 上为 `Loopback`）
3. 设置捕获过滤器：`udp port 37008`
4. 开始捕获

□□ Wireshark □□□□□ S1AP □□□□□□ (InitialUEMessage, Attach request)□LTE RRC □□ (RRCConnectionReject, RRCConnectionReestablishment) □□□ TCE □□□□□□□□□

过滤 MDT 测量

一旦数据流动，使用以下显示过滤器：

```
# 显示所有 RRC 测量报告
lte-rrc.measurementReport

# 显示所有上行 RRC 消息
udp.dstport >= 37011 && udp.dstport <= 37012

# 按低 RSRP 过滤 (< -100 dBm)
lte-rrc.rsrpResult < 40

# 按低 RSRQ 过滤 (< -12 dB)
lte-rrc.rsrqResult < 22
```

理解数据

MDT 测量以 **RRC MeasurementReport** 消息的形式出现，包含：

- 服务小区测量：连接小区的 RSRP 和 RSRQ
- 邻小区测量：附���小区的 RSRP 和 RSRQ
- 小区 **ID**：用于关联的物理小区 ID
- **GPS** 位置：如果配置并由 UE 支持

在 Wireshark 中展开 RRC 消息以查看详细测量：

```
无线资源控制 (RRC)
└─ UL-DCCH-Message
    └─ message: measurementReport
        └─ MeasurementReport
            └─ measResults
                ├─ measResultServCell (服务小区 RSRP/RSRQ)
                └─ measResultNeighCells (邻小区测量)
```

导出以进行分析

要离线分析数据：

1. 文件 → 导出数据包解析 → 作为 **CSV**
2. 包括字段：`lte-rrc.rsrpResult`、`lte-rrc.rsrqResult`、`lte-rrc.physCellId`

3. 在 Excel、Python 或其他工具中处理

常见用例

覆盖分析：查找 RSRP/RSRQ 弱的区域

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

切换分析：查看 UE 报告的邻小区

```
lte-rrc.MeasResultListEUTRA
```

干扰检测：良好的 RSRP 但差的 RSRQ 表示干扰

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

故障排除

Wireshark 中没有数据?

- 验证 TCE 是否正在运行：`ps aux | grep beam`
- 检查 Wireshark 是否使用过滤器 `udp port 37008` 捕获回环
- 确认 AirScale 上的跟踪会话处于活动状态
- 验证基站上配置的 TCE IP/端口是正确

数据不完整?

- 检查许可证是否有效 (MDT + 每通话测量)
- 将跟踪深度增加到最大
- 确保 UE 支持 MDT (LTE Release 10+)

有关置帮助、许可证问题或 **AirScale** 特定问题，请联系 **ONS**。

快速启动检查表

验证 MDT 和每通话测量许可证是否有效
在 AirScale 上配置 TCE IP (RAN Monitor IP) 和端口 49151 在 RAN Monitor 服务器上启动 TCE
在基站上激活 MDT 跟踪会话
在回环上启动 Wireshark 捕获，过滤器为 udp port 37008 应用显示过滤器：lte-rrc.measurementReport
分析测量并根据需要导出

支持

Omnitouch 网络服务 (ONS)：有关 AirScale 配置、许可证和部署的帮助

故障排除指南

RAN 监控器问题解决

□□□□□□□□□□□□□□□

目录

1. □述
2. 设备连接问题
3. 数据收集问题
4. Web UI 问题
5. 数据库问题
6. 性能问题
7. 报警问题
8. 诊断工具
9. 获取帮助

概述

本指南帮助您诊断和解决 RAN 监控器的常见问题。每个部分提供症状、诊断步骤和解决方案。

故障排除方法

1. 确定症状

- 什么没有按预期工作?
- 问题何时开始?
- 最近有什么变化?

2. 收集信息

- 检查应用程序日志
- 在 Web UI 中查看设备状态
- 检查数据库连接
- 查看最近的配置更改

3. 诊断根本原因

- 使用诊断工具
- 查看错误消息
- 测试单个组件
- 隔离问题

4. 实施解决方案

- 根据诊断结果应用修复
- 验证解决方案是否解决了问题
- 监控是否复���
- 记录发现

开始之前

检查基本情况：

- RAN 监控器是否正在运行？（`ps aux | grep ran_monitor`）
- 所需服务是否正在运行？（MySQL, InfluxDB）
- 网络连接是否正常？
- 最近是否有更改？

设备连接问题

问题：设备未注册

症状：

- 设备在 Web UI 中显示“未注册”

- 基站页面显示红色（失败）状态
- 没有从设备收集到指标
- 应用程序日志中有错误消息

诊断步骤：

1. 验证网络连接

```
# 测试基本连接
ping <device-ip>

# 测试管理端口
telnet <device-ip> 8080
```

预期：成功的 ping 和 telnet 连接

如果失败：网络问题 - 检查路由、防火墙、设备状态

2. 检查□置

在 Web UI → 基站 → 单击设备 → 查看配置：

- IP 地址是否正确？
- 端口是否正确（通常为 8080）？
- 凭据是否已配置？

在 `config/runtime.exs` 中：

```
%{
  address: "10.7.15.66",   # 正确的 IP？
  name: "Site-A-BS1",
  port: "8080",            # 正确的端口？
  web_username: "admin",   # 正确的用户名？
  web_password: "password" # 正确的密码？
}
```

3. 检查应用程序日志

Web UI → 应用程序日志 → 按设备名称过滤

查找：

[error] Authentication failed → 凭据不正确 [error] Connection refused → 端口/防火墙问题 [error] Timeout → 网络连接问题
[error] Certificate error → 管理员密钥/证书问题

解决方案：

网络问题：

1. 验证设备已开机并正常运行
2. 检查 RAN 监控器与设备之间的网络路由
3. 验证防火墙允许：
RAN 监控器 → 设备端口 8080
设备 → RAN 监控器端口 9076（webhooks）
4. 从 RAN 监控器服务器直接测试

凭据不正确：

1. 验证凭据在设备 WebLM 界面上是否有效
2. 更新 config/runtime.exs 中的凭据
3. 重启 RAN 监控器
4. 监控日志以确认注册成功

端口/防火墙问题：

1. 验证配置中的端口是否正确
2. 检查两侧的防火墙规则
3. 测试端口可访问性：telnet <device-ip> 8080
4. 查看设备侧的安全设置

管理员密钥/证书问题：

1. 验证文件是否存在：
priv/external/nokia/ne.key.pem priv/external/nokia/ne.cert.der
2. 检查文件权限（应可读）

3. 验证文件是否为有效的 Nokia 管理凭据
4. 如果密钥无效，请联系 Nokia 支持

问题：会话不断过期

症状：

- 设备反复断开和重新连接
- 日志中出现“会话过期”消息
- Web UI 中状态间歇性��示红色/绿色
- 指标收集存在间隙

诊断步骤：

1. 检查会话信息

Web UI → 基站 → 单击设备 → 会话生命周期：

- 会话过期时间是多少？
- 保持活动是否正常？
- 会话多久过期一次？

2. 检查保持活动间隔

在 `config/runtime.exs` 中：

```
nokia: %{
  ne3s: %{
    reregister_interval: 30  # 应为 30-60 秒
  }
}
```

3. 检查网络稳定性

- 是否存在间歇性网络问题？
- 检查丢包情况：`ping <device-ip> -c 100`
- 查看网络日志以查找抖动接口

4. 检查时钟同步

```
# 在 RAN 监控器服务器上
date

# 在设备上（如果可访问）
# 验证时间是否同步
```

解决方案：

保持活动间隔过长：

1. 将 `reregister_interval` 减少到 30 秒
2. 重启 RAN 监控器
3. 监控会话稳定性

网络不稳定：

1. 与网络团队合作进行诊断
2. 检查是否存在间歇性连接
3. 查看交换机/路由器日志
4. 考虑冗余网络路径

时钟同步：

1. 在 RAN 监控器和设备上配置 NTP
2. 验证时钟是否同步
3. 检查是否存在较大时间差异

设备侧问题：

1. 检查设备日志中的错误
2. 验证设备管理接口是否稳定
3. 如果怀疑软件问题，请考虑重启设备

❓❓题：指标未出现

症状：

- 设备在 Web UI 中显示为“已关联”（绿色）
- 但在 InfluxDB 中没有出现指标
- Grafana 仪表板中没有数据
- InfluxDB 状态页面显示零或低计数

诊断步骤：

1. 验证设备是否已关联

Web UI → 基站：

- 设备状态是绿色吗？
- 最近的联系时间戳是最近的吗？
- 会话是否处于活动状态？

2. 检查 InfluxDB 连接

Web UI → InfluxDB 状态：

- 连接状态是绿色吗？
- RAN 监控器能否写入 InfluxDB？

测试连接：

```
# 从 RAN 监控器服务器
curl http://<influxdb-host>:8086/ping
```

3. 检查应用程序日志

查找：

- `[error] InfluxDB write failed` → 连接或权限问题
- `[error] Failed to collect metrics` → 设备通信问题
- `[info] Metrics collected: 0` → 设备未返回数据

4. 直接检查 InfluxDB

查询 InfluxDB 以获取最近的数据:

```
# 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()'
```

解决方案:

InfluxDB 连接问题:

1. 验证 InfluxDB 是否正在运行
2. 检查 `config/runtime.exs` 中的正确配置:
   - 主机地址
   - 端口 (8086)
   - 数据库/桶名称
   - 凭据/API 令牌
3. 从 RAN 监控器服务器测试连接
4. 验证防火墙是否允许端口 8086
5. 修复配置后重启 RAN 监控器

InfluxDB 权限问题:

1. 验证凭据是否具有写入桶/数据库的权限
2. 检查 InfluxDB 日志以查找身份验证错误
3. 重新创建具有适当权限的 API 令牌
4. 使用新令牌更新 `config/runtime.exs`

5. 重启 RAN 监控器

InfluxDB 存储已满:

1. 检查磁盘空间: `df -h`
2. 查看保留策略
3. 清理旧数据或扩展存储
4. 参见 数据保留政策指南

设备未返回数据:

1. 检查设备是否配置为发送指标
2. 验证设备配置中的 webhook URL 是否正确
3. 检查设备日志中的错误
4. 验证 RAN 监控器 webhook 接收器是否正在运行（端口 9076）

数据收集问题

问题：历史数据存在间隙

症状:

- Grafana 仪表板显示时间序列存在间隙
- 某些时间段缺少数据点
- InfluxDB 查询返回不完整的结果

诊断步骤:

1. 检查应用程序正常运行时间

□□间隙期间是否有服务中断?

```
# □查系统日志以获取重启信息
journalctl -u ran_monitor --since "2025-12-29" --until "2025-12-
30"
```

2. 检查设备连接历史

Web UI → 基站 → 设备 → 查看“最后联系”历史

设备在间隙期间是否连接？
是否存在连接问题？

3. 检查 InfluxDB 可用性

在间隙期间是否有 InfluxDB 中断？

检查 InfluxDB 日志
查看监控/警报历史

解决方案：

RAN 监控器停机：

服务中断期间数据间隙是正常的
历史数据无法回填
记录事件并恢复服务

设备断开连接：

调查设备断开的原因
修复连接问题
断开连接期间数据间隙是正常的
未来数据将恢复收集

InfluxDB 中断：

指标可能已收集但未存储
检查 RAN 监控器日志以获取写入失败信息
恢复 InfluxDB 服务
数据间隙无法恢复

预防：

实施 RAN 监控器正常运行时间的监控

- 设置延长断开连接的警报
- 监控 InfluxDB 健康状况
- 考虑对关键系统进行高可用性/冗余

Web UI 问题

问题：无法访问 Web UI

症状：

- 浏览器无法连接到 https://<ran-monitor-ip>:9443
- 连接超时或被拒绝
- SSL 证书错误

诊断步骤：

1. 验证 Web UI 是否正在运行

检查应用程序日志：

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

检查进程：

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

2. 测试连接

从另一台机器：

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

从 RAN 监控器服务器本身：

```
curl -k https://localhost:9443
```

3. 检查防火墙

```
# 查端口是否开放
sudo iptables -L -n | grep 9443

# 或
sudo firewall-cmd --list-ports
```

解决方案：

端口未开放：

1. 添加防火墙规则：

   ```
   sudo firewall-cmd --add-port=9443/tcp --permanent
   sudo firewall-cmd --reload
   ```

2. 再次测试访问

Web UI 未启动：

1. 检查 `config/runtime.exs` 中的 Web 端点配置
2. 验证 SSL 证书文件是否存在
3. 检查应用程序日志以获取启动错误
4. 重启 RAN 监控器

SSL 证书问题：

1. 验证证书文件是否存在且可读：

   ```
   ls -l priv/cert/omnitouch.pem
   ls -l priv/cert/omnitouch.crt
   ```

2. 检查证书有效性：

   ```
   openssl x509 -in priv/cert/omnitouch.crt -text -noout
   ```

3. 如果过期或缺失，请重新生成
4. 重启 RAN 监控器

端口错误：

1. 检查 `config/runtime.exs` 中配置的端口
2. 在浏览器中使用正确的端口
3. 或设置 `CONTROL_PANEL_HTTPS_PORT` 环境变量

问题：**Web UI** 加载但不显示数据

症状：

- Web UI 可访问
- 页面加载但显示空列表或零计数
- 仪表板显示没有设备

诊断步骤：

1. 检查设备□置

在 `config/runtime.exs` 中是否配置了任何内容？

```
airscales: [
  # 至少应有一个设备
]
```

2. 检查数据库连接

设备是否存储在 MySQL 中？

```
mysql -u ran_monitor_user -p ran_monitor -e "SELECT * FROM
airscales;"
```

3. 检查应用程序日志

查找数据库连接错误或查询失败。

解决方案：

未□置设备：

1. 在 `config/runtime.exs` 中添加设备
2. 重启 RAN 监控器
3. 设备应出现在 Web UI 中

数据库连接问题：

1. 验证 MySQL 是否正在运行
2. 检查 `config/runtime.exs` 中的连接配置
3. 测试数据库连接
4. 重启 RAN 监控器

数据库问题

问题：**MySQL** 连接错误

症状：

- 应用程序日志显示数据库连接错误
- Web UI 显示加载页面时出错
- “数据库连接超时”消息

诊断步骤：

1. 验证 **MySQL** 是否正在运行

```
systemctl status mysql
# 或
systemctl status mariadb
```

2. 测试连接

从 RAN 监控器服务器：

```
mysql -h <mysql-host> -u <username> -p <database>
```

3. 检查□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
```

解决方案:

MySQL 未运行:

1. 启动 MySQL 服务:
   ```
   systemctl start mysql
   ```
2. 验证其是否正确启动
3. RAN 监控器将自动重新连接

连接□置错误:

1. 验证主机名、用户名、密码、数据库名称
2. 手动测试连接
3. 如果不正确，请更新 `config/runtime.exs`
4. 重启 RAN 监控器

网络问题:

1. 检查到 MySQL 服务器的网络连接
2. 验证防火墙是否允许端口 3306
3. 检查 MySQL 绑定地址（如果需要，应允许远程连接）

连接过多:

1. 检查 MySQL max_connections 设置
2. 如果需要，减少配置中的 pool_size
3. 重启 RAN 监控器

性能问题

问题：高 **CPU** 或内存使用率

症状：

- RAN 监控器使用过多的 CPU 或 RAM
- 系统变得缓慢或无响应
- 数据库连接超时
- 响应时间降低

诊断步骤：

1. 检查资源使用情况

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

# 详细进程信息
ps aux | grep ran_monitor
```

2. 检查监控的设备数量

配置了多少设备？

- 设备越多 = 需要的资源越多
- 检查设备数量是否最近增加

3. 检查收集间隔

轮询间隔是否非常频繁？

- 更频繁 = 更高的 CPU/网络使用
- 默认是每 10 秒收集一次指标

4. 检查数据库连接池大小

在 `config/runtime.exs` 中:

```
pool_size: 10  # 可能需要调整
```

解决方案:

设备数量过多导致资源不足:

1. 监控资源使用趋势
2. 增加服务器资源（CPU/RAM）
3. 或减少监控的设备数量
4. 考虑水平扩展（多个实例）

数据库连接池过大:

1. 在配置中减少 pool_size
2. 一般规则: 每个设备 2 个连接 + 5 个用于 Web UI
3. 重启 RAN 监控器
4. 监控资源使用情况

内存泄漏:

1. 监控内存使用情况
2. 如果持续增加，可能存在内存泄漏
3. 作为临时修复，重启 RAN 监控器
4. 报告问题并附上日志和指标

InfluxDB 写入性能:

1. 检查 InfluxDB 资源使用情况
2. 验证 InfluxDB 是否是瓶颈
3. 考虑单独的 InfluxDB 服务器
4. 查看保留策略以减少数据量

问题: **Web UI** 响应缓慢

症状:

- Web UI 加载页面需要很长时间
- 仪表板反应迟钝
- 查看设备详细信息时超时

诊断步骤:

1. 检查服务器资源

RAN 监控器服务器是否过载?

```
top
free -h
df -h
```

2. 检查数据库性能

数据库查询是否缓慢?

```
# MySQL 慢查询日志
mysql -u root -p -e "SHOW VARIABLES LIKE 'slow_query_log%';"
```

3. 检查网络延迟

到数据库或客户端的延迟是否很高?

解决方案:

服务器资源问题:

1. 减少服务器负载
2. 增加服务器资源
3. 将数据库移动到单独的服务器

数据库性能：

1. 优化 MySQL 配置
2. 如果需要，添加索引（表应有索引）
3. 增加数据库服务器资源

网络延迟：

1. 调查网络路径
2. 考虑将组件移动得更近
3. 如果可能，使用本地数据库

报警问题

问题：报警未出现

症状：

- 已知故障未显示在报警页面
- 当故障存在时报警计数为零
- 报警通知延迟

诊断步骤：

1. 检查设备是否发送报警

在设备管理界面验证报警是否配置为发送。

2. 检查 Webhook 接收器

Webhook 端点是否正在运行?

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

查找：

```
tcp  0  0.0.0.0:9076  0.0.0.0:*  LISTEN
```

3. 检查 Webhook 配置

在设备配置中，验证 webhook URL 是否指向 RAN 监控器:

```
http://<ran-monitor-ip>:9076/webhook
```

4. 检查应用程序日志

查找 webhook 接收器错误或报警解析失败。

5. 检查 InfluxDB

报警是否被写入?

```
influx -database 'nokia-monitor' -execute '
  SELECT COUNT(*) FROM Alarms WHERE time > now() - 1h
'
```

解决方案:

Webhook 接收器未运行:

1. 检查 `config/runtime.exs` 中的 webhook 端点配置
2. 验证端口 9076 是否已配置
3. 重启 RAN 监控器
4. 验证端口是否在监听

设备未发送:

1. 配置设备以发送报警通知
2. 验证设备配置中的 webhook URL
3. 测试设备上的报警生成

防火墙阻塞:

1. 验证设备是否可以访问 RAN 监控器端口 9076
2. 如有需要，添加防火墙规则
3. 从设备网络测试连接：`telnet <ran-monitor-ip> 9076`

InfluxDB 写入失败：

1. 检查 InfluxDB 连接
2. 验证写入权限
3. 检查 InfluxDB 存储容量
4. 查看应用程序日志以获取写入错误

诊断工具

应用程序日志

通过 **Web UI** 访问：

1. 导航到应用程序日志页面
2. 按日志级别过滤
3. 搜索关键字
4. 暂停以查看特定错误

通过命令行访问：

如果作为 systemd 服务运行：

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

如果通过 mix 运行：

- 日志显示在控制台输出中

日志级别：

- 紧急/警报/严重 - 系统关键问题

- 错误 - 需要关注的错误
- 警告 - 潜在问题
- 信息 - 正常操作消息
- 调试 - 详细诊断信息

有用的搜索词：

- 设备名称（例如，“Site-A-BS1”）
- “error” 或 “failed”
- “InfluxDB” 或 “MySQL”
- “registration” 或 “session”

InfluxDB 查询

查询最近的指标：

```
influx -database 'nokia-monitor' -execute '
  SELECT * FROM PerformanceMetrics
  WHERE basebandName='''Site-A-BS1'''
  AND time > now() - 5m
  LIMIT 10
'
```

按设备计数指标：

```
influx -database 'nokia-monitor' -execute '
  SELECT COUNT(*) FROM PerformanceMetrics
  GROUP BY basebandName
'
```

查询报警：

```
influx -database 'nokia-monitor' -execute '
  SELECT * FROM Alarms
  WHERE time > now() - 1h
'
```

MySQL 查询

检查□置的设备:

```
SELECT name, address, port, registration_status
FROM airscales;
```

检查数据库中的错误:

```
mysql -u ran_monitor_user -p ran_monitor -e "SHOW PROCESSLIST;"
```

网络诊断

测试连接:

```
# 基本连接
ping <device-ip>

# 端口可访问性
telnet <device-ip> 8080
nc -zv <device-ip> 8080

# 路由跟踪
traceroute <device-ip>
```

检查防火墙:

```
# 列出规则
sudo iptables -L -n -v

# □查特定端口
sudo iptables -L -n | grep 8080
```

获取帮助

联系支持之前

收集以下信息：

1. 问题描述

什么没有工作？
何时开始的？
最近有什么变化？

2. 错误消息

复制日志中的确切错误消息
包括时间戳
注意错误的频率

3. 系统信息

RAN 监控器版本
操作系统及版本
数据库版本（MySQL、InfluxDB）
监控的设备数量

4. 诊断结果

上述诊断步骤的结果
相关日志摘录
配置（清理密码）

5. 影响

受影响的设备数量？
这是否阻碍了操作？
业务影响是什么？

文档资源

- **Web UI** 指南 - 控制面板参考
- 常见操作指南 - 日常任务
- 运行时□置指南 - 配置细节
- **Grafana** 集成指南 - 分析设置
- 报警管理指南 - 报警处理
- 数据保留政策指南 - 数据管理
- 操作指南 - 完整□述

自助���源

首先检查日志：

- Web UI 中的应用程序日志页面
- 系统日志：`journalctl -u ran_monitor`
- 数据库日志

查看最近的更改：

- 配置文件修改
- 设备添加/删除
- 网络更改
- 软件更新

测试基本功能：

- 您能访问 Web UI 吗？
- 设备是否显示为连接？
- InfluxDB 是否可访问？
- 指标是否流动？

升级

如果您无法解决问题：

1. 记录所有已采取的诊断步骤
2. 收集上述信息
3. 联系 Omnitouch 支持并提供详细信息
4. 准备提供:
    - 配置文件（清理过的）
    - 日志摘录
    - 相关截图（如适用）
    - 重现步骤

相关文档

- 操作指南 - 完整操作□述
- **Web UI** 指南 - 控制面板用户指南
- 常见操作指南 - 日常任务
- 报警管理指南 - 报警处理程序
- 运行时□置指南 - 配置参考
- **Grafana** 集成指南 - 分析和仪表板
- 数据保留政策指南 - 数据生命周期管理

Web UI 指南

RAN 监控控制面板 - 用户界面参考

□□ RAN □□□□ Web □□□□□□□□□□

目录

1. □述
2. 访问 Web UI
3. 主仪表板
4. 基站页面
5. 设备详细视图
6. 警报页面
7. 配置管理
8. 未配置的 eNodeB 页面
9. 应用日志页面
10. 数据保留政策页面
11. InfluxDB 状态页面
12. 系统指标页面
13. PM 数据收集页面
14. 固件管理页面
15. 数据管理页面
16. Web UI 工作流

相关文档

- 固件管理指南 - 完整的固件库文档

概述

RAN 监控包括一个内置的基于 Web 的控制面板，用于实时操作监控和管理。Web UI 提供设备状态、警报、配置和系统健康的即时可见性。

Web UI 与 Grafana

Web UI 最适合：

- 立即检查设备状态
- 实时警报监控
- 配置管理
- 会话故障排除
- 系统管理

Grafana 最适合：

- 历史趋势分析
- 自定义 KPI 仪表板
- 长期容量规划
- 模式识别
- 高管报告

有关 Grafana 仪表板和分析，请参见 Grafana 集成指南。

访问 Web UI

控制面板通过 HTTPS 访问：

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

默认端口**:** 9443（可通过 `CONTROL_PANEL_HTTPS_PORT` 环境变量配置）

SSL 证书：

- 自签名证书适用于实验室环境

- 生产环境应使用 CA 签名的证书
- 证书配置在 `config/runtime.exs` 中

有关配置详细信息，请参见 运行时配置指南。

自动刷新：大多数页面每 5 秒自动刷新一次，以显示实时数据。

主仪表板

仪表板提供您 RAN 基础设施的□览。

关键部分

系统状态

- 整体健康指标
- 系统正常运行时间和连接性

设备摘要

- 关联/失败设备的计数
- 注册状态□览
- 快速设备健康快照

活动警报

- 当前故障计数按严重性分类
- 颜色编���的严重性级别（关键，主要，次要，警告）
- 快速链接到警报详细信息

最近活动

- 最新事件和更改
- 配置更新
- 会话状态更改

特性

- 每 5 秒自动刷新
- 颜色编码的状态指示器（绿色 = 健康，红色 = 问题）
- 点击导航到详细视图
- 实时指标更新

基站页面

查看所有管理设备及其当前状态和会话信息。

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

□□□□□□□□□□□□□□□□□□□□□□□□ NOKIA eNodeB □□□□□

统计摘要

顶部栏显示设备的汇总计数：

统计信息	描述
总设备	配置设备的数量
已连接	具有活动会话的设备
待处理	等待注册的设备
未连接	没有活动会话的设备

设备表

列	描述
名称	按配置的设备名称
状态	连接状态：“已连接”（绿色）或“未连接”（红色）
地址	设备 IP 地址和端口
会话	会话状态：“活动”（绿色）或“非活动”（灰色）
操作	设备操作按钮

操作按钮

每个设备行都有操作按钮：

按钮	描述
Ping	测试与设备的网络连接
□置	查看当前设备配置
□置操作	访问配置管理操作（下载、上传、验证、激活）
强制重试	强制重新注册未连接设备

设备详细面板

单击设备行显示附加详细信息：

字段	描述
管理器 **ID**	内部管理器标识符
会话 **ID**	当前会话标识符
代理类型	设备代理类型（例如，COMA）
供应商	设备供应商（Nokia）

过滤和搜索

- 按连接状态过滤
- 按设备名称或 IP 地址搜索
- 按任何列排序

设备详细视图

从基站页面单击任何设备以查看全面信息。

注册详细信息

- 管理器身份和认证状态
- 注册时间戳
- 使用的认证凭据
- 管理器密钥和证书

会话生命周期

- 会话创建时间
- 会话到期时间
- 保持活动间隔和状态
- 最后保持活动时间戳
- 到期前剩余时间

最近指标

- 最新性能数据快照
- 计数器值和时间戳
- 指标收集状态
- 数据收集间隔

活动警报

- 此特定设备的当前故障
- 警报严重性和描述
- 警报时间戳
- 可能的原因信息

□置状态

- 当前参数值
- 最近的配置更改
- 配置时间戳
- 参数更改历史

警报页面

在一个集中视图中监控网络中的所有故障。

警报信息

严重性级别：

关键（红色） - 影响服务，需立即采取行动
主要（橙色） - 显著降级，需紧急关注
次要（黄色） - 不影响服务，应该处理
警告（蓝色） - 信息性，监控趋势
已清除（绿色） - 先前活动的警报已解决

警报详细信息：

问题描述
可能原因
受影响的系统（DN - Distinguished Name）
时间戳（警报发生和最后更新时）

特性

颜色编码：

立即可视化识别严重性
红色 = 关键警报
橙色 = 主要警报
黄色 = 次要警报
蓝色 = 警告
绿色 = 已清除

排序和过滤：

按严重性、设备或时间排序

- 按警报类型过滤
- 搜索特定问题

设备链接：

- 单击警报查看受影响设备的详细信息
- 与设备指标交叉引用
- 导航到设备配置

有关详细的警报处理程序，请参见 警报管理指南。

□置管理

Web UI 提供安全高效的设备配置管理工具。

下载□置

目的：检索和备份当前配置

步骤：

1. 导航到设备详细页面
2. 单击“下载配置”
3. 从设备检索配置
4. 将配置保存为 XML 文件

最佳实践：在进行更改之前，始终下载并保存配置

上传□置

目的：将新配置应用到设备

步骤：

1. 选择 XML 配置文件
2. 单击“上传配置”

3. 配置上传到设备（创建“计划”）
4. 系统返回一个计划 ID 以进行跟踪

重要：上传仅创建一个计划 - 不会激活配置

验证□置

目的：在激活之前验证配置是否有效

步骤：

1. 输入上传的计划 ID
2. 单击“验证”
3. 设备验证语法和参数
4. 系统确认准备好激活或报告错误

注意：始终在激活之前进行验证，以防止配置错误

激活□置

目的：应用验证过的配置计划

步骤：

1. 输入验证过的计划 ID
2. 单击“激活配置”
3. 更改立即在设备上生效
4. 监控状态以确认成功/失败

警告：激活是立即生效的，可能会影响服务 - 确保首先通过验证

□置工作流

推荐流程：

1. 下载当前配置（备份）
2. 离线修改配置
3. 上传新配置（获取计划 ID）

4. 验证配置（确保没有错误）
5. 如果验证成功，则激活
6. 验证更改是否生效
7. 监控设备稳定性

有关基站配置的详细信息，请参见 AirScale 配置指南。

未□置的 eNodeBs 页面

发现并管理试图连接但尚未在系统中配置的基站。

目的

未配置的 eNodeBs 页面帮助您:

- 发现网络上的新基站
- 识别试图进行未授权连接的设备
- 在添加到配置之前验证设备标识符
- 跟踪未知设备的连接尝试

显示的信息

代理 **ID**

- 从连接尝试中检测到的设备标识符
- 在将设备添加到配置时使用此 ID

最后一次看到

- 最近一次连接尝试的时间戳
- 有助于识别活动与非活动设备

发生次数

- 设备尝试连接的次数
- 频繁的尝试可能表示配置错误

首次看到

- 设备首次被检测到的时间
- 有助于跟踪新设备

可用的操作

刷新

- 重新加载未配置设备的列表
- 更新时间戳和发生次数

删除

- 从列表中删除单个条目
- 有助于清理旧的/退役的设备

清除所有

- 删除所有未配置设备记录
- 列表的新开始

□置帮助

当设备出现在这里时，请按照以下步骤操作：

1. 记下表中的代理 **ID**
2. 将设备□置添加到 **`config/runtime.exs`** 中：

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

3. 重启 **RAN** 监控 以开始监控设备

有关详细的配置说明，请参见 运行时配置指南。

用例

- 网络发现：查找添加到网络的新基站
- 安全性：识别未授权的连接尝试
- □置：在配置之前验证设备标识符
- 退役：跟踪应处于离线状态的设备的尝试

应用日志页面

实时日志仪表板，用于故障排除和监控系统活动。

日志级别

按日志级别过滤：

- 紧急 - 系统关键故障
- 警报 - 需要立即采取行动
- 关键 - 关键条件
- 错误 - 错误条件
- 警告 - 警告条件
- 通知 - 正常但重要
- 信息 - 信息性消息
- 调试 - 详细调试信息

注意：过滤时，所选级别及所有更高严重性级别将显示。

特性

搜索和过滤：

- 在所有日志消息中进行文本搜索

- 实时日志流（最近 500 条消息）
- 按日志级别过滤

控制：

- 暂停/恢复 - 停止实时日志流以查看消息
- ���除 - 从显示中删除所有日志
- 系统级别 - 动态更改应用程序范围的日志级别

颜色编码：

- 红色 - 紧急/警报/关键级别
- 浅红色 - 错误级别
- 黄色 - 警告级别
- 青色 - 通知级别
- 蓝色 - 信息级别
- 灰色 - 调试级别

用例

故障排除连接问题：

- 过滤特定设备的错误
- 搜索设备名称或 IP 地址
- 查看连接失败消息

监控系统活动：

- 观察信息级日志以获取正常操作
- 跟踪设备注册事件
- 监控数据收集活动

调试问题：

- 暂时设置调试级别
- 重新产生问题
- 查看详细日志

完成后恢复到信息级别

调查故障：

搜索错误消息和堆栈跟踪
查看故障时间附近的时间戳
与设备事件关联

最佳实践

在查看特定错误序列时使用暂停
设置适当的日志级别：
生产使用信息级
故障排除使用调试级
安静生产使用警告级
有效搜索 使用设备名称或错误关键字
日志级别更改在应用程序重启之前保持

数据保留政策页面

管理每个基站在 InfluxDB 中存储数据的时间长度。

全局设置显示

默认保留期

系统范围内的保留政策（小时/天）
在 config/config.exs 中配置默认：720 小时（30 天）

总记录数

所有设备的所有数据点计数
在页面刷新时更新

自动清理状态

- 显示每小时运行的清理
- 后台工作者状态

每设备设置

对于每个配置的基站：

设备信息：

- 设备名称
- 注册状态（已注册/未注册）
- 当前保留期设置

记录计数：

- 性能指标 - 存储的 PM 数据点数量
- □置 - 存储的配置快照数量
- 警报 - 数据库中的警报记录数量
- 总计 - 此设备的所有记录总和

操作：

- 更新保留期 - 更改保留小时（仅适用于此设备）
- 清理旧数据 - 根据保留期手动触发清理
- 清除所有数据 - 删除此设备的所有数据（不可逆）

保留工作原理

1. 全局默认 - 在配置文件中设置，适用于所有设备
2. 每设备覆盖 - 可选择为特定设备设置自定义保留
3. 自动清理 - 每小时运行，删除超过保留期的数据
4. 手动清理 - 使用“清理旧数据”强制立即清理

常见保留期

- **720** 小时（**30** 天） - 短期操作监控
- **2160** 小时（**90** 天） - 大多数部署的标准保留
- **4320** 小时（**180** 天） - 合规的扩展保留
- **8760** 小时（**365** 天） - ���期历史分析

用例

- 通过降低保留期减少存储使用
- 保留关键设备数据时间长于其他数据
- 在生产之前清理测试数据
- 管理 InfluxDB 磁盘空间使用

警告：清除所有数据是永久性的，无法撤销。执行前请务必确认。

有关详细的保留政策信息，请参见 数据保留政策指南。

InfluxDB 状态页面

监控您的 InfluxDB 时间序列数据库的健康和状态。

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

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ InfluxDB □□□□□

□□□□

□□	□□
□□□□	□□□□□□□□□□□□□□□□
□□□	□□□ InfluxDB □□□□□
InfluxDB □□	□□□□□□□□□□□2.x□

□□□□□□

□□□□□□□□□□□□□□□□□

测量	描述
性能指标	从设备收集的 PM 数据点
□置	存储的配置快照
警报	数据库中的警报记录
总计	所有数据点的总和

批处理写入器性能

处理所有数据摄取的 InfluxDB 批处理写入器过程的统计信息：

指标	描述
队列大小	等待写入的点数。颜色编码：绿色（< 1000），黄色（< 10000），橙色（< 20000），红色（>= 20000）
过滤率	被阻止写入的重复数据点的百分比
PM 过滤	被过滤掉的 PM 计数器的数量（不在存储 PM 数据列表中的非仪表板计数器）
队列丢弃	由于队列溢出而丢弃的点数（正常操作时应为 0）
□置缓存	用于增量检测的唯一配置哈希的数量
警报缓存	用于增量检测的活动警报的数量

附加指标：

指标	描述
总写入	自启动以来写入 InfluxDB 的累积点数
刷新次数	批处理刷新操作的数量
过滤	被过滤的重复点的总数（未写入）
写入数据	写入 InfluxDB 的总字节数
吞吐量	当前写入吞吐量（KB/s 或 MB/s）
写入器正常运行时间	自批处理写入器启动以来的时间
最后刷新	自上次成功刷新以来的时间

清除缓存：重置增量检测缓存。使用时希望强制重新写入所有数据（例如，在模式更改后）。

存储信息

字段	描述
保留政策	当前保留设置（默认：无限）
磁盘使用	根据记录计数估算的数据库大小
活动	最后更新的时间戳

置详细信息

字段	描述
主机	InfluxDB 服务器主机名
端口	InfluxDB 服务器端口（默认：8086）
存储桶	InfluxDB 存储桶名称
状态	连接状态徽章
测量	测量类型的数量（3：PerformanceMetrics, Configuration, Alarms）

健康诊断

系统健康的状态指示器：

- **InfluxDB** 连接性 - 数据库可访问并响应
- 数据收集 - 从设备收集性能指标
- 数据保留 - 当前保留政策状态
- 最后同步 - 最近的数据同步

自动刷新

页面每 30 秒自动刷新。

解读状态

条件	意义
连接 + 数据计数增长	系统正常运行
连接 + 无数据	检查设备注册和 PM 计数器配置
未连接	验证 InfluxDB 是否正在运行和网络连接
高队列大小（黄色/红色）	InfluxDB 写入性能问题或网络延迟
高队列丢弃	队列溢出 - 增加批处理大小或减少收集速率
高过滤率	良好 - 表示有效的重复检测

用例

- 验证 InfluxDB 是否���收数据
- 监控批处理写入器的健康和吞吐量
- 故障排除写入性能问题
- 检查增量检测缓存的有效性
- 确认数据正在写入

系统指标页面

实时监控 InfluxDB 批处理写入器和系统资源的性能。

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

□□ InfluxDB □□□□□□□□□□□□□ Airscale □□□□□□□□□□□□

InfluxDB 批处理写入器

批处理写入器过程的摘要统计信息：

指标	描述
队列大小	等待写入到 InfluxDB 的数据点数量
刷新计数	自启动以来的批处理刷新总数
总写入点	写入 InfluxDB 的累积数据点
最后刷新	自上次成功刷新操作以来的时间

按 Airscale 的 InfluxDB 写入统计信息

写入到 InfluxDB 的每个设备的数据分解：

列	描述
Airscale	设备名称
性能指标	写入的 PM 数据点计数
□置	写入的配置快照计数
警报	写入的警报记录计数
总记录	此设备的所有数据点总和
最后写入	此设备最近写入的时间戳

底部的总计行显示所有设备的汇总计数。

系统资源

Erlang VM 资源利用率:

指标	描述
总内存	分配给 Erlang VM 的总内存
进程内存	Erlang 进程使用的内存
二进制内存	用于二进制数据的内存（XML、JSON 负载）
原子内存	用于原子的内存
进程计数	活动 Erlang 进程的数量
运行队列	等待 CPU 时间的进程数量（0 为健康）

自动刷新

页面每 5 秒自动刷新。

用例

- 监控批处理写入器的健康和吞吐量
- 识别数据量大的设备
- 故障排除写入性能问题
- 验证所有设备的数据流
- 监控系统资源使用情况

PM 数据收集页面

控制存储在 InfluxDB 中的性能指标（PM）计数器。Nokia AirScale 基站报告超过 22,000 个独特的 PM 计数器，但通常只需要子集用于仪表板。

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

□□□□□□□□□□□□□□□□□□□□□□□□□□□□ PM □□□□□□□□

PM 过滤工作原理

批处理写入器根据“存储的 PM 数据”列表过滤传入的 PM 数据。未在此列表中的计数器在排队之前被丢弃，从而减少存储并提高查询性能。

界面布局

部分	描述
存储的 **PM** 数据（左）	当前正在收集并写入 InfluxDB 的计数器
可用计数器（右）	可添加的所有 22,000+ 计数器
□置键（底部）	正在跟踪的配置参数（来自 `config_keys.csv`）

存储的 PM 数据面板

此列表中的计数器在从设备接收时写入 InfluxDB。

过滤器：

- 搜索：按计数器 ID 或描述过滤
- 类别：按技术（LTE、5G-NR 等）过滤
- 来源：按默认（来自 CSV）或添加（用户添加）过滤

操作：

- 移除所选：停止收集所选计数器
- 全选：选择所有可见计数器
- 重置为默认：从 `pm_counters.csv` 恢复原始列表

可用计数器面板

浏览并从完整的 Nokia PM 参考中添加计数器。

过滤器：

- ��索：按 ID 或描述关键字查找计数器
- 类别：按技术类别过滤

操作：

- 添加所选：开始收集所选计数器
- 全选：选择所有可见计数器（限制为显示 200 个）

计数器类别

类别	代码前缀	描述
LTE	M8xxx	LTE L1/L2/L3 计数器
WCDMA	M5xxx	3G WCDMA 计数器
5G-NR	M55xxx	5G NR 计数器
5G-移动性	M51xxx	5G 移动性指标
5G-通用	M40xxx	5G 通用计数器

持久性

更改：

- 持久化到磁盘在 `priv/pm_filters.etf` 中
- 在应用程序重启时存活
- 立即生效（无需重启）

相关统计信息

InfluxDB 状态页面显示过滤统计信息：

- **PM** 过滤：丢弃的 PM 点计数（不在白名单中）
- 过滤率：被阻止的重复配置/警报点的百分比

有关计数器选择的详细指导，请参见 PM 数据收集指南。

固件管理页面

管理 Nokia AirScale 固件包以进行远程软件更新。

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

概述

固件管理页面提供基站软件包的集中库。将固件上传到 RAN 监控，并在 WebEM 的“远程服务器”选项中使用生成的 HTTP URL 进行软件更新。

关键特性

特性	描述
上传	存储 Nokia AirScale .zip 固件包
下载 **URL**	复制用于 WebEM 远程服务器配置的 HTTP URL
MD5 校验和	自动校验和计算以进行完整性验证
存储统计	监控库的磁盘使用情况

快速工作流程

1. 将固件包上传到 RAN 监控
2. 单击 复制 **URL** 以获取下载 URL
3. 在 WebEM 软件管理中，选择 远程服务器
4. 粘贴 URL 并单击 连接并刷新
5. 选择固件并开始更新

有关完整文档，包括配置选项和故障排除，请参见 固件管理指南。

数据管理页面

管理缓存数据、临时文件和持久存储。

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

□□ ETS □□□□□□□□ InfluxDB □□□□□□□□□□□□□□□

ETS 缓存（内存中）

在应用程序重启时清除的易失性内存缓存：

缓存	描述
Nokia □置缓存	缓存的设备配置数据
Nokia 警报缓存	缓存的活动警报记录
保留记录缓存	缓存的保留政策记录计数
InfluxDB 状态缓存	缓存的 InfluxDB 连接状态

清除所有 **ETS** 缓存：从内存缓存中删除所有条目。数据将在下一个请求时重新生成。

临时文件

在配置提取和处理期间创建的文件：

- 从设备配置下载的 TAR 提取
- 在处理期间创建的临时归档

清除临时文件：从 `/tmp` 目录中删除临时文件。

InfluxDB 数据（持久）

存储在 InfluxDB 中的每个设备的时间序列数据：

- 性能指标（PM 计数器）
- 配置快照
- 警报记录

每设备清除：单击“清除数据”旁边的设备以删除该设备的所有 InfluxDB 数据。

清除所有：删除所有设备的所有 InfluxDB 数据。请谨慎使用。

设备注册表

显示已注册设备的计数。设备配置在启动时从 `config/runtime.exs` 加载，并存储在 ETS 中。

用例

- 通过清除缓存释放内存
- 在故障排除后清理临时文件
- 在生产使用之前删除测试数据
- 清除退役设备的数据

Web UI 工作流

使用 Web UI 的常见操作工作流。

每日健康检查

目的：在班次开始时验证系统健康

步骤：

1. 打开主仪表板
2. 验证所有设备显示绿色状态
3. 检查警报计数和严重性
4. 审查任何红色/失败的设备
5. 根据需��调查问题
6. 记录采取的任何行动

时间： < 5 分钟

警报调查

目的：响应和解决警报

步骤：

1. 打开警报页面
2. 按严重性排序（关键优先）
3. 单击警报以获取完整详细信息
4. 导航到受影响的设备
5. 与最近的指标交叉引用
6. 确定所需的操作
7. 实施解决方案
8. 验证警报是否清除

有关详细的警报处理程序，请参见 警报管理指南。

置更新

目的：安全更新设备配置

步骤：

1. 下载当前配置（备份）
2. 使用适当的工具离线修改配置
3. 将新配置上传到设备
4. 记录返回的计划 ID
5. 使用计划 ID 验证配置
6. 如果验证成功，则激活配置
7. 验证更改是否生效
8. 监控设备稳定性 15-30 分钟
9. 在变更管理系统中记录更改

安全注意事项：

- 始终在激活之前进行验证
- 尽可能在维护窗口期间进行更改
- 准备好回滚计划
- 监控意外行为

添加新基站

目的：将新部署的基站添加到监控中

步骤：

1. 检查未配置的 eNodeBs 页面以查找设备
2. 记下代理 ID 和 IP 地址
3. 将设备添加到 `config/runtime.exs`
4. 重启 RAN 监控应用程序
5. 验证设备出现在基站页面
6. 确��注册成功（绿色状态）
7. 检查指标是否流入 InfluxDB
8. 如有需要，设置保留政策
9. 添加到 Grafana 仪表板

有关详细操作，请参见 常见操作指南。

故障排除设备连接

目的：诊断和修复设备连接问题

步骤：

1. 检查基站页面以获取设备状态
2. 如果为红色/失败，单击设备以获取详细信息
3. 查看会话信息和最后联系时间
4. 检查应用日志以获取错误消息
5. 验证网络连接（ping 设备）
6. 确认凭据是否正确
7. 检查设备是否在配置的端口上可访问
8. 如有需要，查看设备端日志
9. 根据需要重启连接或设备
10. 验证恢复情况

有关详细故障排除，请参见 故障排除指南。

相关文档

- 入门指南 - 初始设置和部署
- 常见操作指南 - 日常操作任务
- 警报管理指南 - 警报处理和升级
- 运行时□置指南 - 系统配置
- **Grafana** 集成指南 - 分析和仪表板
- 数据保留政策指南 - 数据生命周期管理
- 故障排除指南 - 常见问题���解决方案

RAN 监控操作指南

无线接入网络（**RAN**）监控与管理平台

□ Omnitouch □□□□□□

目录

1. □述
2. RAN 监控的功能
3. 系统架构
4. Web UI □述
5. 使用 Grafana 进行监控
6. 常见操作
7. 文档索引
8. 快速参考
9. 支持

概述

RAN 监控是一个用于 Nokia AirScale 基站在 3GPP LTE 和 5G 网络中的管理和监控平台。它提供对您的 RAN 设备健康状况、性能和配置的实时可见性。

关键特性

- 实时监控 - 持续收集性能指标和警报
- 自动化管理 - 与基站保持持久连接
- 历史分析 - 存储数据以进行趋势分析和容量规划
- **Web** 仪表板 - 通过内置 Web UI 提供实时操作可见性
- **Grafana** 集成 - 高级分析和自定义仪表板

□□□□

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

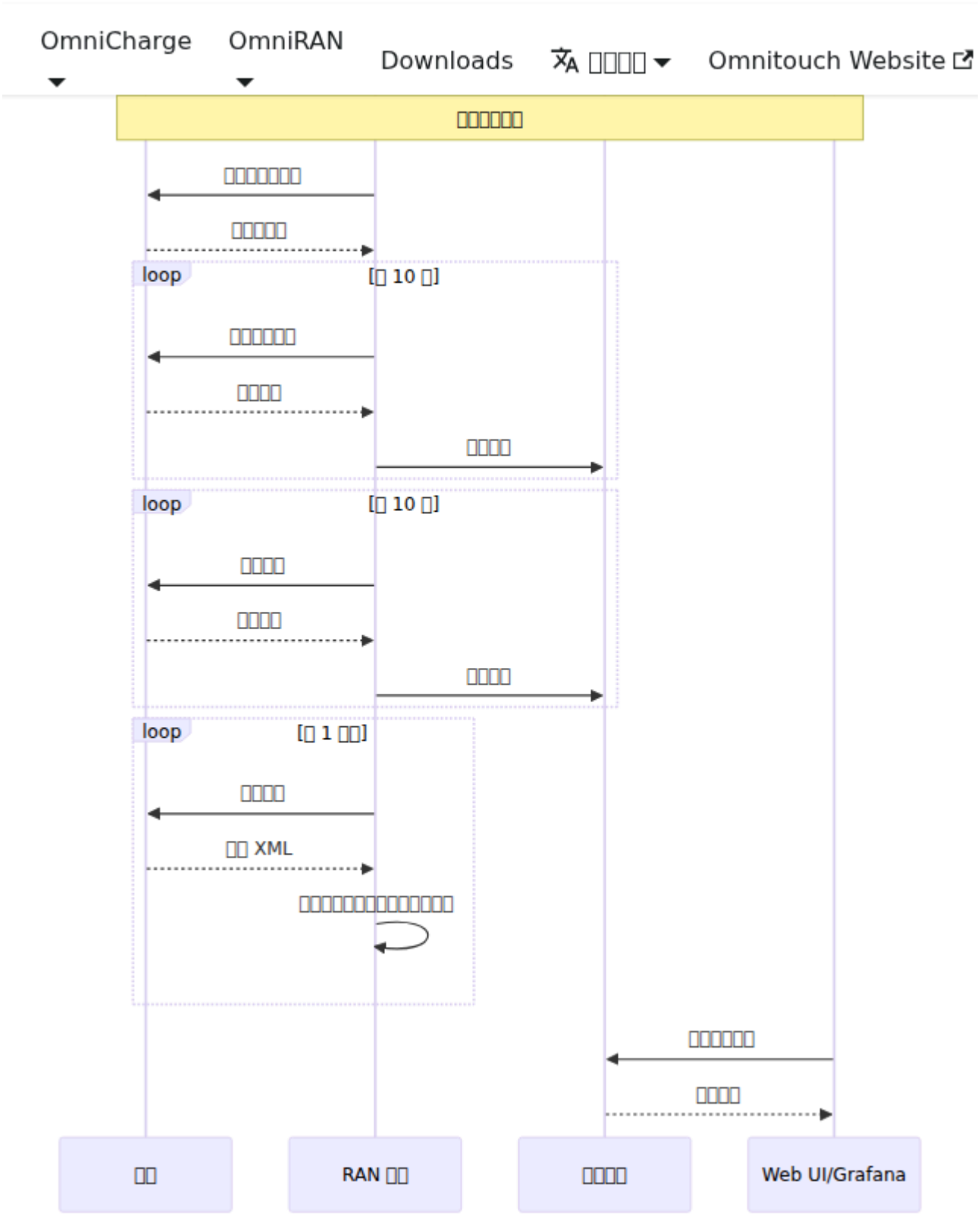
□□□□□□□□□□

*□□□□□□□□□□ LNMME □ S1 □□□□□□□□□□□□□□□□□□□ UE□□□ PRB □□□□□□□□□□□□□□□□□□□

RAN 监控的功能

RAN 监控在后台持续运行，以：

1. 注册和连接 - 与您的 Nokia 基站建立安全连接
2. 收集性能数据 - 每 10 秒收集一次 KPI（可配置）
3. 监控警报 - 跟踪故障及其严重程度
4. 跟踪□置 - 记录系统状态和参数变化
5. 存储历史数据 - 在时间序列数据库中保留指标
6. 提供可见性 - 通过 Web UI 和 Grafana 显示实时状态



数据流

收集的内容

性能指标：

- 小区可用性和正常运行时间
- 流量吞吐量（上行/下行）
- 资源利用率（PRB 使用）
- 呼叫设置成功率
- 切换性能
- 无线质量测量

警报：

- 故障严重性（关键、主要、次要、警告）
- 受影响的系统和组件
- 可能的原因和描述
- 时间戳和持续时间

□置：

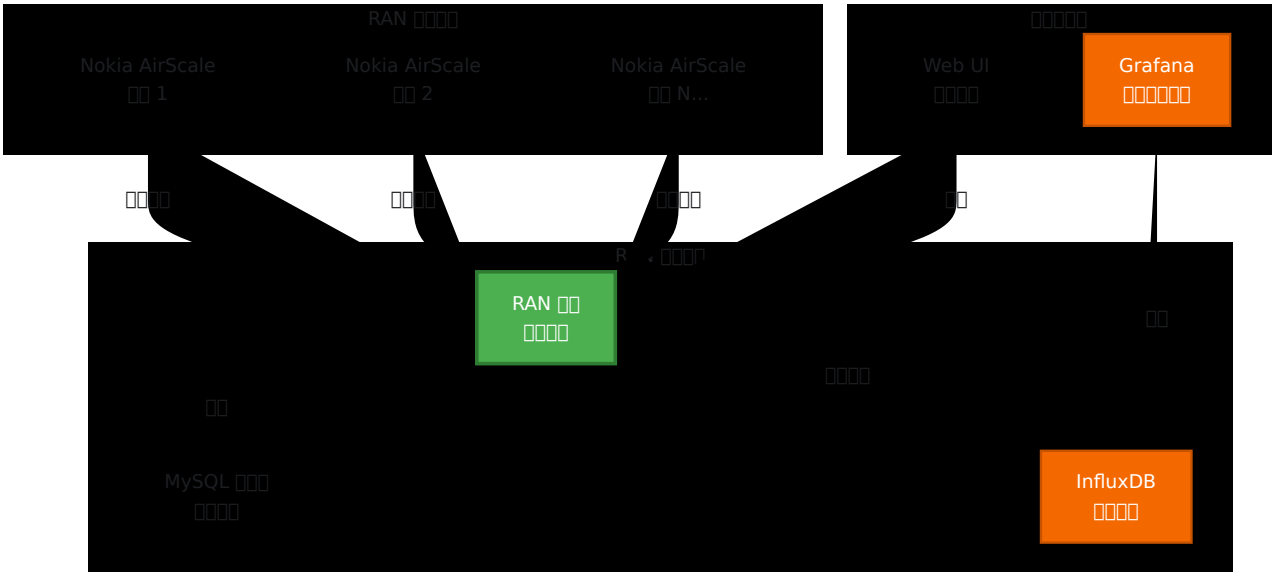
- 完整的 XML 配置快照（存储为版本文件）
- 自动变更检测和版本控制
- ��置历史和审计跟踪
- 每个设备保留最后 10 个版本

有关配置管理的详细信息，请参见 配置档案指南。

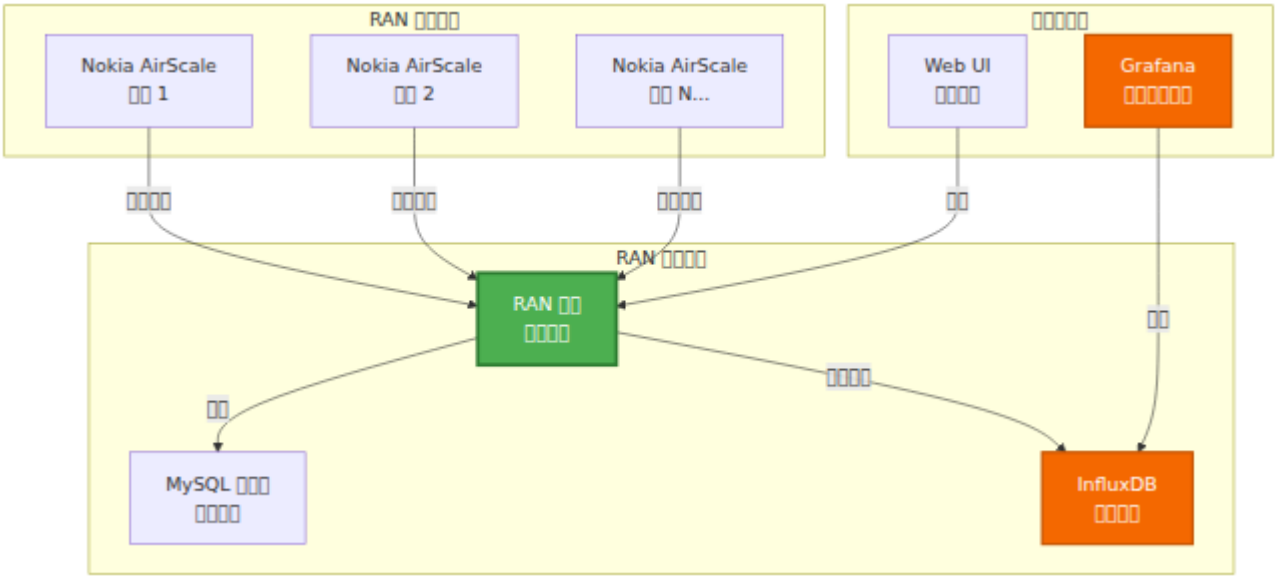
有关详细计数器定义，请参见 Nokia 计数器参考。

□□□□

□□□□□□□


RAN □□□□
Nokia AirScale
□□ 1
Nokia AirScale
□□ 2
Nokia AirScale
□□ N...
Web UI
□□□□
Grafana
□□□□□□
RAN □□
□□□□
InfluxDB
□□□□
MySQL □□□
□□□□

❓❓□□□


RAN □□□□
Nokia AirScale
□□ 1
Nokia AirScale
□□ 2
Nokia AirScale
□□ N...
□□□□□
Web UI
□□□□
Grafana
□□□□□□
RAN □□□□
RAN □□
□□□□
MySQL □□□
□□□□
InfluxDB
□□□□

□□□□□□□□□□□□□□□□ □□□□□□□□□

跟踪收集实体 (TCE)

RAN 监控包括一个集成的跟踪收集实体，用于捕获和分析 LTE/5G 协议消息。这使得详细故障排除、驱动测试和 RF 优化成为可能。

什么是 TCE？

跟踪收集实体接收来自 Nokia AirScale 基站的跟踪数据，包含：

S1-AP 消息 - eNodeB 和 EPC 之间的控制平面信令
RRC 消息 - 无线资源控制信令
NAS 消息 - 非接入层信令
用户平面数据 - PDCP 层吞吐量信息

用例

驱动测试：

捕获终端用户 RF 体验
分析切换性能
测量信号质量（RSRP、RSRQ、SINR）
识别覆盖盲区

故障排除：

调试呼叫设置失败
分析切换问题
调查掉话
审查移动事件

RF 优化：

PCI 规划验证
邻接关系优化
切换参数调整
覆盖和容量分析

有关完整的跟踪收集程序和 Wireshark 分析，请参见 TCE MDT 数据收集指南。

Web UI 概述

RAN 监控包括一个内置的 Web UI，用于实时操作监控和管理。

访问：`https://<ran-monitor-ip>:9443`

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

关键页面

主仪表板

实时系统□述，包含：

- 系统健康指标
- 设备状态摘要（关联/失败计数）
- 按严重性分类的活动警报计数
- 最近活动和事件

每 **5** 秒自动刷新以提供实时可见性。

基站页面

查看所有管理设备及其当前状态：

- 连接状态（绿色 = 关联，红色 = 失败）
- 注册状态和会话信息
- 最后联系时间戳
- 过滤、搜索和排序功能

点击任何设备以查看详细信息，包括注册详情、会话生命周期、最近指标和活动警���。

警报页面

监控网络中的所有故障：

按严重性分类的颜色编码（红色 = 关键，橙色 = 主要，黄色 = 次要，蓝色 = 警告，绿色 = 已清除）
警报详情、可能原因、受影响系统
时间戳和持续时间跟踪
按严重性排序和过滤功能

有关警报处理程序的详细信息，请参见 警报管理指南。

□置管理

安全管理基站配置：

1. 下载 当前配置（备份）
2. 上传 新配置 → 接收计划 ID
3. 验证 使用计划 ID 的配置
4. 激活 验证后的配置
5. 验证 更改已生效

始终在激活之前进行验证，以防止服务中断。

□置档案：所有配置更改都会自动跟踪和版本控制。查看历史配置，下载以前的版本，或通过配置档案页面比较更改。

有关详细程序，请参见 Web UI 指南 - 配置管理 和 配置档案指南。

未□置的 eNodeB

发现尝试连接但尚未配置的基站：

- 代理 ID（添加到配置时使用）
- 最后见时间戳
- 连接尝试次数
- 操作：刷新、删除、清除所有

用例：当部署新的基站时，它们会出现在这里。复制代理 ID 并将其添加到 `config/runtime.exs`。

应用日志

实时日志仪表板用于故障排除：

按日志级别过滤（紧急到调试）
在所有消息中搜索
暂停/恢复实时流
动态更改系统日志级别
按严重性颜色编码

有关故障排除程序的详细信息，请参见 故障排除指南。

数据保留政策

管理数据在 InfluxDB 中存储的时间：

查看全局保留政策和总记录计数
设置每个设备的保留期限
按测量类型查看记录计数（性能指标、配置、警报）
手动触发清理或清除设备的所有数据

有关完整的数据保留信息，请参见 数据保留政策指南。

InfluxDB 状态

监控时间序列数据库健康：

- 连接状态指示器
- 按类型的测量计数
- 存储信息
- 数据库版本和配置
- 每 5 分钟自动刷新

状���解释：

- 连接 + 计数增长 = 正常操作
- 连接 + 无数据 = 检查设备注册
- 未连接 = 检查 InfluxDB 连接性

完整的 Web UI 指南

有关全面的 Web UI 文档，包括所有功能、工作流程和最佳实践，请参见：
Web UI 指南 - 完整控制面板参考

使用 Grafana 进行监控

虽然 Web UI 提供实时可见性，但 Grafana 使深度历史分析和自定义仪表板成为可能。

为什么使用 Grafana？

Grafana 最适合：
几天/几周/几个月的历史趋势分析
根据您的需求量身定制的 KPI 仪表板
长期容量规划
模式识别和异常检测

- 高管报告和 SLA 跟踪
- 具有通知通道的高级警报

Web UI 最适合:

- 立即检查设备状态
- 实时警报监控
- 配置管理
- 会话故障排除
- 系统管理任务

□□ Grafana □□□□□□□□□□□□□□□□□□□□□□□□

仪表板类型

高管摘要仪表板:

- 网络范围的健康□述
- 按严重性分类的总警报计数
- 所有站点的平均小区可用性
- 聚合吞吐量和容量指标
- 设备状态网格

NOC 操作仪表板：

- 实时活动问题表
- 资源利用率仪表
- 流量□述（过去 24 小时）
- 警报趋势图
- 设备状态快速视图

工程深度分析仪表板：

- 流量模式分析
- 小区质量指标（SINR、RSRP 分布）
- 无线性能（RLC 重传、RRC 设置成功）
- 配置审计跟踪
- 相关性分析

Nokia AirScale 性能仪表板：

- PRB 利用率（下行/上行）
- 吞吐量趋势（PDCP 层）
- 活跃 UE 计数
- 小区可用性计算
- 每个小区的资源细分
- RSSI 测量
- RRC 连接设置成功
- 按天线的 VSWR
- 功耗

有关完整的仪表板示例、查询模式和计数器定义，请参见：

Grafana 集成指南 - 完整的分析和仪表板指南

Nokia 计数器参考 - 性能计数器定义

常见操作

日常操作

每日健康检查（5-10 分钟）：

1. 打开 Web UI 仪表板
2. 验证所有设备显示绿色状态
3. 检查警报计数和严重性
4. 审查任何失败的设备
5. 根据需要调查问题

有关详细程序，请参见 Web UI 指南 - 工作流程。

警报调查：

1. 打开警报页面，按严重性排序
2. 点击警报以获取完整详情
3. 导航到受影响的设备
4. 与指标交叉参考
5. 确定所需操作并解决

有关警报处理程序的详细信息，请参见 警报管理指南。

设备管理

添加新基站：

1. 验证设备的网络连接
2. 检查未配置的 eNodeB 页面以查找设备
3. 将设备添加到 config/runtime.exs
4. 重启 RAN 监控
5. 验证注册成功（绿色状态）
6. 确认指标流向 InfluxDB

移除基站：

1. 决定是保留还是删除历史数据
2. 在 `config/runtime.exs` 中注释掉或删除设备
3. 可选地通过数据保留页面清除数据
4. 重启 RAN 监控
5. 更新 Grafana 仪表板

更新设备凭证：

1. 记录当前设备状态
2. 在 `config/runtime.exs` 中更新凭证
3. 重启 RAN 监控
4. 验证重新连接成功

有关完整的操作程序，请参见：

常见操作指南 - 日常管理任务

□置管理

安全□置更新工作流程：

1. 下载 当前配置（备份） - 或从配置档案中检索
2. 修改 离线配置
3. 上传 到设备 → 获取计划 ID
4. 验证 使用计划 ID → 验证无错误
5. 激活 如果验证成功
6. 验证 更改已生效
7. 监控 设备稳定性 15-30 分钟
8. 确认 新版本出现在配置档案中（1 小时内）

重要：始终在激活之前进行验证。尽可能在维护窗口期间安排更改。

置回滚：如果出现问题，从配置档案中下载以前的版本并使用相同的工作流程上传。

有关基站配置的详细信息，请参见 AirScale 配置指南。

有关配置历史和版本控制，请参见 配置档案指南。

文档索引

RAN 监控文档按受众和用例组织：

面向运营团队（**NOC**、管理员）

文档	目的	使用时机
Web UI 指南	完整控制面板参考	日常操作、监控设备
常见操作指南	日常管理任务	添加设备、管理配置、备份
□置档案指南	配置版本控制和历史	跟踪配置更改、回滚、审计
固件管理指南	软件包存储库	上传固件、基站软件更新
警报管理指南	警报处理和升级	调查故障、响应警报
故障排除指南	问题解决程序	当出现问题时，错��诊断
数据保留政策指南	数据生命周期管理	管理存储、设置保留期限

面向工程和分析

文档	目的	使用时机
Grafana 集成指南	仪表板、查询和警报	构建仪表板、设置警报
Nokia 计数器参考	性能计数器定义	理解指标、创建查询
AirScale □置指南	基站设置和配置	配置设备、理解参数
TCE MDT 数据收集指南	MDT 跟踪收集和 Wireshark 分析	收集驱动测试数据、覆盖优化
API 端点参考	REST API 文档	集成、自动化、脚本

面向□置和部署

文档	目的	使用时机
运行时□置指南	完整配置参考	初始设置、修改设置

快速入门

新接触 **RAN** 监控?

1. 从 Web UI 指南 开始学习界面
2. 查看 常见操作指南 以了解常规任务
3. 学习 警报管理指南 以处理警报
4. 保持 故障排除指南 以备问题发生时使用

设置监控?

1. 查看 Grafana 集成指南 以了解仪表板
2. 参考 Nokia 计数器参考 以了解指标
3. 查看 数据保留政策指南 以了解存储管理

快速参考

访问点

服务	**URL**	目的
Web UI 仪表板	`https://<server>:9443`	实时监控和管理
Grafana	`http://<server>:3000`	分析仪表板和警报
InfluxDB	`http://<server>:8086`	指标数据库（通常仅限内部访问）

重要路径

路径	目的
`config/runtime.exs`	主要配置文件（设备、数据库、设置）
`priv/cert/`	用于 HTTPS Web UI 的 SSL 证书
`priv/external/nokia/`	管理员身份验证密钥
`priv/airscale_configs/`	配置档案（版本化的 XML 文件）

关键概念

会话管理：

- RAN 监控与基站建立会话
- 会话具有过期时间并需要保持活动
- 重新注册会自动发生（默认：每 30 秒）
- 会话状态存储在 MySQL 数据库中

数据流：

- 每 10 秒收集一次指标（可配置）
- 每 10 秒通过轮询 + 实时 Webhook 收集警报
- 每 1 小时收集一次配��快照（在更改时保存为版本文件）
- 性能指标和警报写入 InfluxDB 以进行历史存储

数据保留：

- 全局默认：720 小时（30 天）
- 每个设备的覆盖可用
- 自动清理每小时运行
- 通过 Web UI 可手动清理

有关配置详细信息，请参见 运行时配置指南。

常见工作流程

每日健康检查：

1. 打开 Web UI → 仪表板
2. 检查设备状态（全部为绿色？）
3. 审查警报计数
4. 调查任何问题

响应关键警报：

1. Web UI → 警报 → 按严重性排序
2. 点击警报以获取详细信息
3. 导航到设备
4. 审查最近的指标和配置更改
5. 实施解决方案
6. 验证警报已清除

添加新设备：

1. 验证网络连接
2. 编辑 config/runtime.exs
3. 将设备添加到 AirScale 列表
4. 重启 RAN 监控
5. 验证注册（绿色状态）

支持

故障排除资源

资源	用于
故障排除指南	常见问题和解决方案
应用日志页面	实时系统日志和错误
设备详细视图	会话状态、注册问题
InfluxDB 状态页面	数据收集验证

快速诊断步骤

设备未连接：

1. 检查基站页面 → 设备状态
2. 验证网络连接：`ping <device-ip>`
3. 检查 `config/runtime.exs` 中的凭证
4. 审查应用日志以查找错误

Grafana 中没有指标：

1. 检查设备是否已关联（绿色状态）
2. 验证 InfluxDB 状态页面显示计数增长
3. 测试 InfluxDB 连接性
4. 检查 Grafana 数据源配置

Web UI 无法加载：

1. 验证端口 9443 是否可访问
2. 检查防火墙是否允许 HTTPS 流量
3. 验证 SSL 证书是否存在
4. 审查应用日志以查找 Web UI 启动错误

有关完整的故障排除程序，请参见 故障排除指南。

获取帮助

在联系支持之前：

收集以下信息：

问题描述及其开始时间
应用日志中的错误消息
受影响的设备（名称/IP）
最近的配置更改
RAN 监控版本和操作系统

联系：

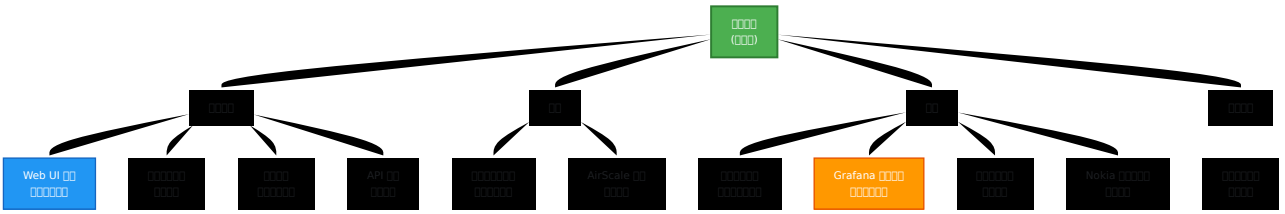
有关 RAN 监控的帮助：

Omnitouch 网络服务支持
包括收集的诊断信息
提供配置文件（清除密码）
包括相关日志摘录

自助服务：

1. 搜索 故障排除指南
2. 检查应用日志以获取特定错误
3. 审查最近的配置更改
4. 测试连接性和基本功能
5. 查阅相关文档指南

□□□□


□□□□
(□□□)
□□□□
□□
□□
□□□□
Web UI □□
□□□□□□□
□□□□□□
□□□□
□□□□□□□
□□□□
□□□□□□□
API □□
□□□□
□□□□□□□□
□□□□□□□
AirScale □□
□□□□
□□□□□□□
□□□□□□□□
Grafana □□□□
□□□□□□□
□□□□□□□
□□□□
Nokia □□□□□□
□□□□
□□□□□□□
□□□□