TARANA

BROADBAND REINVENTED

Webhooks

App Note

What is a Webhook?

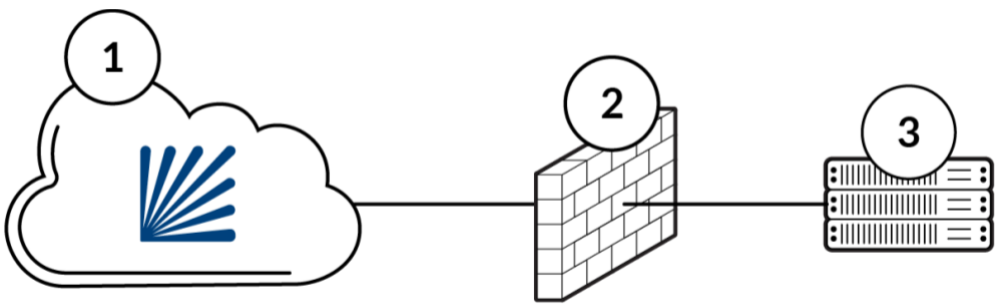
Simply put, webhooks are messages that get sent automatically when something specific happens on a website or web application. Instead of checking all the time for updates, you get notified instantly when something you care about occurs. This helps you customize your experience and integrate different apps together more easily.

A webhook is an event-based notification that allows an application to notify another application, process, or person when a monitored event that generates an alarm occurs. An alarm-raising system event generates an alarm that invokes the webhook, which then sends a message to an application. Webhooks are one-way communications, unlike API calls, which are two-way and require an application or process to request information and wait for service to respond. Webhooks send an HTTP POST request to a specified URL, containing information about the alert that was triggered. Webhooks allow you to stay informed and take action promptly when needed.

Webhooks provide a way for network administrators to receive notifications for alerts they have set up, allowing them to receive real-time updates when those alerts are triggered. Tarana Cloud Suite (TCS) can send alert email messages when events occur. Events that you can configure for webhooks:

- Device connected/disconnected
- Remote node Ethernet port down/up
- Critical alarm raised/cleared
- Base node unreachable
- Radio carrier down/up (status of SAS grant for CBRS)
- Critical alarm raised/cleared
- New device installed
- New base node onboarded

Webhooks respond to the events and automatically send push messages directly to webhook receivers, as illustrated here. A TCS instance running in the cloud creates a webhook message and sends it through the Tarana cloud to the local router. You must configure the router's ACL to allow webhook messages from TCS so it can reach the local message server.



Reference	Description
1	TCS instance running in the cloud. This is the source of the webhook message.
2	Local router firewall or access control list (ACL). The ACL must allow incoming webhook messages from TCS.
3	Local message server or application.

You can configure and test webhooks directly in TCS. If you don't have any webhook receivers configured on your network, you can use public webhook receivers, such as https://webhook.site, to configure and test your webhooks.

The domain *.taranawireless.com needs to be whitelisted on your firewall to allow webhooks.

TCS supports any HTTPS endpoint as a webhook receiver.

To create, edit, or test webhooks, select **Admin > Webhooks** from the navigation pane.

Configuring Webhooks with TCS

Within TCS, the operator admin, TCS admin, or retailer admin can configure webhooks. This includes the ability to create multiple webhooks in the notification configuration section, each requiring a unique name within the operator's domain. Additionally, multiple webhooks can be set up when configuring alerts for specific events within the alert configuration section of TCS. TCS can deliver webhook messages to servers using IPv4 or IPv6.

When a webhook is created, the system will generate a unique and immutable secret (webhookKey), which is specific to each webhook and cannot be updated.

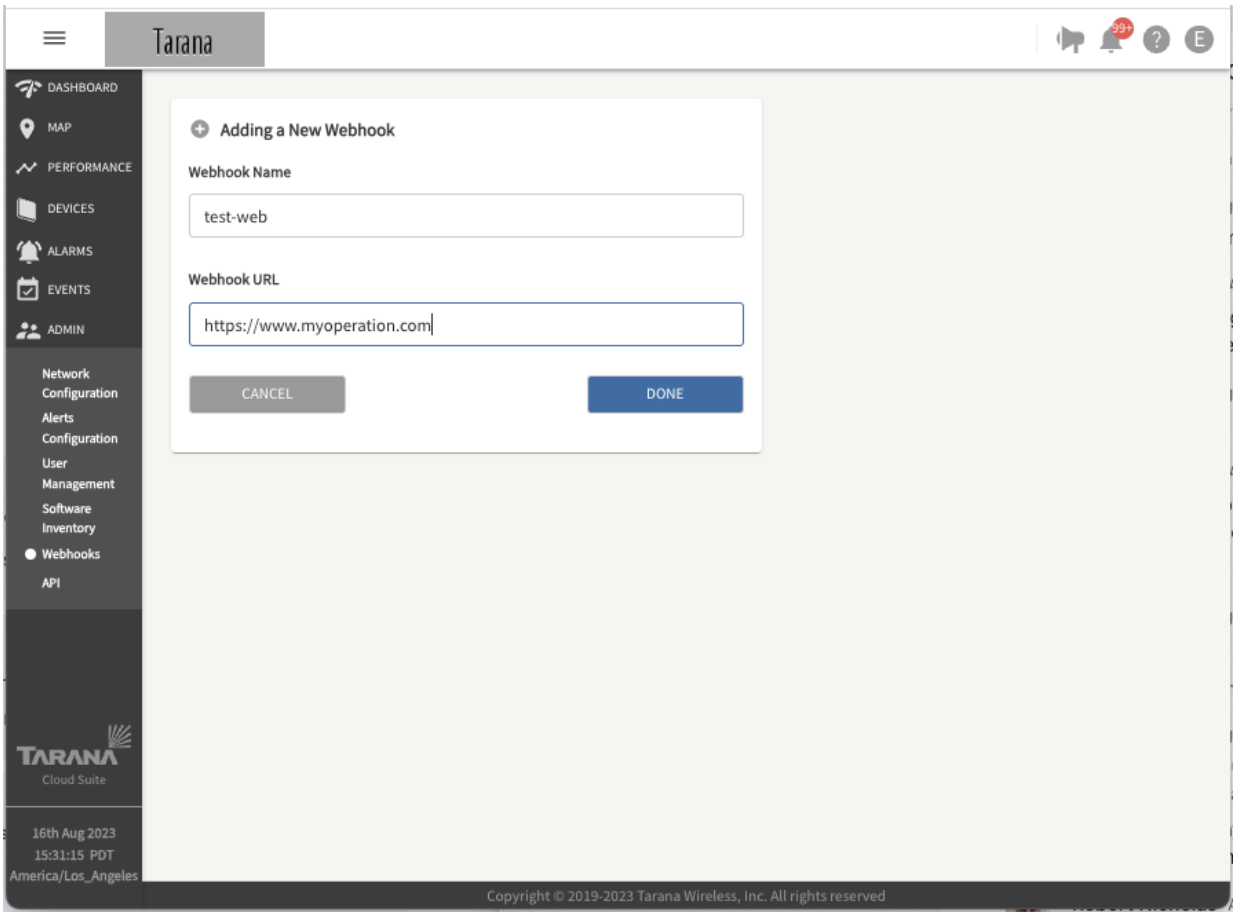
Regarding security, HMAC (Hash-based Message Authentication Code) is utilized. TCS adds a special header (x-tarana-signature) to the POST request based on keys known only to TCS and the customer. The customer application can then use these keys to decrypt the headers, verifying the authenticity and integrity of the message.

Note: Adding a webhook in TCS simply provides a path for configured alerts. Once the alert is created, the user has the option to send the alert as a webhook, or to a specified email(s), or both.

Add a Webhook

To create a new webhook, follow these steps:

1. Select **Admin > Webhooks** from the navigation pane.
2. Select **Add New Webhook**.
3. Enter these values:
   - Webhook Name: A unique and descriptive name for your webhook. This name appears in the list of webhooks that you add to TCS.
   - Webhook URL: URL of the receiving interface.
4. Select **Done**.



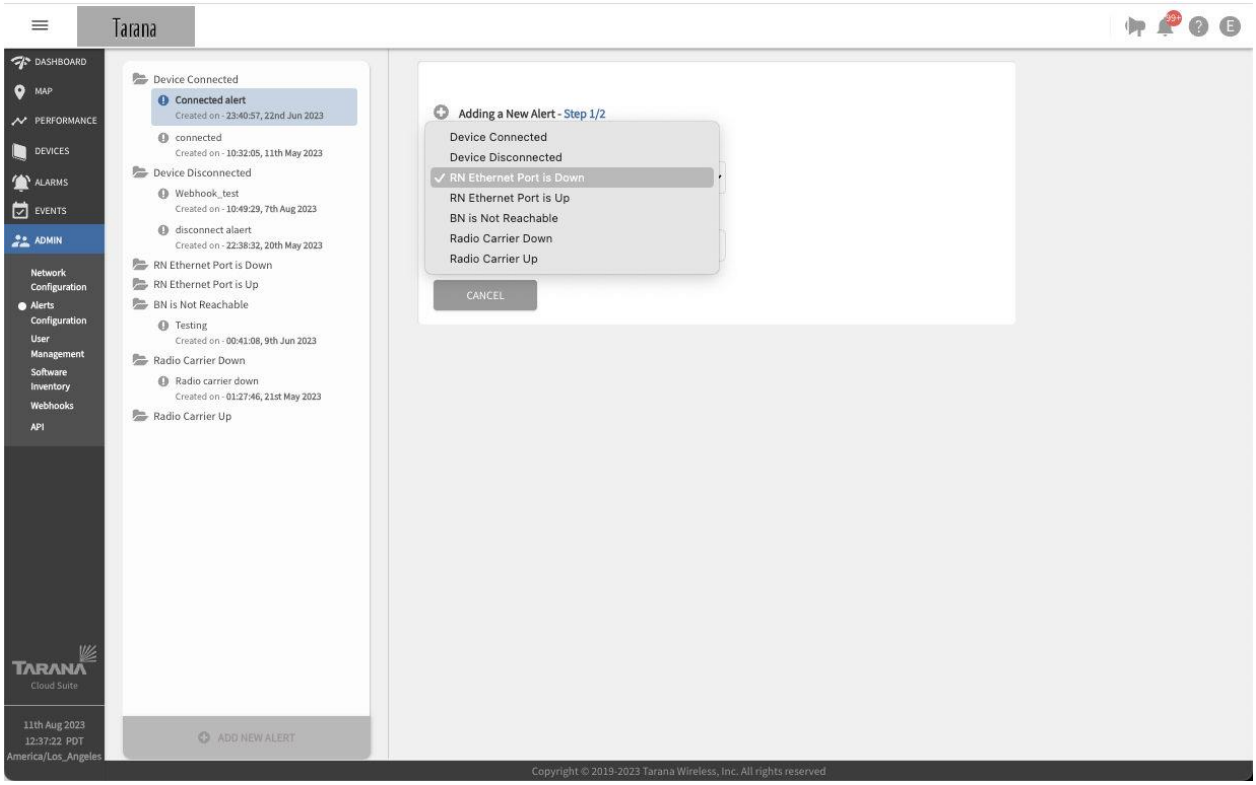
Note: When you create a webhook, TCS automatically generates a secure secret, which you can view when you test the webhook.

Alerts Configuration

To configure webhook email alerts, select **Admin - Alerts Configuration** from the navigation pane. You see a list of folders for the types of alarms you can create, with any alerts created for that type.

Select **Add New Alert**.

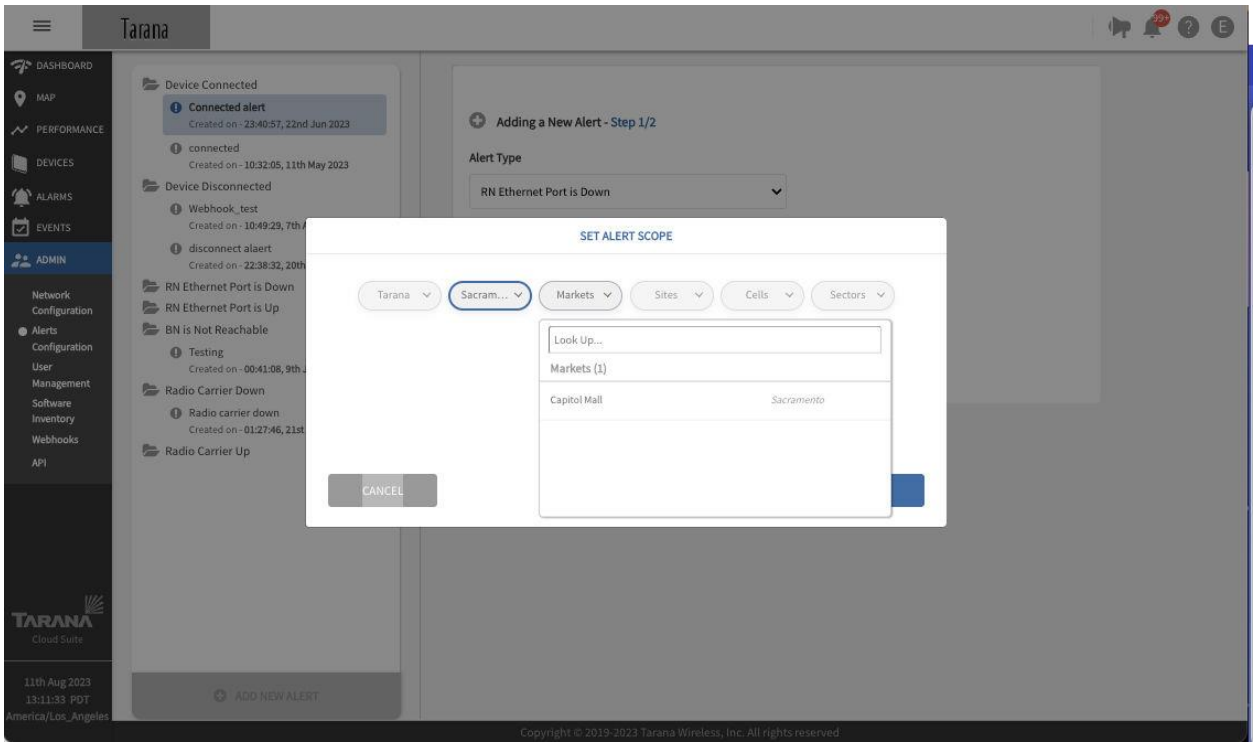
Select the alert type from the drop down.

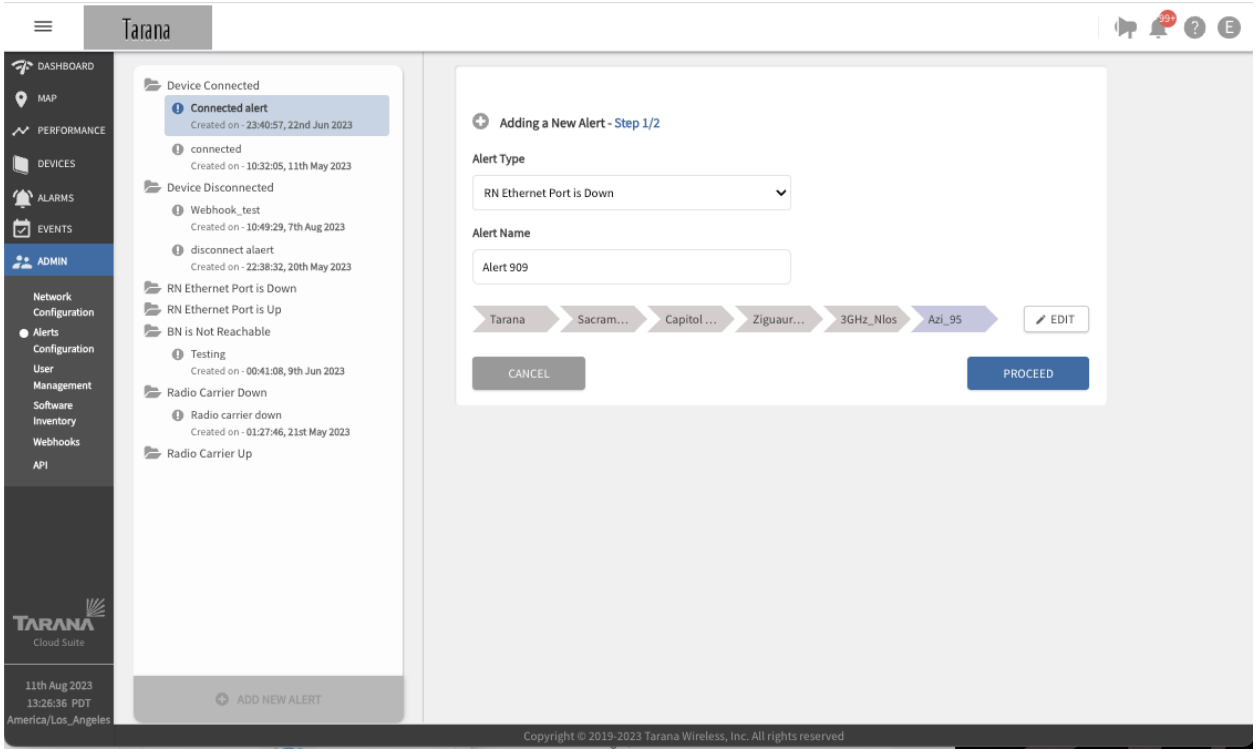


Enter a name for the alert.

For Device Connected, Device Disconnected, Radio Carrier Down, and Radio Carrier Up, use the radio buttons to select the **Device Type**.

Select **Click to Set Alert Scope**, then define the alarm's scope by selecting the applicable Region, Market, Site, Cell, and Sector. Select **Apply**. The screen shows the scope you've selected. Click **Edit** if you want to change it.

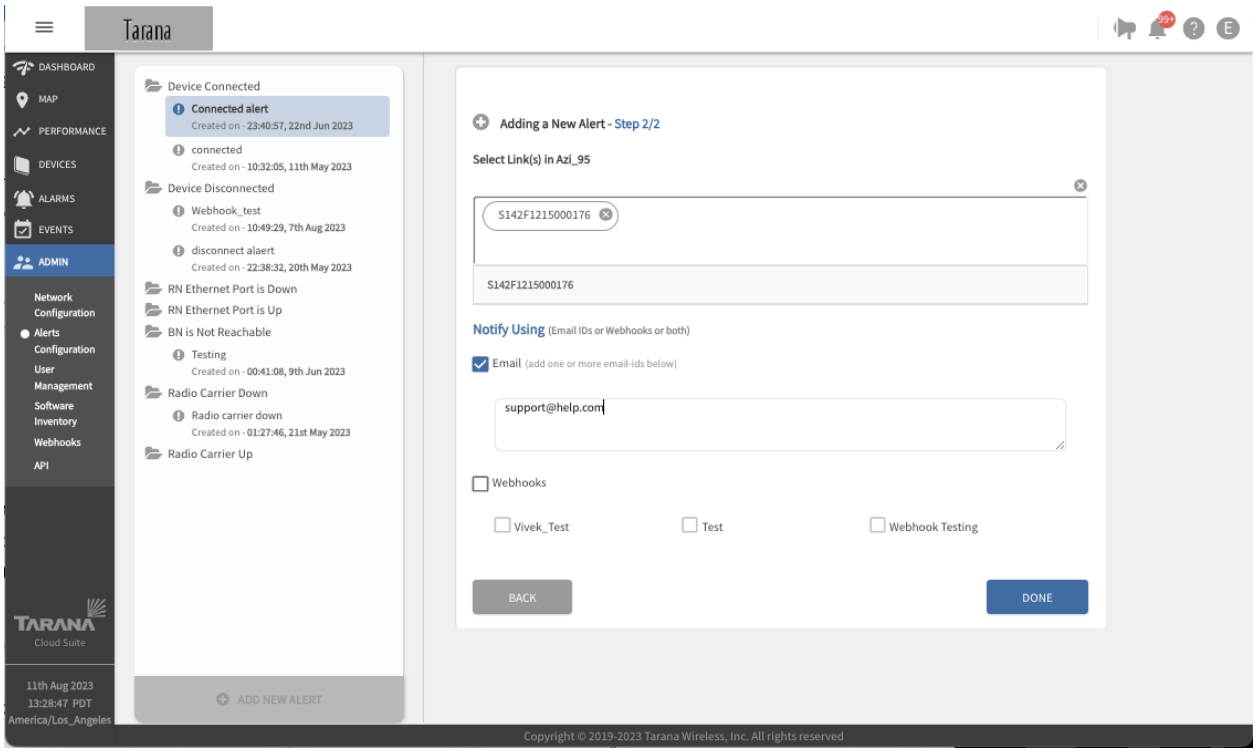




Select **Proceed**.

Select a link. Enter the email addresses that you want to receive this alert, separated by commas.

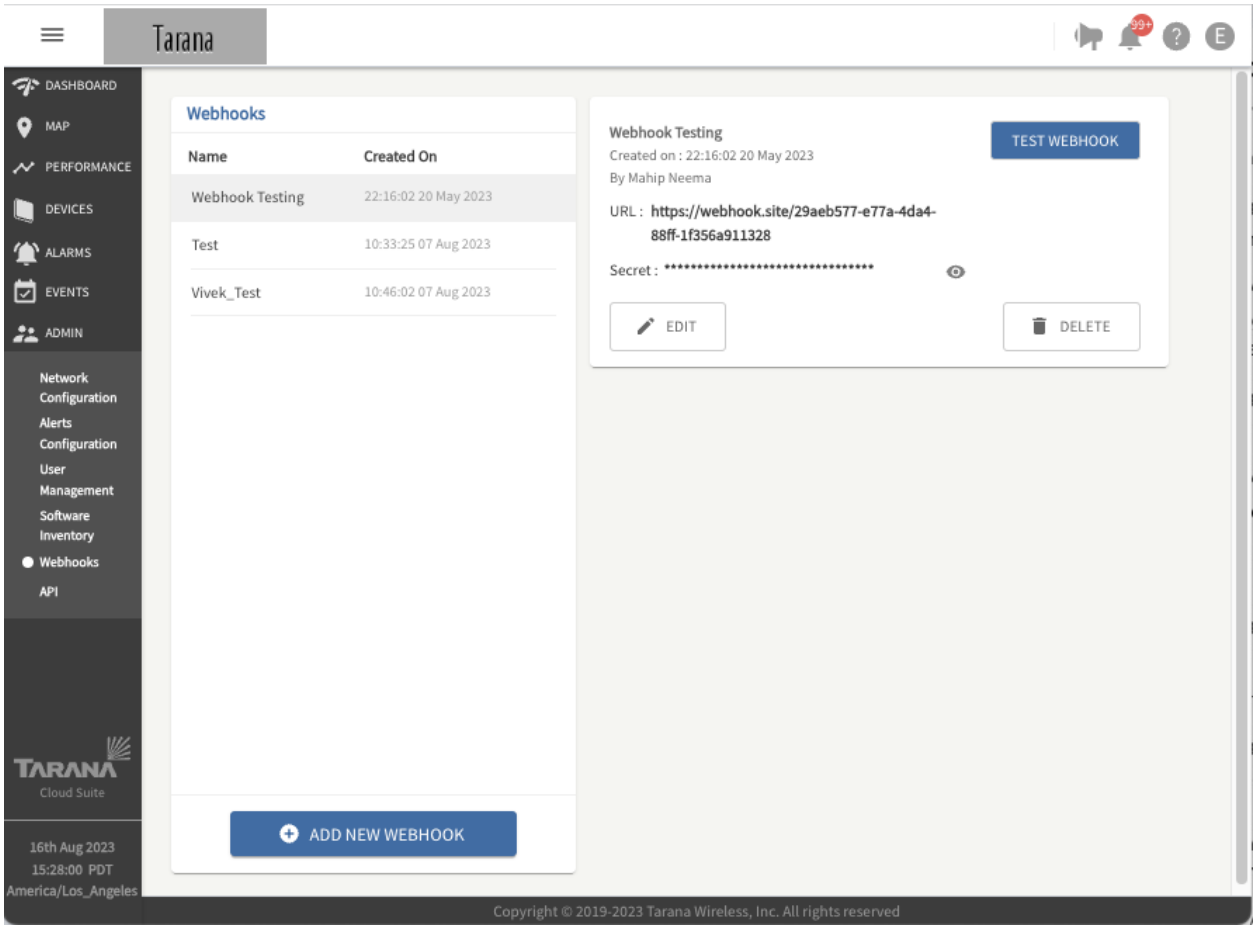
Check the boxes for any webhooks you want to receive this alert. Select **Done**.



Test a Webhook

To test a webhook, follow these steps:

1. Select **Admin > Webhooks** from the navigation pane.
2. Choose the webhook that you want to test from the list of available webhooks.
3. Select **Test Webhook**.
4. TCS displays a status message. Make sure that the expected result is correct. If it isn't, verify that the webhook URL and secret are correct in the webhook.



Webhook Output Templates

Various types of webhook notifications are generated depending on the mapped alerts.

Device Connected

```
{
  "alertStatus": "RAISED",
  "significantData": {
    "Device Hostname": <text string>,
    "Device Serial Number": <15 character alphanumeric string>,
    "Device Type": <"BN/RN">,
    "Device Software": <27 character alphanumeric string>,
    "Operator": <text string>,
    "Region": <text string>,
    "Market": <text string>,
    "Site": <text string>,
    "Cell": <text string>,
    "Sector": <text string>,
    "Device Uptime": <number of seconds>,
    "Device First Seen": <date_string>,
    "Device Boot Reason": <"warm boot/cold boot">,
    "Last Disconnect Reason": <text string>,
    "Device Reported Reason": <text string>,
    "Reboot Reason": <text string>,
    "Device Uptime Since": <time_string>
  },
  "alertName": <text string>,
  "alertId": <UUID>,
  "alertDescription": <text string>,
  "alertCreatedAt": <date_string>"
}
```

Device Disconnected

```
{
  "alertStatus": "RAISED",
  "significantData": {
    "Device Hostname": <text string>,
    "Device Serial Number": <15 character alphanumeric string>,
    "Device Type": <"BN/RN">,
    "Device Software": <27 character alphanumeric string>,
    "Operator": <text string>,
    "Region": <text string>,
    "Market": <text string>,
    "Site": <text string>,
    "Cell": <text string>,
    "Sector": <text string>,
    "Device Uptime": <number of seconds>,
    "Device First Seen": <date_string>,
    "Device Boot Reason": <"warm boot/cold boot">,
    "Disconnect Reason": <text string>,
    "Device Reported Reason": <text string>,
    "Device Uptime Since": <time_string>
  },
  "alertName": <text string>,
  "alertId": <UUID>,
  "alertDescription": <text string>,
  "alertCreatedAt": <date_string>"
}
```

Critical Alarm Raised

```
{
  "alertStatus": "RAISED",
  "significantData": {
    "Alarm Name": <text string>,
    "Alarm State": "Alarm Raised",
    "Alarm Description": <text string>,
    "Alarm Created At": <epoch timestamp in nanoseconds>,
    "Device Hostname": <text string>,
    "Device Serial Number": <15 character alphanumeric string>,
    "Device Type": <"BN/RN">,
    "Device Software": <27 character alphanumeric string>,
    "Operator": <text string>,
    "Region": <text string>,
    "Market": <text string>,
    "Site": <text string>,
    "Cell": <text string>,
    "Sector": <text string>,
    "Device Uptime": <number of seconds>,
    "Device Boot Reason": <"warm boot/cold boot">,
    "Device Uptime Since": <time_string>
  },
  "alertName": <text string>,
  "alertId": <UUID>,
  "alertDescription": <text string>,
  "alertCreatedAt": <date_string>"
}
```

Critical Alarm Cleared

```
{
  "alertStatus": "RAISED",
  "significantData": {
    "Alarm Name": <text string>,
    "Alarm State": "Alarm Cleared",
    "Alarm Description": <text string>,
    "Alarm Created At": <epoch timestamp in nanoseconds>,
    "Device Hostname": <text string>,
    "Device Serial Number": <15 character alphanumeric string>,
    "Device Type": <"BN/RN">,
    "Device Software": <27 character alphanumeric string>,
    "Operator": <text string>,
    "Region": <text string>,
    "Market": <text string>,
    "Site": <text string>,
    "Cell": <text string>,
    "Sector": <text string>,
    "Device Uptime": <number of seconds>,
    "Device Boot Reason": <"warm boot/cold boot">,
    "Device Uptime Since": <time_string>,
  },
  "alertName": <text string>,
  "alertId": <UUID>,
  "alertDescription": <text string>,
  "alertCreatedAt": <date_string>"
}
```

RN Ethernet Port is Down

```
{
  "alertStatus": "RAISED",
  "significantData": {
    "Alarm Type": "connections/connection/system/alarms",
    "Alarm Name": <text string>,
    "Alarm Description": <text string>,
    "Alarm Created At": <epoch timestamp in nanoseconds>,
    "Alarm Severity": "<integer>",
    "Device Hostname": <text string>,
    "Device Serial Number": <15 character alphanumeric string>,
    "Device Type": <"RN">,
    "Device Software": <27 character alphanumeric string>,
    "Operator": <text string>,
    "Region": <text string>,
    "Market": <text string>,
    "Site": <text string>,
    "Cell": <text string>,
    "Sector": <text string>,
    "Device Uptime": <number of seconds>,
    "Device Boot Reason": <"warm boot/cold boot">,
    "Device Uptime Since": <time_string>
  },
  "alertName": <text string>,
  "alertId": <UUID>,
  "alertDescription": <text string>,
  "alertCreatedAt": <date_string>"
}
```

RN Ethernet Port is Up

```
{
  "alertStatus": "RAISED",
  "significantData": {
    "Alarm Type": “connections/connection/system/alarms”,
    "Alarm Name": <text string>,
    "Alarm Description": <text string>,
    "Alarm Created At": <epoch timestamp in nanoseconds>,
    "Alarm Severity": "<integer>",
    "Device Hostname": <text string>,
    "Device Serial Number": <15 character alphanumeric string>,
    "Device Type": <”RN”>,
    "Device Software": <27 character alphanumeric string>,
    "Operator": <text string>,
    "Region": <text string>,
    "Market": <text string>,
    "Site": <text string>,
    "Cell": <text string>,
    "Sector": <text string>,
    "Device Uptime": <number of seconds>,
    "Device Boot Reason": <"warm boot/cold boot">,
    "Device Uptime Since": <time_string>
  },
  "alertName": <text string>,
  "alertId": <UUID>,
  "alertDescription": <text string>,
  "alertCreatedAt": <date_string>"
}
```

Radio Carrier Down

Radio carrier status alerts indicate the status of the SAS grant for CBRS devices, rather than general transmit status of the radio. For example, an alert indicating that the radio carrier is down occurs when the grant is suspended or terminated. Similarly, an alert indicating that the radio carrier is up occurs when the grant is authorized or when a suspension is lifted.

Note: On devices running 0.989 or earlier versions, both carriers must have active grants at both the base node and remote node to pass data traffic. Devices running 0.990 or later can pass data traffic on any carrier with an active grant at the base node and remote node.

```
{
  "alertStatus": "RAISED",
  "significantData": {
    "Device Hostname": <text string>,
    "Device Serial Number": <15 character alphanumeric string>,
    "Device Type": <"BN/RN">,
    "Device Software": <27 character alphanumeric string>,
    "Operator": <text string>,
    "Region": <text string>,
    "Market": <text string>,
    "Site": <text string>,
    "Cell": <text string>,
    "Sector": <text string>,
    "Device Boot Reason": <"warm boot/cold boot">,
    "Carrier Disabled": ”<0/1>”,
    "Lower Frequency [Mhz]": <integer>,
    "Upper Frequency [Mhz]": <integer>
  },
  "alertName": <text string>,
  "alertId": <UUID>,
  "alertDescription": <text string>,
  "alertCreatedAt": <date_string>"
}
```

Radio Carrier Up

```
{
  "alertStatus": "RAISED",
  "significantData": {
    "Device Hostname": <text string>,
    "Device Serial Number": <15 character alphanumeric string>,
    "Device Type": <"BN/RN">,
    "Device Software": <27 character alphanumeric string>,
    "Operator": <text string>,
    "Region": <text string>,
    "Market": <text string>,
    "Site": <text string>,
    "Cell": <text string>,
    "Sector": <text string>,
    "Device Boot Reason": <"warm boot/cold boot">,
    "Carrier Enabled": ”<0/1>”,
    "Lower Frequency [Mhz]": <integer>,
    "Upper Frequency [Mhz]": <integer>
  },
  "alertName": <text string>,
  "alertId": <UUID>,
  "alertDescription": <text string>,
  "alertCreatedAt": <date_string>"
}
```

New Device Installed

```
{
  "alertStatus": "RAISED",
  "significantData": {
    "Device Hostname": <text string>,
    "Device Serial Number": <15 character alphanumeric string>,
    "Device Type": <"RN">,
    "Device Software": <27 character alphanumeric string>,
    "Operator": <text string>,
    "Region": <text string>,
    "Market": <text string>,
    "Site": <text string>,
    "Cell": <text string>,
    "Sector": <text string>,
    "Device Uptime": <number of seconds>,
    "Device First Seen": <"Date">,
    "Device Boot Reason": <"warm boot/cold boot">,
    "Device Uptime Since": <time_string>
  },
  "alertName": <text string>,
  "alertId": <UUID>,
  "alertDescription": <text string>,
  "alertCreatedAt": <date_string>"
}
```

New BN Onboarded

```
{
  "alertStatus": "RAISED",
  "significantData": {
    "Device Serial Number": <15 character alphanumeric string>,
    "Device Type": <"BN">,
    "Operator": <text string>,
  },
  "alertName": <text string>,
  "alertId": <UUID>,
  "alertDescription": <text string>,
  "alertCreatedAt": <date_string>"
}
```

About Tarana

Tarana Wireless, Inc. is the performance leader in next-generation fixed wireless access network solutions, powered by a number of industry-first and well-proven breakthroughs in perfect, multidimensional optimization of radio signals. Its Gigabit 1 fixed access system overcomes previously insurmountable network economics challenges for service providers in both mainstream broadband and underserved markets, using free unlicensed spectrum. The company is headquartered in Milpitas, California, with additional research and development in Pune, India. For more information, visit taranawireless.com.