Logbot IOT Platform Manual

User Guideline

Chapter 1

Introduction

About this document

This document consolidates all proposed revisions to the Logbot IIoT Platform Manual. Its goal is to improve clarity, accuracy, and overall polish. The guidance is organized into four sections: (1) general recommendations, (2) a revised document structure, (3) an updated glossary, and (4) page-by-page change requests.

About Logbot

Logbot is a modern Industrial Internet of Things (IIoT) platform that streamlines and digitizes operations across manufacturing, energy, utilities, logistics, and more. It provides a centralized, web-based environment to connect, manage, and monitor infrastructure—plants, gateways, and devices—in real time. Built for scale and flexibility, Logbot simplifies complex system management with a modular interface that supports secure device communication, automated workflows, event tracking, and license control. It natively supports key industrial protocols (Modbus, OPC UA, S7), ensuring compatibility with both legacy and modern systems. Through an intuitive dashboard, users can configure devices remotely, track performance, respond to alarms, and run scheduled or on-demand commands. Whether deployed at a single plant or across multiple sites, Logbot enables smarter decisions, reduces manual effort, and increases operational transparency—serving as the digital backbone of adaptable, scalable industrial infrastructure.

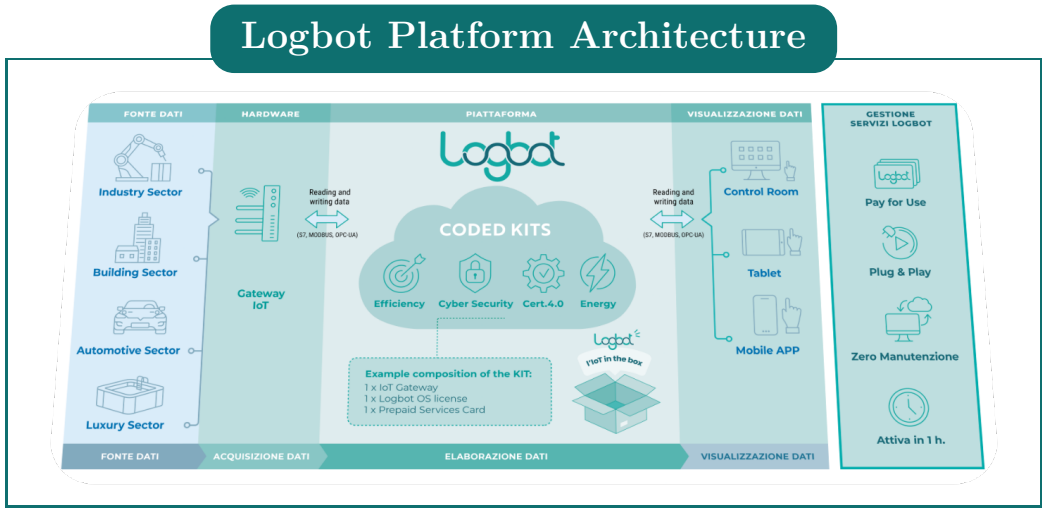


Figure 1.1: Logbot Platform Overview

Chapter 1

Platform Overview

Logbot is a modular Industrial IoT platform for connecting, monitoring, and controlling distributed assets. It provides secure data acquisition, real-time visualization, alarms, user/role management, and workflow tools to operate Plants, Gateways, and Devices at scale. This overview gives you the mental model you need before you dive into setup and daily operations.

Key Platform Modules

- **Home** – Acentralized landing page providing summaries like dashboards, tutorials,blog updates, and helpful links for navigating the platform.
- **Dashboards**: – Create and manage KPIs and data visualizations.
- **Assets Management** – Organize and manage devices, gateways, and plants. This module includes functionalities for creating device groups, mapping device networks, and associating metrics.
- **Alerts** – Define rules and priorities; notify the right people.
- **Users & Roles** – Admins can manage user access through roles (Viewer, Editor, Admin), assign permissions, and control visibility into platform modules using detailed role settings.
- **Groups & Requests** – Supports organizational grouping of users and plants. Users can request access using unique Group Codes and be invited via email. Admins can accept or reject these requests from a dedicated interface
- **Models & Licenses** – Maintain models and ensure license availability.
- **Commands** – Run scheduled or on-demand actions on assets.
- **Events & History** – Review past alarms, actions, and audit logs.

Designed For: This platform is ideal for administrators, operators, and analysts who want to track operational data, enforce security policies, assign controlled access, and gain insights from the IoT infrastructure in real time.

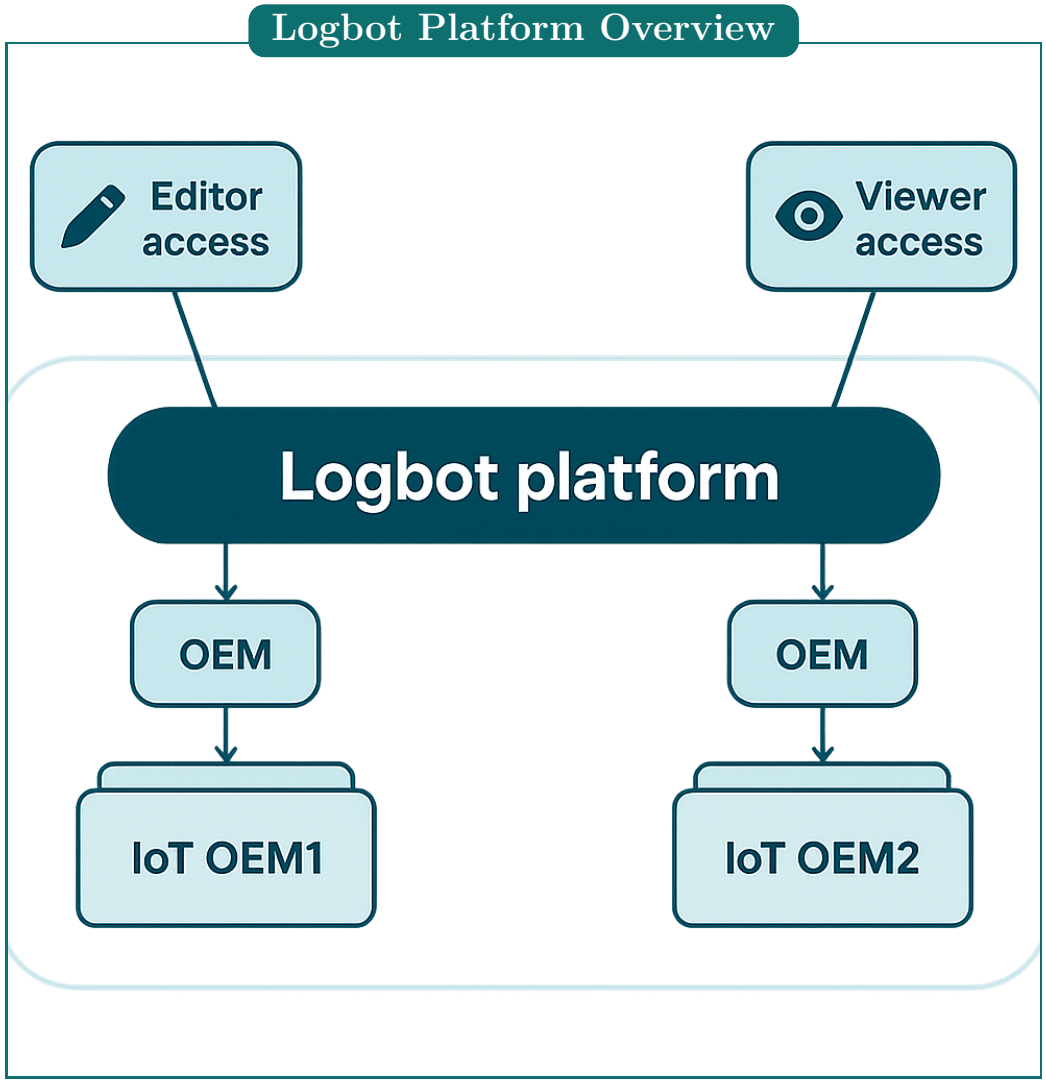


Figure 1.1: Logbot Platform Overview

Chapter 1

Navigating the Home Screen

Prerequisites

Navigation to Home Screen:

- Valid Logbot account (username & password) or SSO access.
- Role **Viewer** (or higher). Editors/Admins may see extra options.
- Network access to the Logbot URL (connect VPN if required by your company).
- Recommended: set your **Time Zone** and **Language** in your user profile.

The Home section of the Logbot platform serves as the initial landing page for users upon logging into the system. It provides an overview of activity and acts as a gateway to other key modules in the platform.

Main Panels

The dashboard is divided into three primary panels:

- **Dashboards:** Displays links to starred dashboards and recently viewed dashboards. This helps users quickly resume work or monitor key metrics.
- **Latest from the Blog:** Shows updates, tips, or technical articles from the Logbot team to keep users informed of changes or best practices.
- **Useful Links:** Provides access to essential resources including:
    1. Documentation
    2. Tutorials
    3. Blog
    4. Report a Bug

User Profile and Language

At the top-right corner, users can:

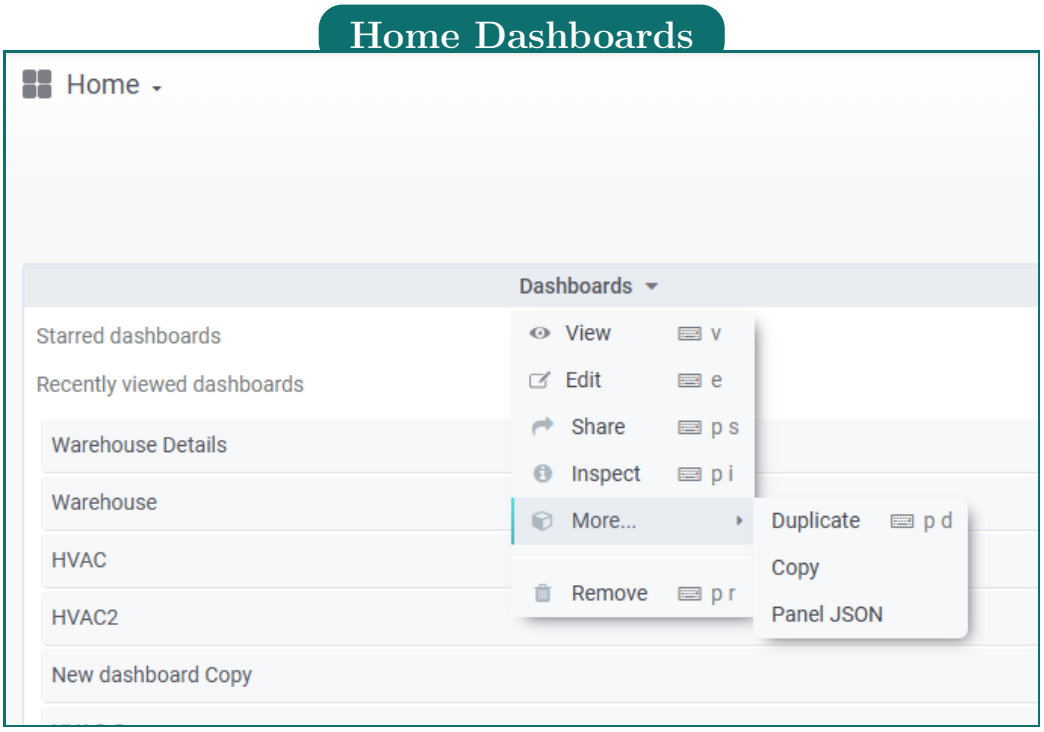
- Change their language preference
- Access their user profile
- Adjust system settings

Time Zone Configuration

The platform allows users to configure the local time zone, which is especially useful for setting up time-based alerts or schedules.

1.1 Dashboards

- **View** – View the selected dashboard.
- **Edit** – Edit dashboard settings or content.
- **Share** – Generate shareable link, embed code, or snapshot.
- **Inspect** – Analyze panel JSON or metrics.
- **More** – Additional options like Duplicate, Panel JSON.
- **Remove** – Delete the dashboard.



1.1.1 View Section

From the **Logbot Home** interface, users can access Dashboards using the left sidebar. Once inside the Dashboard panel, a dropdown appears when hovering over the "Dashboards" label. This menu provides several actions such as View, Edit, Share, and Inspect. Selecting the **View** option allows users to open and explore existing dashboards, including visualizations, graphs, and widgets relevant to their assets or services.

1.1.2 Edit Section

When the **Edit** option is selected for a dashboard, users are presented with a wide selection of visualization tools. This includes widgets such as Graph, Stat, Gauge, Table, Heatmap, HMI display panel, News Panel, and many others.

These panels allow for fully customized dashboards tailored to the needs of industrial monitoring, reporting, and control. Users can

search, filter, and organize their panels by tags, folders, or specific titles from the sidebar.

The Dashboardlist menu in Logbot provides an overview and quick access to different dashboards. Users can easily search, filter, and navigate their dashboards using the visualization panel Visualization.

Options:

- **Starred**: Displays dashboards marked as favorites.
- **Recently Viewed**: Shows dashboards that were recently opened.
- **Search**: Enables search functionality for dashboard titles.
- **Show Headings**: Displays dashboard headings for easier organization.
- **Max Items**: Limits the maximum number of dashboards displayed (default: 30).

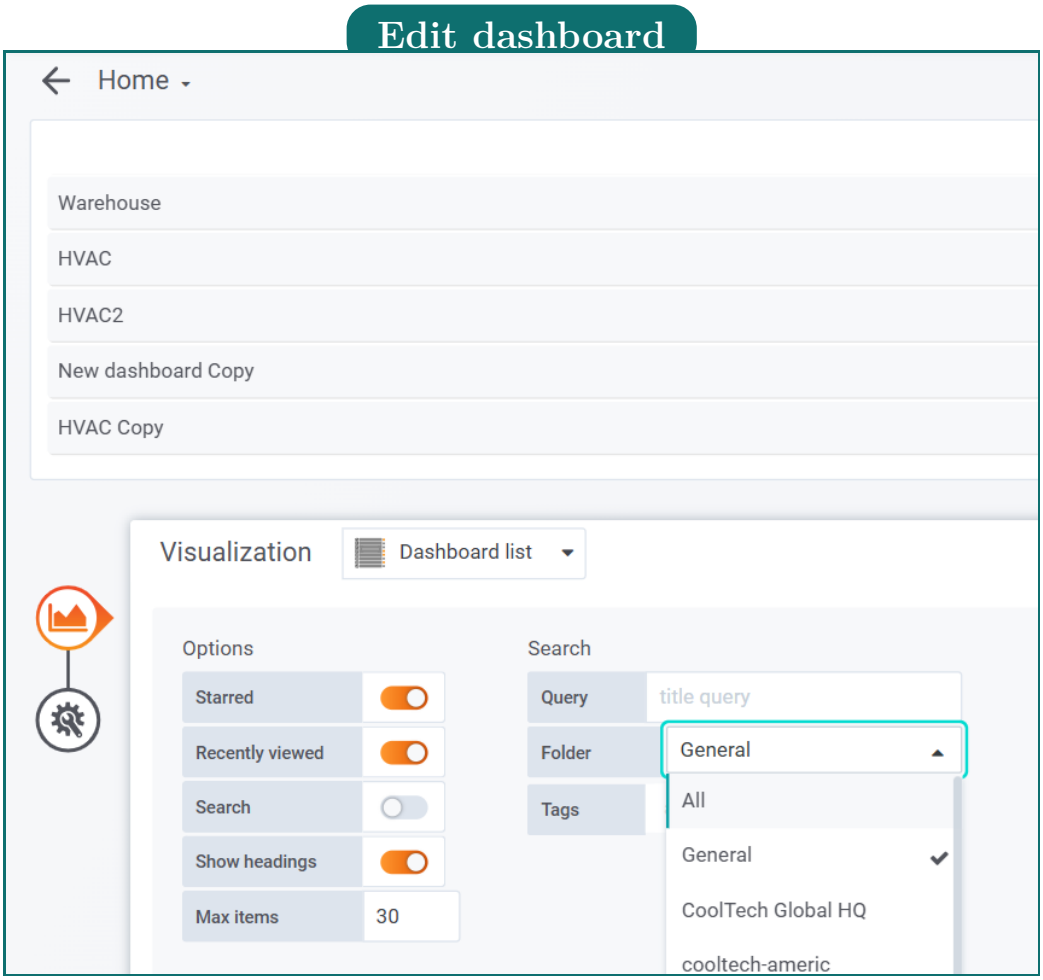
Search Filters: The following filters refine dashboard results:

- **Query**: Enter keywords to search dashboard titles.
- **Folder**: Select a folder (*General*, *Custom*, etc.) to restrict the search within that folder.
- **Tags**: Filter dashboards by user-defined tags for better organization.

Usage.

1. Navigate to the **Dashboards** section in the left menu.
2. Use the **Options** toggles to display favorites, recently viewed, or all dashboards.
3. Apply **Search Filters** such as folder selection or tags to quickly locate a specific dashboard.
4. Adjust the **Max Items** setting if you want to see more or fewer dashboards in the list.

This section ensures users can quickly access and organize their dashboards, improving workflow efficiency.



General Panel

The **General Panel** in dashboards defines the basic settings for how a panel is displayed and behaves.

General Settings:

- **Title:** The name of the panel (e.g., "Dashboards").
- **Description:** Optional field to add context or notes; supports markdown and links.
- **Transparent:** Toggle to enable a transparent background for the panel (ON/OFF).

Repeating:

- Panels can be repeated dynamically based on template variables.
- **Options:** Disabled (default) or repeat based on available template variables.

Panel Links:

- Add clickable links to external dashboards or resources.
- Each link includes:
  - **Title:** Display name of the link (e.g., "Show details").
  - **URL:** Destination link; supports data variables such as series name, labels, and values.
  - **Open in new tab:** Option to open the link in a new browser tab (ON/OFF).

Usage Example:

- A panel titled "Dashboards" with a transparent background.
- Description may include links to documentation.
- Repetition can be used to generate multiple panels for each template variable.
- A panel link could point to another Grafana dashboard with details about the data.

1.1.3 Share Section

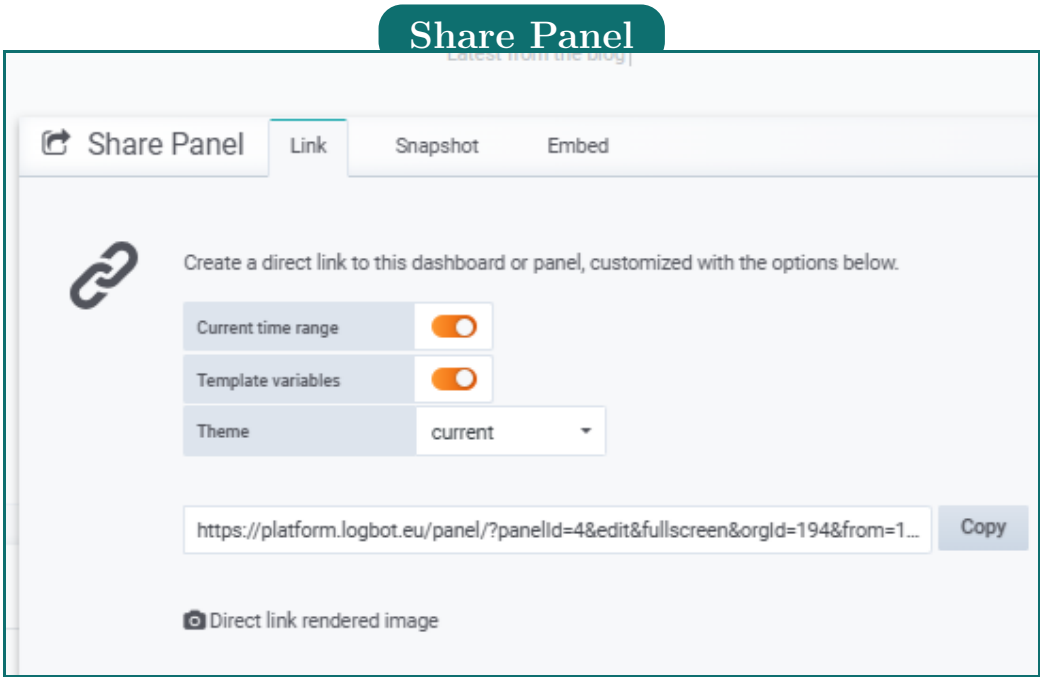
To share a dashboard panel, click the `Share Panel` from the drop-down list. You will be presented with several tabs like `Link`, `Snapshot`, and `Embed`. You can choose to include the current time range and template variables for the shared link, as shown below.

Link Options:

- **Current time range:** If enabled, the shared link preserves the currently selected time range.
- **Template variables:** If enabled, the link includes the active template variable values.
- **Theme:** Choose how the dashboard is displayed in the shared link:
  - **Current:** Uses the viewer's active theme.
  - **Dark:** Forces a dark theme.
  - **Light:** Forces a light theme.
- **Direct link rendered image:** Option to generate a link that displays the panel as a rendered image.

Usage Example:

- Share a panel showing system metrics with a fixed time range and dark theme.
- Embed a panel into documentation using the provided HTML code.
- Send a snapshot to a colleague so they can view the data at a specific point in time without needing access to the live dashboard.



Snapshot Panel:

The **Snapshot** feature allows you to create a static version of your dashboard panel, which can be shared publicly. Snapshots strip out sensitive data (queries, template variables, and annotations) and keep only the visible metrics and series values.

- **Snapshot Name:** The name assigned to the snapshot (e.g., *Home*).
- **Expire:** Defines how long the snapshot remains available.
  - **Never:** The snapshot will not expire.
  - **1 Hour:** Snapshot expires after 1 hour.
  - **1 Day:** Snapshot expires after 24 hours.
  - **7 Days:** Snapshot expires after 1 week.
- **Timeout (seconds):** Specifies the maximum time allowed for collecting the dashboard's metrics (e.g., 4 seconds).

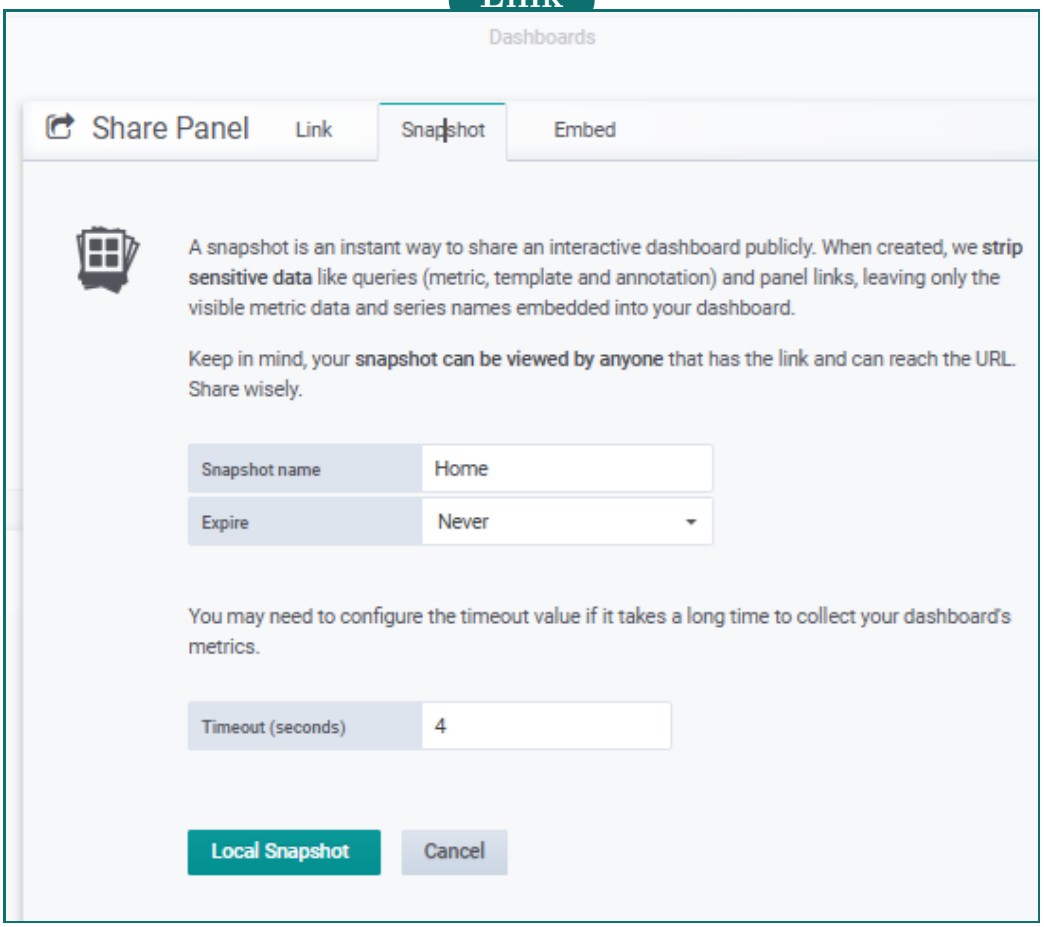
Important Notes.

- Anyone with the snapshot link can view it, so share wisely.
- Snapshots are static and will not update with new incoming data.
- Useful for presentations, reporting, or sharing dashboard states at a specific moment.

Actions.

- **Local Snapshot:** Creates the snapshot and makes it available locally.
- **Cancel:** Discards the snapshot creation.

Link



Embed Panel:

The **Embed Panel** option provides an HTML snippet that allows you to embed a dashboard panel directly into another web page. This is useful for integrating Logbot visualizations into custom portals or external dashboards.

Current time range:
Toggles whether the embedded panel preserves the current time filter from the dashboard.

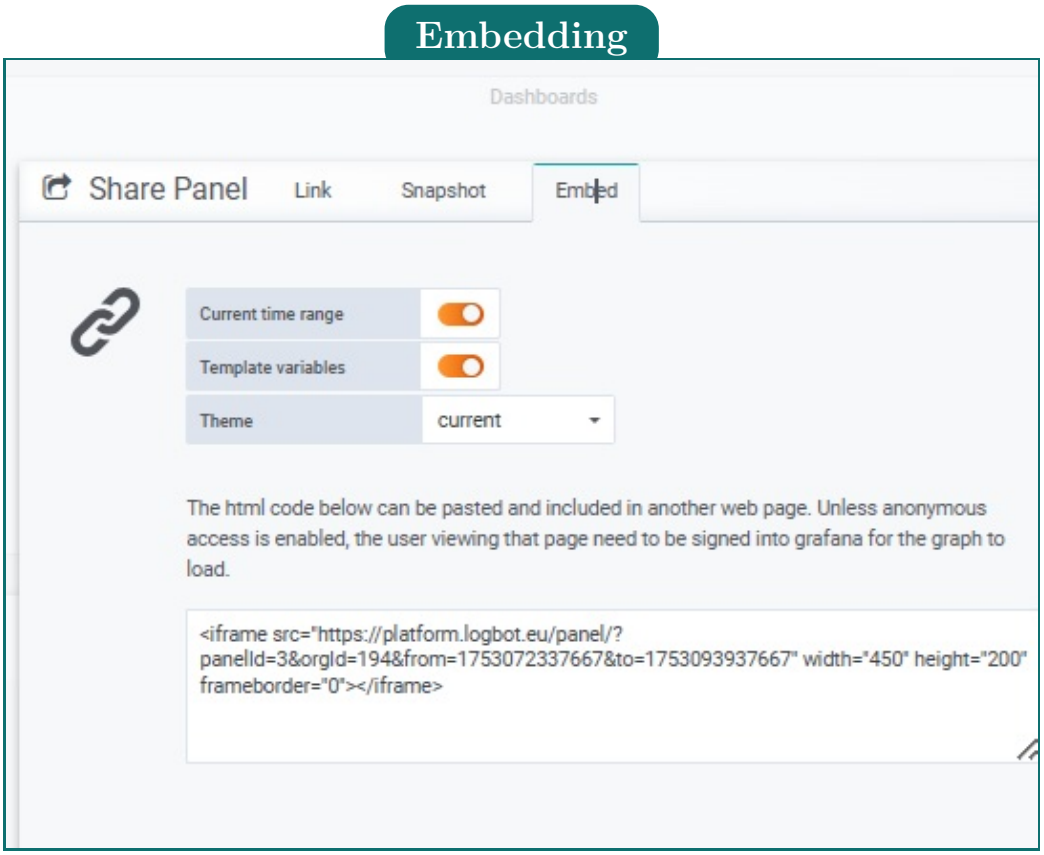
Template variables:
Toggles inclusion of template variables in the embedded panel, ensuring it reflects dynamic dashboard filters.

Theme: **Selects the display theme for the panel:**

- **Current**: Uses the current dashboard theme.
- **Dark**: Forces dark mode.
- **Light**: Forces light mode.

Embed Code:

Provides an HTML `<iframe>` snippet, which can be copied and inserted into external applications or web pages to render the panel.



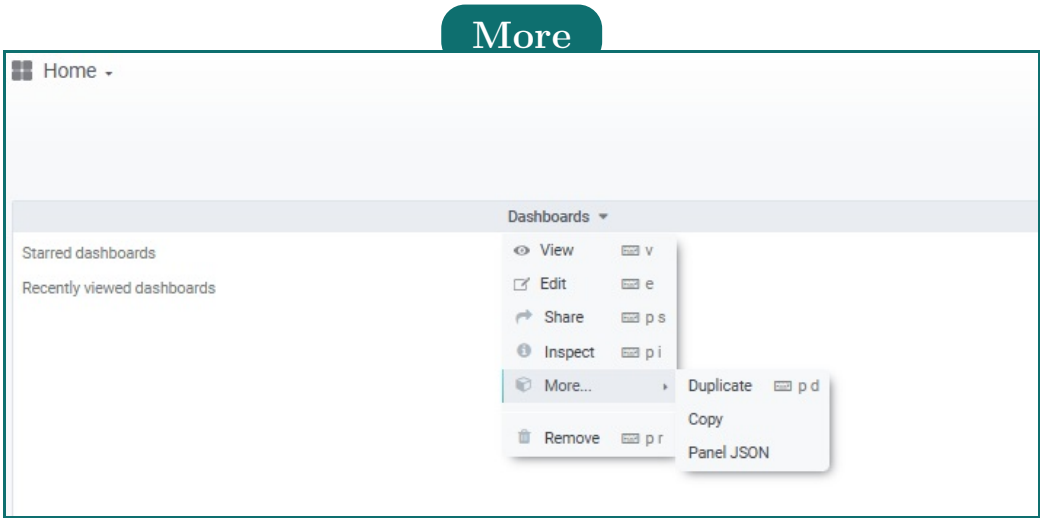
1.1.4 More Section

In the **Dashboard** section, the **More** menu provides additional functionalities for managing panels. When you click on *More*, the following options are available:

- **Duplicate:** Creates an exact copy of the selected panel within the same dashboard.
- **Copy:** Copies the panel configuration so that it can be reused or pasted elsewhere.

- **Panel JSON:** Opens the JSON representation of the panel, allowing advanced users to inspect or modify the raw configuration.

These options are particularly useful for quickly reusing panels, sharing configurations, or performing advanced edits.sou



1.1.5 Remove Section

The **Remove** option allows users to delete a selected dashboard panel or an entire dashboard. When this action is executed, the panel is permanently removed from the workspace.

- **Access:** Open the dropdown menu of the dashboard and click on **Remove**.
- **Shortcut:** You can also use the keyboard shortcut (`p r`) for quick removal.
- **Warning:** Removing a dashboard is irreversible. Ensure that you have backed up or exported any critical data before deletion.

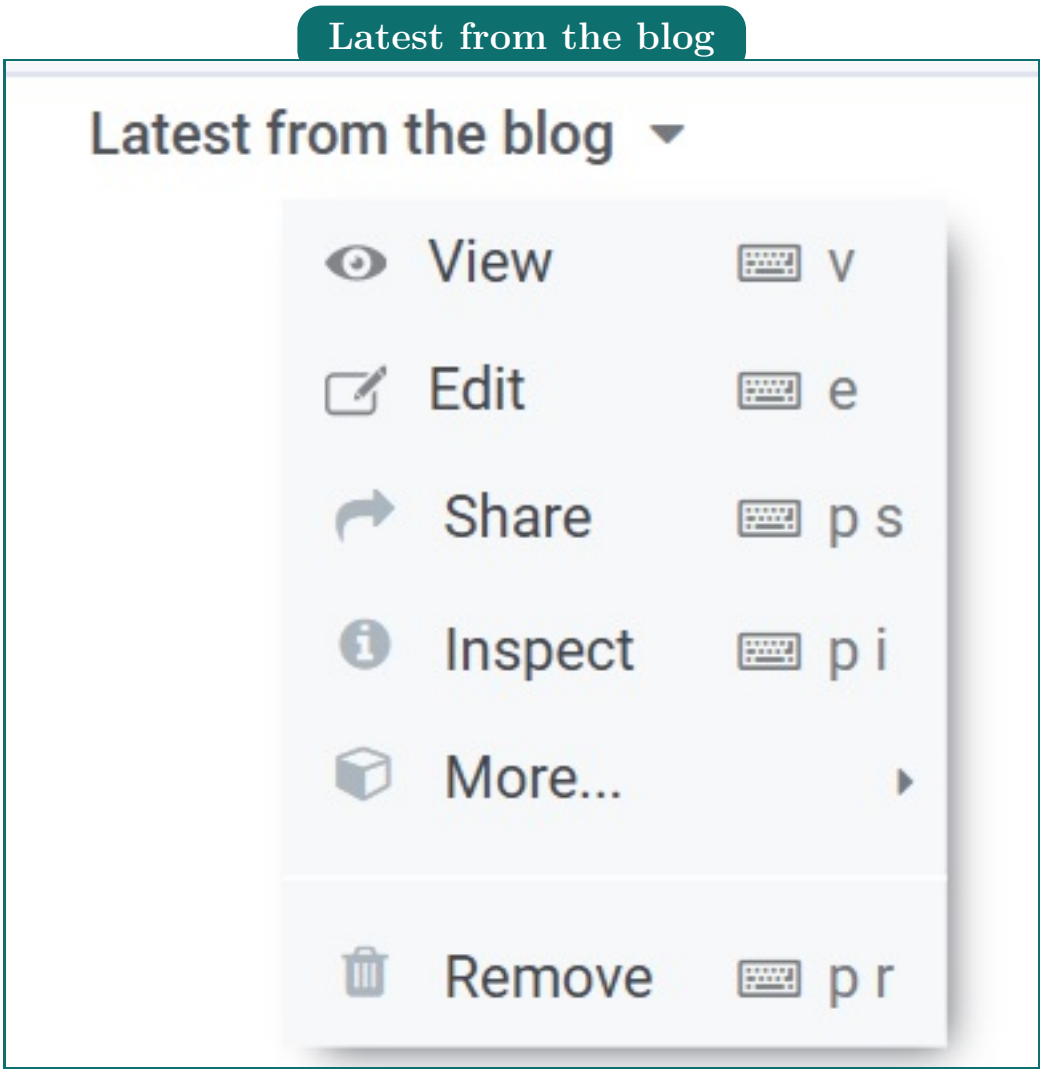
1.2 Latest from the Blog

The **Latest from the Blog** section provides real-time updates and articles related to platform features, best practices, release notes, or system improvements.

It can be accessed from the main **Home** dashboard panel. Users can interact with this widget by:

- Viewing the blog list
- Accessing details such as release notes or usage tutorials
- Managing blog visibility through options like `View`, `Edit`, `Share`, `Inspect`, or `Remove`

- **View** – View the selected dashboard.
- **Edit** – Edit dashboard settings or content.
- **Share** – Generate shareable link, embed code, or snapshot.
- **Inspect** – Analyze panel JSON or metrics.
- **More** – Additional options like Duplicate, Panel JSON.
- **Remove** – Delete the dashboard.



1.2.1 View Section

The **View** option under ***Latest from the Blog*** allows users to open and display the blog panel directly. This section provides a quick overview of recent updates or announcements within the platform. By selecting **View**, users can:

- Instantly access the panel content.
- See the most recent updates posted in the blog section.
- Interact with the dashboard in real-time.

This option is useful when you want to quickly review information without editing or changing the panel.

1.2.2 Edit Section

In the **Latest from the Blog** section, users can customize how blog-related updates are visualized on the dashboard.

By selecting the `Edit` option from the dropdown, the user is presented with various visualization panels. In this example, the **News Panel** has been selected for representing the blog feed.

In the **Latest from the Blog** section lets you place a small **news/feed** widget on your dashboard and choose how it looks.

- Different visualization widgets such as `Graph`, `Stat`, `Table`, and `Pie Chart` are available.
- The `News Panel` is chosen specifically to showcase blog updates in a clean and structured way.
- Feed URLs can be added at the bottom to define external sources.

1.3 Useful Links

In the **Home** section of the Logbot platform, users are welcomed with a dashboard overview. On the right side of the screen, there is a panel labeled **Useful Links**, which provides quick access to helpful resources for navigating and utilizing the platform.

- **View** – View the selected dashboard.
- **Edit** – Edit dashboard settings or content.
- **Share** – Generate shareable link, embed code, or snapshot.
- **Inspect** – Analyze panel JSON or metrics.
- **More** – Additional options like Duplicate, Panel JSON.

- **Remove** – Delete the dashboard.

1.3.1 Documentation

Documentation :Access the official platform documentation containing detailed technical and user information.

1.3.2 Tutorials

Tutorials : View step-by-step guides and practical walkthroughs to help users understand key features.

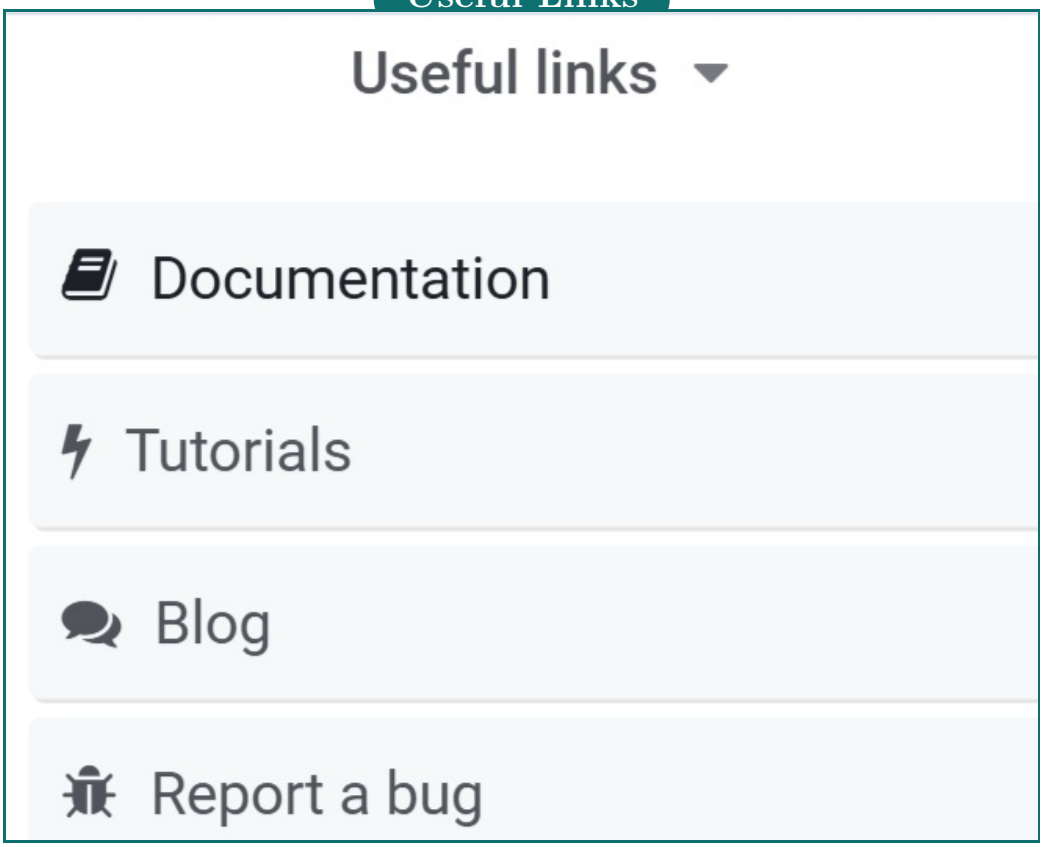
1.3.3 Blog

Blog : Explore the latest articles, updates, and platform insights.

1.3.4 Report a Bug

Report a bug Use this option to report any issues or errors encountered while using the platform. This section is designed to help users find assistance and stay informed about the platform's features and developments.

Useful Links



Chapter 1

The Setup Wizard

1.1 Setup Wizard

The Setup Wizard is a comprehensive, step-by-step workflow designed to streamline the initial configuration of the entire platform. It guides users through all essential components required to deploy an operational environment, ensuring accuracy, consistency, and system readiness.

This interactive process helps prevent misconfigurations by validating each step before proceeding to the next. Whether configuring a single plant or a multi-gateway, multi-device infrastructure, the wizard ensures a smooth onboarding experience for administrators and operators alike.

Steps Overview 

1. **Plant Setup** – Define the name, location, and description of your plant. This acts as the organizational foundation where all gateways and devices will be grouped.

2. **Gateway Registration** – Add gateways to the platform, assigning names, descriptions, models, and licenses. Each gateway serves as a communication bridge between field devices and the backend platform.

3. **Gateway Connectivity Test** – Verify the physical or virtual gateway's online status and network communication to ensure it can interface with the platform.

4. **Device Configuration** – Register the devices connected to each gateway. Specify device details such as protocol type, connection parameters, and address information.

5. **Device Model Creation** – Create templates for devices that define their communication structures, protocols, and expected data schema. Reusable across similar device types.

6. **Create Connection** – Establish how the gateway communicates with the connected device using the selected protocol (e.g., Modbus, S7, OPC UA). Users input protocol-specific settings such as host, port, or serial configurations.

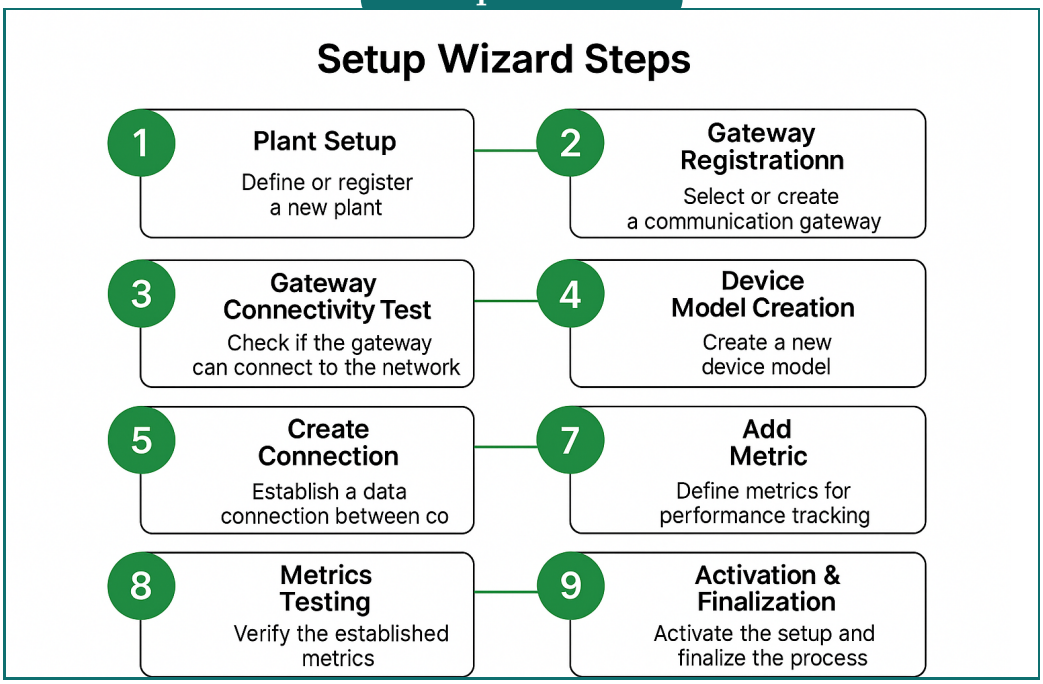
7. **Add Metric** – Define data tags or metrics the system should collect from the device. Parameters include data type, unit, permissions, access level, and static tag mappings.

8. **Metrics Testing** – Run test reads on the defined metrics to ensure correct value retrieval and tag behavior from the devices.

9. **Activation & Finalization** – Review all configurations. Once validated, the environment is activated for real-time monitoring and data acquisition.

This wizard-setup approach eliminates guesswork and helps users deploy scalable, error-free configurations tailored to their specific operational requirements.

Setup Wizard



1.1.1 Step 1: Plant Setup

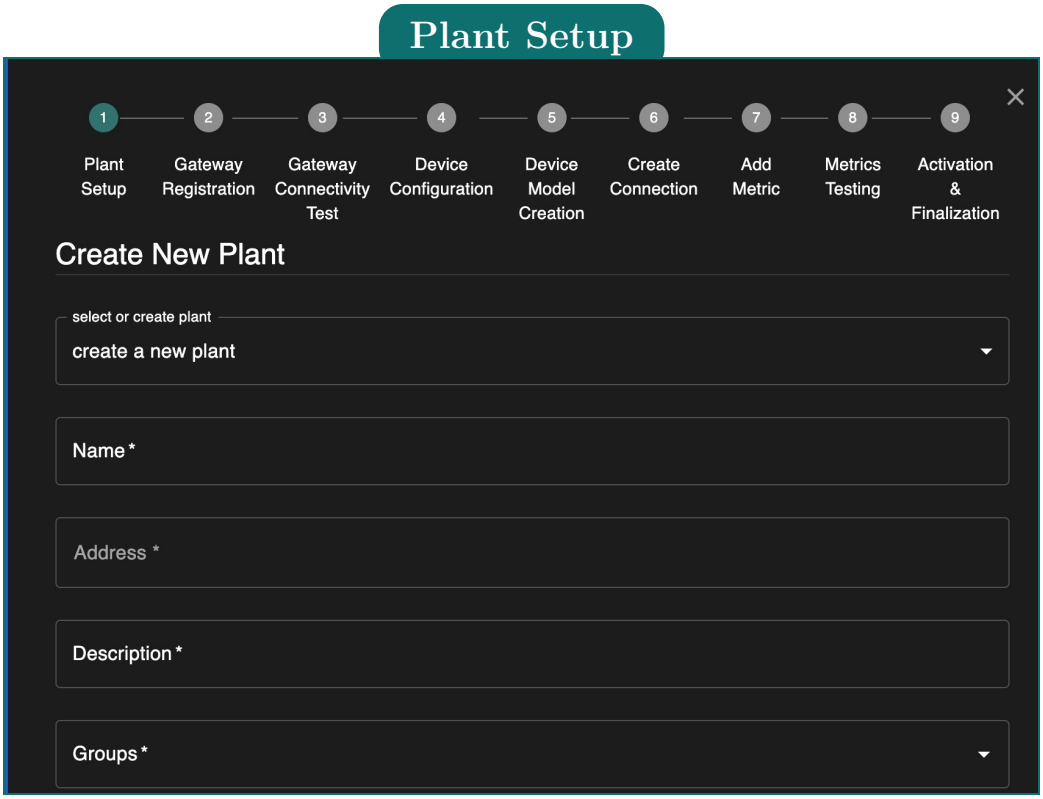
The **Plant** is the highest-level organizational entity in the platform. It represents a physical location or logical environment—such as a factory, building, or system domain—where assets, gateways, and devices operate.

This step lays the foundation for your platform's hierarchy and ensures all components are linked to a defined location.

User define:

- **Name:** A unique, descriptive identifier for your plant.
- **Address:** The geographical or logical location (can be a physical address or facility name).
- **Description:** (Optional) A brief note about the plant's purpose or context.
- **Groups:** Logical clusters to associate the plant with predefined access or operational groups. This allows for role-based visibility and permission segmentation within the platform.

This initial configuration helps structure your deployment clearly and makes it easier to manage access, monitoring, and configurations across your organization.



1.1.2 Step 2: Gateway Registration – Connect the Communication Hub

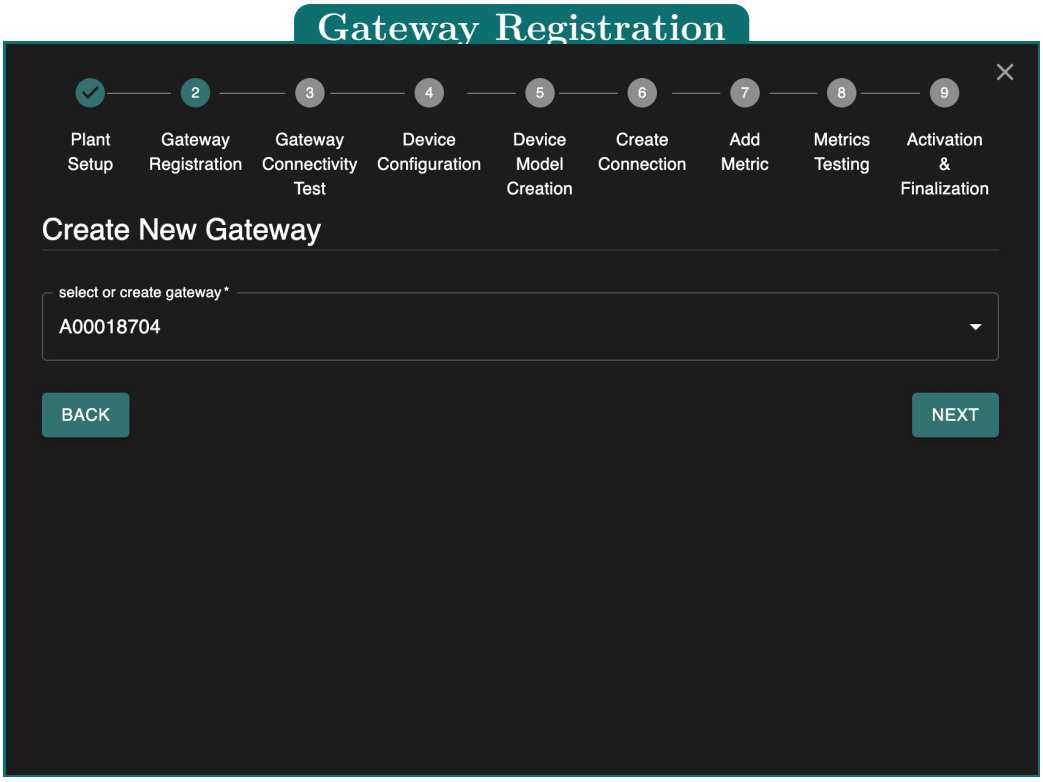
The **Gateway** acts as the communication bridge between edge devices (e.g., PLCs, sensors) and the cloud/server. It handles protocol translation, data transmission, and real-time connectivity.

At this step, users select or register the gateway that will serve as the interface for data collection and device interaction.

User define:

- **Gateway ID:** A unique identifier assigned to the hardware gateway (e.g., `A00018704`).

Once the gateway is selected, the system associates it with the plant defined in Step 1. This registration ensures that data routing and configuration tasks are performed correctly across the platform.



1.1.3 Step 3: Gateway Connectivity Test – Verify Gateway Availability

After Create a New gateway, this step ensures it is properly connected and ready for communication with the platform. This validation is essential before assigning devices or metrics.

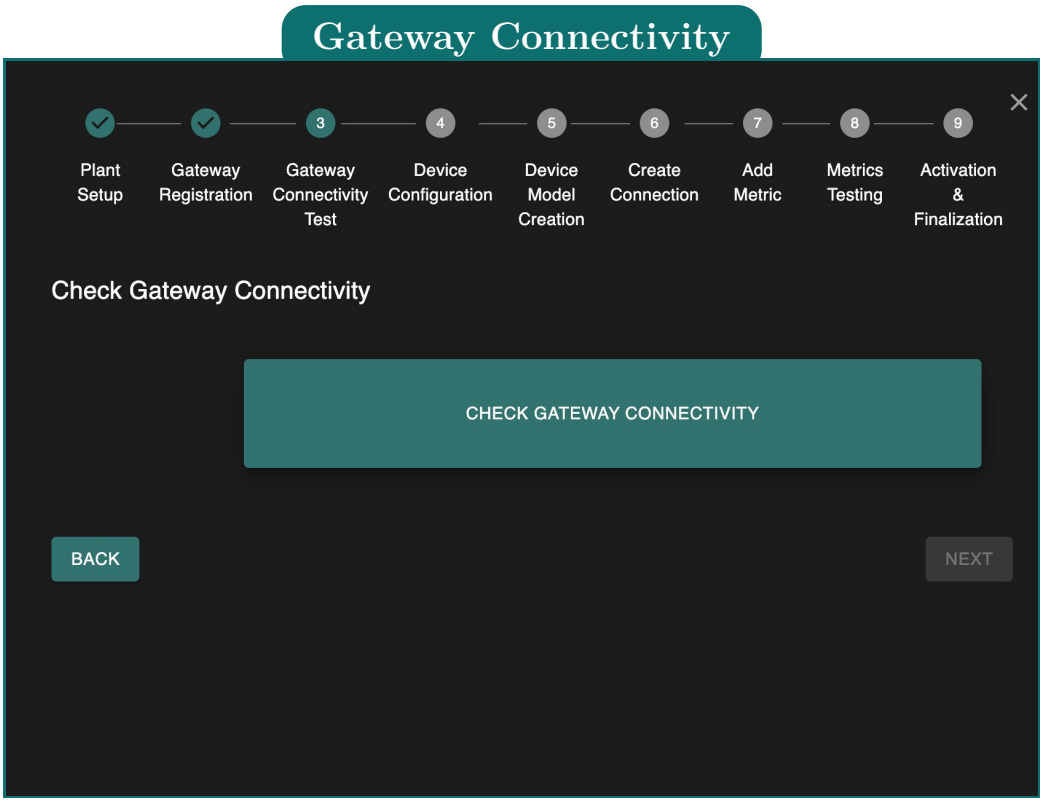
Interface Overview:

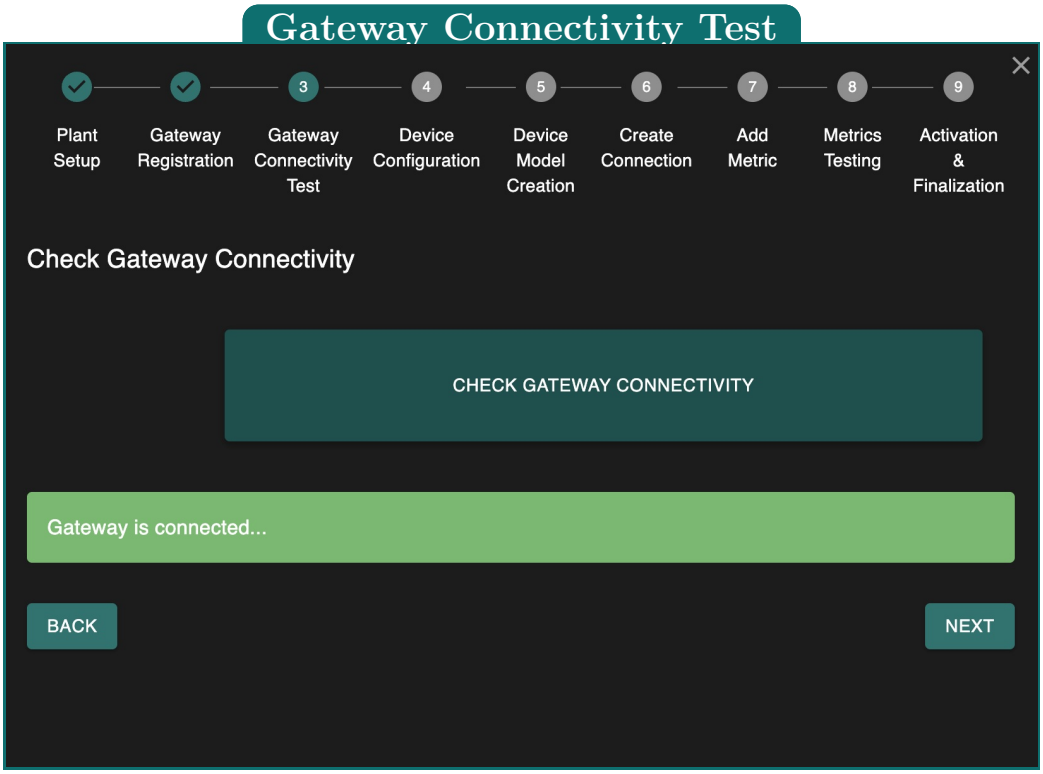
* The user clicks on the **Check Gateway Connectivity** button. * The system initiates a real-time verification with the selected gateway.

Feedback:

* If successful, a green confirmation message—*"Gateway is connected..."*—appears. * If the connection fails, the user is notified and must investigate the gateway's power, network, or configuration.

> This step acts as a safeguard to prevent users from configuring downstream elements on a non-communicating gateway. Only once the connection is confirmed can the user proceed to device-level configuration.





1.1.4 Step 4: Device Configuration – Define Data Sources

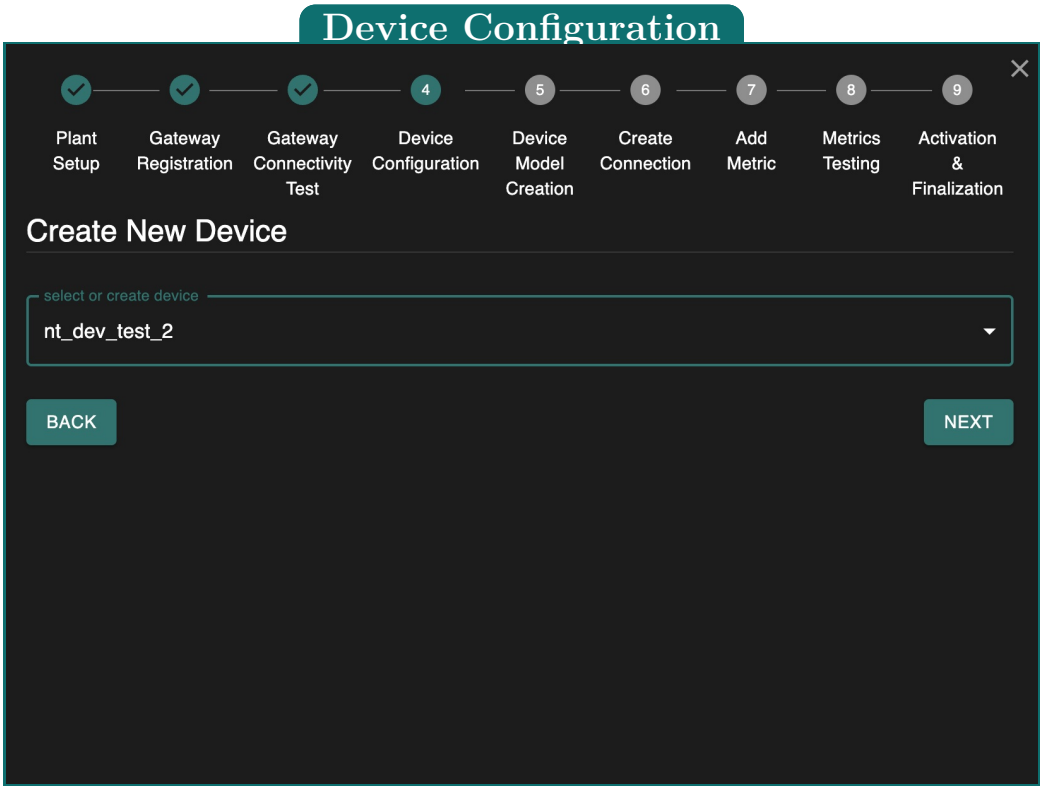
In this step, the user creates or selects a device to be linked with the connected gateway. Devices represent actual hardware or data-generating components in the field—such as sensors, controllers, or PLCs.

✓ Options Available:

* **Select or create device:** Users can either choose from pre-existing device entries or create a new one by entering a unique identifier (e.g.,'nt dev test 2').

This step ensures that each data-producing unit is logically registered in the system before assigning models or connections. It lays the foundation for applying communication protocols, mapping data tags, and structuring device-specific configurations in the upcoming steps.

> Devices created here will later be associated with models, connections, and metrics to enable full functionality and accurate monitoring.



1.1.5 Step 5: Device Model Creation

After selecting or creating the device, the next step is to associate it with a device model. A device model defines the communication protocol, expected data structure, and configuration schema used for interaction with the hardware.

Purpose:

1. It standardizes how the system interprets the data coming from different devices.
2. It defines how tags, metrics, or registers are addressed and processed.

Note: The user either selects a predefined model or creates a custom one based on the device specifications. This step ensures that the system correctly understands the device's behavior and can extract accurate information.

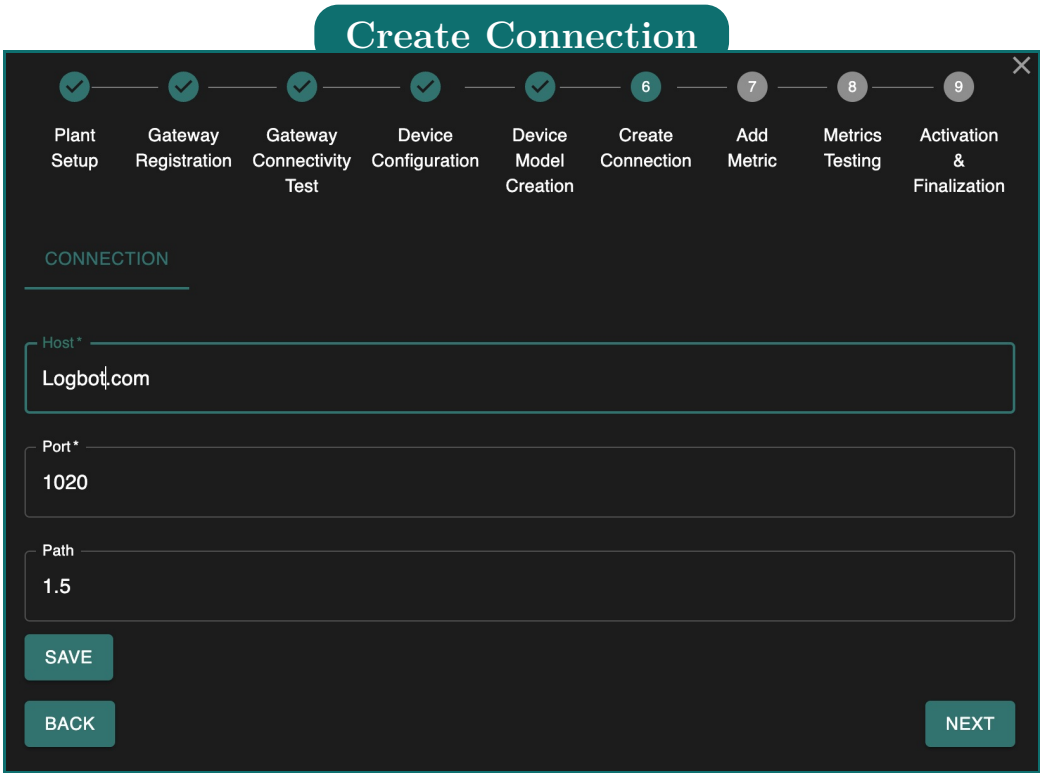
1.1.6 Step 6: Create Connection – Establish Communication Parameters

In this stage, users configure the actual communication settings between the device and the platform. This enables proper data transmission through the defined protocol.

Parameters include:

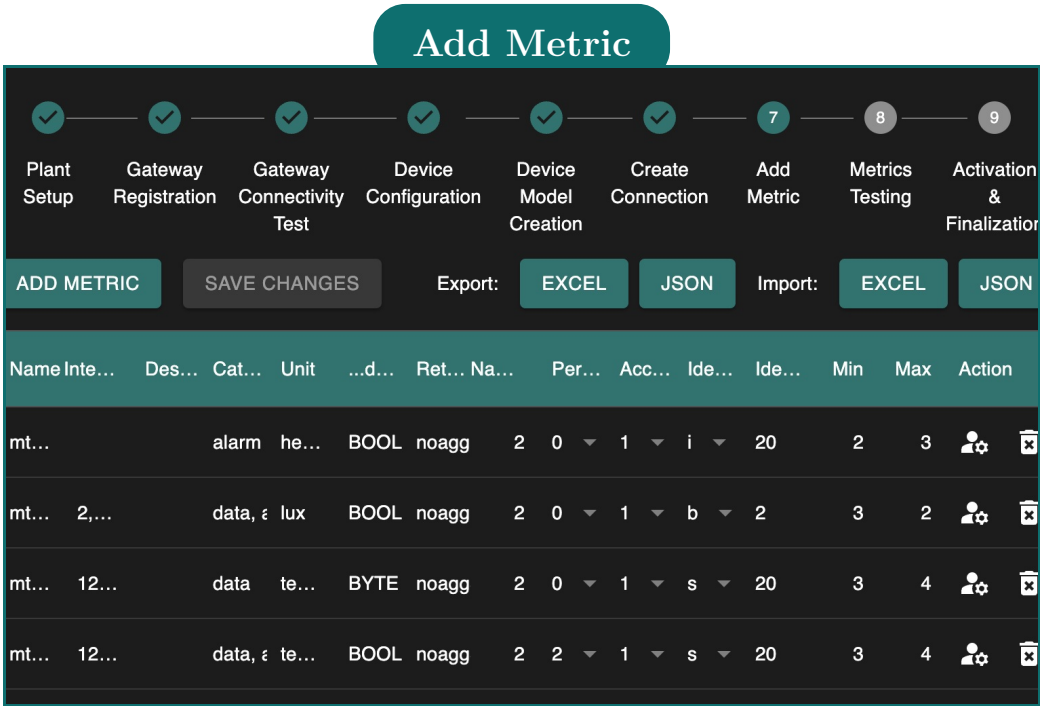
- **Host:** The IP address or domain name of the target system/device (e.g., ‘Logbot.com‘).
- **Port:** Communication port used for data transfer (e.g., ‘1020‘).
- **Path:** Specific sub-path or channel used internally (e.g., ‘1.5‘).

SAVE: must be clicked to store the configuration before proceeding.



1.1.7 7: Add Metric

In this step, you define and configure the data point (or metric) that the system should ingest and monitor from the connected device. This includes setting the datatype, retention policy, access rules, limits, tags, and other static properties. Properly defining the metric is essential for ensuring valid, reliable, and meaningful data flow.

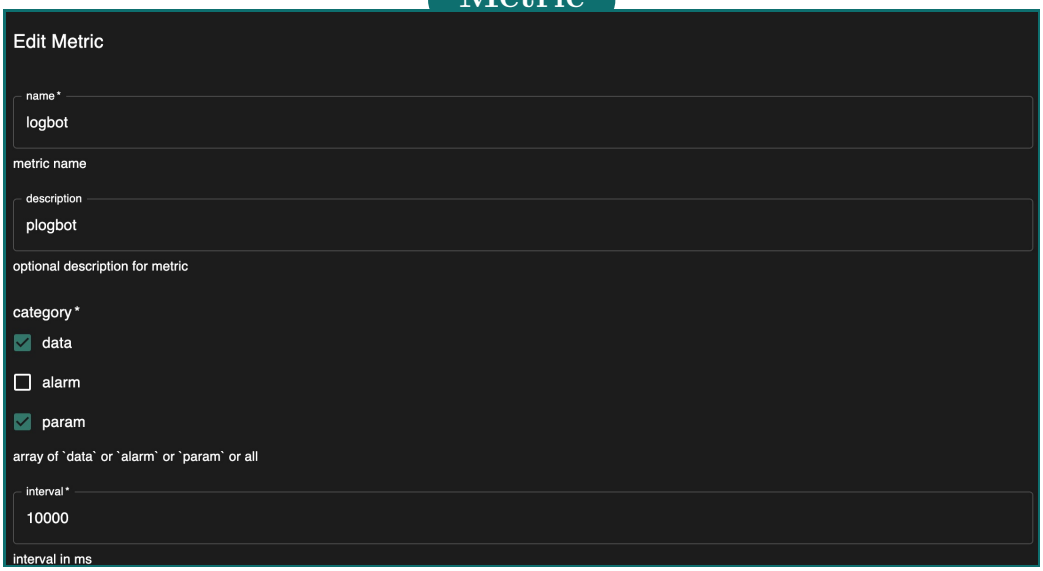


Key configuration fields include:

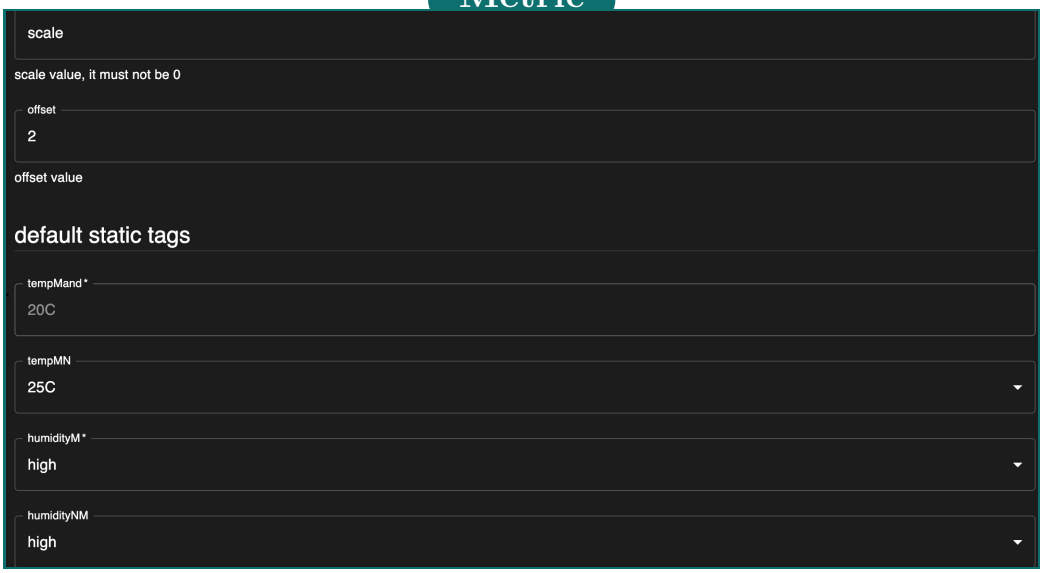
- Data Type: Choose the type of value (e.g., ‘BOOL‘, ‘INT‘, ‘FLOAT‘, etc.).
- Retention Policy**: Define how long and in what format the data should be stored (e.g., ‘noagg6m‘).
- Permissions, Access Level: Set who can read/write this data.
- Identifier Type , Value: Uniquely identifies the metric within a namespace.
- Unit, Ranges: Provide measurement units and define valid value bounds (e.g., min = 2, max = 3 for "meter").
- Scale,Offset**: Apply transformation parameters if needed.
- Static Tags: Label metrics with extra context like ‘tempMand‘, ‘humidityM‘, ‘pressureM‘, etc.

✓ **Note: ‘tempMN‘ stands for *Temperature Mandatory Noun*, indicating required label logic.**

Metric

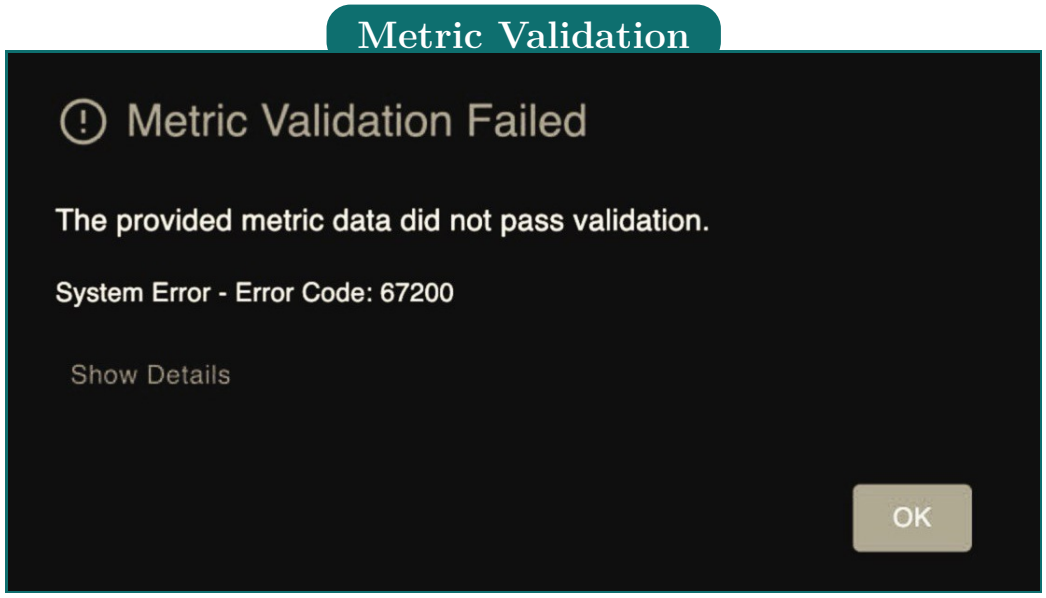


Metric



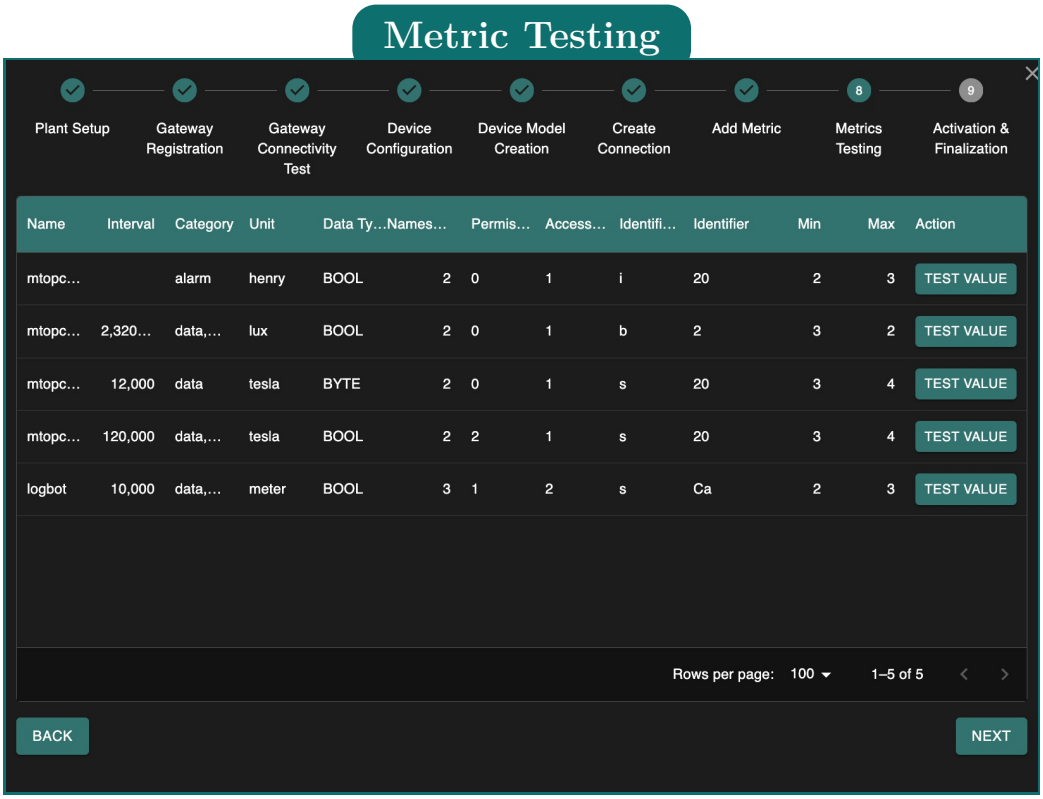
If fields are incorrectly set:

- validation may fail with an error message such as:
  ⊗ Metric Validation Failed
- The provided metric data did not pass validation.
- System Error - Error Code: 67200*



1.1.8 Step 8: Metrics Testing

The **Metrics Testing** phase is designed to validate that the metrics created earlier are properly configured and functioning as expected. This step ensures that each metric can successfully retrieve data from the connected device through the gateway.



Test Metric Functionality

The **Test Value** button available in the **Metrics Testing** step allows users to verify that each defined metric is correctly retrieving data from the connected device.

When clicked, this feature initiates a real-time query through the gateway to the device. It checks:

- Whether the metric is accessible.
- If the data format (e.g., BOOL, BYTE) matches expectations.
- If the retrieved value falls within the defined minimum and maximum range.

This process helps validate that:

- The connection between the platform, gateway, and device is functioning.
- The metric has been configured correctly.
- The system is ready for data collection and monitoring.

If a test fails, users are advised to return to the **Add Metric** step to adjust parameters before proceeding to deployment.

1.1.9 Step 9: Activation, Finalization

In the final step of the Setup Wizard, the platform presents a **Final Review of System Components** of all configured elements, allowing users to **review and confirm** their settings before deployment.

Final Review of System Components:

Plant Information:

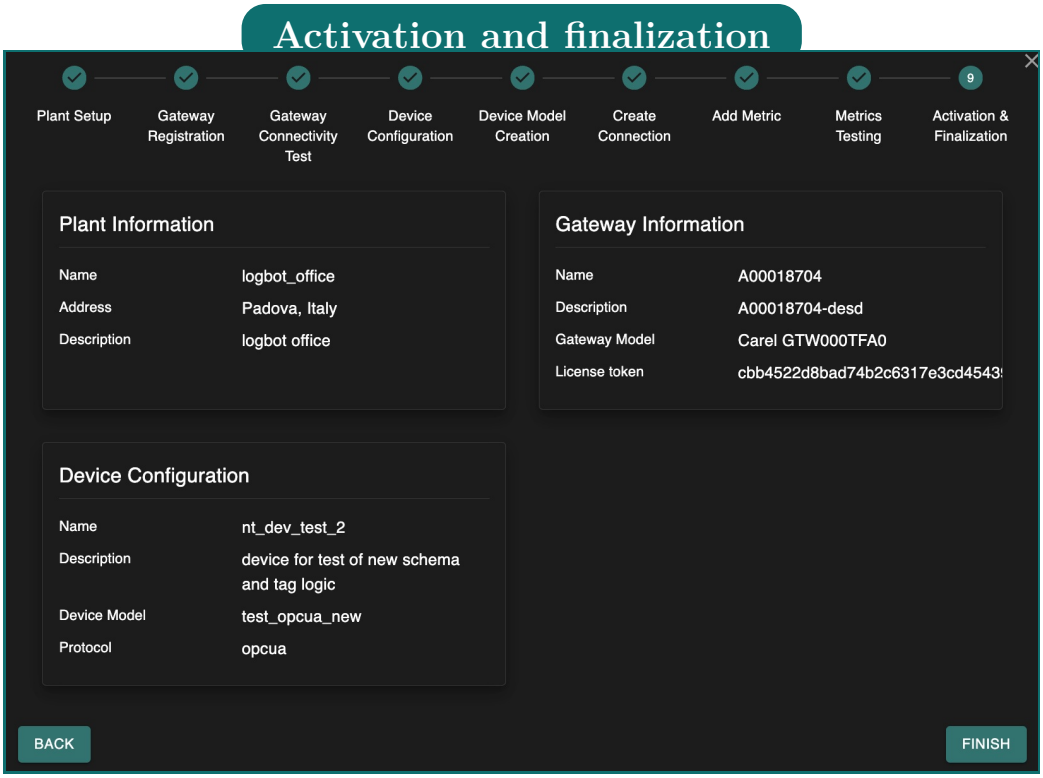
- Displays the plant name, address, and description as initially defined in the setup.

Gateway Information:

- Includes gateway ID, description, model, and license token—verifying correct registration and connection.

Device Configuration:

- Shows the associated device, including its name, description, device model, and communication protocol (e.g., OPC UA).



Chapter 1

Working with Dashboards

1.1 Manages

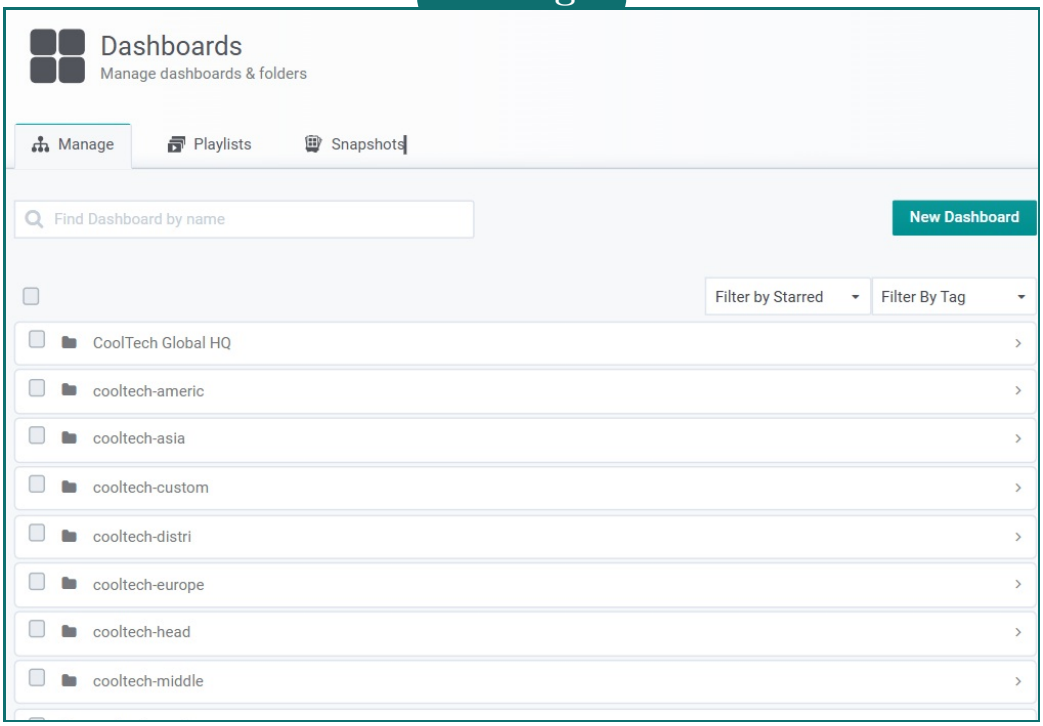
To manage existing dashboards or folders, navigate to the main menu and select:

Manage Tab

This view allows users to:

- Browse all existing dashboards.
- Access folders such as `mohajerorgtest` or `sytwo`.
- Use filtering options like **Starred** or **Tags**.
- Create new dashboards by clicking the **New Dashboard** button.

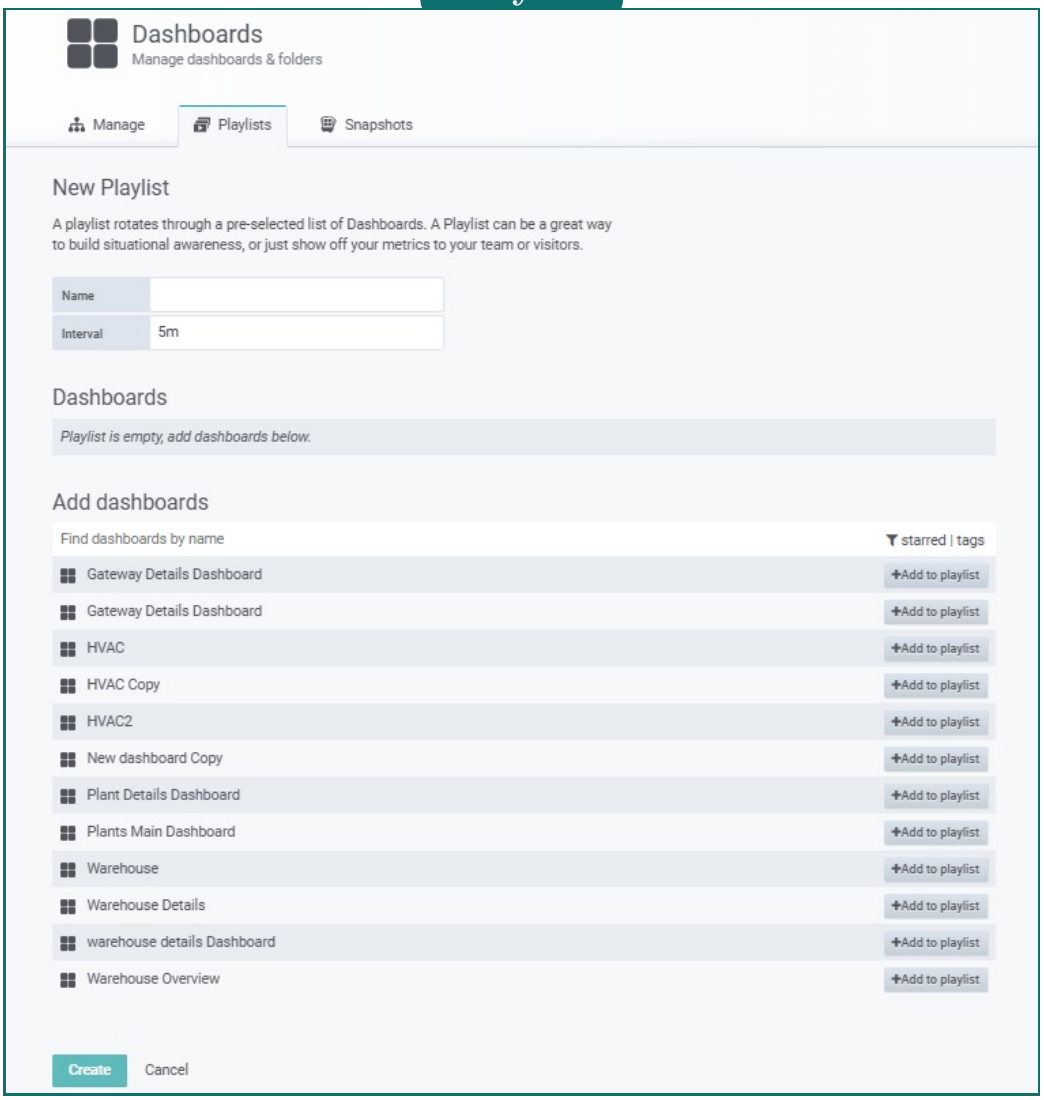
Manage



Playlists Tab

- Used to create rotating dashboards for continuous monitoring.
- Steps to create:
  1. Click **New Playlist**.
  2. Enter playlist **Name**.
  3. Set the switching **Interval** (e.g., 5m).
  4. Add dashboards with **+Add to playlist**.
  5. Click **Create**.
- Once created, use **Start Playlist** to run it.

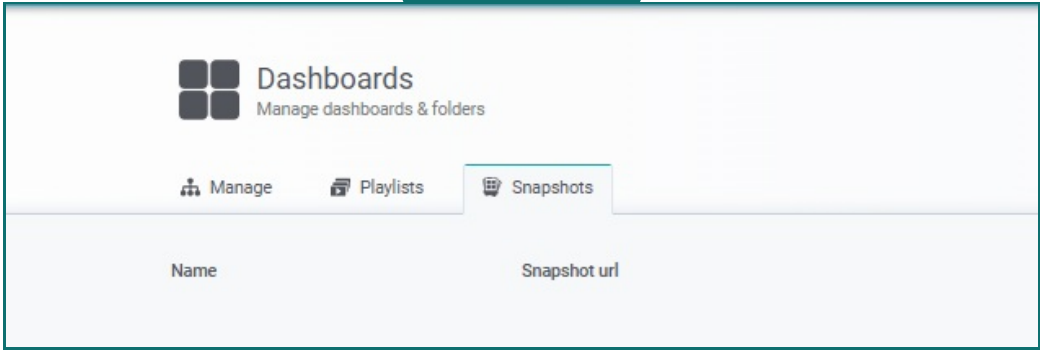
Playlist



Snapshots Tab

- Displays saved dashboard snapshots and their URLs.
- Snapshots are static and can be shared externally.
- If no snapshots exist, the section will be empty.

SnapShot



1.2 Explore – Series Tag Filtering

In the **Explore** section of the platform, users can construct queries to analyze telemetry data using flexible filters.

Within the **Series** area, tag keys can be selected to narrow down the dataset. These tags represent different metadata associated with the telemetry metrics.

Available Tag Keys Include:

- **lbuuid** – Filters using the local unit's unique identifier.
- **lbrep** – Represents internal or location-based reporting.
- **name** – Full metric path (e.g., `CGate_4G_A024279.telemetry.cpu`).
- **vartype** – Describes the type of variable.

Example Query Structures:

- To query using the `name` tag: `seriesByTag('name=CGate_4G_A024279.telemetry.cpu')`
- To query by `lbuuid`: `seriesByTag('lbuuid=71ec7ae')`

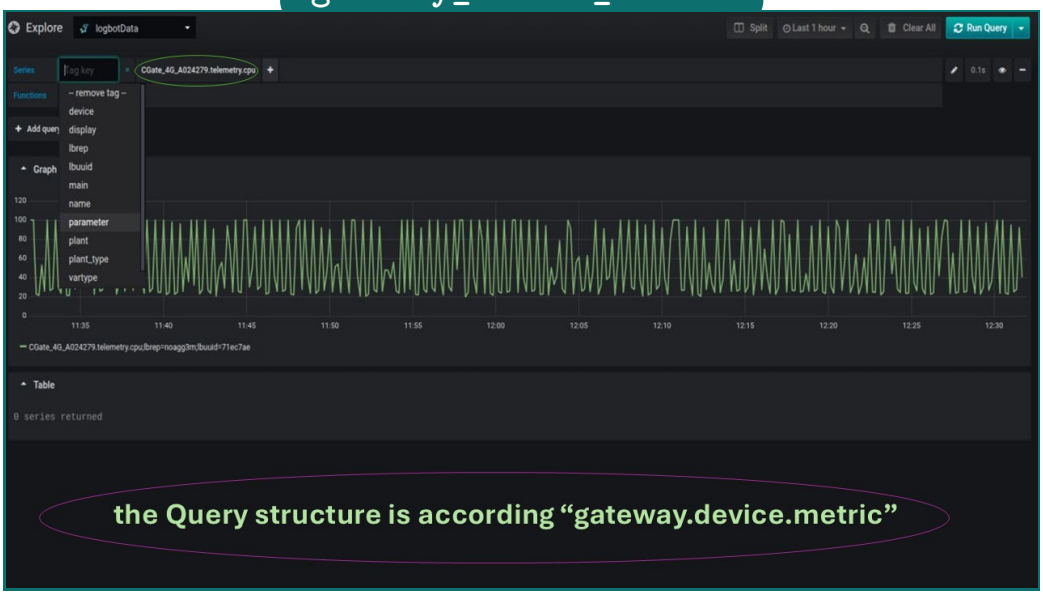
These options allow users to build custom, targeted queries to visualize system behavior over time through graphs and tables.

Query Structure — Visualization Example

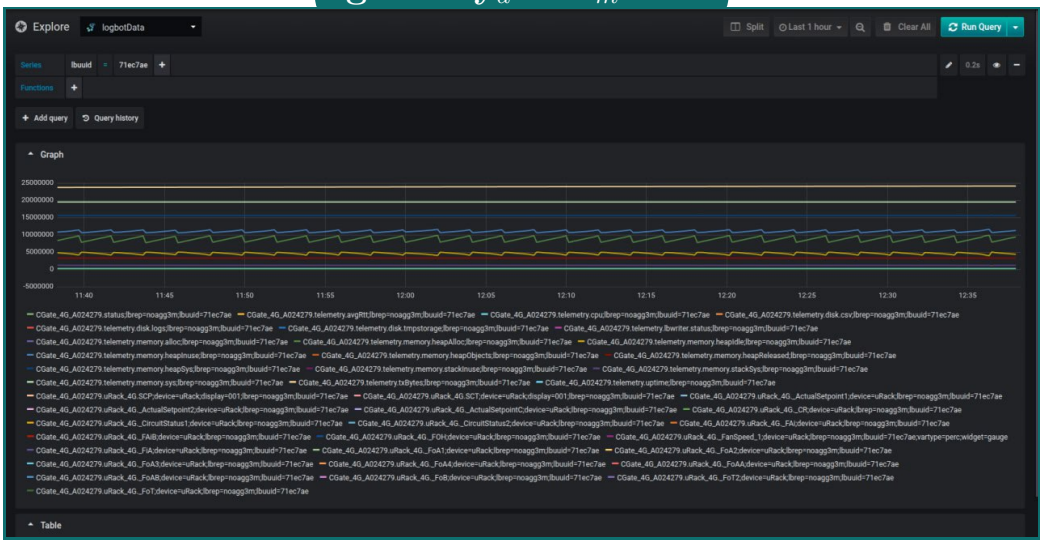
Query pattern	`<gateway>.<device>.<metric>`
Example	`CGate_4G_A024279.telemetry.cpu`

This structure ensures accurate retrieval of telemetry data for specific devices and parameters. Once the query is entered, the results are visualized in a graph, helping users monitor performance trends.

`gateway_device_metric`



$\text{gateway}_{device}{}_{metric}$



1.3 Create New Dashboard

When creating a new dashboard in Logbot, you can define how data panels are added, configured, and visualized. From the left menu, navigate to **Dashboards Section**→ + **Create**.This section introduces the basic options available when creating a new dashboard.

Navigation To

Dashboards → **Manage** → **New Dashboard** → **Dashboard settings**.

General Settings:

In **New dashboard / Settings** → **General**, configure:

- **Name**: A clear, descriptive title (e.g., `LineA_Production_Overview`).
- **Description**: A short statement of purpose (e.g., `Real-time KPIs`).
- **Tags (optional)**: Comma-separated keywords (e.g., `production, lineA, kpi`).
- **Folder**: Choose where to store it (e.g., `submain`). Create a folder first if needed.
- **Editable**: Leave **ON** unless you want to lock the layout.

Time Options:

- **Timezone**:**Default** (inherits org setting) or**Local browser time**.
- **Auto-refresh**: Common choices:**5s, 10s, 30s, 1m, 5m, 30m, 1h, 2h, 1d**.
- **Hide time picker**: Leave **OFF** unless you want a fixed time range.

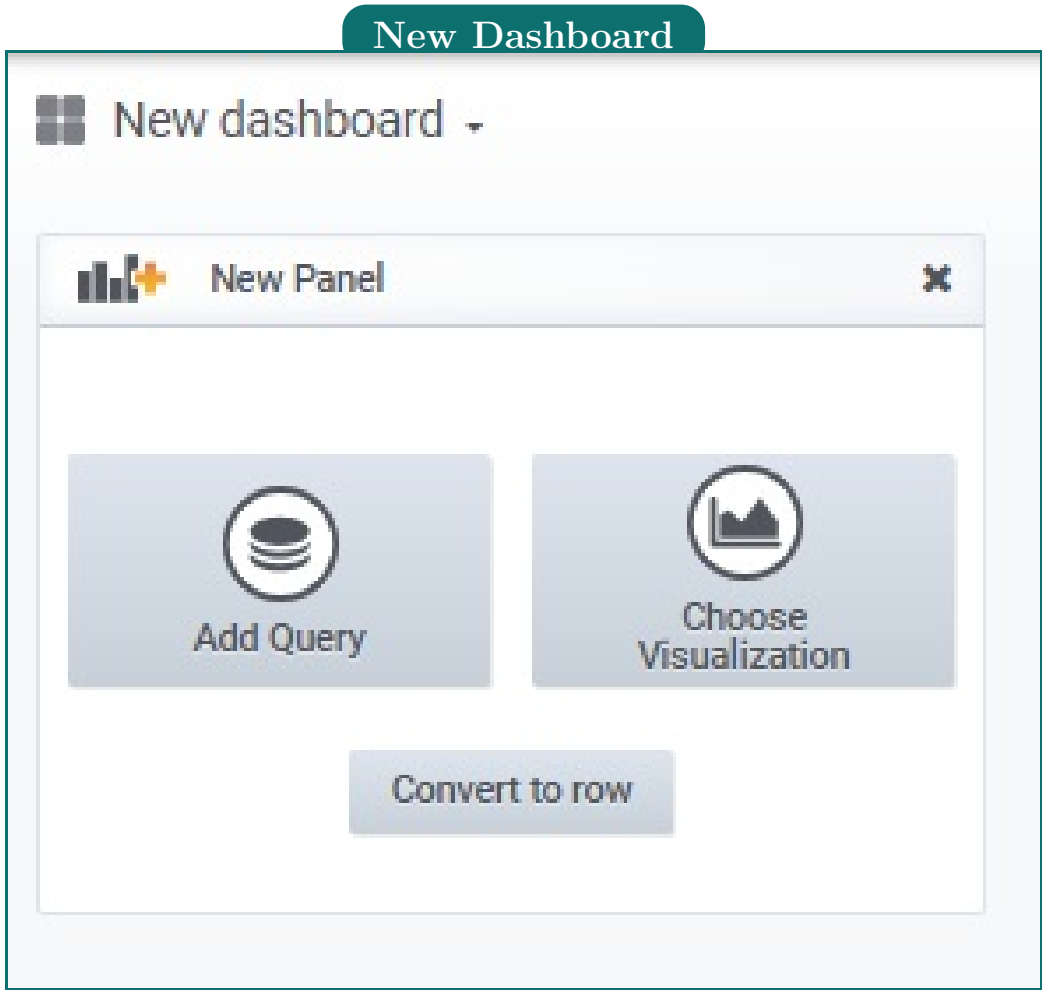
Panel Option:

- **Graph Tooltip**:**Default**. For multi-panel correlation, selectShared crosshair.

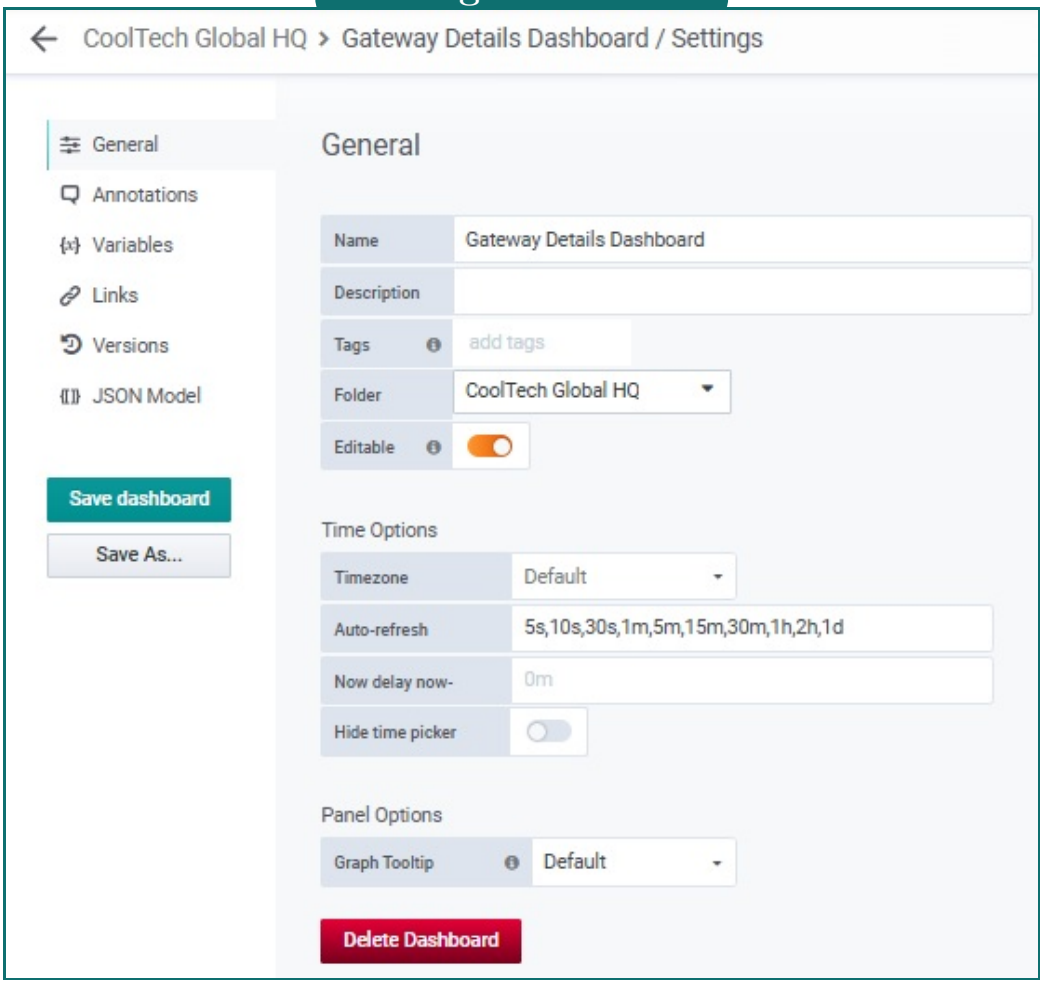
Save

Click **Save dashboard** (left sidebar) and give it a final check. Your empty dashboard is now ready to add panels:

New Dashboard



Setting Dashboard



1.3.1 Dashboard Settings — Advanced Tabs

After configuring the **General** tab (name, folder, time options), use the tabs on the left to add context, parameterization, and lifecycle management to your dashboard.

Annotations

Annotations let you overlay events (e.g., batch start/stop, maintenance) on panels as vertical markers.

Add an Annotation Query

1. Click **Annotations** → **Add Annotation Query**.

2. Choose the data source (usually – *Grafana* – for manual entries, or a time-series source for stored events).

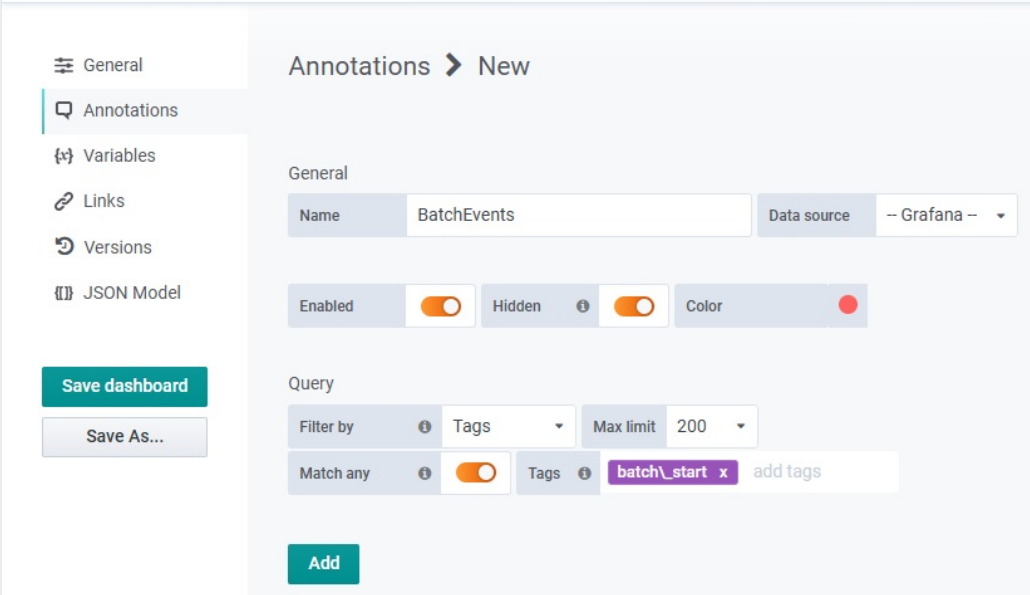
3. Set *Query/Tags* (e.g., `event="batch_start"`).

4. Save the dashboard.

Notes

- Users can add ad-hoc annotations directly on a panel (`Ctrl`/ `Cmd` + Click and drag) when allowed.
- Keep event `tags` short and consistent (e.g., `batch`, `maintenance`).

Annotations



1.3.2 Annotations (Dashboard Markers)

Annotations place time markers (vertical lines with tooltips) on charts to show events such as batch start/end, alarms, or operator actions. They help readers correlate process trends with discrete events.

Prerequisites

- You have created or opened a dashboard, then clicked the **gear** icon (`Dashboard settings`).
- Events exist in the system (from Logbot or your data source) and include either tags or a queryable series.

Open the Annotations tab.

`Dashboards` → `Manage` → `New Dashboard` → **gear** (`Settings`) → `Annotations`

Add an Annotation Query (Tag-based)

Use this option when your events are stored with tags (recommended for most users).

1. Click **Add Annotation Query**.
2. **Name**: Enter a short, clear label (e.g., `BatchEvents`).
3. **Data source**: Select `Grafana` (built–in).
4. **Enabled**: `ON`. **Hidden**: `OFF` (turn `ON` if you want it disabled by default).
5. **Color**: Choose a line color for the marker (e.g., red for `batch_start`).
6. In **Query**:
   - **Filter by**: `Tags`.
   - **Max limit**: Set a sensible cap (e.g., `200`) to avoid clutter.
   - **Match any / all**: Choose `any` to show events that contain *any* of the listed tags; choose `all` to require *every* tag.
   - **Tags**: Type one or more event tags (e.g., `batch_start`, `batch_end`, `alarm_ack`).
7. Click **Add**, then **Save dashboard**.

Add an Annotation Query (Logbot event store)

Use this when events are exposed by the Logbot data source (Graphite–compatible API).

1. Click **Add Annotation Query**.
2. **Name**: e.g., `BatchEvent_Data`.
3. **Data source**: select `logbotData`.
4. **Enabled**: `ON`. **Hidden**: as required. Set a distinctive **Color**.
5. In **Query**, use one of:
   - **Graphite query**: enter the event series path provided by your integration (example pattern only: `events.batch.start`).

- **Graphite events tags**: enter an event tag (e.g., `batch_start`).

6. Click **Add**, then **Save dashboard**.

Good practice (recommended)

- Create two queries with different colors to distinguish `batch_start` and `batch_end`.
- Keep **Max limit** between `100–300` for readability.
- Use consistent naming: short, clear, and machine–safe (letters, numbers, underscores, and hyphens).

How markers appear

On time-series panels, annotation markers render as vertical lines. Hover the marker to see its timestamp and tags. Use the panel menu (`...` on the panel) to quickly hide/show annotations while investigating data.

Troubleshooting

- **No markers visible**: expand the dashboard time range and click **Refresh**.
- **Too many markers**: reduce the time window or lower **Max limit**.
- **Wrong color or overlapping lines**: edit the query and change the **Color**; split complex tags into separate queries.
- **Nothing returns**: verify the correct **Data source**, tag spelling, and that events exist for the selected time range.

Variables

Variables (aka template variables) let viewers filter the same dashboard by plant, gateway, or device without duplicating panels.

Typical Variable Chain

- `$plants` — list of plants
- `$gateways` — gateways constrained by the selected plant
- `$devices` — devices constrained by plant and gateway

Example Definitions

Variable	Type	Definition / Query
`$plants`	Query (label values)	Returns a list of plant names (e.g., `label_values(plant)`).
`$gateways`	Query (chained)	Uses $plants in the filter (e.g., `gateway{plant="$plants"}`).
`$devices`	Query (chained)	Uses both $plants and $gateways in the filter query.

TIPS

- Prefer `Multi-value` for variables like `$devices` when panels can summarize multiple selections.
- In panel queries, replace fixed names with variables (e.g., `{plant="$plants", gateway="$gateways"}`).
- Keep variable names short and lowercase (e.g., `$plants`, `$gateways`, `$devices`).

Links

Dashboard Links place quick navigation buttons below the header.

Add a Link

1. Click **Links** → **Add Dashboard Link**.
2. Set a short *Title* (e.g., `Device Details`).
3. Choose the *Type*: link to another dashboard, an external URL, or a relative path.
4. (Optional) Use variables in the URL to pass the current selection (e.g., `?var-devices=$devices`).
5. Save the dashboard.

Versions

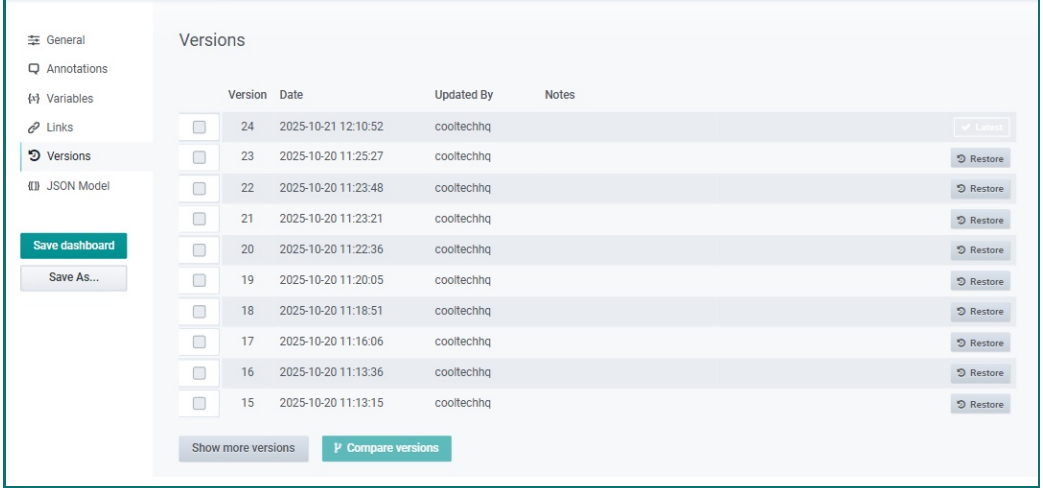
Every save creates a version you can compare or restore.

- Use **Compare versions** before restoring to see what will change.
- Add a short note when you make big edits (e.g., *"Added line KPIs and device selector"*).

Good Practice

Tag stable points (e.g., releases) in **Versions** so teams can roll back confidently.

Version



JSON Model (Advanced)

Shows the dashboard's full JSON definition (panels, variables, layout). Useful for backup or migrating dashboards.

- Click **JSON Model** to *view or copy*; avoid editing directly unless you know the schema.
- For backups, copy the JSON and store it with your project files.

Warning

Editing JSON incorrectly can break the dashboard. Prefer the UI for normal changes; use JSON only for import/export or scripted updates.

1.4 Add Query or Choose Visualization

1.4.1 Add Query

Clicking the **Add Query** button opens the **Query Panel**, where you can define the data series, select metrics, and apply functions. This is the first step in building your panel. After adding queries, you can continue editing the panel Section.

1.4.2 Choose Visualization

The **Choose Visualization** option allows you to select the type of panel representation. Available visualizations include:

- Graphs for time series data
- Tables for structured information
- Single Stat, Gauges, Bar Charts, Pie Charts, etc.

Choosing the correct visualization ensures that the selected query results are displayed in a meaningful way.

Visualization Panel

From the **Home** section, navigate to the ***Latest from the Blog*** panel.

To customize how blog updates are displayed:

1. Click the **Edit** option from the panel menu.
2. In the **Visualization** dropdown, select **News Panel**.
3. Under the **Feed** section, paste the blog feed URL into the **URL** field.

   Example: `https://cors-anywhere.herokuapp.com/https://logbot.cloud/feed`
4. Save the configuration to update the panel.

This setup ensures that the ***Latest from the Blog*** widget retrieves and displays the most recent updates directly on your Home dashboard.

News Panel

Visualization

News Panel

Select Data Source

The first dropdown in the Query editor specifies the data source. Available options may include:

- `default`
- `logbotData`
- `Grafana`
- `Dashboard`
- `Mixed`
- `logbot Json`

With Graph chosen as the visualization, the panel shows the Query section of default builder:

- **Data source**: defaults to *default* (dropdown at the top of the query bar).
- **Series** row: click *select metric* to choose the time series for this panel.
- **Functions** row: click + to add transformations (e.g., aggregate, math).
- **Advanced query controls**: *Add Query*, *Add Expression*, and *Query Inspector* (top right).
- **Timing options**: *Cache timeout*, *Max data points*, *Relative time*, and *Time shift*.

Open Functions Menu In Section Of Default Below the Query row, click on the + button under **Functions.This opens a categorized list of operations that can be applied to your series.**

1- Alias Functions

Alias functions help in renaming, formatting, or adjusting the labels of time series in the graph. This is useful when series names are long or when you want to create readable legends.

- `alias` — Renames a metric series with a given string.
- `aliasByMetric` — Uses the full metric path as alias.
- `aliasByNode` — Uses a selected node in the metric path as alias.

- `aliasByTags` — Generates alias names using tags associated with the series.
- `aliasQuery` — Applies the query string as alias.
- `aliasSub` — Substitutes text in the alias using regex rules.
- `legendValue` — Appends statistical values (e.g., avg, min, max) to legend.
- `lower` — Converts alias name to lowercase.
- `upper` — Converts alias name to uppercase.

2- Calculate Functions

The Calculate functions are used to apply mathematical or statistical operations directly to the time series data. These help in smoothing, forecasting, and deriving additional insights from the metrics.

- `aggregateLine` — Draws a straight line representing the average value of the series.
- `exponentialMovingAverage` — Applies an exponential moving average for smoothing data points.
- `holtWintersAberration` — Detects anomalies based on the Holt-Winters forecasting algorithm.
- `holtWintersConfidenceArea` — Displays confidence intervals for Holt-Winters forecasts as shaded areas.
- `holtWintersConfidenceBands` — Adds upper and lower confidence bounds using Holt-Winters.
- `holtWintersForecast` — Generates predicted future values based on historical data.
- `identity` — Returns the same series, useful for validation or as a placeholder.
- `linearRegression` — Fits a regression line across the series, showing the trend.
- `movingAverage` — Smooths the series by calculating the average within a moving window.
- `movingMax` — Returns the maximum value within a moving window.
- `movingMedian` — Applies a moving median filter to smooth data.
- `movingMin` — Returns the minimum value within a moving window.
- `movingSum` — Calculates the cumulative sum over a moving window.
- `movingWindow` — Generalized function to apply custom operations in a moving window.
- `nPercentile` — Returns the Nth percentile value across the series.
- `stdev` — Computes the standard deviation of the time series.

3- Combine Functions

Combine functions operate on multiple time series simultaneously. They allow aggregation, comparison, and grouping across multiple metrics.

- `aggregate` — Aggregates values from multiple series into one.
- `aggregateSeriesLists` — Aggregates across a list of series.
- `aggregateWithWildcards` — Aggregates while ignoring selected path nodes.
- `applyByNode` — Applies a function by node index in the metric path.
- `asPercent` — Converts values into percentage form.
- `averageSeries` — Returns the average of multiple series.
- `averageSeriesWithWildcards` — Averages series ignoring parts of the path.
- `avg` — Alias for average across series.
- `countSeries` — Returns the number of series.
- `diffSeries` — Calculates differences between series.
- `diffSeriesLists` — Calculates differences across lists of series.
- `divideSeries` — Divides one series by another.
- `divideSeriesLists` — Divides across series lists.
- `group` — Groups multiple series.
- `groupByNode` — Groups by a single node index in the metric path.
- `groupByNodes` — Groups by multiple node indices.
- `groupByTags` — Groups by metric tags.
- `map` — Applies a mapping function.
- `mapSeries` — Similar to map, but works with series lists.
- `maxSeries` — Returns maximum values across series.
- `minSeries` — Returns minimum values across series.
- `multiplySeries` — Multiplies series values together.
- `multiplySeriesLists` — Multiplies across two lists of series.
- `multiplySeriesWithWildcards` — Multiplies while ignoring path wildcards.
- `pct` — Converts to percentage values.
- `percentileOfSeries` — Computes percentile (e.g., 90th) across series.
- `powSeries` — Raises one series to the power of another.
- `rangeOfSeries` — Calculates range (max–min) across series.
- `reduce` — Reduces multiple series into a single series summary.
- `reduceSeries` — Works like reduce but on lists of series.
- `stddevSeries` — Computes standard deviation across series.
- `sum` — Sums values.
- `sumSeries` — Sums across series into one.
- `sumSeriesLists` — Sums across lists of series.
- `sumSeriesWithWildcards` — Sums ignoring path wildcards.

- `weightedAverage` — Computes weighted average across series.

4- Filter Data

Filter Data functions remove unwanted values from a single time series based on thresholds or percentiles.

- `removeAbovePercentile` — Removes values above a defined percentile.
- `removeAboveValue` — Removes values greater than a specified threshold.
- `removeBelowPercentile` — Removes values below a defined percentile.
- `removeBelowValue` — Removes values less than a specified threshold.

5- Filter Series

Filter Series functions refine which entire series are displayed, based on conditions.

- `averageAbove` — Keeps only series whose average is above a threshold.
- `averageBelow` — Keeps only series whose average is below a threshold.
- `averageOutsidePercentile` — Keeps only series outside a percentile range.
- `currentAbove` — Keeps series where the latest value is above a threshold.
- `currentBelow` — Keeps series where the latest value is below a threshold.
- `exclude` — Excludes series that match a pattern.
- `filterSeries` — Filters series based on a regex expression.
- `grep` — Keeps only series that match a regex pattern.
- `highest` — Selects the N series with the highest values.
- `highestAverage` — Selects series with the highest average values.
- `highestCurrent` — Selects series with the highest current values.
- `highestMax` — Selects series with the highest maximum value.
- `limit` — Limits the number of series shown.
- `lowest` — Selects the N series with the lowest values.
- `lowestAverage` — Selects series with the lowest average values.
- `lowestCurrent` — Selects series with the lowest current values.
- `maximumAbove` — Keeps series where the maximum exceeds a threshold.
- `maximumBelow` — Keeps series where the maximum is below a threshold.
- `minimumAbove` — Keeps series where the minimum exceeds a threshold.
- `minimumBelow` — Keeps series where the minimum is below a threshold.
- `mostDeviant` — Selects series with the highest standard deviation.
- `removeBetweenPercentile` — Removes series within a percentile band.
- `removeEmptySeries` — Removes series with no data.
- `unique` — Keeps only unique series by name.
- `useSeriesAbove` — Keeps series with values above a threshold.

6- Sorting Functions

Sorting functions reorder time series based on values, totals, or names.

- `sortBy` — Sorts series by average values.
- `sortByMaxima` — Sorts by maximum value in each series.
- `sortByMinima` — Sorts by minimum value in each series.
- `sortByName` — Sorts alphabetically by series name.
- `sortByTotal` — Sorts by the total sum of values in each series.

7- Special Functions

Special functions provide utility transformations and visual adjustments.

- `cactiStyle` — Formats graphs in Cacti-style visualization.
- `changed` — Returns only data points that have changed value.
- `consolidateBy` — Adjusts how multiple points are consolidated (avg, sum, etc.).
- `constantLine` — Adds a horizontal constant line to the graph.
- `cumulative` — Returns a running cumulative sum of the series.
- `events` — Plots external events on the graph.
- `fallbackSeries` — Provides a fallback series if the main series is missing.
- `randomWalk` — Generates random walk data (for testing).
- `randomWalkFunction` — Function version of random walk.
- `seriesByTag` — Selects series using metadata tags.
- `setXFilesFactor` — Adjusts tolerance for missing data points.
- `substr` — Extracts substrings from series names.
- `xFilesFactor` — Sets how missing values are handled.

8 - Transform Functions

Transform functions perform mathematical or functional transformations on the series.

- `absolute` — Returns absolute values.
- `add` — Adds a constant value to the series.
- `bitwiseAnd` — Performs bitwise AND on series values.
- `compressPeriodicGaps` — Compresses gaps in periodic data.
- `delay` — Shifts the series forward in time.
- `derivative` — Returns the derivative (rate of change).
- `exp` — Applies the exponential function.
- `hitcount` — Converts event counts into rates.
- `integral` — Computes the integral (cumulative sum).
- `integralByInterval` — Computes integrals over defined intervals.

- `interpolate` — Fills missing values by interpolation.
- `invert` — Inverts series values (1/x).
- `isNonNull` — Returns 1 when value is not null, else 0.
- `keepLastValue` — Replaces nulls with last known value.
- `log` — Applies natural logarithm.
- `logit` — Applies logit transformation.
- `minMax` — Normalizes values between min and max.
- `nonNegativeDerivative` — Like derivative, but ignores negative drops.
- `offset` — Shifts values by a constant.
- `offsetToZero` — Shifts series to start at zero.
- `perSecond` — Converts values into per-second rates.
- `pow` — Raises values to a power.
- `round` — Rounds values to nearest integer.
- `scale` — Multiplies values by a constant.
- `scaleToSeconds` — Normalizes values per second.
- `sigmoid` — Applies sigmoid function.
- `sin` — Applies sine function.
- `sinFunction` — Generates synthetic sine wave.
- `smartSummarize` — Summarizes values intelligently by time interval.
- `squareRoot` — Applies square root function.
- `summarize` — Aggregates data into larger time buckets.
- `time` — Returns time index values.
- `timeFunction` — Applies a function relative to time.
- `timeShift` — Shifts series by a time offset.
- `timeSlice` — Slices series into a specified time window.
- `timeStack` — Stacks multiple series over time intervals.
- `transformNull` — Replaces null values with a chosen constant.

Monitoring:

The Graph visualization displays the output of the selected queries and metrics over time. Each metric is shown as a colored line, bar, or point depending on the chosen draw mode, making it possible to compare multiple signals in a single chart.

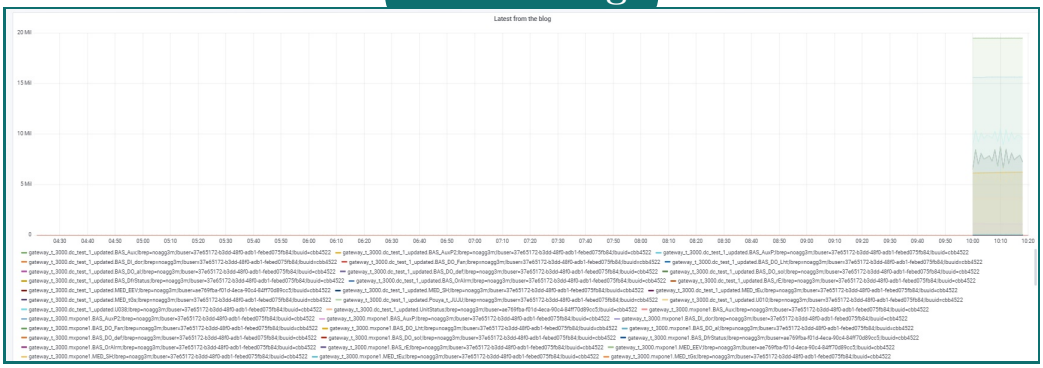
- **X-Axis:** Represents the time dimension, showing how the data evolves continuously.
- **Y-Axis:** Represents the values of the selected metrics. Multiple Y-axes can be used when different units or scales are required.
- **Legend:** Displays all active metrics with their corresponding colors, names, and current values, allowing users to quickly identify which line or series belongs to each data source.

- **Data Points:** Each plotted value reflects the real-time or historical data pulled from the query. Hovering over the graph provides detailed tooltips with exact values and timestamps.
- **Thresholds and Regions:** Highlighted colored zones or horizontal lines indicate warning or critical levels, making it easier to detect anomalies or out-of-range values.

This visualization provides an intuitive way to monitor system performance, track changes over time, and detect unusual patterns across different metrics.

Visualization

Monitoring



1.5 Visualization Section:

The Visualization panel Visualization allows users to control how the graph is displayed, including drawing styles, tooltips, stacking behavior, and handling of null values.

1.5.1 Graph Visualization

From the dashboard editor, select **Graph** from the visualization menu. This opens the main query editor where you can define metrics and apply functions. When configuring a Graph panel in Logbot, the parameters you set depend on your dataset and how you want the graph to display information. The following explains each option:

- Navigate to a dashboard.
- Click **Add Panel** → **Visualization** → **Graph**.

Section of Draw Modes

- **Bars**: Displays the data as vertical bars.
- **Lines**: Connects data points with lines.
- **Points**: Shows individual data points.
- **Staircase Line**: Draws steps instead of smooth lines.
- **Legend** : Displays all active metrics with their corresponding colors, names, and current values, allowing users to quickly identify which line or series belongs to each data source.
- **Dashes** : changes line style (visual distinction).
- **Hidden series** :completely hides the series from the chart view but still keeps it in the query.

Multiple modes can be enabled at the same time (e.g.,lines and points together).

Mode Options

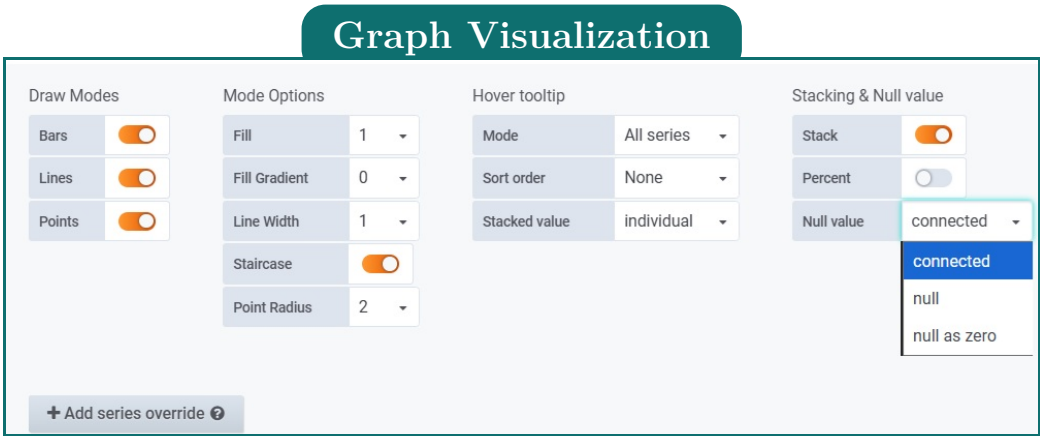
- **Fill**: Controls the filled area under a line (0 = no fill, 1–10 = increasing intensity).
- **Fill Gradient**: Applies a gradient fill under the line.
- **Line Width**: Adjusts the thickness of lines (0–10).
- **Staircase**: Converts the line into a step-like style.
- **Point Radius**: Defines the size of data points (1-5).

Hover Tooltip

- **Mode**: Show values for all series or only a single series.
- **Sort Order**: Determines order of values (None, Increasing, Decreasing).
- **Stacked Value**: Display stacked values as *individual* or *cumulative*.

Stacking and Null Value

- **Stack**: Enables stacking of multiple series.
- **Percent**: Shows stacked values as a percentage of the total.
- **Null Value**: Defines how missing data is treated:
  - Null – leaves gaps in the chart.
  - Connected – connects available values.
  - Null as zero – replaces missing values with zero.



Series Override (Advanced Visualization)

Series override provides fine-grained control for individual data series, allowing users to adjust how each series is displayed, compared, or emphasized in the visualization.

Draw Modes (True/False). These options define how a series is drawn on the graph:

- **Bars**: Displays series as vertical bars. Useful for histogram-like or comparative views.
- **Lines**: Displays data as connected lines. Best for trends over time.
- **Points**: Displays individual data points. Suitable for sparse datasets.
- **Staircase line**: Shows data as steps, holding the previous value until the next update.
- **Hidden Series**: Temporarily hides the series from the graph without deleting it.
- **Dashes**: Renders lines as dashed, helping differentiate overlapping series.
- **Legend**: Toggles whether the series appears in the legend.
- **Hide in tooltip**: Excludes the series from hover tooltips to reduce clutter.

Stacking. Stacking determines how multiple series overlay each other vertically:

- **True/False**: If true, values are stacked (summed). If false, series are drawn independently.
- **Groups (A–D)**: Assign series to stack groups, e.g., group A for traffic in/out, group B for CPU usage.

Axis Assignment. Assigns a series to a specific Y-axis:

- **1**: Assigns to the left Y-axis (default).
- **2**: Assigns to the right Y-axis. Useful for comparing values with different scales.

Z-Index. Controls the drawing order of overlapping series:
- Range: **-3 to +3**.
- Negative values push the series into the background.
- Positive values bring the series to the foreground.

Transformations. Apply transformations to adjust how the series is displayed:
- **Constant**: Shifts the series by a fixed constant value.
- **Negative-Y**: Flips the series along the Y-axis for mirrored comparison.

Null Point Mode. Controls the treatment of missing data points:
- **Connected**: Draws a continuous line across null values.
- **Null**: Leaves visible gaps in the series where data is missing.
- **Null as zero**: Interprets missing data as zero.

Point Radius. Controls the size of the points:
- Range: **1 to 5**.
- Larger values highlight individual points, smaller values make them subtle.

Other Numeric Options. These options fine-tune line and fill properties:
- **Line Fill (0–10)**: Controls how much of the area under the line is filled.
- **Fill Gradient (0–10)**: Adds gradient shading to the filled area.
- **Line Width (0–10)**: Adjusts the thickness of the line.

Section of Axes

Left Y Configure the primary vertical axis for values.
- **Show** (toggle): show/hide the axis.
- **Unit** (dropdown): common choices include `none`, `short`, `percent (0-100)`, `percent (0.0-1.0)`, `Humidity (%H)`, `decibel`, `hexadecimal (0x)`, `hexadecimal`.
- **Scale** (dropdown): `linear`, `log (base 2)`, `log (base 10)`, `log (base 32)`, `log (base 1024)`.
- **Y-Min/Y-Max**: numeric limits; `auto` lets Logbot pick optimal bounds.
- **Decimals**: fixed number of decimals or `auto`.
- **Label**: free text shown on the axis.

Right Y Configure a secondary vertical axis (for series assigned to Y=2 in overrides).

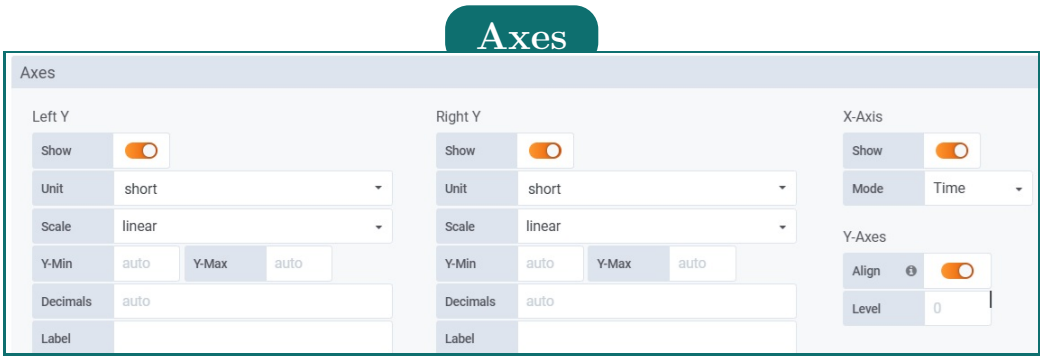
- Same options as Left Y: **Show**, **Unit**, **Scale**, **Y-Min**/**Y-Max**, **Decimals**, **Label**.

X-Axis

- **Show** (toggle): show/hide the axis.
- **Mode**: `Time` (time series), `Series` (categorical series names), `Histogram` (bucketed counts).

Y-Axes (global alignment)

- **Align** (toggle): align left and right Y axes to a shared zero or range.
- **Level**: alignment level (integer), used when alignment is enabled.



Section Of Legend

Options

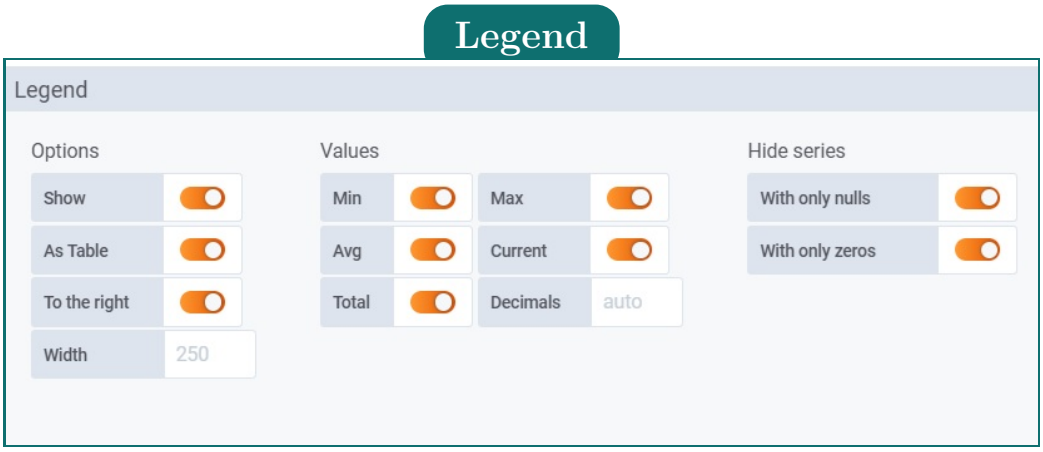
- **Show** (toggle): display/hide the legend.
- **As Table** (toggle): render legend as a table with columns.
- **To the right** (toggle): place legend at the right side instead of below.
- **Width**: pixel width when the legend is on the right.

Values Choose which statistics appear in the legend (toggles).

- **Min**, **Max**, **Avg**, **Current**, **Total**.
- **Decimals**: fixed number of decimals or `auto`.

Hide series

- **With only nulls** (toggle): hide series composed entirely of null values.
- **With only zeros** (toggle): hide series composed entirely of zeros.



Sections Of Thresholds & Time Regions

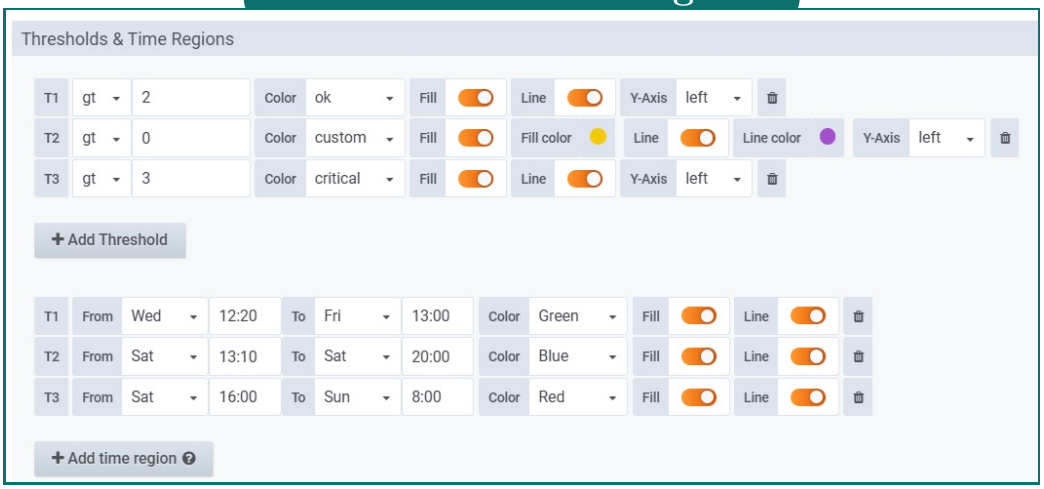
Thresholds Add one or more threshold rules that style values crossing a condition.

- **Add Threshold**: inserts a new rule row.
- **Operator**: `gt` (greater than) or `lt` (less than).
- **Value**: numeric trigger level.
- **Color**: preset `ok`, `warning`, `critical` or `custom`.
- **Fill** (toggle): fill background in the region where the rule matches.
- **Line** (toggle): draw a horizontal threshold line.
- **Y-Axis**: `left` or `right` target axis for the line/area.
- **Delete** (trash icon): remove the rule.

Time Regions Highlight specific time windows regardless of values.

- **Add time region**: inserts a new region.
- **From/To**: time bounds (`hh:mm`) or `Any` for open ends.
- **Color**: region color.
- **Fill** (toggle): fill the background in the region.
- **Line** (toggle): draw boundary lines at region edges.

Thresholds Time Regions



Sections Of Data Links

Data links provide interactive navigation by associating parts of a visualization with external or internal resources.

Definition.

- Each data link consists of a **Title** (displayed text) and a **URL**.
- URLs can reference dashboards, external websites, or specific data resources.

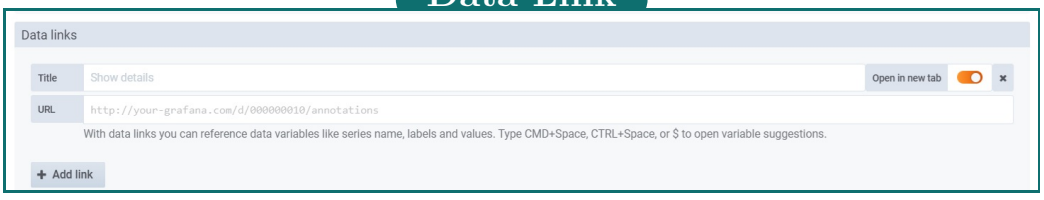
Dynamic Variables.

- Data links support variables such as `$series_name`, `$label`, or `$value`.
- These variables automatically substitute real values during user interaction.

Usage.

- Clicking on a data link opens the referenced resource.
- Links can be configured to open in a **new tab** or within the same application.

Data Link



TIPS:

Always match your Y-axis ranges and thresholds with the actual data range. If unsure, leave `Y-Min` and `Y-Max` as `auto`, so Logbot adjusts dynamically.

General Section

The **General** Panel provides configuration for the panel's overall settings.

Title.

- Allows setting a custom panel title.
- The title is displayed at the top of the panel.

Transparent.

- Toggle **on/off** to make the panel background transparent.
- Useful when layering over dashboards with custom backgrounds.

Description.

- Field for entering a text description.
- Supports markdown and hyperlinks.
- The description is visible as a tooltip when hovering over the panel.

Repeating.

- Allows the panel to repeat for different variable values.
- Options:
  - **Disabled** – no repetition.
  - **Variable-based** – panel repeats dynamically for each selected variable value.
- Note: variable configuration may need to be adjusted to see this feature in action.

Panel Links.

- Add hyperlinks to external dashboards, documentation, or related resources.
- Options:
  - **Title** – display name of the link.
  - **URL** – target address (can reference variables like series name, labels, or values).
  - **Open in new tab** – toggle to open the link in a new browser tab.

Alert Section

The **Alert** feature allows users to create rules that monitor data and trigger notifications when certain conditions are met. This ensures that issues or anomalies are detected and reported in real time.

Creating a New Alert

1. From the panel editor, scroll down to the **Alert** tab.
2. Click on **Create Alert**. If no alert rule is defined, this option will be shown by default.
3. Provide a **Name** for the alert (e.g., *Latest from the Blog alert*).
4. Configure the evaluation rule:
   - **Evaluate every**: defines how often the condition is checked (e.g., every 1 minute).
   - **For**: specifies how long the condition must hold true before an alert is triggered.
5. Add **Conditions**:
   - Select the query (e.g., Query A).
   - Choose the function (e.g., `avg()`).
   - Set the operator (e.g., `IS ABOVE`).
   - Define the threshold value.

Error Handling

- **If no data or all values are null**: choose how the alert should behave (`No Data`, `Alerting`, or `Keep Last State`).
- **If execution error or timeout**: set the state to `Alerting` or `Keep Last State`.

Allerts

Notifications

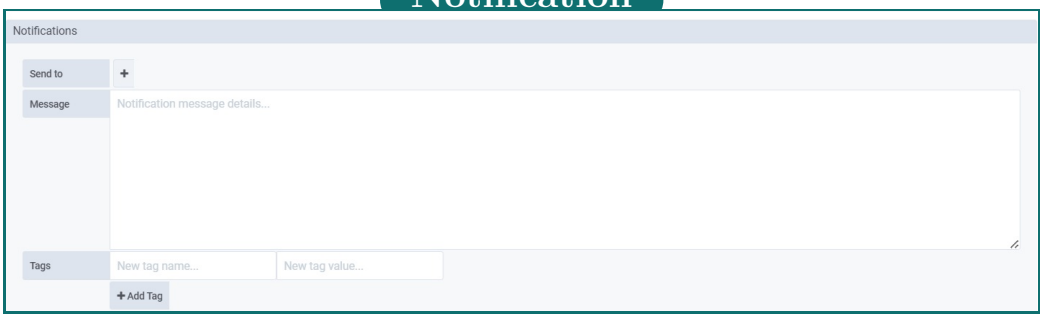
- Define **Send to**: the notification channel (e.g., Email, Slack).
- Add a **Message**: custom notification text.
- Optionally, add **Tags** to classify alerts.

Testing and Management

- Use **Test Rule** to simulate and verify the alert logic.
- **State History** displays the past states of the alert (OK, Alerting, No Data).
- Alerts can be deleted at any time using the **Delete** button.

Note: Alerts are only available on time-series visualizations (e.g., Graph panel). Ensure that queries are correctly configured before setting alert rules.

Notification



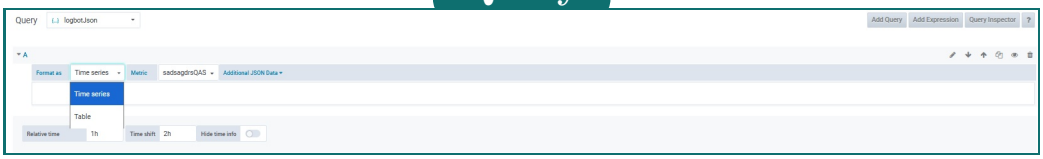
Logbot JSON Data

This view shows the Graph panel *after* a query is created. Use it to choose the data source (`logbot.json`), pick the metric/series to plot, and set the time window. As you change the query, the chart above updates immediately, so you can verify the result and adjust the visualization settings with instant feedback.

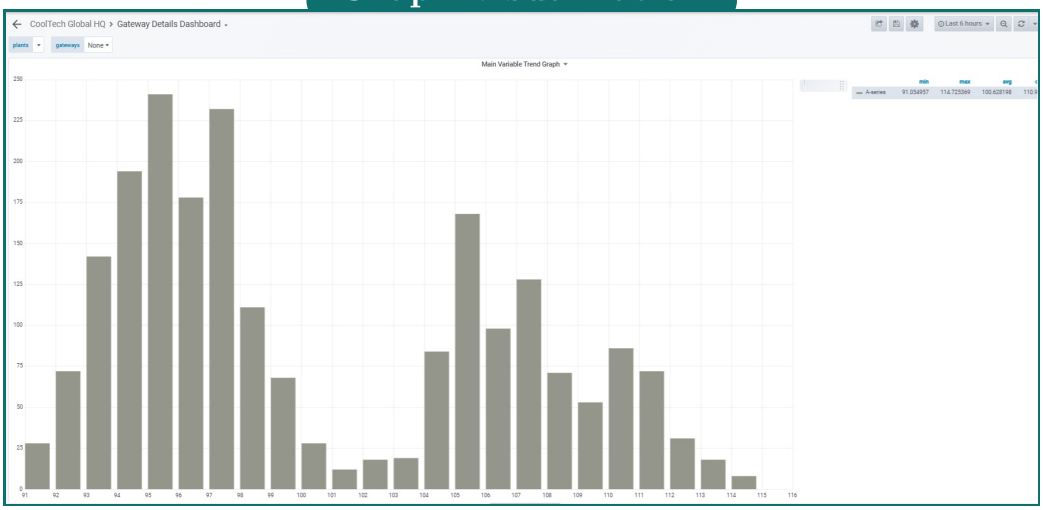
- **Graph Area (Top):** The large empty grid represents the time-series graph. The horizontal axis (**X-axis**) shows the time dimension, while the vertical axis (**Y-axis**) shows the values of the selected metrics. If the query returns data, the graph will display real-time or historical trends.
- **Panel Title:** At the top of the graph, the panel title (*Latest from the blog*) identifies the panel. Users can rename this title to reflect the type of data being visualized.
- **Query Section (Bottom):** The query editor is where users define what data to display. In this example:
  - The **Query Source** is set to `logbot.json`.
  - The query format is defined as **Time series**.
  - The selected **Metric** is shown in the drop-down (placeholder value in the screenshot).
  - Additional options like **Additional JSON Data** allow advanced filtering or custom query details.
- **Time Controls:** Below the query box, users can fine-tune the displayed time range:
  - **Relative Time:** Limits the graph to a chosen interval (e.g., last 1 hour).
  - **Time Shift:** Allows offsetting the data for comparison (e.g., shifting data by 2 hours).
  - **Hide Time Info:** Removes the time overlay for cleaner visuals.
- **Action Buttons:** On the right side, buttons allow users to:
  - **Add Query:** Insert an additional data series for comparison.
  - **Add Expression:** Create calculated fields or derived metrics.
  - **Query Inspector:** Debug or inspect the raw data response.

This setup provides users with full control over how data is retrieved and visualized, enabling customized dashboards that match specific monitoring needs.

Query



Graph visualization



1.6 Stat Visualization

1.6.1 Stat Panel

The **Stat** panel shows a single, large value (optionally with a small trend sparkline in the background). It is ideal for KPIs such as "Temperature (°C) now", "Alarms count", or "Uptime (%)".

- **Big value area (top)** — The large green number is the most recent value from your query. If the query returns a time series, the latest point is used.
- **Trend band (optional)** — A faint filled line behind the value (sparkline) shows recent history for context.
- **Query editor (bottom)** — Where you choose the data source and define the query that feeds the Stat panel.

1) Choose a Data Source

In the **Query** dropdown you can select where the panel pulls data from:

- `default`

- **logbotData**: Query builder with field "chips" for Logbot streams.
- **Grafana**: Built-in sample series (e.g., Test data: random walk) for quick checks.
- **Dashboard** : Re-use results from another panel or the entire dashboard (no external call)
- **Mixed** : Combine multiple sources in one panel (the editor shows a note if a source has no custom editor).
- **logbot Json** : Query using the JSON endpoint; supports Format as Time series or Table and a Metric dropdown.

2) Build the Query

A. Using logbotData

- Click **Series** and add filters as chips. Example: `lbuser` → `cbb4522` **AND** `name` → *your_tag*.
- Use the + button to add more conditions or groups as needed.
- Optional controls:
  - **Relative time** (e.g., *1h*) — Query only recent history for faster panels.
  - **Time shift** (e.g., *1h*) — Compare with the same window in the past.
  - **Hide time info** — Hides the time range note in the panel header.

B. Using `logbotJson`

- Set **Format as**: *Time series* (for Stat/sparkline) or *Table*.
- Choose a **Metric** from the dropdown (e.g., `pit_test_1v`, `test_june_10_0`, ...).
- Use **Additional JSON Data** for advanced filters if required.
- Adjust **Relative time**, **Time shift**, and **Hide time info** as above.

C. Quick check with `Grafana test data`

- Pick *Grafana* → *Test data: random walk*. You should immediately see a green value and a pale history band. This is useful to confirm the panel itself is fine.

D. Re-use results via - `Dashboard` -

- Select – **Dashboard** – and pick *Panel* or *Dashboards* from the menu to feed the Stat from an existing result in the same dashboard (no extra calls).

E. Mixed source

- Choose – **Mixed** – to add multiple queries from different sources. If a source has no special editor, you'll see: *"Data source plugin does not export any Query Editor component."*

3) Actions and Tools

- **Add Query** — Start another query (B-series, C-series, . . .) for comparisons or mixed sources.
- **Add Expression** — Create arithmetic across series (e.g., A - B, AVG(A,B)).
- **Query Inspector** — See raw requests/responses and timing to diagnose issues.

4) "No data" — What it means & how to fix

- The large green **"No data"** indicates your query returned no points in the selected time range.
- **Fixes**:
  - Widen **Relative time** (e.g., from *1h* to *24h*).
  - Verify **lbuser**, **name**, and **Metric** values.
  - Try **Grafana test data** to confirm the panel renders.
  - Open **Query Inspector** for errors or empty responses.

TIPS:

- Pick *Time series* when using the Stat sparkline; pick *Table* only if you need tabular output.
- If unsure about scales, leave axes implicit—Stat uses the latest point; history is only contextual.
- Start with a simple filter (e.g., only `lbuser`) to confirm data flows, then refine with `name`/metric.

Data Displayed

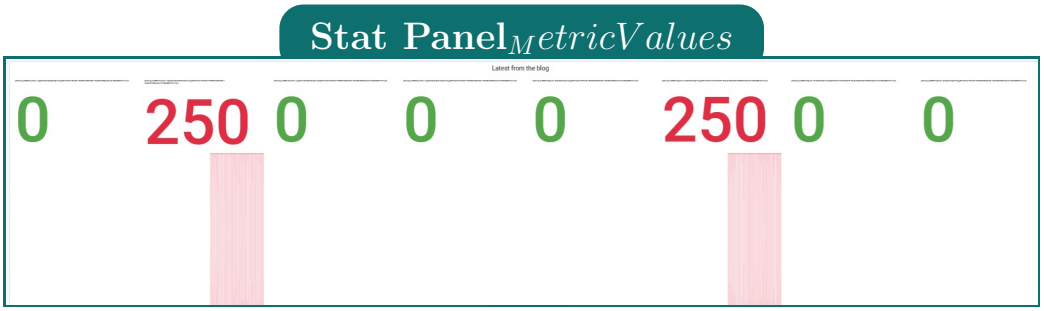
When the **query condition** `main = yes` matches available data, the visualization panel displays the latest values of each series. **Display Behavior:**

- Each series is shown as a large number in the panel.
- The value represents the most recent data point within the selected time range.
- Colors indicate status:

- Green = Normal values (e.g., 0).
- Red = Critical or abnormal values (e.g., 250).

Example:

- A metric value of `0` is displayed in green (normal).
- A metric value of `250` is displayed in red (warning/critical).



TIPS:

Each number corresponds to a series label shown above the value, helping the user identify which metric or device the reading belongs to.

Configuring Queries with Main Filter

This subsection explains how to configure a query using the `Main = Yes` filter. By applying this condition, only the records marked as `Main` will be retrieved and displayed in the visualization panel.

This is especially useful when focusing on primary metrics or key signals, while ignoring non-essential data.

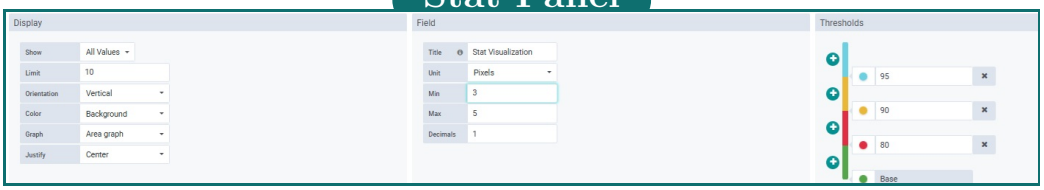
Why "No Data" may appear:

- No records in the dataset satisfy the condition `main = yes`.
- The selected time range (e.g., Last 1h) does not include any matching data.
- Case sensitivity: `yes` differs from `Yes`.

Settings:

- `Cache timeout:` Refresh interval, does not generate new data.
- `Max data points:` Controls resolution of the chart.
- `Relative time:` Restricts to recent time window.
- `Time shift:` Moves query window back in time for comparison.

Stat Panel



TIPS:

If no data is shown, first extend the time range (e.g., to 24h), then verify the query condition.

Display Settings (Stat Visualization)

The `Display` section controls how values are computed, arranged, and styled in the Stat panel.

- **Show:**
  - `Calculation` — render a single computed value (e.g., `First`, `Last`, `Min`, `Max`, `Average`).
  - `All Values` — render every returned value from the query.
- **Calc (when Show = `Calculation`):** choose the aggregation method such as `First`, `Last`, `Min`, `Max`, or `Average`.
- **Limit:** maximum number of values to render (useful with `All Values`).
- **Orientation:** `Horizontal`, `Vertical`, or `Auto` layout for the rendered values.
- **Color:** choose whether threshold colors apply to the `Value` (text) or the `Background`.
- **Graph:** optional tiny trend, e.g., `Area graph`, behind each value.
- **Justify:** alignment of the content: `Auto`, `Left`, or `Center`.

Field Settings (Stat Visualization)

The **Field** section defines how the selected metric values are displayed in the Stat Visualization panel:

- **Title**: Custom label for the field (e.g., *CPU Usage*, *Temperature*).
- **Unit**: Measurement unit for the value (e.g., percent, pixels, ms, decibel).
- **Min / Max**: Define the lower and upper limits for the displayed data range.
- **Decimals**: Controls the number of decimal places shown (e.g., 2 = 25.36).

These options ensure the raw data is formatted clearly, making the visualization more meaningful and easy to interpret.

Field: Unit Options

In the **Field** section of the Stat Panel, the **Unit** option allows you to define how your data values are displayed. By choosing an appropriate unit, numbers are automatically formatted with correct scaling and symbols. This makes the visualization clearer and easier to interpret.

- **none** — Show the number with no unit (e.g., `42`).
- **short** — Compact abbreviations for large/small numbers (e.g., `12.3k`, `4.5M`).
- **percent (0–100)** — Treats `1` as `1%` and `80` as `80%`. Use when your data already spans $[0, 100]$.
- **percent (0.0–1.0)** — Treats `0.8` as `80%`. Use when your data is fractional.
- **Humidity (%H)** — Displays humidity as a percent with %H suffix (e.g., `55%H`).
- **decibel (dB)** — Formats acoustic/electrical level values (e.g., `72 dB`).
- **scientific notation** — Exponential format for very large/small values (e.g., `1.23e-6`).
- **hexadecimal (0x)** — Unsigned integer shown in hex with `0x` prefix (e.g., `0x7F`).
- **hexadecimal** — Hex without prefix (e.g., `7F`).
- **locale format** — Adds thousands separators per locale (e.g., `1,234,567`).
- **Pixels** — UI/graphics counts (e.g., `250 px`).
- **Square Meters (m^2)** — Area values (e.g., `250.00` m^2).
- **Acceleration / Angle / Area / Concentration / Currency / Data (IEC/Metric)** — Domain presets that append the appropriate unit symbol and scale (e.g., `9.81` m/s^2, `45°`, `USD 120.00`, `128 MiB`, `128 MB`).

TIPS:

Pick the unit that matches your *stored* data (e.g., use *percent (0.0–1.0)* for fractions like `0.73`).
Use **Min**, **Max**, and **Decimals** (in the same *Field* box) to clamp the display range and control precision. Changing **Unit** only affects formatting; calculations still use the raw numeric values.

- **Acceleration:** meters/sec^2, feet/sec^2, g unit.
- **Angle:** degrees (°), radians, gradian, arc minutes, arc seconds.
- **Area:** square meters (m^2), square feet (ft^2), square miles (mi^2).
- **Computation:** FLOP/s, MFLOP/s, GFLOP/s, up to YFLOP/s.
- **Concentration:** ppm, ppb, mg/m^3, µg/m^3, mmol/L, etc.
- **Currency:** Dollar ($), Euro (€), Pound (£), Yen (¥), Bitcoin (BTC), and many others.
- **Data (IEC):** bits, bytes, KiB, MiB, GiB, TiB, PiB.
- **Data (Metric):** bits, bytes, kB, MB, GB, TB, PB.
- **Data Rate:** bits/sec, bytes/sec, kB/s, MB/s, GB/s.

TIPS:

Selecting the correct unit ensures that your Stat panel values are consistent and readable.

For example:

energy consumption might use kWh, network traffic could use MB/s, and environmental sensors might use ppm or mg/m^3. The **Unit** setting in the **Field** section allows users to define the measurement unit of the displayed data. This ensures that values are properly formatted and meaningful for the context. The available units are organized into categories:

- **Date & Time:** YYYY-MM-DD HH:mm:ss, MM/DD/YYYY hh:mm a, "From Now".
- **Energy:** watt (W), kilowatt (kW), megawatt (MW), gigawatt (GW), watt-hour (Wh), kilowatt-hour (kWh), ampere-hour (Ah).
- **Electrical:** volt (V), kilovolt (kV), millivolt (mV), ohm (), kilo-ohm (k), mega-ohm (M).
- **Capacitance/Inductance:** farad (F), microfarad (µF), nanofarad (nF), henry (H), millihenry (mH), microhenry (µH).
- **Flow:** gallons/min (gpm), cubic meters/sec (cms), liters/min (L/min), cubic feet/sec (cfs).

- **Force:** newtons (N), kilonewtons (kN), newton-meters (Nm).
- **Hash Rate:** hashes/sec, kilohashes/sec, MH/s, GH/s, TH/s, PH/s, EH/s.
- **Light/Illumination:** lumens (Lm), lux (lx), candela (cd).
- **Length:** meters (m), centimeters (cm), millimeters (mm), kilometers (km), miles (mi), feet (ft), inches (in).
- **Mass:** grams (g), kilograms (kg), milligrams (mg), tonnes (t), pounds (lb).
- **Pressure:** pascal (Pa), kilopascal (kPa), bar, psi.
- **Temperature:** Celsius (°C), Fahrenheit (°F), Kelvin (K).
- **Velocity:** meters/sec, kilometers/hour (km/h), miles/hour (mph), knots.
- **Radiation:** Becquerel (Bq), Curie (Ci), Gray (Gy), rad, Sievert (Sv), millisievert (mSv), microsievert (µSv), rem.
- **Time:** Hertz (Hz), nanoseconds (ns), microseconds (µs), milliseconds (ms), seconds (s), minutes (min), hours (h), days (d).
- **Throughput:** counts/sec (cps), operations/sec (ops), requests/sec (rps), reads/sec (rps), writes/sec (wps), I/O ops/sec (iops), counts/min (cpm), operations/min (opm).
- **Velocity:** meters/second (m/s), kilometers/hour (km/h), miles/hour (mph), knots (kn).
- **Volume:** milliliter (mL), liter (L), cubic meter (m^3), normal cubic meter, cubic decimeter, gallons.

TIPS:

Always choose a unit that matches the real-world meaning of the metric. For example, select mg/m^3 for air quality, kWh for energy consumption, or psi for pressure.

Thresholds

The **Thresholds** configuration defines how numerical values are translated into visual color indicators. Thresholds help users quickly identify whether a value is in a normal, warning, or critical state.

- **Adding Thresholds:** Use the icon to add new thresholds. Each threshold is represented by a numerical limit and a color assignment.
- **Order of Thresholds:** Threshold values are evaluated from top to bottom. The **minimum number (lowest threshold)** is always placed at the bottom of the list and acts as the baseline state (often displayed in **green** as the "Base").
- **Behavior of Minimum Threshold:** When you set a minimum value, this number is automatically positioned under all other thresholds. Any

value below this minimum will adopt the color assigned to the "Base" threshold.

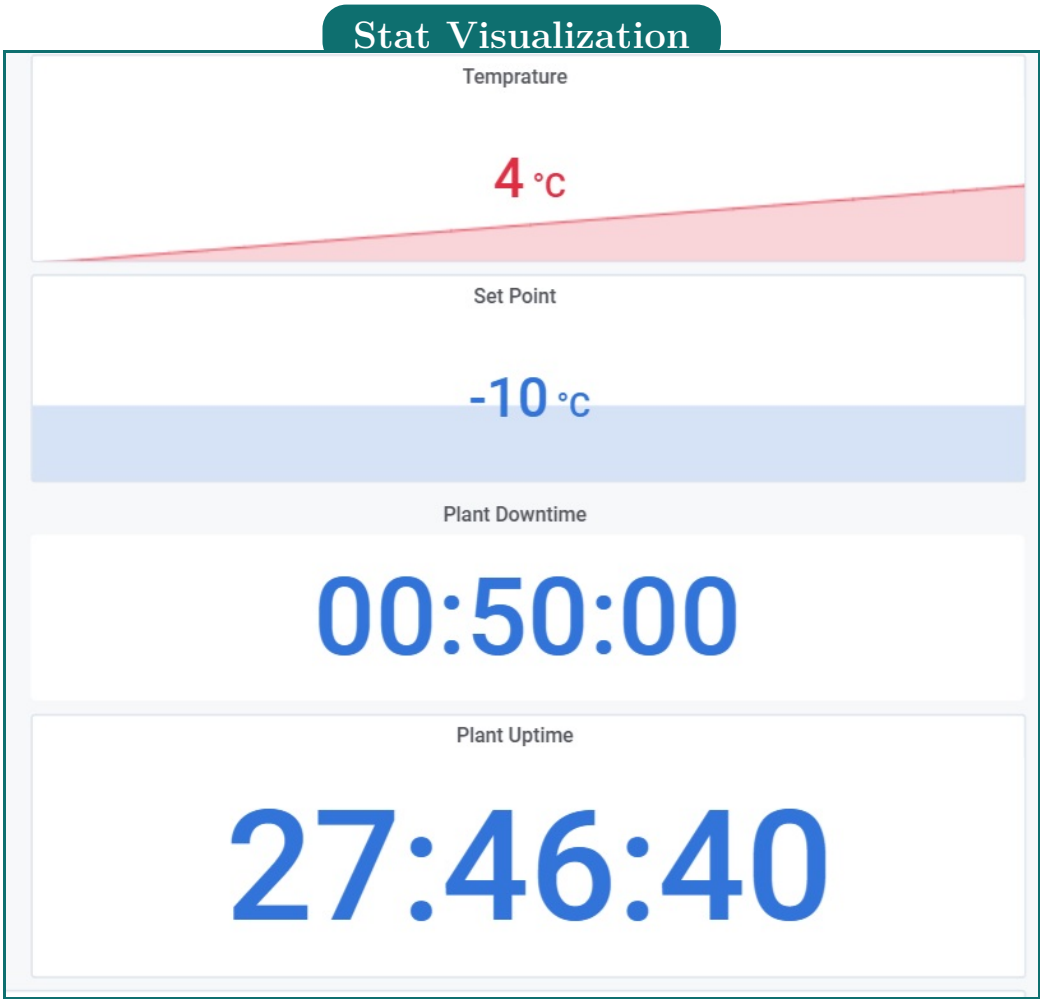
- **Example:** In the configuration shown below:
  - Values $\geq 45 \rightarrow$ Red (critical)
  - Values $\geq 40 \rightarrow$ Blue (warning)
  - Values $\geq 19 \rightarrow$ Yellow (caution)
  - Values below $19 \rightarrow$ Green (base, safe zone)

This structure ensures that the lowest threshold (minimum) anchors the color scale at the bottom, while higher thresholds define escalation levels.

Thresholds

How Thresholds Work

- **Add level:** Click `+` to create a new threshold.
- **Value:** Enter the numeric trigger (e.g., `80`). When the panel value crosses this number, the visual style changes.
- **Color:** Pick a preset (green, yellow, red, blue, orange, purple) or define a *Custom* color.
- **Base:** The default style used when no threshold condition is met.
- **Multiple levels:** Stack thresholds for finer granularity, e.g., `40` (normal) $\rightarrow$ `80` (warning) $\rightarrow$ `85, 90` (critical).



TIPS:

- Order thresholds from lowest to highest for predictable bands.
- Combine with *Min*, *Max*, and *Decimals* in the **Field** section to control scale and precision.
- Use consistent colors across panels (e.g., green = normal, yellow = warning, red = critical).

Value Mappings Section

The **Value Mappings** configuration provides a mechanism to substitute raw numerical output values with human-readable labels or descriptive text. This functionality is particularly valuable when numeric readings alone do

not convey sufficient context, or when end-users require direct interpretation (e.g., `250` → `LOG`).

Mapping Types Two primary mapping strategies are available:

- **Value Mapping:** Associates a single, discrete numeric value with a specified label. Example: Mapping `250` to `LOG` ensures that every occurrence of the value `250` is displayed as `LOG`.
- **Range Mapping:** Associates a continuous interval of values with a specified label. Example: Defining a range of `20-240` and assigning it the label `LOG` ensures that all values within this interval are replaced with `LOG`.

Practical Application When applied, value mappings transform how data is presented in the Stat panel:

- Numeric data is replaced by text that carries operational meaning.
- Users no longer need to interpret thresholds manually; the visualization conveys state directly.
- Complex datasets can be abstracted into simplified, high-level categories.

Example: Mapping Raw Values to Labels

- **Discrete Mapping:** Input: `250` → Displayed as `LOG`.
- **Range Mapping:** Input: Any value between `20-240` → Displayed as `LOG`.
- **Result in Panel:** Instead of showing raw numbers (e.g., `250`), the Stat panel will display the label `LOG`, allowing faster recognition of status without numeric interpretation.

Recommendation

It is recommended to apply value mappings in scenarios where:

- Thresholds or categorical states are more meaningful than exact numbers.
- User interpretation must be simplified for decision-making.
- Data is to be standardized across multiple dashboards for consistency.

General Settings

After defining thresholds for data visualization, additional configuration in the **General** and **Panel Links** sections ensures that the panel is well structured and integrated within the dashboard.

The General tab provides options to define the identity and layout of the panel:

- **Title:** Assign a clear name (e.g., "Log") so the panel can be easily recognized.
- **Description:** Panel Description, and also Supports Markdown& Links
- **Transparent:** Enable to blend the panel with the dashboard background.
- **Repeating:** When template variables are defined, this option allows the panel to repeat dynamically for each variable instance. This is useful in monitoring scenarios with multiple devices or metrics.

Panel Links:

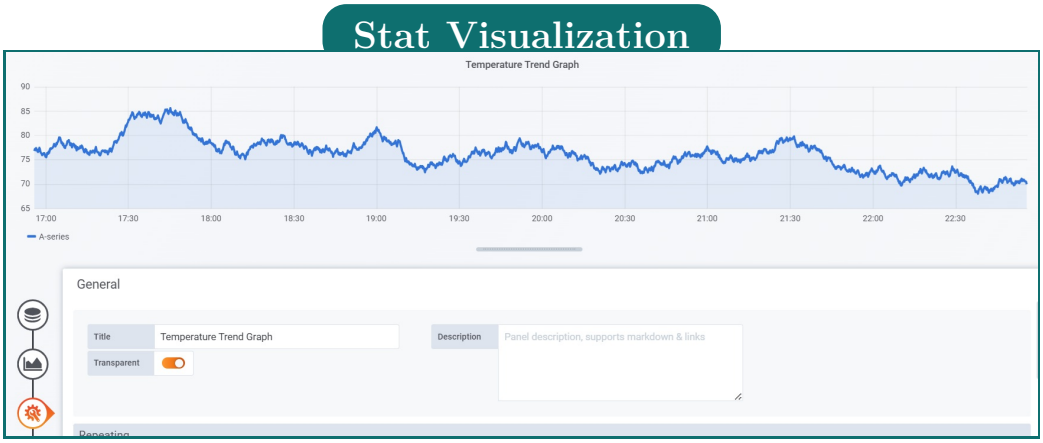
The Panel Links feature connects the panel to external systems or supporting resources:

- **Title:** Provide a descriptive label (e.g., "Platform.Log.com").
- **URL:** Insert the target website or dashboard link for quick access.
- **Open in New Tab:** Recommended for smoother navigation, ensuring that the main monitoring dashboard remains open.

Practical Use

By configuring General settings and Panel Links, monitoring panels become:

- **Clear and identifiable** through proper titling.
- **Flexible** with repeating panels for scalable monitoring.
- **Connected** by linking to external dashboards, reports, or documentation.



Convert to Row

The **Convert to Row** feature lets you organize panels into rows. This is particularly useful when working with multiple panels in the same dashboard,

as it allows for grouping and better layout management.

Add Panel

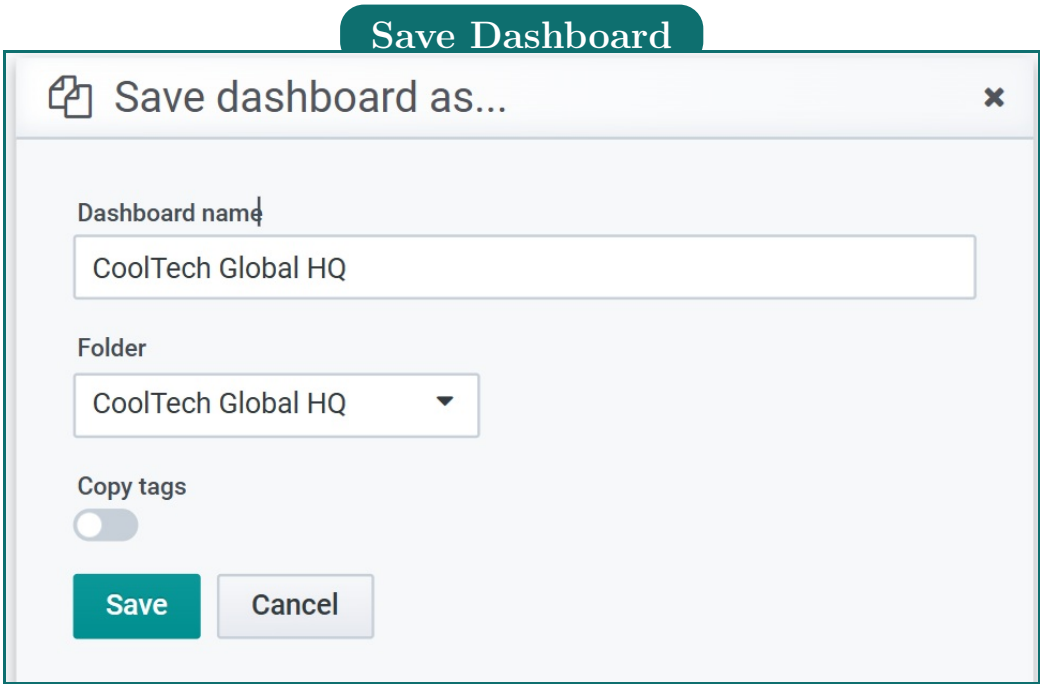
The **Add Panel** button (top toolbar) allows you to create additional panels in the same dashboard. Each panel can contain its own queries and visualizations, making it possible to monitor multiple datasets together.

Save Dashboard

After setting up panels, click **Save Dashboard** to store your configuration. You can choose a folder (e.g., General, etc.), set dashboard permissions, and configure options such as:

- Timezone (Default, Local Browser Time, or UTC)
- Auto-refresh intervals
- Tooltip mode (Default, Shared Tooltip, or Crosshair)

At this point, the dashboard has been created successfully. The next step is to return to the **Edit Section**, where detailed customization of queries, panels, and styles can be applied.

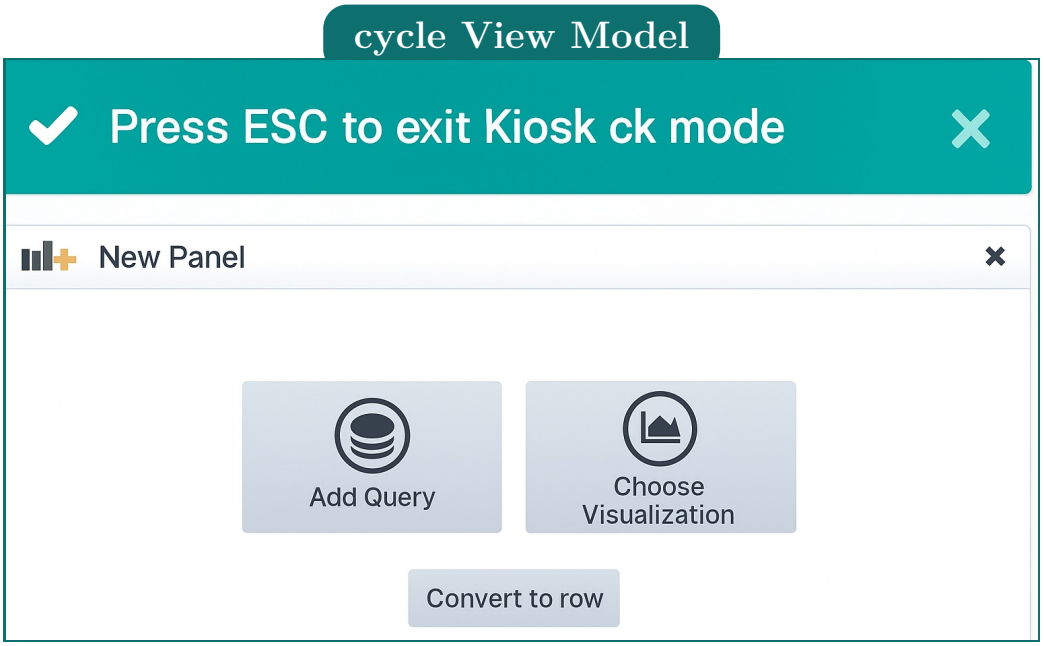


Dashboard Settings

The **Dashboard Settings** (gear icon) provide configuration options for the entire dashboard such as title, description, tags, and refresh intervals. See Section 1.3 (Create New Dashboard)for details on how to start a new dashboard.

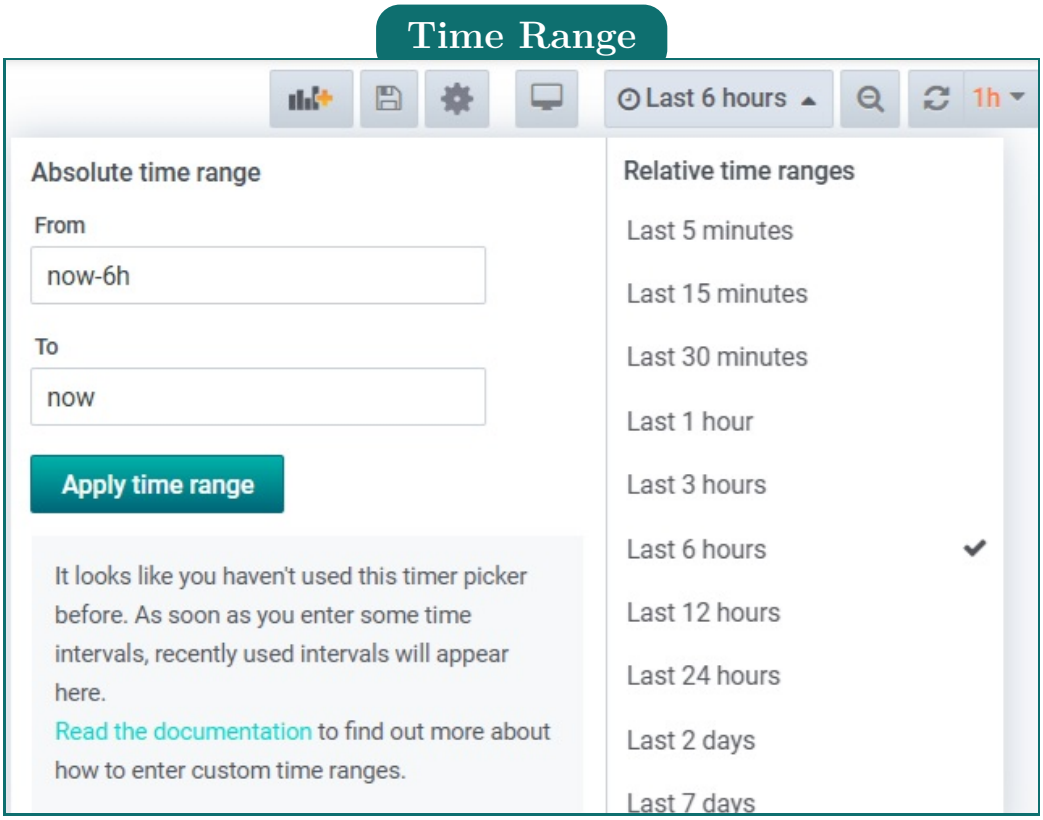
Cycle View Mode

The **Cycle View Mode** (monitor icon) toggles between different dashboard view modes, including kiosk/fullscreen, for easier presentation.



Time Range

The **Time Range** (clock menu) controls the time interval displayed in the panel. Options include relative ranges (e.g., last 5 minutes, last 6 hours, last 7 days) or custom absolute ranges. This is useful for focusing analysis on specific time periods.

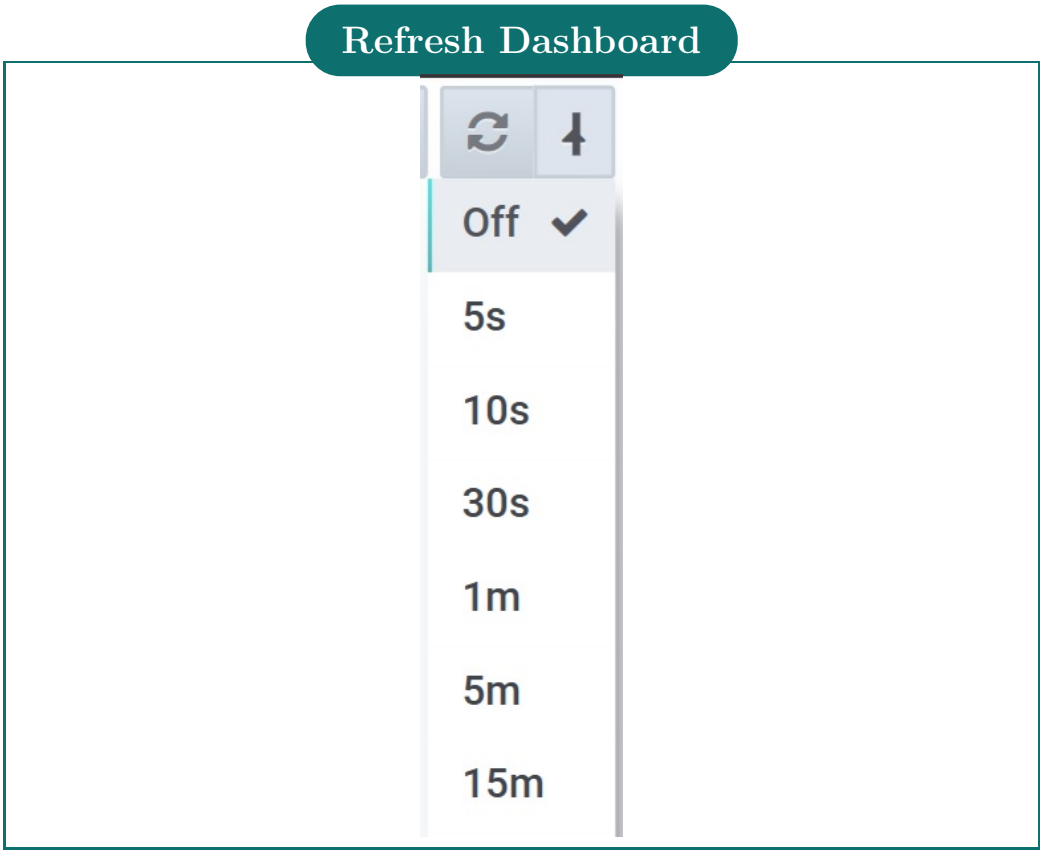


Time Range Zoom

The **Zoom Out** (magnifier icon) allows quick zooming out to view a wider time span for the selected data.

Auto-Refresh

The **Auto-Refresh** dropdown enables automatic refreshing of dashboard data at chosen intervals (e.g., every 5s, 1m, 5m, 30m, 1h, etc.). This ensures that real-time data is always up-to-date.



1.7 Gauge Panel Overview

The **Gauge Panel** is a visualization tool designed to present a single numeric value within a circular dial, providing users with an intuitive and immediate understanding of system status. It is especially effective when users need to:

- **Monitor thresholds visually** – Color-coded segments highlight whether the current value falls within safe, warning, or critical ranges.
- **Track performance or capacity** – Ideal for monitoring metrics such as CPU usage, memory, network load, or sensor readings.
- **Simplify data interpretation** – A speedometer-like display replaces detailed charts or tables with an at-a-glance view of the current state.

The Gauge panel typically consists of:

- **Dial/Needle or Filled Arc** – Indicates the position of the current value on the scale.

- **Threshold Colors** – Provide quick visual cues for normal, warning, or danger conditions.
- **Numeric Value Display** – Shows the exact measurement alongside the graphical indicator.

In summary, the Gauge panel transforms raw numeric data into a *speedometer-like visualization*, making it easier to assess live metrics at a glance.

Select Data Source

The first dropdown in the Query editor specifies the data source. Available options may include:

- `default`
- `logbotData`
- `Grafana`
- `Dashboard`
- `Mixed`
- `logbot Json`

logbotData in Query Panel

Configuring Queries in Logbot Data

To make the **Gauge Panel** functional, users must configure a query. The query defines *what data* will be displayed, *where the data comes from*, and *how it is processed*. Once configured, the Gauge panel transforms raw input into a clear, real-time indicator.

TIPS:

The Gauge panel only becomes meaningful when it is connected to live data. By configuring queries correctly:

- You ensure that the panel shows the correct metric (e.g., CPU load, temperature, network usage).
- You can refine the scope of data displayed by adjusting time ranges and refresh intervals.
- You maintain performance by limiting unnecessary data points.

Configuration Steps

1. **Select a Data Source** From the query section, choose the correct source (e.g., `logbotData`, `default`). This determines the system or database the panel retrieves values from.

2. **Choose the Metric** Define which metric should be monitored (e.g., temperature, CPU usage, requests per second). The Gauge will display the latest value of this metric in real time. *Why:* Choosing the right metric ensures the panel provides information relevant to your monitoring goals.

3. **Set Query Parameters** (optional) Fine-tune the data request to optimize accuracy and performance:

   - **Cache Timeout** – Defines how often the query is refreshed to keep data current.
   - **Max Data Points** – Improves performance by limiting how much data is processed.
   - **Relative Time / Time Shift** – Focuses the Gauge on a specific time window or shifts the analysis period for comparisons.

   Why: Parameters allow you to balance between data accuracy and system performance.

4. **View the Result** After configuration, the Gauge panel displays the live metric as a dial, arc, or needle. Colors update according to thresholds, and the numeric value is shown for precision.

Outcome Once queries are configured, the Gauge panel is no longer static — it becomes a dynamic monitoring element that provides immediate insight into critical values. Users can see both the precise numeric value and its status (normal, warning, critical) based on thresholds, all within a single glance.

Gauge

Gauge Visualization in Logbot Data

The **Gauge Visualization** in Logbot is one of the most intuitive ways to monitor live metrics.It provides users with a speedometer-like dial that reflects real-time data values in a visual format. This visualization is particularly useful for interns and new users who may not yet be familiar with raw datasets or advanced chart analysis.
When opening the Gauge panel, the following main areas appear:

- **Visualization Panel (Monitoring Area):** Displays the active gauges at the top of the dashboard. Each dial shows the measured value relative to defined thresholds, helping users instantly recognize if the system is performing normally, approaching a warning zone, or entering a critical state.
- **Configuration Tabs (Below the Monitoring Area):** These tabs allow users to control how the Gauge behaves and what data it represents. The key configuration sections are:
  - **Display:** Controls how the Gauge should look and what calculation to apply (e.g., *Max*, *Min*, or *Average*). Additional options include enabling labels and markers to make the gauge more descriptive.
  - **Field:** Defines technical parameters for the Gauge:
    - **Title:** Custom name for the Gauge.
    - **Unit:** Measurement unit (e.g., bits, bytes,
    - **Min/Max:** Scale boundaries of the Gauge to match the expected data range.
    - **Decimals:** Precision of the displayed value.
  - **Thresholds:** Colored ranges that indicate safe, warning, and danger zones. For example, values above 100 might display in purple, between 20–100 in yellow, and below 20 in green.
  - **Value Mappings:** Converts raw values into user-friendly labels. For

instance, a numerical output of `0` could be mapped to `Inactive`, while `1` is mapped to `Active`.

- **Data Links:** Adds hyperlinks for deeper inspection or navigation. Users can link the gauge to external dashboards, reports, or Logbot logs for quick reference.

TIPS:

Without proper configuration, the Gauge may either display misleading data or remain empty. By carefully setting the fields, thresholds, and mappings:

- Users gain a reliable, real-time monitoring view.
- Complex numeric data becomes accessible and meaningful, even for new team members.
- Quick decisions can be made based on visual indicators, reducing the need to interpret raw datasets.

In short, the Gauge Visualization within Logbot Data transforms technical system outputs into **clear, actionable insights**. For interns or new users, it serves as both a monitoring tool and a learning aid, bridging the gap between raw data and operational awareness.

General Settings

The **General tab** in the Gauge Panel allows you to configure the basic properties of the visualization. These settings define how the panel is presented on the dashboard and ensure clarity when monitoring data.

- **Title:** Enter a descriptive name for the panel. For example, instead of leaving it as *"Gauge"*, you may write *"CPU Usage Monitor"* or *"Network Load"*. This helps other users immediately understand the purpose of the panel.
- **Description:** Add an optional note explaining what the panel shows. Descriptions can support markdown and links, making it easier to document important context or reference related dashboards.
- **Transparent Mode:** Toggle this option if you want the panel background to blend with the dashboard design. It is particularly useful for clean or minimal layouts.
- **Repeating Panels:** The *Repeat* option allows you to duplicate the panel automatically based on a variable (e.g., one panel per server or sensor). This is especially useful when monitoring multiple data sources with the same settings.

- **Panel Links:** Add shortcuts that link to external sites, documentation, or other dashboards. For instance, you can provide a direct link to detailed logs, system metrics, or troubleshooting guides. Links open in a new tab, ensuring that monitoring is not interrupted.

In summary, the **General section** controls the identity and behavior of the Gauge panel. By giving panels clear titles, adding context with descriptions, and linking to useful resources, users can create dashboards that are both informative and easy to navigate.

1.8  Bar Gauge

For new users or interns, it is important to note:

- Every visualization (Graph, Stat, Gauge, Bar Gauge, etc.) depends on a correctly configured query.
- The query panel is not just a text box, but an interactive configuration area where you select data sources, apply filters, and define calculations.

Selecting a Data Source

At the top of the query panel, you must choose the **data source**. This selection tells Logbot where the data is coming from and how it should be handled. The available sources are:

- `default` – The fallback data source used when no specific source is configured.
- `logbotData` – The main source for real-time monitoring, pulling metrics directly from connected devices and sensors.
- {...} `logbotJson` – Allows structured JSON data to be used, with the option to format results as *time series* or *tables*.
- `Grafana (Test Data)` – Useful for testing queries and visualizations without live data, providing built-in examples such as random walk or sine waves.
- `Dashboard` – Enables reusing queries or values that are already defined in other panels within the same dashboard.
- `Mixed` – Combines multiple data sources into one query, allowing comparisons or overlays of different datasets.

Configuring Query Fields

Once the source is selected, the following fields appear:

- **Series / Metric** – Defines the specific measurement (e.g., temperature, CPU load, traffic).
- **Functions** – Allows applying transformations, calculations, or aggregations (e.g., average, max, moving average).
- **Cache Timeout** – How often the query result is refreshed.
- **Max Data Points** – Limits the number of points displayed (important for performance).
- **Relative Time / Time Shift** – Adjusts the time window (e.g., last 1h, shift by -15m).

Additional Options (per source) Depending on the chosen data source, additional settings may appear:

- For **logbotJson**: Option to switch between *Time series* and *Table*.
- For **Grafana Test Data**: Predefined patterns (e.g., sine wave, random walk).
- For **Dashboard**: Dropdown to select existing panels or links.
- For **Mixed**: Options to define how different sources are merged.

Practical Usage Example:

Suppose you want to monitor sensor data from a device:

1. Open the **Query Panel**.
2. Select `logbotData` as the data source.
3. Choose the metric (e.g., `temperature`).
4. Apply a function, such as `mean()` to smooth the values.
5. Set `Relative Time` to the last 1 hour.

The panel will now show a real-time graph or gauge (depending on the chosen visualization).
In summary, the Query Panel is where you *tell the system what data to fetch and how to display it*. Correct query configuration ensures that your visualizations (graphs, stats, gauges, bar gauges, etc.) are accurate and meaningful.

Bar Gauge Visualization

The **Bar Gauge** visualization panel is used to represent numerical values in the form of horizontal or vertical bars. It is particularly effective for comparing multiple metrics side-by-side or monitoring a single metric with thresholds.

Monitoring View: When the Bar Gauge is added, the panel initially displays empty or default bars. This provides a placeholder view, showing the area where live data will later appear once queries and thresholds are configured.

Configuration View: In the configuration mode, several options are available:

- **Calculation (Calc):** Determines how values are aggregated, such as *Mean*, *Max*, or *Min*.
- **Orientation:** Controls whether the bars are displayed *Horizontally* or *Vertically*.
- **Mode:** Allows different display styles, such as *Retro LCD* for classic digital looks.
- **Field Options:** - *Title:* Sets the label for the metric. - *Unit:* Defines the measurement unit (e.g., - *Min / Max:* Adjusts the data scale boundaries. - *Decimals:* Controls the precision of displayed numbers.
- **Thresholds:** Add colored ranges (e.g., green = safe, yellow = warning, red = critical). This helps users identify states quickly.
- **Value Mappings:** Enables assigning text labels to values or ranges (e.g., 1 = ON, 0 = OFF).
- **Data Links:** Provides clickable links attached to values, which can open external dashboards or documentation.

By moving from the monitoring view to the configuration view, users can customize the Bar Gauge to provide both a clear visual comparison and actionable context for decision-making.

General Settings

The **General** tab provides essential configuration options for naming, organizing, and linking your Bar Gauge panel. These settings do not affect the metric data itself but control how the panel is presented within the dashboard.

Main Fields:

- **Title:** Defines the name displayed at the top of the panel (e.g., *"CPU Usage"* or *"Network Status"*). A clear title helps other users quickly identify the panel's purpose.
- **Description:** An optional field where you can enter additional notes. Descriptions support text formatting (Markdown/links), making it useful for documentation or providing instructions.

- **Transparent:** Toggle this switch to remove the panel's background color. A transparent setting helps panels blend seamlessly with the dashboard background, improving readability in dense layouts.

Repeating Panels:

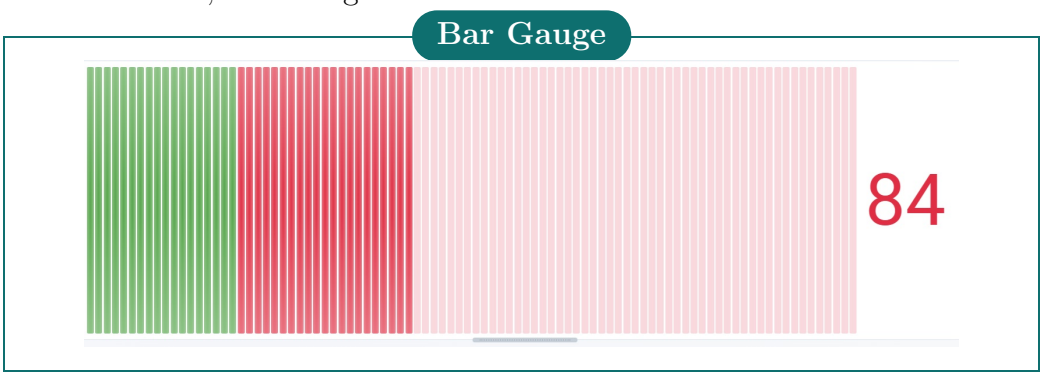
- **Repeat:** This option allows the panel to be automatically duplicated based on a variable. For example, if you monitor multiple servers, the same Bar Gauge template can repeat for each server without manually creating separate panels.

Panel Links:

- **Add Link:** Creates custom links attached to the panel. These links can redirect to:
  - Another dashboard inside the platform.
  - An external website (e.g., documentation, ticketing system).
  - A report or related visualization.

  This feature helps connect monitoring with deeper analysis or action tools.

In summary, the **General Settings** ensure that the Bar Gauge panel is well-labeled, easy to understand, and connected to additional resources. While visualization and queries handle data, the General tab improves usability, documentation, and navigation across the dashboard.



1.9 Table

1.9.1 Table Panel Overview

The **Table Panel** is a visualization tool designed to display raw or processed data in a structured, grid-based format. Unlike gauges or charts, which provide summarized or visualized values, the Table panel is useful when users need to:

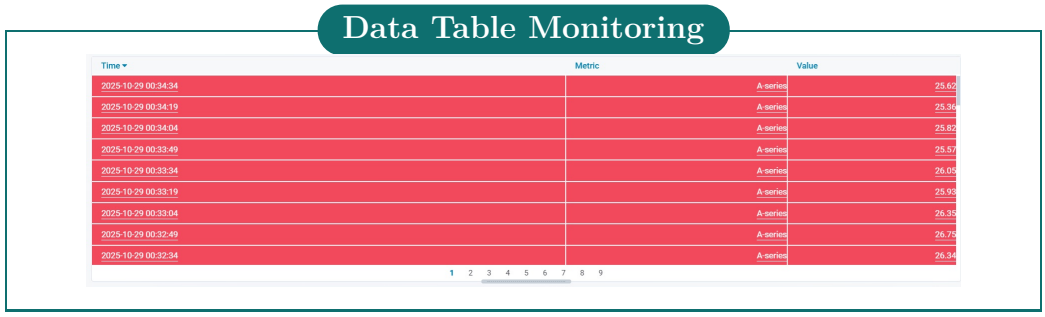
- **Inspect detailed data points** – Every row corresponds to a data entry, making it easy to review logs, metrics, or sensor outputs.
- **Compare multiple values at once** – Columns represent metrics or fields, allowing direct comparison.
- **Export or analyze structured results** – Data in tables can be sorted, filtered, and later exported for further analysis.

This makes the Table panel especially helpful for monitoring events, log data, or datasets that require a precise tabular view.

Configuring Queries for the Table Panel

As with all panels in Logbot, the **Query Section** is the starting point for the Table panel. Without a configured query, the table will display *"No data"*. To set up a query:

1. **Select a Data Source:** From the **Query dropdown menu**, choose the appropriate source:
   - `default`
   - `logbotData`
   - `logbotJson`
   - `Grafana`
   - `Dashboard`
   - `Mixed`
2. **Define the Query:** - Specify the `Series` or metric you want to display. - Add `Functions` if calculations or transformations are needed. - Configure additional options like `Relative time` or `Time shift` to adjust the data range shown.
3. **Choose the Format:** In the `Format as` dropdown, select:
   - `Time series` – to display chronological data with timestamps.
   - `Table` – to view results as structured rows and columns.
4. **Preview the Output:** Once the query is set, the panel will update and show live data in tabular format. Columns can include metrics, timestamps, or labels, depending on the query configuration.



TIPS:

the Table panel transforms query results into a structured and highly readable format, making it indispensable for detailed monitoring and analysis.

Table Visualization

The **Table Panel** is designed to present data in a structured grid format, making it ideal for viewing multiple metrics, logs, or time-series results in a clear and organized way. Unlike gauges or graphs, the table does not summarize a single value — instead, it provides a detailed view of raw or transformed datasets. This makes it especially useful when users need to compare values, apply styles, or link results to external resources.
Configuring the table involves several key sections:

1. **Data Transformation (Table Transform):** Defines how the raw data is reshaped for display. Options include:
   - *Time series to rows* – each data point becomes a row.
   - *Time series to columns* – each metric is displayed as a column.
   - *Time series aggregations* – summarize values (e.g., average, min, max).
   - *Annotations or JSON Data* – integrate external metadata or JSON fields.

2. **Paging and Font Size:** Controls how many rows are visible per page and adjusts the text size (e.g., `Rows per page = 100`, `Font size = 100%`). This helps balance readability with the amount of data displayed.

3. **Column Styles:** Allows fine-grained control over how each column is displayed.
   - **Apply to column name:** Target a specific column (e.g., `Time`).

- **Type:** Choose between `Number`, `String`, `Date`, or `Hidden`.
- **Date Format:** If a date column is selected, formats such as `YYYY-MM-DD HH:mm:ss` can be applied.
- **Alignment:** Controls text placement (left, center, right).
- **Units and Decimals:** For numeric columns, specify measurement units (e.g., %) and decimal precision.

4. **Thresholds and Color Coding:** Define ranges that automatically change the background or text color of a column. For example, values above `80` could be highlighted in red, while safe values remain green. This enables quick pattern recognition.

5. **Data Links:** Each table cell can be turned into a hyperlink that opens external dashboards, reports, or documentation. Users may configure:

   - `URL` – target destination (static or variable-based).
   - `Tooltip` – description shown when hovering.
   - `Open in new tab` – ensures the main dashboard stays open.

TIPS:

the **Table Panel** transforms raw metrics into a customizable, Excel-like view. By combining transformations, column styles, thresholds, and links, users can turn complex datasets into structured, interactive reports directly within the Logbot dashboard.

Table Visualization

The **Table Panel** is designed to present data in a structured grid format, making it ideal for viewing multiple metrics, logs, or time-series results in a clear and organized way. Unlike gauges or graphs, the table does not summarize a single value — instead, it provides a detailed view of raw or transformed datasets. This makes it especially useful when users need to compare values, apply styles, or link results to external resources.
Configuring the table involves several key sections:

1. **Data Transformation (Table Transform):** Defines how the raw data is reshaped for display. Options include:

   - *Time series to rows* – each data point becomes a row.
   - *Time series to columns* – each metric is displayed as a column.

- *Time series aggregations* – summarize values (e.g., average, min, max).
- *Annotations or JSON Data* – integrate external metadata or JSON fields.

2. **Paging and Font Size:** Controls how many rows are visible per page and adjusts the text size (e.g., `Rows per page = 100`, `Font size = 100%`). This helps balance readability with the amount of data displayed.

3. **Column Styles:** Allows fine-grained control over how each column is displayed.
   - **Apply to column name:** Target a specific column (e.g., `Time`).
   - **Type:** Choose between `Number`, `String`, `Date`, or `Hidden`.
   - **Date Format:** If a date column is selected, formats such as `YYYY-MM-DD HH:mm:ss` can be applied.
   - **Alignment:** Controls text placement (left, center, right).
   - **Units and Decimals:** For numeric columns, specify measurement units (e.g., %) and decimal precision.

4. **Thresholds and Color Coding:** Define ranges that automatically change the background or text color of a column. For example, values above `80` could be highlighted in red, while safe values remain green. This enables quick pattern recognition.

5. **Data Links:** Each table cell can be turned into a hyperlink that opens external dashboards, reports, or documentation. Users may configure:
   - `URL` – target destination (static or variable-based).
   - `Tooltip` – description shown when hovering.
   - `Open in new tab` – ensures the main dashboard stays open.

1.10  Text Panel

The**Text Panel** is used to display static or dynamic information directly on the dashboard. Unlike numerical or graphical panels, the Text Panel does not visualize metrics, but instead provides context, explanations, or documentation for the dashboard. It can also be used to embed formatted text, links, or even HTML content for more advanced customization.

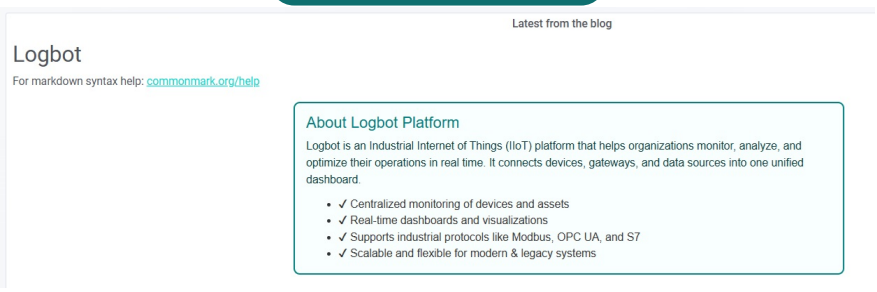
Modes of Display: The panel supports two content modes:

- **Markdown Mode:** Allows you to write content using markdown syntax. This is useful for adding formatted text, headers, bullet points, and links. Example:

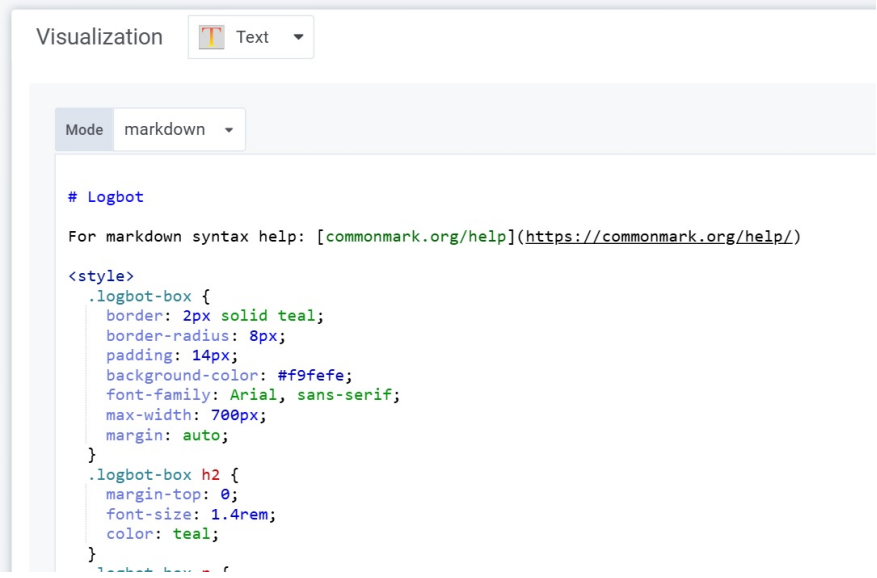
```
# System Overview
This dashboard monitors the **CPU usage**.
[Help](https://commonmark.org/help/)
```

- **HTML Mode:** Allows advanced users to write raw HTML code for complete control over formatting and layout, including embedding images or styling text.

Text Visualization



Text visualization



General Settings:

- **Title:** Defines the title shown at the top of the text panel.
- **Transparent:** Toggles the background. When enabled, the panel blends into the dashboard background.
- **Description:** Adds supporting notes about the panel's purpose. This is especially helpful for documenting dashboards.

Repeating: This option allows the text panel to be repeated for each selected variable. For example, if you have a variable representing different servers, the panel can repeat once per server, each with the same text template.

Panel Links: You can add links to other dashboards, documentation, or external resources. This is useful for creating navigation menus or quick references directly inside the dashboard.

Why use the Text Panel? The Text Panel is essential for providing explanations, instructions, or contextual information within dashboards. It ensures that users — especially new team members — understand the purpose of the dashboard, the meaning of its metrics, and how to interpret the data being shown.

1.11 Import

The **Import** feature allows users to bring existing dashboards into the Logbot platform. Each dashboard in Logbot is internally represented as a `JSON` file, which contains the complete configuration of its panels, queries, and visualizations. This means that whenever a user creates or customizes

a dashboard, the system automatically generates a corresponding `JSON` definition that can be exported, shared, or re-imported.

1.11.1 Import Methods

Users can import dashboards in the following ways:

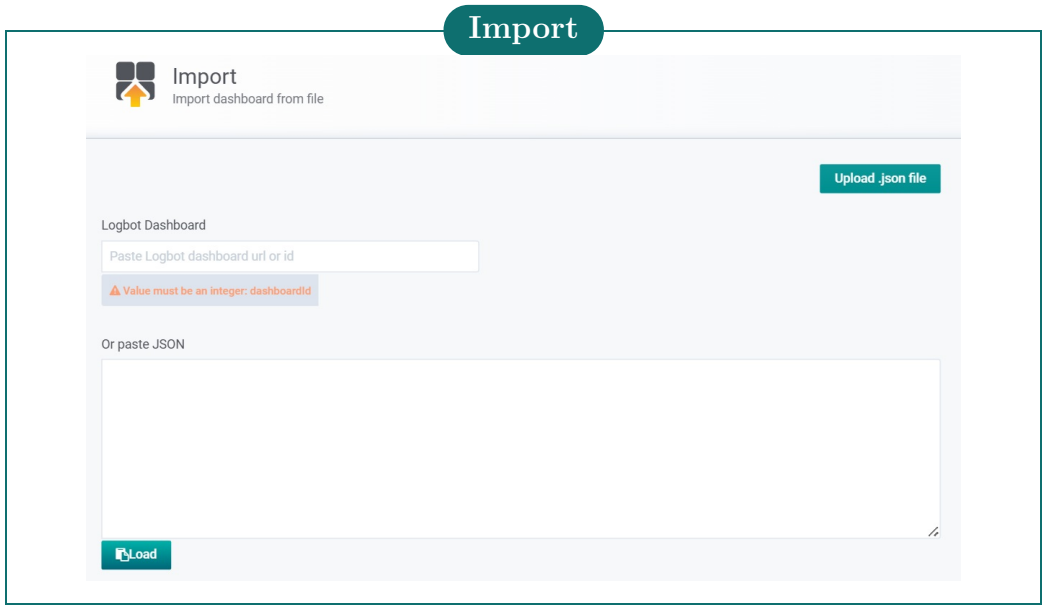
- **Paste Dashboard ID or URL** – Enter the Logbot dashboard identifier (`dashboardId`) or URL.
- **Upload JSON File** – Import a dashboard from a saved `.json` file that was exported earlier.
- **Paste JSON Content** – Directly paste the raw JSON definition into the editor box.

TIPS

Every dashboard is stored as a `JSON` definition. This makes it possible to easily back up dashboards, move them between environments, or share configurations with other users.

1.11.2 Validation and Loading

When entering a dashboard ID, the system checks that the input is a valid integer (**`dashboard Id`**). If the value is invalid, a warning message will be shown. Once valid input is provided, clicking the **Load** button completes the import and adds the dashboard to the workspace.



1.12 Alerts

1.12.1 Channels

Alert Management

The **Alert System** in Logbot helps users detect problems in real-time by sending notifications when certain conditions are met.

Alerts are divided into two sections:

- **Notification Channels** – define *where* alerts will be sent (Email, Slack, Teams, etc.).
- **Alert Rules (Roles)** – define *when* alerts are triggered (e.g., CPU > 80%, No Data, etc.).

Notification Channels

A **Notification Channel** specifies the delivery method for alerts. For example: Email, Slack, Telegram, or Microsoft Teams.

Steps to configure:

Open the **Alerts** → **Channels** menu.

Click **New Notification Channel**.

Enter a **Name** (e.g., "System Email Alerts").

Select a **Type**:

- Email
- Slack
- Telegram
- Microsoft Teams

Fill in the required settings (e.g., email addresses or webhook URLs).

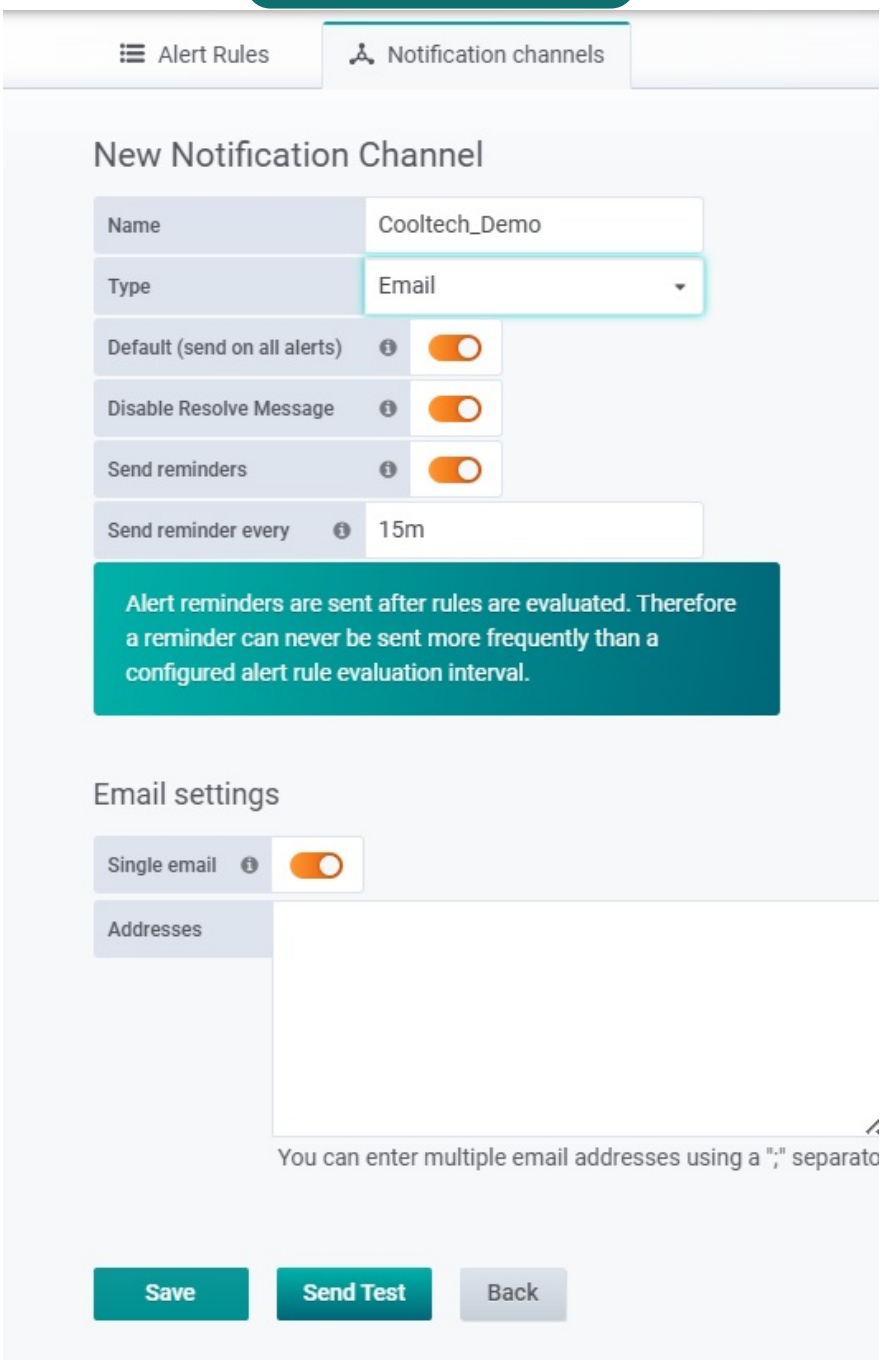
Test with **Send Test**.

Click **Save**.

Tip

Always create a notification channel first — without it, alerts cannot be delivered.

Notification Channels



1.13 Alert Notifications — Slack Channel

Create a Slack notification channel so alert rules can automatically deliver alert messages to your workspace.

1.13.1 Open the Page

Navigate to:

`Alerts` → `Notification Channels` → `New Notification Channel`.

1.13.2 Header Fields

Name	A clear label for this channel (e.g., `cooltech-hq`).
Type	Select **Slack**.
Default (send on all alerts)	Turn **ON** to use this channel automatically for new alert rules; leave **OFF** to add it manually per rule.
Disable Resolve Message	Turn **ON** to suppress "resolved" notifications; leave **OFF** to send a message when alerts close.
Send Reminders	Turn **ON** to resend alerts periodically until they are resolved.
Send Reminder Every	Interval between reminders (e.g., `15m`). *Note:* The interval cannot be shorter than the alert evaluation interval.

1.13.3 Slack Settings — Field Guide

Slack supports **two integration methods**. Choose one and complete only the fields relevant to that method.

Method A — Incoming Webhook (no bot token) Use this when you already have a Slack *Incoming Webhook URL*.

Url	*Required.* webhook URL (e.g., `https://hooks.slack.com/services/T000/B000/XXXX`).
Recipient	Optional. Override destination channel or user — use `#channel-name`, `@username` (lowercase, no spaces), or a Slack ID (`C.../U...`).
Username	Optional bot display name (e.g., `CoolTech Industries`).
Icon Emoji	Optional emoji avatar (e.g., `:rotating_light:`). Overrides Icon URL.
Icon URL	Optional image URL for avatar (e.g., company logo). Used only if *Icon Emoji* is empty.
Mention Users	Optional. Comma-separated Slack **User IDs** (`U...`), e.g., `U024BE7LH,U061F1EUR`.
Mention Groups	Optional. Comma-separated **User Group IDs** (`S...`);.
Mention Channel	Optional. Choose **Disabled** (no mass mention) or **Every active channel member** (equivalent to `@channel/@here`).
Token	Leave blank in Method A.

Method B — Bot Token (no webhook URL) Use this when you have a Slack *Bot/User OAuth Token* (starts with `xoxb-`).

Url	Leave blank in Method B.
Recipient	*Required.* Target channel or user — `#channel-name`, `@username`, or Slack ID (`C.../U...`).
Username	Optional. Override sender name shown in Slack.
Icon Emoji / Icon URL	Optional. Same behavior as above.
Mention Users	Optional. Comma-separated **User IDs** (`U...`) to mention.
Mention Groups	Optional. Comma-separated **User Group IDs** (`S...`) to mention.
Mention Channel	Optional. Choose **Disabled** or **Every active channel member**.
Token	*Required.*,token: (e.g., `xoxb-1234567890-0987654321-ABCDEFGHijklmnop`).

1.13.4 Example Values

Webhook Example

- Url: `https://hooks.slack.com/services/T000/B000/1a2b3c4d5e6f`
- Recipient: `#cooltech-hq` (or `@oncall`)
- Username: `CoolTech Industries`
- Icon Emoji: `:rotating_light:`
- Mention Users: `U024BE7LH,U061F1EUR`
- Mention Groups: `S012ABC34,S056XYZ78`
- Mention Channel: `Every active channel member`

Bot Token Example

- Recipient: `#ops-eu`
- Token: `xoxb-1234567890-0987654321-ABCDEFGHijklmnop`

1.13.5 Save and Test

1. Click **Save**.
2. Click **Send Test**. Confirm that a message appears in the Slack destination channel.

1.13.6 Toggle Reference (Quick Guide)

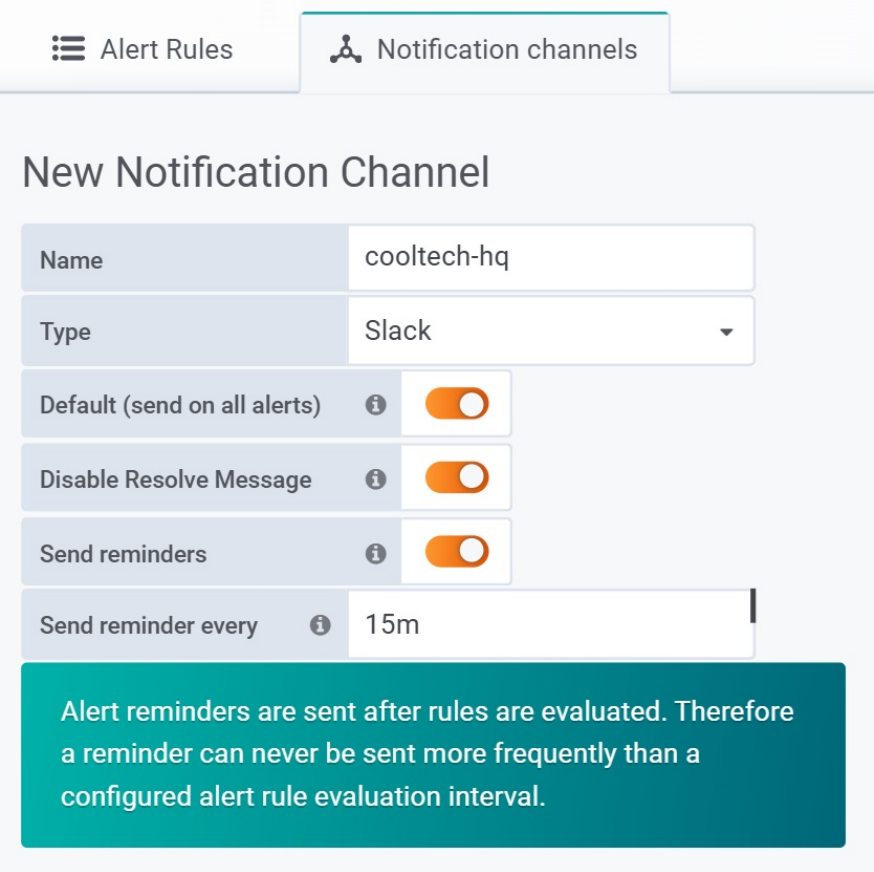
Default (send on all alerts)	When **ON**, new alert rules inherit this channel automatically.
Disable Resolve Message	When **ON**, only active ("firing") alerts are sent; resolved messages are suppressed.
Send Reminders	When **ON**, repeated notifications are sent until resolution.
Send Reminder Every	Frequency for reminders (e.g., `15m`). Must not be shorter than the rule's evaluation interval.

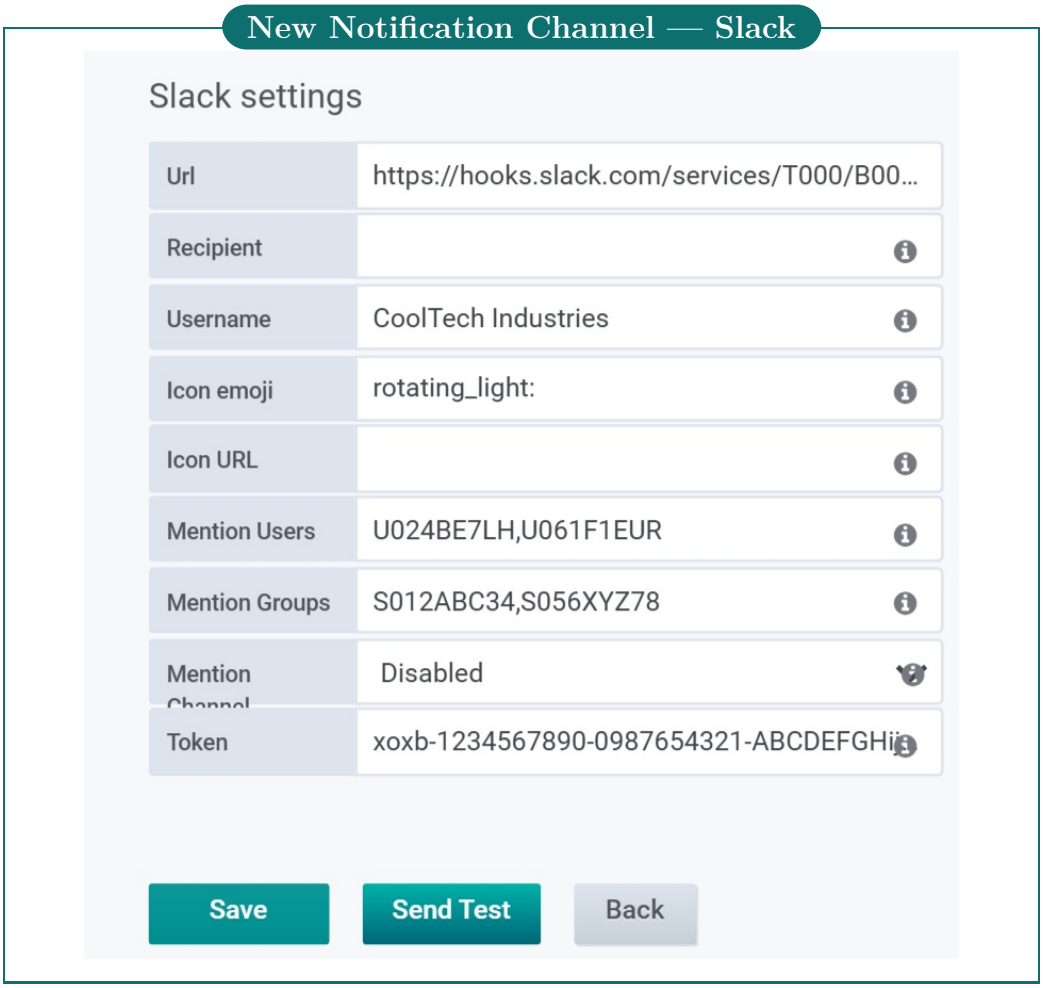
1.13.7 Validation and Best Practices

- **Recipient format:** Use `#channel-name`, `@username` (lowercase, no spaces), or a Slack ID (`C.../U...`). Copy IDs directly from the Slack profile or browser URL.
- **Webhook vs Token:** Use only one integration type at a time — Webhook (**Url** filled, **Token** empty) or Token (**Token** filled, **Url** empty).

- **Mentions:** *Mention Users/Groups* fields require Slack **IDs** (not names). Example: `U024BE7LH`. The *Mention Channel* option will ping all members — use sparingly.
- **Branding:** Prefer **Icon URL** for company branding; fallback to **Icon Emoji** for lightweight styling.
- **Troubleshooting:** If no message appears, verify: Recipient channel exists, Slack app permissions are correct, and the token/webhook is valid. Also check that the alert rule actually fired within the selected time range.

New Notification Channel — Slack





1.14 Roles

Alert Rules (Roles)

An **Alert Rule** defines the conditions for triggering notifications.

Available States:

- **OK** – everything is normal.
- **Not OK** – condition is violated.
- **Alerting** – system is actively sending notifications.
- **No Data** – no input is available.
- **Paused** – the rule is disabled.
- **Pending** – waiting before firing an alert.

Steps to configure:

Go to **Alerts** → **Roles**.

Click **Add Alert Rule**.

Define the **Condition** (e.g., CPU usage > 80%).

Set an **Evaluation Interval** (how often the rule is checked).

Enable **Reminders** (e.g., send every 15 minutes until resolved).

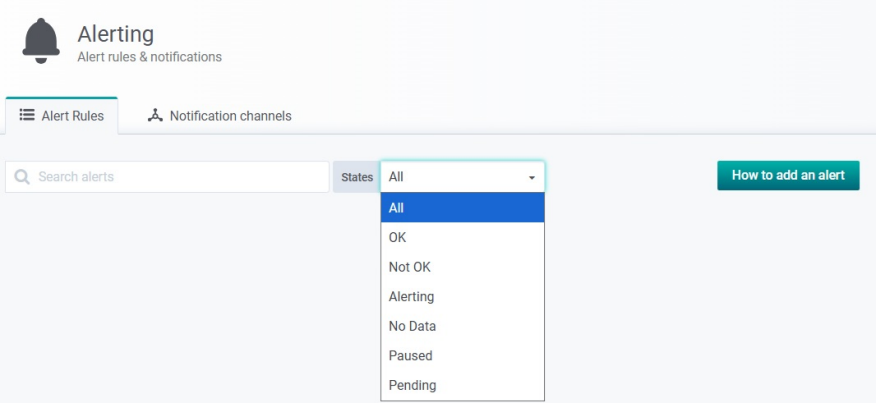
Assign a **Notification Channel** (created earlier).

Click **Save**.

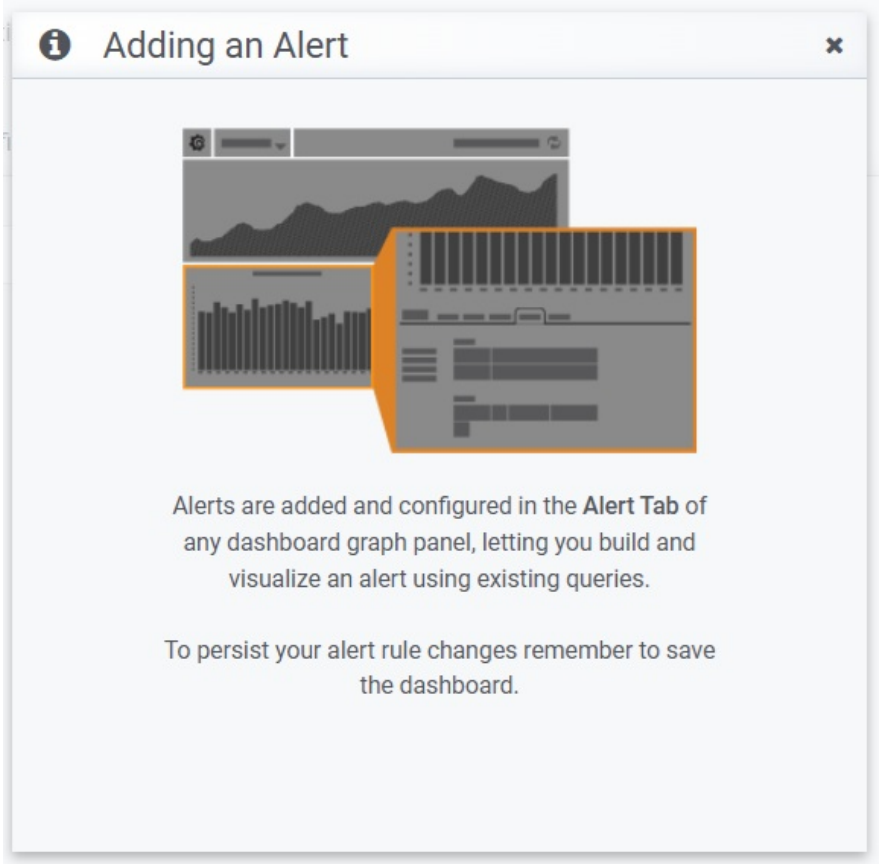
Summary for Interns

- Always create a **Notification Channel** first.
- Then configure **Alert Rules**.
- Use the states (OK, Alerting, etc.) to quickly monitor system health.

Alerting



Adding An Alerts



Chapter 1

Alarms & Notifications

1.1 Alarms

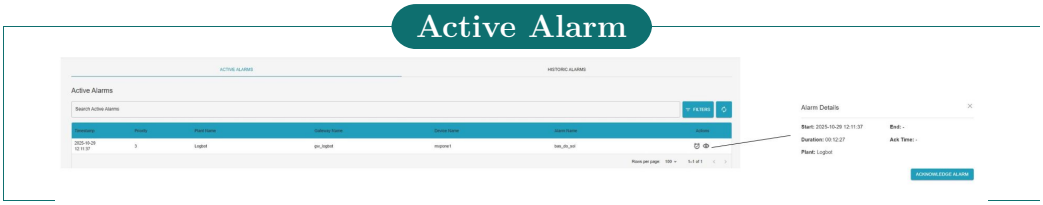
The **Alarms** section in the Logbot platform provides users with the ability to monitor and manage both current and historical alarm data related to IoT devices.

1.1.1 Active Alarms

The **Active Alarms** tab displays alarms that are currently being triggered by online devices. This indicates that:

1. Some devices are actively online and sending alarm metrics.
2. Alarm metrics have defined *intervals* and types.
3. Real-time monitoring of alarms is available including details such as:

✓ Timestamp
- Priority
- Plant Name
- Gateway Name
- Device Name
- Alarm Name
- Available actions

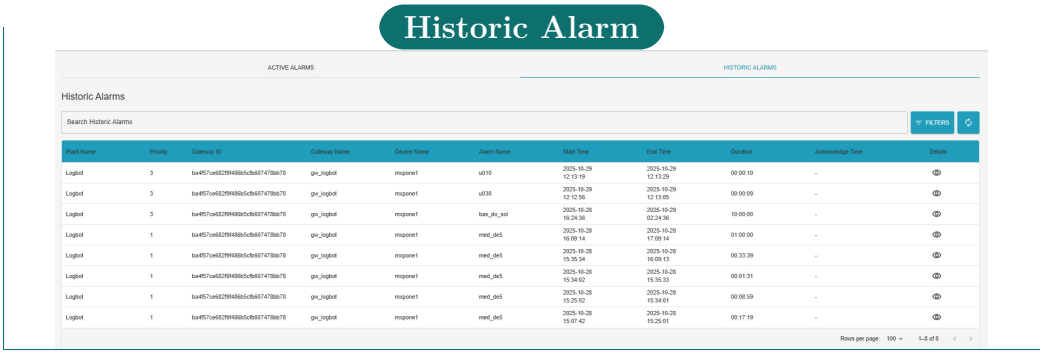


1.1.2 Historic Alarms

The **Historic Alarms** tab displays previously triggered alarms that have been recorded. This includes:

- Alarms sent in the past by devices.
- Ability to filter by time range, plant, gateway, or device.
- Table columns include:

1. Plant Name
2. Priority
3. Gateway ID
4. Gateway Name
5. Device Name
6. Alarm Name
7. Start Time, End Time
8. Duration



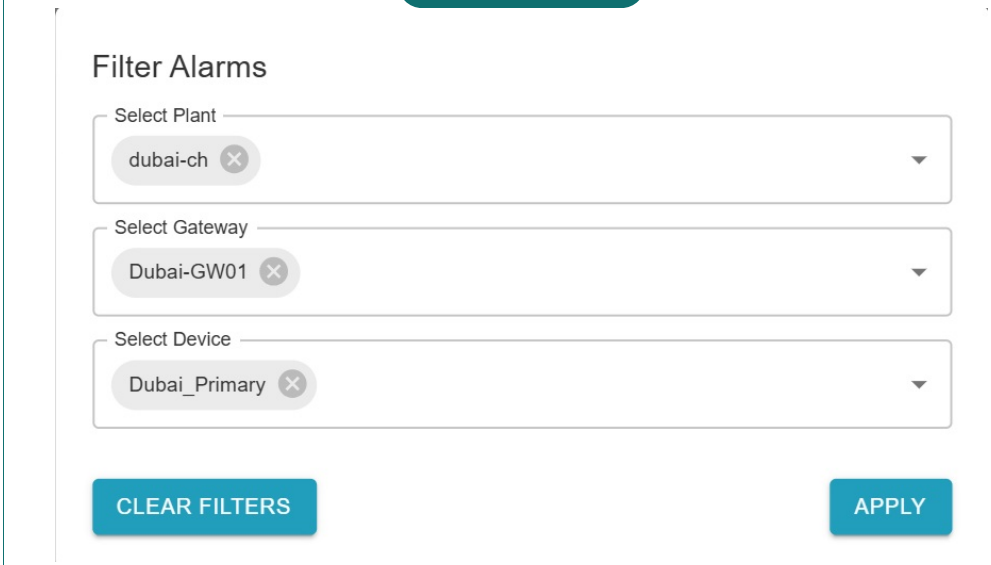
1.1.3 Filter Alarms

Users can apply filters to refine their search for both active and historical alarms. Filters include:

- Plant selection (e.g., `dubai-ch`)
- Gateway selection (e.g., `Dubai-GW01`,)
- Device selection (e.g., `Dubai_Primary`)
- Time range (start and end time)

The **Alarms** section enhances visibility into system events and device behavior, supporting effective monitoring and rapid response.

Filter Alarm



1.2 Notification

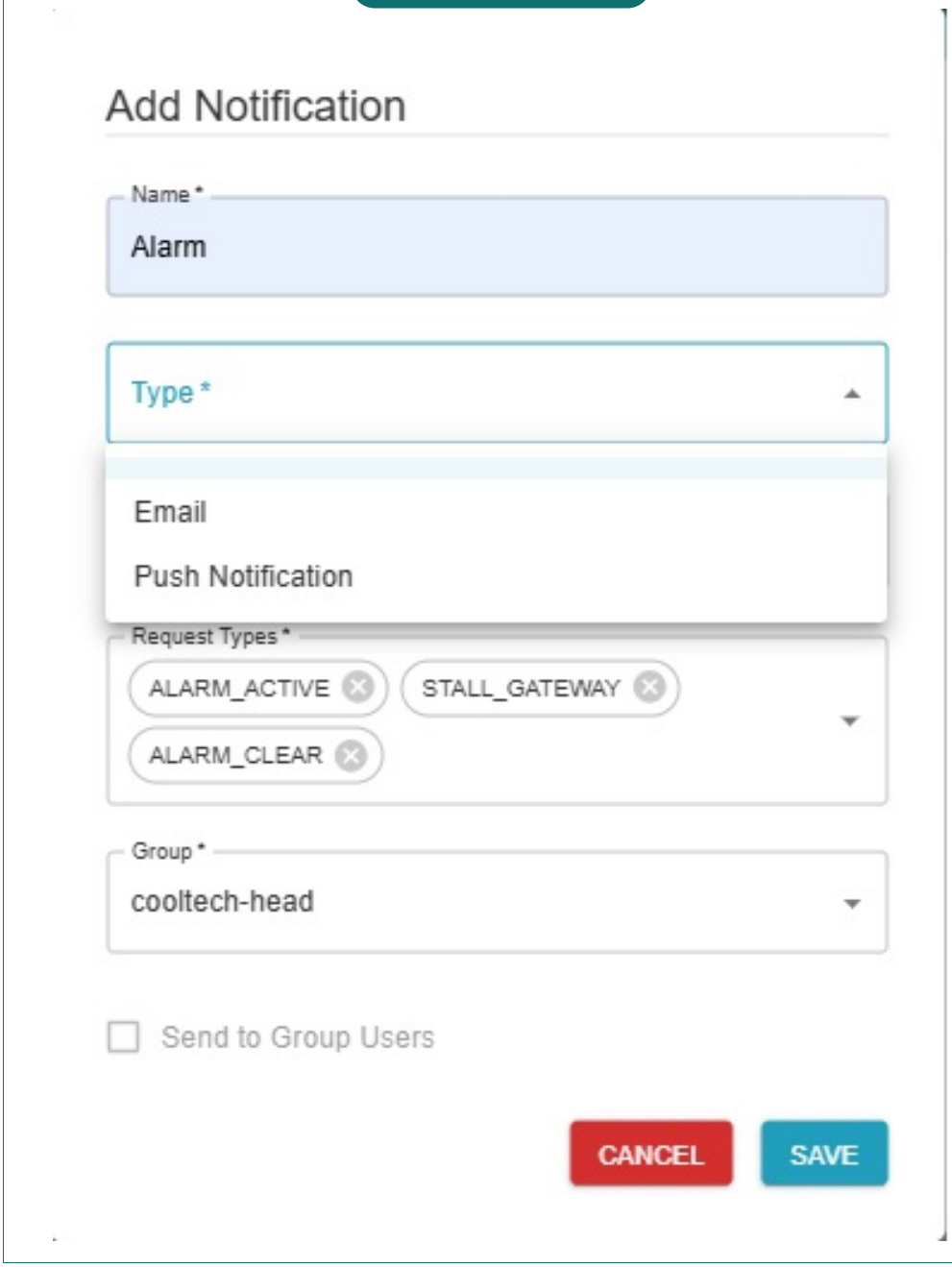
1.2.1 Add Notification

From the left sidebar menu, go to:
`Notifications → Add Notification`

This will open the notification creation form. **Notification Parameters:**

- **Name:** e.g., `gateway2`
- **Type:** Select one of:
  - `Email` – If this type is selected, you must manually check the option `Send to Group Users`.
  - `Push Notification` – When selected, the `Send to Group Users` option is automatically enabled.
- **Priority:** Choose a level from 0 to 3, where 0 is the lowest and 3 is the highest priority.
- **Request Types:** You can select one or both of the following:
  - **Alarm_Active:** Indicates that an alarm condition is currently active in the system. The underlying issue or trigger is still present and requires attention. The system will continue to show this state until the condition is resolved.
  - **Alarm_Cleared:** Indicates that a previously active alarm has been resolved or acknowledged. The root cause is no longer present, and the alarm state has returned to normal. Clearing can be automatic (when the condition is fixed) or manual (acknowledgement by a user).
  - **Stall_Gateway (Does Not Respond):** This status occurs when a gateway device stops responding to system queries or heartbeats. It may be due to network connectivity issues, device malfunction, or power loss. Requires prompt investigation to restore normal communication.
- **Group:** Choose a user group from the dropdown (e.g., `mohajerit`)

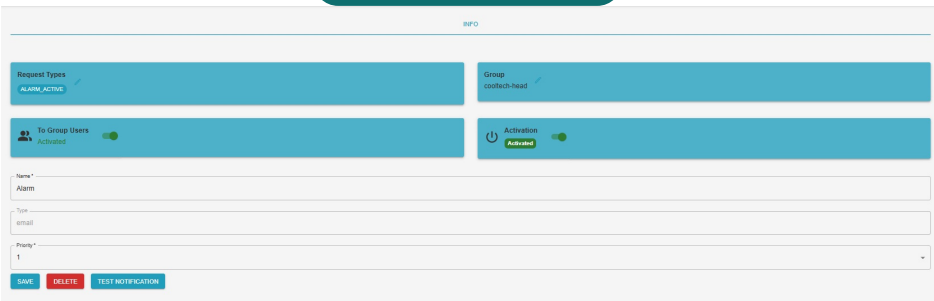
Add Notification



Notification Info and Contact Settings Explanation:

- **Info Tab:**
  - When **Email** is selected as the notification type:
    - The `Send to Group Users` option can be manually activated or deactivated.
    - If deactivated, users must manually define recipients via the **Contacts** tab.
  - When **Push Notification** is selected:
    - The `Send to Group Users` option is always enabled by default and cannot be changed.
- **Contacts Tab:**
  - Only visible when **Email** is selected and `Send to Group Users` is disabled.
  - Allows the addition of multiple contact emails through the **Add Contact** option.
  - These manually added emails will receive alerts when group messaging is turned off.
- **System Behavior:**
  - If `Send to Group Users` is deactivated, the status is shown as `false` in the notification list.
  - For **Push Notification** type, the system always marks it as `true` and disables manual control.

Notification Info



Chapter 1

Reviewing Events

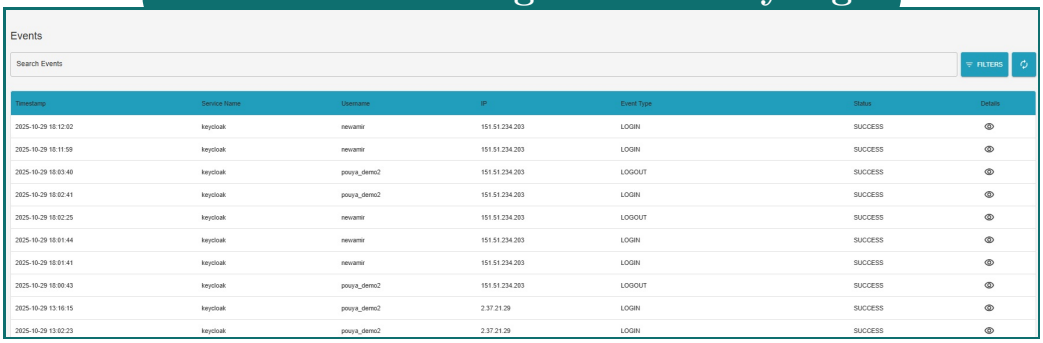
1.1 Events

The **Events** tab provides a detailed audit trail of user activities, including actions such as logins, logouts, and system interactions. Each log entry captures essential metadata such as the timestamp, username, service name, IP address, event type, and status, allowing administrators to monitor and trace system activity effectively.

In the Events window, these activity logs are displayed in a structured format, enabling easy review and filtering of recorded events. Each entry includes important metadata:

- **Timestamp:** Date and time the event occurred (e.g., `2025-07-08 12:42:40`)
- **Service Name:** Source service that triggered the event (e.g., `assets`, `keycloak`)
- **Username:** User who triggered the event (e.g., `shirintest`)
- **IP:** IP address of the originator (e.g., `2.37.21.29`)
- **Event Type:** Nature of the activity (e.g., `LOGIN`, `PLANT_CREATED`, `GATEWAY_UPDATED`)
- **Status:** Result of the event (e.g., `SUCCESS`)
- **Details:** You can click the (eye icon) to view detailed information about the event.

Events list showing user activity logs



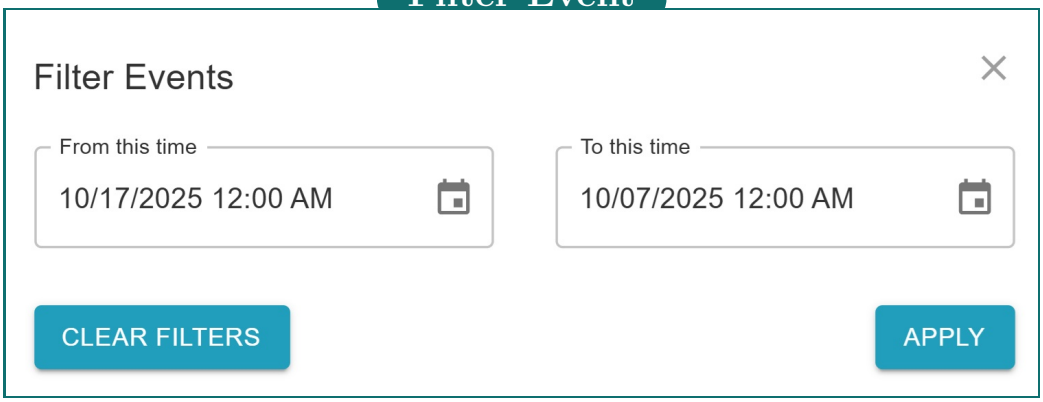
Filtering Events

To refine the displayed log entries, use the **Filter Events** feature. Specify a date and time range in the **From this time** and **To this time** fields to display only events within that period.

- Click **Apply** to filter the results.
- Click **Clear Filters** to reset and show all events.

This feature enables users to quickly identify specific activities or audit records within a defined timeframe.

Filter Event



Chapter 1

Managing Groups

1.1 Groups

The **Groups** section allows administrators to manage group structures, assign users, and organize access control within the platform. It is divided **Info** several tabs: Info, **Users**, **Roles**, **Requests**, and **Plants**. Each tab serves a specific purpose in the group management process.

1.2 Groups Management

The **Groups** module enables users to manage organizational structures including subgroups, associated users, permissions (roles), requests, and plants.

Group Code : Shareable alphanumeric code used during user registration to request membership in the group.

1.2.1 Info Tab

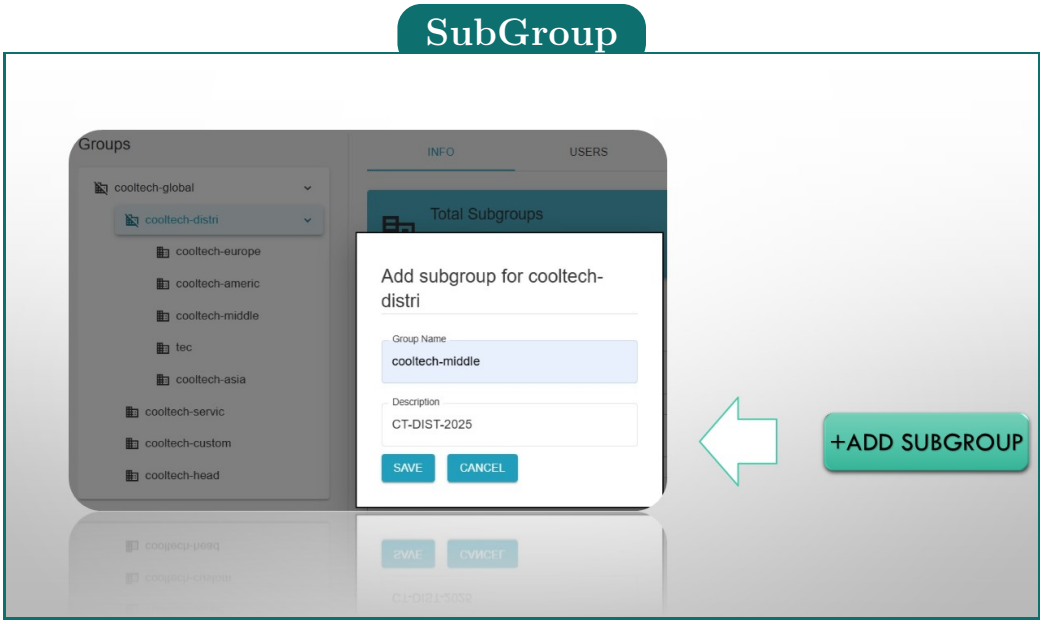
The **INFO** tab provides general information about the group.

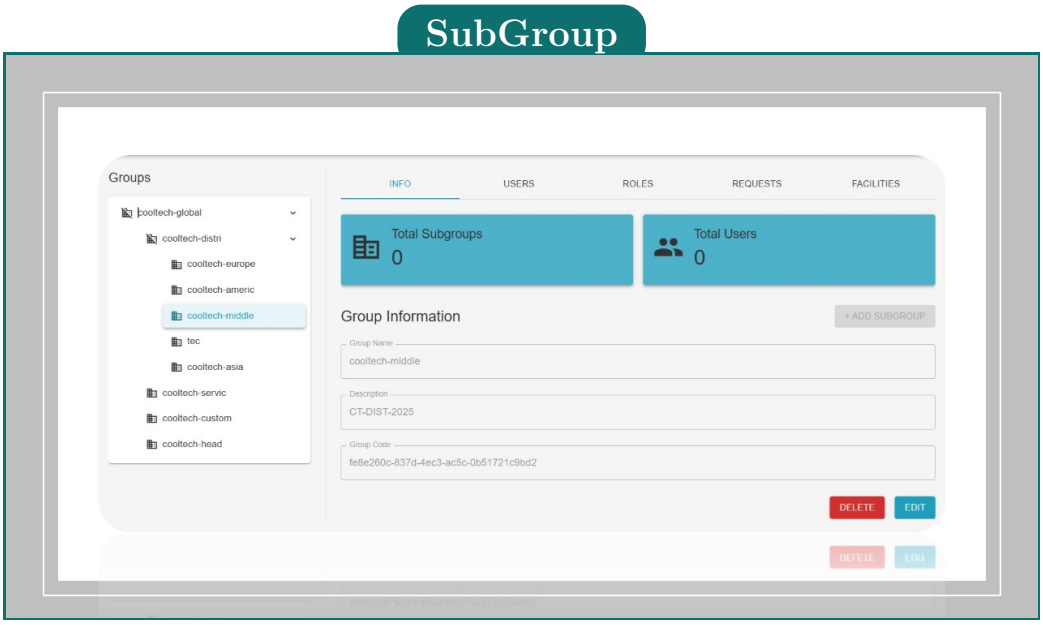
The **Groups** module enables administrators to structure organizations by creating subgroups, managing users, assigning roles, handling requests, and linking Plants.

Info Tab Overview

The **Info** tab provides a summary and key editable information related to a specific group.

- **Total Subgroups:** Indicates how many subgroups are associated with the current group.
- **Total Users:** Displays the number of users currently assigned to the group.
- **Group Information:**
  - **Group Name** – the name of the group (editable).
  - **Description** – a brief summary of the group's purpose or content.
  - **Group Code** – a unique identifier used to connect users during registration.
- **Add Subgroup:** Create new subgroups by clicking the `+ ADD SUBGROUP` button. Subgroups can have their own users and hierarchical structure.



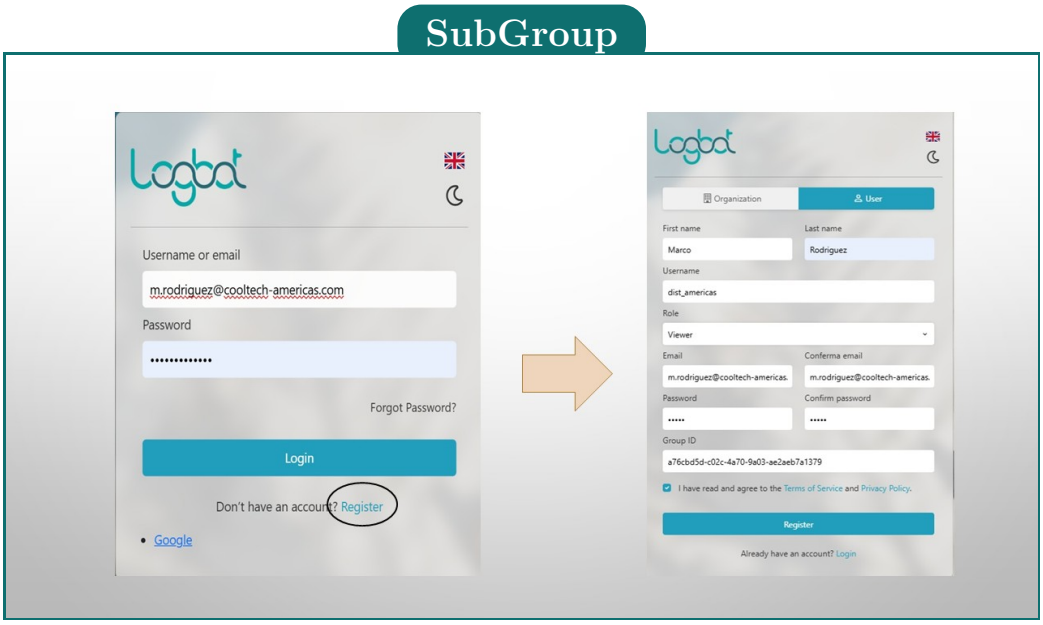


Important: The **Group Code** must be securely shared with intended users. It is required at the time of user registration to correctly associate users with their respective group.

1.2.2 User Registration with Group Code

Group Code (Registration) During sign-up, new users enter the **Group Code** to request membership in a group.

- Admin shares the **Group Code**.
- User goes to: `https://platform.logbot.eu`
- Fills in their information and enters the **Group Code**.
- Submits the form. A request is generated and visible in the **Requests** tab.
- Admin reviews and accepts the request to grant access.



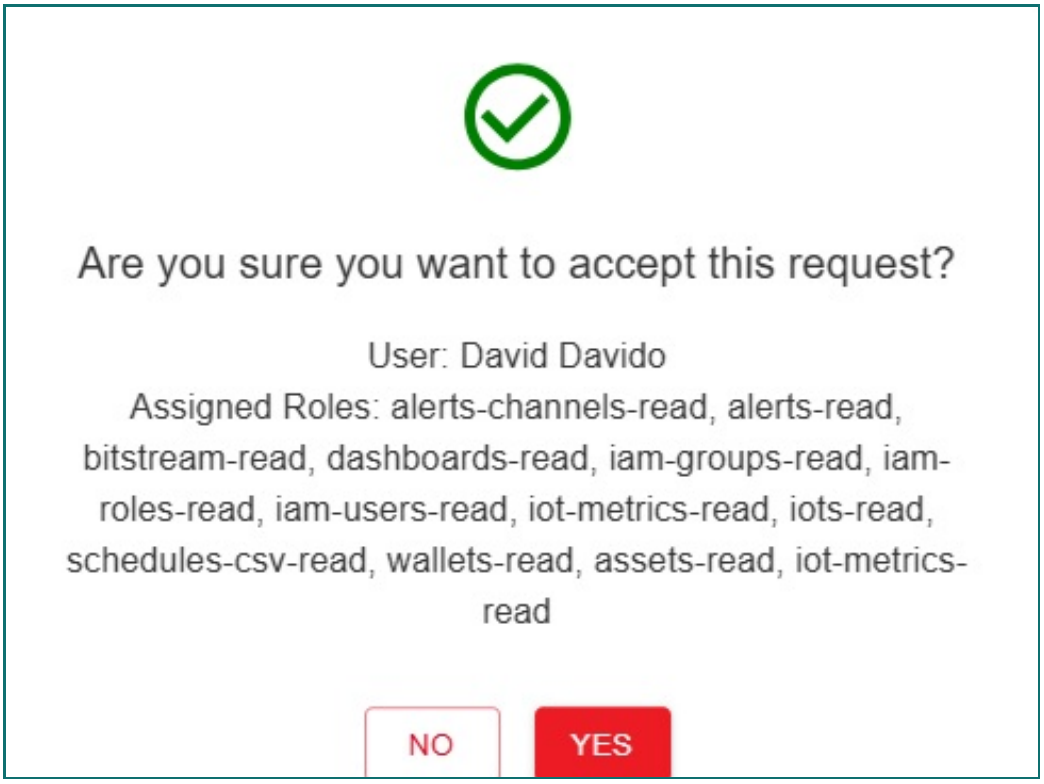
1.2.3 Users Tab

This tab displays all users assigned to the selected group and provides tools for managing their access and activity.

- Lists user details including: **Name**, **Surname**, **Username**, **Email**, and **Activation Status**.
- Admins can view and manage each user's role and status.
- User-specific actions (e.g., view info, remove) are accessible via the action buttons on the right.

Features:

- **Activation Status Bar:** Indicates the percentage of active users in the group.
- **Search Functionality:** Easily filter users by name, email, or username.
- **Enabled Column:** Displays whether a user is currently active or not.



1.2.4 Roles Tab

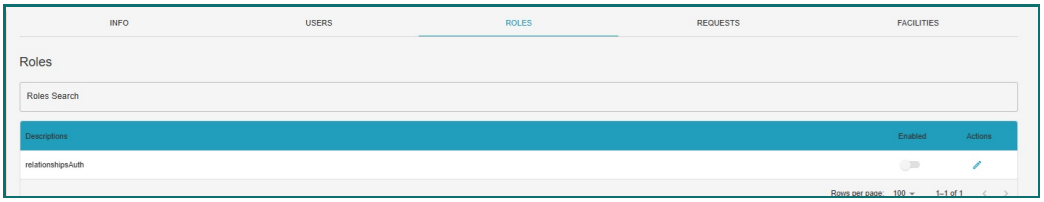
The Roles tab allows administrators to assign and manage permissions for users within the selected group. It provides a clear overview of available roles and their configurations. **Roles Management:**

- View a list of all available roles along with their descriptions.
- Use toggle switches to **enable** or **disable** specific roles for the group.
- Modify role settings by clicking the pencil icon under the **Actions** column.

Types of Roles:

- **View** – Grants view-only access to specific modules or data.
- **Edit** – Allows users to edit or create content within allowed sections.
- **Admin** – Provides full control, including access to configuration and management tools.

These role types define the level of access and interaction each user has with the platform's features.



1.2.5 Requests Tab

The **Requests** tab displays all pending requests from users who wish to join the group using the Group Code.

- Lists all users who have registered with the corresponding **Group Code**.
- Admins can approve and activate user accounts directly from this panel.
- Helps manage group membership through manual invitation and approval processes.

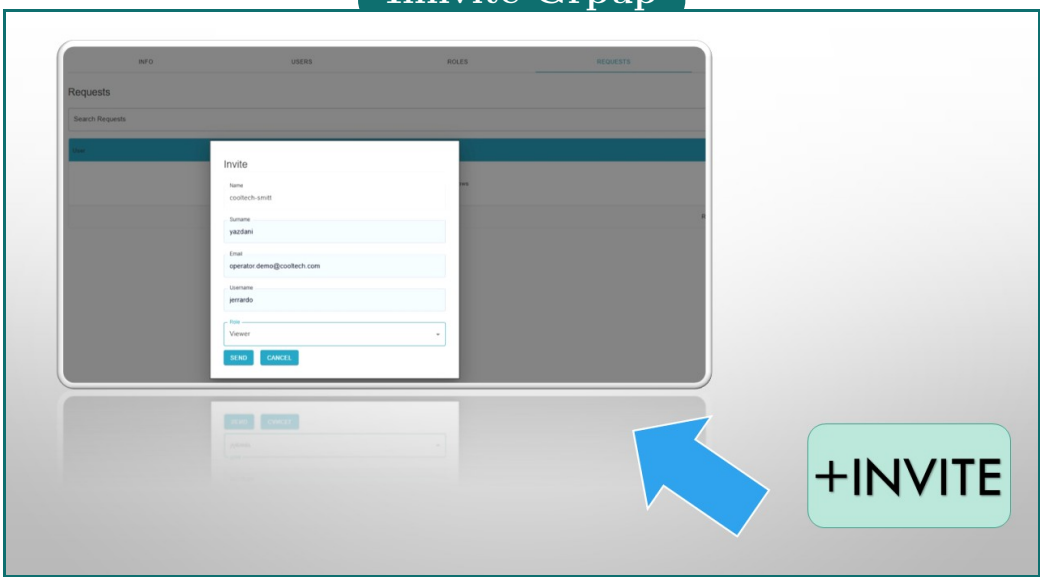
Capabilities:

- View and manage incoming join requests.
-
- Search and filter requests by **Name**, **Email**, or **Group Code**.
- Use the `+ INVITE` button to manually invite new users via email.

Invite Users

- Click on the `Invite` button.
- Enter the user's email address.
- User must register using the same email address.
- Admin approves the user request to activate the account.

Innvite Grpup



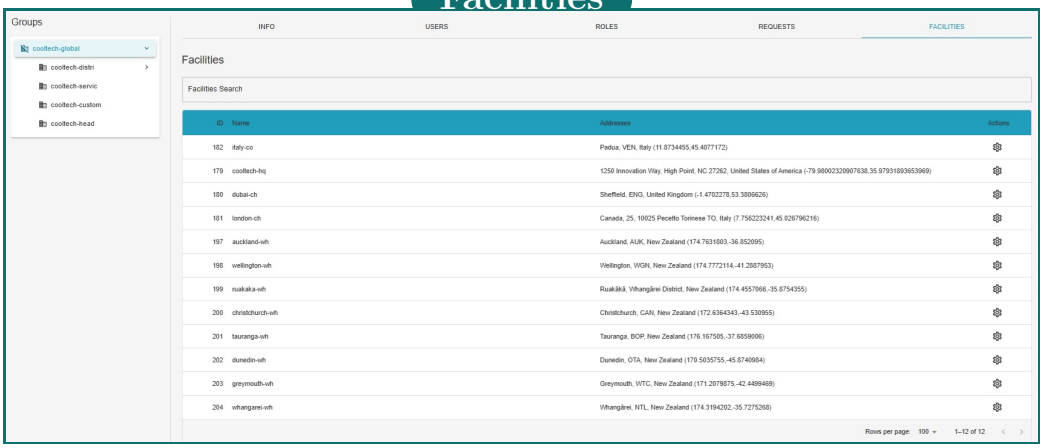
1.2.6 Plants Tab

This tab provides access to all Plants (Sites) that are associated with the group.

Functionality:

- View and manage linked Plants under the group structure.
- Each Plant entry displays its **ID**, **Name**, and **Address**.
- A configuration icon under the **Actions** column allows for adjusting Plant-specific settings.

Facilities



Action

The Action icon allows further configuration of each Plant.

- INFO: Basic Plant info (name, address, description).
- TAGS: Optional tagging for search/filtering.
- CONNECTED GATEWAYS: Shows the number of gateways connected.
- ASSIGNED GROUPS: Lists groups associated with the Plant.

The users can:

- Edit details (name, address, description).
- Save or delete the Plant.
- View associated groups and gateways for better configuration insight.

  Purpose: Provides full control over each facility's settings.

Enables efficient linking of Plant with groups and gateways.

Info Section:

The Info section displays the essential details about a selected facility. It is commonly accessed when you click the gear icon under the Actions column in the Plant table.

- **Connected Gateways:** Shows the number of gateways currently linked to this facility. This number helps confirm that the Plant is correctly integrated with the network and devices.
- **Assigned Groups:** Indicates how many groups are associated with the facility. Helps ensure that the right users and roles are connected to the plant.
- **Name:** Editable field showing the current name of the facility. You can click here to rename the facility if needed. Example: Change from "Logbot-Office" to "Production-Hub-1".
- **Address:** A field displaying the physical address of the facility. Important for location tracking or documentation. Can be updated directly in this field.
- **Description:** Optional field for adding a short explanation or reference for the facility. Example: "Ps752 – Secondary testing area".
- SAVE Button: After making any changes to Name, Address, or Description, click SAVE to apply and store those changes.
- **DELETE Button:** Permanently removes the plant from the system.

Group Facilities-Info

Tags Tab

Tags allow administrators to assign key-value metadata to a facility. These can be used for filtering, categorizing, or searching Plants.

Fields:

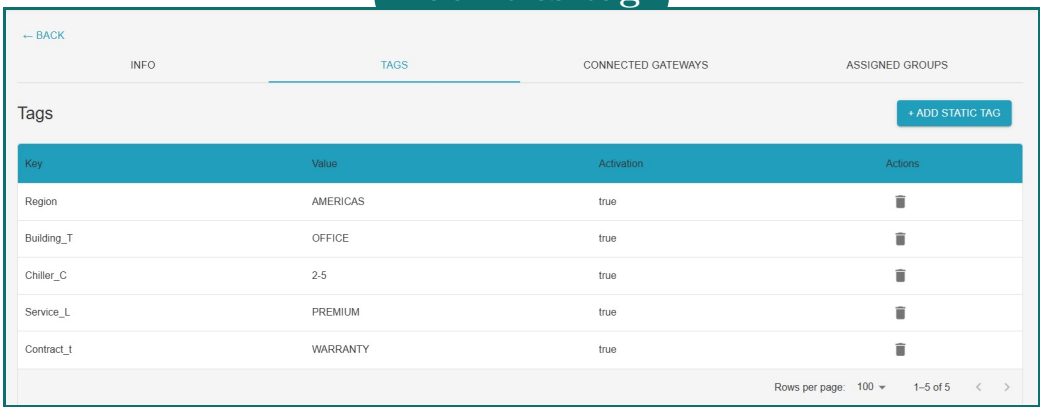
- Key: The category of the tag (e.g., “Chiller-C”).
- Value: The specific value of the tag (e.g., “2-5”).
- Activation: A boolean field indicating if the tag is active.
- Actions: Trash icon allows deletion of the tag.

Actions:

- + ADD STATIC TAG: Adds a new tag entry manually.

Note:

Facilities-tag



Connected Gateways Tab

This section shows a list of all gateways that are or can be connected to the facility.

ID: Unique internal identifier for the gateway.

Name: The assigned name of the gateway (editable from gateway settings).

Description: Optional descriptive text.

License UUID: A unique license identifier used to authorize the gateway.

Activation: Indicates whether the gateway is currently activated or not.

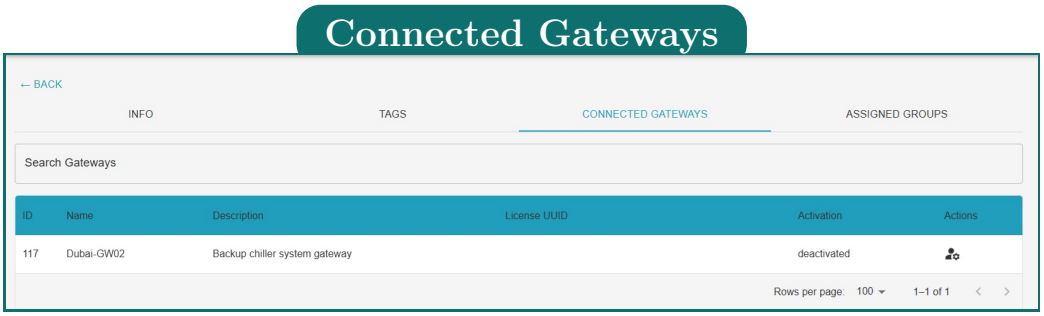
Actions:

The user icon might be used for access or assignment settings.

The gear icon usually provides configuration options.

Usage: Allows managing which physical or virtual gateways are linked to a Plant.

Admins can activate, deactivate, or reassign gateways.



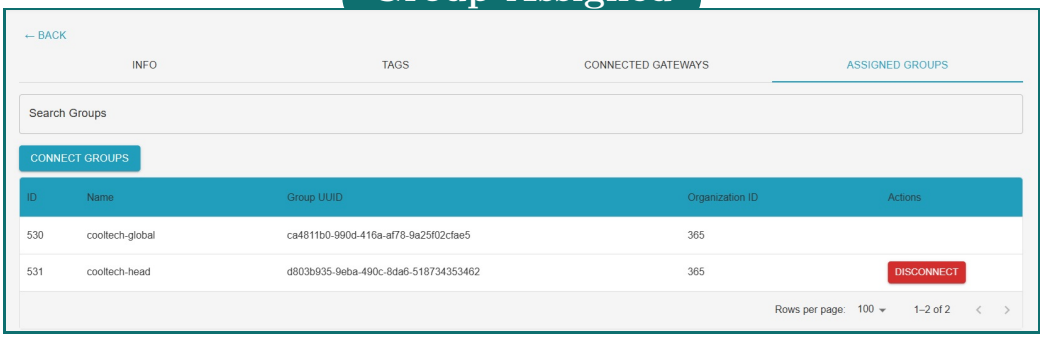
Assigned Groups Tab

This view allows the user to manage which groups are associated with the selected Plant. These groups typically define user access and organizational hierarchy.

Typical Actions:

- View a list of associated groups.
- Add or remove groups to/from the facility.
- Enables better organization and separation of data visibility based on groups.

Group Assigned



Chapter 1

Managing Users

1.1 USER

This module lets administrators create, **enable/disable**, and assign roles to users. Supports both permanent and temporary user accounts.

1.2 User Management

The **Users** module in the Logbot platform allows administrators to manage user accounts, assign roles, and monitor group affiliations.

User Table View

The user interface displays a searchable table with the following columns:

- **Name** and **Surname**: The first and last name of the user.
- **Username**: The login ID used for platform access.
- **Email**: The user's contact email address.
- **Group**: Indicates the organization/group the user is affiliated with.
- **Enabled**: Shows whether the user account is currently active.

Actions: Provides icons for viewing user details, editing settings, or deleting the user.

Two buttons at the top-right allow:

- + **ADD USER**: Opens a form to register a permanent user.
- + **ADD TEMPORARY USER**: Adds a user account with limited duration or scope.

1.2.1 +ADD USER

The **Add User** form allows administrators to register new users to the platform. Each field in the form must be completed with accurate user information.

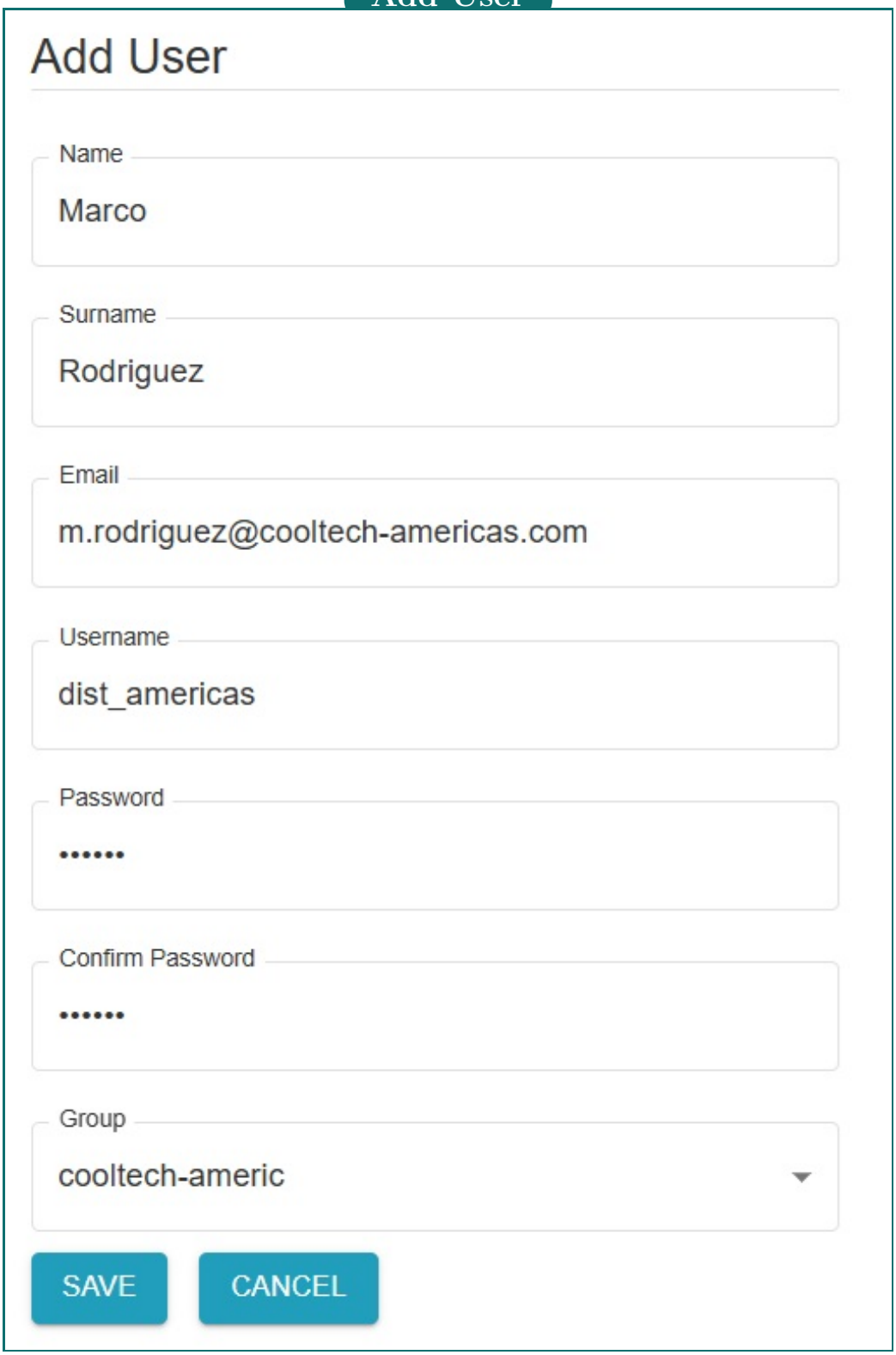
- **Name:** The user's first name.
  Example: `Marco`
- **Surname:** The user's last name.
  Example: `Rodriguez`
- **Email:** The user's valid email address for contact and login purposes.
  Example: `m.rodriguez@cooltech-americas.com`
- **Username:** A unique identifier used for login.
  Example: `dist_americas`
- **Password:** A secure password set by the user.
- **Confirm Password:** Re-enter the password to ensure accuracy.
- **Group:** Select a user group from the dropdown menu to assign the user to a specific role or access level.
  Example: `cooltech-americ`

Buttons:

- `SAVE` – Submits the form and adds the user.
- `CANCEL` – Cancels the operation and clears the form.

This interface is used primarily by administrators to manage user access and assign users to the correct functional group within the system.

Add User

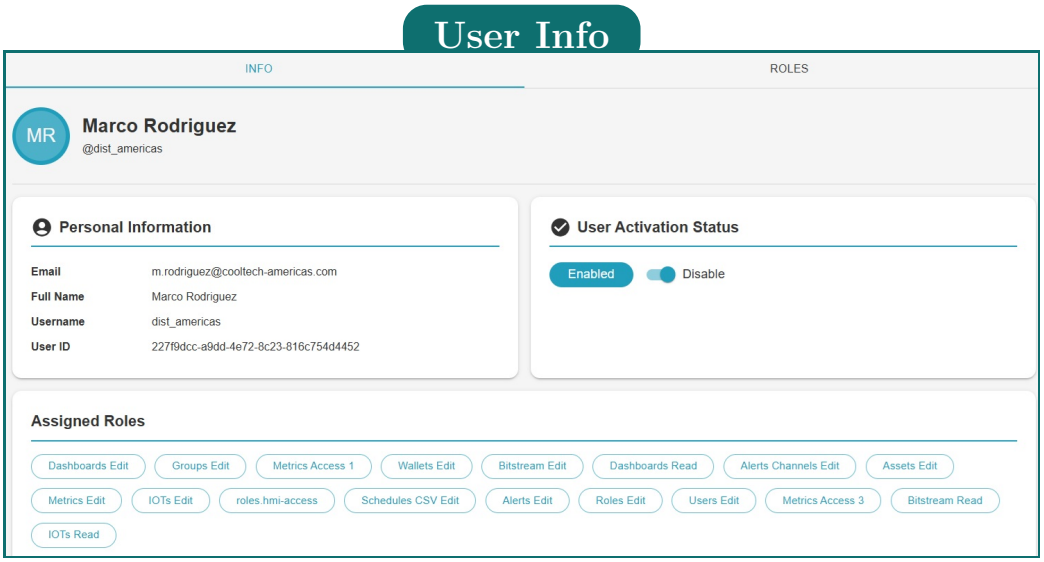

Add User
Name
Marco
Surname
Rodriguez
Email
m.rodriguez@cooltech-americas.com
Username
dist_americas
Password
••••••
Confirm Password
••••••
Group
cooltech-americ
SAVE
CANCEL

1.2.2 Info

This section displays the detailed profile of a registered user. It includes:

- **Personal Information** for identification.
- **User Activation Status**, where administrators can enable or disable the user account.
- **Assigned Roles**, which define what sections or data the user can access, such as `Dashboards Read`, `Alerts Read`, or `Metrics Read`.

Users can also change their password or edit their profile from this panel. This centralized view helps administrators manage access and privileges easily and securely.



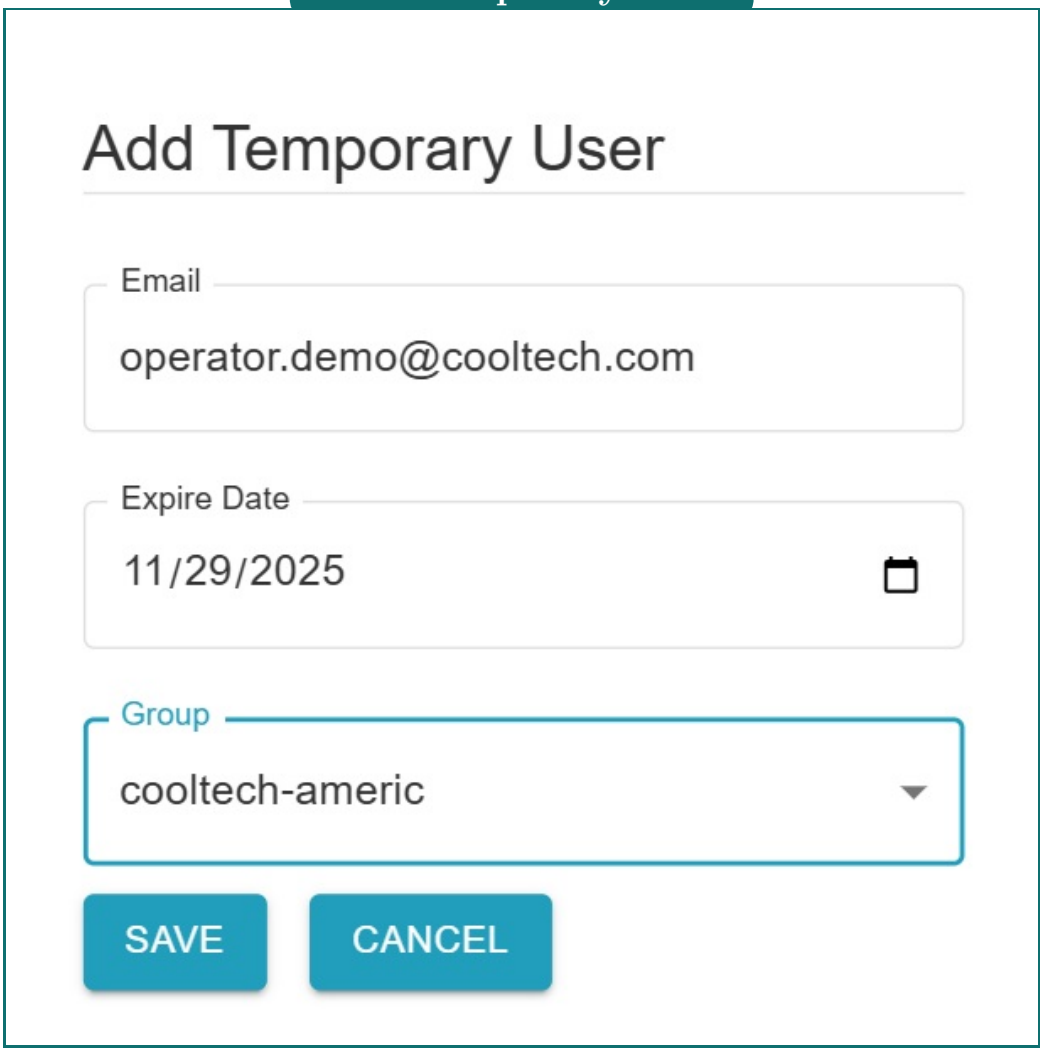
1.2.3 + ADD TEMPORARY USER

To add a temporary user to the system:

- Enter the **Email** of the temporary user.
- Set the **Expire Date**, after which the user will no longer have access.
- Choose the appropriate **Group** from the dropdown menu.
- Click `SAVE` to confirm or `CANCEL` to abort the process.

Once saved, the user will appear in the user list with an expiration label showing the remaining days.

Add Temporary User



1.3 Action

1.3.1 Info

The system will manage temporary users by automatically disabling them when the expiration date is reached. Details such as roles and activation status can be viewed in the user profile.

In the **Temporary User** section of the platform, two main information panels are presented:

- **Personal Information:**
  - **Email**: The email address registered to the temporary user.

  - **Full Name**: The user's complete name (auto-generated if not entered manually).
  - **Username**: A unique identifier used for login purposes.
  - **User ID**: A system-generated unique ID to distinguish users in the backend.
  - **Expire Date**: Shows how many days remain before the temporary account is automatically deactivated. This ensures limited-time access for temporary users.
- **User Activation Status:**
  - This panel controls whether the temporary user's access is currently **enabled** or **disabled**.
  - A toggle button is provided:
    * **Enabled** (green): The user is currently active and has platform access.
    * **Disabled** (Gray): The user is blocked from accessing the platform, without deleting the account.
- **Assigned Roles:**
  - **Groups Read**: Allows the user to view user groups and their details.
  - **Users Read**: Grants access to view registered users on the platform.
  - **Schedules CSV Read**: Enables the user to read and download scheduled data in CSV format.
  - **Alerts Channels Read**: Allows access to view configured alert channels and settings.
  - **Dashboards Read**: Permits the user to view visual dashboards and monitoring tools.
  - **Wallets Read**: Allows the user to view digital wallets or credit balances.
  - **Metrics Read**: Grants permission to access performance and usage metrics.
  - **Roles Read**: Enables viewing of all defined system roles and permissions.
  - **Assets Read**: Permits access to view all registered assets or devices.
  - **Bitstream Read**: Allows the user to read raw data streams from devices.
  - **IOTs Read**: Grants visibility into connected IoT devices and configurations.
  - **Alerts Read**: Provides access to view generated alerts and their statuses.
  - **Groups Edit**: Allows modification of group settings and user assignments.
  - **Users Edit**: Enables the user to edit, update, or deactivate user accounts.
  - **Schedules CSV Edit**: Permits editing of scheduled data exports.

- **Alerts Channels Edit**: Grants permission to modify alert channel settings.
- **Dashboards Edit**: Allows users to customize or create dashboards.
- **Wallets Edit**: Enables editing of wallet balances or transaction settings.
- **Metrics Edit**: Allows updating or configuring metric sources and thresholds.
- **Roles Edit**: Grants permission to modify system roles and their privileges.
- **Assets Edit**: Enables editing asset/device configurations or properties.
- **Bitstream Edit**: Allows the user to adjust raw data stream settings.
- **IOTs Edit**: Permits configuration of connected IoT devices.
- **Alerts Edit**: Enables modification or acknowledgment of alerts.
- **Wallets Products Edit**: Allows editing of linked wallet products or subscriptions.

These two panels give administrators an easy and fast way to **view essential user data** and **control account status**, especially for short-term or limited-access roles.

1.3.2 Password Reset via Random Generation

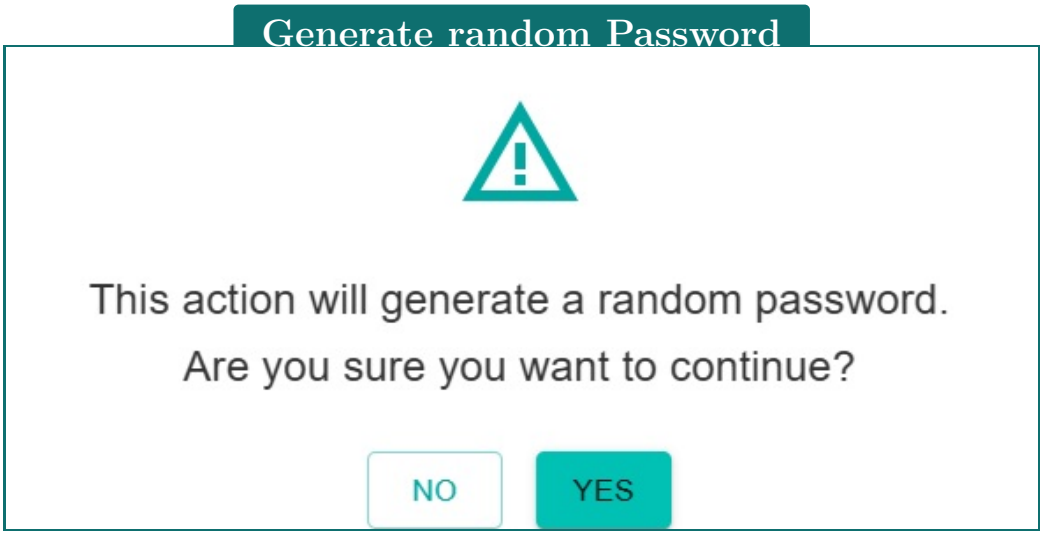
In case a user forgets their password, the system allows administrators to reset it using a secure random password generation method.

- Navigate to the **Users** section.
- Select the user whose password needs to be reset.
- Click on the **Change Password** button.
- A warning prompt appears:

  "This action will generate a random password. Are you sure you want to continue?"

- The administrator can then choose:
  - **YES** – to confirm and generate a secure random password for the user.
  - **NO** – to cancel the operation and retain the current password.
- Once confirmed, the system generates a new password automatically and updates the user's credentials.

This method ensures account recovery while maintaining security through non-predictable password generation.

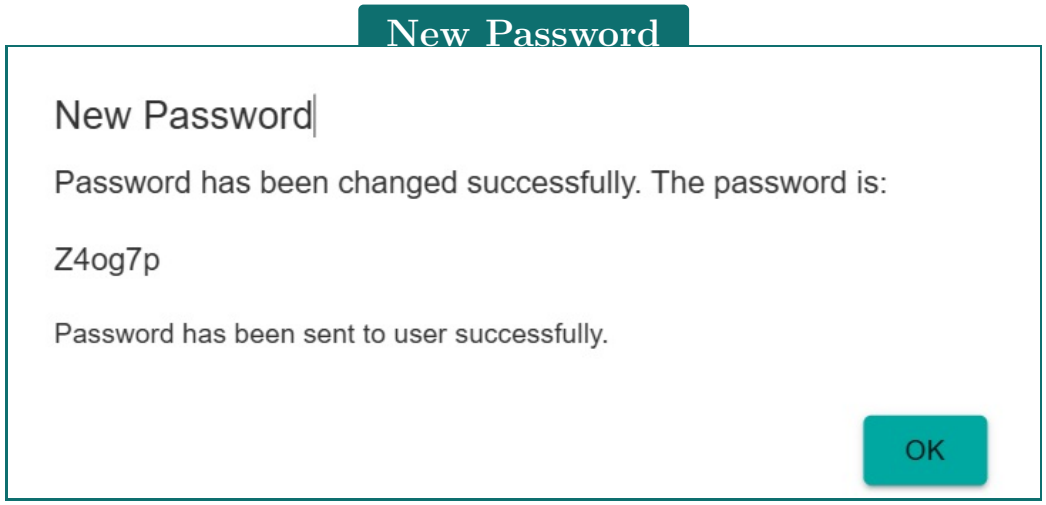


1.3.3 Random Password Confirmation

After confirming the random password generation by clicking **YES**, the system generates a secure password automatically.

- A success message appears stating that the password has been changed.
- The new randomly generated password is displayed on the screen.
- Example: `Z4og7p`
- A confirmation also notes: *"Password has been sent to user successfully."*

The administrator or user must copy and securely save the new password provided for **login**. After clicking the **OK** button, the password reset process is completed, and the new password is also sent to the user's **email**.



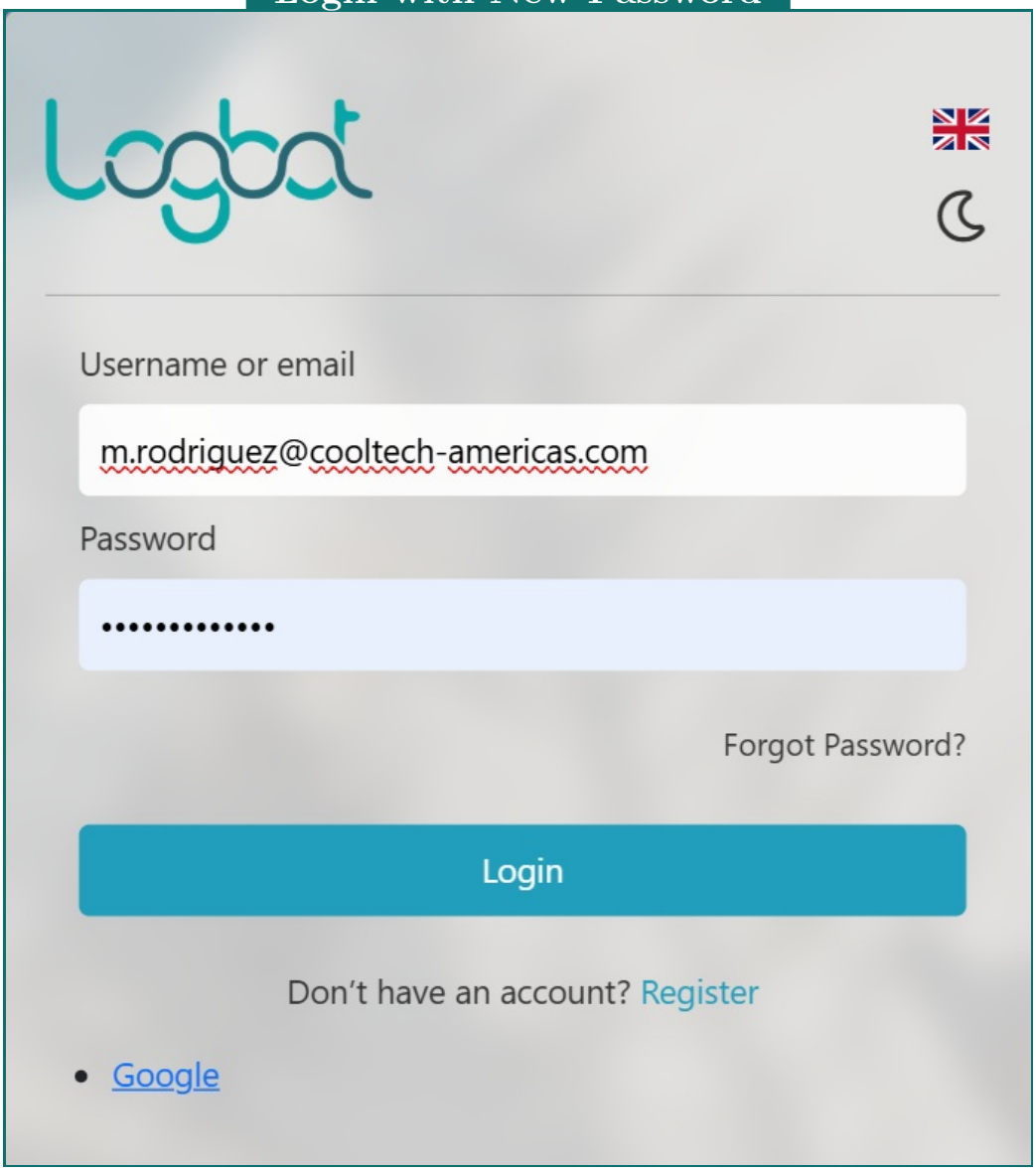
1.3.4 Login with New Password

Once the new random password is generated and shown to the administrator or user, it must be used to log in.

- Navigate to the login screen of the Logbot platform.
- Enter the temporary user's **username or email** in the appropriate field.
- Paste or type the **newly generated password** received from the password reset step.
- Click the **Login** button to access the account with the new credentials.

This step completes the password recovery process and enables the user to securely re-enter the system.

Login with New Password



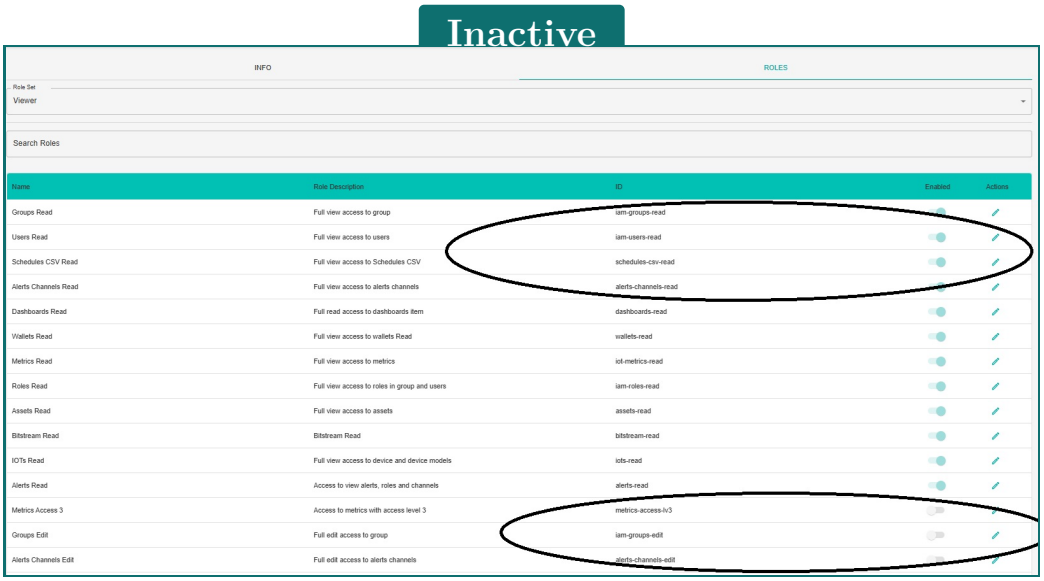
1.3.5 Roles

Login with new password and limited access permissions

- The **Assigned Roles** section lists all permissions granted to the user, determining their access level in the platform.
- If a user is assigned only `Read` roles, they can **only view** information. All action buttons (like edit or delete) will appear disabled.
- This is a security measure that restricts temporary or limited users from

making changes in the system.

- When a random password is generated for a new or temporary user, the default behavior often includes only `Read` roles — ensuring that the user must be reviewed and upgraded manually by an admin before being granted edit or admin privileges.
- Roles like `Groups Read`, `Users Read`, `Assets Read`, etc., allow the user to navigate and observe system data, but **not** to modify or interact with it in a functional way.



Read-Only Roles: These roles allow users to view content within specific modules of the platform without the ability to modify or interact with it. Users assigned only Read roles cannot perform edit or delete actions. This access type is typically assigned to observers, auditors, or new users awaiting full role assignment.

Table 1.1: Read-Only Access Roles

Name	Role Description	ID	Description
Groups Read	Full view access to groups	iam-groups-read	Grants visibility into user group structure without editing rights
Users Read	Full view access to users	iam-users-read	Allows inspection of user profiles and roles
Assets Read	Full view access to assets	assets-read	Permits viewing all registered platform assets
Dashboards Read	Full view access to dashboards	dashboards-read	Enables viewing analytics and dashboards only
Roles Read	View roles in group and users	iam-roles-read	Displays user role assignments and their hierarchy
Alerts Channels Read	View access to alert channels	alerts-channels-read	Lets users observe alert routing setup
Schedules CSV Read	View access to Schedules CSV	schedules-csv-read	Grants read-only access to schedule exports
Bitstream Read	Bitstream view access	bitstream-read	Allows monitoring of raw or stream-based data
Wallets Read	View access to wallets	wallets-read	Permits users to see wallet usage and balance info
Metrics Read	View access to metrics	iot-metrics-read	Displays collected device or asset metrics
IOTs Read	View device and model info	iots-read	Enables inspection of IoT device models and metadata
Alerts Read	View alerts, roles, and channels	alerts-read	View alerts log and routing config

1.3.6 Metrics Access Levels:

Metrics access is divided into three levels (LV1, LV2, and LV3). Each level grants visibility to progressively more detailed metrics:

- **LV1** – Basic-level metrics.
- **LV2** – Includes LV1 and additional intermediate metrics.
- **LV3** – Includes LV1 and LV2, plus advanced-level metrics.

Assigning a higher-level role automatically includes access to all lower levels.

Table 1.2: Metrics Access Levels

Name	Access Scope	Description	ID
Metrics Access 1	Can access metrics level 1 only	Basic-level metrics visibility	metrics-access-lv1
Metrics Access 2	Can access metrics levels 1 and 2	Adds intermediate-level metrics on top of Level 1	metrics-access-lv2
Metrics Access 3	Can access metrics levels 1, 2, and 3	Grants access to all metrics including advanced-level	metrics-access-lv3

1.3.7 Edit Roles (Full Access):

These roles grant users full permissions to modify, create, or delete data within the respective modules. They are typically assigned to administrators, editors, or power users who manage the platform's content and configurations.

Table 1.3: Full Edit Access Roles

Name	Role Description	Description	ID
Groups Edit	Full edit access to groups	Allows creation, modification, and deletion of user groups	iam-groups-edit
Users Edit	Full edit access to users	Manage user details, status, and roles	iam-users-edit
Assets Edit	Full edit access to assets	Add, edit, or remove asset records	assets-edit
Dashboards Edit	Full edit access to dashboards	Customize and modify dashboard widgets and data	dashboards-edit
Roles Edit	Edit roles in group and users	Assign or revoke user permissions across groups	iam-roles-edit
Alerts Channels Edit	Full edit access to alerts channels	Configure alert routes and notification channels	alerts-channels-edit
Schedules CSV Edit	Full edit access to schedules CSV	Import/export and configure CSV-based schedules	schedules-csv-edit
Bitstream Edit	Edit bitstream data	Modify raw data streams or signal feeds	bitstream-edit
Wallets Edit	Edit access to wallets and products	Full wallet management including limits and types	wallets-edit
Wallets Products Edit	Edit access to wallets/products	Manage wallet-linked products and associations	wallets-products-edit
Metrics Edit	Edit access to metrics and models	Create or adjust metrics and analytical models	iot-metrics-edit
IOTs Edit	Edit device and model info	Update IoT device metadata and configurations	iots-edit
Alerts Edit	Edit alerts, roles, and channels	Full access to create or disable alerts	alerts-edit

1.3.8 Editor Role Assignment Workflow:

The **User Management** interface provides an intuitive way to assign roles to users. Under the **Roles** tab, administrators can assign predefined role sets. Two main options are available:

- **Editor** – Grants full edit and configuration permissions.
- **Viewer** – Provides read-only access to platform modules.

When the **Editor** role set is selected:

1. The section titled **Roles To Be Assigned** automatically populates with all available edit roles. These include:
   - *Dashboards Edit, Groups Edit, Assets Edit, Alerts Channels Edit, Users Edit*, etc.
2. Additionally, the user is granted the highest available **Metrics Access Level** — `Metrics Access 3`, which enables access to all levels of platform metrics (Levels 1, 2, and 3).
3. To apply the selection, the user must click the **Apply** button.

1.3.9 Customizing Role Activation:

Each role listed under the **Assigned Roles Table** can be individually enabled or disabled using the toggle switch in the **Enabled** column. Once configured:

- The user must click the **Save** icon under the **Actions** column to confirm and activate each role.
- Disabled roles can be activated later using the same process — enabling the toggle and saving.

This workflow ensures that users with the **Editor** role receive complete access privileges and can manage platform resources, users, alerts, schedules, and metrics comprehensively.

Chapter 1

Integrating with Grafana

1.1 LB Assets and Alarms Data Source

1.1.1 Grafana Query Language

Grafana

The Logbot platform supports Grafana queries to retrieve data about plants, gateways, devices, and alarms. This guide helps users understand the structure and purpose of each query.

1.1.2 Basic Structure

Grafana queries are composed of:

- **Non-functions**: entity names such as plants, gateways, or devices.
- **Functions**: start with an underscore (_) and define the action.

Syntax:

`[Plant].[Gateway].[Device]._Function`

Example:

```
plantA.gatewayX.deviceY._list
```

This query returns a list of all devices in gateway `gatewayX` under plant `plantA`.

1.1.3 Common Queries and Their Meanings

_plants Returns all available plants in the system.

```
_plants
```

_gateways Returns all gateways (across all plants).

```
_gateways
```

_devices Lists all devices in the system.

```
_devices
```

Example:

plantA._gateways Returns all gateways associated with `plantA`.

```
plantA._gateways
```

plantA.gatewayB._devices Lists all devices in gateway `gatewayB` in plant `plantA`.

```
plantA.gatewayB._devices
```

plantA.gatewayB.deviceC._list Displays a list of data or configurations for `deviceC`.

```
plantA.gatewayB.deviceC._list
```

1- Alarm Queries:

_active_alarm Shows all current active alarms.

```
_active_alarm
```

_historical_alarm Returns a list of past alarms.

```
_historical_alarm
```

__priority__N Filters alarms by priority level `N` (e.g., `_priority_1`).

```
_priority_1
_priority_2
```

2- Wildcard Queries

Use `*` to refer to all entities at that level.

Wildcard Example 1: Active status for all plants:

```
*._active
```

Wildcard Example 2: All devices under any gateway and any plant:

```
*.*._devices
```

3 - Multi-Selection Queries

Use the `(A|B)` form to request data from multiple entities.

Multiple Plants Device lists from two plants:

```
("Plant1"|"Plant2").gatewayX._devices._list
```

1.2 Grafana JSON Data Source

1.2.1 _alias function

_alias Function

Use the field `_alias` to set a *custom title* for a panel directly from your JSON query payload. This value overrides the default title displayed in the Grafana panel.

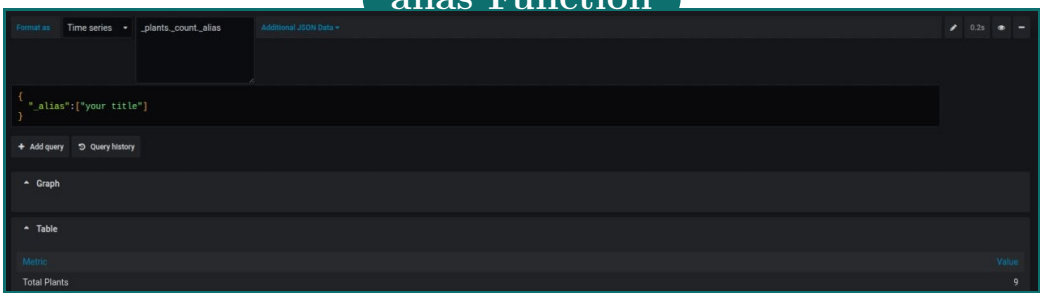
In this example JSON payload:

```
{
  "_alias": { "my title": "" }
}
```

Notes

- Replace `"my title"` with your desired display name.
- The empty string `""` as value is acceptable when only the key matters.
- Keep your payload valid JSON (matching braces and quotes).

alias Function



Common Queries

Typical endpoints used in this dashboard:

- `_plants_list_count` (returns total number of plants.)
- `_plants_list` (returns plant rows:) `id`, `name`, `address`, `desc`.
- `plt_test_1v._gateways_list` (returns gateways for a given plant.)

1.2.2 __plants__list__count

The `_plants_list_count` query quickly returns the **total count of plants** in the platform.

This query is helpful for:

- Getting a quick overview of how many plants are being monitored.
- Performing system audits to confirm all expected plants are registered.
- Comparing counts before and after data imports to ensure completeness

1.2.3 __plants__list

Retrieves a detailed table of all registered plants (ID, name, address, description).

- **Plant ID** — A unique numerical identifier assigned to the plant.
- **Plant Name** — The descriptive name given to the plant.
- **Address** — The plant's physical location, which may include GPS coordinates.

- **Description** — Any additional notes or context provided by the user.
- **Usage Scenario:** This query is often used by operators and administrators to:
- **Identify and confirm plant entries**.
- **Cross-check plant addresses for mapping or logistics.**
- **Review descriptive notes for operational context.**

The figure shows multiple sample entries, such as '**plant_test**' and '**testTags**', along with their **IDs, addresses, and descriptions**.

1.2.4 plt_test_1v._gateways_list

Returns all gateways associated with the selected plant.

The '<plant_name>._gateways_list' query is used to retrieve all gateways that are associ ated with a specific plant. Replace '<plant_name>' with the actual plant name from your system.

The table output contains:

- **Gateway Name** — A unique identifier for the gateway device.
- **Plant Name** — The plant to which the gateway belongs.
- **Status** — The operational state of the gateway (activated or deactivated).

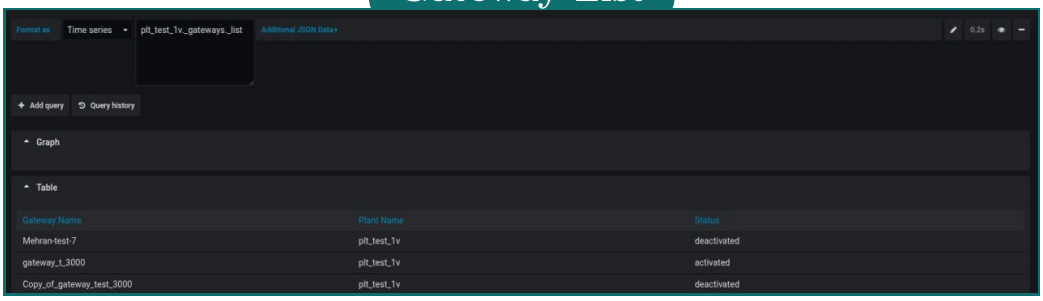
Usage Scenario: Operators use this query to:

- Check which gateways are linked to a particular plant.
- Verify that all expected gateways are active.
- Identify inactive gateways for troubleshooting.

In the example,

the plant 'plt_test_1v' has three gateways, with two in a '**deactivated**' state and one '**activated**'.

Gateway List



Function Reference Table

Entity	Functions
Plants	_list, _alias, _count, _active_alarm, _historical_alarm, _priority_N
Gateways	_actives, _inactives, _list, _active, _inactive, _online, _offline, _licensed, _unlicensed, _alias, _count, _active_alarm, _historical_alarm, _priority_N
Devices	_list, _active, _inactive, _alias, _count, _active_alarm, _historical_alarm, _priority_N

Table 1.1: Available functions by entity

Notes

All function names begin with an underscore. In paths they are appended to the entity, e.g., `plant_list`, `gateway_online`, `device_active_alarm`.

Function Descriptions:

- **_list**: Retrieves the complete list of entities of the specified type.
- **_active / _inactive**: Returns entities that are currently active or inactive.
- **_actives / _inactives** (Gateways only): Batch queries for active/inactive gateways.
- **_online / _offline** (Gateways only): Filters gateways by online/offline connectivity
- status.
- **_licensed / _unlicensed**: Filters entities based on license availability.
- **_alias**: Retrieves alternate display names or aliases for the entity.

- **__count**: Returns the total number of entities for the given type.
- **__active__alarm**: Lists currently active alarms for the entity.
- **__historical__alarm**: Retrieves past alarms for the entity.
- **__priority__N**: Filters alarms by priority level N (replace N with the priority number).

1.3 Historical Alarms JSON

User JSON (verbatim)

```
{
  "historicalAlarmFilter": {
    "_from": "2025-07-26T00:00:00",
    "_to":   "2025-07-26T23:00:00",
    "_limit_groups": 5
  },
  "tag_keys":   ["key1", "key2"],
  "tag_filter": [["key2", "val2"]],
  "alias": "historical sample alias"
}
```

How to apply it (panel setup)

1. Panel → *Query editor*.
2. Choose the *LB / JSON* data source.
3. Paste the JSON into **Additional JSON Data**.
4. Use a metric for historical alarms, e.g. `$plants.$gateways.*._historical_alarm`.
5. Panel type: **Table** (lists) or **Stat** (counts).

What the user will see

- Historical alarms for **26/07/2025, 00:00–23:00**.
- Rows restricted to `key2 = val2`.
- Columns include the requested tags: `key1`, `key2`.
- Series/table labelled *historical sample alias*.

1.4 Grafana Variables for LB Assets & Alarms

Variables:

Think of variables as *drop-down selections* the dashboard needs (for example: which Plant, which Gateway, which Device). When you pick values in the drop-downs, Grafana puts those choices into the query automatically.

Panels:

A panel is a chart or a table. Panels read your variable choices and then ask the data source for the right data.

we'll create three drop-downs (`$plants`, `$gateways`, `$devices`) and then build two panels. This section explains how to create the dashboard variables used in the queries: `$plants`, `$gateways`, and (optional) `$devices`.

1.4.1 `$plants` — Plant selector (first step)

Name	`plants`
Type	`Query`
Data source	`logbotJson`
Query	`_plants_list`
Regex	`/(.*)/` *(pass-through; or use* `^([^.]+).*$` *to keep only the plant token)*
Sort / Multi / All	*Alphabetical (asc)*; *ON*; *ON*

Open *Dashboard* → *Settings* (gear icon) → *Variables*.

1.4.2 $plants — "Which plant?"

Purpose. Lets the user choose one or more plants. Other drop-downs will depend on this.

Do this (click-by-click)

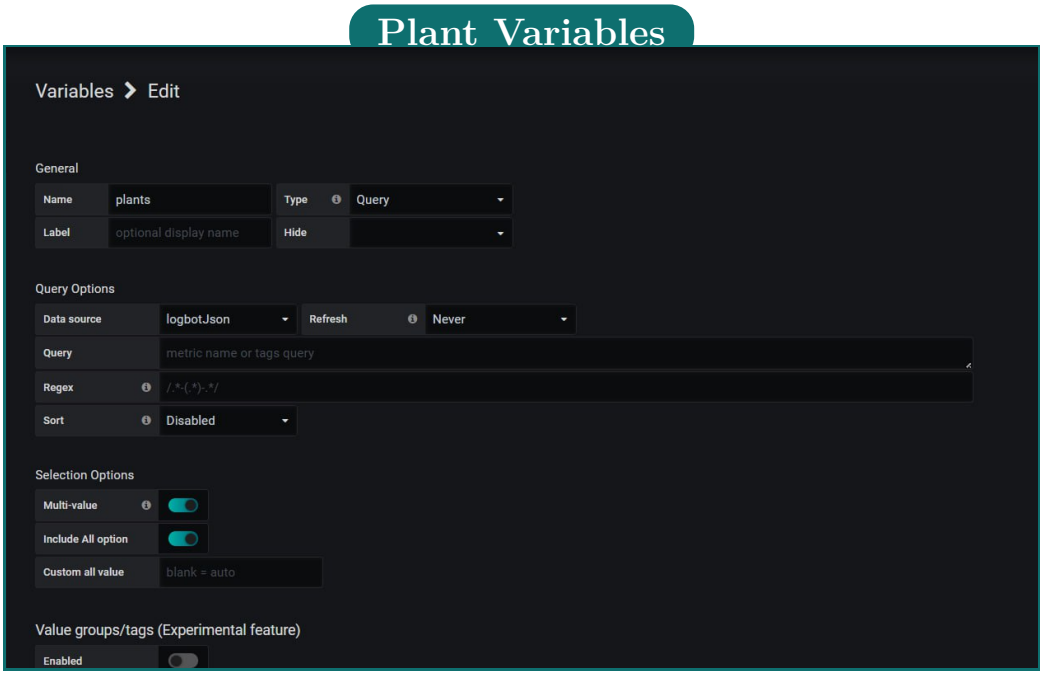
1. Click **Add variable**.
2. **Name:** `plants` **Type:** `Query`
3. **Data source:** `logbotJson`

4. **Query:** `_plants_list` (if you are unsure, start with this; it's the safest)
5. **Regex:** `/(.*)/` (keeps the full plant name as it is)
6. Turn **Multi-value** ON. You may also turn **Include All** ON.
7. **Save**.

What you should see A screen like FigurePlant.Alarm. The drop-down will show plant names (e.g. *plt_test_1v*).

If it doesn't work

- The list is empty? Set **Query** to `_plants` or `*._plants` and try again.
- You see long dotted paths? Keep it for now; we'll strip dots in the next variable.



1.4.3 $gateways — Gateway selector

Name	`gateways`
Type	`Query`
Data source	`logbotJson`
Query	`$plants.`
Regex	`^.*\.(.*)$` *(keeps the token after the last dot)*
Sort	`Disabled` *(or Alphabetical (asc))*
Multi / All	*ON / ON (as desired)*

1.4.4 $gateways — "Which gateway?"

Purpose. Show only the gateways that belong to the selected plant(s).

Do this

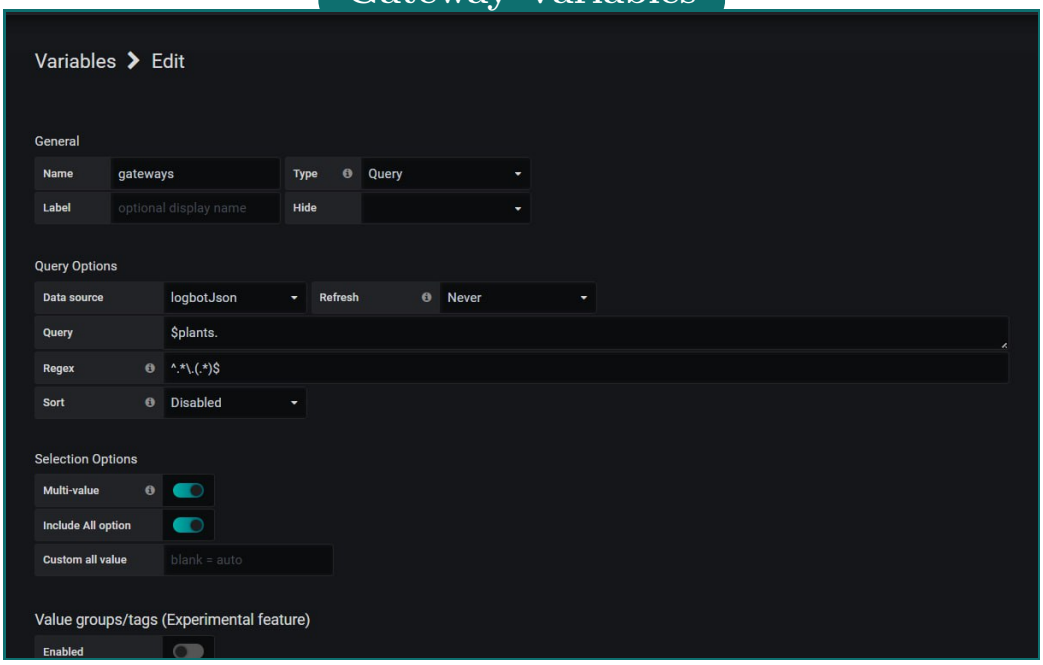
1. **Name:** `gateways` **Type:** `Query`
2. **Data source:** `logbotJson`
3. **Query:** `$plants.` (the dot means "list what is under this plant")
4. **Regex:** `^.*\.(.*)$` (keeps only the text after the last dot, so you see clean names)
5. Turn **Multi-value** ON if you want to pick several gateways.
6. **Save**.

A screen like Figure Gateway.Alarm. When you open the drop-down, gateway names appear without dots.

If it doesn't work

- You still see dotted paths? Double-check the **Regex**—copy it carefully.
- Empty list? Pick a plant first from `$plants`. The gateway list depends on it.

Gateway Variables



1.4.5 `$devices` — Device selector (optional)

Name	`devices`
Type	`Query`
Data source	`logbotJson`
Query	`$plants.$gateways.`
Regex	`^(?:[^.]+\.){2}(.*)` *(captures the third token)*
Sort / Multi / All	`Disabled`; optional; optional

1.4.6 $devices — "Which device?" (optional)

Purpose. Show devices for the chosen plant *and* gateway.

Do this

1. **Name:** `devices` **Type:** `Query`
2. **Data source:** `logbotJson`
3. **Query:** `$plants.$gateways.`
4. **Regex:** `^(?:[^.]+\.){2}(.*)` (keeps the third token = device)

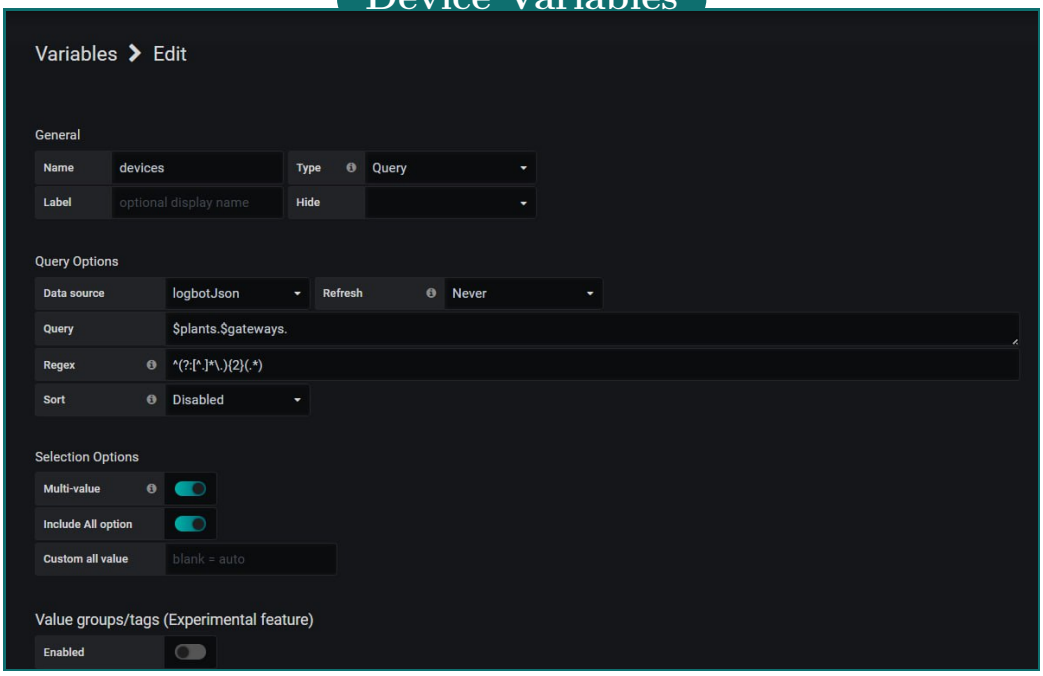
5. Multi-value is optional. **Save**.

A screen like Figure **??**. The list shows device names (no dots).

If it doesn't work

- Make sure you have picked both a *Plant* and a *Gateway*.
- Still empty? Temporarily remove the regex to check what raw values come back.

Device Variables



Using the Variables (Examples)

- Historical alarms per gateway: `$plants.$gateways.*._historical_alarm`
- Active alarms per device: `$plants.$gateways.$devices._active_alarm`
- **List gateways:** `$plants._gateways` or `$plants._gateways_list`

1.5 Active Alarms Table (Reference)

A standard panel listing **currently active alarms** for the selected scope.

Goal:

Show a live table of currently *active* alarms, filtered by your drop-downs and by priority.

Do this

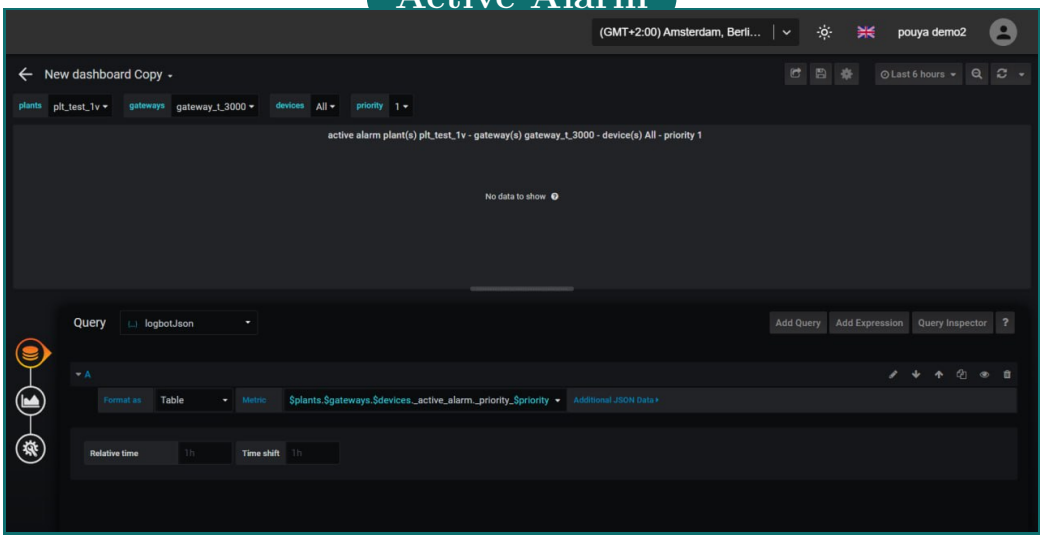
1. Add a new **Panel**.
2. **Data source:** `logbotJson`.
3. **Format as:** `Table`.
4. **Metric:**
   `$plants.$gateways.$devices._active_alarm__priority_$priority`
5. Pick a **Relative time** that suits operations (e.g. Last 1h, Last 6h).
6. **Apply**.

A table like Figure **??**. If it says "No data", see the fixes below.

If it doesn't work

- Try replacing `$devices` with `*` to include *all* devices: `$plants.$gateways.*._active_alarm__`
- Make **Relative time** larger (e.g. Last 24h).
- Clear the priority filter (or set it to a different value).

Active Alarm



1.6 Panel 2 — Historical Alarms (One Day, With Tag Filter)

Goal:

Show a table of *past* alarms for a specific day, optionally filtering by tags.

Do this:

1. Add a new **Panel**.
2. **Data source:** `logbotJson`.
3. **Format as:** `Table`.
4. **Metric (scope example):** `$plants.$gateways.*._historical_alarm`
5. Click **Additional JSON Data** and paste:

```
{
  "historicalAlarmFilter": {
    "_from": "2025-08-13T00:00:00",
    "_to":   "2025-08-13T23:00:00",
    "_limit_gtoups":
  },
  "alias": "historical",
  "tag_filter": [["key2", "val2"], ["key1", "val1"]]
}
```

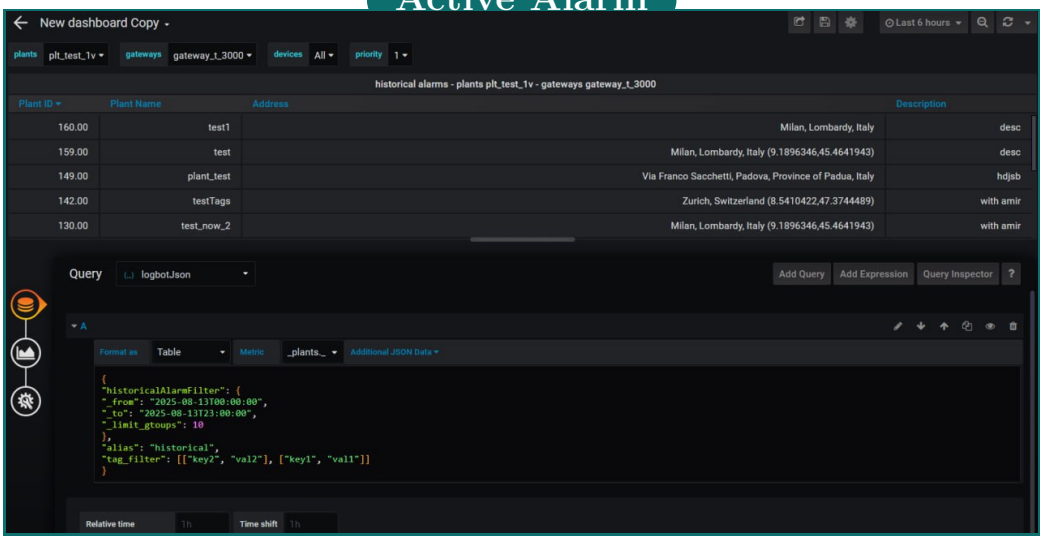
6. **Apply**.

A table similar to Figure **??**. Rows appear only for the chosen day and tags.

If it doesn't work

- Remove `"tag_filter"` temporarily to check if any rows exist.
- Widen the time range (change `_from`/`_to` to a bigger window).
- If the table is huge, lower `_limit_gtoups` or narrow scope (pick a single gateway).

Active Alarm



1.7 Quick Testing in Explore (Autocomplete Helps)

Use **Explore** with data source `logbotJson`.

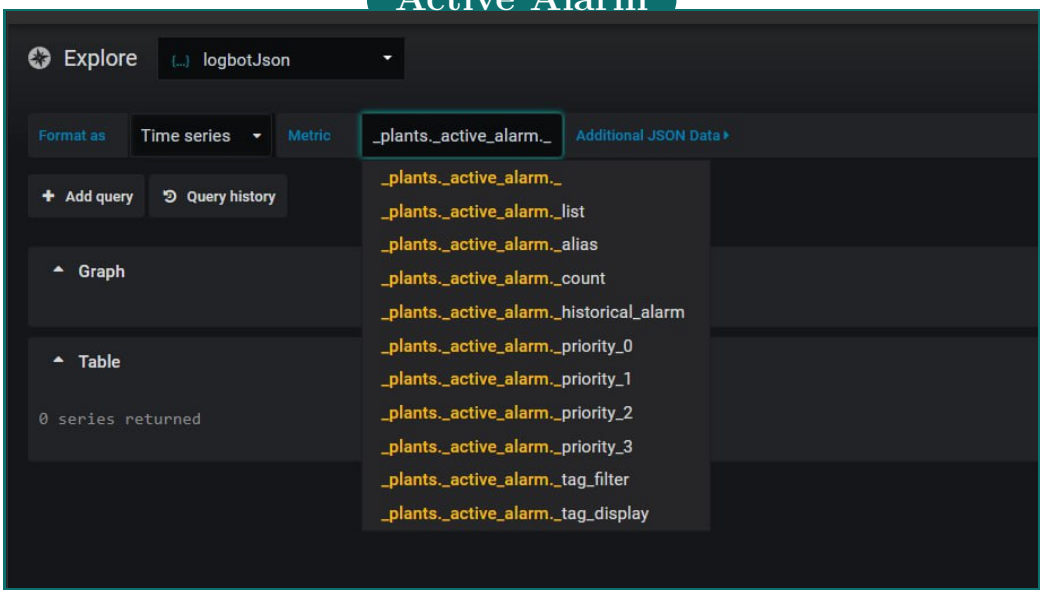
In the Metric box, start typing:

`_plants._active_alarm_`. You'll see suggestions such as: `__list`, `__count`, `__priority_0` ... `__priority_3`, `__tag_filter`, `__tag_display`.

Why this is useful

- Confirms that a path actually exists for your selections.
- Shows the function endings ("suffixes") you can use in panels.

Active Alarm



1.8 Common Problems and Fast Fixes

Nothing shows in the table

Try these in order:

1. Increase the **Relative time** range (e.g., *Last 24h*).
2. Replace `$devices` with `*` to include all devices.
3. In the Historical panel, temporarily remove `tag_filter`.
4. Re-select `$plants` and `$gateways` to refresh the query scope.

Drop-downs show weird dotted text

Keep the **Query** but fix the **Regex**:

- Gateways: `^.*\.(.*)$` (keeps the part after the last dot)
- Devices: `^(?:[^.]+\.){2}(.*)` (keeps the third part)

I don't know which metric to type

Open the **Explore** view and let the picker show valid endings such as `__list`, `__count`, or `__priority_1`. Copy one that matches your data field.

Chapter 1

Managing Schedules

1.1 Schedules

Schedules automate tasks across devices, allowing users to define start times and repeat intervals. Each schedule includes a title, run history, and device targeting.

Schedules

The **Schedules** section allows users to automate queries and notifications at fixed intervals. Instead of executing queries manually, schedules ensure that data is collected and reported continuously.

Creating a New Schedule

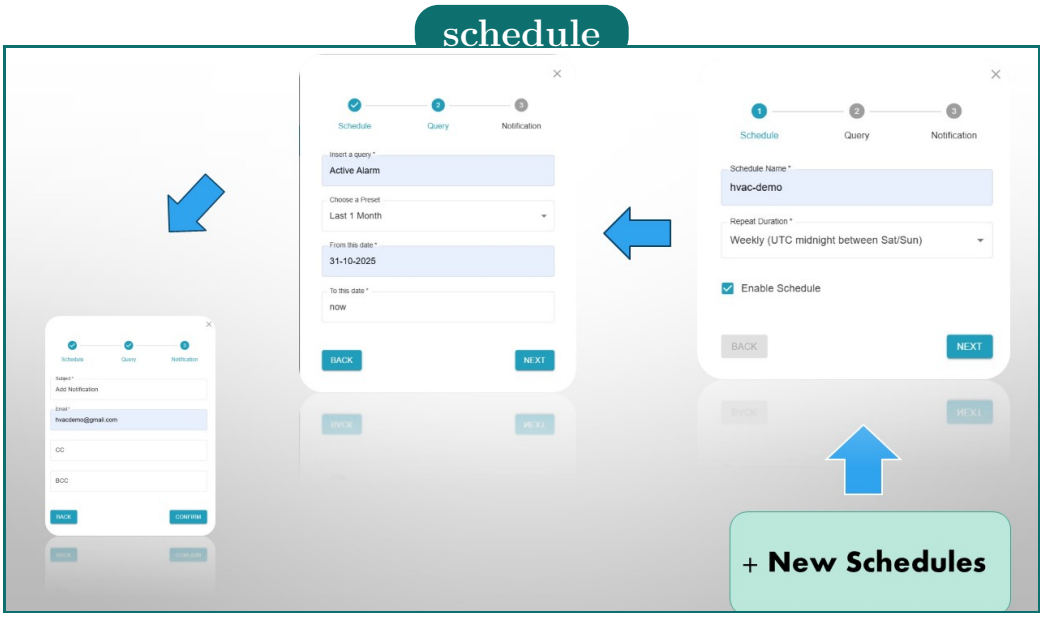
- Click on **"+ New Schedule"** to open the configuration wizard.
- The setup includes three main steps:
  1. **Schedule**: Define the name and repetition frequency.
  2. **Query**: Select the data source or panel to execute.
  3. **Notification**: Configure how results are sent.

Schedule Options

- **Schedule Name**: A required identifier for the schedule.
- **Repeat Duration**: Choose from:
  - Daily (UTC midnight)
  - Weekly (UTC midnight between Sat/Sun)
  - Monthly (UTC midnight, first day of month)
- **Enable Schedule**: Toggle to activate or deactivate the schedule.

Managing Schedules

Created schedules are listed in the overview panel, where users can edit, enable/disable, or delete them. This ensures continuous monitoring and automated delivery of results through configured notification channels.



Chapter 1

Billing & Settings

1.1 Billing

Tracks user platform usage via Logbot credits. Users can view transaction history, buy credits, or redeem coupons through a wallet-style dashboard.

Billing Panel

The **Billing Panel** allows users to manage their Logbit credits, review past transactions, and add funds to their account. This section ensures full visibility into credit usage and provides multiple ways to maintain account balance.

Overview

The Billing page displays the following components:

- **Balance:** Shows the total available Logbit credits.
- **Last Month:** Indicates the amount of credits used during the previous month.
- **Buy Credits:** Button to purchase additional Logbit credits.
- **Redeem Coupon:** Option to enter a coupon code to recharge your wallet.

Wallet History

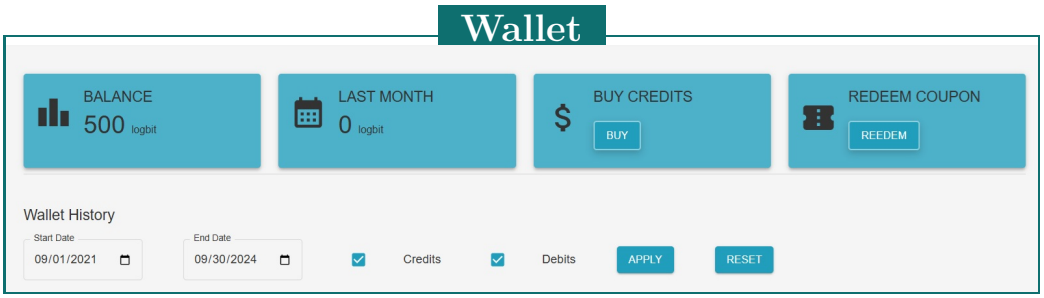
The **Wallet History** provides a transaction log with detailed information about credits and debits.

- Use the **Start Date** and **End Date** fields to filter transactions by period.
- Enable or disable the **Credits** and **Debits** checkboxes to refine results.
- Click **Apply** to refresh results or **Reset** to clear filters.

Adding Credits

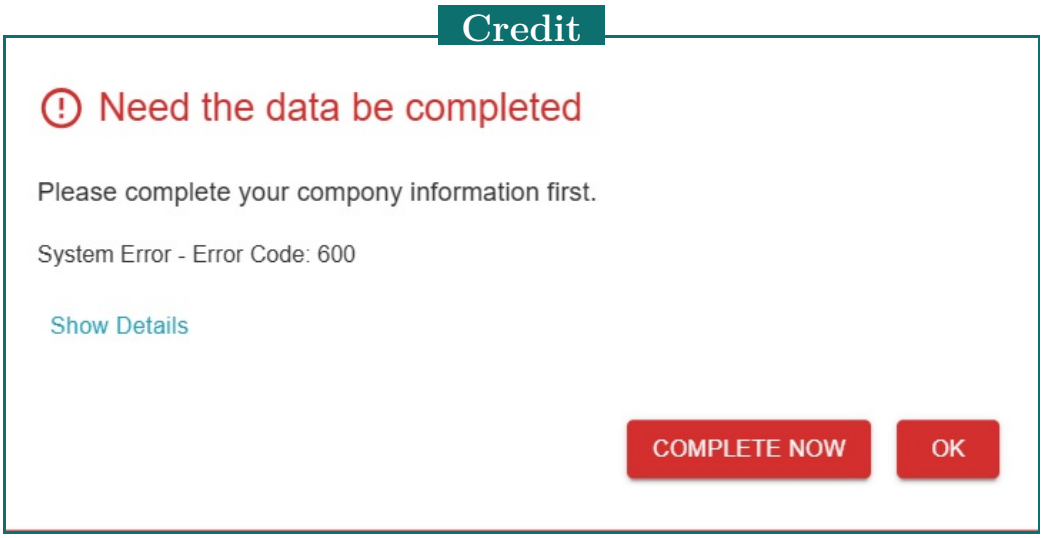
Users can increase their wallet balance in two ways:

- **Buy Credits:** Opens a payment form to purchase Logbits directly.
- **Redeem Coupon:** Enter a valid coupon code in the pop-up dialog and press **Submit**.



Error Notifications

If the billing profile is incomplete, the system blocks transactions and displays an error message:



1.2 Settings

1.2.1 Settings

The **Settings** section allows users to configure their company profile and billing details. This information must be completed before financial operations, such as *redeeming coupons* or *purchasing credits*, can be performed.

Available Fields

- **Company Name** – Name of the company or organization.
- **Tax Code** – Tax identification code of the company.
- **VAT Number** – Value Added Tax number, required for billing compliance.
- **SDI Number** – System interchange code (used in Italy).
- **Nation** – Country of the company.
- **Region / City / Road / ZIP Code** – Full company address.

Saving Changes

After entering the required data, click on the **Edit** button at the bottom of the page to save the changes. If the information is incomplete, billing features will not be available and an error message will be displayed.

Chapter 1

Troubleshooting Common Errors

1.1 Platform Errors and Troubleshooting

Gateway Activation: When creating a Gateway (GT) in the Logbot platform and attempting to activate its status, the system may show the following error:

1.2 Gateway Model has no transform

1. Gateway Model has no transform" – To resolve this:
   - Go to the Gateway Model section.
   - Edit the configuration by setting the Transform to the latest available version (e.g., Iot-config-2.2.0).

1.3 Gateway not connected to a Plant

2. After updating the transform, the system may prompt: Gateway not connected to a Plant: Then, it may show:

1.4 No Active Device Found

   - Ensure at least one active Device is created and assigned to the gateway.

3. Finally, you may see:

1.5 No Metric Found

- Go to the Device Metric section.
- Use Edit → Advanced, and add at least one Metric.
  Once a valid Transform, Plant, Device, and Metric are configured, the Gateway status can be successfully activated, confirming full connectivity.

1.6 License type not Allowed

When attempting to connect a license to a gateway model, the following error may appear:

> **License Type Not Allowed**
> *System Error - Error Code: 63500*
> License and Gateway Model must connect to the same Supported Gateway.

Cause:
This error occurs when the selected License and Gateway Model are not compatible. Specifically, the system checks that both resources:

- Belong to the **same License Type**.
- Have matching values for **Supported Hardware**.

License Incompatibility Example:

- **License Type:** `lba_iot3000_mehran`
- **Connected Gateway:** `Not Connected`

In contrast, a gateway model such as `gateway2` may only support:

- **Supported Hardware:** `lba_000000000000_cgatet000`

If you attempt to bind these mismatched entries, the operation will fail with status code 400.

Solution:

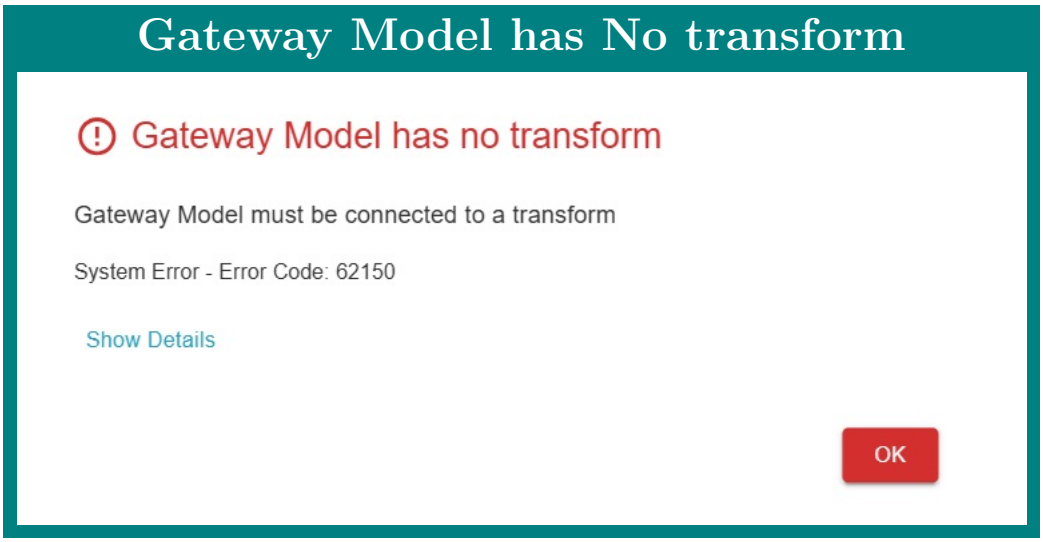
- Make sure to select a **Gateway Model** that has the same **Supported Hardware** as the **License Type**.
- For example, a license of type `lba_iot3000_mehran` should only be connected to gateway models where the supported hardware is also `lba_iot3000_mehran`.
- Avoid mixing License Tokens of different formats (e.g., `4:...` vs. `1:...`) if your platform distinguishes token schemes.

Recommended Gateway Models for Each License:

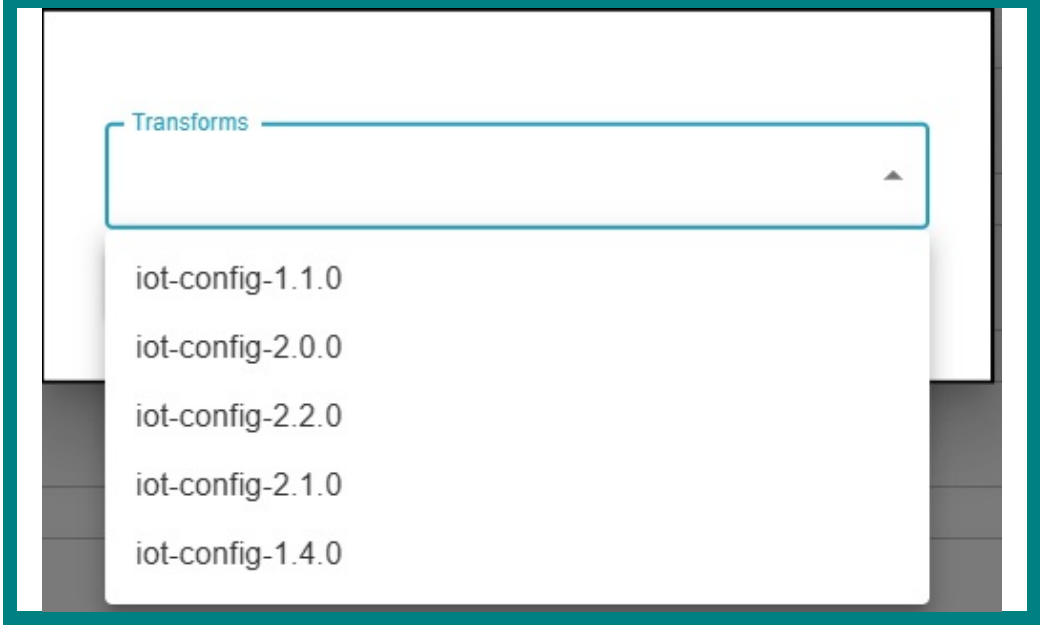
License Type	Compatible Gateway Models
`lba_iot3000_mehran`	Gateway4, Log, Gt3
`lba_000000000000_cgatet000`	gateway2

Table 1.1: Gateway Model Compatibility by License Type

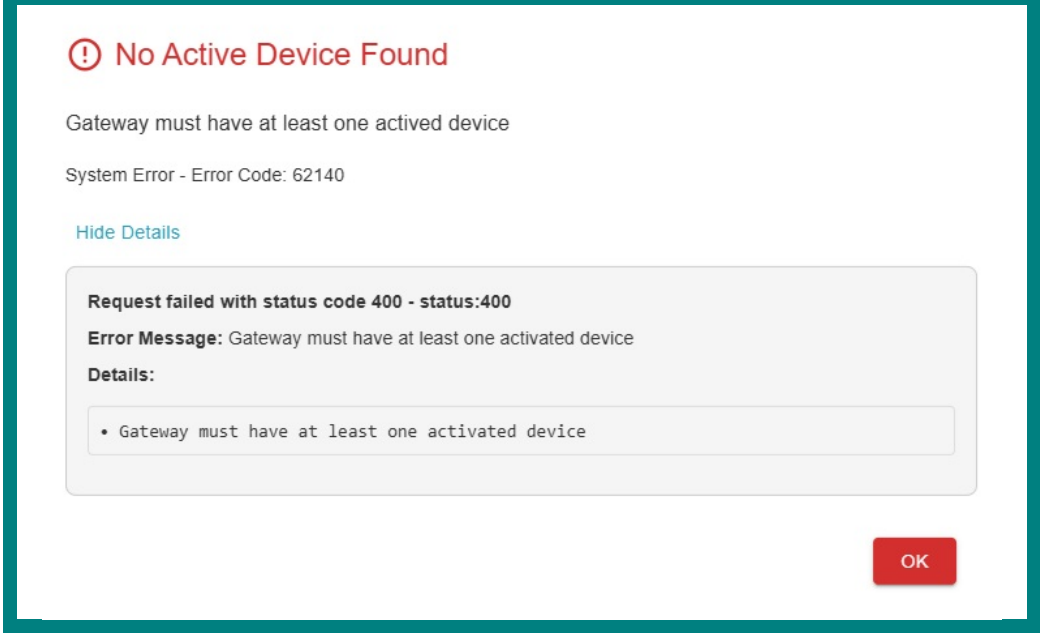
By aligning your license selection with the appropriate gateway model, you can avoid connection issues and ensure successful device activation.



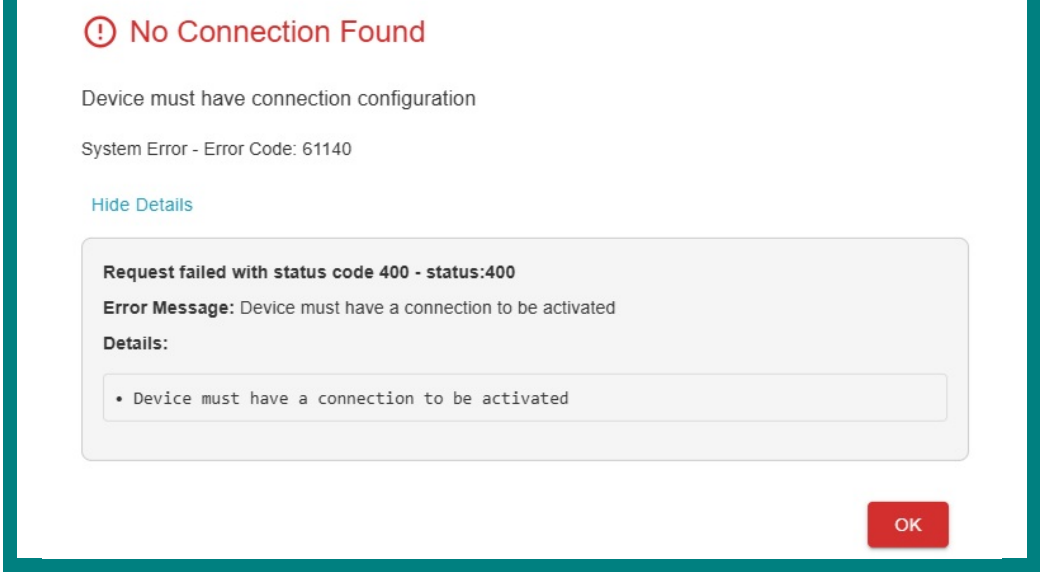
GT model error



No active device find

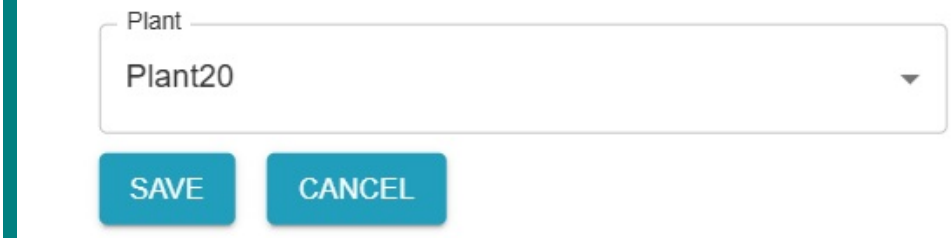


No connection found

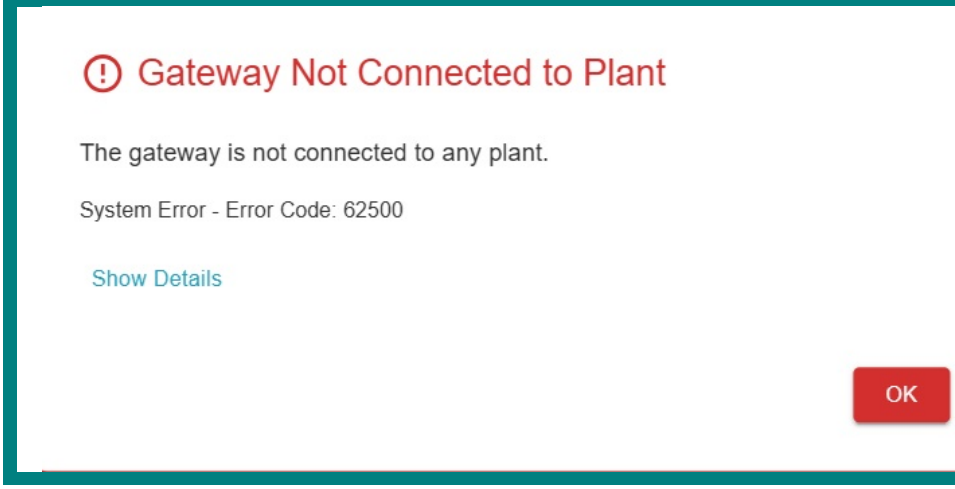


No metric found

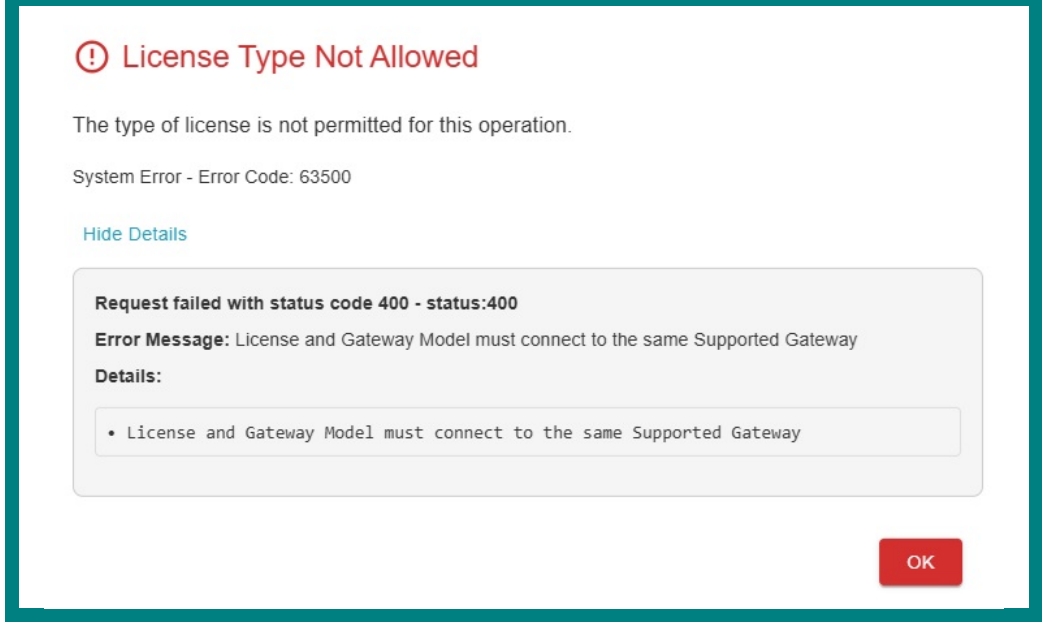
Plant



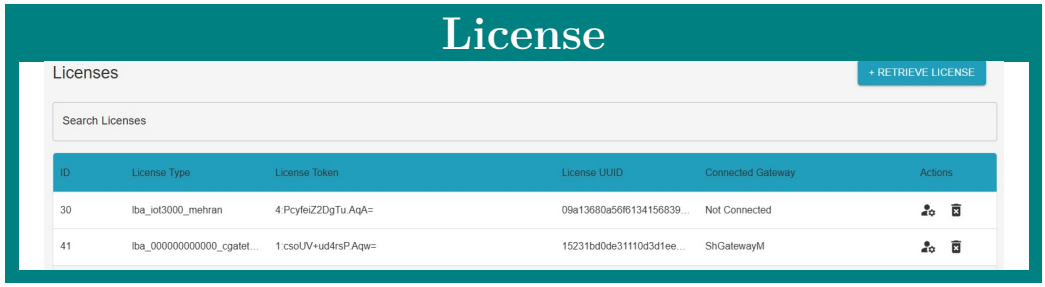
Not Connected to Plant



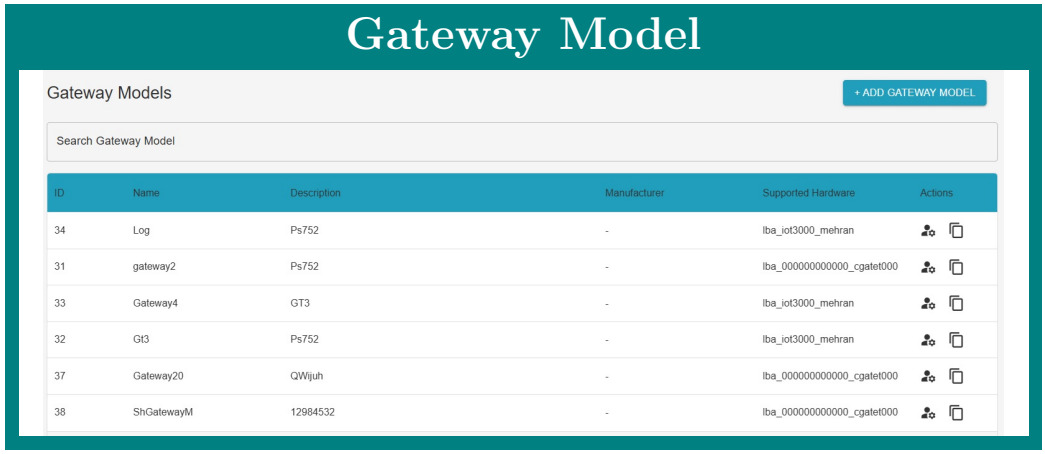
License Type Not Allowed



License



Gateway Model



Chapter 1

Glossary of Terms

This glossary defines key terms used throughout the Logbot platform and this manual. Entries are listed alphabetically.

Asset Any manageable component within the Logbot platform, including *Plants*, *Gateways*, and *Devices*.

Device The physical or logical endpoint that generates data (e.g., a sensor, PLC, motor). Devices connect to a *Gateway*.

Device Model A digital template defining a device's properties, including its protocol (e.g., Modbus, S7) and available metrics.

Gateway A hardware or software component that acts as a secure bridge between Devices and the Logbot platform.

Gateway Model A template used for configuring Gateways, specifying the supported hardware and the configuration version (*Transform*).

Group A collection of users, organized to manage permissions and control access to specific Plants.

Group Code A unique, shareable code that allows new users to request membership in a specific Group during registration.

License A digital token required to authorize and activate a Gateway on the platform.

Metric A single stream of data from a Device (e.g., temperature, pressure, status).

Plant	The highest-level organizational container, typically representing a physical location like a factory or building.
Protocol	A set of rules governing data exchange between devices (e.g., Modbus RTU, Siemens S7, OPC UA).
Role	A named set of permissions (e.g., read, edit) that define a user's capabilities.
Scale & Offset	Mathematical values used to convert raw device data into human-readable engineering units.
Static Tag	A permanent, descriptive key–value pair of metadata attached to an asset for organizing and filtering.
Transform	A specific configuration script (e.g., `iot-config-2.2.0`) assigned to a Gateway Model that defines its core logic.

Chapter 1

Conclusion

1.1 conclusion

This documentation provides a complete walkthrough for configuring and managing industrial assets within the Logbot IIoT platform. By systematically addressing the setup of plants, gateways, devices, models, and licensing, it ensures seamless data communication and system integrity. Special emphasis was given to protocol-specific configurations, such as Modbus RTU and Siemens S7, to support diverse industrial environments. The user-centric interface, combined with logical workflows and detailed metric configurations, facilitates robust monitoring and control. Ensuring each component—from device registration to license activation—is properly configured guarantees reliable performance, making the platform a powerful tool for smart industrial asset management


Logbat
524 kW
152 kWh
InstantApp
Extractor
45:00
Levels
OK