Node graph

- Configure a node graph visualization
- Supported data formats
  - Nodes dataset
    - Example
  - Edges dataset
    - Example
  - Data requirements
    - Nodes
    - Edges
- Node graph navigation
  - Pan
  - Zoom
  - Switch layouts
  - Hidden nodes
- Configuration options
  - Panel options
  - Node graph options
    - Zoom mode
    - Layout algorithm
  - Nodes options
  - Edges options
  - Data links
  - Field overrides
- Data API
  - Edges data frame structure
  - Nodes data frame structure

Node graphs are useful when you need to visualize elements that are related to each other. This is done by displaying circles—or *nodes*—for each element you want to visualize, connected by

lines—or *edges*. By default, the visualization uses a layered layout that positions the nodes into a network of connected circles.

Node graphs display useful information about each node, as well as the relationships between them, allowing you to visualize complex infrastructure maps, hierarchies, or execution diagrams.

Node graph visualization

The appearance of nodes and edges can also be customized in several ways including color, borders, and line style.

You can use a node graph visualization if you need to show:

- Solution topologies
- Networks
- Infrastructure
- Organizational charts
- Critical path diagrams
- Family trees
- Mind maps

Configure a node graph visualization

The following video provides beginner steps for creating node panel visualizations. You'll learn the data requirements and caveats, special customizations, and much more:

{{< youtube id="VrvsMkRwoKw" >}}

{{< docs/play title="Node graph panel" url="https://play.grafana.org/d/bdodfbi3d57uoe/" >}}

Supported data formats

To create node graphs, you need two datasets: one containing the records for the displayed elements (nodes) and one dataset containing the records for the connections between those elements (edges).

Nodes dataset

The nodes dataset must contain one alphanumeric ID field that gives each element a unique identifier. The visualization also accepts other options fields for titles, subtitles, main and secondary stats, arc information for how much of the circle border to paint, details, colors, icons, node size, and indicators for element highlighting. For more information and naming conventions for these fields, refer to the Nodes data frame structure section.

Example

id	title	subtitle	mainstat	secondary stat	color	icon	highlighted
node1	PC	Windows	AMD	16gbRAM	blue		true
node2	PC	Linux	Intel	32gbRAM	green	eye	false
node3	Mac	MacOS	M3	16gbRAM	gray	apps	false
node4	Alone	SoLonely	JustHere	NotConnected	red		false

If the icon field contains a value, it's displayed instead of the title and subtitle. For a list of of available icons, refer to Icons Overview.

Edges dataset

Similar to the nodes dataset, the edges dataset needs one unique ID field for each relationship, followed by two fields containing the source and the target nodes of the edge; that is, the nodes the edge connects. Other optional fields are main and secondary stats, context menu elements, line thickness, highlight indications, line colors, and configurations to turn the connection into a dashed line. For more information and naming conventions for these fields, refer to the Edges data frame structure section.

Example

id	source	target	mainstat	seconddarystat	thickness	highlighted	color
edge1	node1	node2	TheMain	TheSub	3	true	cyan
edge2	node3	node2	Main2	Sub2	1	false	orange

If a node lacks edge connections, it's displayed on its own outside of the network.

Data requirements

A node graph requires a specific shape of the data to be able to display its nodes and edges. This means not every data source or query can be visualized with this graph. If you want to use this as a data source developer see the section about data API.

A node graph consists of *nodes* and *edges*.

- A *node* is displayed as a circle. A node might represent an application, a service, or anything else that is relevant from an application perspective.
- An *edge* is displayed as a line that connects two nodes. The connection might be a request, an execution, or some other relationship between the two nodes.

Both nodes and edges can have associated metadata or statistics. The data source defines what information and values is shown, so different data sources can show different type of values or not show some values.

Nodes

{{< admonition type="note" >}} Node graphs show up to 200 visible nodes by default. If the graph has more nodes, Grafana shows a warning at the top of the panel and hides some nodes for performance reasons. You can expand hidden parts of the graph by clicking the "Hidden nodes" markers in the graph. {{< /admonition >}}

Usually, nodes show two statistical values inside the node and two identifiers just below the node, usually name and type. Nodes can also show another set of values as a color circle around the node, with sections of different color represents different values that should add up to 1.

For example, you can have the percentage of errors represented by a red portion of the circle. Additional details can be displayed in a context menu which is displayed when you click on the node. There also can be additional links in the context menu that can target either other parts of Grafana or any external link.

Node context menu

Edges

Edges can also show statistics when you hover over the edge. Similar to nodes, you can open a context menu with additional details and links by clicking on the edge.

The first data source supporting this visualization is X-Ray data source for its Service map feature.

For more information, refer to the X-Ray plugin documentation.

Node graph navigation

You can use pan, zoom, and other functions to navigate a node graph.

Pan

You can pan the view by clicking outside any node or edge and dragging your mouse.

Zoom

Use the buttons in the lower right corner to zoom in or out. You can also use the mouse wheel or touchpad scroll, together with either Ctrl or Cmd key to do so.

Switch layouts

Switch quickly between displaying the visualization in graph or grid layout.

Click a node and select either **Show in Grid layout** or **Show in Graph layout**, depending on the current layout of the visualization:

{{< figure src="/media/docs/grafana/panels-visualizations/screenshot-node-graph-grid-menu.png" max-width="750px" alt="Node graph in grid layout with node menu open" >}}

In grid layout, you can sort nodes by clicking on the stats inside the legend. The marker next to the stat name shows which stat is currently used for sorting and the sorting direction:

{{< figure src="/media/docs/grafana/panels-visualizations/screenshot-node-graph-legend-sort.png" max-width="550px" alt="Node graph legend sorting" >}}

Switching between grid and other layouts this way only changes the layout temporarily. The visualization maintains the layout algorithm selected in the panel editor, and reverts to it when the dashboard refreshes.

For more information about layouts, refer to Layout algorithm.

Hidden nodes

The number of nodes shown at a given time is limited to maintain a reasonable visualization performance. Nodes that are not currently visible are hidden behind clickable markers that show an approximate number of hidden nodes that are connected by a particular edge. You can click on the marker to expand the graph around that node.

Node graph exploration

Configuration options

{{< docs/shared lookup="visualizations/config-options-intro.md" source="grafana" version="<GRAFANA_VERSION>" >}}

Panel options

{{< docs/shared lookup="visualizations/panel-options.md" source="grafana" version="<GRAFANA_VERSION>" >}}

Node graph options

Use the following options to refine your node graph visualization.

Zoom mode

Choose how the node graph should handle zoom and scroll events:

- **Cooperative** - Allows you to scroll the visualization normally.
- **Greedy** - Reacts to all zoom gestures.

Layout algorithm

Choose how the visualization layout is generated:

- **Layered** - Default. Creates a predictable and orderly layout, especially useful for service graphs.
- **Force** - Uses a physics-based force layout algorithm that's useful with a large number of nodes (500+).
- **Grid** - Arranges nodes into a grid format to provide a better overview of the most interesting nodes in the graph. This layout shows nodes in a grid without edges and can be sorted by the stats shown inside the node or by the ones represented by the a colored border of the nodes.
  {{< figure src="/media/docs/grafana/panels-visualizations/screenshot-node-graph-grid.png" max-width="650px" alt="Node graph in grid layout" >}}

  For more information about using the graph in grid layout, refer to Switch layouts.

{{< admonition type="note" >}} Layered layout can be slow when the graph has more than 500 nodes. For large graphs, use **Force** layout. {{< /admonition >}}

Nodes options

The **Nodes** options section provides configurations for node behaviors.

- **Main stat unit** - Choose which unit the main stat displays in the graph's nodes.
- **Secondary stat unit** - Choose which unit the secondary stat displays in the graph's nodes.
- **Arc sections** - Configure which fields define the size of the colored circle around the node and select a color for each. You can add multiple fields.

{{< admonition type="note" >}} Defining arc sections overrides the automatic detection of `arc__*` and `color` fields described in the **Optional fields** section of Nodes data frame structure. {{< /admonition >}}

Edges options

The **Edges** options section provides configurations for node edges behaviors.

- **Main stat unit** - Choose which unit the main stat displays in the graph's edges.
- **Secondary stat unit** - Choose which unit the secondary stat displays in the graph's edges.

Data links

{{< docs/shared lookup="visualizations/datalink-options.md" source="grafana" version="<GRAFANA_VERSION>" >}}

In node graphs, some data fields may have pre-configured data links. To add a different data link in those cases, use a field override.

Field overrides

{{< docs/shared lookup="visualizations/overrides-options.md" source="grafana" version="<GRAFANA_VERSION>" >}}

Data API

This visualization needs a specific shape of the data to be returned from the data source to correctly display it.

Node graphs, at minimum, require a data frame describing the edges of the graph. By default, node graphs compute the nodes and any stats based on this data frame. Optionally a second data frame describing the nodes can be sent when you need to show more node-specific metadata. Grafana recognizes node graph frames when `frame.meta.preferredVisualisationType = 'nodeGraph'`, when the frame name or `refId` is `nodes` or `edges`, or when a frame contains an `id` field.

Edges data frame structure

Required fields:

Field name	Type	Description
id	string	Unique identifier of the edge.
source	string	Id of the source node.
target	string	Id of the target.

Optional fields:

Field name	Type	Description
mainstat	string/number	First stat shown in the overlay when hovering over the edge. It can be a string showing the value as is or it can be a number. If it is a number, any unit associated with that field is also shown
secondarystat	string/number	Same as mainStat, but shown right under it.
detail__*	string/number	Any field prefixed with `detail__` will be shown in the header of context menu when clicked on the edge. Use `config.displayName` for more human readable label.
thickness	number	The thickness of the edge. Default: `1`
highlighted	boolean	Sets whether the edge should be highlighted. Useful, for example, to represent a specific path in the graph by highlighting several nodes and edges. Default: `false`
color	string	Sets the default color of the edge. It can be an acceptable HTML color string. Default: `#999`

Field name	Type	Description
strokeDasharray	string	Sets the pattern of dashes and gaps used to render the edge. If unset, a solid line is used as edge. For more information and examples, refer to the `stroke-dasharray` MDN documentation.

{{< admonition type="caution" >}} Starting with 10.5, `highlighted` is deprecated. It will be removed in a future release. Use `color` to indicate a highlighted edge state instead. {{< /admonition >}}

Nodes data frame structure

Required fields:

Field name	Type	Description
id	string	Unique identifier of the node. This ID is referenced by edge in its source and target field.

Optional fields:

Field name	Type	Description
title	string	Name of the node visible in just under the node.
subtitle	string	Additional, name, type or other identifier shown under the title.
mainstat	string/number	First stat shown inside the node itself. It can either be a string showing the value as is or a number. If it is a number, any unit associated with that field is also shown.
secondarystat	string/number	Same as mainStat, but shown under it inside the node.
arc__*	number	Any field prefixed with `arc__` will be used to create the color circle around the node. All values in these fields should add up to 1. You can specify color using `config.color.fixedColor`.
detail__*	string/number	Any field prefixed with `detail__` will be shown in the header of context menu when clicked on the node. Use `config.displayName` for more human readable label.

Field name	Type	Description
color	string/number	Can be used to specify a single color instead of using the `arc__` fields to specify color sections. It can be either a string which should then be an acceptable HTML color string or it can be a number in which case the behavior depends on `field.config.color.mode` setting. This can be for example used to create gradient colors controlled by the field value.
icon	string	Name of the icon to show inside the node instead of the default stats. Only Grafana built in icons are allowed.
nodeRadius	number	Radius value in pixels. Used to manage node size.
highlighted	boolean	Sets whether the node should be highlighted. Useful for example to represent a specific path in the graph by highlighting several nodes and edges. Default: `false`
fixedX	number	Fixed x-coordinate for the node. If you use `fixedX`, all nodes must provide a value.
fixedY	number	Fixed y-coordinate for the node. If you use `fixedY`, all nodes must provide a value.

□□□ #2
□ □□□ □□□ 2026-10-31 16:38:24 UTC
□ □□□ □□□ 2026-10-31 16:38:30 UTC