+++ title = "Node graph" keywords = ["grafana", "dashboard", "documentation", "panels", "node graph", "directed graph"] aliases = ["/docs/grafana/latest/panels/visualizations/node-graph/"] weight = 850 +++

Node graph panel

> **Note:** This panel is currently in beta. Expect changes in future releases.

The *Node graph* can visualize directed graphs or networks. It uses a directed force layout to effectively position the nodes, so it can display complex infrastructure maps, hierarchies, or execution diagrams.

Node graph panel

Data requirements

The Node graph panel requires 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 in this panel. If you want to use this as a data source developer see the section about data API.

The Node graph visualization 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

> **Note:** Node graph can show only 1,500 nodes. If this limit is crossed a warning will be visible in upper right corner, and some nodes will be hidden. You can expand hidden parts of the graph by clicking on the "Hidden nodes" markers in the graph.

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 graph navigation

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 it's Service map feature. For more information, refer to the X-Ray plugin documentation.

Navigating the node graph

You can pan and zoom in or out the node graph.

Pan

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

Zoom in or out

Use the buttons in the upper left corner or use the mouse wheel, touchpad scroll, together with either Ctrl or Cmd key to zoom in or out.

Explore 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

Grid view

You can switch to the grid view to have a better overview of the most interesting nodes in the graph. Grid view shows nodes in a grid without edges and can be sorted by stats shown inside the node or by stats represented by the a colored border of the nodes.

Node graph grid

To sort the nodes, click on the stats inside the legend. The marker next to the stat name shows which stat is currently used for sorting and sorting direction.

Node graph legend

Click on the node and select "Show in Graph layout" option to switch back to graph layout and focus on the selected node, to show it in context of the full graph.

Node graph grid to default

Data API

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

Data source needs to return two data frames, one for nodes and one for edges. You have to set `frame.meta.preferredVisualisationType = 'nodeGraph'` on both data frames or name them **nodes** and **edges** respectively for the node graph to render.

Node parameters

Required fields:

Field name	Type	Description
id	string	Unique identifier of the node. This ID is referenced by edge in it's 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 that will be shown right under the title.
mainStat	string/number	First stat shown inside the node itself. Can be either string in which case the value will be shown as it is or it can be a number in which case any unit associated with that field will be also shown.
secondaryStat	string/number	Same as mainStat but shown right 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.

3

Edge parameters

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. Can be either string in which case the value will be shown as it is or it can be a number in which case any unit associated with that field will be 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.