MLOps Monitoring Practices with Prometheus & Grafana

Palani Karthikeyan

Sr.Technical Consultant, Krosum Labs

karthikeyan@krosum.com

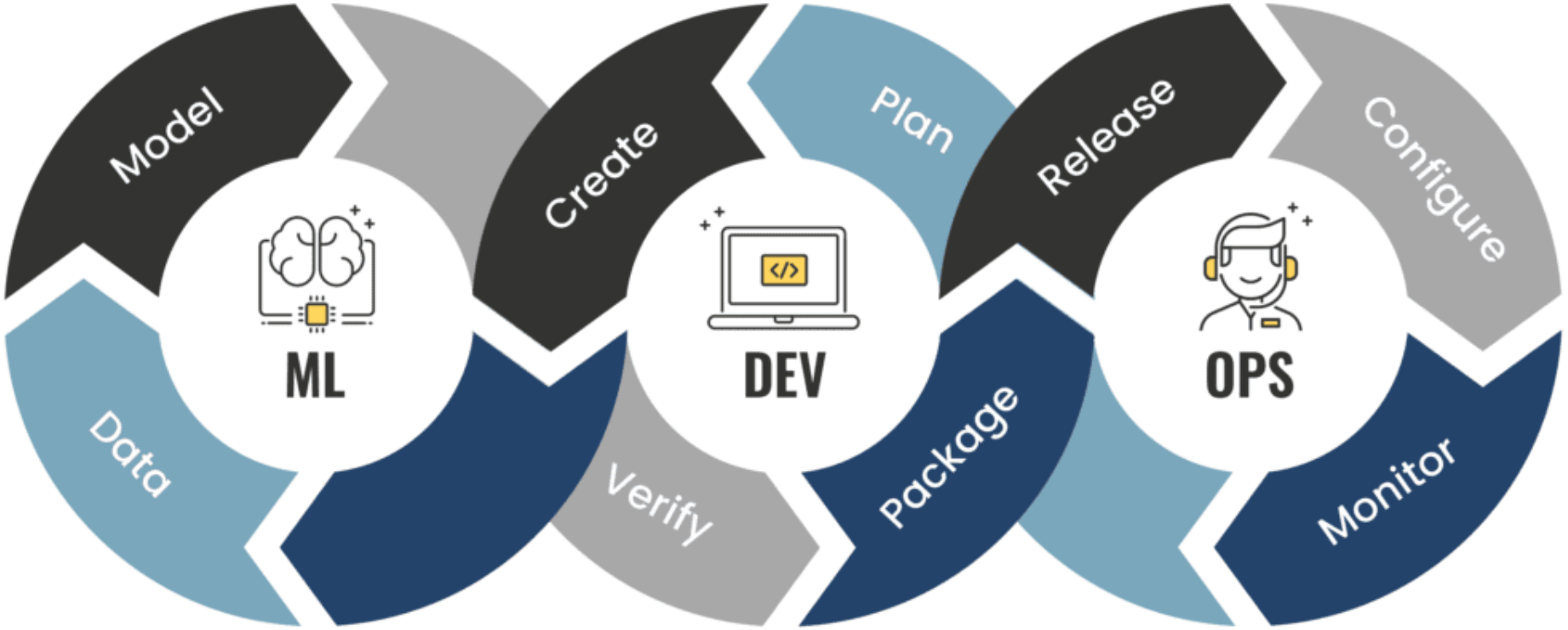
Agenda

* MLOps and Devops
* Grafana – Deep dive into Grafana
* Prometheus
* Mlfow – Case studies

MLOps

* Machine Learning Operations for Production, is a collection of defined methods for building, deploying, and governing the lifespan of machine learning models.
* MLOps approaches also improve the ML systems' scalability, security, and reliability, resulting in faster development cycles and more revenues from ML initiatives.

MLOps


Model
Create
Plan
Release
Configure
Data
Verify
Package
Monitor
ML
DEV
OPS

MLOps

* MLOps is a methodology of operation that aims to facilitate the process of bringing an experimental Machine Learning model into production and maintaining it efficiently.
* MLOps focus on bringing the methodology of DevOps used in the software industry to the Machine Learning model lifecycle.

Features of a MLOPs project

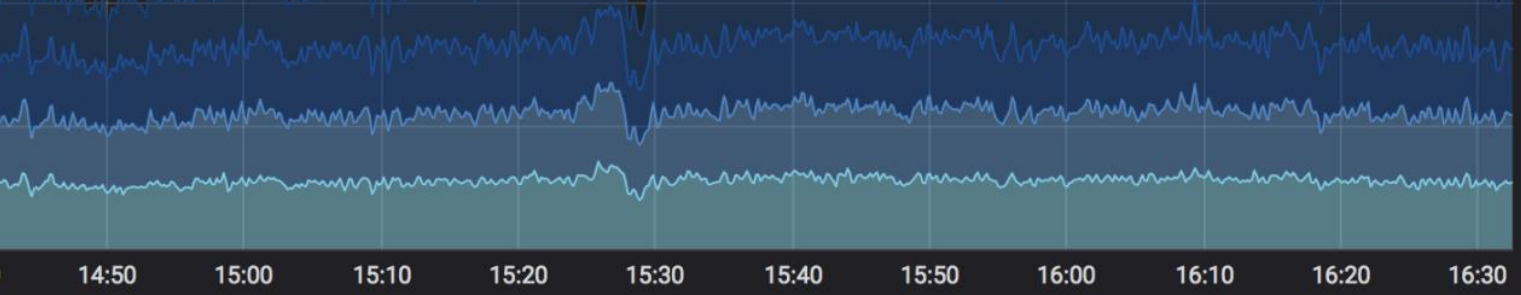
* Data and Model Versioning
* Feature Management and Storing
* Automation of Pipelines and Processes
* CI/CD for Machine Learning
* Continuous Monitoring of Models

Grafana

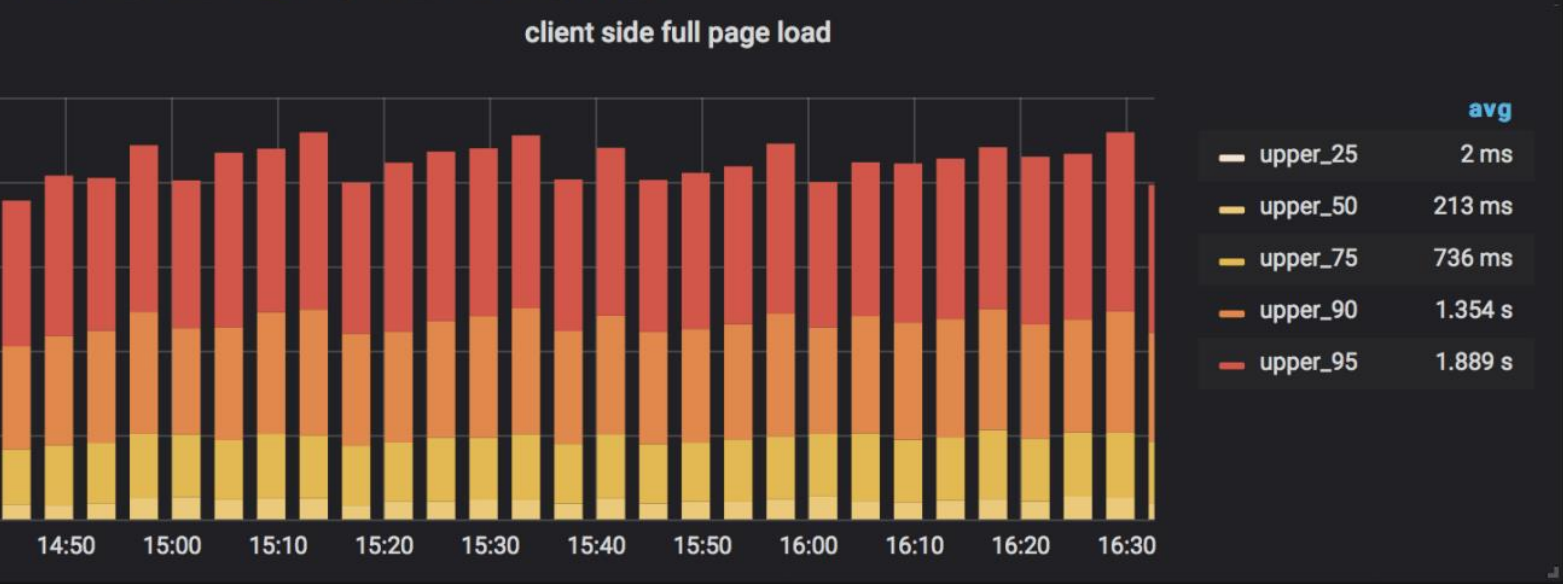
* Grafana is an **open-source data visualization** and **analysis tool** which allows us to view our data in the form of **beautiful graphs.**
* Grafana designed by **Torkel Odegaard** in **January 2014.**

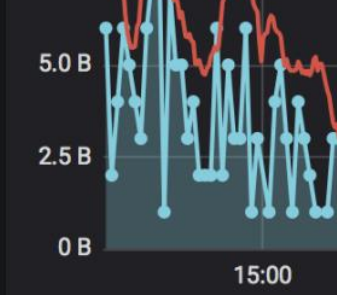
Grafana

* It enables us to create a dashboard for **collecting**, **processing**, **storing**, and **analyzing data** from various different sources.
* It allows us to query, visualize, alert on, and understand our metrics.



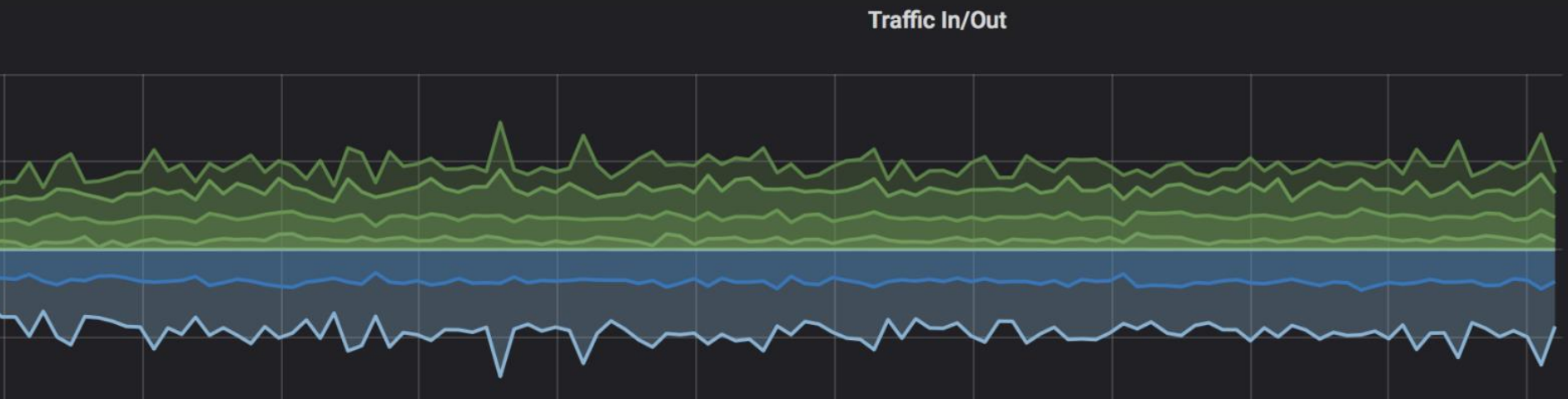
01 web_server_02 web_server_03 web_server_04





memory

logins logins



Grafana

* It includes a variety of visualization option that helps us to understand our data, beautifully.
* Grafana is written in **Go** and **Node.js LTS** language with a strong API (Application Programming Interface).

Grafana Dashboard

* Grafana Dashboard is used to pull data from various data sources such as **Prometheus, Influx DB, Graphite, ElasticSearch, MySQL, PostgreSQL, CloudWatch, Microsoft SQL Server**, and many more.
* A Grafana Dashboard contains various visualization options such as **heat maps, geomaps, histograms, tables, free text panels**, and different types of **charts & graphs** to study and understand business data easily.

Why Grafana?

* Easy virtualization
* Drag and Drop panels
* Flexible to use
* Supports different data source
* Open source
* Multi-platform support

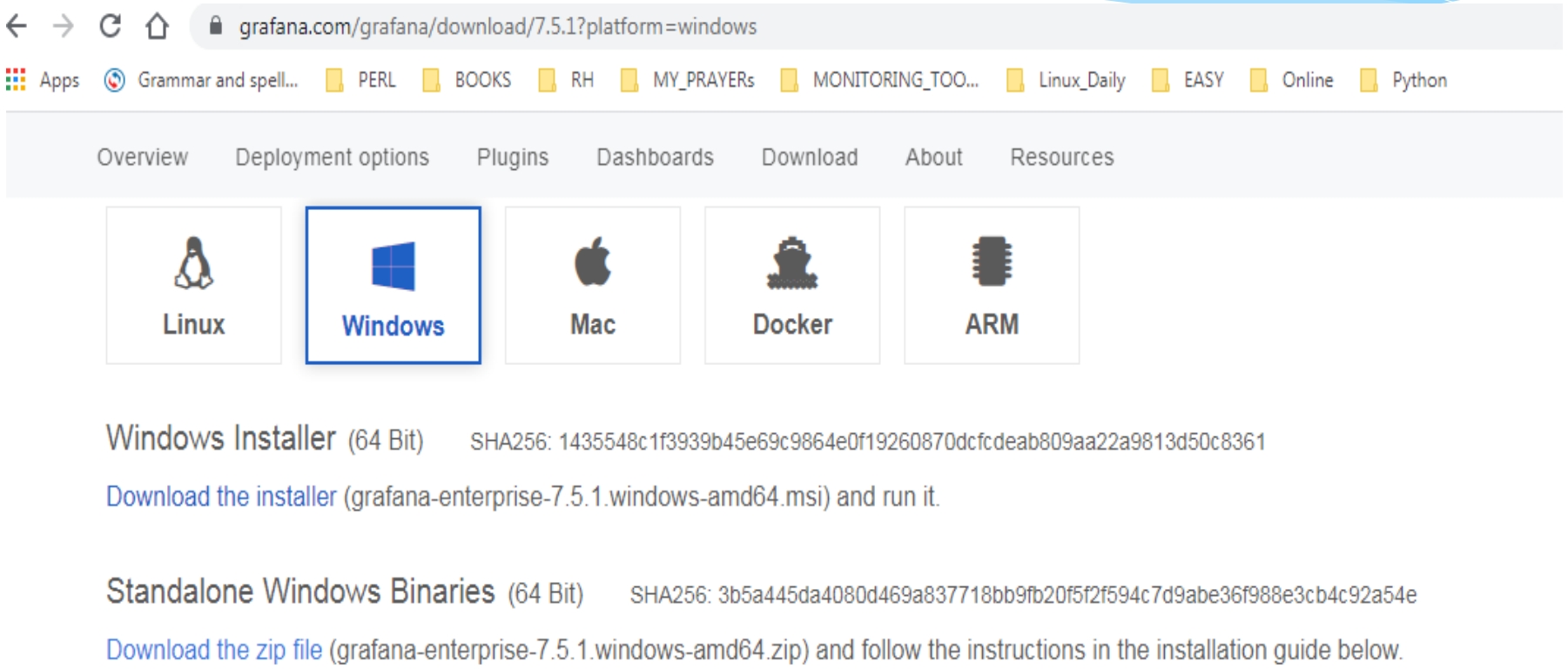
Features of Grafana

* Plugins Platform
* Transformation
* Dynamic Dashboards
* Authentication
* Explore metrics and Logs
* Alerting
* Annotations
* Mixed data sources

Installing Grafana

* To download the latest versions of Grafana, check out **https://grafana.com/grafana/download**

Install Grafana on Windows



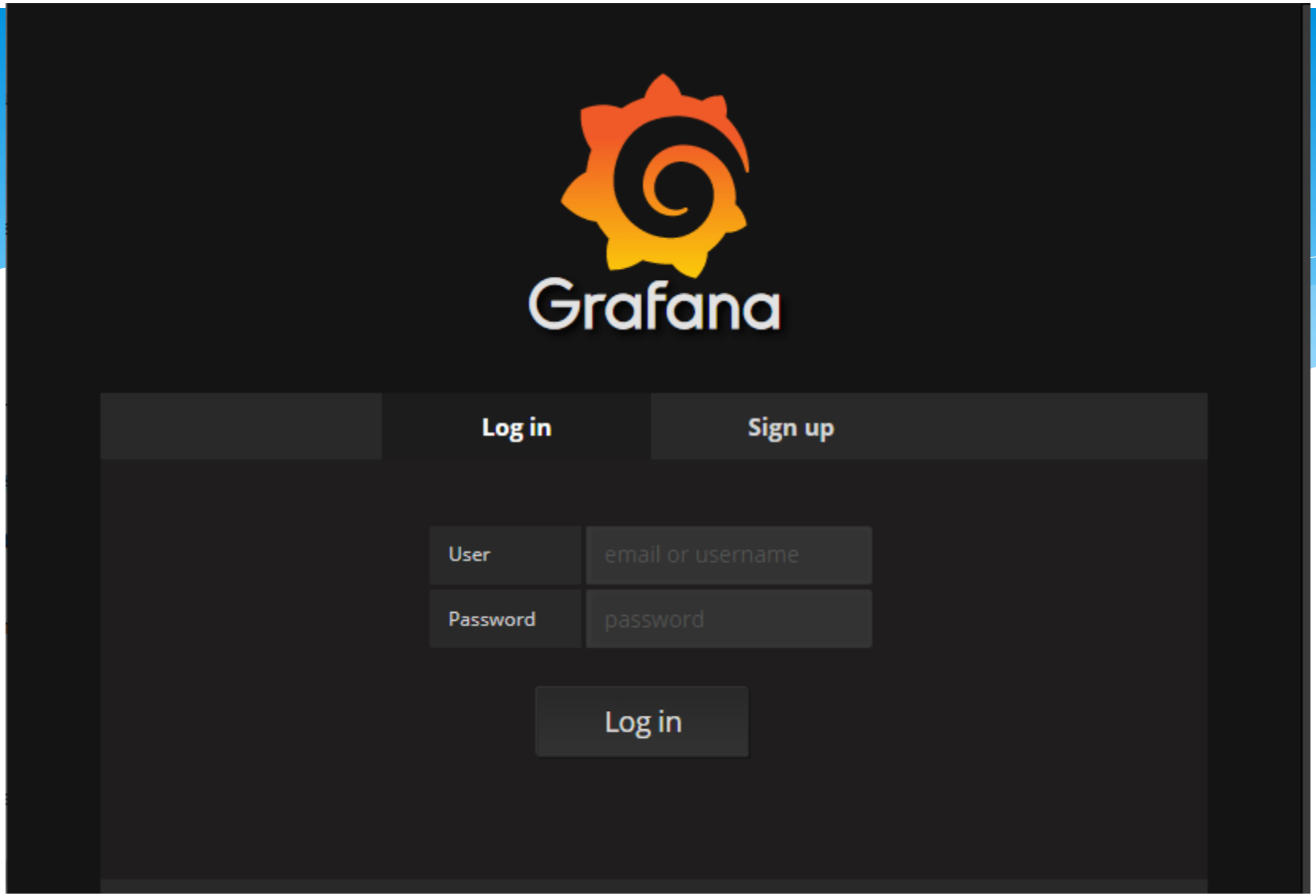
* **Step 1:** Go to any web browser, copy-paste the below link, or simply click on the below link.

* https://grafana.readthedocs.io/en/latest/sources/installation/windows.html

* **Step 2:** Once the link is opened, the below window appears on the screen, in which click on the **grafana.4.4.1.windows-x64.zip**.

Connecting to the Grafana server

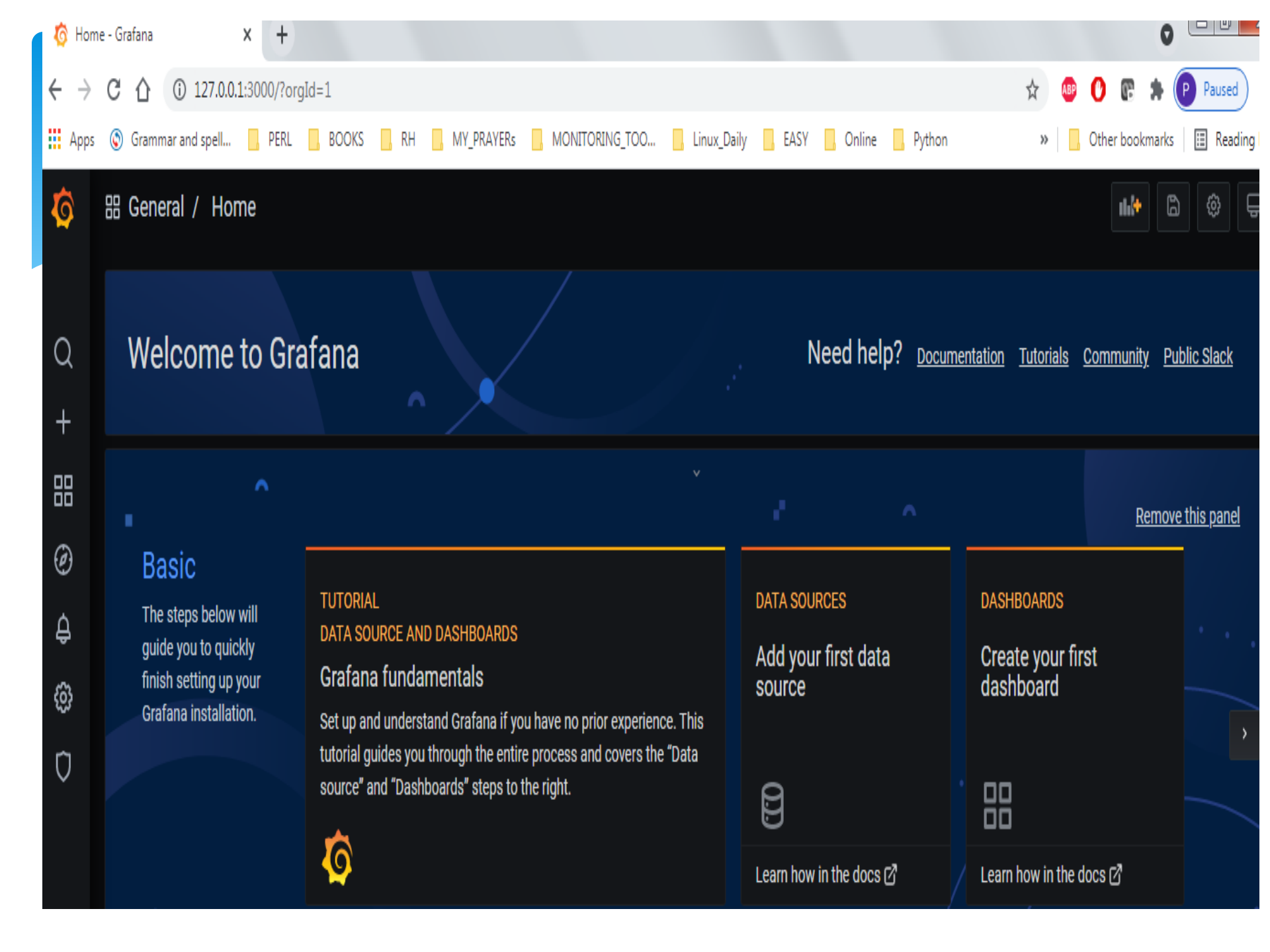
* Once you have installed and launched Grafana, open a browser page to access
* the Grafana application. It can be found at **http://localhost:3000**



Log in with the **admin** username and the **admin** password

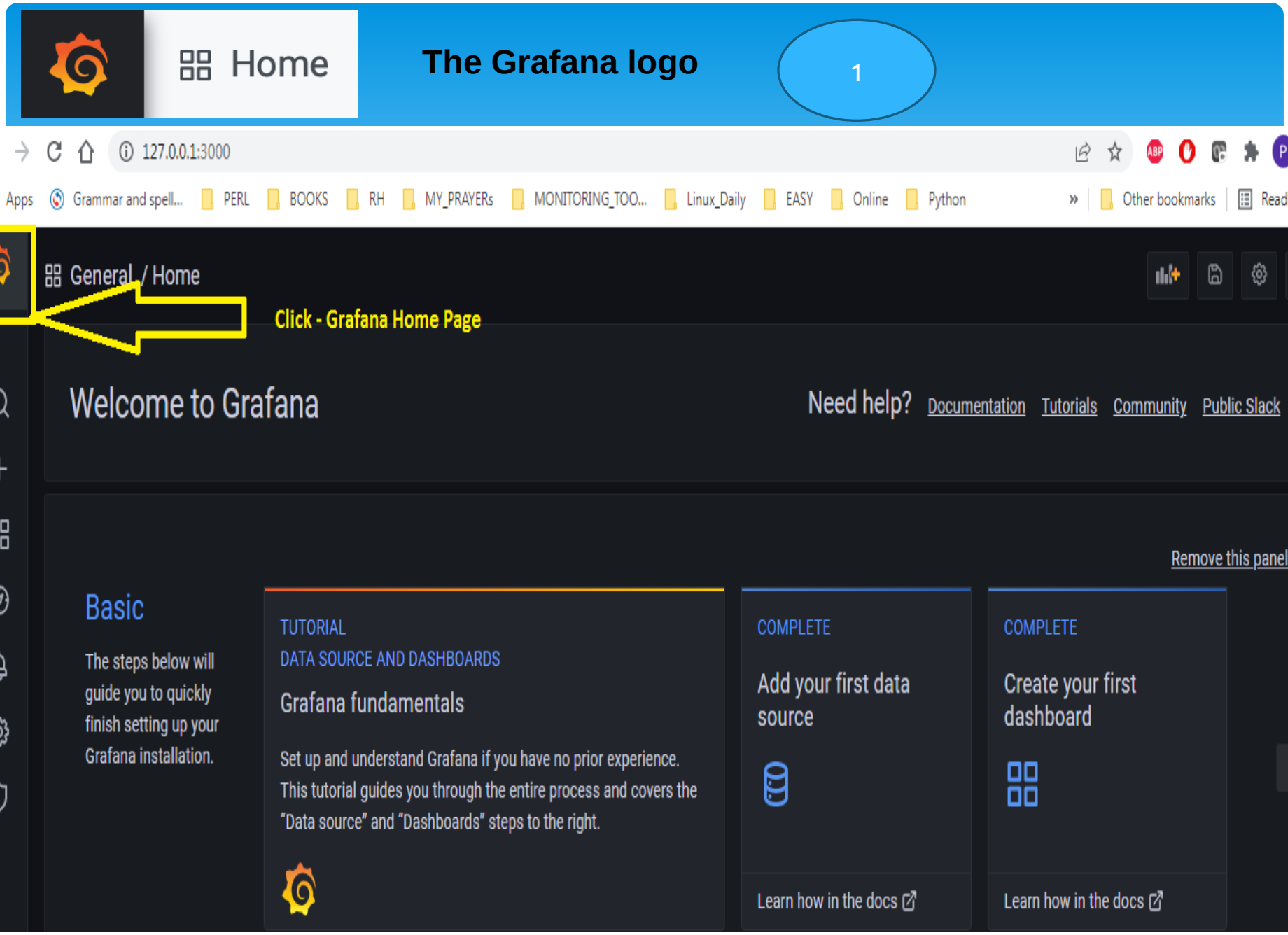
Grafana User Interface

* The default Home dashboard
* The sidebar menu.
* Navigation hub, providing both quick access to simple creation pages and links to more complex functions, including data source creation, Explore mode, alert management, and server administration.

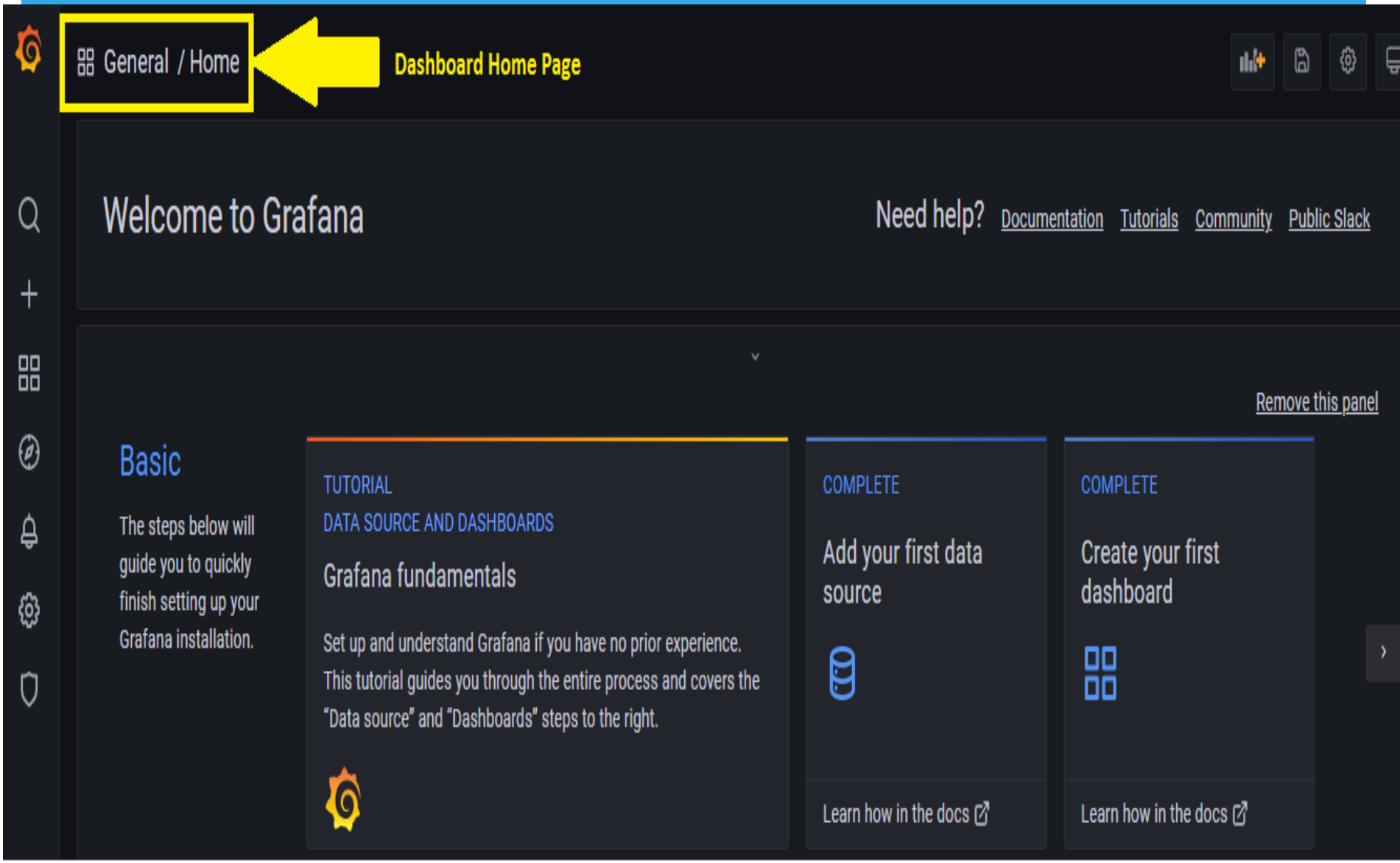

Home - Grafana
127.0.0.1:3000/?orgId=1
Paused
Apps
Grammar and spell...
PERL
BOOKS
RH
MY_PRAYERs
MONITORING_TOO...
Linux_Daily
EASY
Online
Python
Other bookmarks
Reading
General / Home
Welcome to Grafana
Need help?
Documentation
Tutorials
Community
Public Slack
Remove this panel
Basic
The steps below will guide you to quickly finish setting up your Grafana installation.
TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals
Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.
DATA SOURCES
Add your first data source
Learn how in the docs
DASHBOARDS
Create your first dashboard
Learn how in the docs

Grafana's left sidebar

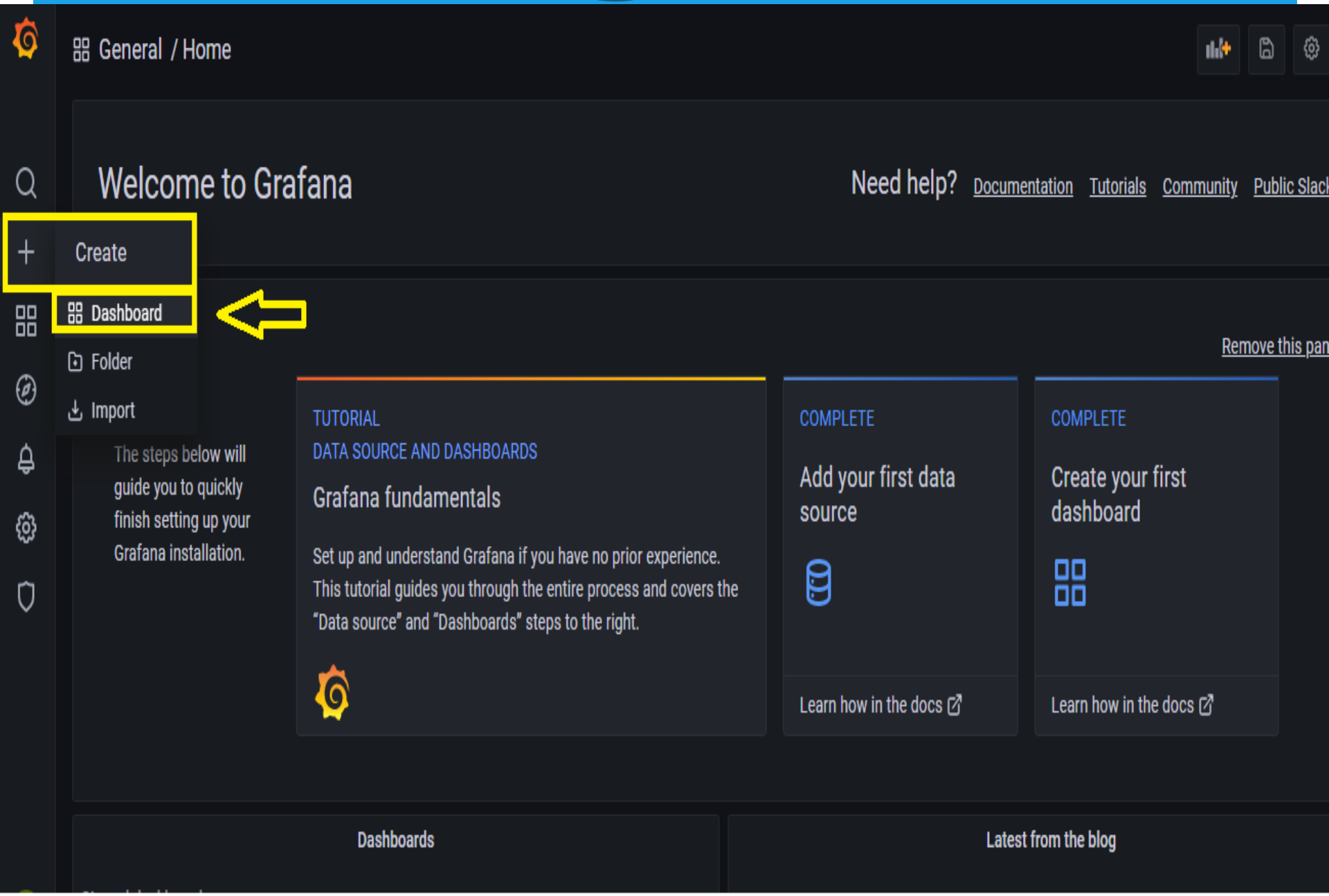
* To the left of the dashboard itself is the left sidebar.
* These icons lead to some of the most powerful of Grafana's impressive features.
* Search for dashboards
* Create and import dashboards and folders
* Find dashboards
* Manage dashboards, dashboard playlists, and dashboard snapshots
* Explore data sources in a free-form fashion
* Manage alert rules and notification channels
* Configure data sources, users, and teams, download plugins, set
* preferences, and generate API keys
* Administer Grafana users and organizations and view the server
* settings and stats
* Return to the Home dashboard
* Set personal preferences
* Get help


Home
The Grafana logo
1
127.0.0.1:3000
Apps
Grammar and spell...
PERL
BOOKS
RH
MY_PRAYERs
MONITORING_TOO...
Linux_Daily
EASY
Online
Python
Other bookmarks
Reading
General / Home
Click - Grafana Home Page
Welcome to Grafana
Need help?
Documentation
Tutorials
Community
Public Slack
Remove this panel
Basic
The steps below will guide you to quickly finish setting up your Grafana installation.
TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals
Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.
COMPLETE
Add your first data source
Learn how in the docs
COMPLETE
Create your first dashboard
Learn how in the docs

2


General / Home
Dashboard Home Page
Welcome to Grafana
Need help?
Documentation
Tutorials
Community
Public Slack
Remove this panel
Basic
The steps below will guide you to quickly finish setting up your Grafana installation.
TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals
Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.
COMPLETE
Add your first data source
Learn how in the docs
COMPLETE
Create your first dashboard
Learn how in the docs

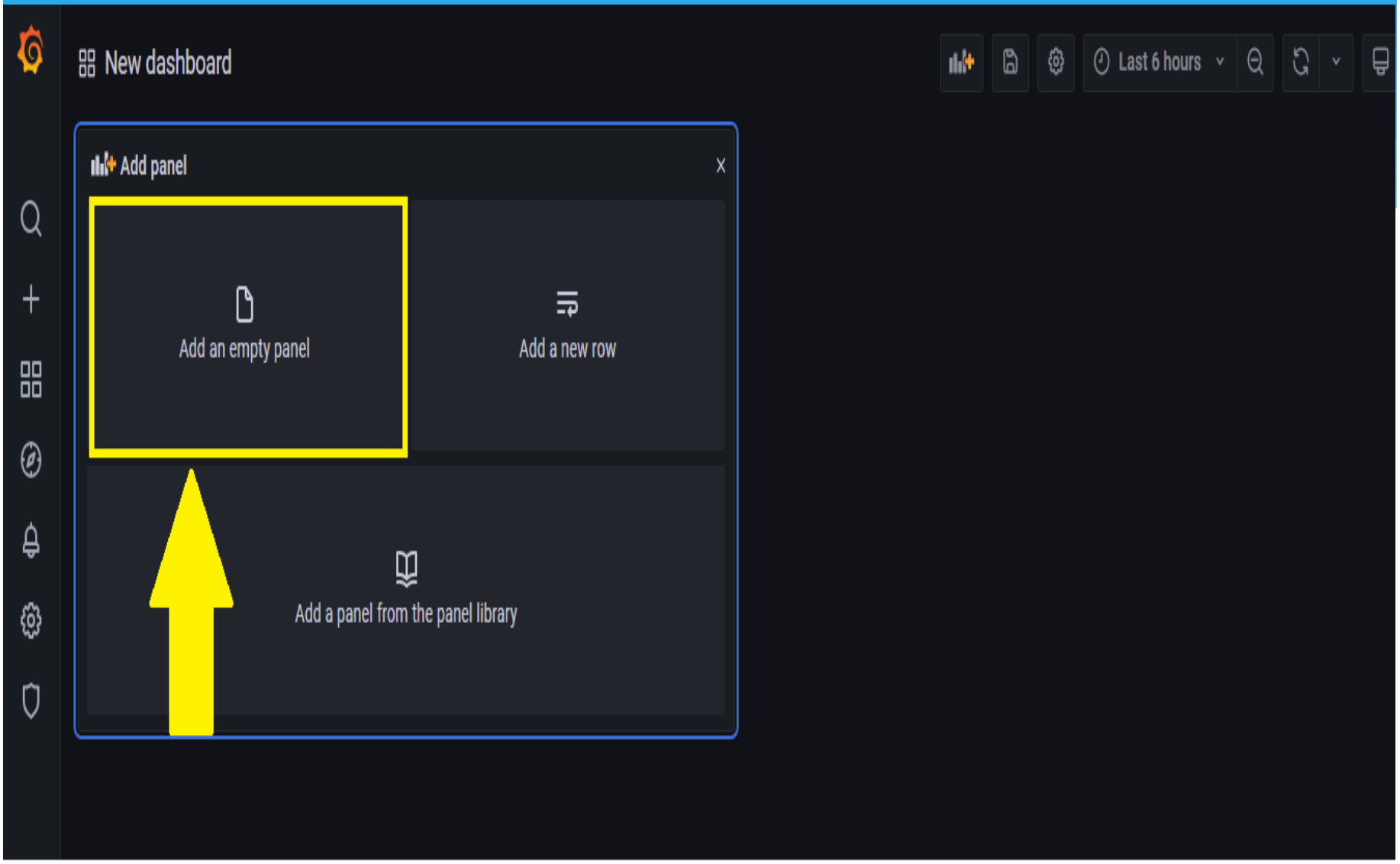
3


General / Home
Welcome to Grafana
Need help?
Documentation
Tutorials
Community
Public Slack
Create
Dashboard
Folder
Import
Remove this pan
The steps below will guide you to quickly finish setting up your Grafana installation.
TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals
Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.
COMPLETE
Add your first data source
Learn how in the docs
COMPLETE
Create your first dashboard
Learn how in the docs
Dashboards
Latest from the blog

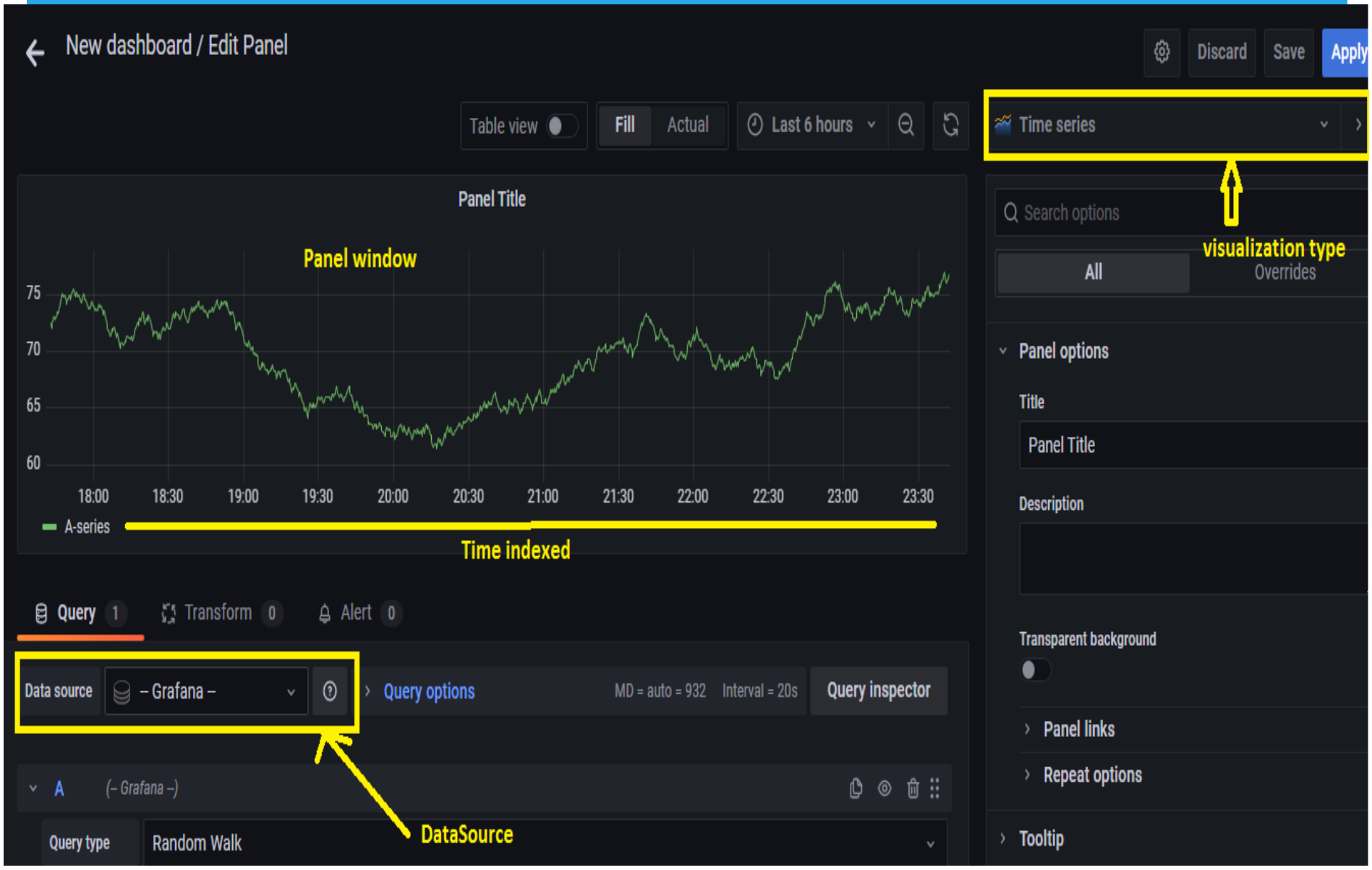
The panel's UI

* The panel's UI can be broken down into roughly three main functional areas:
* 1. **Panel display:** Preview display, and time picker
* 2. **Display settings:** Panel visualization type, styles, and links
* 3. **Data configuration:** Data query, data transformation, and alerting

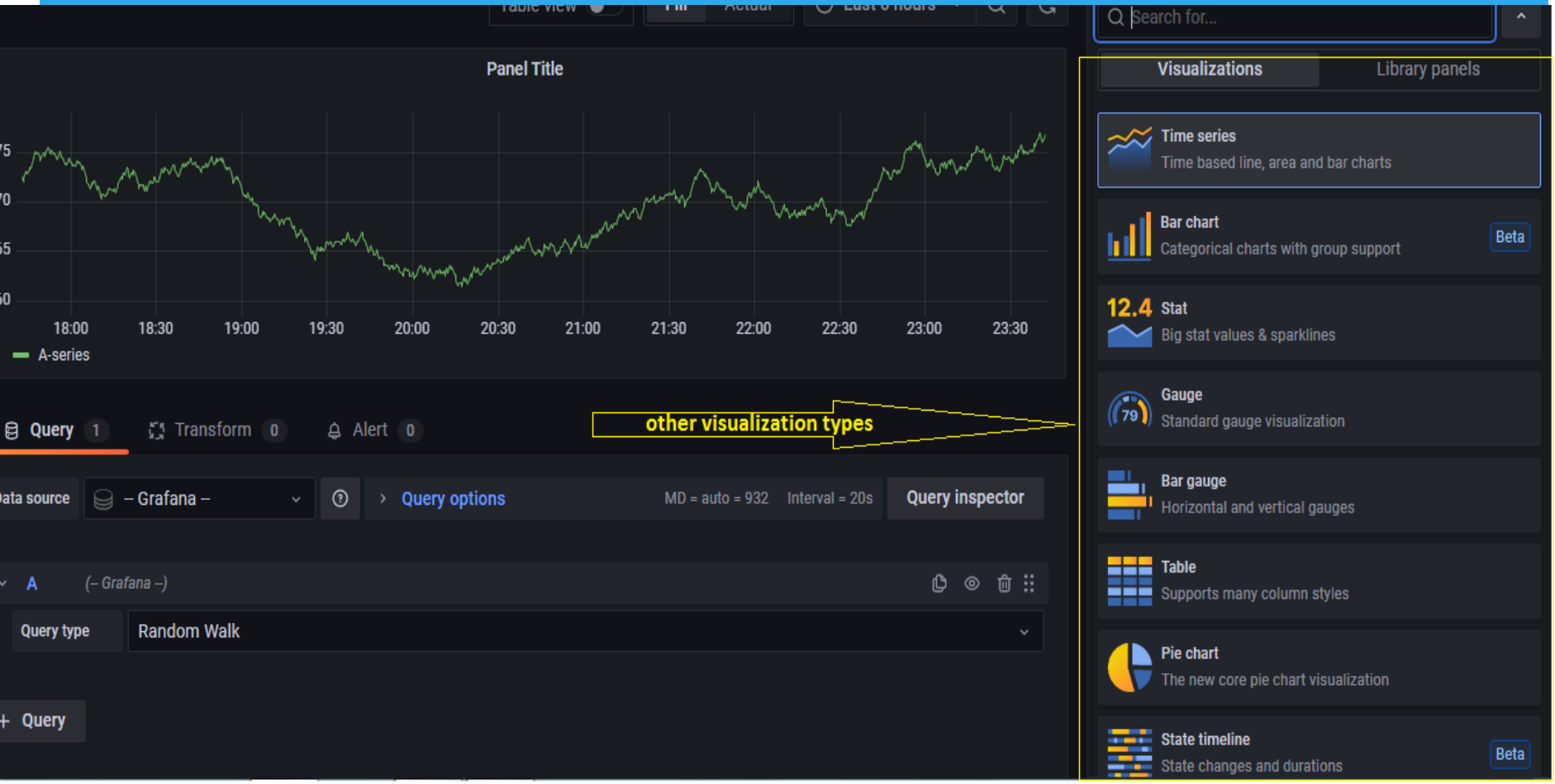
4


New dashboard
Last 6 hours
Add panel
X
Add an empty panel
Add a new row
Add a panel from the panel library

5


New dashboard / Edit Panel
Discard
Save
Apply
Table view
Fill
Actual
Last 6 hours
Time series
Panel Title
Panel window
visualization type
Search options
All
Overrides
75
70
65
60
18:00
18:30
19:00
19:30
20:00
20:30
21:00
21:30
22:00
22:30
23:00
23:30
A-series
Time indexed
Panel options
Title
Panel Title
Description
Query 1
Transform 0
Alert 0
Transparent background
Data source
– Grafana –
Query options
MD = auto = 932
Interval = 20s
Query inspector
Panel links
Repeat options
A
(– Grafana –)
DataSource
Query type
Random Walk
Tooltip

6


Search for...
Visualizations
Library panels
Panel Title
18:00
18:30
19:00
19:30
20:00
20:30
21:00
21:30
22:00
22:30
23:00
23:30
A-series
Time series
Time based line, area and bar charts
Bar chart
Categorical charts with group support
Beta
12.4
Stat
Big stat values & sparklines
Gauge
Standard gauge visualization
Bar gauge
Horizontal and vertical gauges
Table
Supports many column styles
Pie chart
The new core pie chart visualization
State timeline
State changes and durations
Beta
Query 1
Transform 0
Alert 0
other visualization types
Data source
– Grafana –
Query options
MD = auto = 932
Interval = 20s
Query inspector
A
(– Grafana –)
Query type
Random Walk
+ Query

Dashboard

* A dashboard is a kind of canvas upon which you can display **one or more panels** in a grid style arrangement.
* It also serves as a web page, so you can bookmark or share it with a simple URL.
* The entire dashboard can even be imported and exported in JSON text file format, making it easy to share, save, or transfer to another version of Grafana.

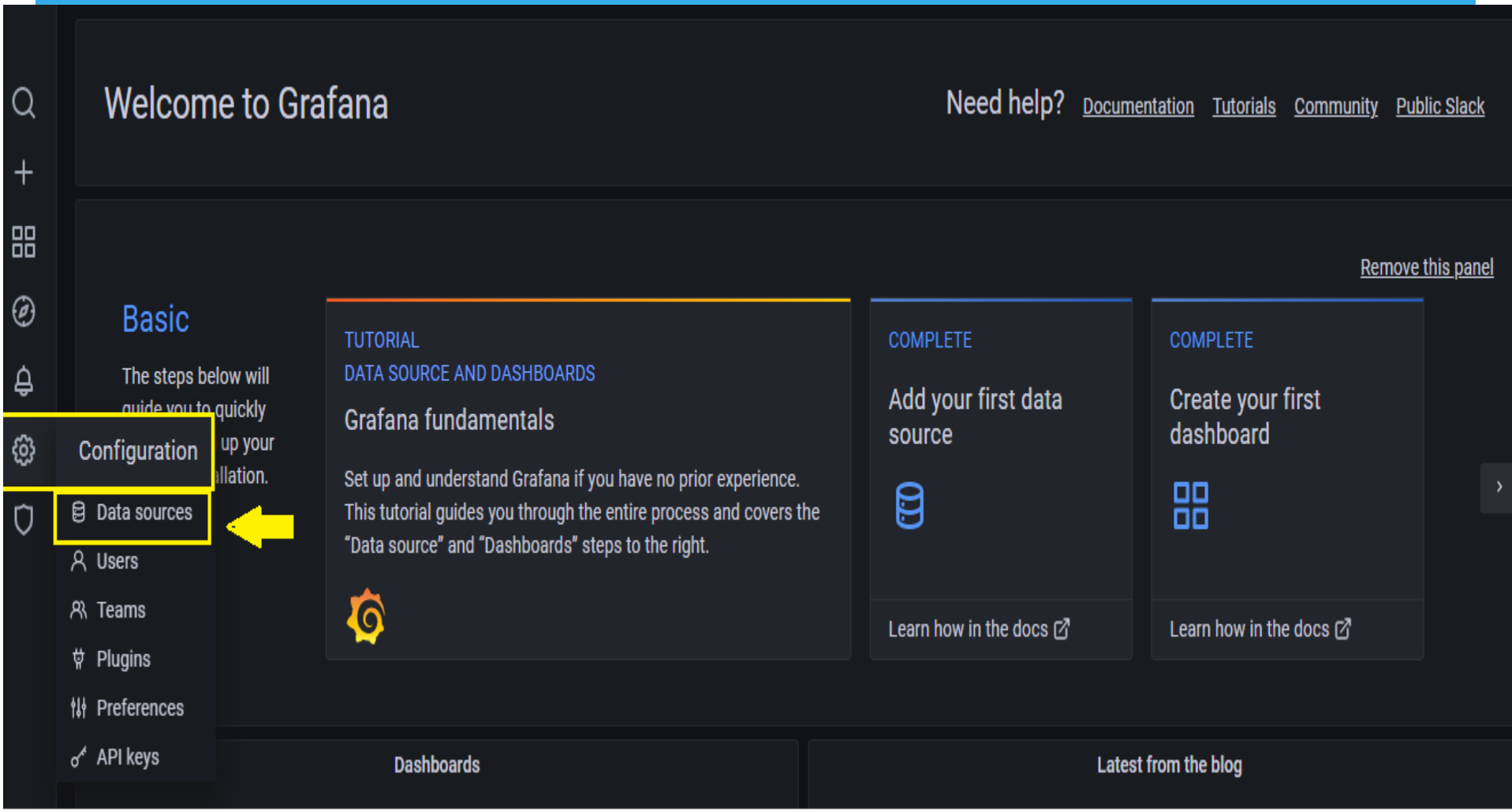
Unit (or) service file & Grafana server daemon

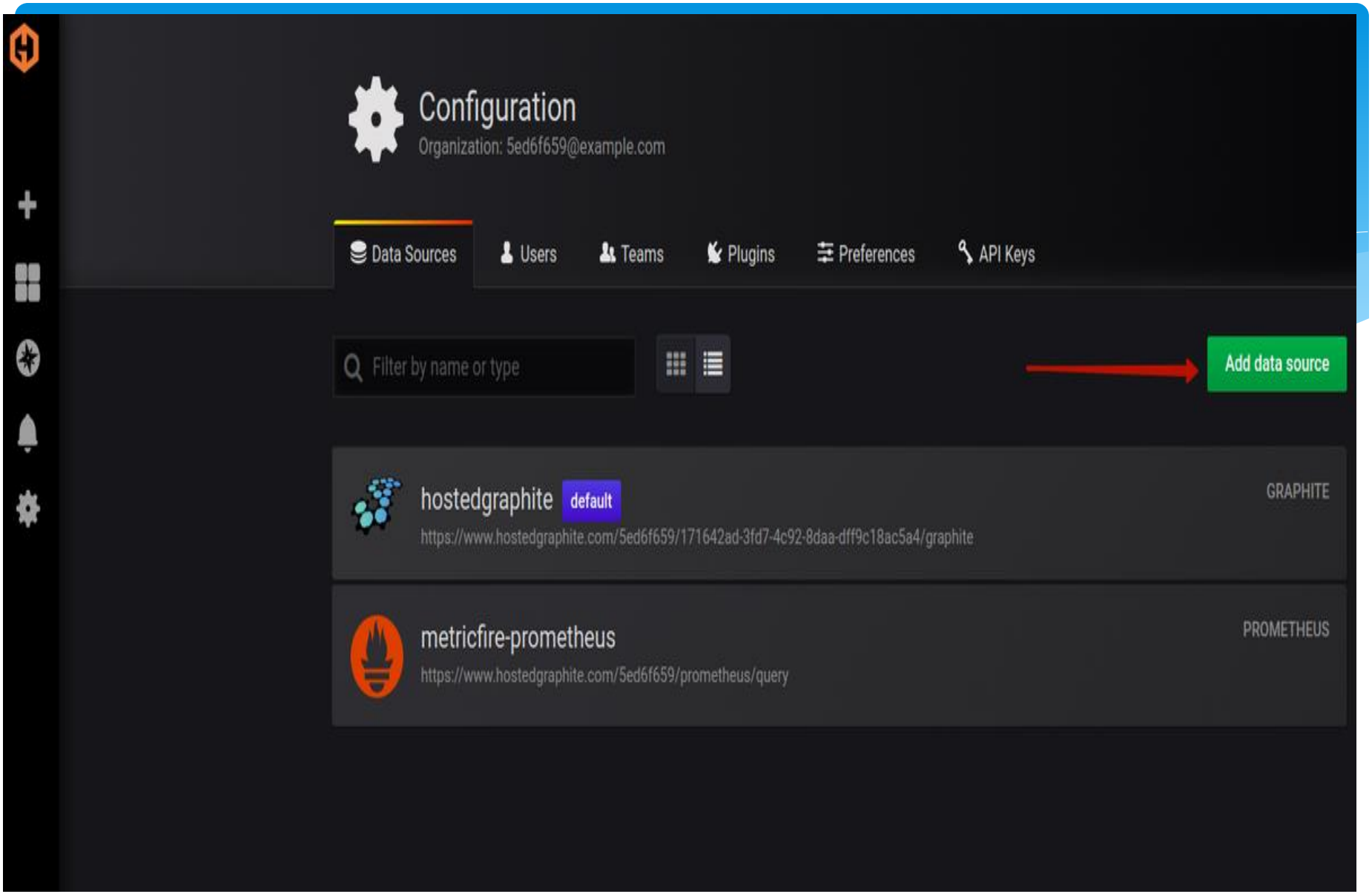
* Windows: grafana-server.exe
* Linux : grafana-server.service file
* sudo systemctl daemon-reload
* sudo systemctl start grafana-server
* sudo systemctl status grafana-server
* Configure the Grafana server to start at boot:
* sudo systemctl enable grafana-server.service

Data Source

* Grafana works with data, which must be stored in a database before it can be accessed by Grafana.
* There are several different kinds of databases.
* Additionally, some other systems can be used to store data, even though their main purpose is not focused on data storage.
* A Grafana data source is any place from which Grafana can pull data.

How to add a data source in Grafana





When you click the *Add data sources* button, you will see the list of officially supported data sources available for connection

ne series databases

Prometheus

Open source time series database & alerting

Graphite

Open source time series database

OpenTSDB

Open source time series database

InfluxDB

Open source time series database

gging & document databases

Loki

Like Prometheus but for logs. OSS logging solution from Grafana Labs

Elasticsearch

Open source logging & analytics database

Data Sources / Elasticsearch

Type: Elasticsearch

Settings

Name Elasticsearch Default

HTTP

URL http://localhost:9200

Access Server (Default) Help

Whitelisted Cookies Add Name

Auth

Basic Auth

With Credentials

TLS Client Auth

With CA Cert

Skip TLS Verify

Forward OAuth Identity

Elasticsearch details

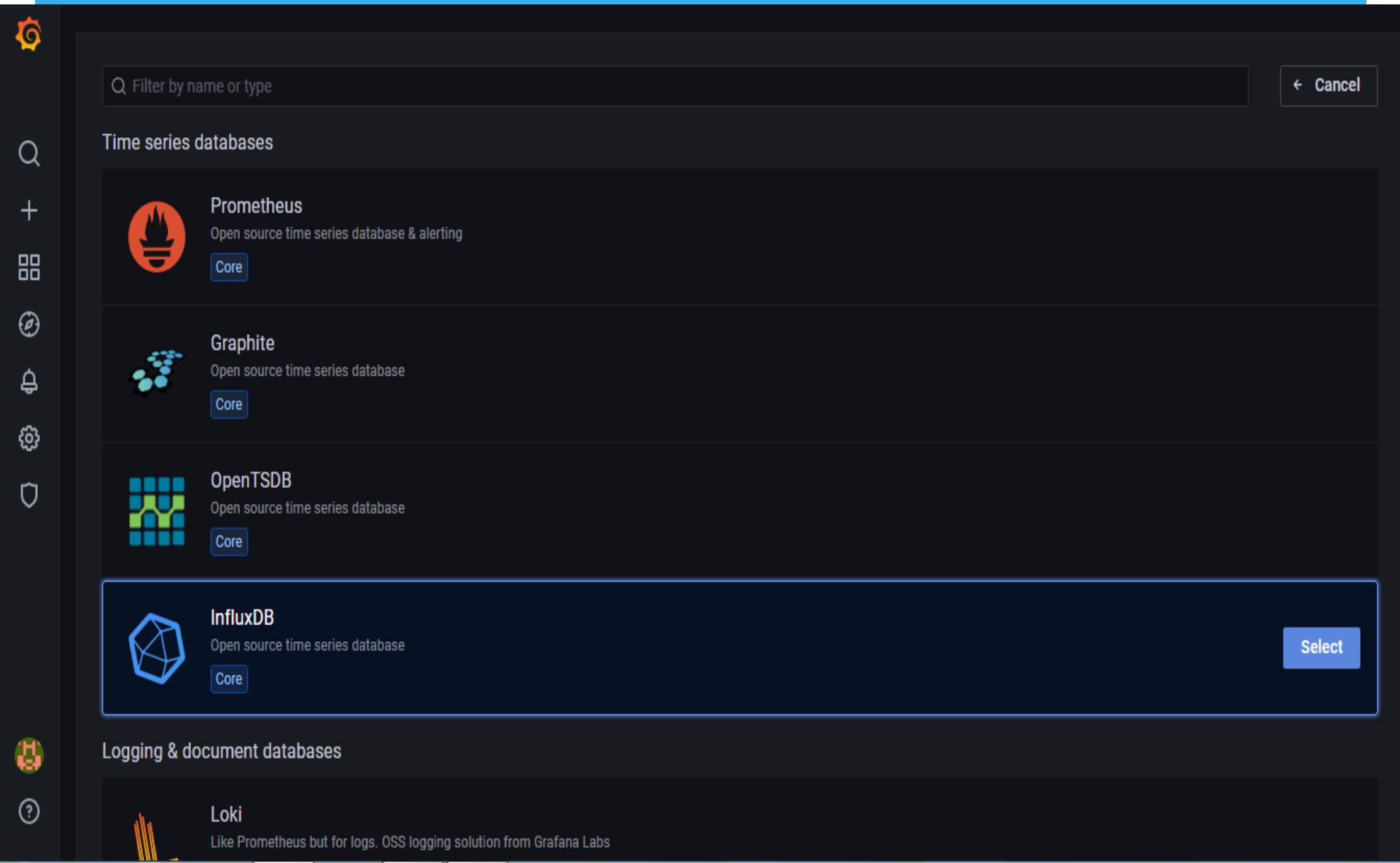
Index name

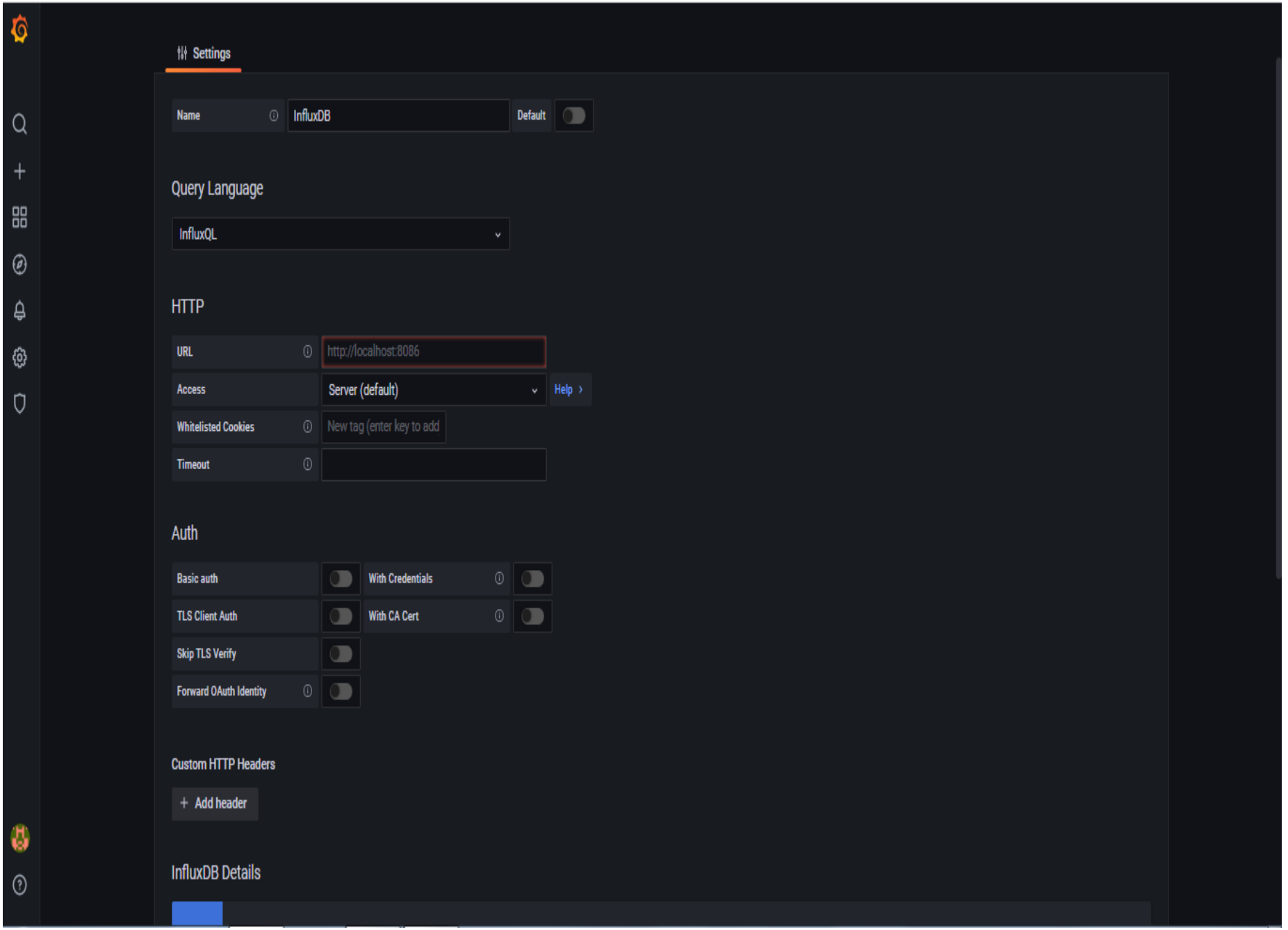
Pattern No pattern

Time field name @timestamp

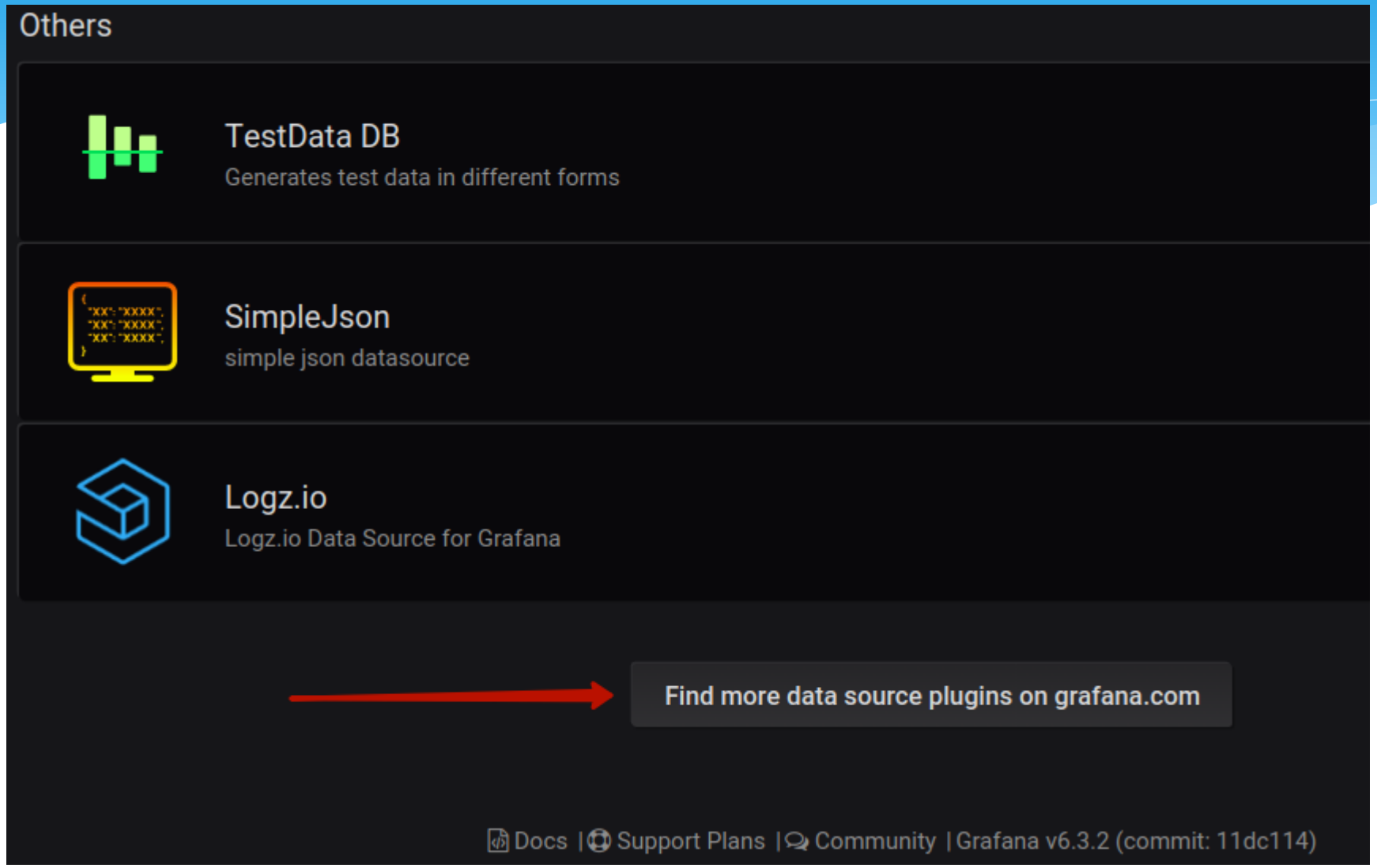
Version 5.x

Influx DB with Grafana integration




Settings
Name
InfluxDB
Default
Query Language
InfluxQL
HTTP
URL
http://localhost:8086
Access
Server (default)
Help
Whitelisted Cookies
New tag (enter key to add
Timeout
Auth
Basic auth
With Credentials
TLS Client Auth
With CA Cert
Skip TLS Verify
Forward OAuth Identity
Custom HTTP Headers
+ Add header
InfluxDB Details

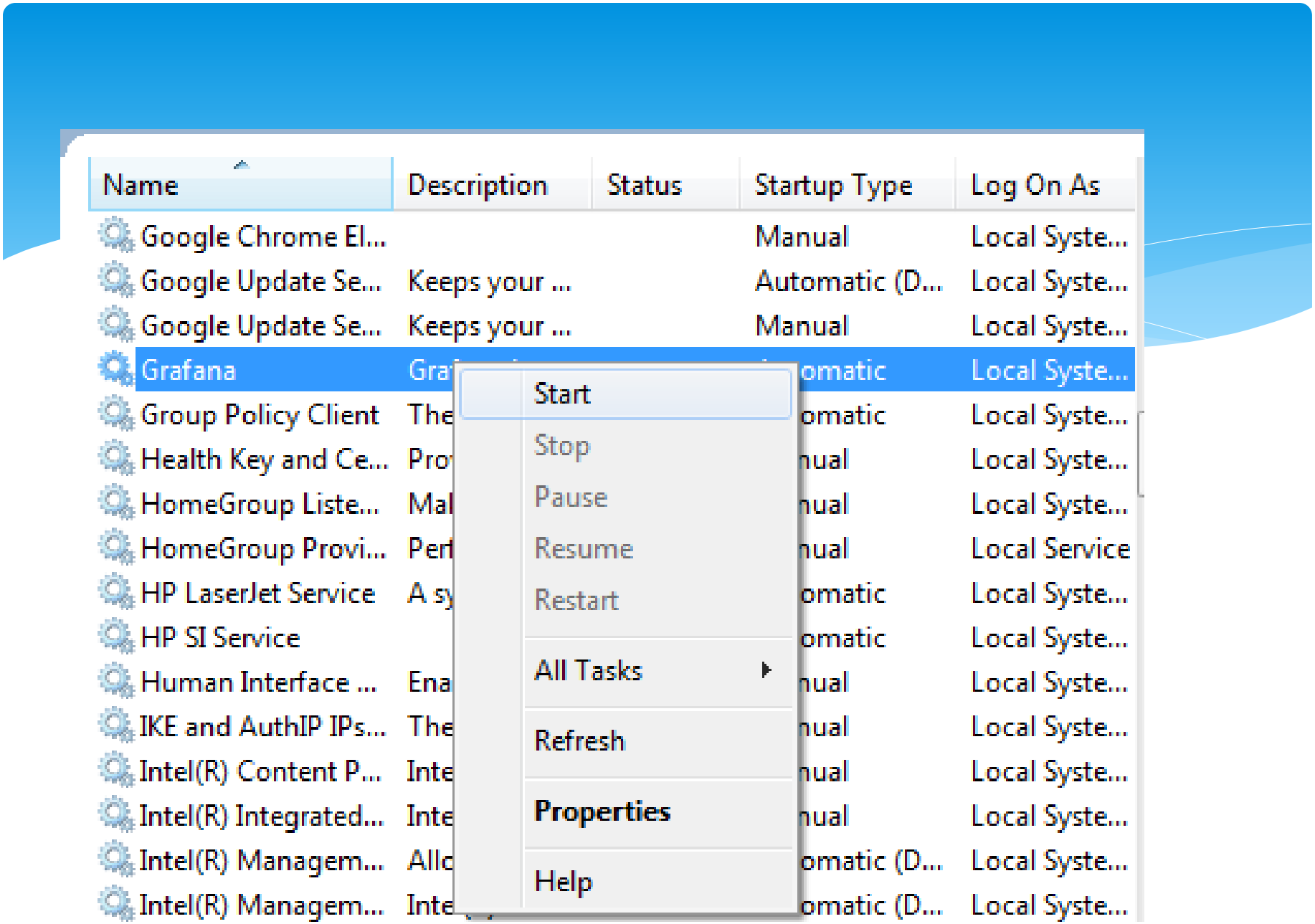
Installing third-party data sources



grafana-cli

```
D:\GRAFANA\grafana-enterprise-8.1.5.windows-amd64\grafana-8.1.5\bin>grafana-cli.
exe plugins install grafana-simple-json-datasource
-[32m←[0m Downloaded grafana-simple-json-datasource v1.4.2 zip successfully

-[32mPlease restart Grafana after installing plugins. Refer to Grafana documenta
tion for instructions if necessary.
```


Name
Description
Status
Startup Type
Log On As
Google Chrome El...
Manual
Local Syste...
Google Update Se...
Keeps your ...
Automatic (D...
Local Syste...
Google Update Se...
Keeps your ...
Manual
Local Syste...
Grafana
omatic
Local Syste...
Group Policy Client
omatic
Local Syste...
Health Key and Ce...
nual
Local Syste...
HomeGroup Liste...
nual
Local Syste...
HomeGroup Provi...
nual
Local Service
HP LaserJet Service
omatic
Local Syste...
HP SI Service
omatic
Local Syste...
Human Interface ...
nual
Local Syste...
IKE and AuthIP IPs...
nual
Local Syste...
Intel(R) Content P...
nual
Local Syste...
Intel(R) Integrated...
nual
Local Syste...
Intel(R) Managem...
omatic (D...
Local Syste...
Intel(R) Managem...
omatic (D...
Local Syste...
Start
Stop
Pause
Resume
Restart
All Tasks
Refresh
Properties
Help

```
D:\GRAFANA\grafana-enterprise-8.1.5.windows-amd64\grafana-8.1.5\bin>grafana-cli.
exe plugins install grafana-simple-json-datasource
-[32m←[0m Downloaded grafana-simple-json-datasource v1.4.2 zip successfully

-[32mPlease restart Grafana after installing plugins. Refer to Grafana documenta
tion for instructions if necessary.
```

```
D:\GRAFANA\grafana-enterprise-8.1.5.windows-amd64\grafana-8.1.5\bin>grafana-cli
plugins install marcusolsson-csv-datasource
←[32m←[0m Downloaded marcusolsson-csv-datasource v0.6.1 zip successfully

←[32mPlease restart Grafana after installing plugins. Refer to Grafana documenta
tion for instructions if necessary.
```

Prometheus

* Prometheus is an open source, metrics-based monitoring system.
* Prometheus is primarily written in Go and licensed under the Apache 2.0 license.
* In 2016 the Prometheus project became the second member of the Cloud Native Computing Foundation (CNCF).

Categories of Monitoring

* Receiving a HTTP request
* Sending a HTTP 400 response
* Entering a function
* Reaching the else of an if statement
* Leaving a function
* A user logging in
* Writing data to disk
* Reading data from the network
* Requesting more memory from the kernel

Prometheus main features

* A multidimensional data model with time series data identified by metric name and key/value pairs.
* A flexible query language to leverage this dimensionality.
* No reliance on distributed storage; single server nodes are autonomous.

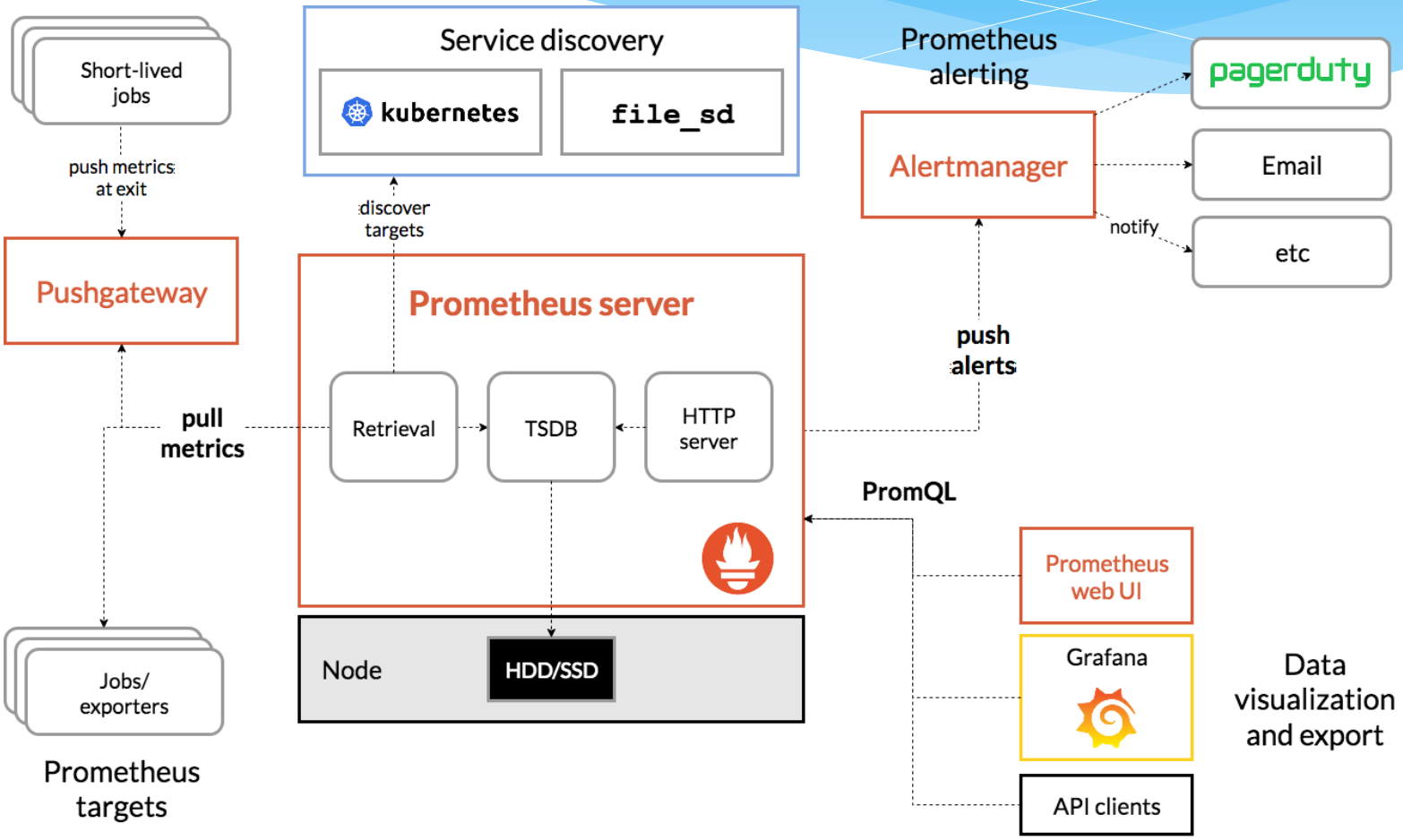
Prometheus main components

* The Prometheus server which scrapes and stores time series data.
* Client libraries for instrumenting application code.
* The Alert manager.
* A push gateway for supporting short lived jobs (optional)

Prometheus data model

* Prometheus fundamentally stores all data as time series: streams of time stamped values belonging to the same metric and the same set of labeled dimensions.
* Besides stored time series, Prometheus may generate temporary derived time series as the result of queries.

Prometheus Architecture



Prometheus Alert Manager

* The Promtheus Alertmanager handles alerts sent by client applications such as the Prometheus server.
* It takes care of de-duplicating, grouping, and routing them to the correct receiver integration such as email, PagerDuty, or OpsGenie.

Prometheus Alert Manager

* AlertManager is also capable of silencing and inhibition of alerts.
* In this context, inhibition means suppressing notifications for certain alerts if certain other alerts are already firing.

promQL

* PrometheusQueryLanguage(promQL)
* Provides built in operators and functions
* vector-based calculations like Excel
* Expressions over time-series vectors

Expression

Instant Vector-set of time series containing

single sample for each time series, all sharing same timestamp.

Examples:-

```
http_request_count
http_request_count  {status = "200"}
http_request_count  {status != "200"}
```

* Range Vector - set of time series containing a range of data points over time for each series
* http_request_count[5m]
* scalar - as a literal and as result of an expression
* string - only currently as a literal in an expression

Time Series Selectors

* **Instant Vector Selectors**
* num_nodes
* num_nodes{role="backend"}
* **Range Vector Selectors(s,m,h,d,w,y)**
* num_nodes{role="backend"}[5m]
* **Offset Modifier**
* num_nodes{role="backend"}[5m} offset 1w

Prometheus metric types

* The four Prometheus metric types
* **Counters**
* **Gauges**
* **Histograms**
* **Summaries**

Node exporter

* Set up and configured Node Exporter to collect Linux system metrics like CPU load and disk I/O.
* Node Exporter will expose these as Prometheus-style metrics.
* Configured Prometheus to scrape Node Exporter metrics and optionally ship them to Grafana Cloud.

Node exporter

* Set up a preconfigured and curated set of recording rules to cache frequent queries.
* Imported Grafana dashboards to visualize your metrics data.
* Set up Prometheus alerting rules to alert on your metrics data.

Installation

* **Step 1: Setting up Node Exporter**
* In this step you'll set up Node Exporter on your Linux machine to collect and expose system metrics.
* To begin, log in to your machine and download the relevant Node Exporter binary. In this guide we'll use linux-amd64 but you should choose the one corresponding to your system's OS and architecture:
* **wget https://github.com/prometheus/node_exporter/releases/download/v1.1.1/node_exporter-1.1.1.linux-amd64.tar.gz**

* https://prometheus.io/docs/guides/node-exporter/

* Unzip the tarball and cd into the directory:
* tar xvfz node_exporter-*.*-amd64.tar.gz cd node_exporter-*.*-amd64 Run the Node Exporter binary:
* **./node_exporter**
* If you see the above output, you successfully ran Node Exporter.
* Node Exporter publishes your system metrics in Prometheus format on port 9100.
* You can test this using curl. You will need to open a new SSH session or background the Node Exporter process to use curl.
* **curl http://localhost:9100/metrics**

* **Step 2: Scraping Node Exporter using Prometheus**
* Node Exporter is up and running on your machine, you can configure a Prometheus scrape job to collect and store Node Exporter metrics.
* Add the following scrape job config to the scrape_configs section of your **prometheus.yml** configuration file:

```
- job_name:
  node static_configs:
  - targets:
      ['linux_machine_IP_address:9100']
```

Prometheus Push gateway

* The Prometheus Pushgateway exists to allow ephemeral and batch jobs to expose their metrics to Prometheus.
* Since these kinds of jobs may not exist long enough to be scraped, they can instead push their metrics to a Pushgateway.
* The Pushgateway then exposes these metrics to Prometheus

pushgateway

* The Pushgateway is explicitly not an *aggregator or distributed counter* but rather a metrics cache.
* The metrics pushed are exactly the same as you would present for scraping in a permanently running program.
* The Pushgateway is not an *event store*. While you can use Prometheus as a data source for Grafana annotations, tracking something like release events has to happen with some event-logging framework.

Install Prometheus Push Gateway by executing following commands

sudo useradd -M -r -s /bin/false pushgateway

wget https://github.com/prometheus/pushgateway/releases/download/v1.2.0/pushgateway-1.2.0.linux-amd64.tar.gz

tar xvfz pushgateway-1.2.0.linux-amd64.tar.gz

sudo cp pushgateway-1.2.0.linux-amd64/pushgateway /usr/local/bin/

sudo chown pushgateway:pushgateway /usr/local/bin/pushgateway

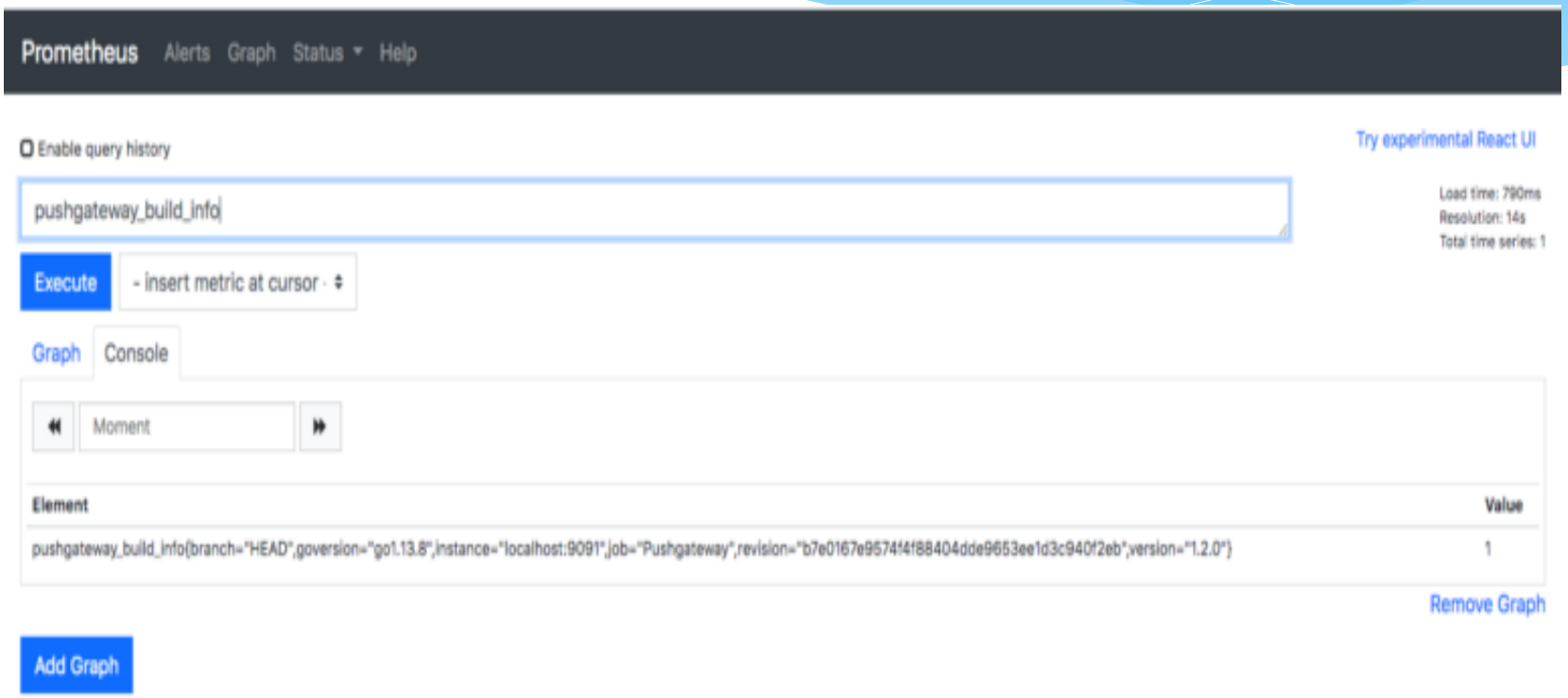
/etc/systemd/system/pushgateway.service

* Create a systemd unit file named /etc/systemd/system/pushgateway.service with following content:
* [Unit]
  Description=Prometheus Pushgateway
  Wants=network-online.target
  After=network-online.target[Service]
  User=pushgateway
  Group=pushgateway
  Type=simple
  ExecStart=/usr/local/bin/pushgateway[Install]
  WantedBy=multi-user.target

* Start PushGateway service by executing following commands:
* sudo systemctl enable pushgateway
  sudo systemctl start pushgateway
  sudo systemctl status pushgateway

* To configure Pushgateway as a Scrape Target for Prometheus Server.
* Add following lines at the end of /etc/prometheus/prometheus.yml file
* - job_name: 'Pushgateway'
  honor_labels: true
  static_configs:
  - targets: ['localhost:9091']

pushgateway_build_info query in Prometheus expression browser.



* Service discovery is designed to integrate with the machine and service databases.
* Prometheus 2.2.1 has support for Azure, Consul, DNS, EC2, OpenStack, File, Kubernetes, Marathon, Nerve, Serverset, and Triton service discovery in addition to the static discovery you have already seen.

* A good service discovery mechanism will provide you with *metadata*.
* Metadata is what you will convert into target labels.

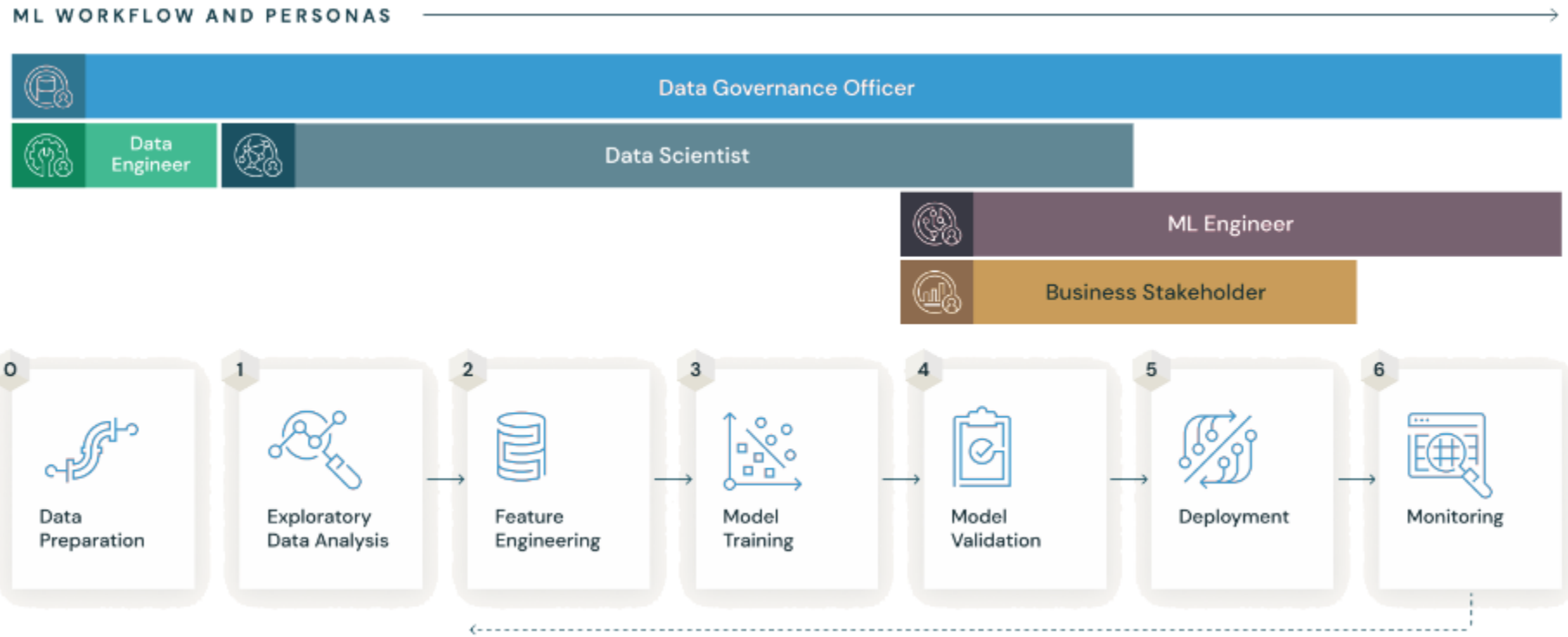
MLflow

* MLflow is an open source platform for managing machine learning workflows.
* MLflow REST API allows you to create, list, and get experiments and runs, and allows you to log parameters, metrics, and artifacts.
* It is used by MLOps teams and data scientists.
* It Tracking ML experiments to record and compare model parameters, evaluate performance, and manage artifacts

Components of MLflow

* Tracking
* Projects
* Models
* Model Registry

MLflow


ML WORKFLOW AND PERSONAS
Data Governance Officer
Data Engineer
Data Scientist
ML Engineer
Business Stakeholder
0
Data Preparation
1
Exploratory Data Analysis
2
Feature Engineering
3
Model Training
4
Model Validation
5
Deployment
6
Monitoring

Reference

* https://grafana.com/docs/
* https://prometheus.io/docs/introduction/overview/
* https://ml-ops.org/content/mlops-principles
* https://docs.h2o.ai/mlops/

Thank you