پاکستان سول ایوی ایشن اتھارٹی

Prepared by – Sohail Ahmed

Position – IT Intern

Assignment Title – Grafana Installation, Configuration & Building the First Dashboard

Supervisor – Sir Noman Rajput, Sr. Assistant Director IT

Grafana Installation, Configuration, and Building the First Dashboard

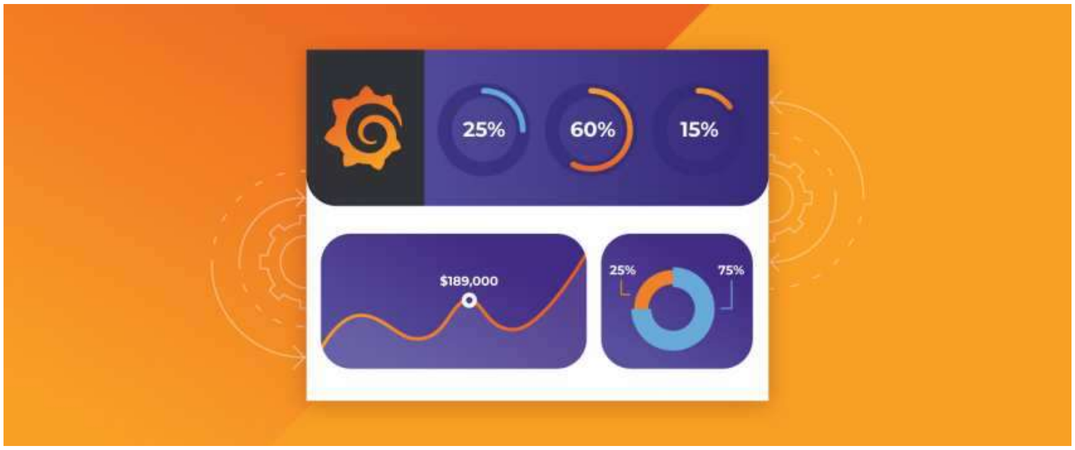


Table of Contents

1. Introduction to Grafana ____ 03
2. Installation ____ 03
   i. Linux ____ 03
   ii. Windows ____ 03
3. Getting Started ____ 04
4. Data Sources ____ 05
5. Dashboards ____ 06
6. Panels ____ 07
7. Visualization ____ 07
8. Alerts ____ 09
9. Administration ____ 09
10. Integrations ____ 11

1. Introduction to Grafana

Grafana is an open-source platform for monitoring and observability, allowing users to query, visualize, alert, and understand metrics from various data sources. It provides a rich set of features for creating dynamic dashboards and exploring data insights.

2. Installation

Linux

System Requirements:

Linux system with a supported version (e.g., Ubuntu, CentOS)

Access to the internet for downloading Grafana packages

Steps:

1. **Update Package Repository:**

   sudo apt update

2. **Install Grafana:**

   sudo apt install -y grafana

3. **Start Grafana Server:**

   sudo systemctl start grafana-server

4. **Access Grafana:**

Open a web browser and navigate to **http://localhost:3000.** Log in with the default credentials (username: **admin**, password: **admin**). Change the password after logging in for the first time.

Windows

System Requirements:

- Windows 7 or later (64-bit)
- 1GB RAM or higher
- 1.5 GHz processor or faster
- 800x600 resolution or higher
- Access to the internet for downloading Grafana installer

Steps:

1. **Download Grafana Installer:**

   Download the Grafana installer for Windows from the official Grafana website.

2. **Run Installer:**

   Run the downloaded installer (**grafana-x.x.x.windows-amd64.msi**), follow the installation wizard, and complete the installation process.

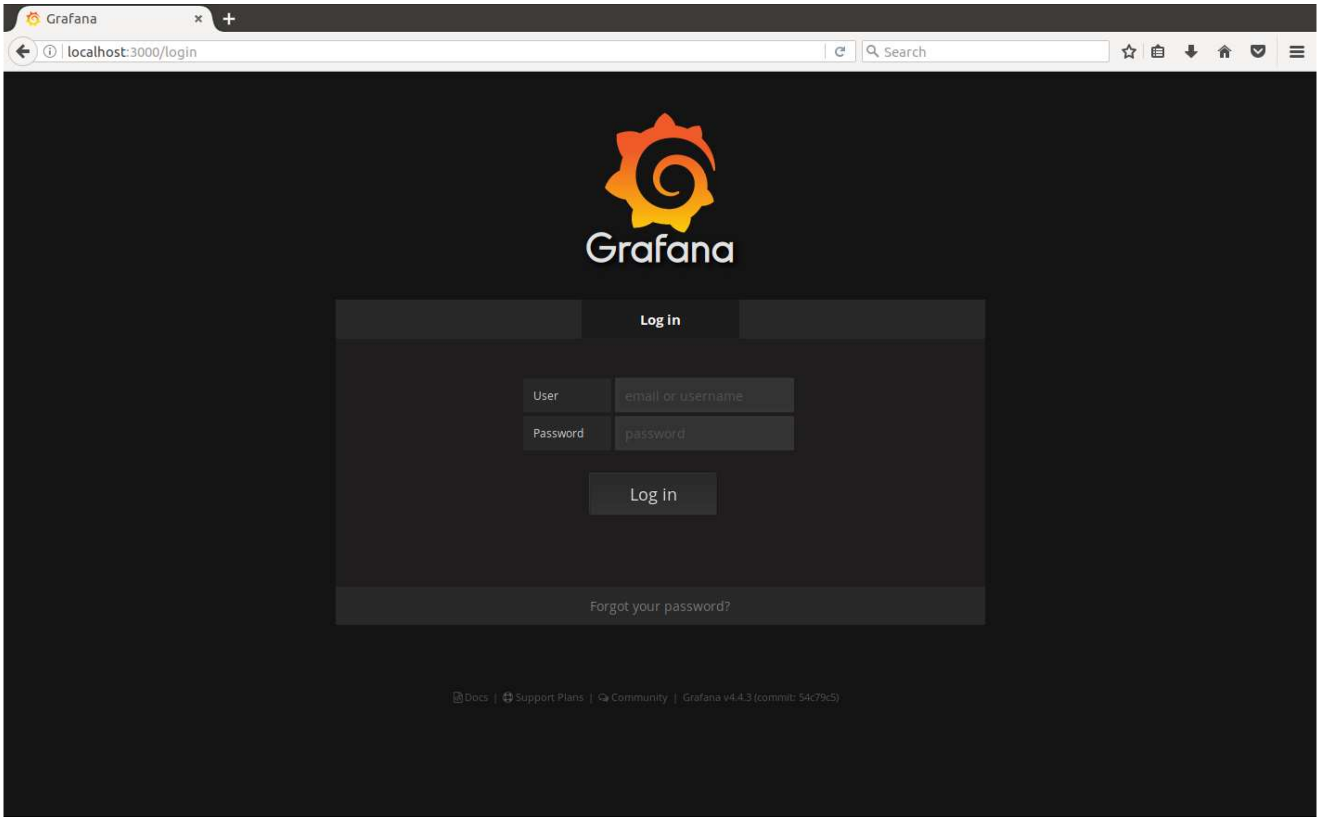
3. Access Grafana:

Open a web browser and navigate to http://localhost:3000. Log in with the default credentials (username: **admin**, password: **admin**). Change the password after logging in for the first time.

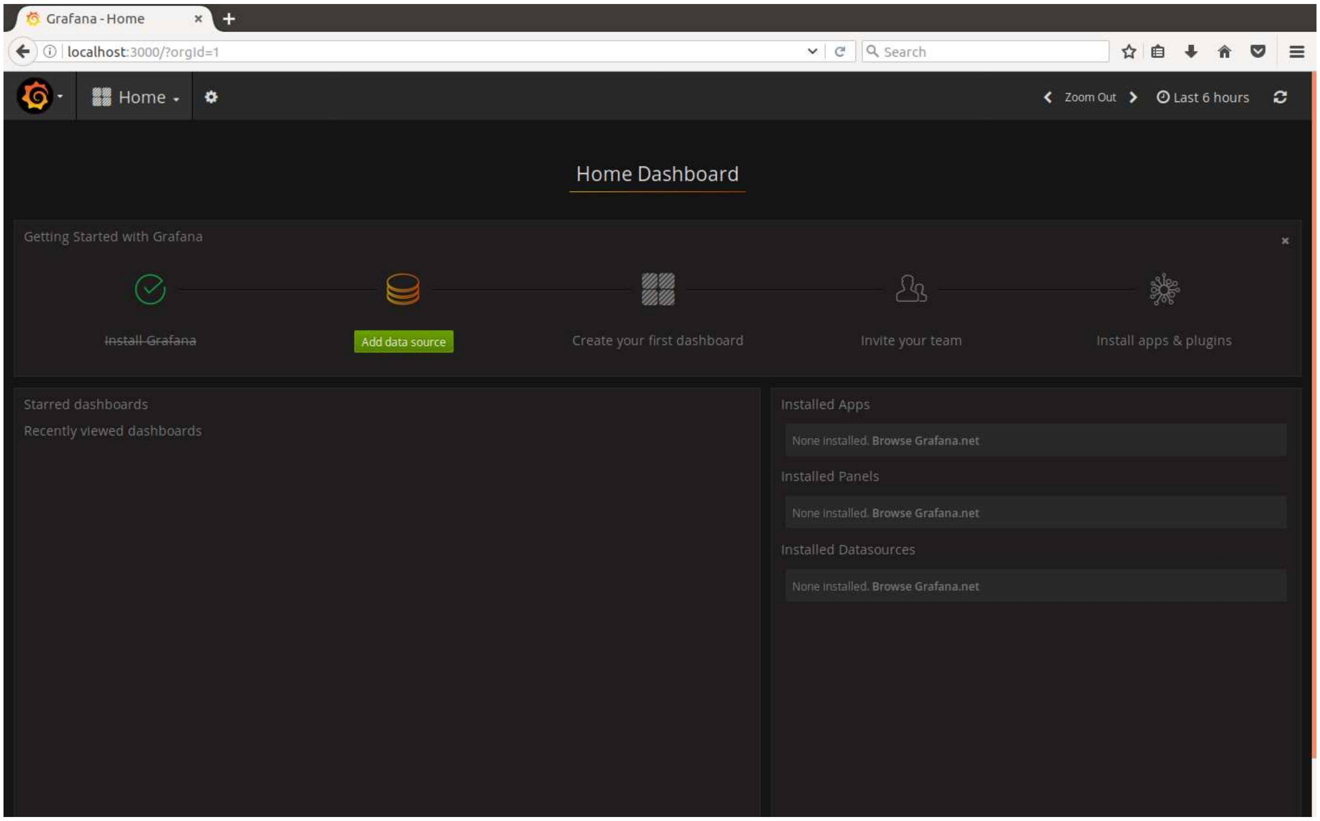
3. Getting Started

Logging In

1. **Open Web Browser:** Launch your preferred web browser.
2. **Navigate to Grafana**: In the address bar, type http://localhost:3000and press Enter. If Grafana is installed on a remote server, replace **localhost** with the server's IP address or hostname.



3. **Login:** You will be directed to the Grafana login page. Use the default credentials:
   - Username: **admin**
   - Password: **admin**

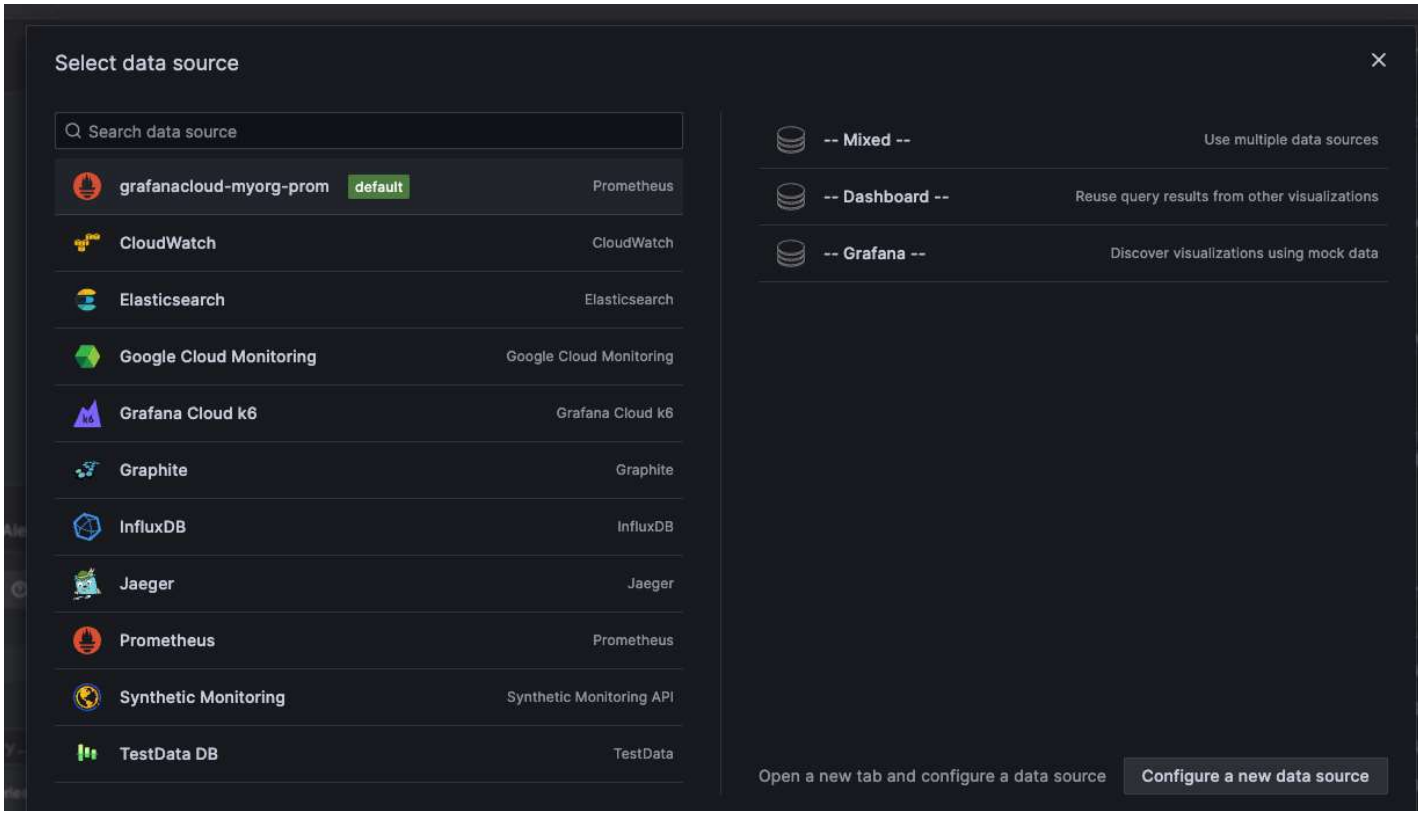


4. **Change Default Password (Optional):** After logging in for the first time, Grafana will prompt you to change the default password. Follow the on-screen instructions to set a new password.

4. Adding Data Sources

1. **Navigate to Data Sources:** After logging in, click on the gear icon (Configuration) in the sidebar, then select "Data Sources".
2. **Add Data Source:** Click on the "Add data source" button.
3. **Select Data Source Type:** Choose the type of data source you want to add (e.g., Prometheus, InfluxDB, Elasticsearch).
4. **Configure Data Source:** Fill in the required fields such as URL, access credentials, and other settings specific to your data source.

5. **Test Connection:** Click on the "Save & Test" button to test the connection to the data source. If successful, you will see a confirmation message.



Select data source

Search data source

grafanacloud-myorg-prom default	Prometheus
CloudWatch	CloudWatch
Elasticsearch	Elasticsearch
Google Cloud Monitoring	Google Cloud Monitoring
Grafana Cloud k6	Grafana Cloud k6
Graphite	Graphite
InfluxDB	InfluxDB
Jaeger	Jaeger
Prometheus	Prometheus
Synthetic Monitoring	Synthetic Monitoring API
TestData DB	TestData

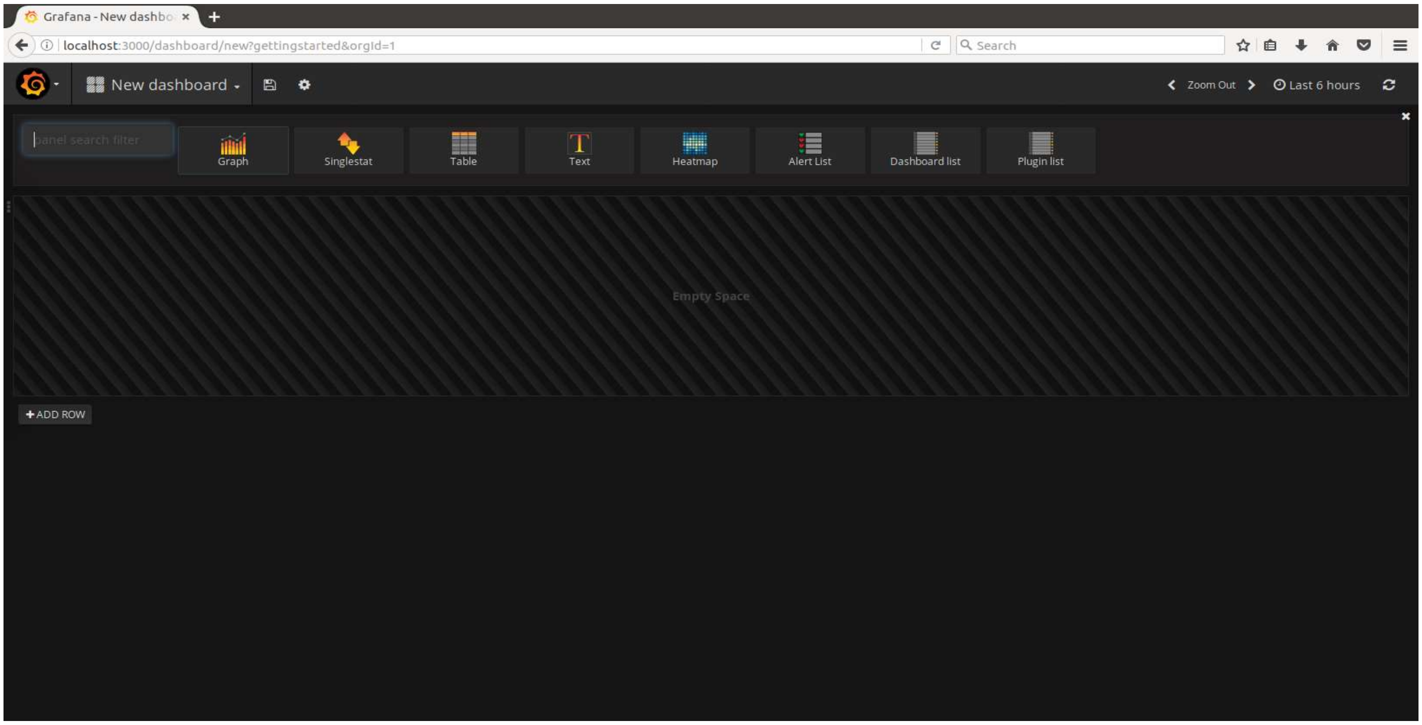
-- Mixed --	Use multiple data sources
-- Dashboard --	Reuse query results from other visualizations
-- Grafana --	Discover visualizations using mock data

Open a new tab and configure a data source

Configure a new data source

5. Creating Dashboards

1. **Navigate to Dashboards:** Click on the Grafana logo in the top left corner to return to the homepage. From the sidebar, click on the "Create" (+) icon, then select "Dashboard".
2. **Add Panel:** Click on the "Add new panel" button.



3. **Select Visualization:** Choose the type of visualization you want to add to your panel (e.g., Graph, Singlestat, Table).
4. **Configure Panel:** Customize the panel settings, including data source, queries, and visualization options.
5. **Save Dashboard:** Once you have configured your panels, click on the "Save" icon in the top right corner and give your dashboard a name.

6.Adding Panels

1. **Edit Dashboard:** Open the dashboard you want to add panels to.
2. **Add Panel:** Click on the "Add panel" button in the top right corner of the dashboard.
3. **Select Visualization:** Choose the type of visualization you want to add from the list of available panel types.
4. **Configure Panel:** Customize the panel settings according to your requirements.
5. **Position Panel:** Drag and drop the panel to the desired location on the dashboard layout.
6. **Save Dashboard:** Once you have added and configured all the panels, click on the "Save" icon to save your changes to the dashboard.

7.Visualization

Graph

Description: The Graph panel in Grafana allows users to visualize time-series data as line graphs, bar graphs, or stacked area graphs. It supports multiple data series and offers various customization options for axes, legends, and annotations.

Steps:

1. Add a Graph panel to your dashboard.
2. Configure the data source and query to fetch the time-series data.
3. Customize the graph appearance, including axes, legends, and visualization options.
4. Apply any additional settings or annotations as needed.
5. Save the panel configuration.

Singlestat

Description: The Singlestat panel is used to display a single value from a metric query, such as the current value of a sensor, the total count of events, or the average response time of a service.

Steps:

1. Add a Singlestat panel to your dashboard.
2. Configure the data source and query to fetch the metric value.
3. Customize the appearance of the single value display, including thresholds, colors, and units.
4. Apply any additional settings or annotations as needed.
5. Save the panel configuration.

Table

Description: The Table panel allows users to display tabular data retrieved from a data source query. It supports features such as sorting, filtering, and pagination, making it suitable for presenting detailed data sets.

Steps:

1. Add a Table panel to your dashboard.
2. Configure the data source and query to retrieve the tabular data.
3. Customize the table appearance, including column layout, sorting, and filtering options.
4. Apply any additional settings or annotations as needed.
5. Save the panel configuration.

Heatmap

Description: The Heatmap panel visualizes data as a grid of colored cells, where the color intensity represents the data value. It is useful for identifying patterns or trends in large data sets, such as server response times or network traffic.

Steps:

1. Add a Heatmap panel to your dashboard.
2. Configure the data source and query to retrieve the data for the heatmap.
3. Customize the appearance of the heatmap, including color mapping, gradient scales, and tooltips.
4. Apply any additional settings or annotations as needed.
5. Save the panel configuration.

Gauge

Description: The Gauge panel displays a single value within a specified range, represented as a gauge with a needle or dial. It is commonly used to visualize metrics such as CPU usage, disk space, or temperature.

Steps:

1. Add a Gauge panel to your dashboard.
2. Configure the data source and query to fetch the metric value.
3. Define the gauge settings, including minimum and maximum values, thresholds, and colors.
4. Apply any additional settings or annotations as needed.
5. Save the panel configuration.

Plugins and Custom Visualizations

Description: Grafana supports a wide range of plugins and custom visualizations that extend its capabilities beyond the built-in panels. Users can install community-developed plugins or develop their own custom visualizations using JavaScript libraries such as D3.js or Plotly.

Steps:

1. Explore the Grafana Plugin Repository to find plugins or custom visualizations that meet your requirements.
2. Install the desired plugin or follow the documentation to create a custom visualization.
3. Add the plugin or custom visualization to your dashboard and configure it according to your needs.
4. Save the panel configuration.

8. Alerts

Alerting Rules

Description: Alerting rules in Grafana allow users to define conditions based on metrics queries and trigger alerts when these conditions are met. Alerts can be configured to send notifications via various channels, such as email, Slack, or PagerDuty.

Steps:

1. Navigate to the dashboard containing the panel for which you want to set up an alert.
2. Edit the panel and click on the "Alert" tab.
3. Click on the "Create Alert" button to define a new alert rule.
4. Configure the alert rule by specifying the conditions that trigger the alert, such as thresholds or patterns in the data.
5. Define the alert notification settings, including the severity level, notification channels, and message templates.
6. Save the alert rule configuration.

Notification Channels

Description: Notification channels in Grafana define how alerts are sent to users or external systems. Supported notification channels include email, Slack, PagerDuty, and more.

Steps:

1. Navigate to the Grafana main menu and select "Alerting" > "Notification channels".
2. Click on the "Add Channel" button to create a new notification channel.
3. Choose the type of notification channel you want to add (e.g., Email, Slack, PagerDuty).
4. Configure the settings specific to the selected notification channel, such as email addresses, API tokens, or webhook URLs.
5. Test the notification channel to ensure that alerts are delivered successfully.
6. Save the notification channel configuration.

Alerting Configuration

Description: Alerting configuration in Grafana allows users to manage global settings related to alerting, such as default notification channels, alert rules evaluation frequency, and alert annotations.

Steps:

1. Navigate to the Grafana main menu and select "Alerting" > "Alert rules".
2. Click on the "Notification channels" tab to configure default notification channels for alerts.
3. Configure other alerting settings as needed, such as evaluation frequency and alert annotations.
4. Save the alerting configuration.

9. Administration

User Management

Description: User management in Grafana allows administrators to create, modify, and delete user accounts, as well as assign roles and permissions to control access to dashboards and features.

Steps:

1. Navigate to the Grafana main menu and select "Configuration" > "Users".
2. Click on the "Add new user" button to create a new user account.
3. Fill in the user details, including username, email, and password.
4. Assign roles and permissions to the user, such as Viewer, Editor, or Admin.
5. Save the user account.

Security

Description: Security features in Grafana help administrators protect sensitive data and secure access to the Grafana instance. This includes configuring authentication methods, enabling SSL/TLS encryption, and implementing access controls.

Steps:

1. Navigate to the Grafana main menu and select "Configuration" > "Server Admin" > "Settings".
2. Configure authentication settings, such as LDAP, OAuth, or built-in authentication.
3. Enable SSL/TLS encryption to secure communications between clients and the Grafana server.
4. Set up access controls and permissions to restrict access to sensitive data and features.
5. Save the configuration changes and restart the Grafana server if necessary.

Plugins

Description: Grafana plugins extend the functionality of the platform by adding new data sources, visualizations, or integrations. Administrators can manage installed plugins, configure plugin settings, and install or remove plugins as needed.

Steps:

1. Navigate to the Grafana main menu and select "Configuration" > "Plugins".
2. View installed plugins and their status (enabled/disabled).
3. Configure plugin settings, such as data source connections or visualization options.
4. Install new plugins from the Grafana Plugin Repository or remove existing plugins.
5. Save the plugin configuration changes.

Backup and Restore

Description: Backup and restore procedures in Grafana ensure that critical data, configurations, and dashboards are preserved and can be recovered in case of system failures or data loss events.

Steps:

1. Backup Grafana configuration files, database, and any additional data sources or plugins.
2. Implement regular backup schedules to capture changes and updates to Grafana settings and dashboards.
3. Store backup files in a secure location, such as an external server or cloud storage provider.
4. Test the backup and restore procedures periodically to ensure they are functional and reliable.
5. In case of a system failure or data loss event, follow the restore procedures to recover the Grafana instance and its data.

Logging

Description: Logging in Grafana helps administrators monitor system activities, diagnose issues, and troubleshoot errors. Grafana logs contain information about user actions, server events, and system performance metrics.

Steps:

1. Configure logging settings in Grafana to specify log levels, log file locations, and log rotation policies.
2. Monitor log files regularly for any abnormal activities, errors, or warnings.
3. Use log analysis tools or Grafana dashboards to visualize log data and identify trends or patterns.
4. Investigate and resolve any issues or errors reported in the logs to maintain system reliability and performance.

Metrics

Description: Metrics monitoring in Grafana allows administrators to track system performance, resource utilization, and application health metrics. Grafana integrates with various monitoring systems and data sources to collect and visualize metrics data.

Steps:

1. Configure data sources to ingest metrics data from monitoring systems such as Prometheus, InfluxDB, or Graphite.
2. Create dashboards and panels to visualize metrics data in real-time or historical views.
3. Set up alerting rules to trigger notifications based on predefined thresholds or conditions in the metrics data.
4. Monitor system metrics regularly to identify performance bottlenecks, capacity issues, or anomalies.
5. Use Grafana's built-in or custom visualization tools to analyze metrics data and gain insights into system behavior and performance trends.

10. Integrations

Grafana with Prometheus

Description: Prometheus is a popular open-source monitoring and alerting toolkit. Integrating Prometheus with Grafana allows users to visualize Prometheus metrics data and create custom dashboards for monitoring systems and applications.

Steps:

1. Install and configure Prometheus according to your environment and monitoring requirements.
2. Add Prometheus as a data source in Grafana by specifying the Prometheus server URL.
3. Create dashboards and panels in Grafana to visualize Prometheus metrics using PromQL queries.
4. Customize dashboard layouts, visualization options, and alerting rules based on Prometheus metrics data.
5. Monitor system performance, resource utilization, and application health metrics using Grafana dashboards integrated with Prometheus data.

Grafana with InfluxDB

Description: InfluxDB is a time-series database designed for handling high volumes of timestamped data. Integrating InfluxDB with Grafana allows users to query and visualize time-series data stored in InfluxDB and create dynamic dashboards for monitoring and analytics.

Steps:

1. Install and configure InfluxDB according to your environment and data storage requirements.
2. Add InfluxDB as a data source in Grafana by providing the InfluxDB server URL and authentication credentials.
3. Define queries in Grafana to retrieve time-series data from InfluxDB and configure visualization options for displaying the data.
4. Create dashboards in Grafana to visualize metrics such as CPU usage, memory usage, disk I/O, and network traffic stored in InfluxDB.
5. Customize dashboard layouts, panel settings, and alerting rules based on InfluxDB metrics data to monitor system performance and application health.

Grafana with Elasticsearch

Description: Elasticsearch is a distributed search and analytics engine designed for storing and querying large volumes of structured and unstructured data. Integrating Elasticsearch with Grafana allows users to visualize Elasticsearch query results and create interactive dashboards for exploring log data, metrics, and analytics.

Steps:

1. Install and configure Elasticsearch according to your environment and data indexing requirements.
2. Add Elasticsearch as a data source in Grafana by specifying the Elasticsearch server URL and authentication credentials.
3. Define queries in Grafana using Elasticsearch Query DSL to retrieve data from Elasticsearch indices.
4. Configure visualization options in Grafana to display Elasticsearch query results as graphs, tables, or other visualizations.
5. Create dashboards in Grafana to visualize log data, metrics, and analytics stored in Elasticsearch and monitor system performance, application logs, and user activity.

Grafana with MySQL

Description: MySQL is a popular open-source relational database management system (RDBMS) used for storing structured data. Integrating MySQL with Grafana allows users to visualize MySQL query results and create interactive dashboards for monitoring database performance, query latency, and system health metrics.

Steps:

1. Install and configure MySQL according to your environment and database schema requirements.
2. Add MySQL as a data source in Grafana by providing the MySQL server URL and authentication credentials.
3. Define SQL queries in Grafana to retrieve data from MySQL tables and views.
4. Configure visualization options in Grafana to display MySQL query results as graphs, tables, or other visualizations.
5. Create dashboards in Grafana to monitor database performance metrics such as query throughput, response time, and resource utilization.

Grafana with Graphite

Description: Graphite is a scalable and highly available metrics storage and visualization platform. Integrating Graphite with Grafana allows users to query and visualize metrics data stored in Graphite and create custom dashboards for monitoring system performance, application metrics, and infrastructure health.

Steps:

1. Install and configure Graphite according to your environment and data storage requirements.
2. Add Graphite as a data source in Grafana by specifying the Graphite server URL and authentication credentials.
3. Define queries in Grafana to retrieve metrics data from Graphite metrics.
4. Configure visualization options in Grafana to display Graphite query results as graphs, tables, or other visualizations.
5. Create dashboards in Grafana to monitor system performance metrics such as CPU usage, memory usage, disk I/O, and network traffic stored in Graphite.