

Monitoring System Guide

SYSTEM DESCRIPTION AND INSTALLATION GUIDE

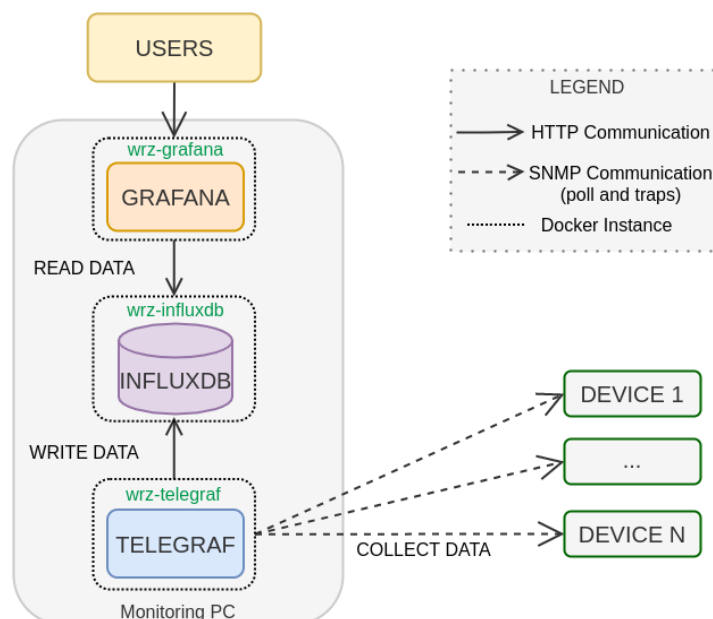
1. Introduction

The Safran Spain Monitoring System is a set of software and tools which is distributed with the main objective of providing knowledge about the timing network. It is installed externally and queries the deployed systems to know the state of the network.

This document describes the system architecture and the instructions for the system installation. The deployment steps, including the dependencies installation, will be shown. Different commands and services will be explained, so the user can manage the tool on his own.

2. Monitoring System Solution

The monitoring suite is a set of three containers that run the services. They communicate with each other to read and write data in InfluxDB. The containers will run on the same server.



3. Deployment steps

This section shows how to install and configure the monitoring suite. The 3.3 subsection includes a step-by-step explanation. After the tool installation, as will be shown below, some actions will be necessary to be applied in Grafana to visualize all the provided functionalities.

3.1. Requirements

First, install the tools list below in the server:

- Docker and Docker Compose
- Python v3.7 or later and pip

3.2. Directory structure

The tar file received with this document includes the following content:

- config.yml
- docker-compose.yml
- Data source and dashboard templates
- Templates to schedule tasks in InfluxDB
- Mibs files
- README.md
- Telegraf configuration files
- Python module wrz_mon
- wrz_monitoring.py

3.3. Install Monitoring System

As shown in the previous section, the installation file includes a Python script for easy deployment and configuration.

To run the Python file, first, extract the content of the tar file by using the following command:

```
sudo tar -xf wrz_monitoring.tar.gz
```

It is recommended to edit the **config.yml** file before performing the next action (before the installation). See section 4 to understand how this file should be modified. If the specific network configuration is applied at this point it will not be necessary to perform a configuration update after the installation.

Then, execute the following command to perform the installation. The installation script (wrz_monitoring.py) modifies the administrator account in Influx and Grafana services. It sets the user and password provided as input.

```
sudo ./wrz_monitoring.py up-services --username USER --password PASS
```

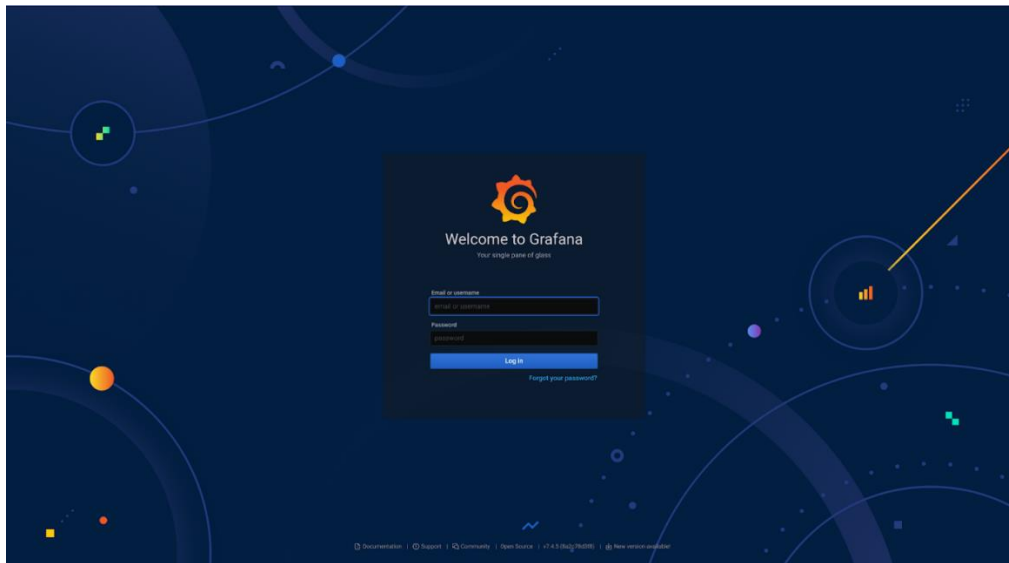
A correct installation should print the following logs:

```
wrz-monitoring: INFO: Configuration OK
wrz-monitoring: INFO: Docker containers started
wrz-monitoring: WARNING: InfluxDB doesn't respond. Sleep 5 seconds
wrz-monitoring: INFO: InfluxDB service started
wrz-monitoring: INFO: InfluxDB token generated
wrz-monitoring: INFO: InfluxDB configured
wrz-monitoring: INFO: NTP Server Status Tasks Check imported successfully
wrz-monitoring: INFO: Healthingd Status Tasks Check imported successfully
wrz-monitoring: INFO: System Status Tasks Check imported successfully
wrz-monitoring: INFO: Timing Status Tasks Check imported successfully
wrz-monitoring: INFO: Leapsec File Status Tasks Check imported successfully
wrz-monitoring: INFO: InfluxDB successfully configured
wrz-monitoring: INFO: Telegraf configuration files copied to docker instance
wrz-monitoring: INFO: Telegraf restarted
wrz-monitoring: INFO: Telegraf successfully configured
wrz-monitoring: INFO: Grafana service started
wrz-monitoring: INFO: Grafana token generated
wrz-monitoring: INFO: Grafana organization changed
wrz-monitoring: INFO: Grafana user and password updated
wrz-monitoring: INFO: Folder Custom created
wrz-monitoring: INFO: InfluxDB datasource imported
wrz-monitoring: INFO: Dashboard Time sources imported to Grafana
wrz-monitoring: INFO: Dashboard Device Interfaces imported to Grafana
wrz-monitoring: INFO: Dashboard TSRC VCSCodes imported to Grafana
wrz-monitoring: INFO: Dashboard Offset from master imported to Grafana
wrz-monitoring: INFO: Dashboard Device Load imported to Grafana
wrz-monitoring: INFO: Dashboard Device SFP imported to Grafana
wrz-monitoring: INFO: Dashboard External Reference Signals imported to Grafana
wrz-monitoring: INFO: Dashboard Interfaces imported to Grafana
wrz-monitoring: INFO: Dashboard Device Timing imported to Grafana
wrz-monitoring: INFO: Dashboard Vclock VCSCodes imported to Grafana
wrz-monitoring: INFO: Dashboard Device Overview imported to Grafana
wrz-monitoring: INFO: Dashboard Healthing imported to Grafana
wrz-monitoring: INFO: Dashboard Home imported to Grafana
wrz-monitoring: INFO: Dashboard Device PWS imported to Grafana
wrz-monitoring: INFO: Dashboard Device Timing Servo imported to Grafana
wrz_monitoring: To visualize data in Grafana UI, insert InfluxDB token in the Grafana datasource.
InfluxDB Token:
b0e1b8c3dc88dcee4b50c7cad2a84daa28d66ff006085a216e94d6891ae01121b38475be643de52
4000d37==
```

To conclude the installation, access Grafana web interfaces, set the InfluxDB token in Grafana, change the Grafana Home page, and draw the network topology. The following subsections explain the full procedure.

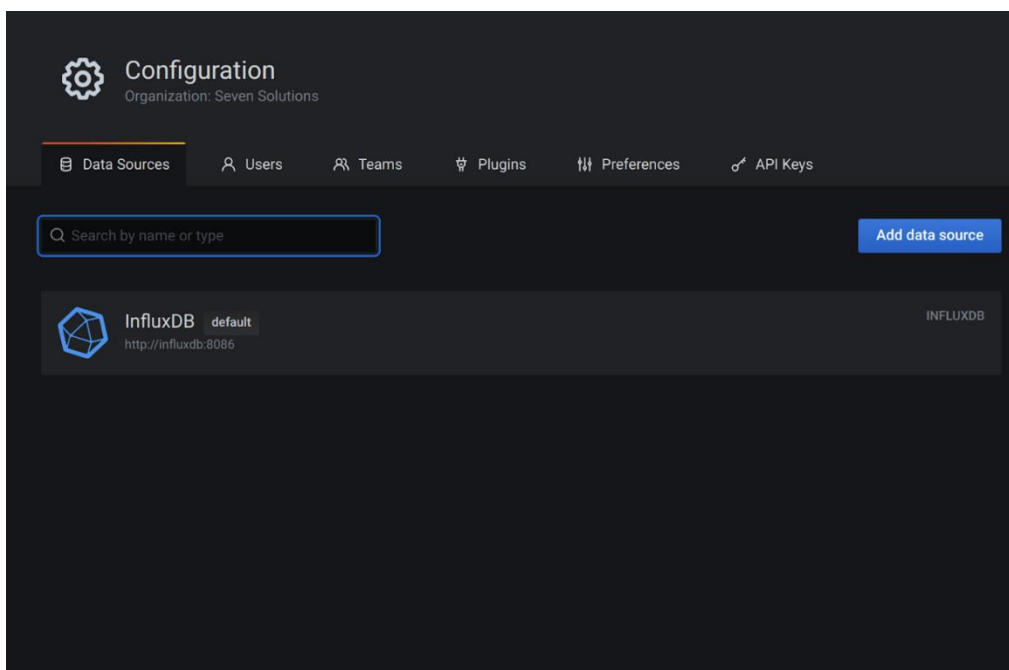
3.3.1. Login to Grafana

Grafana runs on port 3000. For this reason, it is possible to access through the web performing a localhost connection.



3.3.2. Set InfluxDB token in Grafana

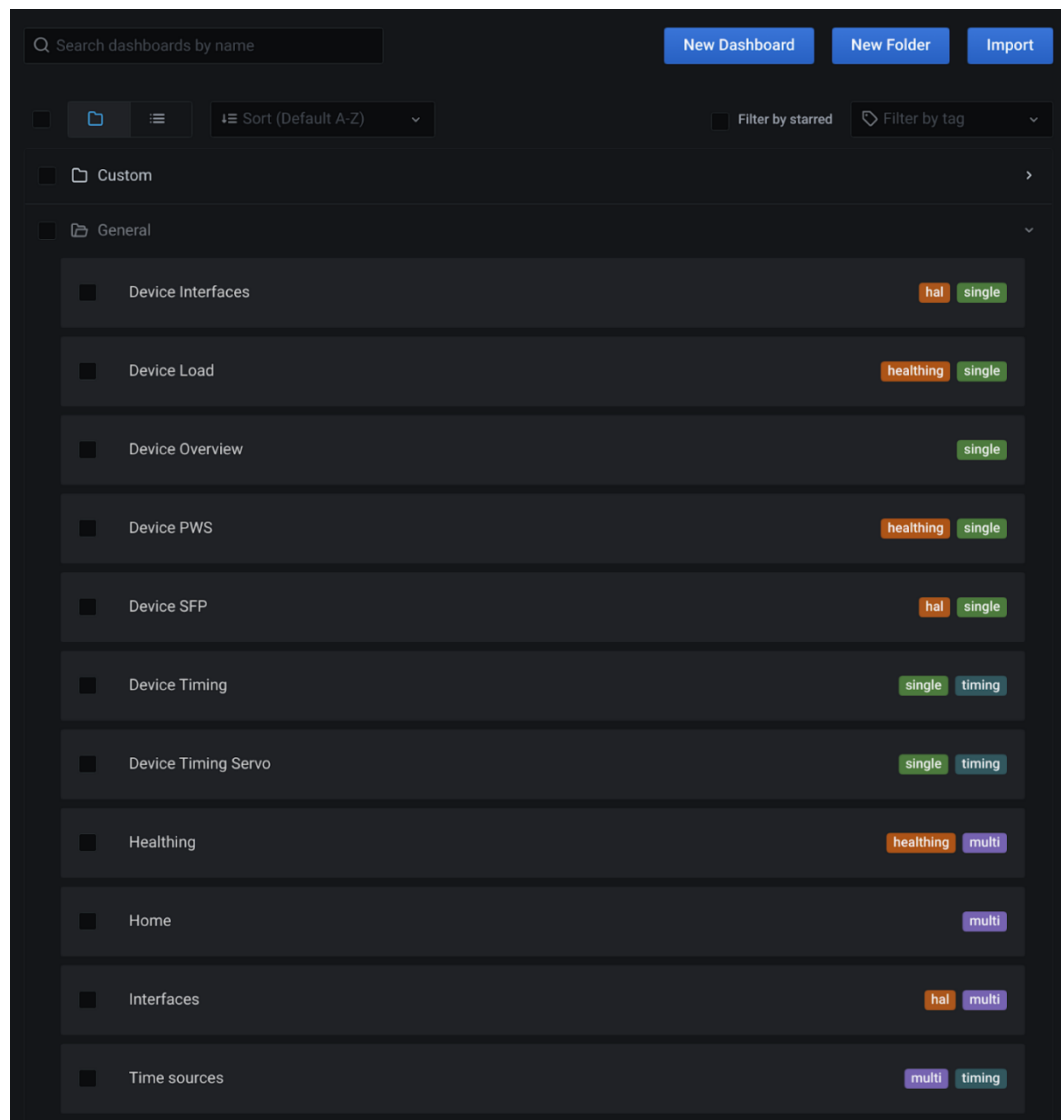
Click in **Configuration > Data Sources**. This tab list all Data Sources configured.



Press in the InfluxDB data source and set the token in its configuration. Then, press the **save & test** button. The correct configuration shows the following message:



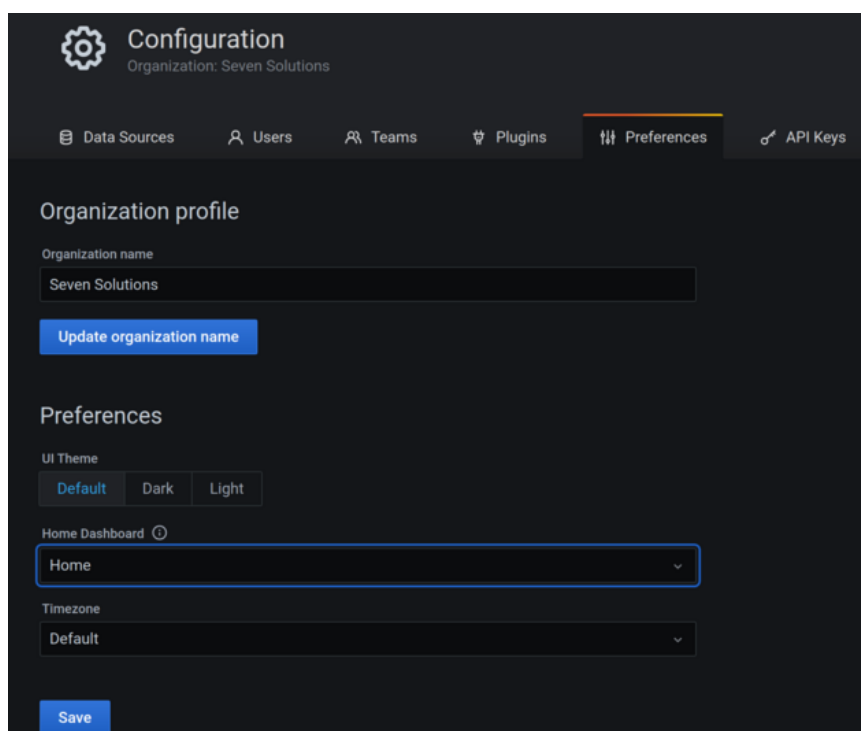
To change the Grafana home page, go to **Dashboard > Manage** tab, which shows all dashboard defined:



Click on dashboard **Home** and press the start button to mark it as favorite:



Set the Home dashboard as the default home page in the **Configuration > Preferences** tab.



The screenshot shows the Configuration > Preferences tab with the following settings:

- Organization profile:**
 - Organization name: Seven Solutions
 - Update organization name button
- Preferences:**
 - UI Theme: Default (selected), Dark, Light
 - Home Dashboard: Home (selected in dropdown)
 - Timezone: Default (selected in dropdown)
 - Save button

To draw the topology, go through the **Network Topology** tab (in the Home dashboard) and click on the **Edit** option. On the right page side, include the mermaid syntax of the specific topology in the **Diagram definition** section

4. Configuration file

Before deploying the system services, the devices to monitor must be added to the configuration file. Open **config.yml** file and add the devices in the agent's section as shown below:

```
agents:
  z16-252:
    ip: 192.168.1.216
    port: 161
    protocol: udp
  z16-251:
    ip: 192.168.1.253
    port: 161
    protocol: udp
```

In case the configuration file needs to be modified after the tool installation, save the changes in the **config.yml** file and run the following command:

```
sudo ./wrz_monitoring.py update --username USER --password PASS
```

4.1. Configuration options

| NAME | DESCRIPTION |
|--------------|--|
| organization | Organization to set in InfluxDB and Grafana services |
| email | System administrator email |

4.2. Device configuration

This section divides the configuration by SNMP information (version, password, user, OIDs, etc). By default, there are three groups of devices: 7s, wrs, and ptpd. All groups must have the following configuration options:

| NAME | DESCRIPTION |
|-------------|--|
| agents | List of agents to monitor. |
| device-type | Type of device (WRZ, WRS, PTP4L, etc.) |
| version | SNMP version |

| | |
|---------------|--|
| auth_password | SNMP password (in case of SNMP v3) |
| auth_protocol | SNMP auth protocol (in case of SNMP v3) |
| community | SNMP community |
| priv_password | SNMP password (in case of SNMP v3) |
| priv_protocol | SNMP protocol (in case of SNMP v3) |
| sec_level | SNMP security level (in case of SNMP v3) |
| sec_name | SNMP user (in case of SNMP v3) |

5. Deployment script (wrz_monitoring.py)

This section explains the different functionalities that the deployment script offers to the user. The *up-services* and the *update* commands have been shown before in sections 3.3 and 4. The following tables describe the different commands and options for the deployment procedure.

The ***wrz_monitoring.py*** script automates the deployment and management procedures. To run this script, docker, docker-compose, Python 3.7 or later, and the python modules *jsonschema* and *toml* must be installed.

wrz_monitoring.py has the following functions:

| COMMAND | DESCRIPTION |
|--------------------|---|
| up-services | Up containers using Docker compose file, configure InfluxDB and generate a token to use in HTTP communications, configure Telegraf, configure Grafana, generate a token for Grafana communications and import dashboards. |
| down-services | Stop and remove Docker containers |
| configure-telegraf | Generate Telegraf configuration files and copy them in Telegraf Docker container. Then, restart the container. |
| test-configuration | Validate configuration file |
| update | Update the configuration if it has changed. Additionally, it updates dashboards and datasource |

Additionally, ***wrz_monitoring.py*** has the following options:

| OPTION | DESCRIPTION |
|---------------------------|--|
| -u, --username | Provide the user that will be set in InfluxDB and Grafana services. |
| -p, --password | Provide the password that will be set in InfluxDB and Grafana services. |
| --second-user | Provide a second user that only has a view permissions |
| --second-user-password | Provide the second user password |
| --new-username | Provide a name to update the username of InfluxDB and Grafana |
| --new-password | Provide a password to update the InfluxDB and Grafana password |
| -c, --config-file | Set a different configuration file. |
| -w, --wrz-monitoring-path | To set the path of the monitoring-7s repository. The repository path must be given, when executing this script from a different directory. |
| -l, --log-file | Print logs in wrz_monitoring.log instead of standard output. |
| -h, --help | Show the help message. |