AMORPH.pro
SMARTUNIFIER

SMARTUNIFIER High Availability Demo Guide

Release 1.4.0

Amorph Systems GmbH

Apr 25, 2022

TABLE OF CONTENTS

CHAPTER

ONE

WHAT IS SMARTUNIFIER HIGH AVAILABILITY DEMONSTRATOR

SMART**UNIFIER** High Availability Demonstrator is a package that allows users to simulate the connection between a production equipment and an Influx DB, using two Instances with a load balancer.

This illustrates that SMART**UNIFIER** is capable to run in a high availability mode, becoming the tool of choice for critical use cases. The package contains all the necessary tools to run a complete communication scenario out of the box.

1.1 Benefits of SMARTUNIFIER High Availability Mode

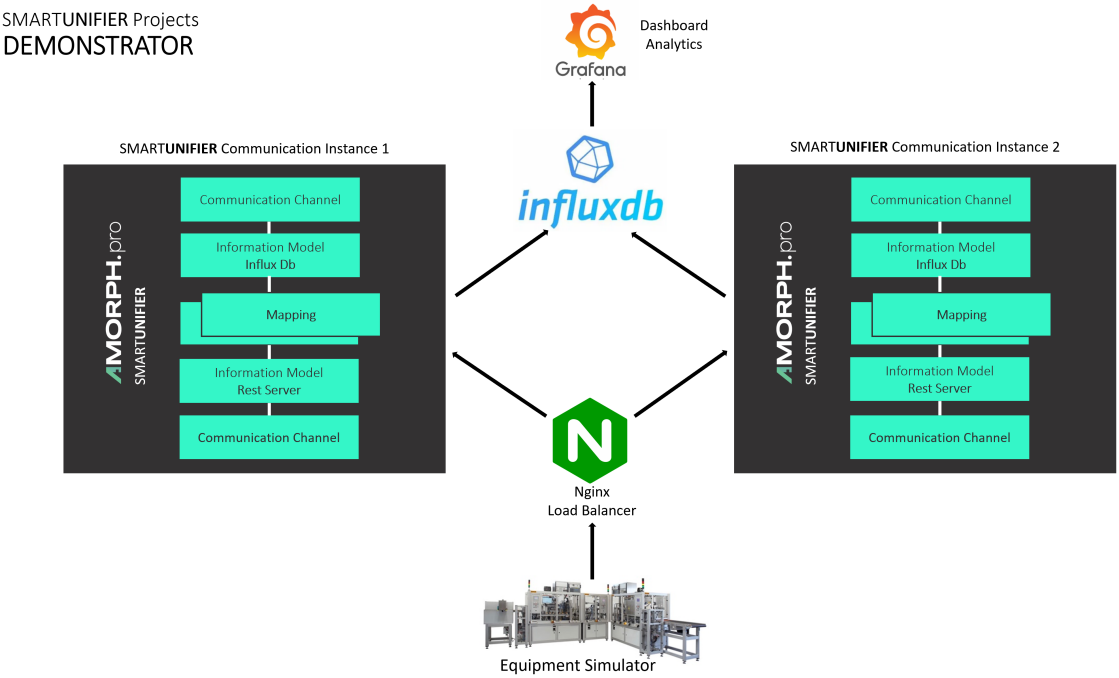
- Downtime reduction
- Performance improvements
- Instance update without downtime

1.2 Components

The Demonstrator package contains the following components:

- **SMARTUNIFIER Manager** – a modern web application to create SMART**UNIFIER** Instances that enable the communication between the Rest Server and the Influx DB.
- **Influx Database** - a Docker image containing the Influx Database where the data coming in from the Rest Server will be stored and used to build visualizations.
- **Grafana** - a Docker image containing a preconfigured Grafana application that has a built-in dashboard displaying the key parameters sent by the Rest Server.
- **Nginx** - a Docker image representing the load balancer for the two SMART**UNIFIER** Instances.
- **Send_Data Script** - a script that generates random values for two parameters: PRESSURE and TEMPERATURE.

1.3 Data Flow Diagram

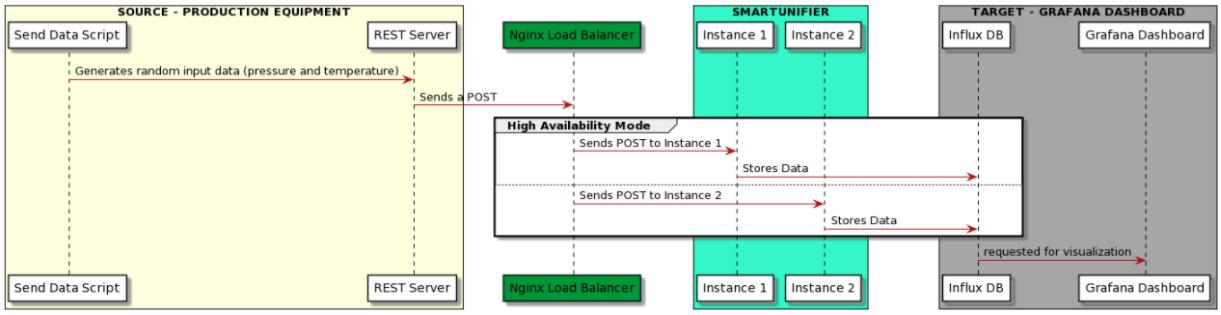


1.4 Demonstrator Artefacts Structure

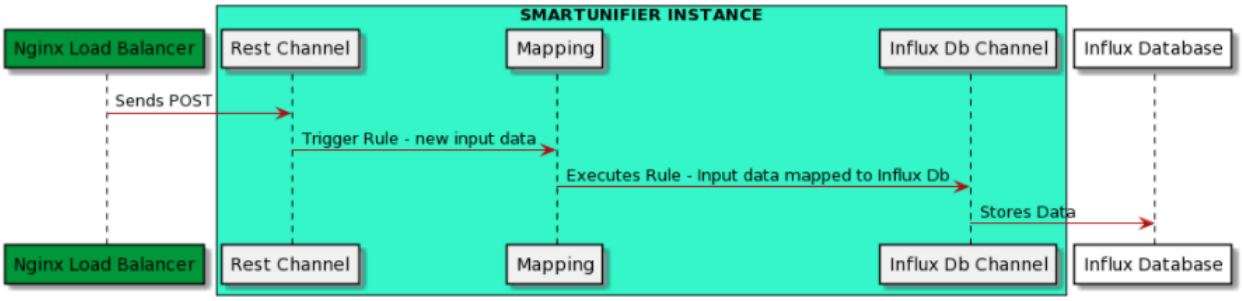
The following table shows the SMART**UNIFIER** artefacts that are used to create this demo:

Type	Name	Description
Information Models	InfluxDb	Stores data from the Rest Server on InfluxDb
	RestData	Represents the Rest Server structure
Communication Channels	InfluxDBChannel	Is used to transmit data from the Rest Server to InfluxDb
	RestChannel	Represent the Rest Server communication protocol
Mappings	RestToInflux	Defines when and how to extract data from the Rest Server and store it on the InfluxDb
Device Types	RestToInfluxDeviceType	Represents the template for the SMARTUNIFIER Instance
Instances	RestToInfluxInstance_1	Represents the configuration for the runnable application
	RestToInfluxInstance_2	

1.5 Simulation Process Flow Diagram



1.6 SMARTUNIFIER Instance Process Flow Diagram



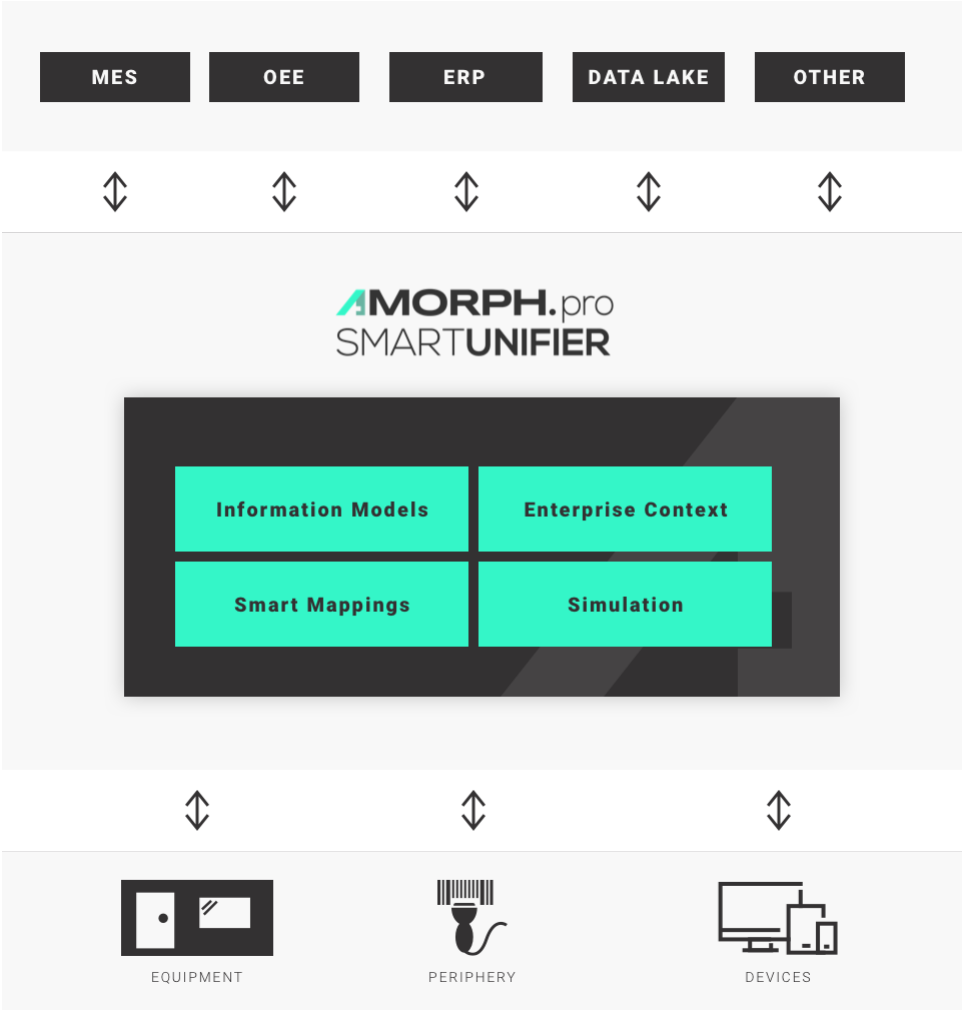
CHAPTER

TWO

WHAT IS SMARTUNIFIER

SMART**UNIFIER** represents a powerful but very easy to use decentralized industrial connectivity platform for interconnecting all industrial devices and IT systems including equipment, peripheral devices, sensors/actors, MES, ERP as well as cloud-based IT systems.

SMART**UNIFIER** is the tool of choice for transforming data into real value and for providing seamless IT interconnectivity within production facilities.



CHAPTER

THREE

WHAT DOES SMARTUNIFIER DO

- SMART**UNIFIER** provides an easy way to collect data from any Data Source and is able to transmit this data to any Data Target.
- Data Sources and Data Targets (commonly referred to as Communication Partners) in this respect may be any piece of equipment, device or IT system, communicating typically via cable or Wi-Fi and using a specific protocol like e.g., OPC-UA, file-based, database, message bus.
- With SMART**UNIFIER** several Communication Partners can be connected simultaneously.
- With SMART**UNIFIER** it is possible to communicate unidirectional or bidirectional to each Communication Partner. I.e., messages and events can be sent and received at the same time.
- SMART**UNIFIER** is able to translate and transform data to any format and protocol that is required by a certain Data Target. This includes different pre-configured protocols and formats, e.g., OPC-UA, file-based, database, message bus, Webservices and many direct PLC connections. In case a certain protocol or format is currently not available it can be easily added to SMART**UNIFIER**.
- By applying so called Information Models, SMART**UNIFIER** enables the same view to data regardless of the protocol or format being used to physically connect an equipment, device or IT system.
- A big advantage of SMART**UNIFIER** is, that in many cases there is no need for coding when providing connectivity between different Communication Partners. SMART**UNIFIER** Mappings enable users to assign data sources to data targets via drag and drop.

CHAPTER

FOUR

SYSTEM REQUIREMENTS

Minimum requirements for running the SMART**UNIFIER** Manager

- **Computer and Processor**: 1 GHz or faster, x86-bit- or x64-bit-processor.
- **Memory**: 512 MB RAM.
- **Hard Disk / SSD**: 1 GB free space.
- **Display PC (Engineering, Dashboard)**: 1280 x 1024 Resolution.
- **Mobile Devices (Dashboard)**: Apple iPhone 6 or higher, Android.
- **Operating System**: Windows 10, Windows 8, Windows 7, Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Linux, MacOS - For an optimal user experience always use the newest version of the operating system.
- **Browser**: Latest version of Chrome, Microsoft Edge/Internet Explorer, Firefox.
- **Other**: Latest version of Docker Daemon (including Docker-compose). For more details follow the on-screen instructions from https://docs.docker.com/compose/install/.

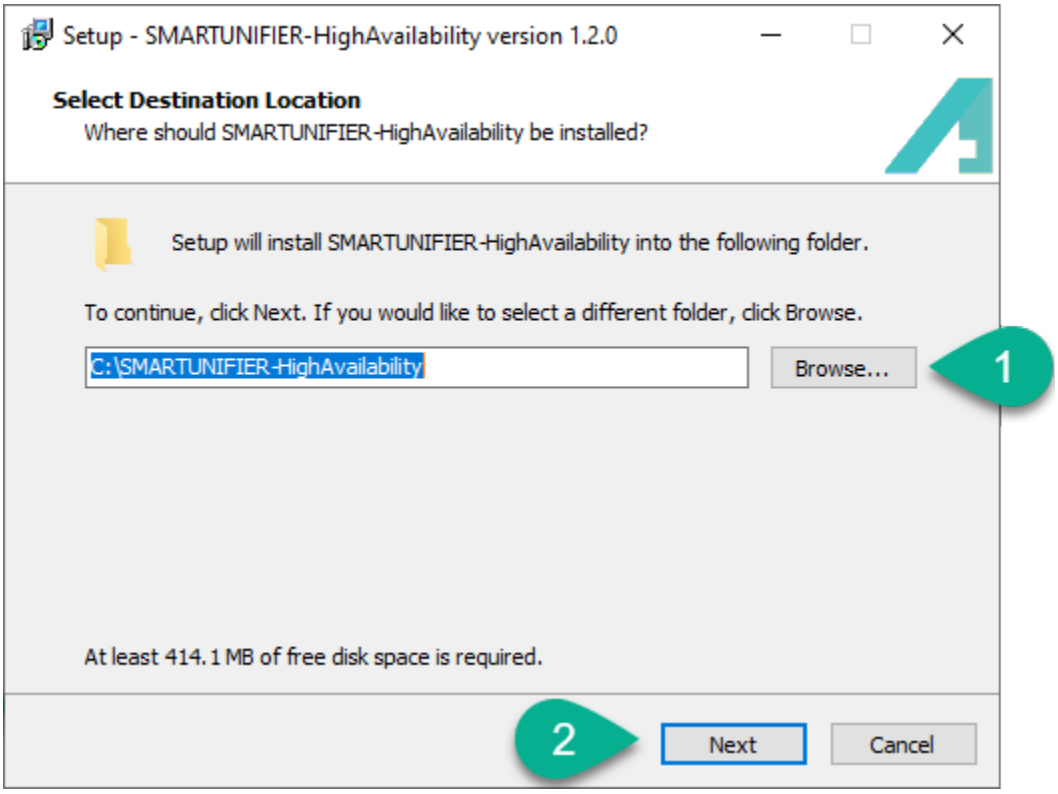
CHAPTER

FIVE

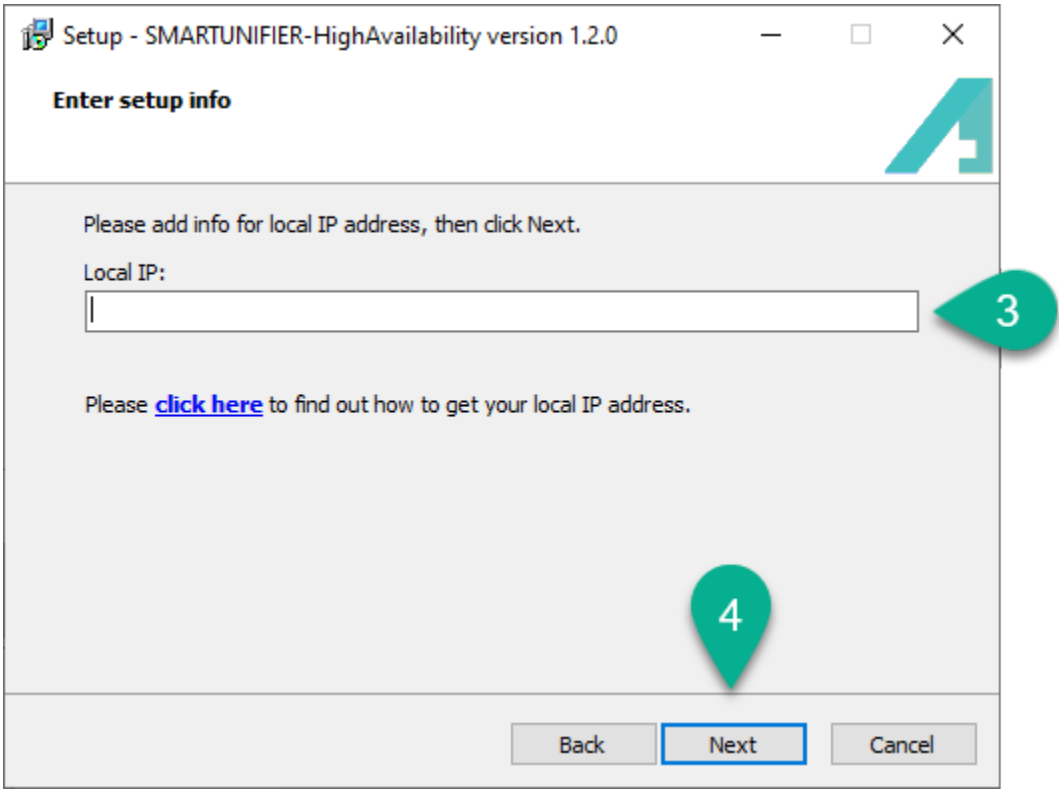
WINDOWS

5.1 Step 1 - Install the SMARTUNIFIER and the Docker Components:

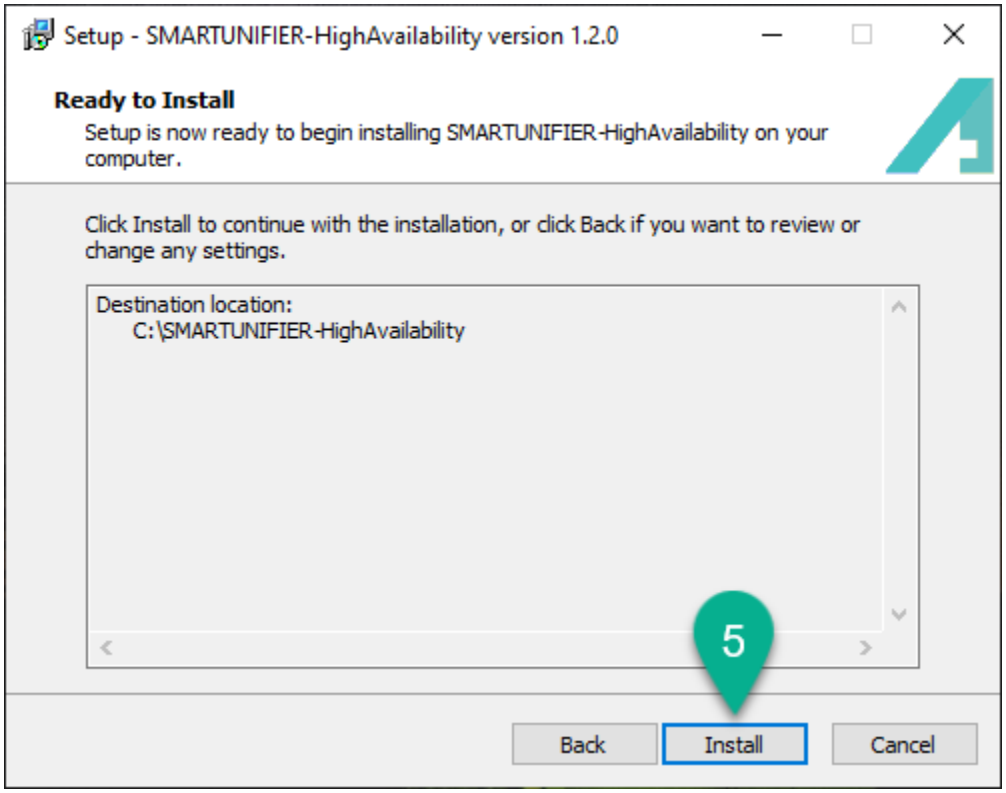
- Move the installation package to a suitable location.
- Open “Docker Desktop” application.
- Run the **.exe** file to start the setup.
- Click on the “Browse” button **(1)** to change the location of the installation.
- Select the “Next” button **(2)** to continue.



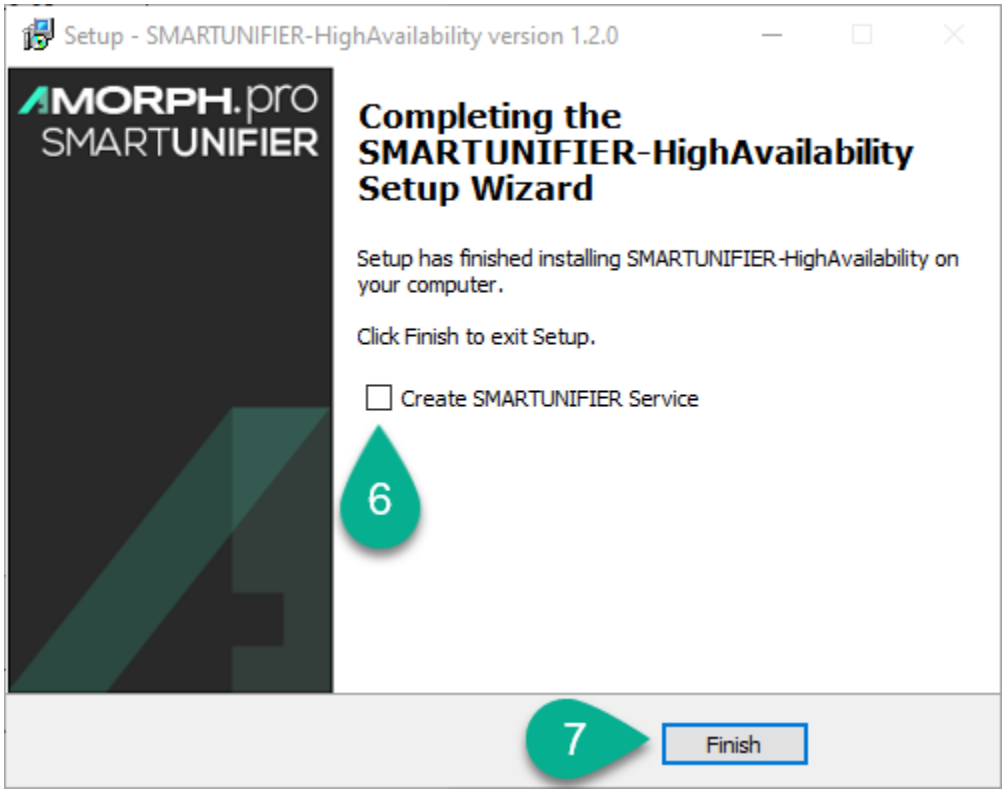
- Enter the local IP address **(3)**. Select the “Next” button **(4)** to continue.



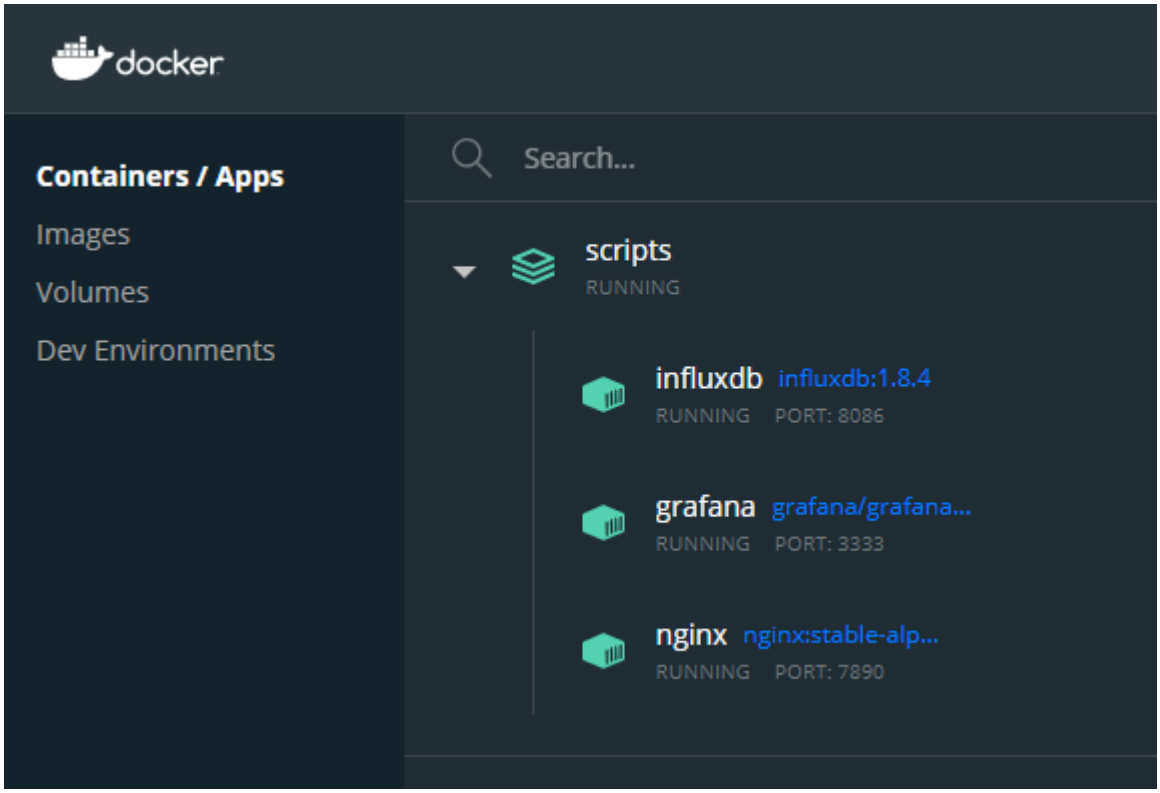
- Click on the “Install” button **(5)** to start the installation.



- Check the box **(6)** to create SMART**UNIFIER** Service (optional).



- Select the “Finish” button **(7)** to finallize installing the SMART**UNIFIER**. The console will open and the extraction of the Docker components starts.
- After the console is closed, the Docker components are installed in a few seconds:
  - influxdb
  - grafana
  - nginx



5.2 Deploy Components Manually

The SMART**UNIFIER** Service can also be created manually. From the **SMARTUNIFIER-HighAvailability** folder (install_location/SMARTUNIFIER-HighAvailability) run the console (CMD) as administrator and execute:

```
create_service.bat
```

If a Docker container is deleted, it can be redeployed. From the **scripts** folder (install_location/SMARTUNIFIER-HighAvailability/scripts) open the console (CMD) and execute:

- to deploy InfluxDb container:

```
deploy.bat influxdb
```

- to deploy Grafana container:

```
deploy.bat grafana
```

- to deploy Nginx container:

```
deploy.bat nginx
```

- to deploy all components:

```
deploy.bat
```

5.3 Uninstalling a container

To remove the SMART**UNIFIER** Service, from the **SMARTUNIFIER-HighAvailability** folder (install_location/SMARTUNIFIER-HighAvailability) run the console (CMD) as administrator and execute:

```
remove_service.bat
```

To undeploy a docker container, from the **scripts** folder (install_location/SMARTUNIFIER-HighAvailability/scripts) open the console(CMD) and execute:

- to remove InfluxDb container:

```
cleanup.bat influxdb
```

- to remove Grafana container:

```
cleanup.bat grafana
```

- to remove Nginx container:

```
cleanup.bat nginx
```

- to remove all containers:

```
cleanup.bat
```

Note: If the user needs to connect with a different local IP, first undeploy and redeploy the docker containers and in the end input the new local IP.

5.4 Uninstalling All

Follow the steps below to uninstall all the Demonstrator components (folders, files, Docker components):

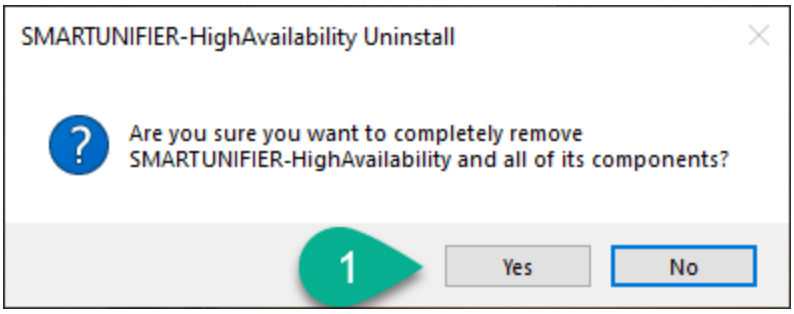
- Make sure the Demonstrator Instance is **NOT** running.
- Make sure the "Docker Desktop" application **IS** running.
- Make sure the **SMARTUNIFIER-HighAvailability** folder (install_location/SMARTUNIFIER-HighAvailability) is **NOT** opened by any application.
- There are two options to uninstall the Demonstrator:

1. Using Windows System settings section:

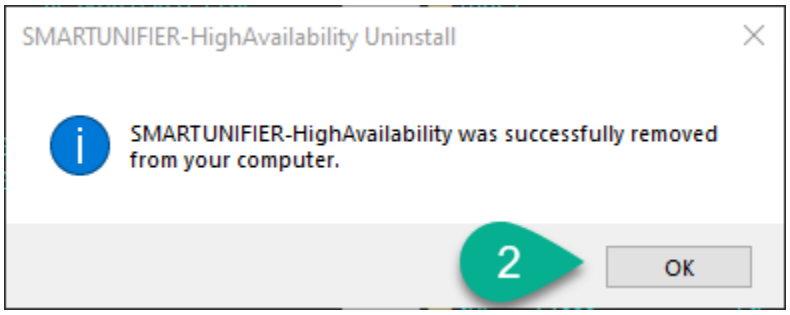
- Go to "Add or remove programs" and uninstall SMARTUNIFIER-HighAvailability application.

2. Run the **unins000.exe** file from the "SMARTUNIFIER-HighAvailability" folder.

- Select the "Yes" button **(1)** to confirm.



- Select the “OK” button **(2)** to finish.



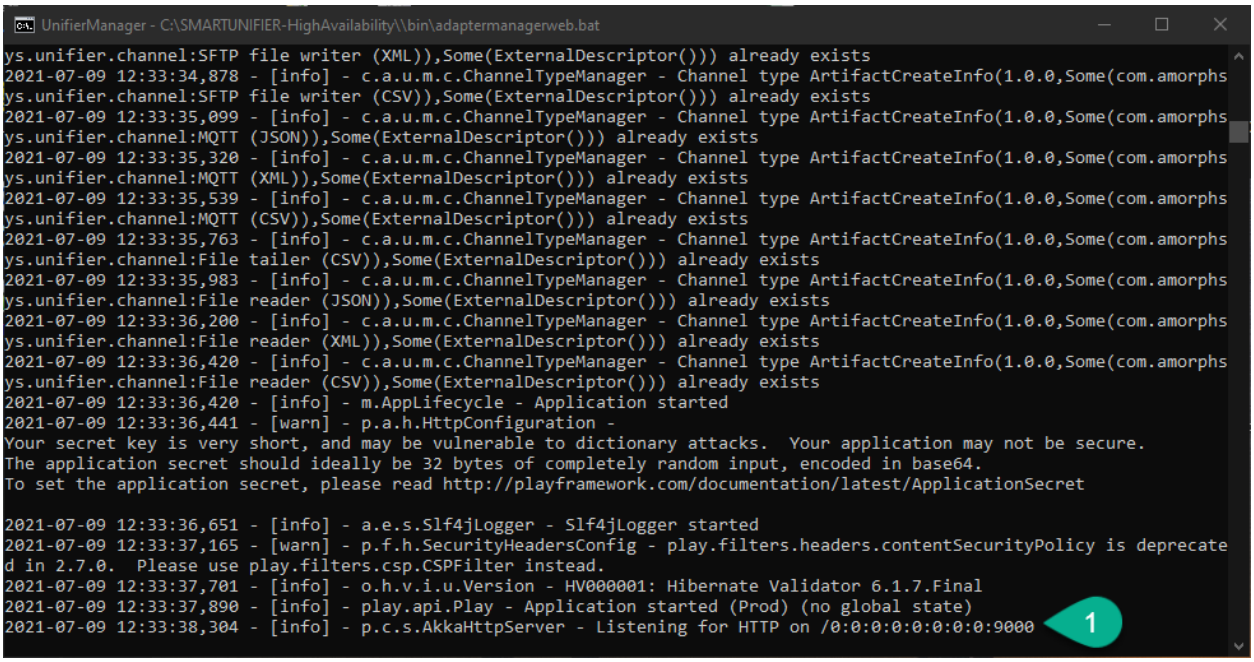
All the Demonstrator components are removed.

5.5 Step 2 - Running the SMARTUNIFIER

SMART**UNIFIER** can be started as an application or as a service.

5.6 Running SMARTUNIFIER as an Application

- If SMART**UNIFIER** was not installed as a service, execute the **UnifierManager.bat** script located in the installation folder. Afterwards the SMART**UNIFIER** Manager Console appears on the screen.

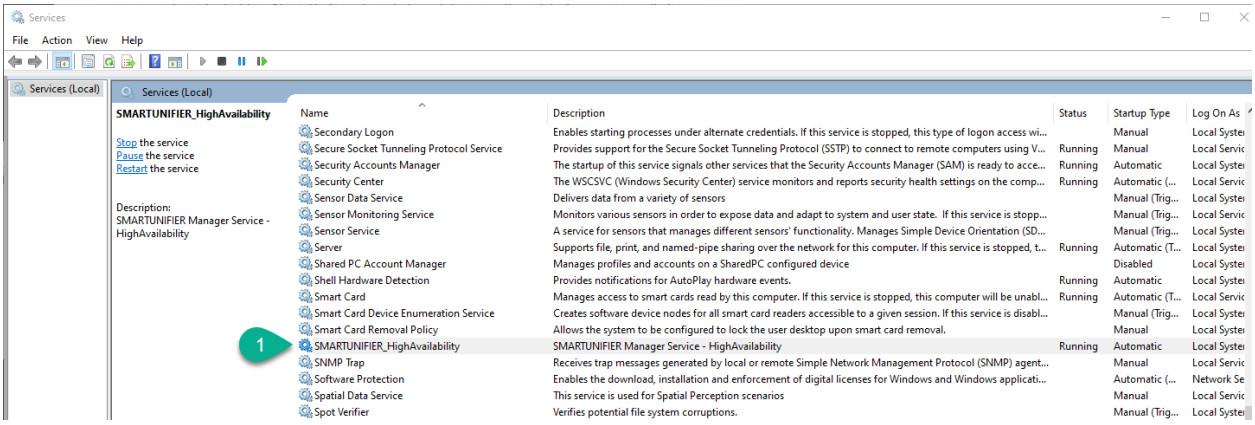


- After successfully starting up the SMART**UNIFIER** Manager, it can be accessed by opening an Internet Browser (e.g., Chrome or Firefox) and navigating to http://localhost:9000. Use the administrator credentials to login:
- Username: **admin**
- Password: **admin**

Note: The console is for information purposes only. It can be moved to any suitable location on your screen or it can be hidden. Nevertheless, do not close it, because the related processes will also be terminated.

5.7 Running SMARTUNIFIER as a Service

- If SMART**UNIFIER** was installed as a service, the service is already running.
- To check open “Services” in Windows (press the “Windows” button and type “services”) and search “SMART**UNIFIER**" from the list **(1)**.



- Open an Internet Browser (e.g., Chrome or Firefox) and navigating to http://localhost:9000. Use the administrator credentials to login:
- Username: **admin**
- Password: **admin**

CHAPTER

SIX

LINUX

6.1 Step 1 - Install SMARTUNIFIER and Docker Components:

- Move the installation package to a suitable location. Make sure the path to the directory does not include any white spaces!
- Extract the **.tar.gz**-archive.

```
tar -xvzf SMARTUNIFIER-Manager-linux-x64.tar.gz
```

- To change the deploy file properties and to convert it's format to Unix, open the terminal from the **scripts** folder (install_location/SMARTUNIFIER-HighAvailability/scripts) and execute the following commands:

```
chmod 775 deploy.sh
```

```
dos2unix *.sh
```

- To deploy the Docker components execute the following commands (terminal opened from the **scripts** folder):

```
bash deploy.sh demonstrator
```

- During the Docker components deploy, the input of the local IP address is required. To check how to get the local IP go to https://support.microsoft.com/en-us/windows/find-your-ip-address-f21a9bbc-c582-55cd-35e0-73431160a1b9
- After entering the local IP address, the Docker components are deployed in a few seconds:
  - influxdb
  - grafana
  - nginx

6.2 Step 2 - Running SMARTUNIFIER:

- Start the SMART**UNIFIER** Manager by executing in a terminal the following commands:

```
chmod +x UnifierManager.sh
```

```
./UnifierManager.sh
```

- After successfully starting the SMART**UNIFIER** Manager, it can be accessed by opening an Internet Browser (e.g., Chrome or Firefox) and navigating to http://localhost:9000. Use the administrator credentials to login:
- Username: **admin**
- Password: **admin**

Note: The console is for information purposes only. It can be moved to any suitable location on your screen or it can be hidden. Nevertheless, do not close it, because the related processes will also be terminated.

CHAPTER

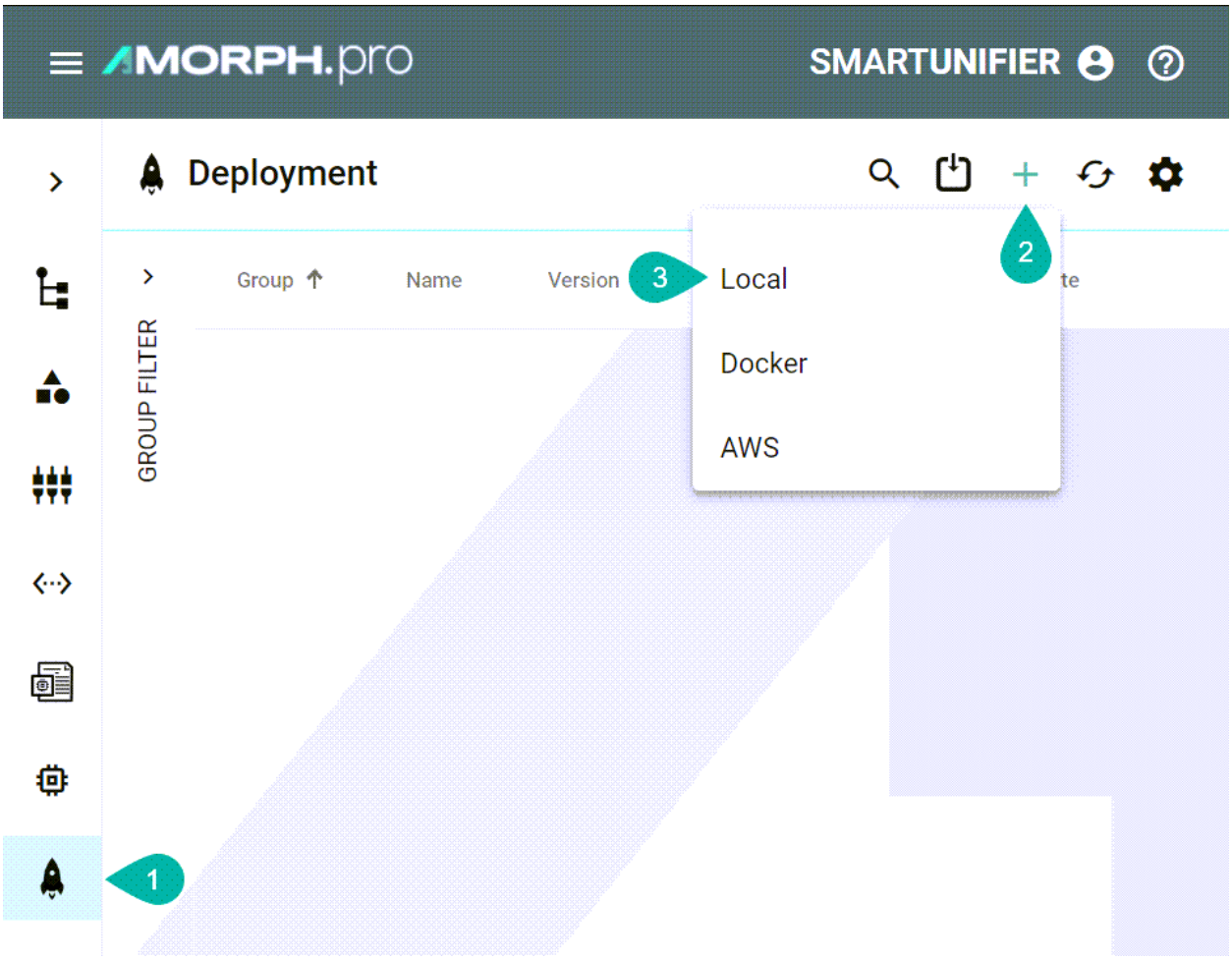
SEVEN

RUN THE INSTANCE

The communication between the Rest Server and the Influx DB is facilitated by two SMART**UNIFIER** Instances.

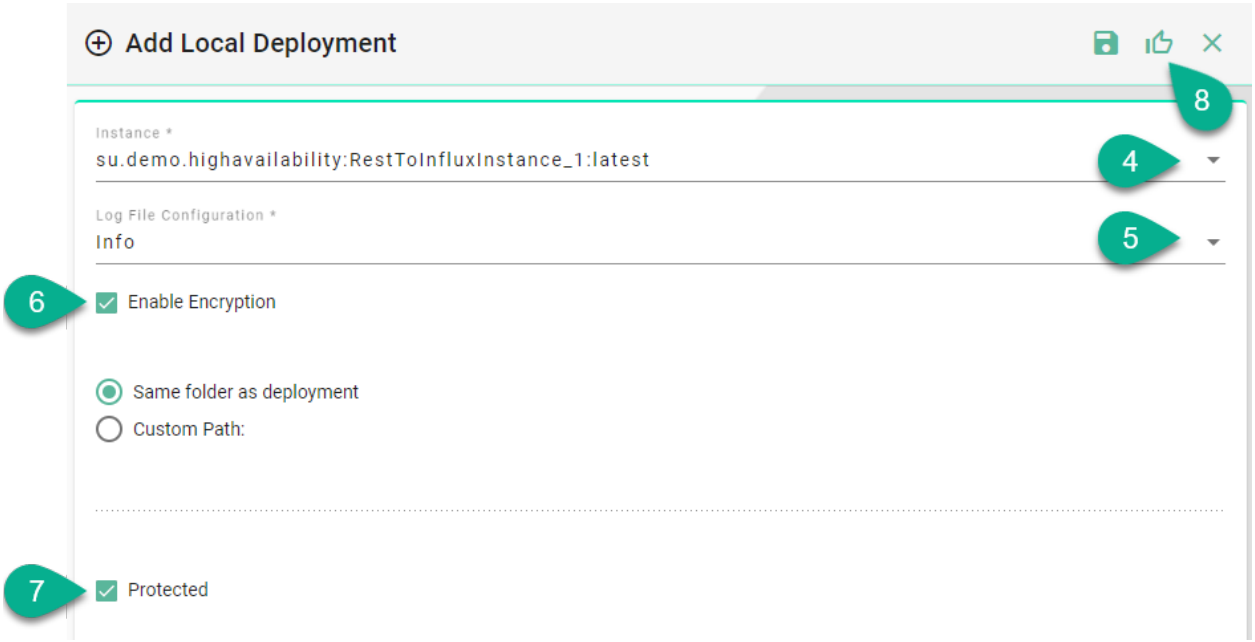
First add each Instance to a local deployment:

- Open “Deployments” section **(1)**.
- Click on the “Add” button **(2)**.
- Select the “Local” option **(3)**.



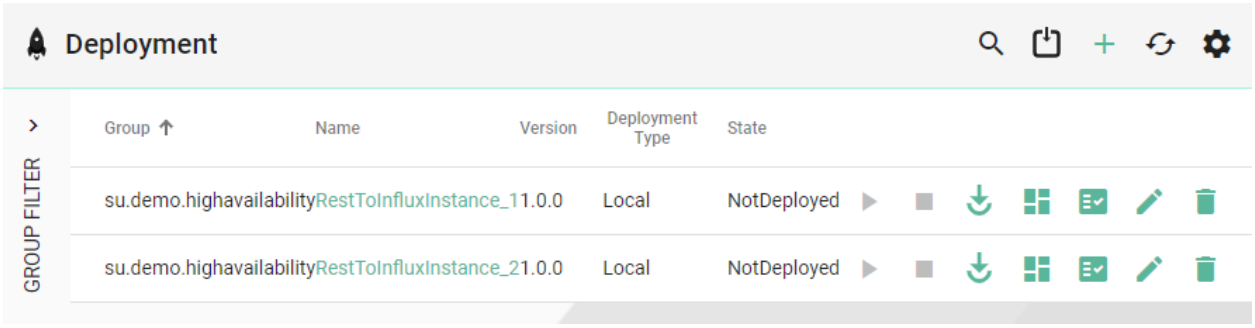
- Select the “Instance” from the dropdown **(4)**.

- Select “Info” from the “Log File Configuration” dropdown **(5)**.
- Check the box for “Enable Encryption” **(6)** to have all the credentials encrypted in the configuration files.
- Check the box for “Protected” **(7)** and a confirmation will be required for each Instance action (e.g., deploy, undeploy, start, stop).



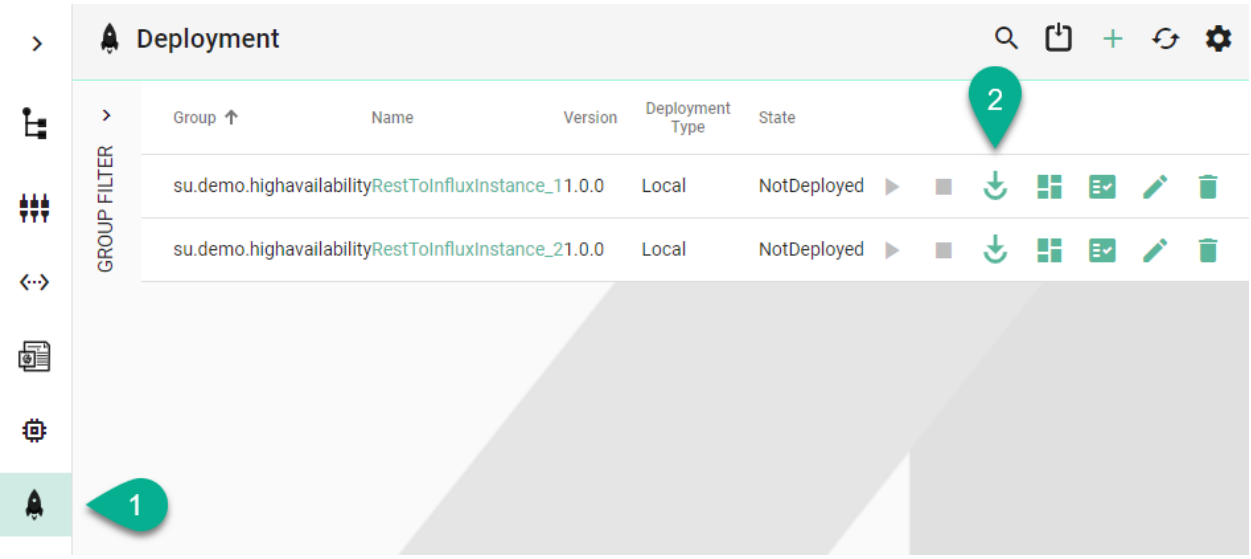
- Click the “Save and Close” button **(8)**.

The first Instance is added, follow the steps above to add the second Instance.

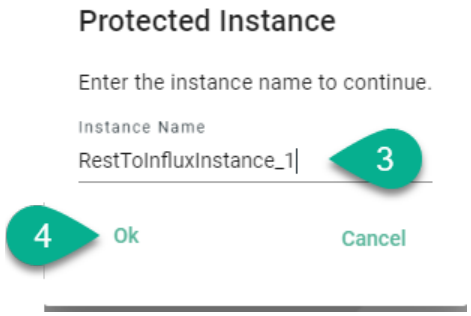


Then run each Instance:

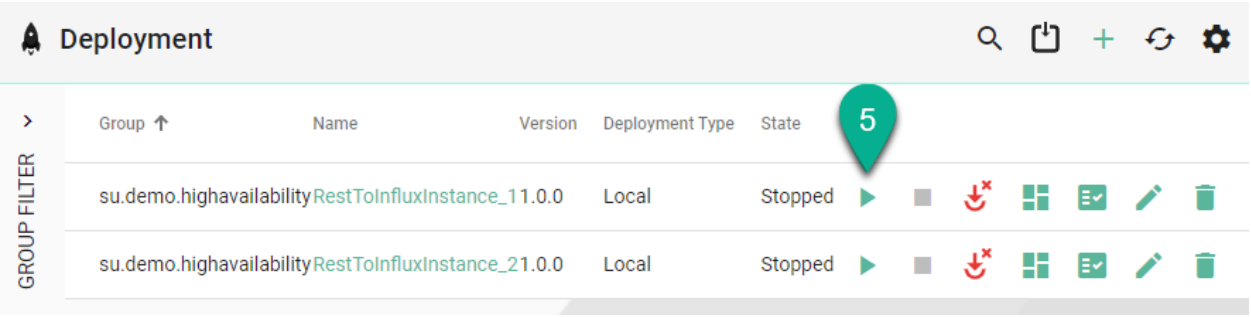
- Open the “Deployments” section **(1)**.
- Click the “Deploy” button **(2)** for each Instance.



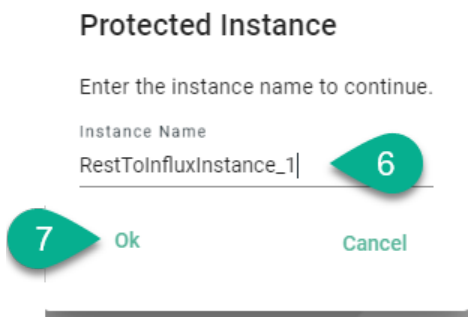
- Type the Instance name **(3)** to confirm the action and click the “Ok” button **(4)**.



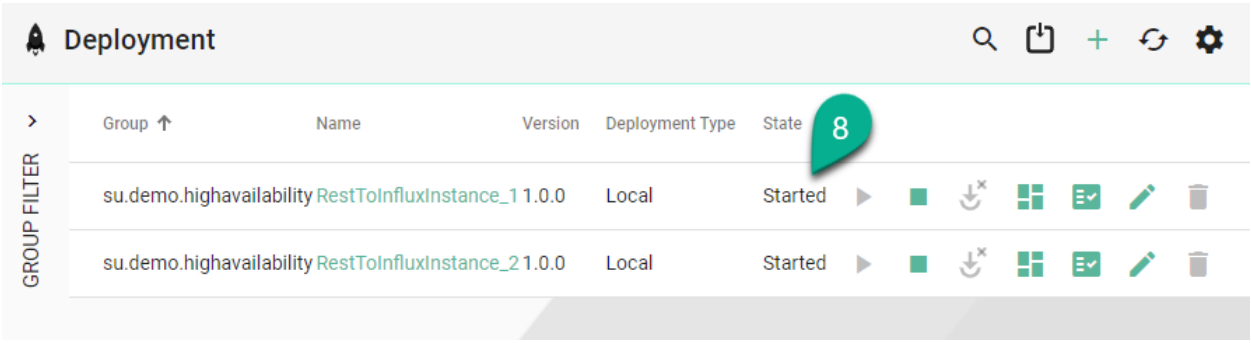
- Click the “Start” button **(5)** to run each Instance.



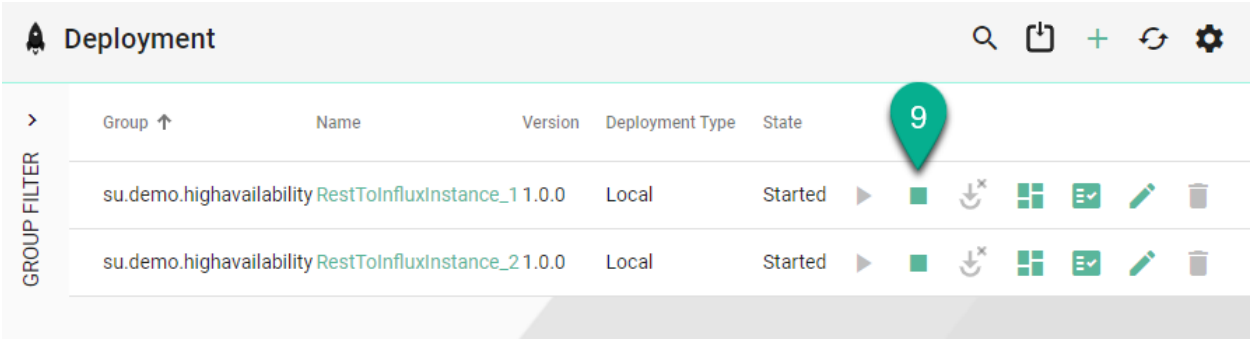
- Type the Instance name **(6)** to confirm the action and click the “Ok” button **(7)**.



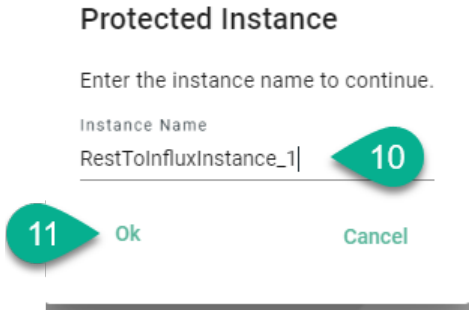
- The Instances are running **(8)**.



To stop an Instance click on the "Stop" button **(9)** as seen below.



- Type the Instance name **(10)** to confirm the action and click the "Ok" button **(11)**.



CHAPTER

EIGHT

GRAFANA

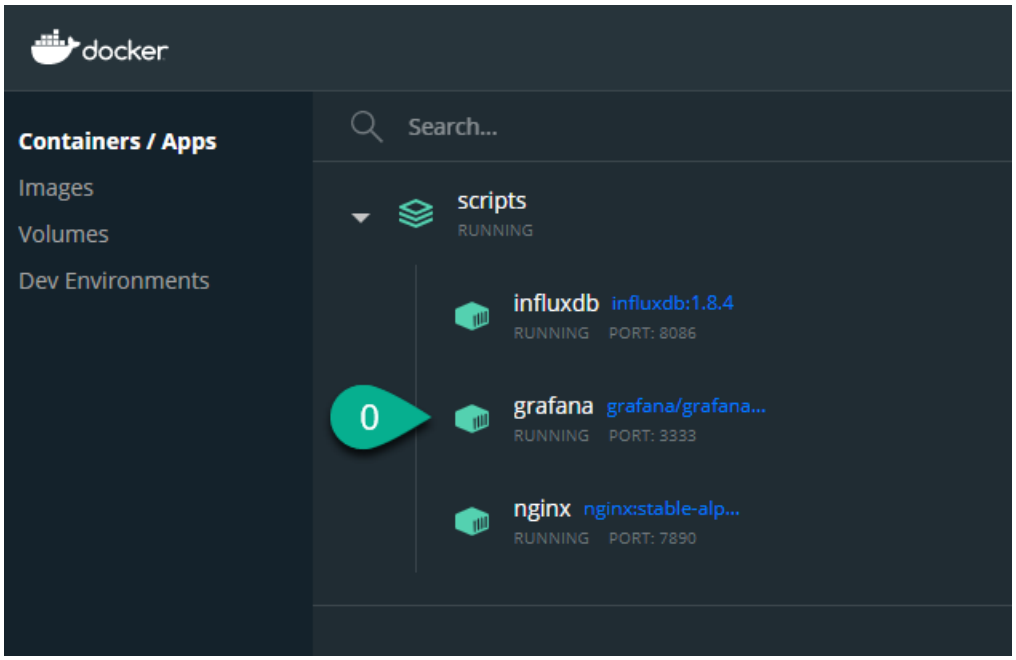
Grafana is an open source analytics and interactive visualization web application. It provides charts, graphs and alerts for the web when connected to the supported data sources.

In the current demonstrator, Grafana is used to display the key parameters sent by the Rest Server.

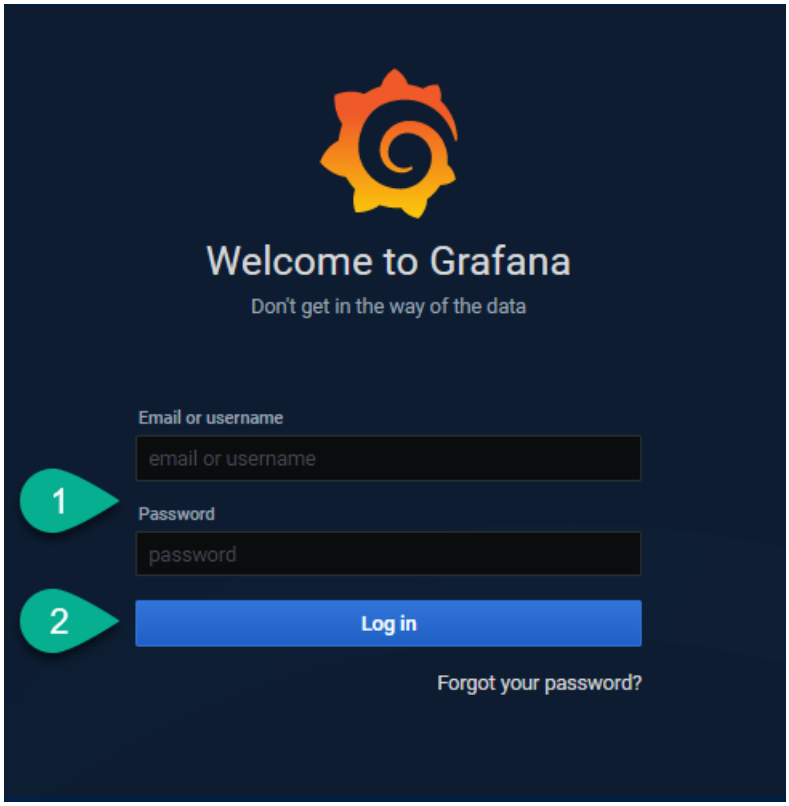
Access Grafana

Follow the steps bellow to access Grafana:

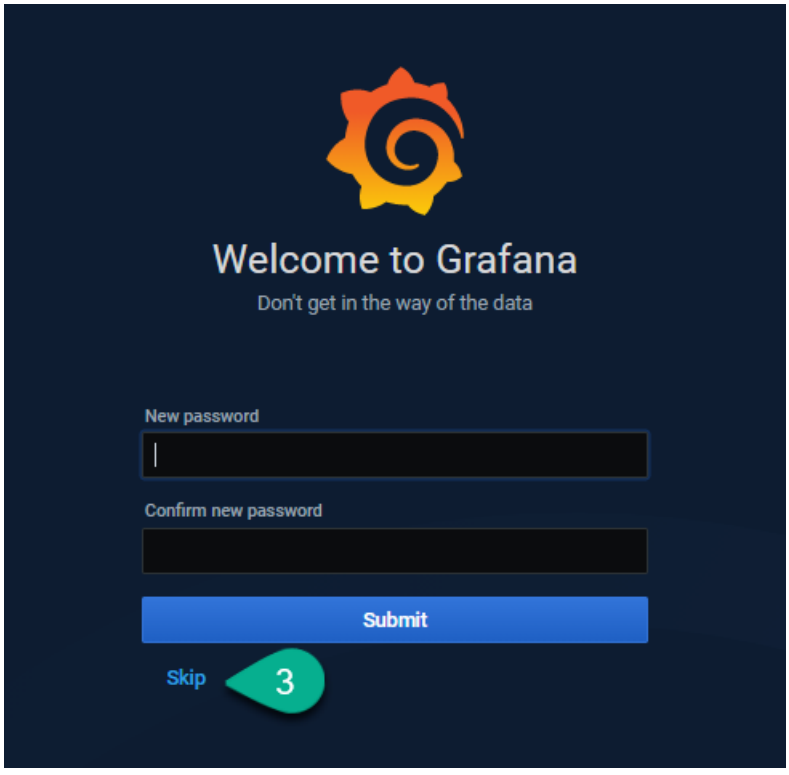
- Grafana Docker container must be running.



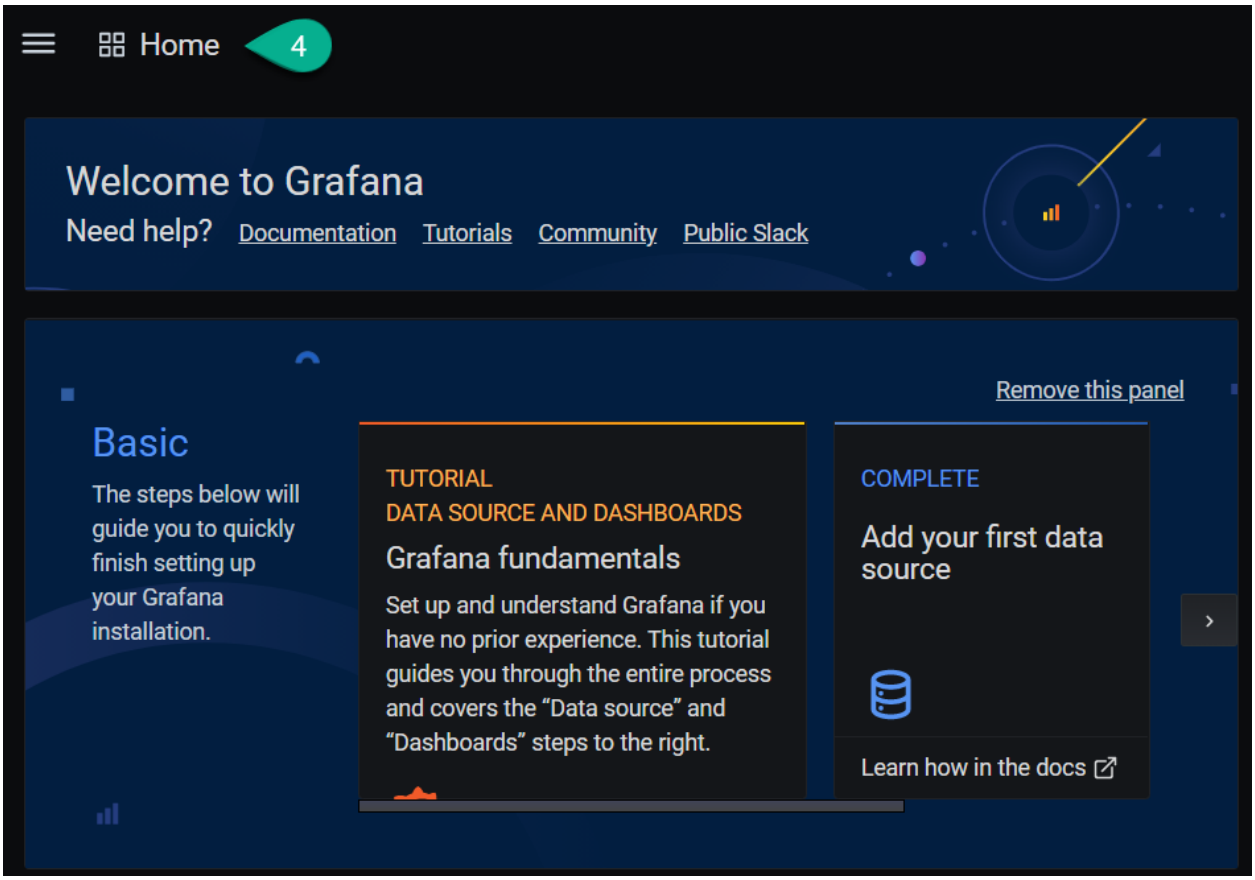
- Open an Internet Browser (e.g., Chrome or Firefox) and navigate to http://localhost:3333/. For "Username" and "Password" use **admin (1)**. Select the "Log in" button **(2)**.



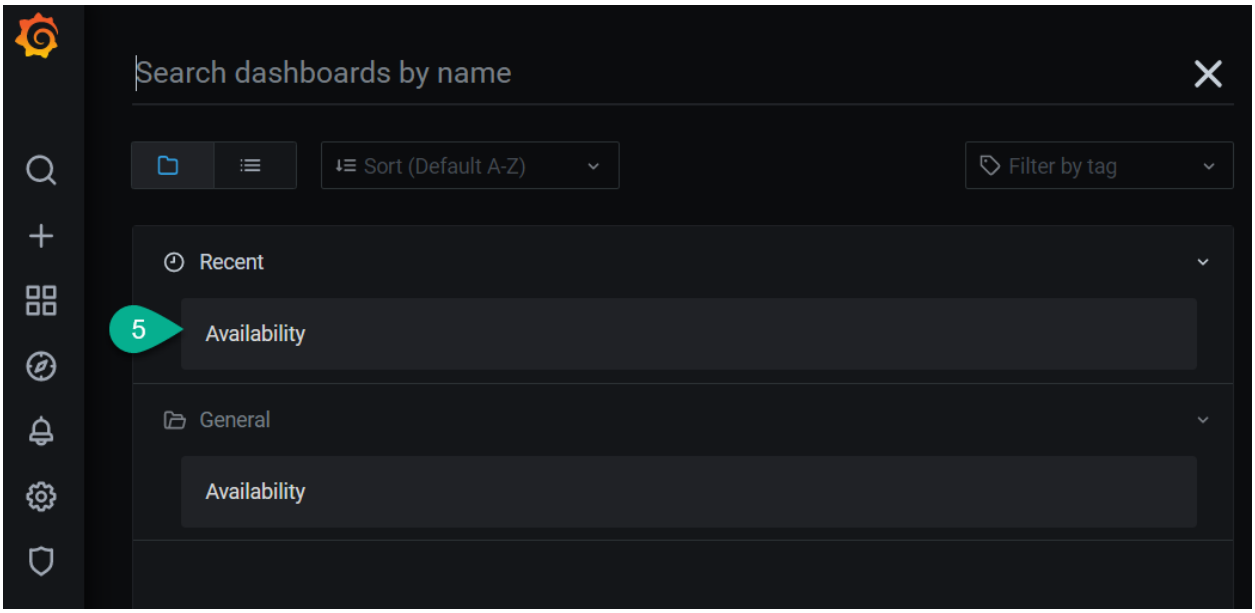
- Change the “Password” or just select the “Skip” button **(3)**.



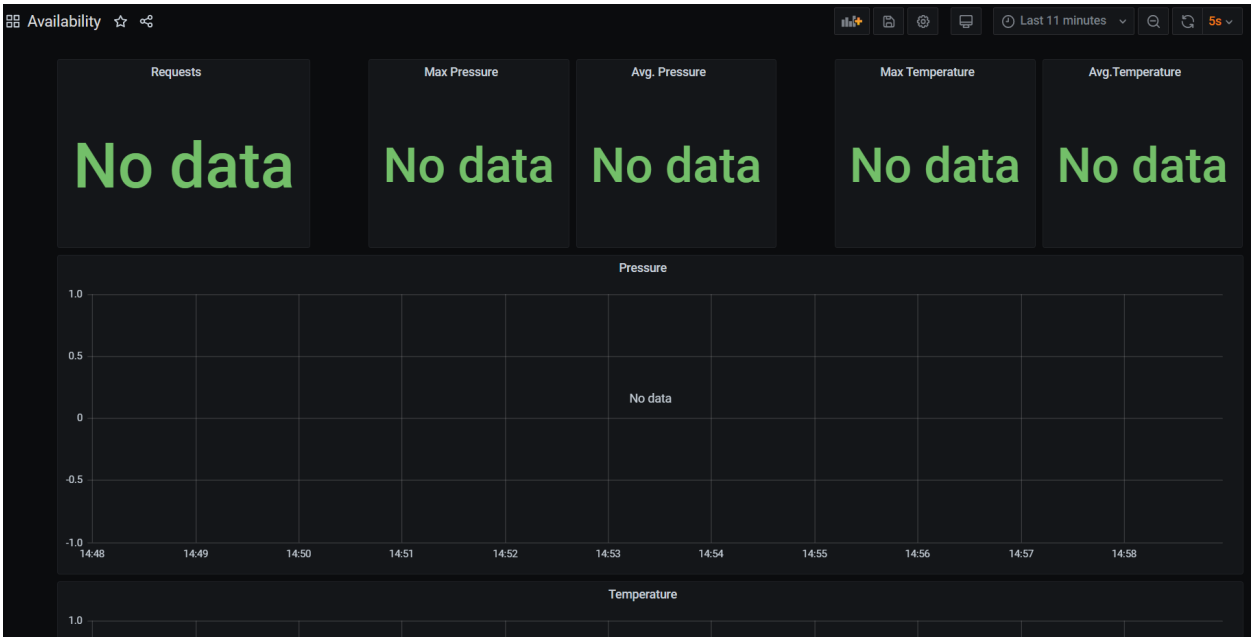
- Select the “Home” button **(4)**.



- Select "Availability" options **(5)**.



- The Grafana Availability Dashboard is visible.


Availability
Last 11 minutes
5s
Requests
No data
Max Pressure
No data
Avg. Pressure
No data
Max Temperature
No data
Avg.Temperature
No data
Pressure
1.0
0.5
0
-0.5
-1.0
No data
14:48
14:49
14:50
14:51
14:52
14:53
14:54
14:55
14:56
14:57
14:58
Temperature
1.0

CHAPTER

NINE

START THE SIMULATION

The Demonstrator use cases the seamless connection between a production equipment and an InfluxDb, connection facilitated by the SMART**UNIFIER**. Two Instances are configured with a load balancer to provide the connection in high availability mode.

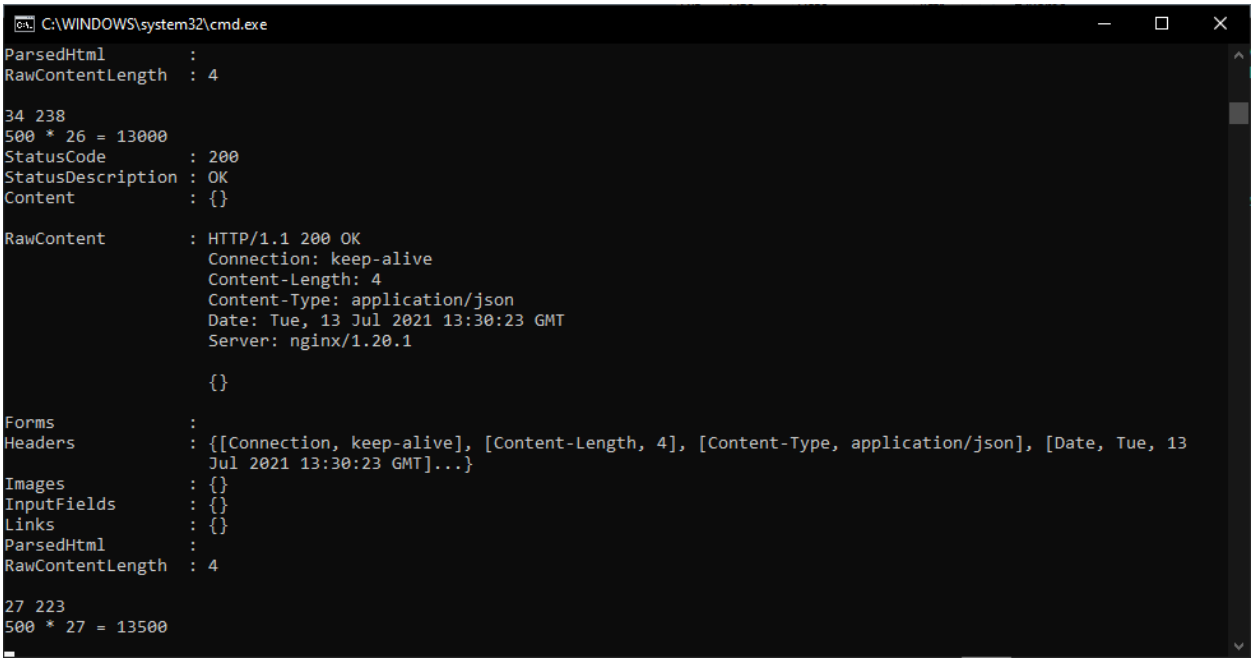
To run the simulation, two steps are required:

- Generate the input data - random values.
- Test the High Availability - by stopping one of the Instances.

9.1 Generate the Input Data

To simulate the data provided by a production equipment, the input data is generated by the **send_data** script, located in the **scripts** folder (install_location/SMARTUNIFIER-HighAvailability/scripts).

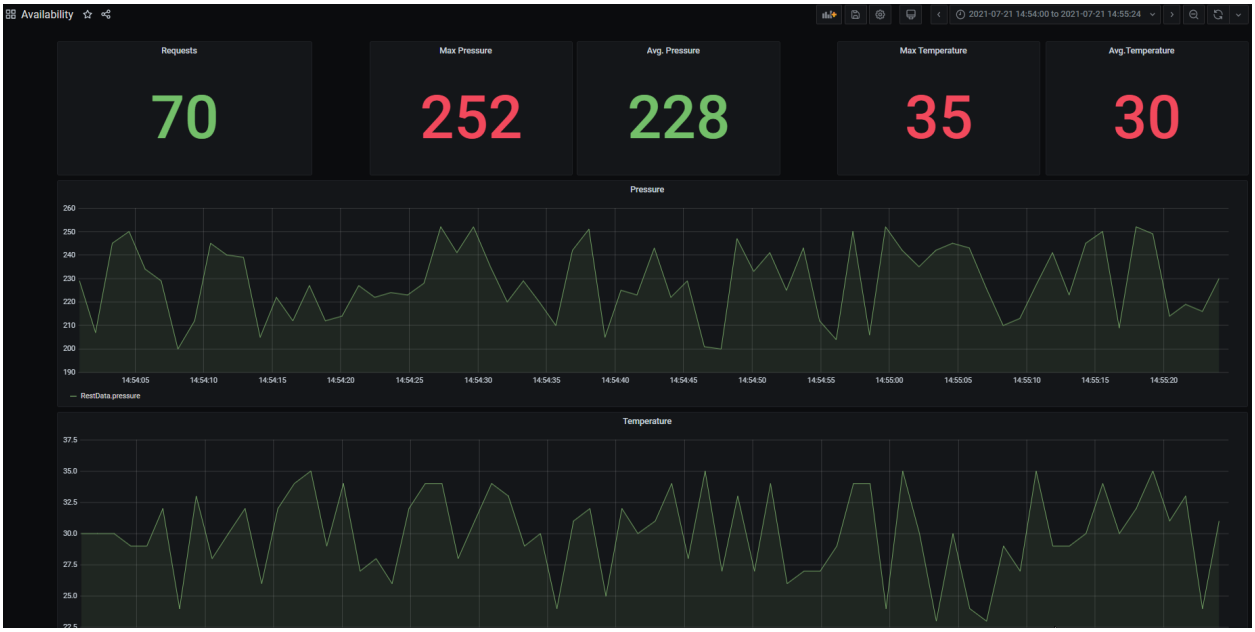
Run the script to generate random values for two parameters: pressure and temperature.



Note: For Linux, to generate the input data run the below command:

```
bash send_data.sh
```

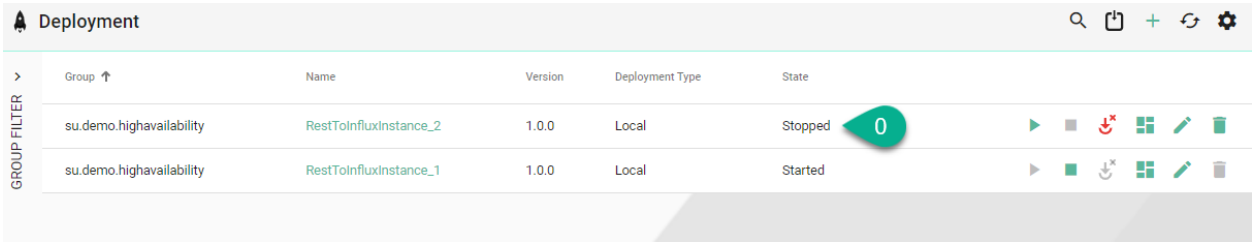
The Rest Server picks up the new generated values and sends a post that is stored in the Influx database, using the SMART**UNIFIER**. The overall process can be viewed on the Grafana dashboard.



9.2 Test the High Availability

The SMART**UNIFIER** uses two Instances configured with a load balancer (Nginx). This means that if one of the two Instances is stopped, the load is handled by the other working Instance.

Stop one of the two Instances.



When one Instance is stopped, the communication fails over to the other running Instance and all the data is sent to the receiver, as seen below.

The script stops to generate values when it reaches the cycle limit, configured at 500.

```
C:\WINDOWS\system32\cmd.exe
RawContentLength  : 4

23 251
500 * 500 = 250000
StatusCode        : 200
StatusDescription : OK
Content           : {}

RawContent        : HTTP/1.1 200 OK
                    Connection: keep-alive
                    Content-Length: 4
                    Content-Type: application/json
                    Date: Tue, 13 Jul 2021 13:40:36 GMT
                    Server: nginx/1.20.1

                    {}

Forms             :
Headers           : {[Connection, keep-alive], [Content-Length, 4], [Content-Type, application/json], [Date, Tue, 13
                    Jul 2021 13:40:36 GMT]...}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        :
RawContentLength  : 4

33 237


PS C:\SMARTUNIFIER-HighAvailability\scripts>
```


Availability
2021-07-21 14:53:08 to a few seconds ago
Requests
500
Max Pressure
252
Avg. Pressure
227
Max Temperature
35
Avg.Temperature
29
Pressure
260
250
240
230
220
210
200
190
14:53:30
14:54:00
14:54:30
14:55:00
14:55:30
14:56:00
14:56:30
14:57:00
14:57:30
14:58:00
14:58:30
14:59:00
14:59:30
15:00:00
15:00:30
15:01:00
15:01:30
15:02:00
15:02:30
15:03:00
15:03:30
15:04:00
15:04:30
15:05:00
15:05:30
15:06:00
RestData.pressure
Temperature
37.5
35.0
32.5
30.0
27.5
25.0
22.5

CHAPTER

TEN

INSTANCE SETUP

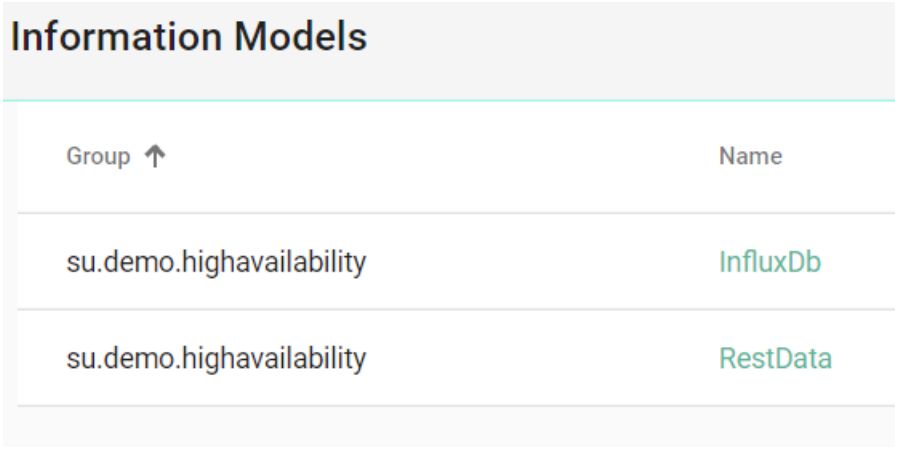
A SMART**UNIFIER** Instance is a dynamically created application that can be deployed to any suitable IT resource (e.g., Equipment PC, Server, Cloud), and which provides the connectivity functionality configured. Therefore, a SMART**UNIFIER** Instance uses one or multiple Mappings, selected Communication Channels and Information Models.

10.1 Information Models

Within the SMART**UNIFIER** an Information Model describes the communication related data that is available for a device or IT system. One device or one IT system therefore is represented by one Information Model.

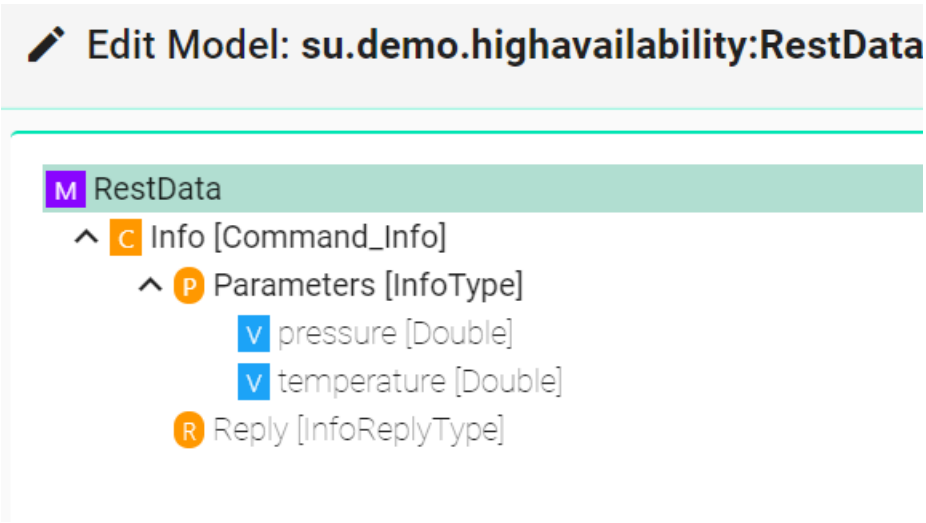
The Information Model perspective lists the information models currently configured within the SMART**UNIFIER** Manager:

1. **RestData**
2. **InfluxDb**



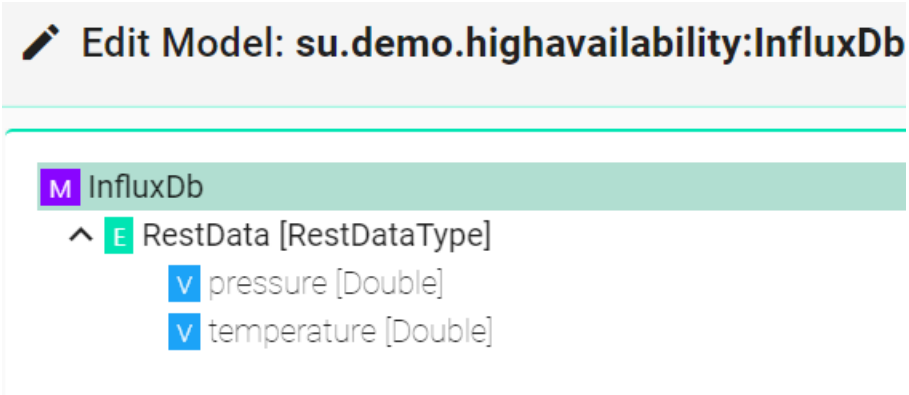
1. **RestData**

The RestData Information Model represents the Rest Server structure. It's structure is simple, containing two parameters, as seen below.



2. **InfluxDb**

The InfluxDb Information Model stores data from the Rest Server. The data provided by the Rest Server is stored using the SMART**UNIFIER** in the Influx database and the overall process can be viewed on the Grafana dashboard. As seen below, the Model data structure matches the source data structure (Rest Server).



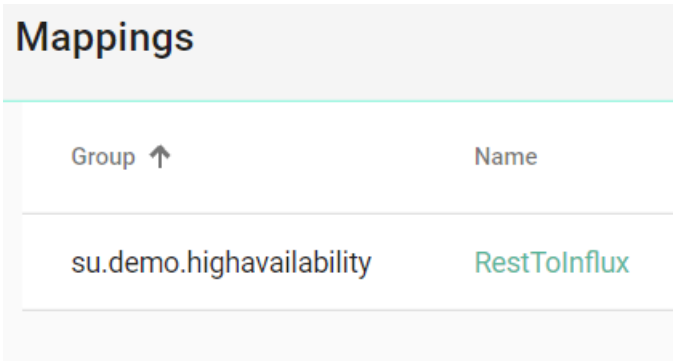
10.2 Mappings

The Mapping represents the SMART**UNIFIER** component that defines when and how to exchange/transform data between two or multiple Information Models. In other words, it is acting as a translator between the different Information Models.

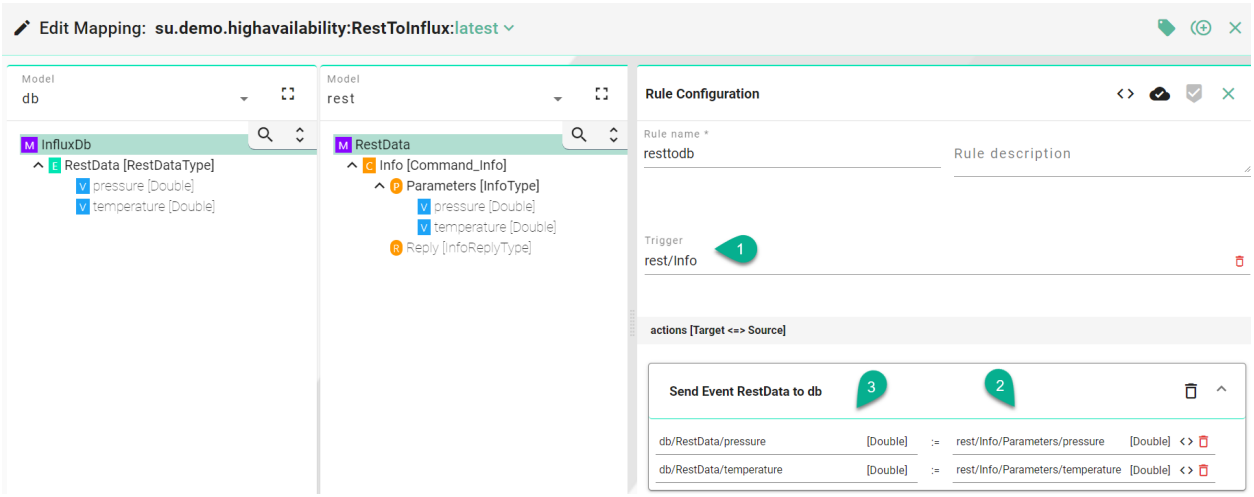
One Mapping consists of one or multiple Rules. A Rule contains a Trigger, which defines when the exchange/transformation takes place, and a list of actions that are defining how the exchange/transformation is done.

The Mapping perspective lists the currently configured Mapping within the SMART**UNIFIER** Manager:

- **RestToInflux**



This Mapping defines when and how to process data from the Rest Server and store it on the InfluxDb.



The trigger to send data is represented by new values for the Rest Server parameters: pressure and temperature **(1)**. The Mapping is done one on one, the source (Rest Server) parameters **(2)** to the destination (Influx Db) parameters **(3)**.