Assignment 2

COMPUTING SYSTEMS AND INFRASTRUCTURES

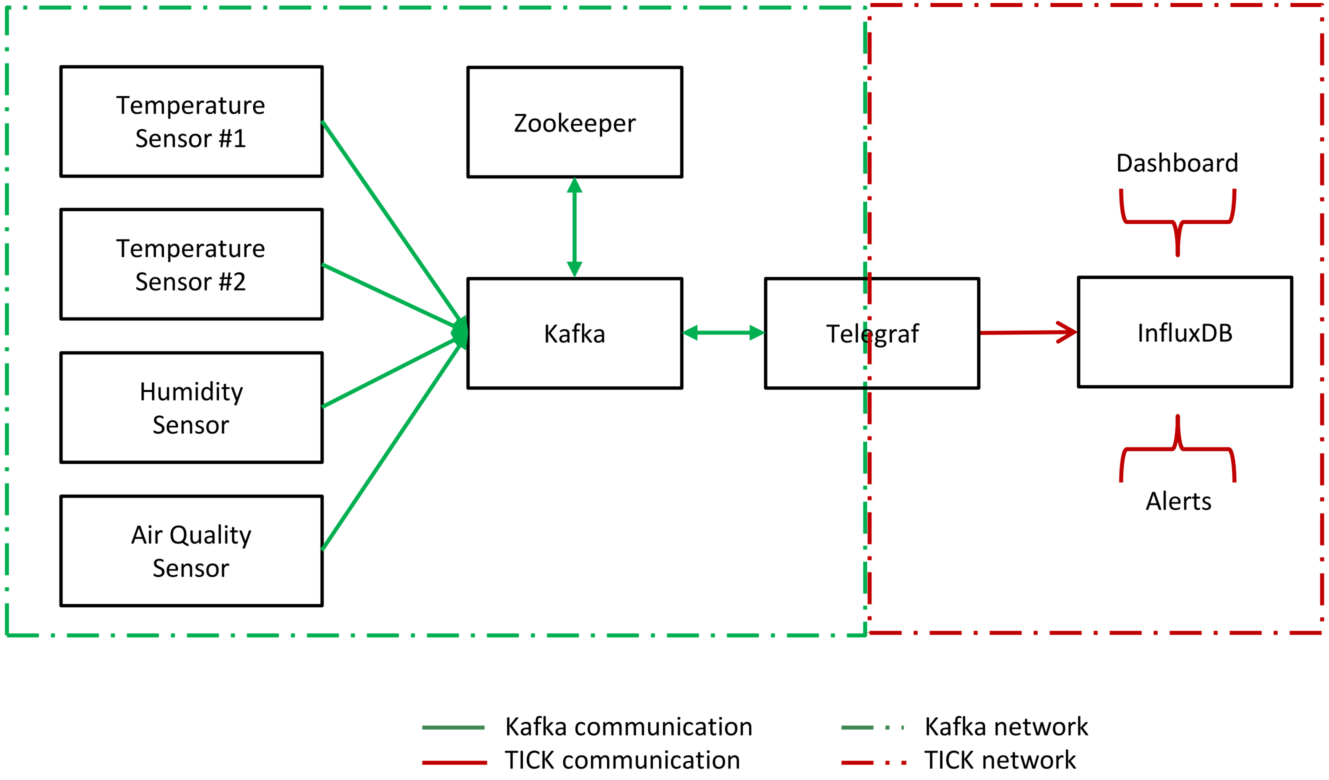
(SISTEMAS E INFRAESTRUTURAS DE COMPUTAÇÃO)

Context and Motivation

- This assignment is focused on taking the first steps to transform a Home into a Smart Home using the concepts and skills learned and trained during the SIC course
- A key component in a Smart Home are the sensors deployed in the premise to gather environmental data. However, before acquiring these devices, it is a common approach to run some tests and this is what you have to do
- 4 data sources are considered: 2 temperature sensors (living room and main room), 1 humidity sensor (living room), and 1 air quality sensor (corridor between the living room and the main room). These sensors will be emulated using Python scripts
- It will be necessary to configure and instantiate a set of applications and services, namely:
  - Zookeeper
  - Kafka
  - TICK stack

Assignment 2

- This assignment aims to develop a solution for Smart Homes that includes:
  - Temperature, humidity, and air quality sensors
  - Time series database that stores all data
  - Visualization dashboard to monitor the data collected
  - Alert notification system based on threshold checks
- Create a **Docker compose** file that is able to instantiate the required goals by instantiating and interconnecting the required services:
  - Zookeeper
  - Kafka broker
  - TICK stack
  - Python scripts → 2 temperature sensor, humidity sensor, air quality sensor



Temperature Sensor #1

Temperature Sensor #2

Humidity Sensor

Air Quality Sensor

Zookeeper

Kafka

Telegraf

InfluxDB

Dashboard

Alerts

Kafka communication

TICK communication

Kafka network

TICK network

About the Temperature Sensor container

- Python docker with a script to simulate the temperature collected each 30 seg following a normal distribution with a mean and standard deviation passed by environmental variables. This script has to use the kafka-python library
- The range of temperature simulated is [-5, 45]
- The sensor data must be sent to the platform through Kafka in the topic “sh-temperature”. The message must follow the InfluxDB line protocol
    - https://docs.influxdata.com/influxdb/v1/write_protocols/line_protocol_tutorial
- The image should be → python:3.11

About the Humidity Sensor container

- Python docker with a script to simulate the humidity collected each 30 seg following a normal distribution with a mean and standard deviation passed by environmental variables. This script has to use the kafka-python library
- The range of temperature simulated is [0, 100]
- The sensor data must be sent to the platform through Kafka in the topic “sh-humidity”. The message must follow the InfluxDB line protocol
  - https://docs.influxdata.com/influxdb/v1/write_protocols/line_protocol_tutorial
- The image should be → python:3.11

About the Air Quality Sensor container

- Python docker with a script to simulate the humidity collected each 30 seg following a normal distribution with a mean and standard deviation passed by environmental variables. This script has to use the kafka-python library
- The range of temperature simulated is [0, 200]
- The sensor data must be sent to the platform through Kafka in the topic “sh-AirQuality”. The message must follow the InfluxDB line protocol
  - https://docs.influxdata.com/influxdb/v1/write_protocols/line_protocol_tutorial
- The image should be → python:3.11

About the Zookeeper container

- The image should be → bitnami/zookeeper:latest
- The Zookeeper data must be preserved in a volume managed by the docker engine
  - zookeeper-data
- This container must allow anonymous login
- This container must have a health-check

About the Kafka container

- The image should be → bitnami/kafka:latest
- The Kafka data must be preserved in a volume managed by the docker engine
  - kafka-data
- This container must have a health-check

About the Telegraf container

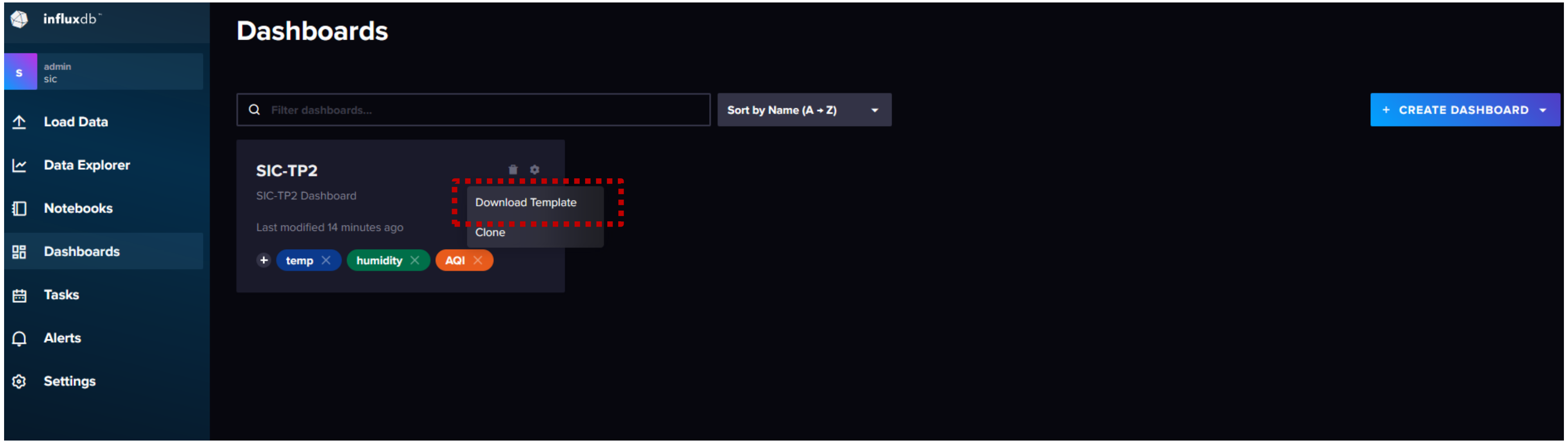
- The image should be → telegraf:latest
- The Telegraf service must use a configuration file bound to the VM local file system
  - ./telegraf/telegraf.conf → This must use a relative path inside your solution folder
  - This container must have a health-check

About the InfluxDB container

- The InfluxDB data must be preserved in a volume managed by the docker engine
  - influxdb-data
- The InfluxDB should store all the time series data in the database “tp2”
- User name for the DB → admin
- Password for the DB → password
- Organization → sic
- Bucket → tp2
- Admin token → LRk4Qb4QtnFqdn83kYDG-1EXsNRkhFGrdZqJQhfy2IOTN7IlyVhWdwEu8PeKlvhXDHomucMZVOpE7tAPLSx0xQ==
- This container must have a health-check

Dashboard

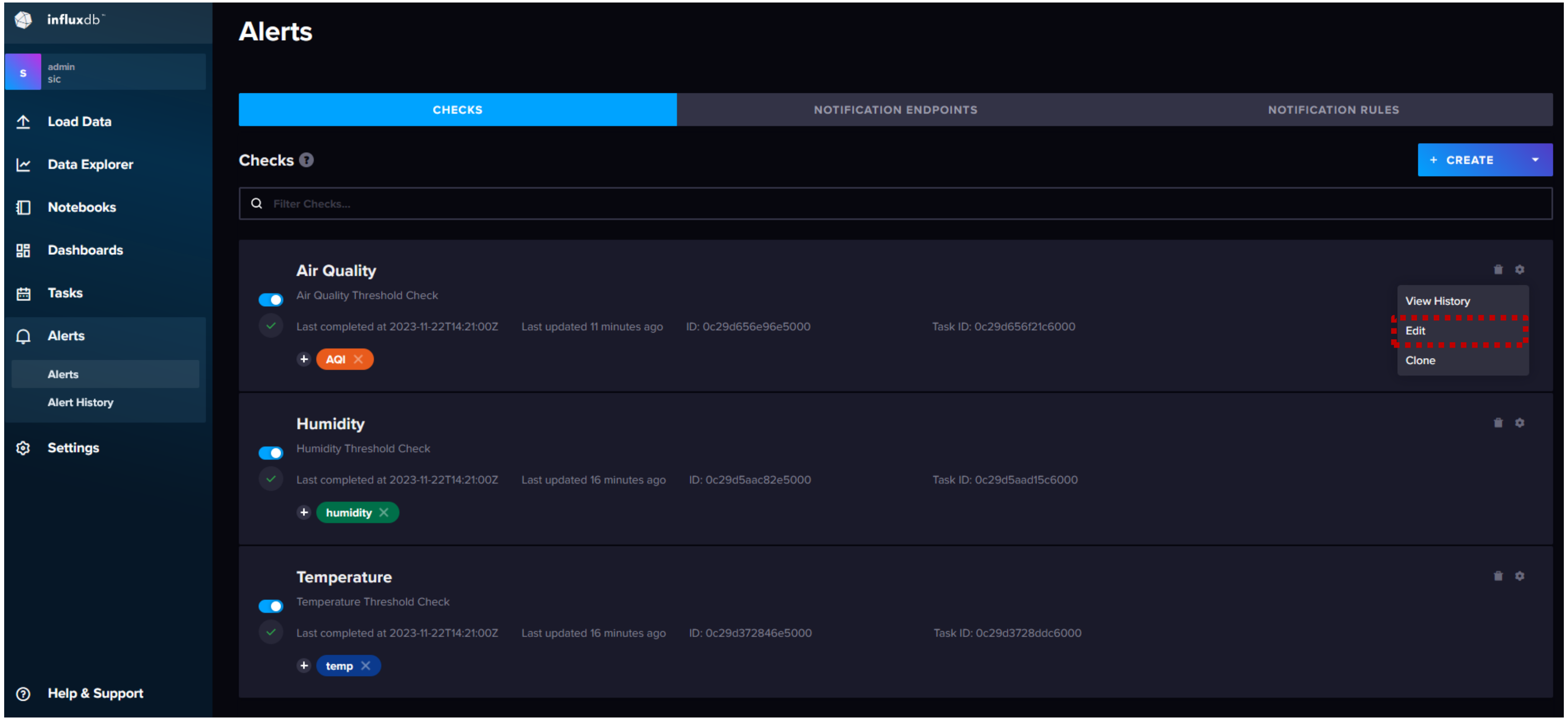
- Create at least one visualization dashboard to allow the monitoring of the data collected by your solution. This dashboard must include at least one visualization type of: graph, table, stat, gauge, etc.
- You have to provide a dashboard.json file during the submission that allows importing your dashboard



Alerts (1/2)

- Create an alert based on a threshold check for the temperature
  - Critical → Value is outside the range [10, 35]
- Create an alert based on a threshold check for the humidity
  - Critical → Value is above 70, Warn → Value is above 60, OK → Value is inside the range [30, 59]
- Create an alert based on a threshold check for the air quality
  - Critical → Value is above 151, Warn → Value is above 101, Info → Value is in the range [51, 100], OK → Value is inside range [0, 50]
- You have to provide 3 files (temp-flux, hum-flux, aqi-flux) with the FLUX code of the alerts specified above
  - See in the next where to access the code

Alerts (2/2)



Air Quality

Air Quality Threshold Check

Last completed at 2023-11-22T14:21:00Z Last updated 11 minutes ago ID: 0c29d656e96e5000 Task ID: 0c29d656f21c6000

AQI

Humidity

Humidity Threshold Check

Last completed at 2023-11-22T14:21:00Z Last updated 16 minutes ago ID: 0c29d5aac82e5000 Task ID: 0c29d5aad15c6000

humidity

Temperature

Temperature Threshold Check

Last completed at 2023-11-22T14:21:00Z Last updated 16 minutes ago ID: 0c29d372846e5000 Task ID: 0c29d3728ddc6000

temp

General guidelines (1/2)

- Rely only on official images to develop this exercise, paying attention to the specific suggestion for each component
- For the Python scripts, create the required Dockerfile to build the custom images
- Usage of volumes is required when appropriate to have persistent data
- The implementation of your solution must consider network isolation, for this consider two different networks for your containers → tick-net and kafka-net
- Only exposed ports when where required
- Address the container dependency requirements (create custom health checks)

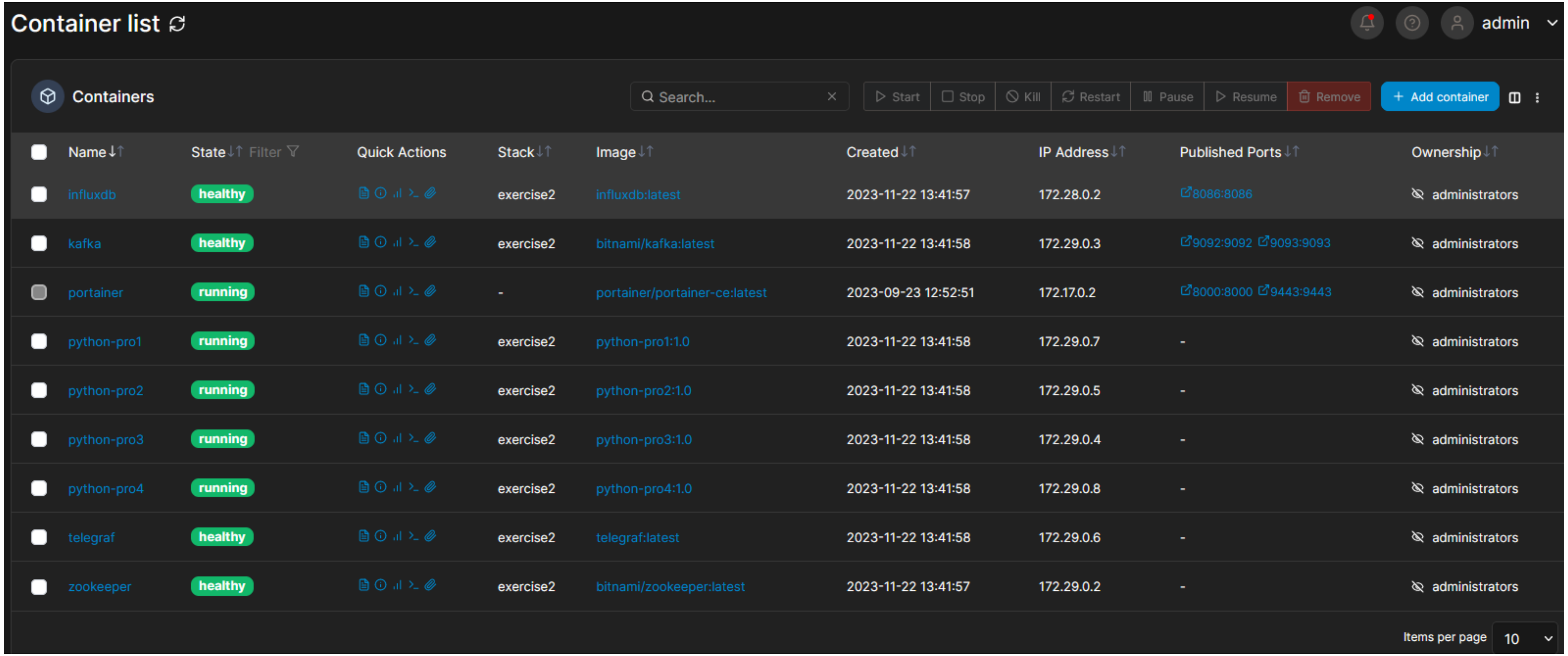
General guidelines (2/2)

- The compose file must be self-contained (to not depend on existing custom images, volumes or networks)
- You can use a .env file to provide configuration values and parameters
- No configuration can depend on the operating system or absolute file paths. Use only relative paths to your assignment folder
- Create a work plan to avoid “try” to do anything at the last moment
  - Stick to your work plan and take advantage of the PLs dedicated to the support of the assignment

Examples of the expected outcome (1/4)

- View of the directory to be submitted to the InforEstudante (.zip)
  - exercise2
    - compose.yaml
    - dashboard
      - dashboard.json
    - flux
      - temp, humidity, aqi
    - telegraf
      - telegraf.conf
    - producerN → N= 1, 2, 3, 4
      - producerN.py → N= 1, 2, 3, 4
      - Dockerfile
  - For the produces N=1,2 → temp; N=3 → humidity; N=4 → AQI

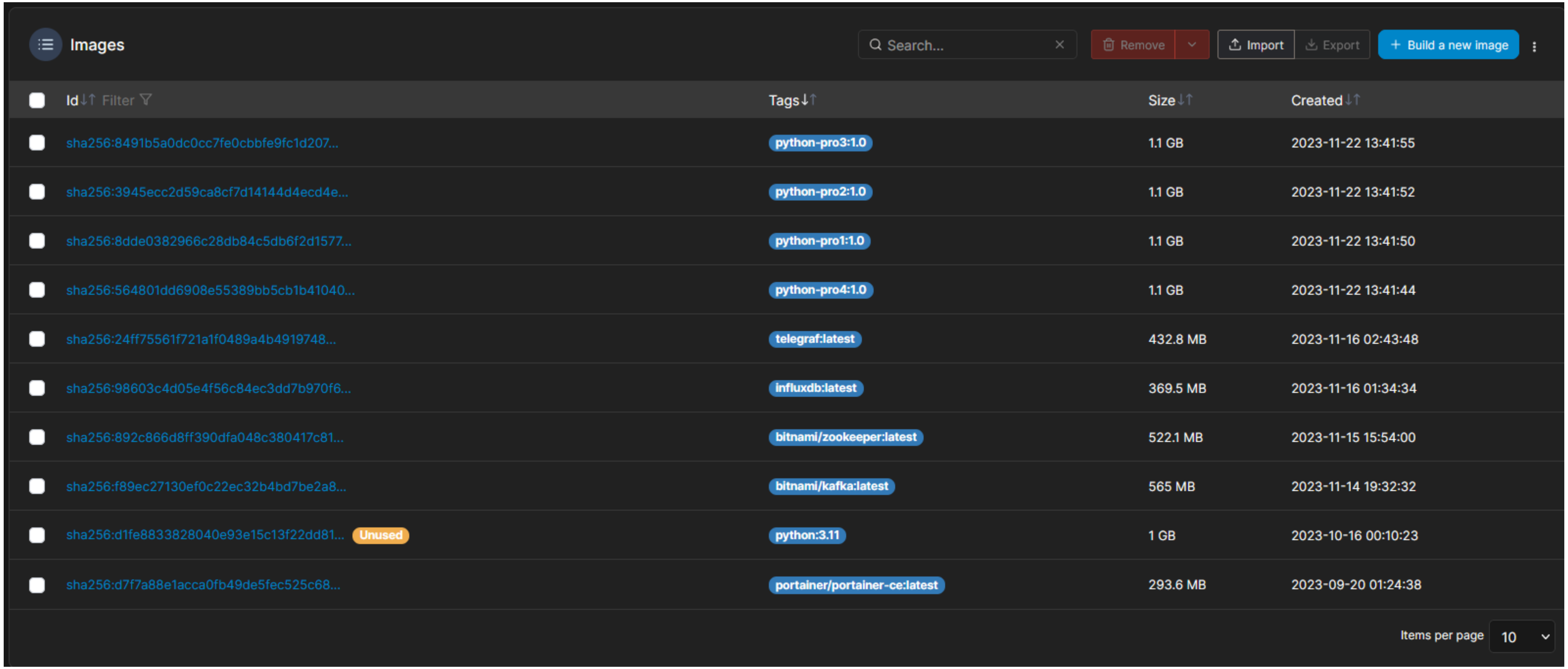
Examples of the expected outcome (2/4)



☐	Name	State Filter	Quick Actions	Stack	Image	Created	IP Address	Published Ports	Ownership
☐	influxdb	healthy		exercise2	influxdb:latest	2023-11-22 13:41:57	172.28.0.2	8086:8086	administrators
☐	kafka	healthy		exercise2	bitnami/kafka:latest	2023-11-22 13:41:58	172.29.0.3	9092:9092 9093:9093	administrators
☐	portainer	running		-	portainer/portainer-ce:latest	2023-09-23 12:52:51	172.17.0.2	8000:8000 9443:9443	administrators
☐	python-pro1	running		exercise2	python-pro1:1.0	2023-11-22 13:41:58	172.29.0.7	-	administrators
☐	python-pro2	running		exercise2	python-pro2:1.0	2023-11-22 13:41:58	172.29.0.5	-	administrators
☐	python-pro3	running		exercise2	python-pro3:1.0	2023-11-22 13:41:58	172.29.0.4	-	administrators
☐	python-pro4	running		exercise2	python-pro4:1.0	2023-11-22 13:41:58	172.29.0.8	-	administrators
☐	telegraf	healthy		exercise2	telegraf:latest	2023-11-22 13:41:58	172.29.0.6	-	administrators
☐	zookeeper	healthy		exercise2	bitnami/zookeeper:latest	2023-11-22 13:41:57	172.29.0.2	-	administrators

Items per page 10

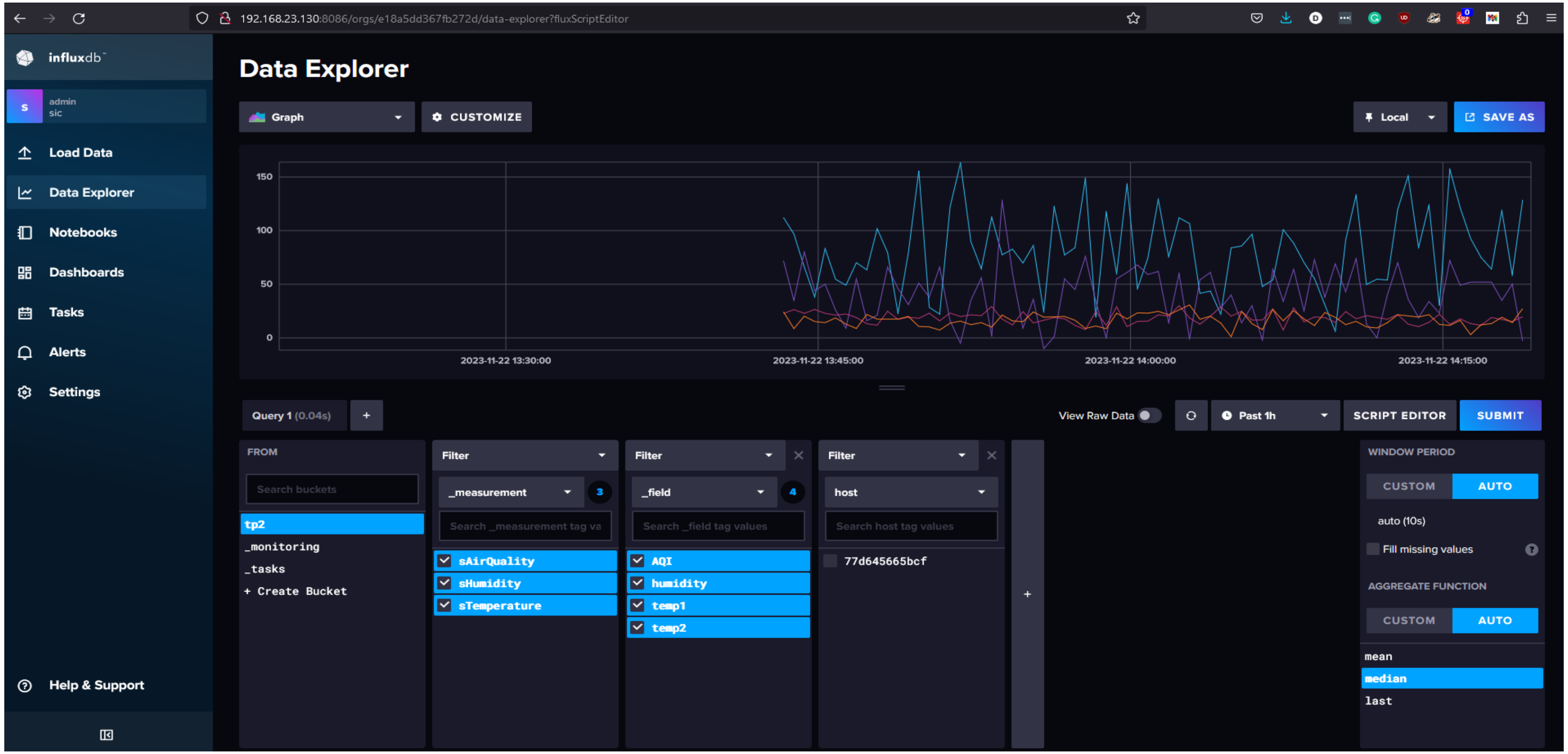
Examples of the expected outcome (3/4)



☐	Id Filter	Tags	Size	Created
☐	sha256:8491b5a0dc0cc7fe0cbbfe9fc1d207...	python-pro3:1.0	1.1 GB	2023-11-22 13:41:55
☐	sha256:3945ecc2d59ca8cf7d14144d4ecd4e...	python-pro2:1.0	1.1 GB	2023-11-22 13:41:52
☐	sha256:8dde0382966c28db84c5db6f2d1577...	python-pro1:1.0	1.1 GB	2023-11-22 13:41:50
☐	sha256:564801dd6908e55389bb5cb1b41040...	python-pro4:1.0	1.1 GB	2023-11-22 13:41:44
☐	sha256:24ff75561f721a1f0489a4b4919748...	telegraf:latest	432.8 MB	2023-11-16 02:43:48
☐	sha256:98603c4d05e4f56c84ec3dd7b970f6...	influxdb:latest	369.5 MB	2023-11-16 01:34:34
☐	sha256:892c866d8ff390dfa048c380417c81...	bitnami/zookeeper:latest	522.1 MB	2023-11-15 15:54:00
☐	sha256:f89ec27130ef0c22ec32b4bd7be2a8...	bitnami/kafka:latest	565 MB	2023-11-14 19:32:32
☐	sha256:d1fe8833828040e93e15c13f22dd81... Unused	python:3.11	1 GB	2023-10-16 00:10:23
☐	sha256:d7f7a88e1acca0fb49de5fec525c68...	portainer/portainer-ce:latest	293.6 MB	2023-09-20 01:24:38

Items per page 10

Examples of the expected outcome (4/4)



192.168.23.130:8086/orgs/e18a5dd367fb272d/data-explorer?fluxScriptEditor

Data Explorer

Query 1 (0.04s)

Past 1h

FROM

tp2
_monitoring
_tasks
+ Create Bucket

Filter: _measurement 3

☑ sAirQuality
☑ sHumidity
☑ sTemperature

Filter: _field 4

☑ AQI
☑ humidity
☑ temp1
☑ temp2

Filter: host

□ 77d645665bcf

WINDOW PERIOD

auto (10s)

□ Fill missing values

AGGREGATE FUNCTION

mean
median
last

Final remarks

- The assignment must be developed in groups of **2** students
- The assignment must be submitted in InforEstudante in a **.zip** file containing all required files
- Students must enroll in one of the available defense time slots available in InforEstudante

- Submission deadline: **09/12/2023**
- Defence: **11/12/2023**