Scicat Perfomance Testing

SCT Meeting

20.01.2025

Armando Bermúdez Martínez, Tim Wetzel

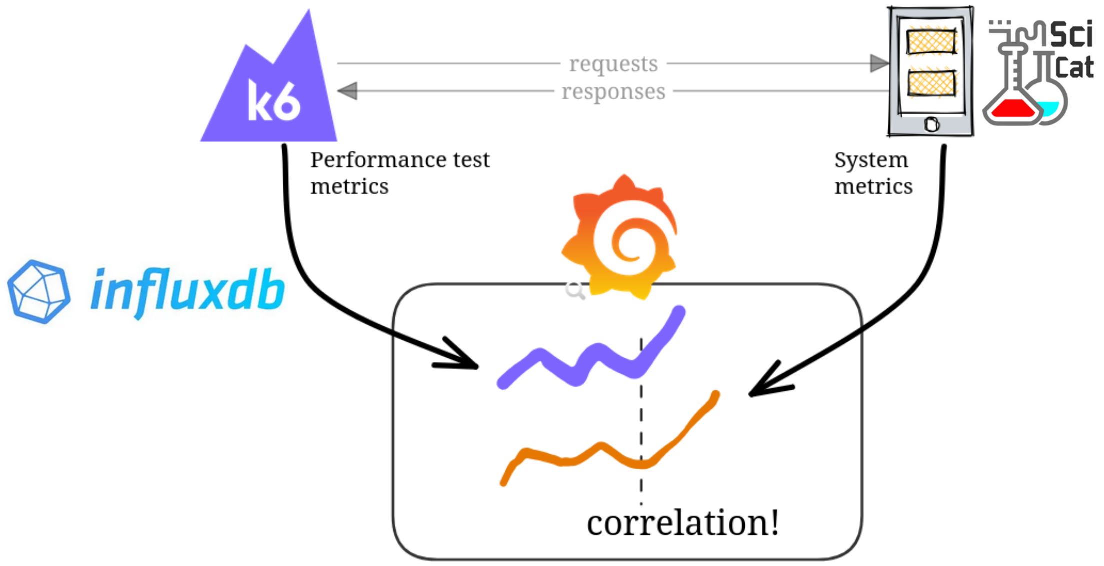
Objectives

- Ensure it can handle large datasets and high user loads.
- Test the reliability and responsiveness under different scenarios.

Types of tests performed

- **Load Testing**: Determine how the system behaves under expected traffic.
- **Stress Testing:** Test limits by simulating heavy loads.
- **Soak Testing:** Assess stability over an extended period
- **Spike Testing:** Measure response to sudden surges in traffic.

Framework



Setup:

Docker Compose: Combines K6 for generating load, Grafana for visualization, and InfluxDB for storing performance metrics.

Benefits of this stack

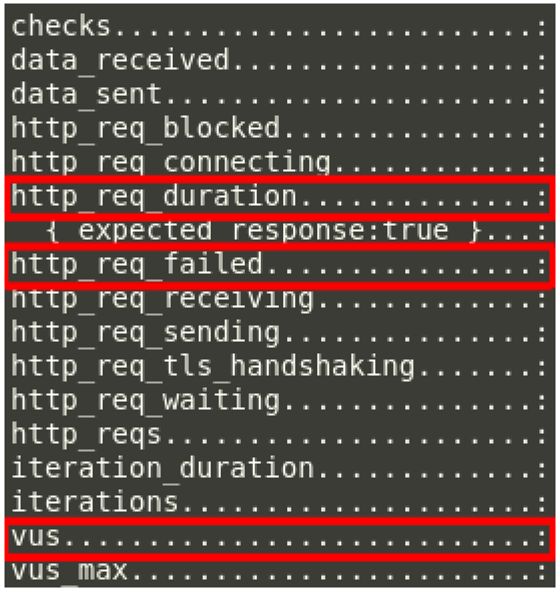
- Easy setup and teardown.
- Centralized performance monitoring.
- Real-time data visualization.

Framework

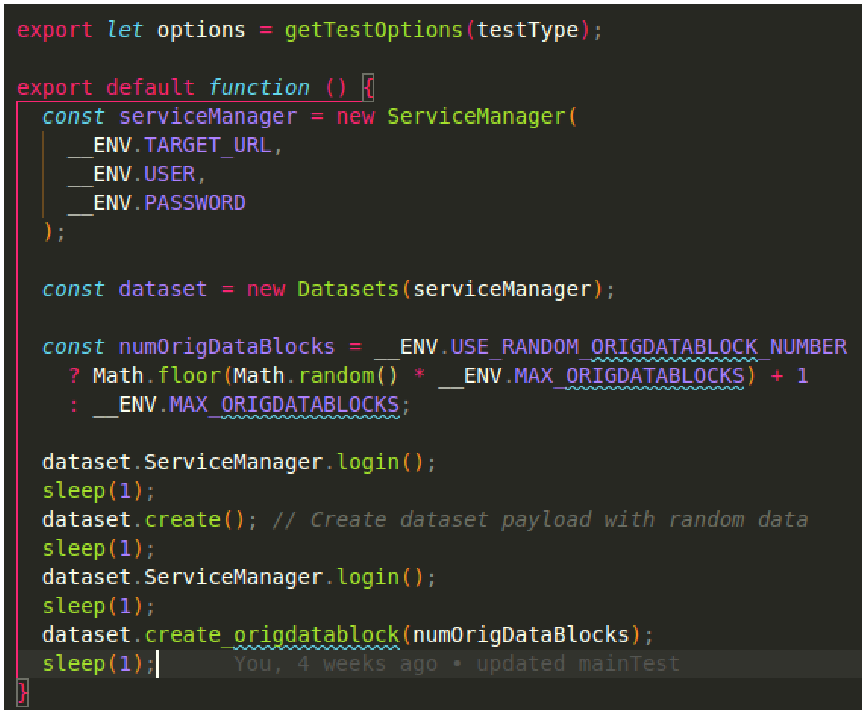
- Custom app based on k6: an open-source tool for load testing.
- Allows you to simulate traffic, measure performance, and find bottlenecks.

Logic organized in classes and methods.

Possibility to generate random datasets, origdatablocks (**Here up to 50 odb per data set**)



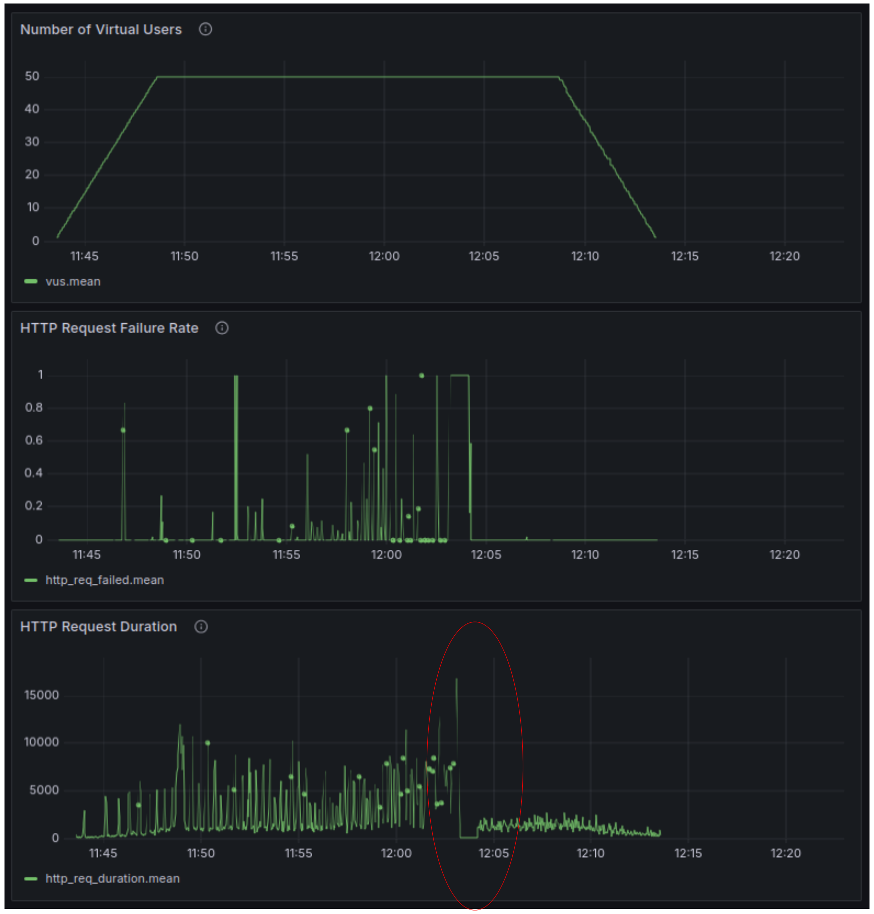
```
checks.........................:
data_received..................:
data_sent......................:
http_req_blocked...............:
http_req_connecting............:
http_req_duration..............:
  { expected_response:true }...:
http_req_failed................:
http_req_receiving.............:
http_req_sending...............:
http_req_tls_handshaking.......:
http_req_waiting...............:
http_reqs......................:
iteration_duration.............:
iterations.....................:
vus............................:
vus_max........................:
```

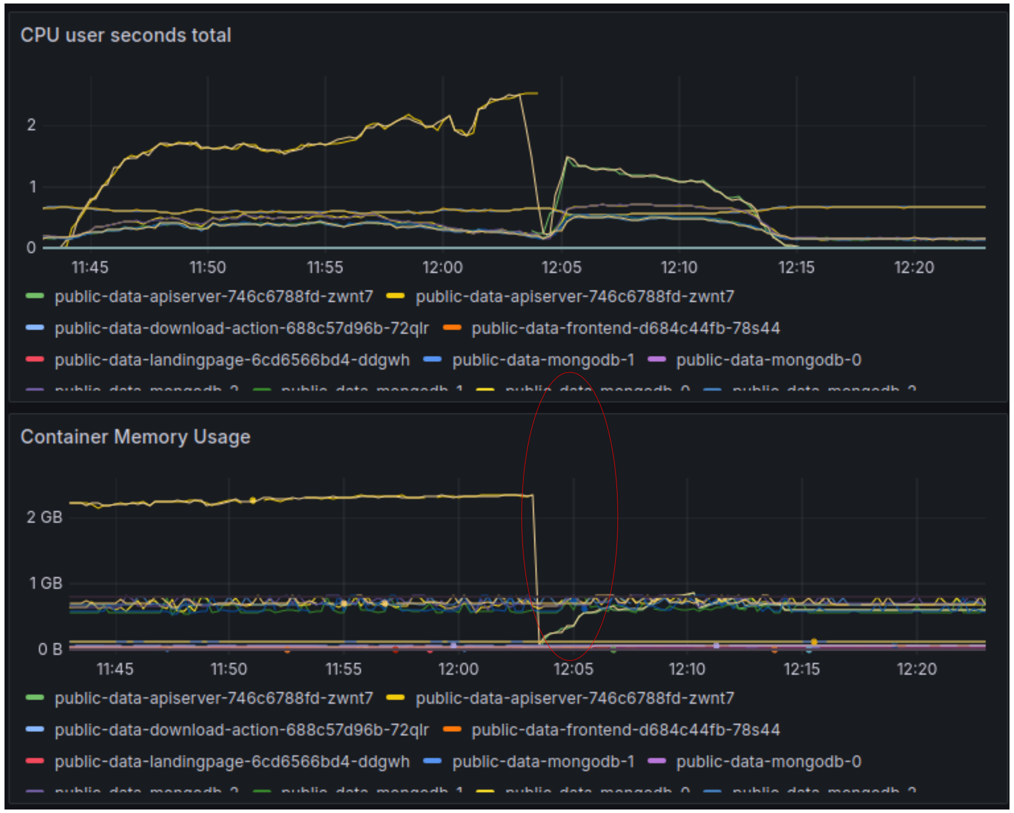


```
export let options = getTestOptions(testType);

export default function () {
  const serviceManager = new ServiceManager(
    __ENV.TARGET_URL,
    __ENV.USER,
    __ENV.PASSWORD
  );

  const dataset = new Datasets(serviceManager);

  const numOrigDataBlocks = __ENV.USE_RANDOM_ORIGDATABLOCK_NUMBER
    ? Math.floor(Math.random() * __ENV.MAX_ORIGDATABLOCKS) + 1
    : __ENV.MAX_ORIGDATABLOCKS;

  dataset.ServiceManager.login();
  sleep(1);
  dataset.create(); // Create dataset payload with random data
  sleep(1);
  dataset.ServiceManager.login();
  sleep(1);
  dataset.create_origdatablock(numOrigDataBlocks);
  sleep(1);      You, 4 weeks ago • updated mainTest
}
```

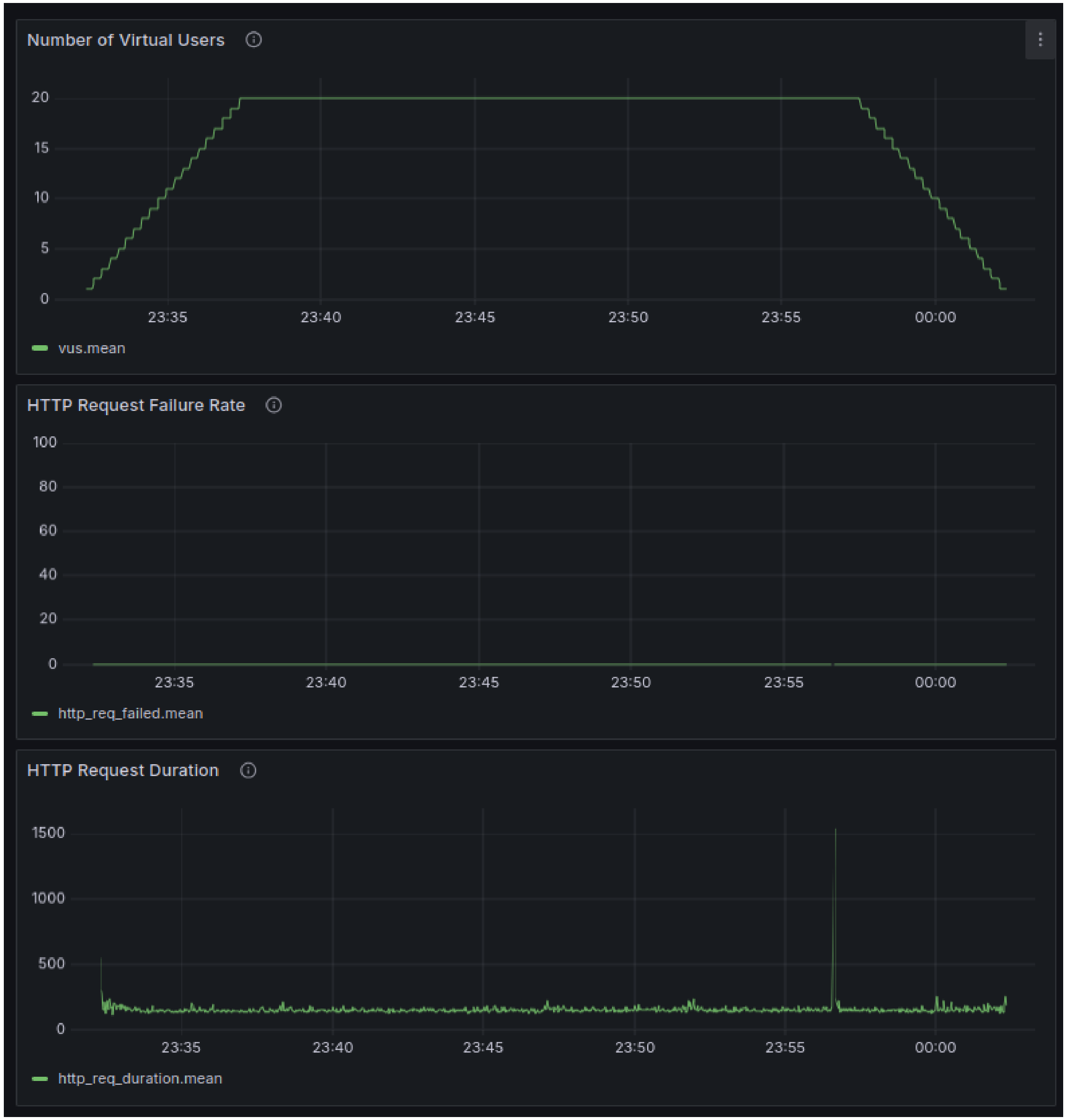
Resuts and observations: first tests

- Scicat first subjected to extreme conditions: 50 Vus per second making simultaneous api calls.
- We found indications of memory leak for the backend. This is unlikely to be a problem. Nevertheless we now use two replicas for the backend to circumvent it.

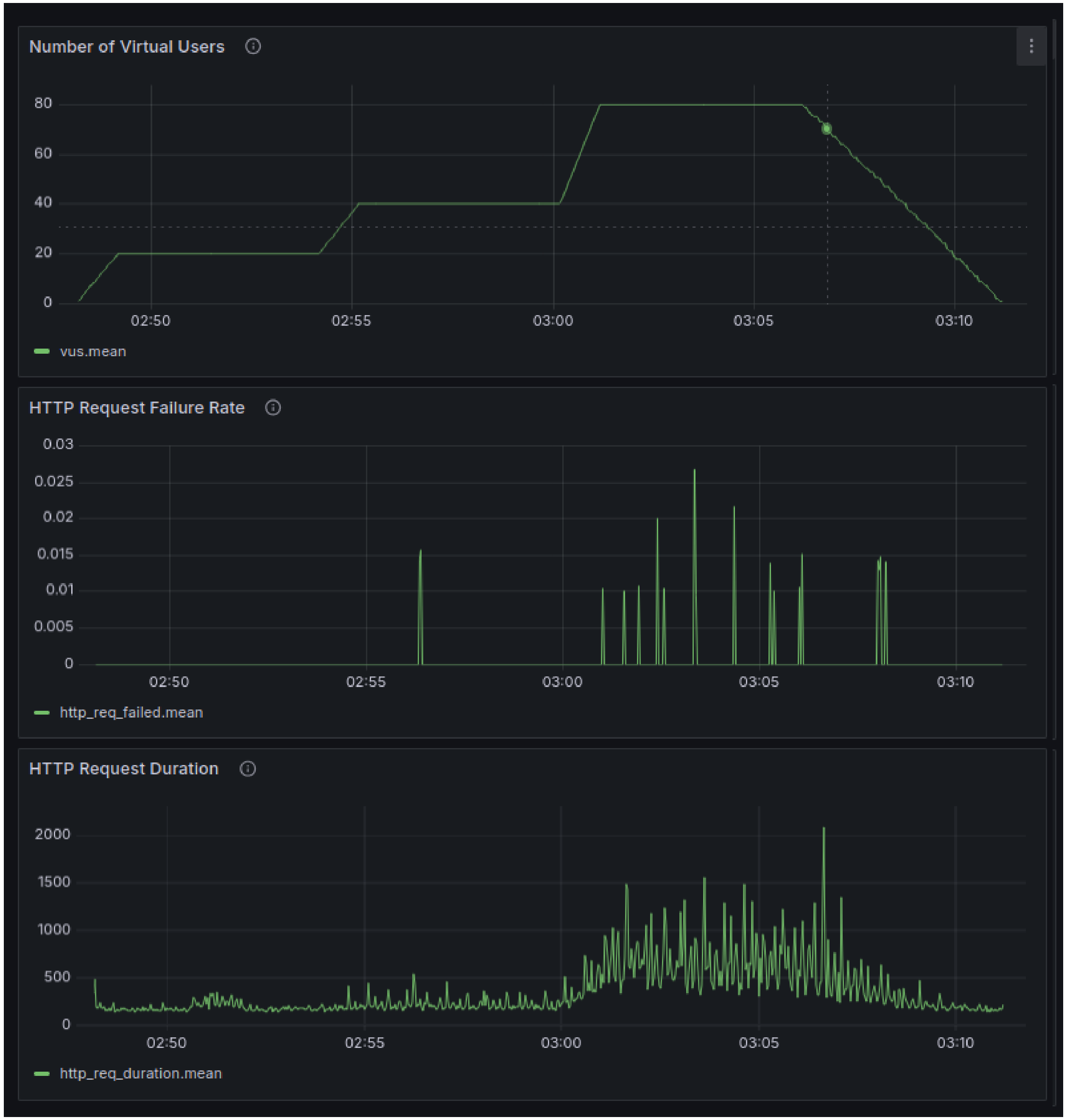




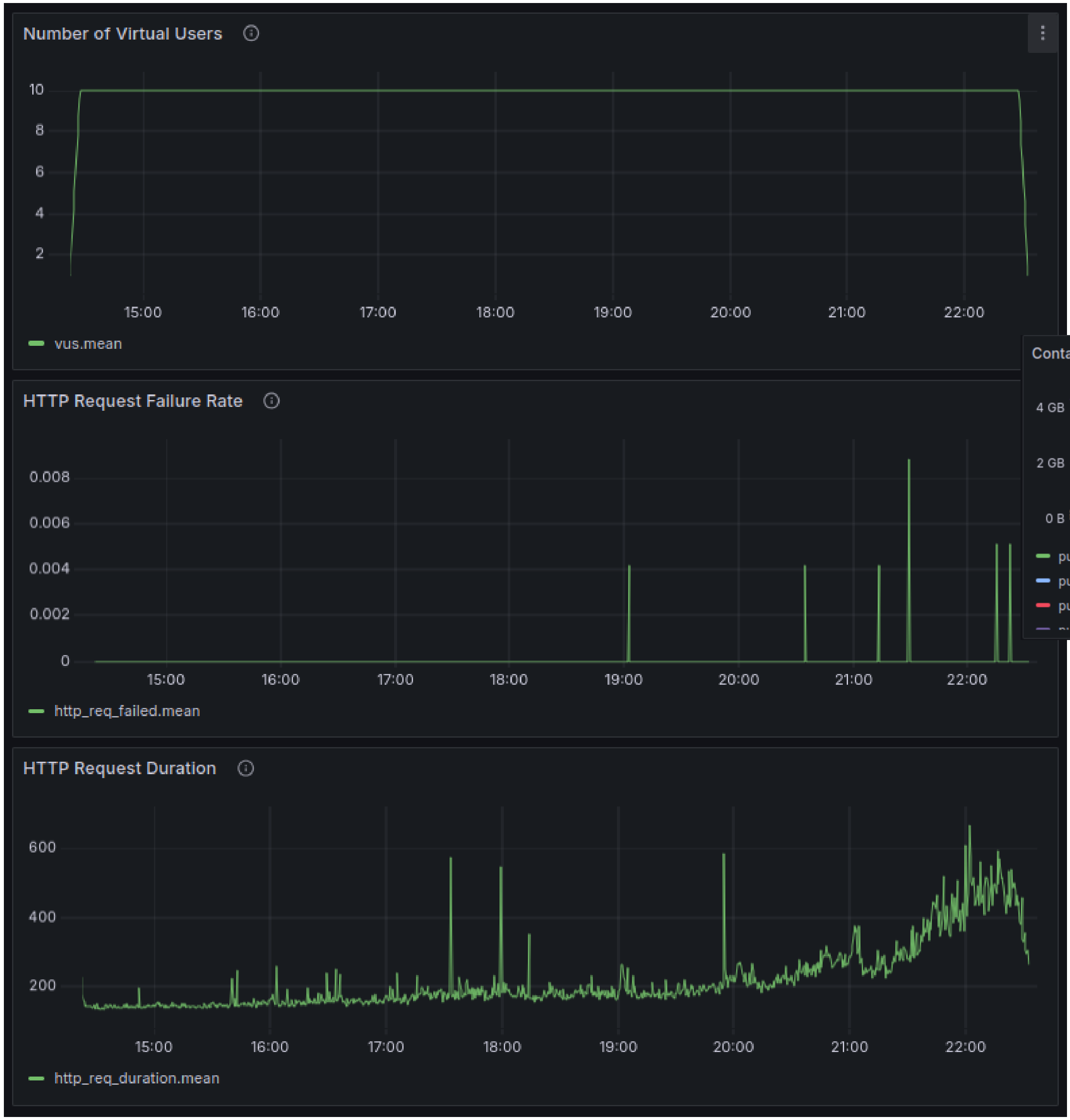
Results and observations: Load

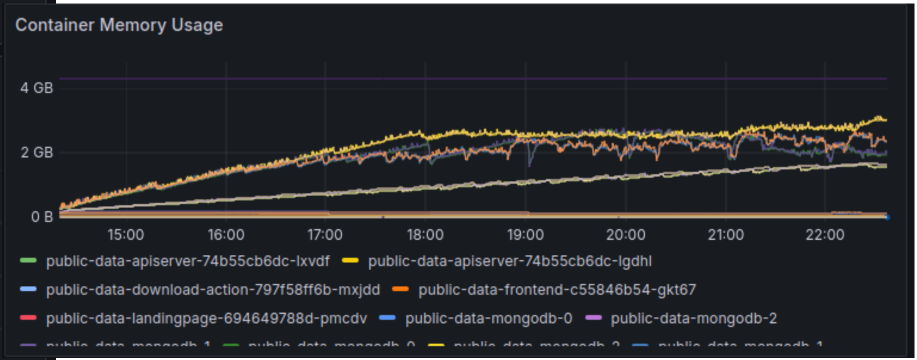


Results and observations: Stress

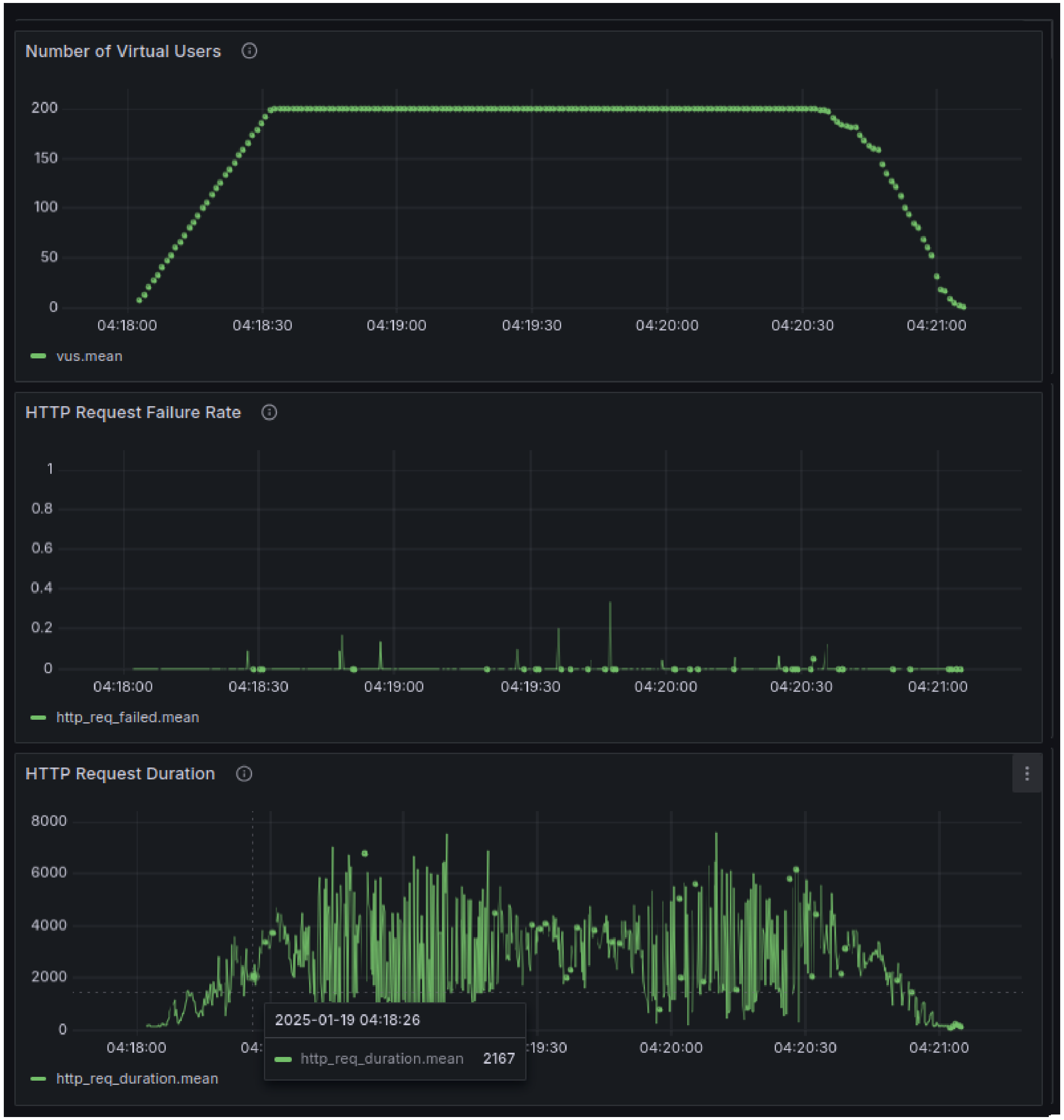


Results and observations: Soak

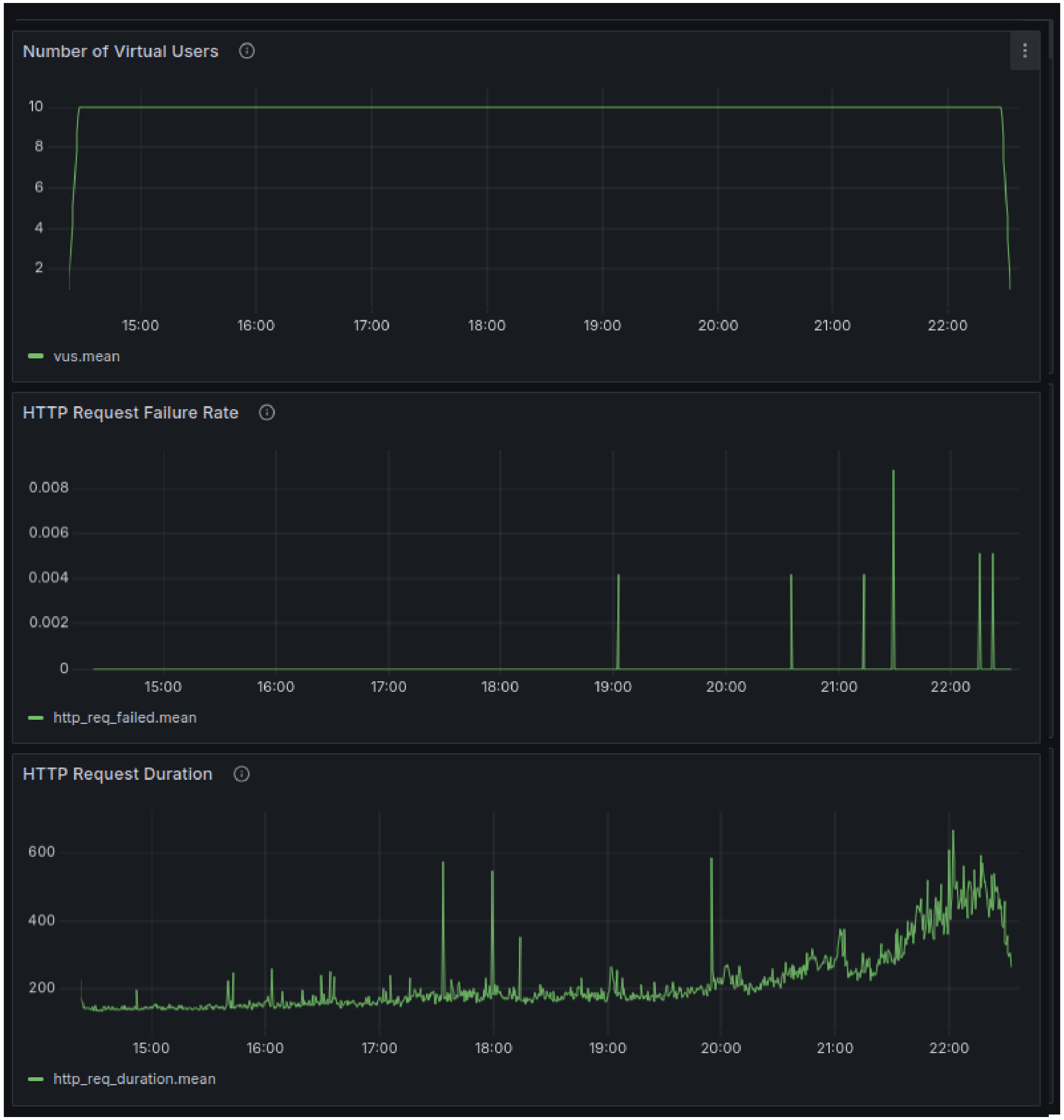




Results and observations: Spike



Results and observations: Spike


Number of Virtual Users
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
vus.mean
HTTP Request Failure Rate
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
http_req_failed.mean
HTTP Request Duration
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
http_req_duration.mean