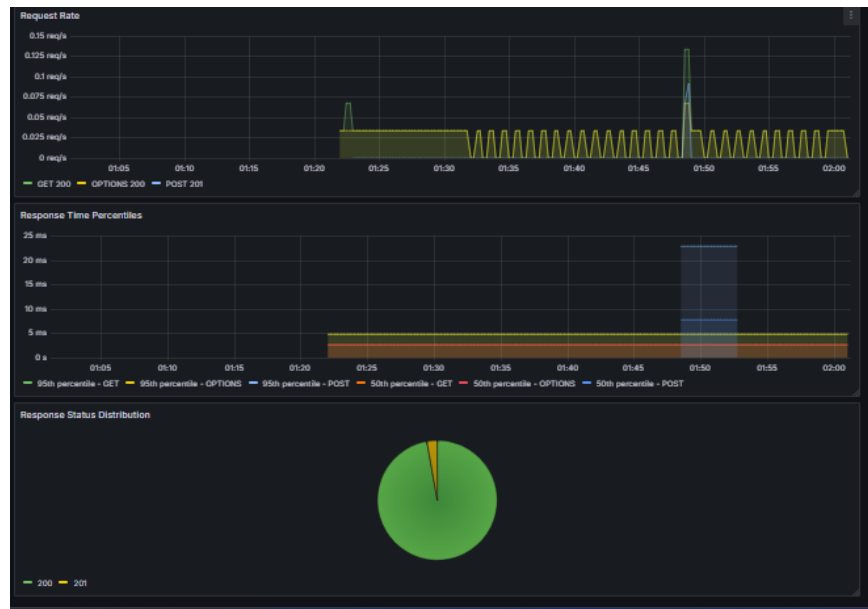


DevOps Final project

Lazare Bedukadze

Grafana dashboard

The dashboard displays 9 different statistics of the CPU and back-end service.



Vulnerability scans

Trivy scan results:

No critical vulnerability was detected in any of the services, except for postgres, which I could not fix.

- backend

Python (python-pkg) =====							
Total: 2 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 2, CRITICAL: 0)							
Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title	
setuptools (METADATA)	CVE-2024-6345	HIGH	fixed	65.5.1	70.0.0	pypa/setuptools: Remote code execution via download functions in the package_index module in... https://avd.aquasec.com/nvd/cve-2024-6345	
	CVE-2025-47273				78.1.1	setuptools: Path Traversal Vulnerability in setuptools PackageIndex https://avd.aquasec.com/nvd/cve-2025-47273	

- db

usr/local/bin/gosu (gobinary) =====							
Total: 61 (UNKNOWN: 0, LOW: 1, MEDIUM: 26, HIGH: 31, CRITICAL: 3)							
Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title	
stdlib	CVE-2023-24538	CRITICAL	fixed	v1.18.2	1.19.8, 1.20.3	golang: html/template: backticks not treated as delimiters https://avd.aquasec.com/nvd/cve-2023-24538	
	CVE-2023-24540				1.19.9, 1.20.4	golang: html/template: improper handling of Java whitespace https://avd.aquasec.com/nvd/cve-2023-24540	
	CVE-2024-24790				1.21.11, 1.22.4	golang: net/netip: Unexpected behavior from Is...	

- frontend

devops_final-frontend (alpine 3.21.3) =====							
Total: 2 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 2, CRITICAL: 0)							
Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title	
libxml2	CVE-2025-32414	HIGH	fixed	2.13.4-r5	2.13.4-r6	libxml2: Out-of-Bounds Read in libxml2 https://avd.aquasec.com/nvd/cve-2025-32414	
	CVE-2025-32415					libxml2: Out-of-bounds Read in xmlSchemaSect... https://avd.aquasec.com/nvd/cve-2025-32415	

- Grafana

```
usr/share/grafana/bin/grafana (gobinary)
=====
Total: 1 (UNKNOWN: 0, LOW: 1, MEDIUM: 0, HIGH: 0, CRITICAL: 0)
```

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	
github.com/cloudflare/circl	GHSA-2x5j-vhc8-9cwm	LOW	fixed	v1.6.0	1.6.1	CIRCL-Four incorrect https://github.com/cloudflare/circl

- Node-exporter

```
bin/node_exporter (gobinary)
=====
Total: 4 (UNKNOWN: 0, LOW: 0, MEDIUM: 4, HIGH: 0, CRITICAL: 0)
```

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	
golang.org/x/net	CVE-2025-22872	MEDIUM	fixed	v0.37.0	0.38.0	golang.org/x/net/html: Inconsistent handling of in os in syscall... https://avd.aquasec.com/nvd/cve-2025-22872
stdlib	CVE-2025-0913			v1.23.7	1.23.10, 1.24.4	Inconsistent handling of in os in syscall... https://avd.aquasec.com/nvd/cve-2025-0913

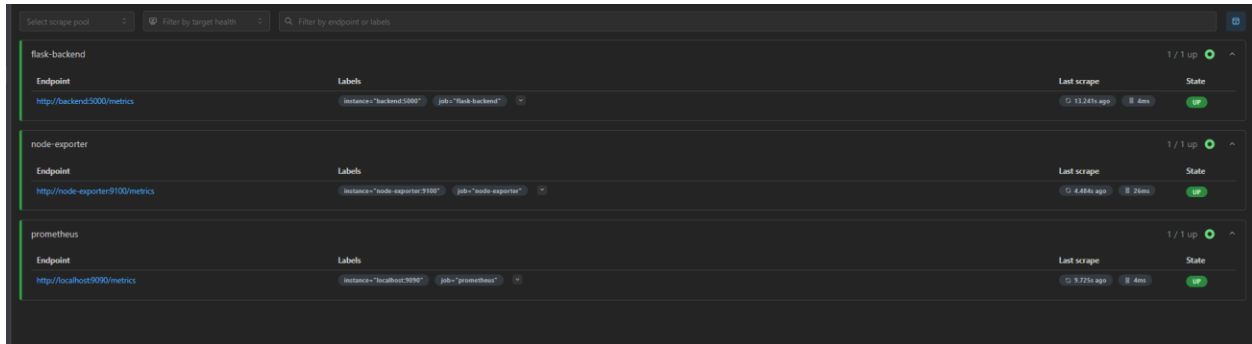
- Prometheus

```
bin/prometheus (gobinary)
=====
Total: 3 (UNKNOWN: 0, LOW: 0, MEDIUM: 2, HIGH: 1, CRITICAL: 0)
```

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title
stdlib	CVE-2025-22874	HIGH	fixed	v1.24.3	1.24.4	crypto/x509: Usage of ExtKeyUsage validation in crypto/x509 https://avd.aquasec.com/nvd/cve-2025-22874
	CVE-2025-0913	MEDIUM			1.23.10, 1.24.4	Inconsistent handling of 0_CREATE in os in syscall... https://avd.aquasec.com/nvd/cve-2025-0913

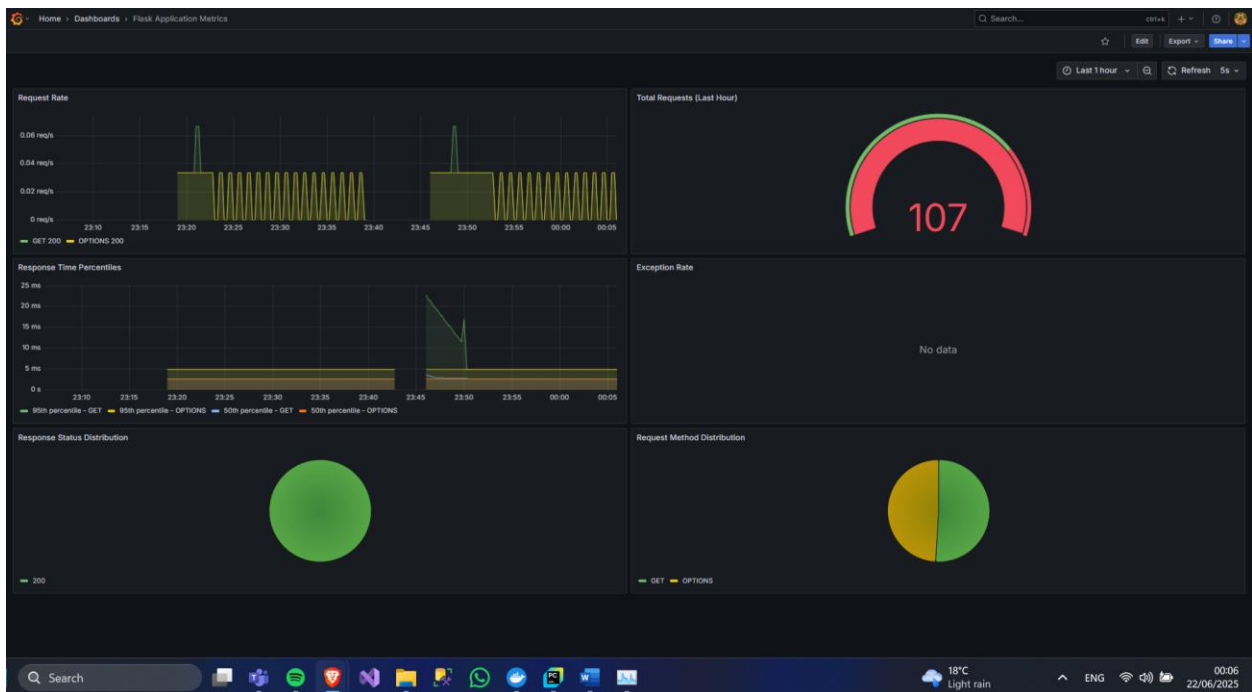
Simulated incident

Before the simulation here are the stats:



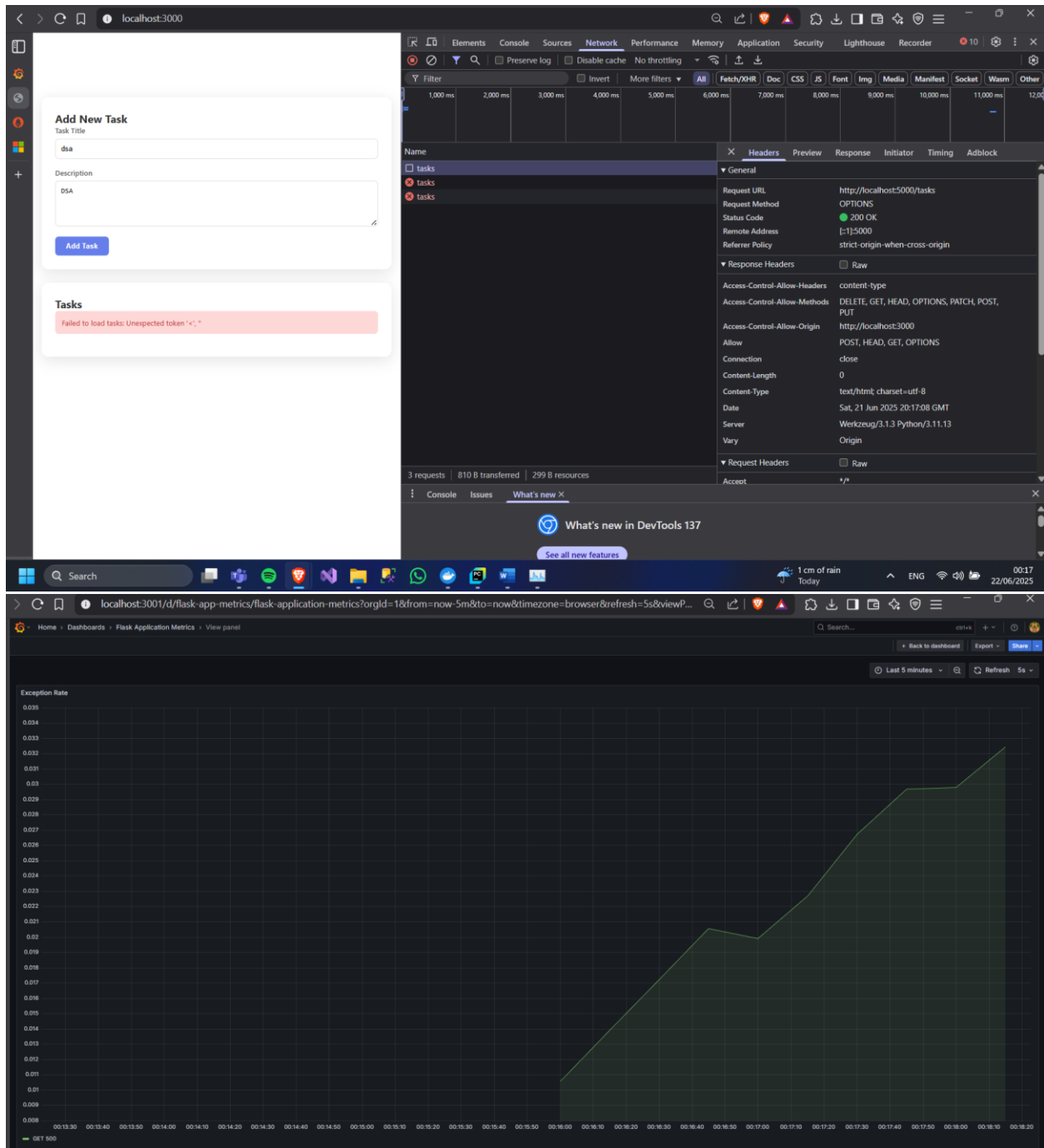
In Prometheus we see the flask-backend clearly running, alongside node-exporter and Prometheus.

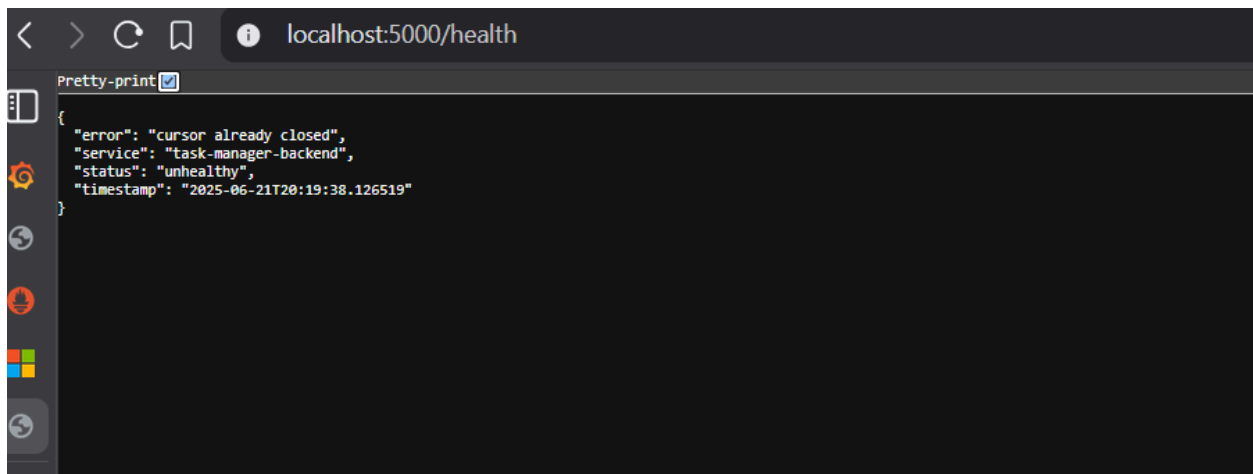
Here are the analytics of the backend



Now we kill the database service, which will cause the app to stop working.

Killing the db caused the application to stop working.





The Health-check also shows that the app has had a problem.

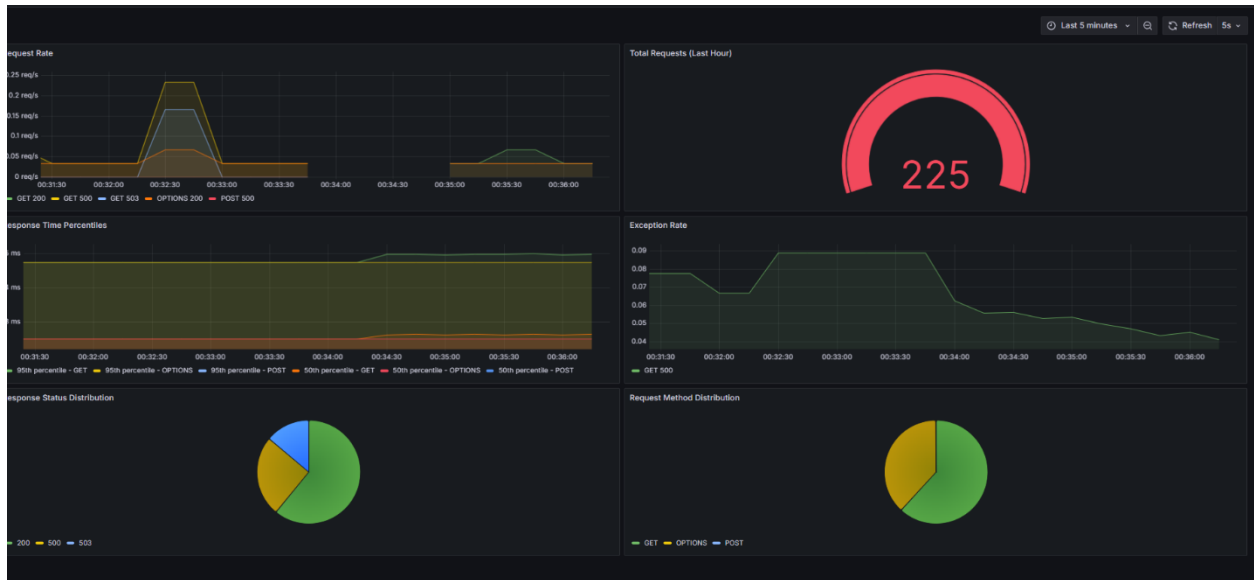
Postmortem

Summary:

The database container (devops_db) went down, causing the backend service to fail and rendering the entire application non-functional.

Impact:

Due to the backend's dependency on the database, the backend service stopped working. As a result, the frontend (UI) could no longer function correctly. All core services of the application were affected.



Timeline:

00:15 – Last confirmed successful request from the application

00:16 – Ui stopped working

00:17 – Checking the back-end service's health-check, it revealed, there was a problem with the database.

00:31 – After having trouble with re-running the sole database container, I had to restart the entire application, which resolved the problem.

00:32 – Application up and running again, health-check returns healthy, UI working, backend – working, database – working

Root Cause:

The database container was manually killed (docker kill devops_db), leading to the failure of all dependent services.

Resolution:

Restarted the entire application with docker compose to bring all services back online

Lessons Learned:

- Application won't run properly unless all 3 services (database, frontend, backend) are healthy
 - The database and backend are especially crucial.
- Don't kill containers :D

Action Items:

- Document procedures for resolving such problems.
- Set up auto-restart policies for containers