universidade de aveiro
deti departamento de electrónica, telecomunicações e informática

Doctalk

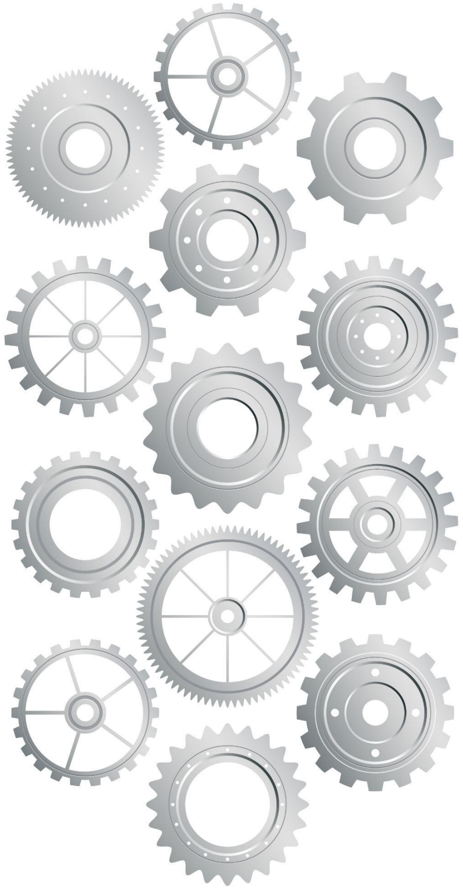
Your online medical appointments

- Jodionísio Muachifi «97147»
- Rúben Castelhano «97688»
- Matilde Costa «98507»
- Rúben Saldanha «98241»

TP1 - Prof. Diogo Gomes

June 2023 | EGS Project Presentation

TABLE OF CONTENTS



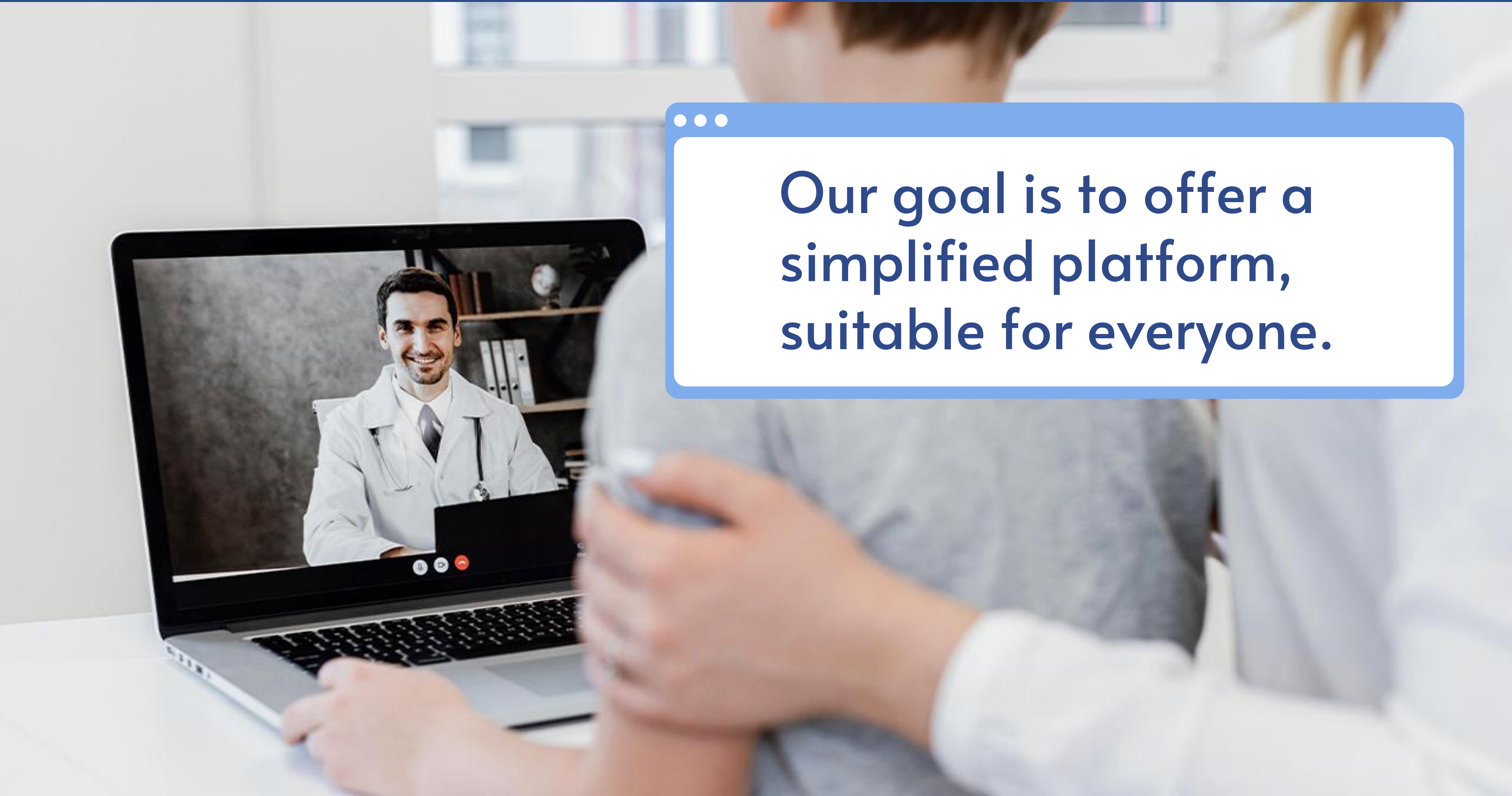
INTRODUCTION

- Engineering and Services Management
- Service Oriented Architecture (SOA)
- Service Monitoring

01.

OUR IDEA

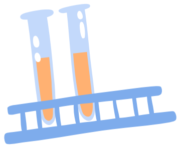
Create an online system for medical appointments, with support for doctor-patient video calls



Our goal is to offer a simplified platform, suitable for everyone.

MAIN FEATURES

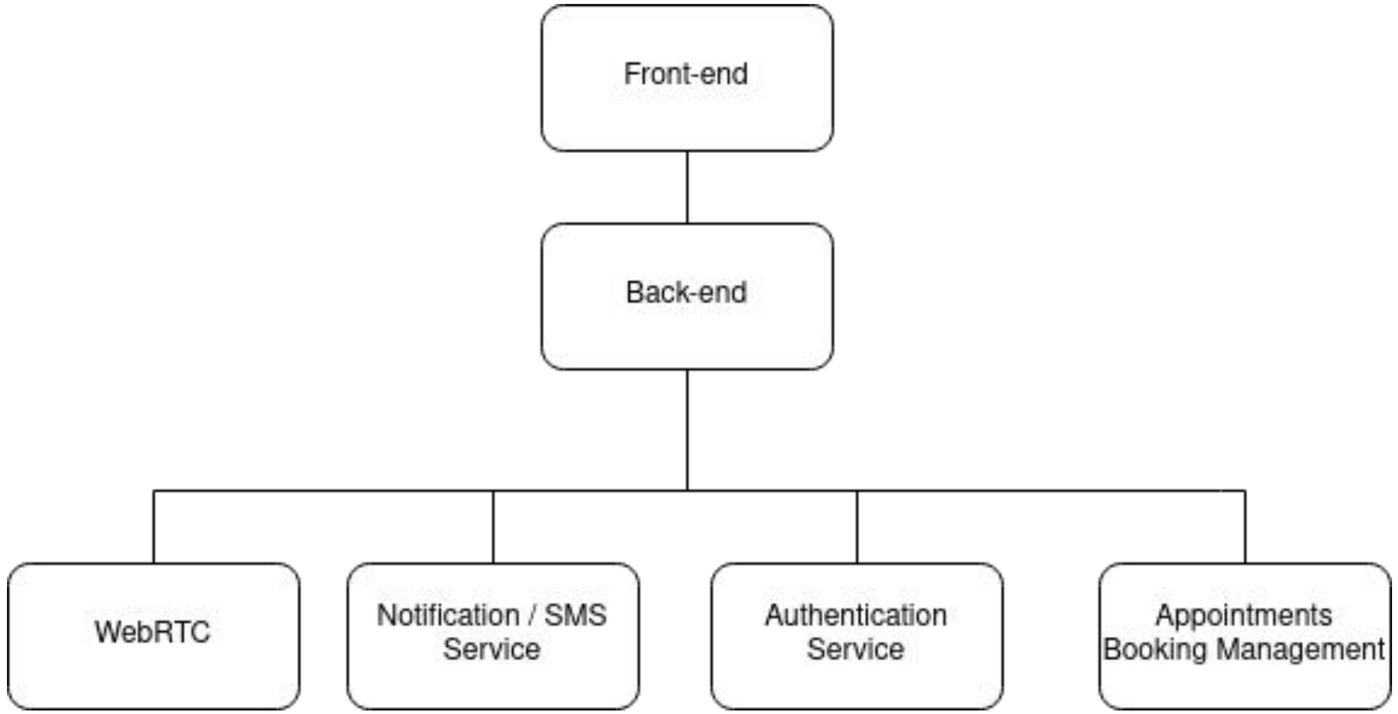
- Fast contact with the doctor
- Better management of the medical resources
- Centralized health information, as vaccines, medication, appointment historic, medical exams
- Simple and fast way to schedule a medical appointment, reducing the wait time



02. DEVELOPMENT

THE SERVICES

ARCHITECTURE



Notification Service

The notification service provides a simple way to send email and text messages using external providers

AWS SES vs AWS SNS

API for Notification Service

- AWS SNS - Simple Notification Service
  POST - /v1/notifications/sms: responsible to send SMS to the user

- AWS SES - Simple Email Service
  POST - /v1/notifications/email: responsible to send EMAIL to the user

API for Notification Service(continuation)

POSTMAN

```
{
            sender: "egs-notify@example.xyz",
            recipients: ["teste@gmail.com"],
            subject: "Test Subject",
            body: `<img alt="Embedded Image" src="cid:pain.png"/>`,
            attachments: [
            {
            attachment_name: "pain.ics",
            attachment_data: calendar,
            attachment_mime: "text/calendar",
            },
            {
            attachment_name: "pain.png",
            attachment_data: data,
            attachment_mime: "image/png",
            },
            ],
}
```

EMAIL API test

```
{
            "msg_body":"EGS SMS NOTIFY",
            "send_to":"+351xxxxxx"
}
```

SMS API test

Authentication Service

The authentication service provides third party authentication from Google and Facebook

API for Authentication Service

- Homepage
  GET - /login: home page

- Google Provider
  GET - /auth/google: redirects to Google's external authentication
  GET - /google/auth/callback: obtains user data through the authorized URL

- Facebook Provider
  GET - /auth/facebook: redirects to Facebook's external authentication
  GET - /facebook/auth/callback: obtains user data through the authorized URL

WebRTC - Node.js & PeerJS

PeerJS simplifies WebRTC peer-to-peer communication with an easy-to-use interface.

PEERJS

API for WebRTC

- GET - v1/video-call: request a video call ID
- POST - v1/{id}: join a call using a specific id

Appointment Service

The appointment service provides a way to simply manage appointments with multiple participants. It also exports them into the widely used iCal format.

iCal.NET Microsoft® SQL Server® Entity Framework Core ASP.net

Appointments Service

- GET - /v1/appointments: searches for appointments using several criteria
- POST - /v1/appointments: creates a new appointment
- GET - /v1/appointments/{appointment_id}: gets an appointment by id
- PUT - /v1/appointments/{appointment_id}: updates an appointment by id
- DELETE - /v1/appointments/{appointment_id}: deletes an appointment by id

Backend Service

The backend service acts as an orchestrator for other services providing a unique straightforward API to interface with the platform.

Entity Framework Core

Microsoft® SQL Server®

ASP.net

Backend

- GET - /v1/login: redirects to the authentication service
- GET - /v1/self: gets the logged in user data
- POST - /v1/doctors: register a new doctor user
- GET - /v1/doctors: searches doctors based on several criteria
- GET - /v1/doctors/{doctor_id}: gets a doctor by id
- PUT - /v1/doctors/{doctor_id}: updates a doctor by id
- POST - /v1/patients: register a new patient user
- GET - /v1/patients/{patient_id}: gets a patients by id
- PUT - /v1/patients/{patient_id}: updates a patients by id
- GET - /v1/appointments: gets appointments for a user
- POST - /v1/appointments: created a new appointment
- PUT - /v1/appointments/{appointment_id}: updates an appointment by id

Frontend

- Gathering of all the services above, with a simplified User Interface
- Pages:
  - Homepage
  - Log In / Sign In
  - Register (Doctor / Patient)
  - Create Appointment
  - Profile (Doctor / Patient)

React

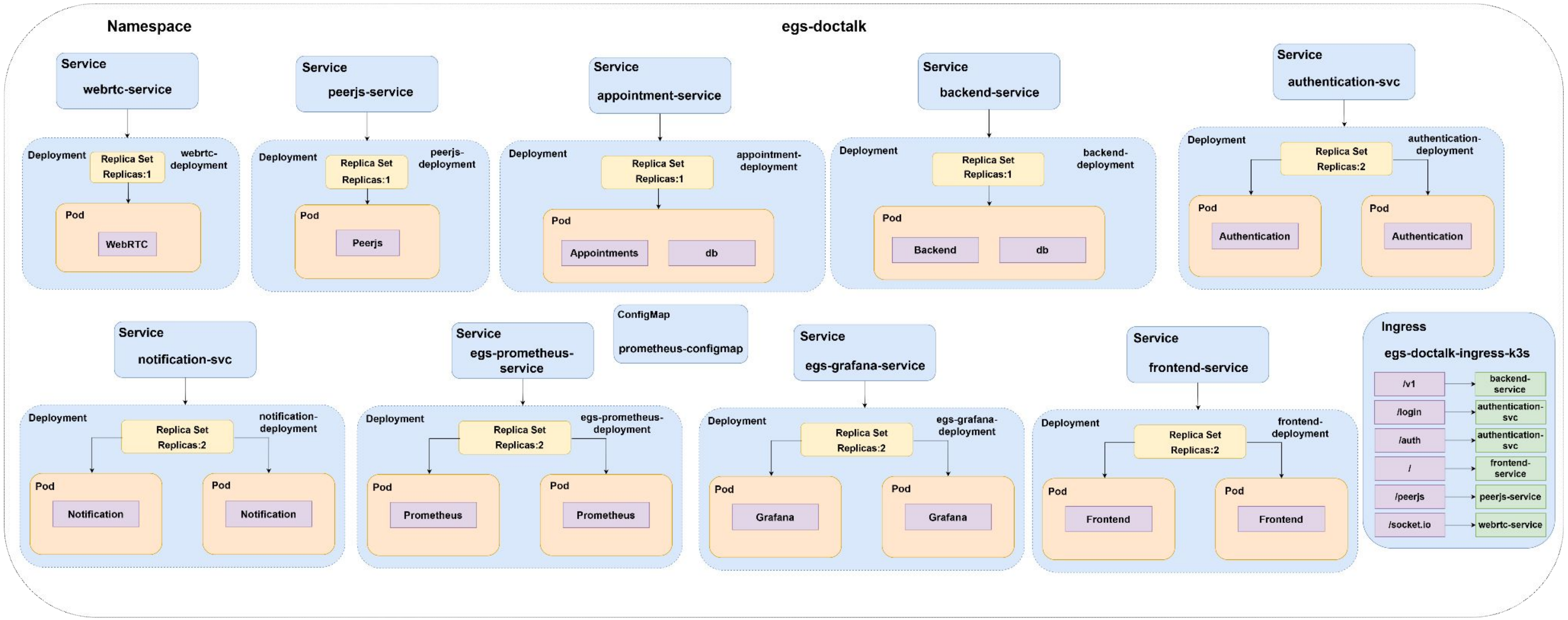
03.

DEPLOYMENT

Deployment

- Every service was initially developed and tested locally
- Once every service was working we created a docker container for each and deployed them all with docker-compose
- The final step was to move the deployment to kubernetes

Full Deployment Diagram

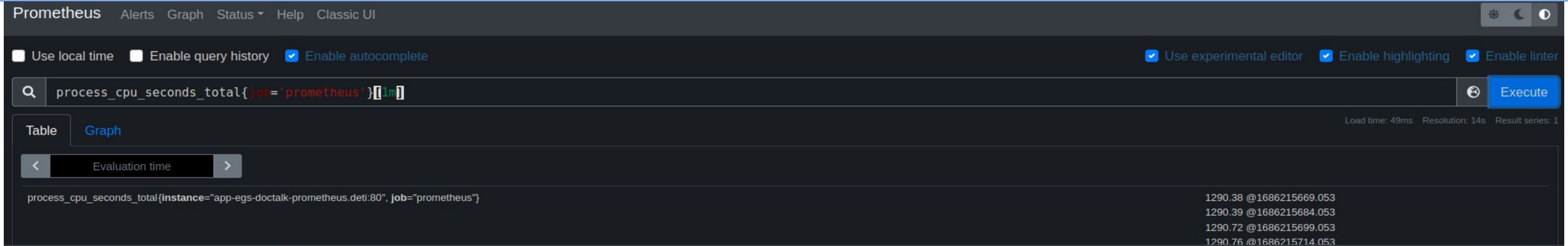


04.

DIFFICULTIES

Difficulties

- Setting up a prometheus exporter in every service
- Making prometheus detect service's health
- Consequently, grafana does not display dashboard metrics because there's no data being sent by prometheus
- Contacting external services from kubernetes pods due to internal name resolution



- [ ] Use local time
- [ ] Enable query history
- [x] Enable autocomplete
- [x] Use experimental editor
- [x] Enable highlighting
- [x] Enable linter

```
process_cpu_seconds_total{job='prometheus'}[1m]
```

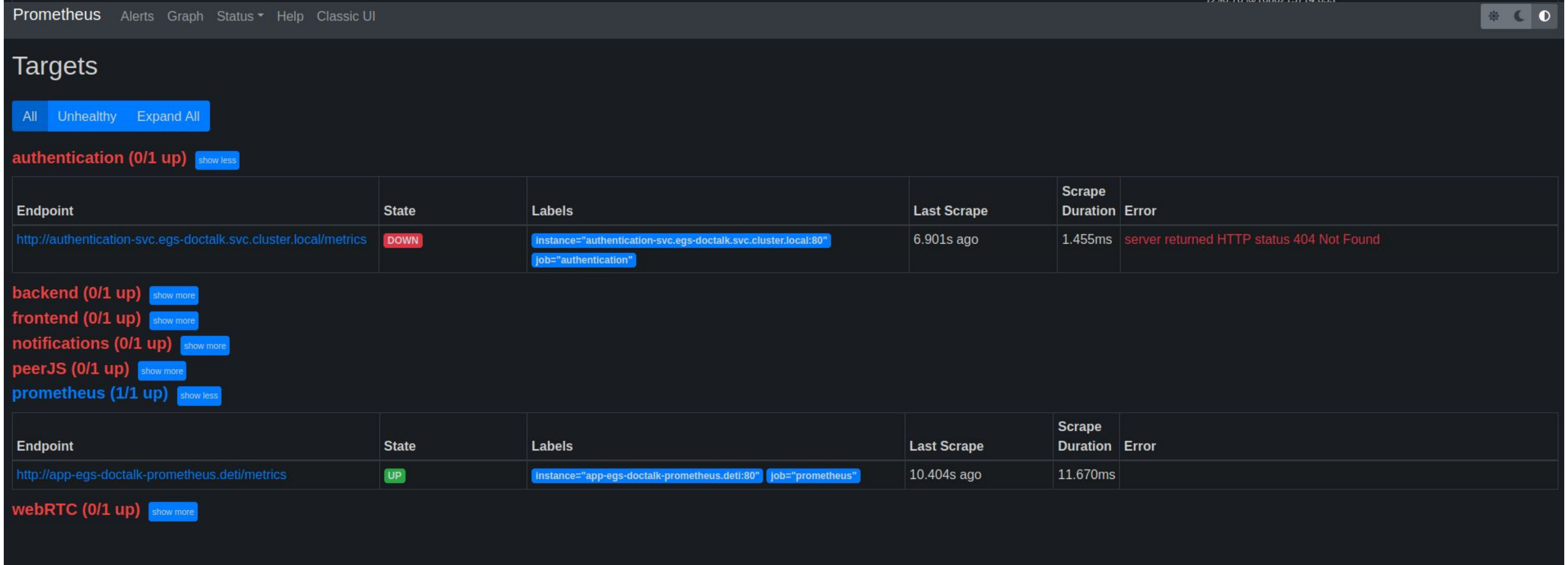
Execute

Load time: 49ms Resolution: 14s Result series: 1

Table | Graph

Evaluation time

```
process_cpu_seconds_total{instance="app-egs-doctalk-prometheus.deti:80", job="prometheus"}    1290.38 @1686215669.053
                                                                                              1290.39 @1686215684.053
                                                                                              1290.72 @1686215699.053
                                                                                              1290.76 @1686215714.053
```



Targets

authentication (0/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://authentication-svc.egs-doctalk.svc.cluster.local/metrics	DOWN	instance="authentication-svc.egs-doctalk.svc.cluster.local:80" job="authentication"	6.901s ago	1.455ms	server returned HTTP status 404 Not Found

backend (0/1 up) show more

frontend (0/1 up) show more

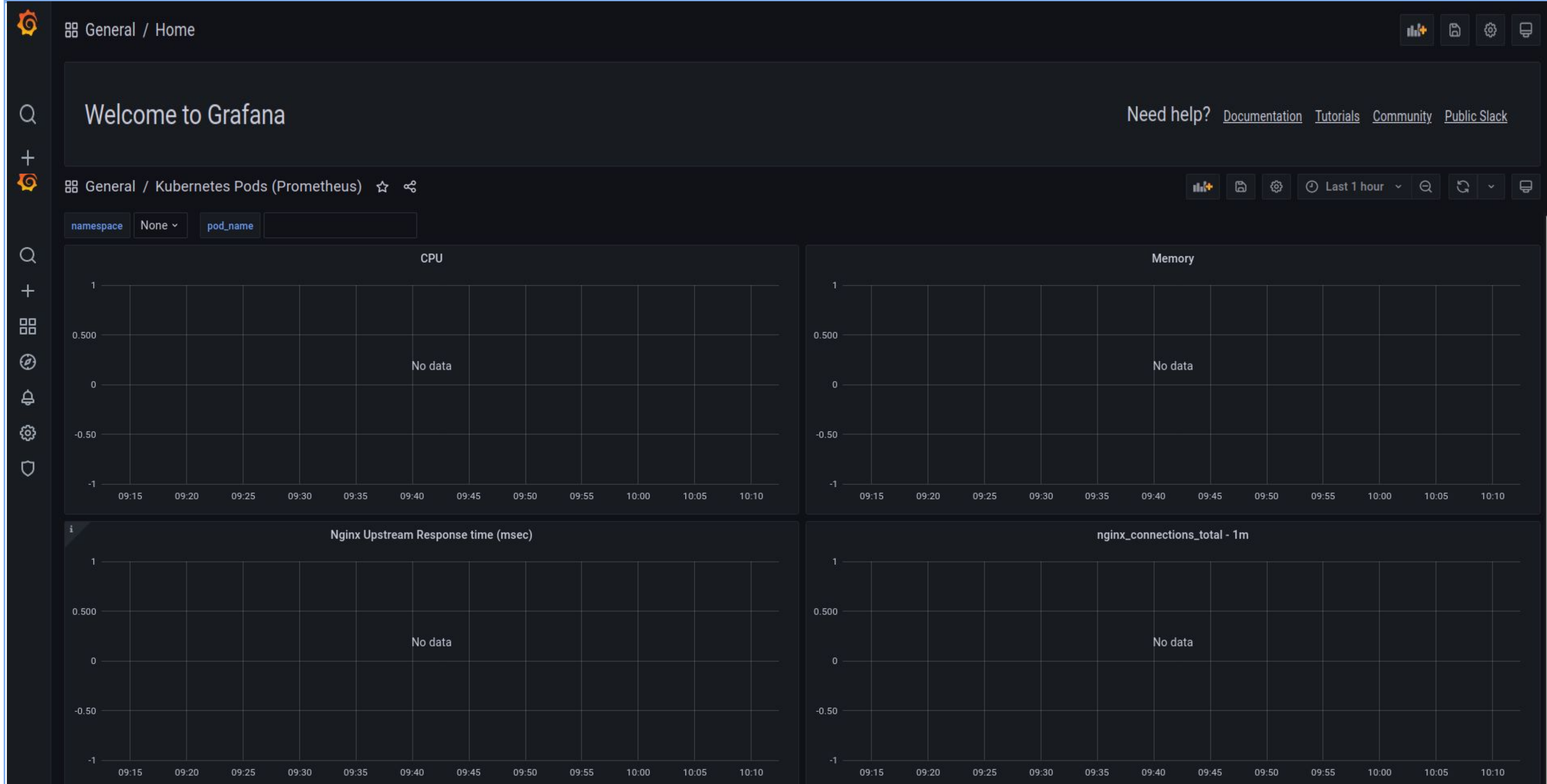
notifications (0/1 up) show more

peerJS (0/1 up) show more

prometheus (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://app-egs-doctalk-prometheus.deti/metrics	UP	instance="app-egs-doctalk-prometheus.deti:80" job="prometheus"	10.404s ago	11.670ms	

webRTC (0/1 up) show more


General / Home
Welcome to Grafana
Need help? Documentation Tutorials Community Public Slack
General / Kubernetes Pods (Prometheus)
Last 1 hour
namespace None
pod_name
CPU
No data
Memory
No data
Nginx Upstream Response time (msec)
No data
nginx_connections_total - 1m
No data

05. DEMONSTRATION

THANK YOU FOR YOUR ATTENTION

Do you have any questions?

Repository

[GitHub Organization]