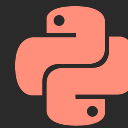


OpenTelemetry na prática

Live de Python #267



1. Como vamos fazer?

Pensando um pouco antes

2. Implementação do código

O básico necessário com FastAPI + SQLAlchemy

3. Instrumentação

Colocando a telemetria para rodar

4. O que der tempo

O céu é o limite xD

Título	Data
Observabilidade - Uma introdução	11/03
Observabilidade - Métricas (Opentelemetry / Prometheus)	25/03
Observabilidade - Rastreamento / Tracing (Opentelemetry / Tempo / Jeager)	08/04
Observabilidade - Logs (Opentelemetry / Loki)	15/04
OpenTelemetry + Python na prática	Agora!



Isso é uma série! – Playlist na descrição!





apoia.se/livedepython



pix.dunossauro@gmail.com



patreon.com/dunossauro



Ajude o projeto <3



Ademar Peixoto, Adriano Casimiro, Alexandre Harano, Alexandre Lima, Alexandre Takahashi, Alexandre Villares, Alfredo Braga, Alisson Souza, Allan Kleitson, Alysso Oliveira, Andre Paula, Antônio Filho, Apc 16, Aslay Clevisson, Aurelio Costa, Ber_dev_2099, Bernarducs, Bruno420g, Bruno Almeida, Bruno Barcellos, Bruno Batista, Bruno Freitas, Bruno Guizi, Bruno Lira, Bruno Ramos, Bruno Santos, Caio Nascimento, Carlos Ramos, Christian Fischer, Clara Battesini, Controlado, Cristian Firmino, Daniel Aguiar, Daniel Bianchi, Daniel Real, Daniel Wojcickoski, Danilo Boas, Danilo Silva, David Couto, David Kwast, Davi Souza, Dead Milkman, Denis Bernardo, Dgeison Peixoto, Diego Guimarães, Dino, Douglas Oliveira, Edgar, Eduard0ml, Elton Guilherme, Emerson Rafael, Ennio Ferreira, Erick Andrade, Érico Andrei, Everton Silva, É X, Fabio Barros, Fábio Barros, Fabiokleis, Fabio Valente, Fabricio Biazzotto, Felipe Augusto, Felipe Rodrigues, Fernanda Prado, Fernandocelmer, Filipe Monteiro, Frederico Damian, Fsouza, Gabrieimoreira, Gabriel Espindola, Gabriel Mizuno, Gabriel Paiva, Gabriel Ramos, Gabriel Simonetto, Geilton Cruz, Giovanna Teodoro, Giuliano Silva, Guibeira, Guilherme, Guilherme Felitti, Guilherme Gall, Guilherme Ostrock, Guilherme Piccioni, Guilherme Silva, Gustavo Suto, Haelmo Almeida, Harold Gautschi, Heitor Fernandes, Helvio Rezende, Henrique Andrade, Henrique Machado, Henrique Sebastião, Hiago Couto, Igor Taconi, Jairo Lenfers, Janael Pinheiro, Jean Victor, Jefferson Antunes, Jlx, Joelson Sartori, Jonatas Leon, Jônatas Silva, Jorge Silva, Jose Barroso, Jose Edmario, José Gomes, Josefto Júnior, Jose Mazolini, Jose Terra, Josir Gomes, Jrborba, Juan Felipe, Juliana Machado, Julio Batista-silva, Julio Franco, Júlio Gazeta, Júlio Sanchez, Kaio Peixoto, Leandro Silva, Leandro Vieira, Leonan Ferreira, Leonardo Mello, Leonardo Nazareth, Lucas Carderelli, Lucas Lattari, Lucas Mello, Lucas Mendes, Lucas Nascimento, Lucas Schneider, Lucas Simon, Luciano Filho, Luciano Ratamero, Luciano Teixeira, Luis Leite, Luiz Carlos, Luiz Duarte, Luiz Lima, Luiz Paula, Mackilem Laan, Marcelo Araujo, Marcelo Silva, Marcio Freitas, Marcio Novaes, Marcio Silva, Marco Gandra, Marco Mello, Marcos Almeida, Marcos Gomes, Marcos Oliveira, Maria Gabriela, Marina Passos, Mateus Lisboa, Matheus Silva, Matheus Vian, Mírian Batista, Mlevi Lsantos, Murilo Carvalho, Murilo Viana, Natália Araújo, Nathan Branco, Ocimar Zolin, Otávio Carneiro, Pedro Henrique, Peterson Santos, Philipe Vasconcellos, Phmmdev, Pytonyc, Rafael, Rafael Araújo, Rafael Faccio, Rafael Lino, Rafael Lopes, Rafael Romão, Raimundo Ramos, Ramayana Menezes, Renan, Renan Sebastião, Renato José, Renato Moraes, Rene Pessoto, Renne Rocha, Ricardo Combat, Ricardo Silva, Ricardo Viana, Richard Costa, Rinaldo Magalhaes, Riverfount, Rjribeiro, Rodrigo Barretos, Rodrigo Oliveira, Rodrigo Quiles, Rodrigo Santana, Rodrigo Vaccari, Rodrigo Vieira, Rogério Nogueira, Rui Jr, Samanta Cicilia, Santhiago Cristiano, Selmison Miranda, Shinodinho, Téo Calvo, Terramotta, Thiago Araujo, Thiago Borges, Thiago Lucca, Thiago Paiva, Tiago Ferreira, Tiago Henrique, Tony Dias, Tyrone Damasceno, Valdir, Valdir Tegon, Varlei Menconi, Vinicius Bego, Viniciusccosta, Vinicius Silva, Vinicius Stein, Vitor Silva, Vladimir Lemos, Wagner Gabriel, Washington Teixeira, Williangl, Willian Lopes, Wilson Duarte, Zeca Figueiredo, Zecarlos Junior



Obrigado você



Uma explicação
básica da
arquitetura

O que
farem
os?

Reduzindo o escopo ao máximo

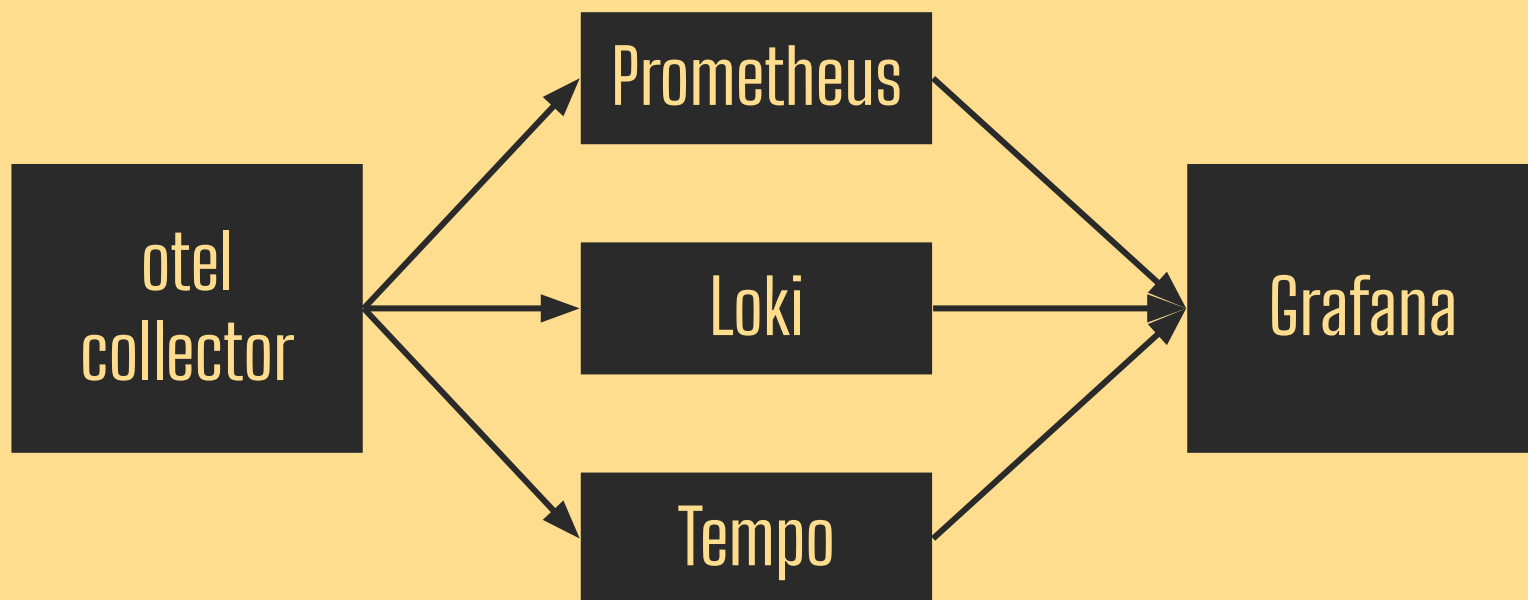


Aplicação

otel/lgtm

PostgreSQL

Containers



otel/lgtm

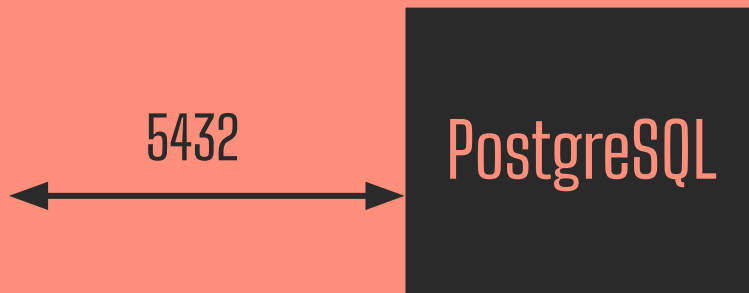




```
1 services:
2   otel:
3     image: grafana/otel-lgtm
4     ports:
5       - 3000:3000 # grafana
6       - 4317:4317 # otel
```

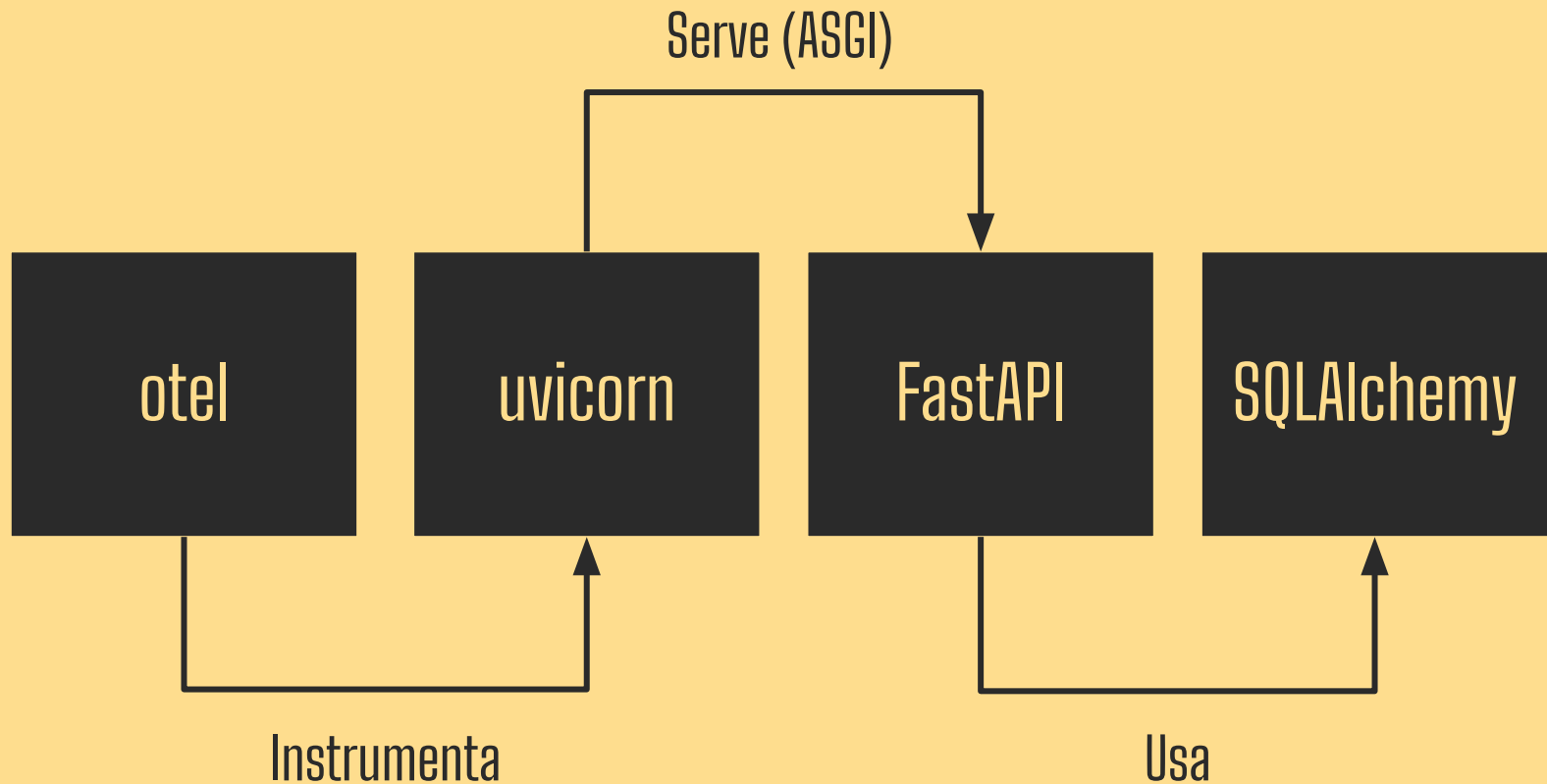
compose.yaml

PostgreSQL



```
postgres:
  image: postgres
  volumes:
    - pgdata:/var/lib/postgresql/data
  environment:
    POSTGRES_USER: app_user
    POSTGRES_DB: app_db
    POSTGRES_PASSWORD: app_password
  ports:
    - "5432:5432"
```

Aplicação



Aplicação



Serve (ASGI)

otel

uvicorn

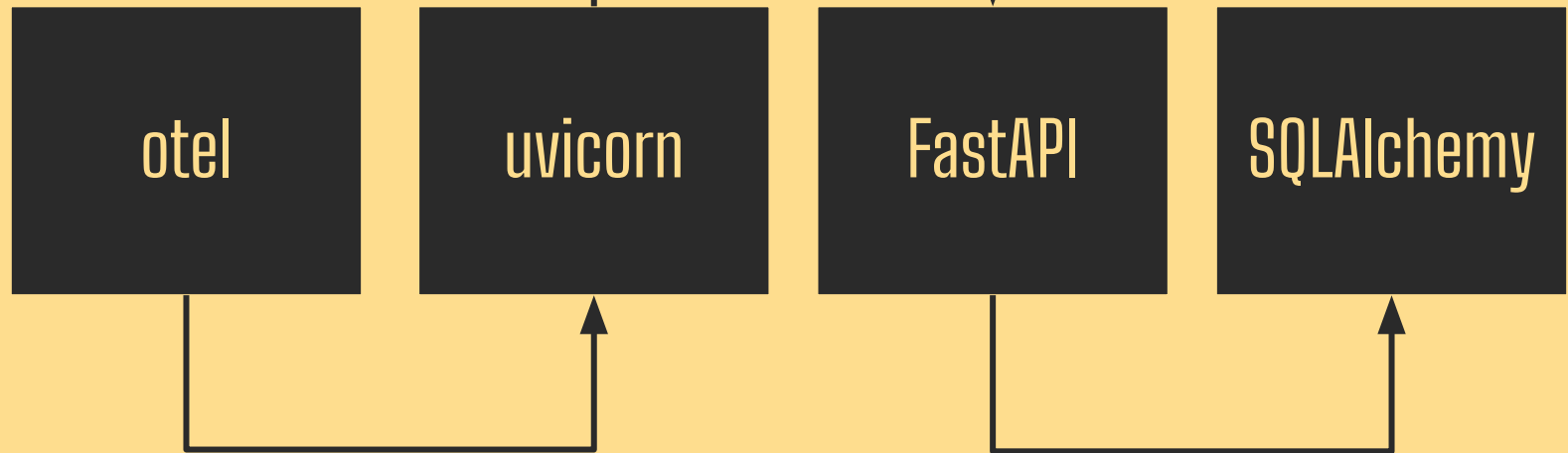
FastAPI

SQLAlchemy

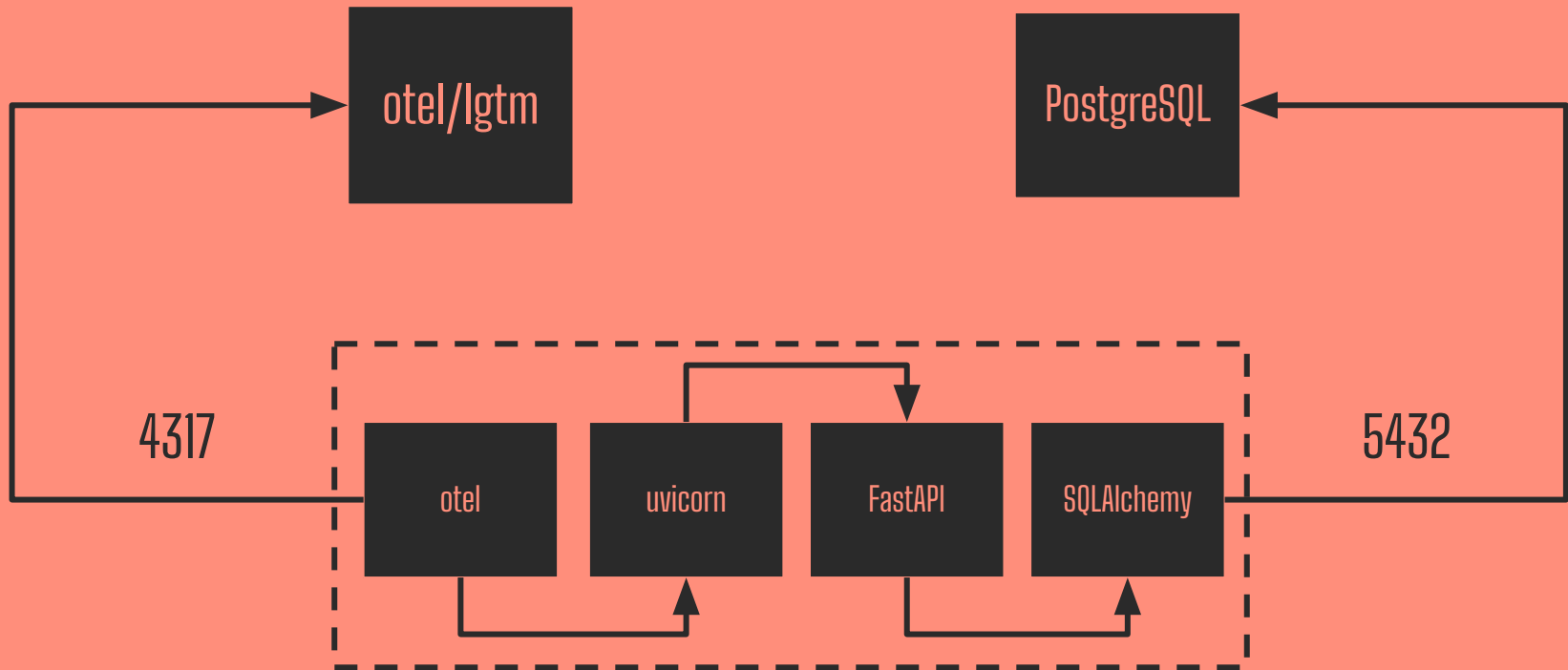
Instrumenta

Usa

Container



Conexões



Container

Bora botar a mão no
código!

Aplica
ção

Um olá mundo!



Vamos fazer a configuração mais básica possível antes de tudo!

1. Instalações
2. Inicialização
3. Instrumentação

Ambiente e instalações



```
$ poetry new Live267  
$ poetry add fastapi uvicorn sqlalchemy  
$ poetry add --group dev poethepoet
```

Iniciando o projeto



```
from fastapi import FastAPI

app = FastAPI()

@app.get('/check')
def check():
    return {'message': 'OK'}
```

Instalando os componentes para instrumentação



```
$ poetry add --group otel \  
    opentelemetry-distro  
    opentelemetry-exporter-otlp
```

Variáveis de ambiente



```
OTEL_SERVICE_NAME=app
OTEL_EXPORTER_OTLP_ENDPOINT=localhost:4317
OTEL_EXPORTER_OTLP_INSECURE=true
OTEL_PYTHON_LOGGING_AUTO_INSTRUMENTATION_ENABLED=true
OTEL_TRACES_EXPORTER=otlp
OTEL_METRICS_EXPORTER=otlp
OTEL_LOGS_EXPORTER=otlp
```

Simplificando a vida com o Edgar

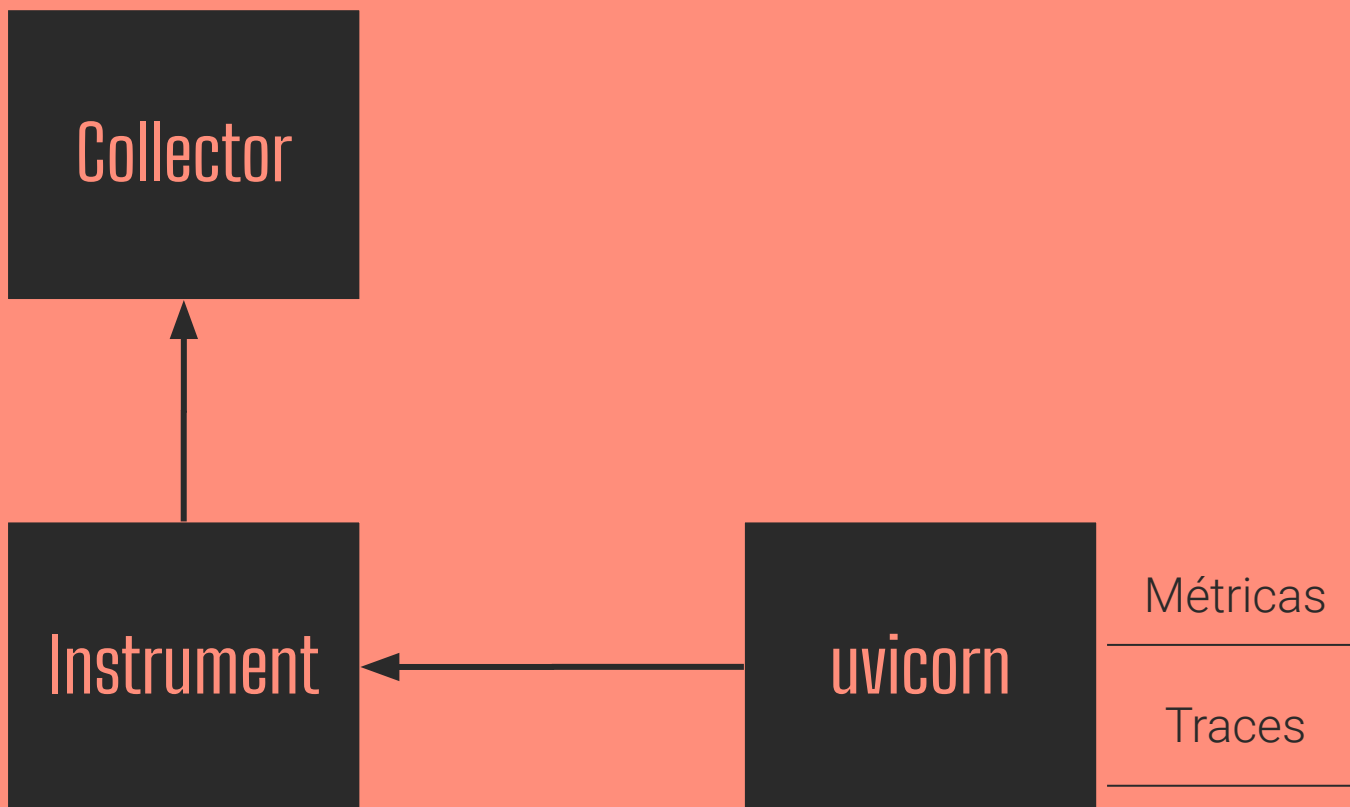
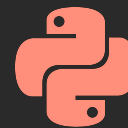


```
[tool.poe]
envfile = '.env'

[tool.poe.tasks.serve]
shell = 'opentelemetry-instrument uvicorn live267.app:app'

[tool.poe.tasks]
otel-install = 'opentelemetry-bootstrap -a install'
```

Os efeitos práticos dessa configuração



Os efeitos práticos dessa configuração



E os logs?

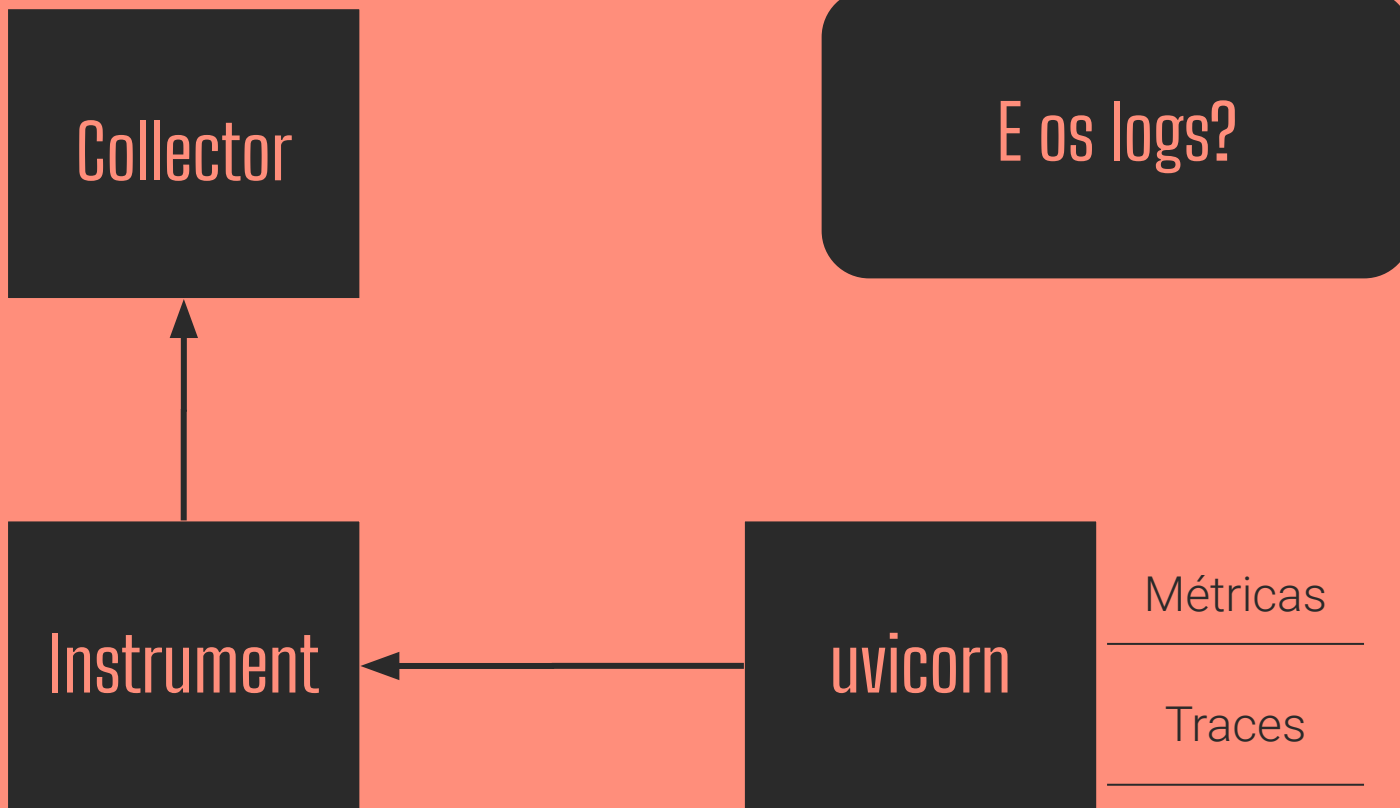
Collector

Instrument

uvicorn

Métricas

Traces



Colocando os logs para funcionar



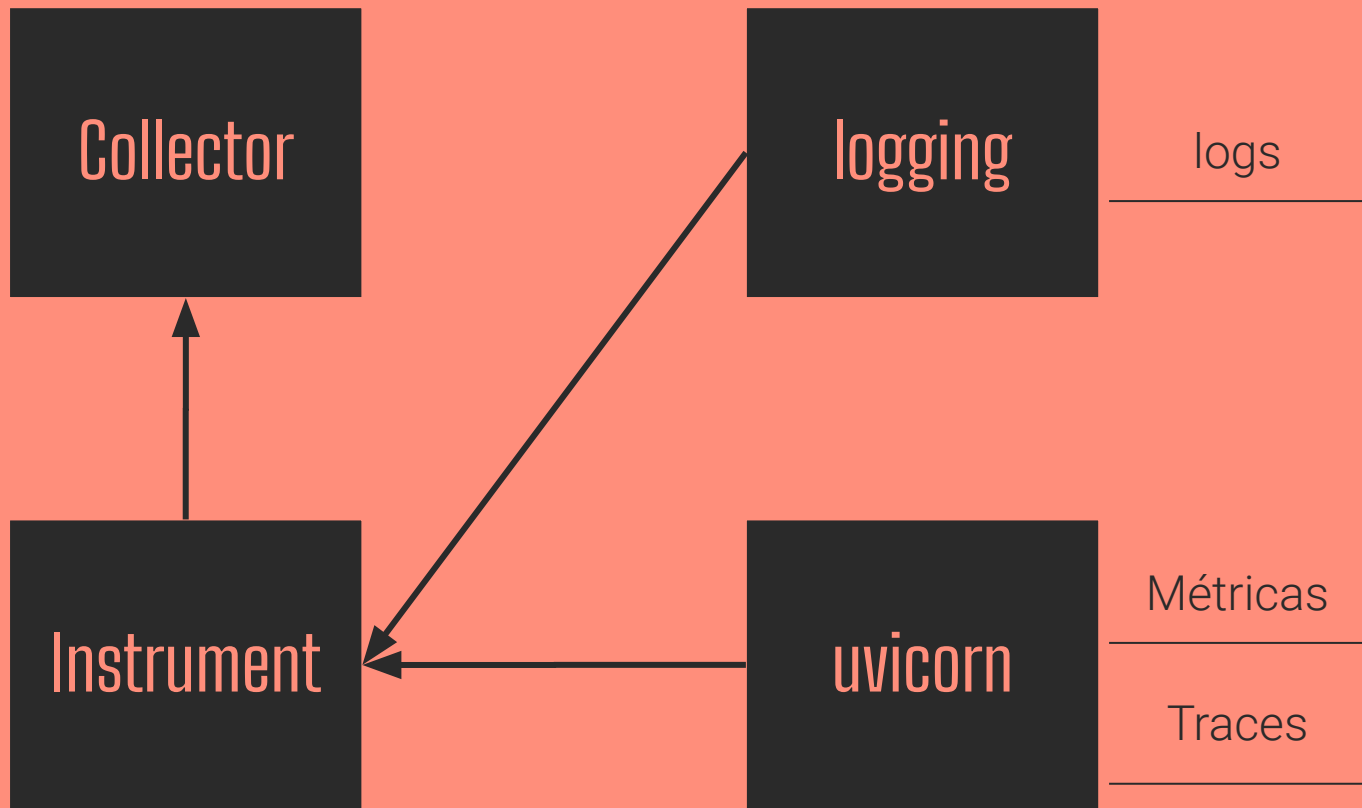
```
import logging
from contextlib import asynccontextmanager
from fastapi import FastAPI

logger = logging.getLogger()
logger.setLevel(logging.INFO)

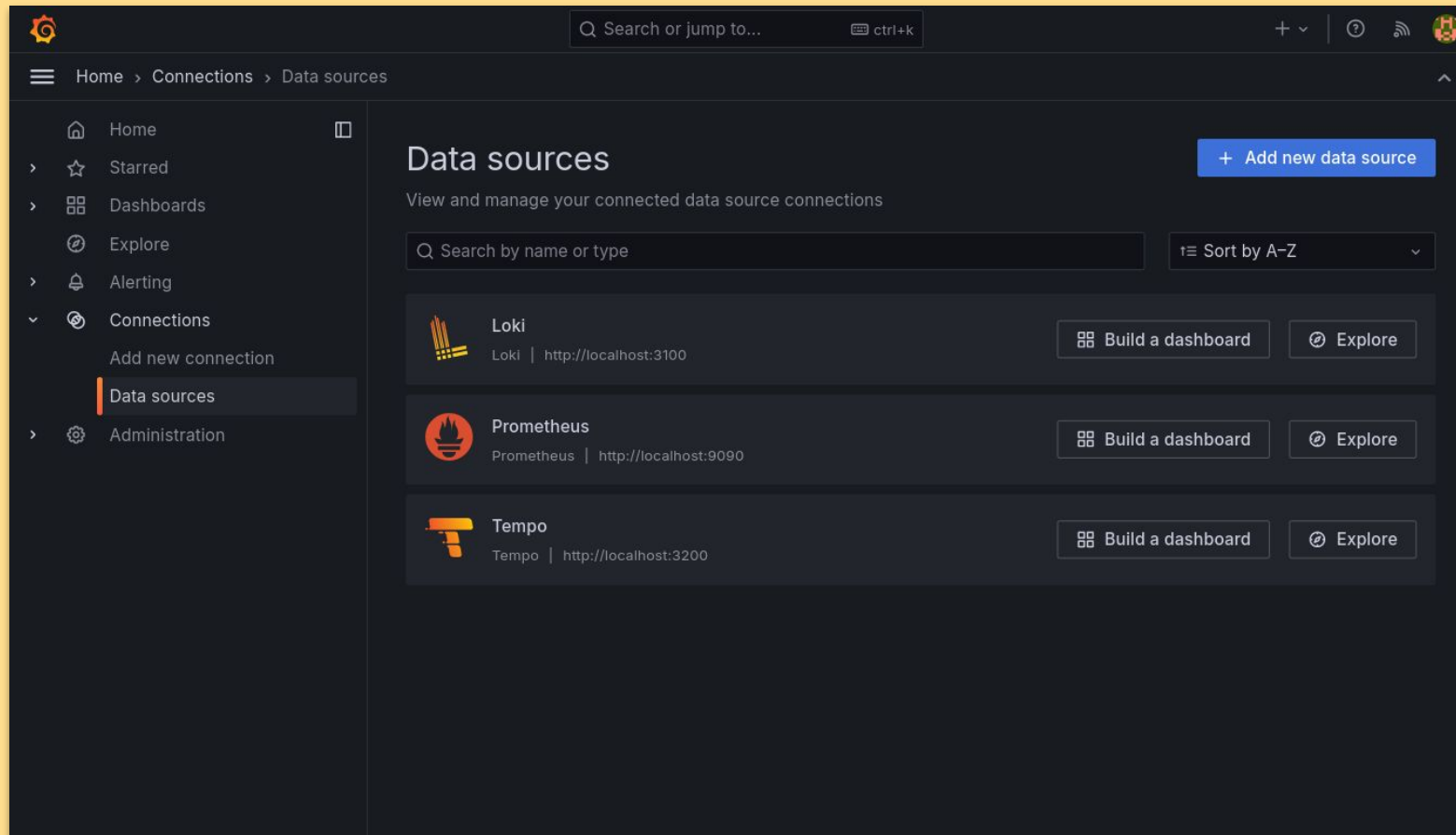
@asynccontextmanager
async def lifespan(app):
    logger.info('Iniciando aplicação!')
    yield
    logger.info('Finalizando aplicação!')

app = FastAPI(lifespan=lifespan)
```


Os efeitos práticos dessa configuração



Validando se a POC deu certo



The screenshot shows the Grafana web interface. The top navigation bar includes a search bar, a 'ctrl+k' shortcut, and user profile icons. The left sidebar contains a menu with 'Home', 'Starred', 'Dashboards', 'Explore', 'Alerting', 'Connections', 'Add new connection', 'Data sources' (highlighted), and 'Administration'. The main content area is titled 'Data sources' and includes a subtitle 'View and manage your connected data source connections'. A search bar and a 'Sort by A-Z' dropdown are present. Three data sources are listed: 'Loki' (http://localhost:3100), 'Prometheus' (http://localhost:9090), and 'Tempo' (http://localhost:3200). Each entry has 'Build a dashboard' and 'Explore' buttons.

Name	URL	Actions
Loki	http://localhost:3100	Build a dashboard, Explore
Prometheus	http://localhost:9090	Build a dashboard, Explore
Tempo	http://localhost:3200	Build a dashboard, Explore

localhost:3000

Agora o céu é o limite

(ou então o período marcado de 2 horas de live xD)



- **Métricas**

- **Quantidade de requests:** `http_server_duration_milliseconds_count`
- **Requests ativos:** `http_server_active_requests`
- **Tempo das requisições:**
 - **piores 95%:** `histogram_quantile(0.95, sum(rate(http_server_duration_milliseconds_bucket[$__rate_interval]) by (le))`

- **Traces**

- **Erros:** `{resource.service.name="app" && status=error}`

- **Logs**

- **Volume:** `sum by(level) (rate({job="app"} | json [5m]))`



apoia.se/livedepython



pix.dunossauro@gmail.com



patreon.com/dunossauro



Ajude o projeto <3

