Если 500 ошибка, то провреить поднялся ли контейнер командой docker ps. Если не поднялся, то посмотреть логи контейнера, падает ли ошибка  
В Sentry смотримлог ошибки. Если дело в редисе, то перезапускаем его  
**docker stop окружение-redis-1  
docker start окружение-redis-1**

Если дело в БД, то перезапускае ее **sudo systemctl restart postgresql**

Если не работает какая то функциональность (не отсылается запрос), то узнать как она работает – по REST или WebSocket, что бы знать что отвалилось

**Зачастую каждый микросервис развернут в docker контейнере.**

Что бы посмотреть что данные передаются из одного микросервиса в другой, то можно посмотреть логи этих двух микросервисов

**docker logs -f –tail=100 {container\_name}**

И проверить что верные данные передаются

**Так же можно зайти в Sentry (система ловли эксепшенов), где будет видна эта ошибка, с указанием места в коде, где она произошла**

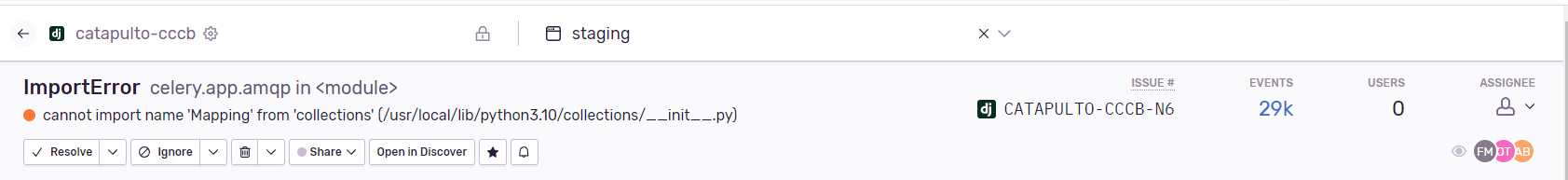
По сути это те же логи, но только уже показано место в коде, где она произошла (**Exception Monitoring**)

1. Заходим в Issues > My Projects > выбираем нужный компонент

* Catapulto-front (Front backend)
* Catapulto-cs (Back backend)
* Catapulto-react (Frontend)

1. Заходим в Issues > All Environments > выбираем окружение где производили тестирование

* test a / b / c / d
* staging
* prod
* dev



**Так же можно зайти в админку каждого микросервиса и посмотреть его дамп**

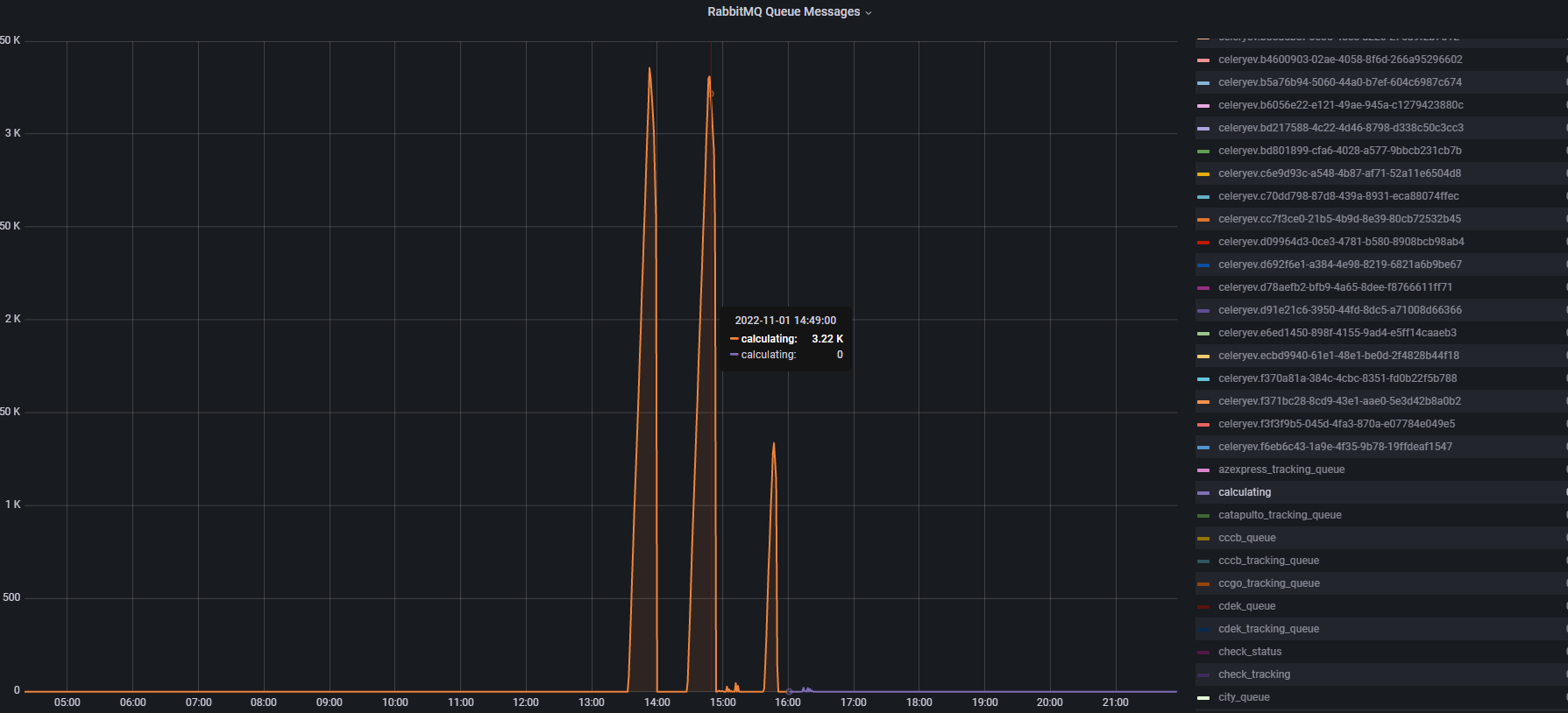
**Так же можно подключиться к БД и посмотреть что нужные сущности создались/удалилились и тд**

**Так же нужно бывает смотреть Redis Cache**

Redis > **docker exec -it staging\_redis\_1 sh**

* + **redis-cli** – переход в оснастку
  + **select 1** – выбор базы данных
  + **keys \*** - получить список всех закешированных записей (их ключей)
  + **get key\_name** – получить информацию по закешированной записи
  + **flushdb** – сбросить кэш
  + **ctrl + d** – вернуться на прошлый шаг

**Так же нужно бывает смотреть RabbitMQ**



Можно смотреть как рассасывается очередь

**Так же нужно бывает смотреть Django-RQ (Django Redis Queue)**

https://{env}api.ctplt.ru/django-rq/

**Так же нужно бывает смотреть Celery Worker**