

Министерство образования Республики Беларусь

Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет	компьютерных систем и сетей
Кафедра	информатики
Дисциплина	обеспечение качества программного обеспечения

Отчёт по лабораторной работе  
на тему

**AWS LAMBDA**

Группа:	858641
Магистранты:	А.С. Долматович Н.Р. Ровдо

МИНСК 2019

## СОДЕРЖАНИЕ

Введение .....	3
1 Детали реализации .....	3
1.1 Топология .....	3
1.2 Swagger .....	5
2 Нагрузочное тестирование .....	6
3 Код реализации.....	11
Список использованной литературы.....	12

## **Введение**

AWS Lambda – это сервис бессерверных вычислений, который запускает программный код в ответ на определенные события и отвечает за автоматическое выделение необходимых вычислительных ресурсов. AWS Lambda можно использовать для расширения возможностей других сервисов AWS с помощью специальной логики или для создания собственных серверных сервисов с применением возможностей масштабирования, производительности и безопасности AWS. AWS Lambda может автоматически запускать программный код в ответ на различные события, такие как HTTP-запросы через Amazon API Gateway, изменение объектов в корзине Amazon S3, обновление таблиц в Amazon DynamoDB или смена состояний в AWS Step Functions.

Lambda запускает код в высокопроизводительной вычислительной среде и занимается административной поддержкой всех ресурсов, включая обслуживание серверов и операционных систем, распределение производительности и автоматическое масштабирование, установку ПО и исправлений уязвимостей, а также мониторинг кода и ведение журналов. От вас требуется только предоставить программный код.

## **1 ДЕТАЛИ РЕАЛИЗАЦИИ**

### **1.1 Топология**

Для реализации API была выбрана топология с AWS Lambda + AWS Elastic Compute Cloud + AWS Elastic Load Balancer. AWS Lambda представляет собой точку входа в приложение. Lambda идеально подходит для создания программных интерфейсов. Есть различные способы использования функций Lambda и варианты организации бессерверной архитектуры — каждый сможет выбрать что-то подходящее с учетом поставленной цели. В данной топологии ELB распределяет нагрузку, EC2 выполняет бизнес-требования.

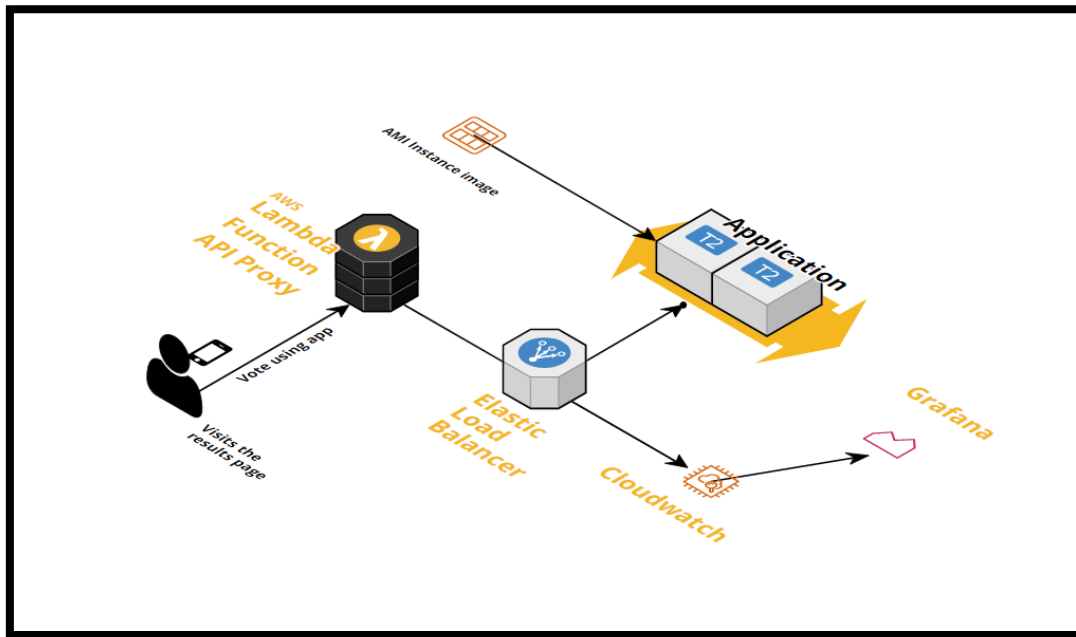


Рисунок 1 Топология приложения

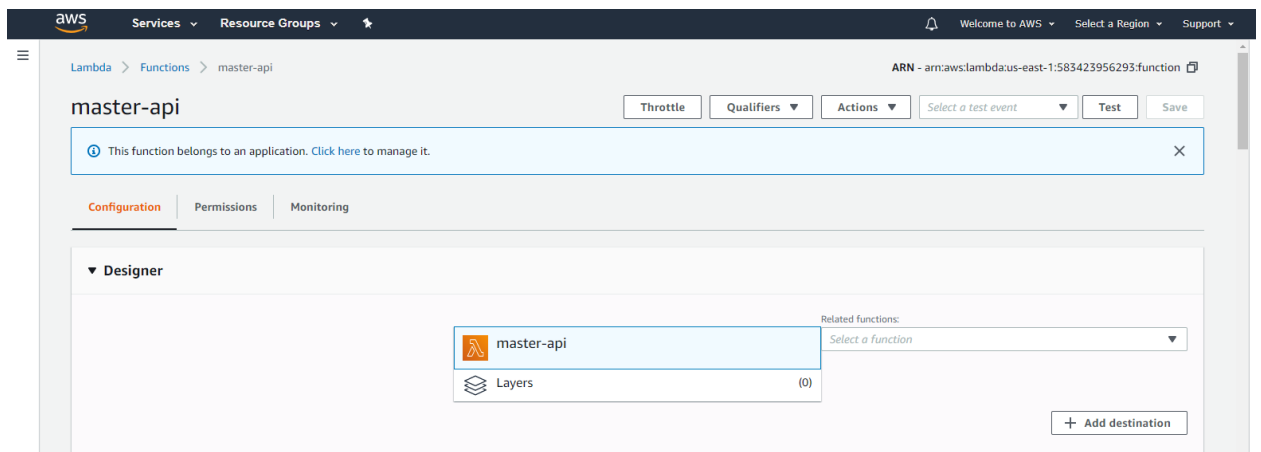


Рисунок 2 Lambda функция

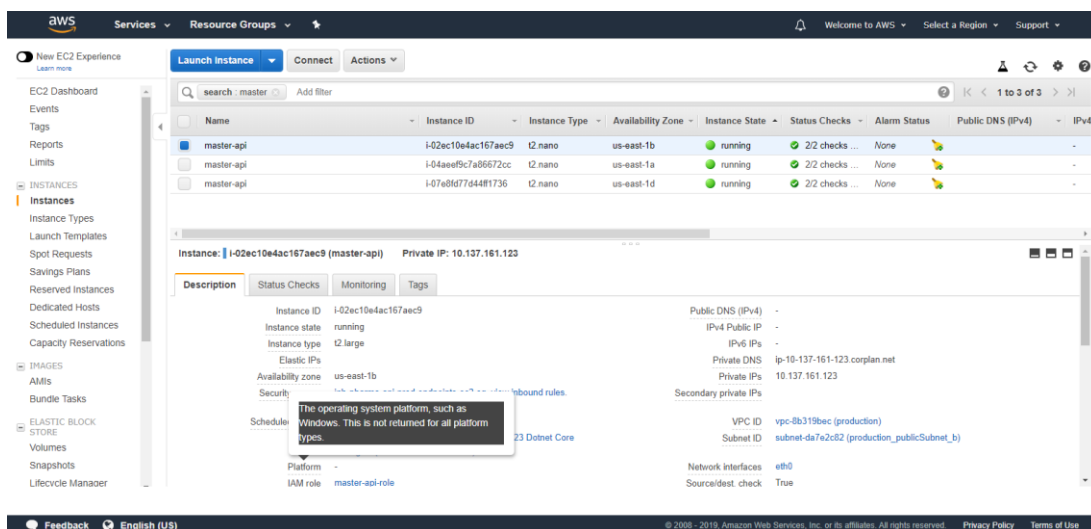


Рисунок 3 Развёрнутые машины в сервисе EC2

## 1.2 Swagger

Для автоматической генерации документации в приложения был подключен сервис Swagger.

Swagger – это фреймворк для спецификации RESTful API. Суть заключается в том, что он дает возможность не только интерактивно просматривать спецификацию, но и отправлять запросы.

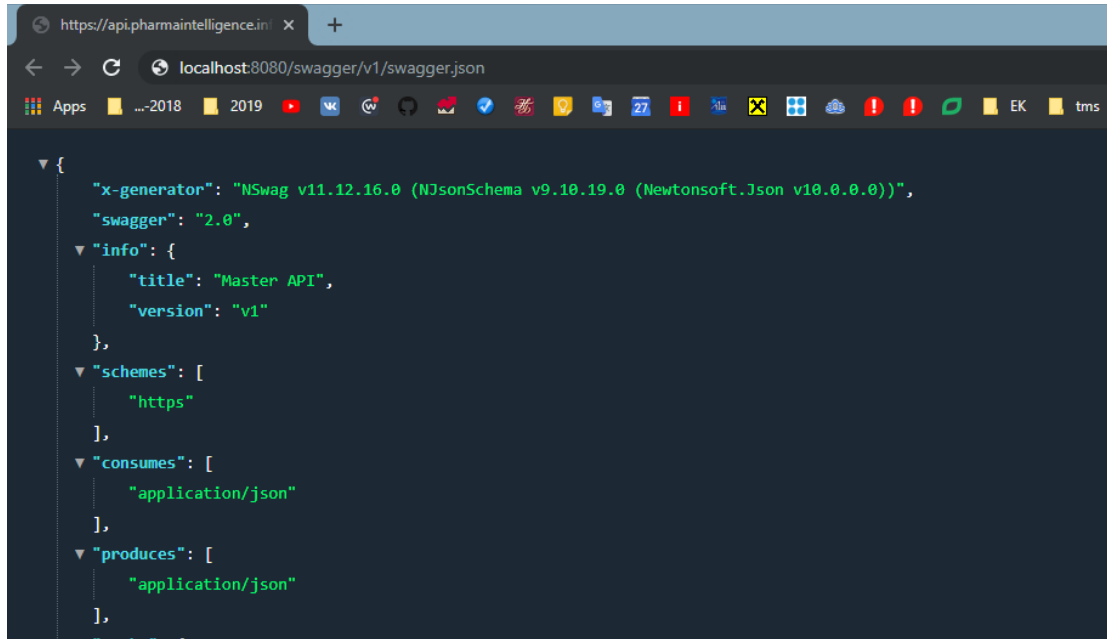


Рисунок 4 Результат работы Swagger

### Код подключения Swagger

```
public static class SwaggerExtensions
{
    public static void UseSwaggerBuilder(this IApplicationBuilder app)
    {
        var operationProcessorHook =
app.ApplicationServices.GetRequiredService<IOperationProcessor>();
        var documentProcessorHook =
app.ApplicationServices.GetRequiredService<IDocumentProcessor>();
        var deployment = app.ApplicationServices.GetRequiredService<IDeployment>();
        var pharmaRouteProvider =
app.ApplicationServices.GetRequiredService<IPharmaRouteProvider>();

        var securityDefinition = new SecurityDefinitionAppender(PharmaAuthSchemes.Jwt,
new SwaggerSecurityScheme
{
    Type = SwaggerSecuritySchemeType.ApiKey,
    Name = "Authorization",
    Description = "Copy 'Bearer ' and a valid JWT token into field",
    In = SwaggerSecurityApiKeyLocation.Header
});
        var securityScope = new
OperationSecurityScopeProcessor(PharmaAuthSchemes.Jwt);
    }
}
```

## 2 НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ

Для обеспечения мониторинга API был подключен CloudWatch. Amazon CloudWatch служит для мониторинга тех или иных параметров состояния сервера.

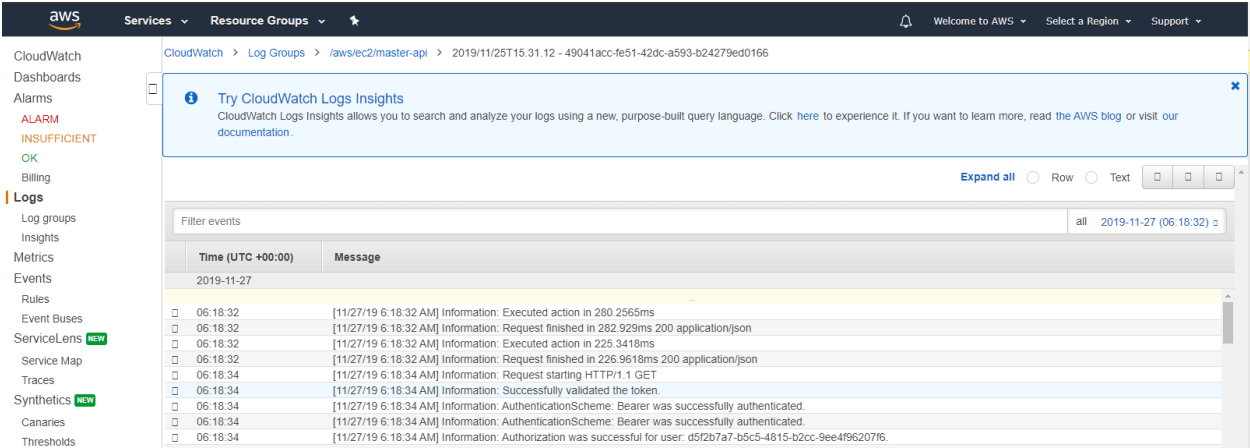


Рисунок 5 Логи Cloudwatch

Также к функции был подключен X-Ray, данный сервис используется для поиска ошибок и узких мест в приложении. В X-Ray доступны сегменты и аннотации, которые позволяют добавлять метаданные к отслеживаниям.



Рисунок 6 X-ray

Пример запроса в Postman:

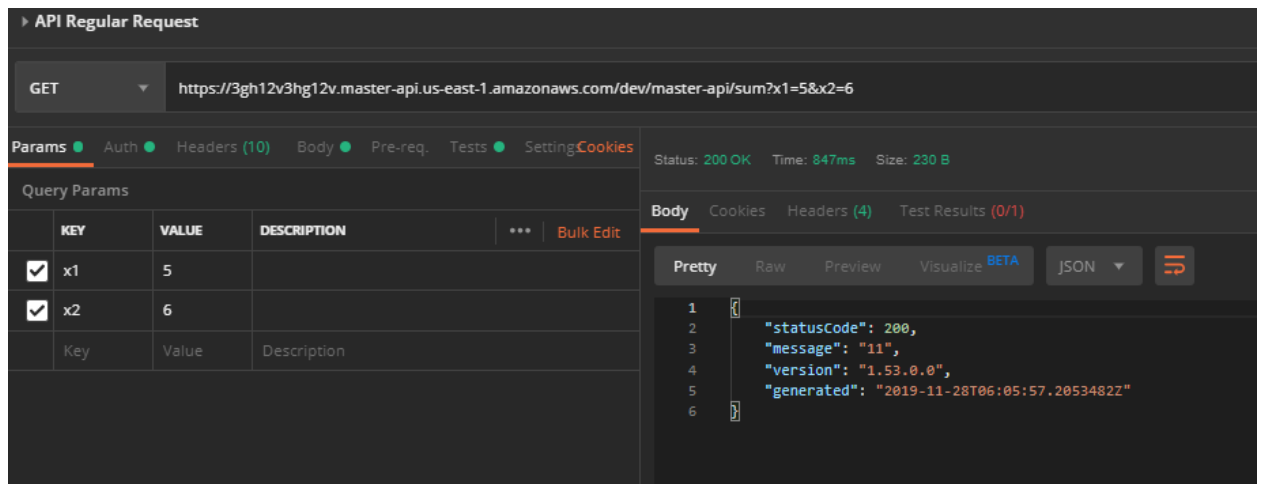


Рисунок 7 Пример использования приложения

Для нагрузочного тестирования использовался инструмент Apache Jmeter, так как он позволяет описывать запросы с помощью простого GUI, поддерживает выполнение нагрузочных тестов, распределенных по нескольким процессам и потокам, таким образом он может использоваться для симуляции большого количества запросов.



Рисунок 8 Логотип приложения JMeter

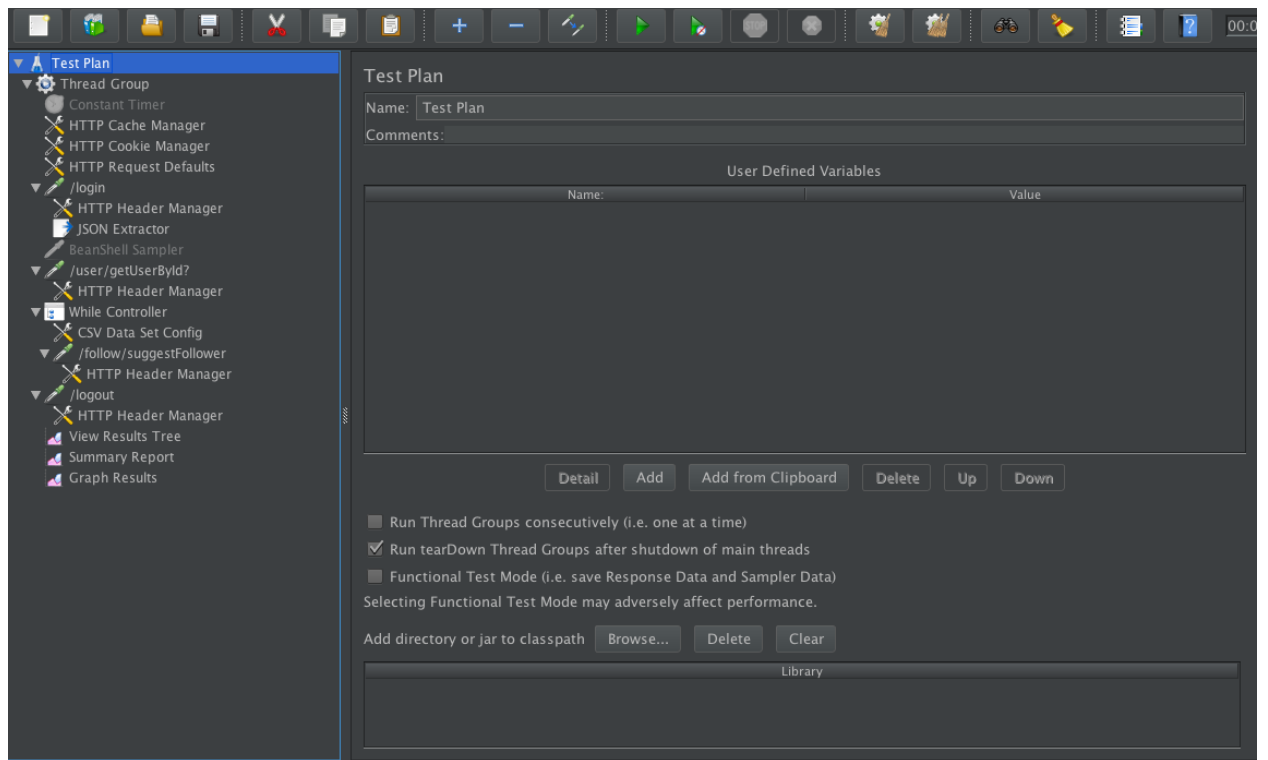


Рисунок 9 Рабочее пространство JMeter с загруженным проектом тестирования приложения

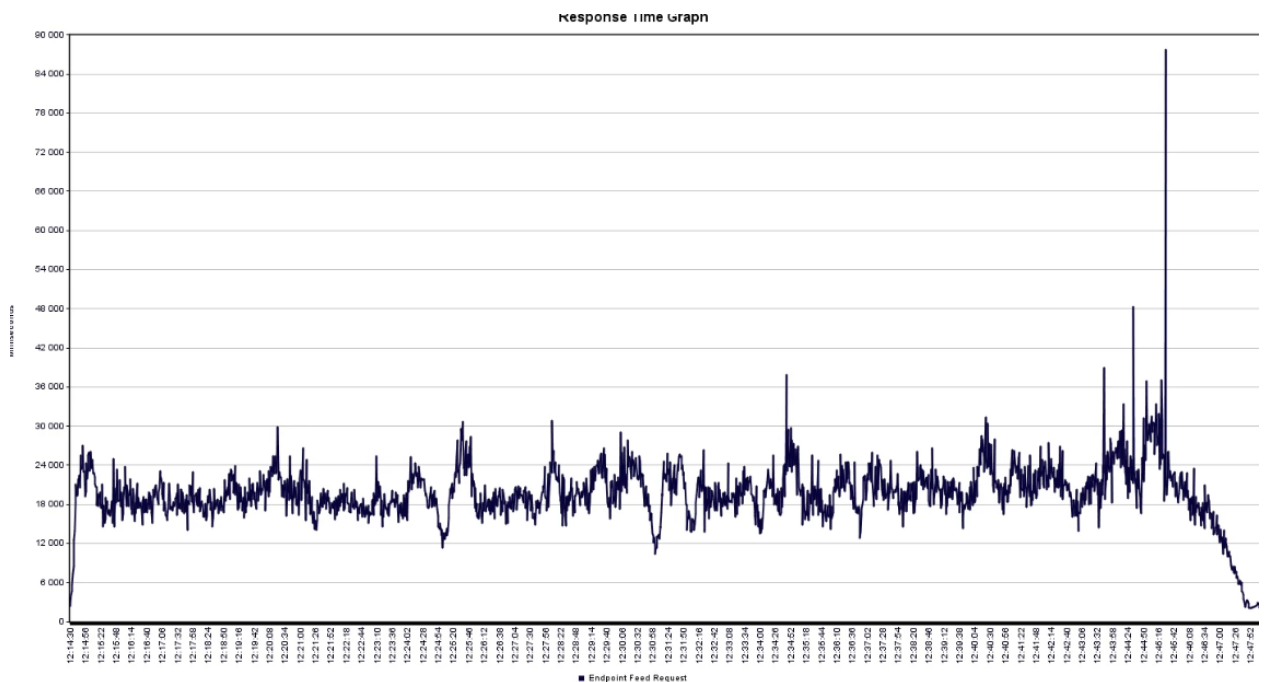


Рисунок 10 Мониторинг исходящих запросов из JMeter

## Результаты тестирования

Для анализа результатов был использован сервис визуализации Grafana предоставляющий гибкий инструмент для построения графиков.



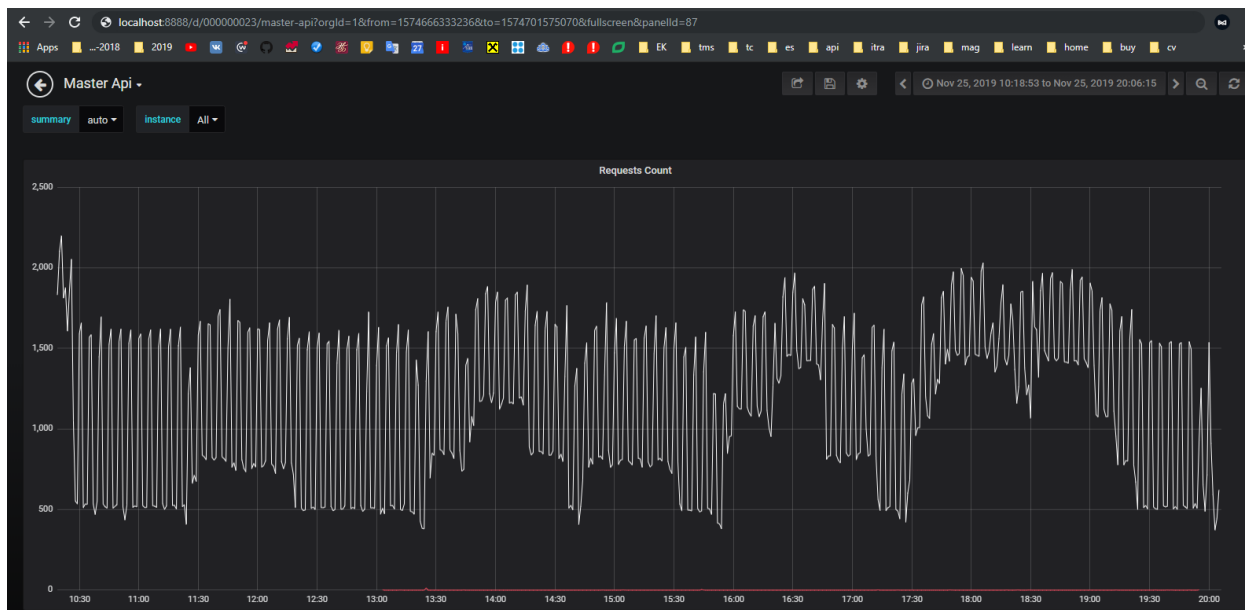


Рисунок 11 Запросы принятые через сервис визуализации Grafana

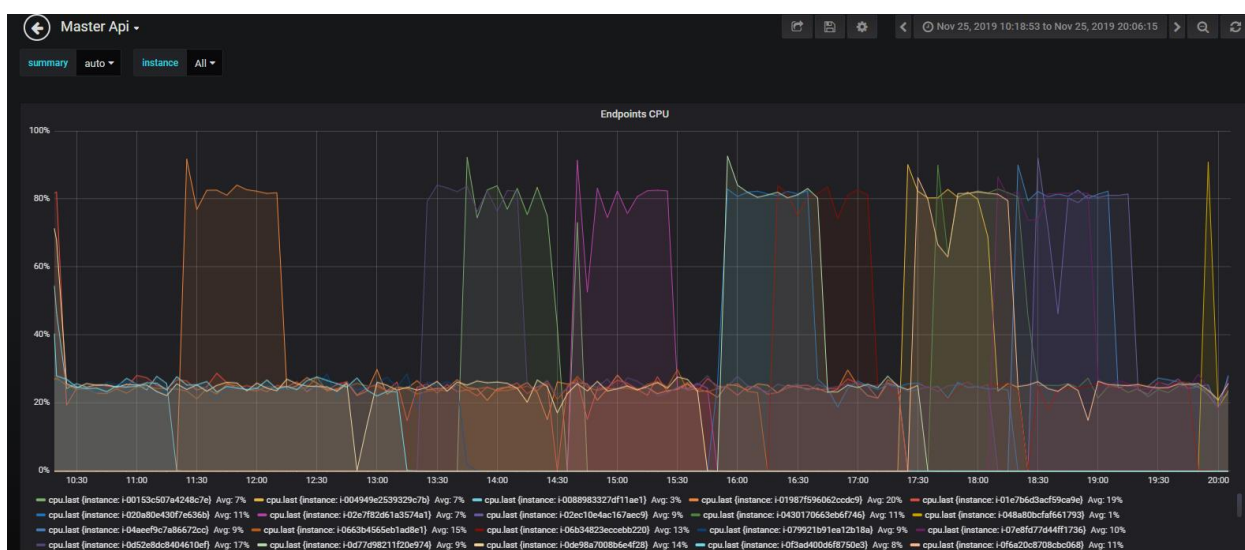


Рисунок 12 Мониторинг нагрузки на процессор

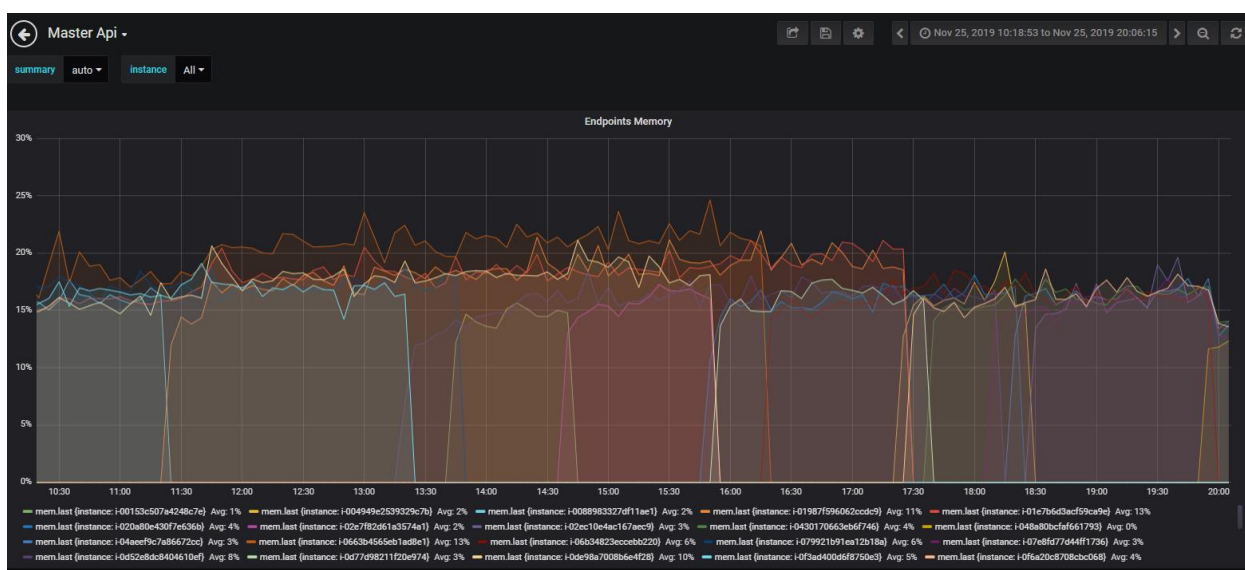


Рисунок 13 Мониторинг памяти приложения

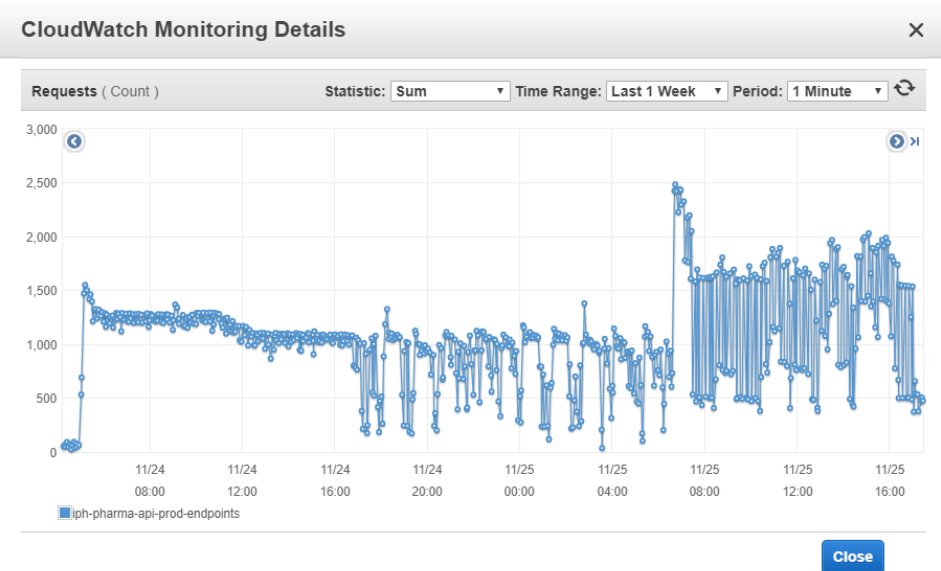


Рисунок 14 Мониторинг входящих запросов через Cloudwatch

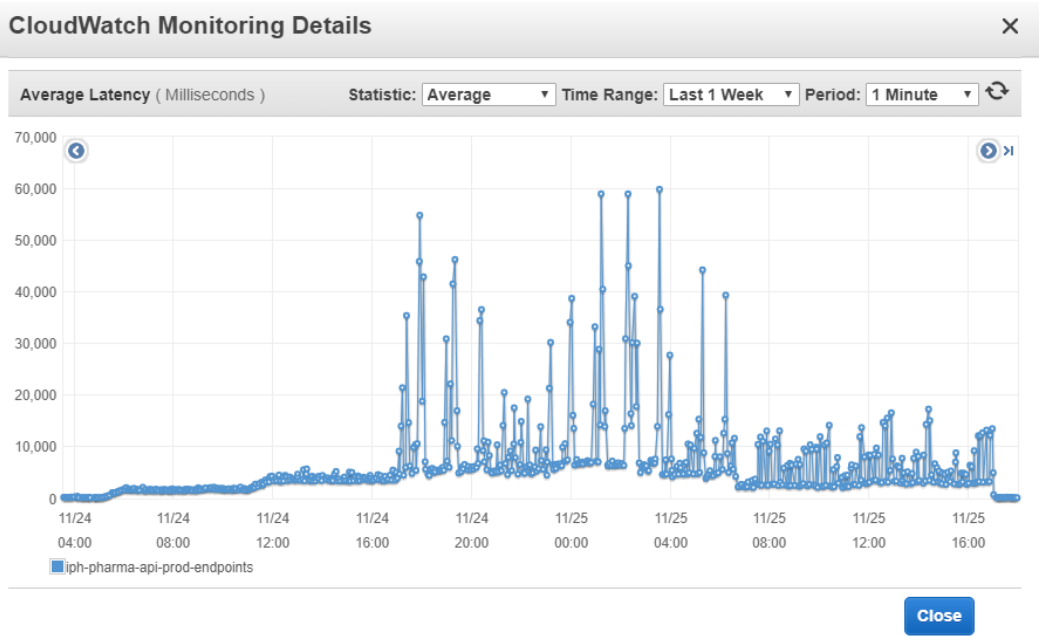


Рисунок 15 Задержка запросов в течении тестирования

	50	75	100	150	175	200
Avg response time	6846ms	9354ms	11413ms	15600ms	24220ms	19598ms
Execution time	11:47	16:20	19:36	26:48	40m	33:51

Рисунок 16 Длительность тестирования

### 3 Код реализации

```
[HttpGet("sum")]
public IActionResult Sum(int x1, int x2)
{
    if (!IsAcceptable)
    {
        return ObjectResultStatusCode(StatusCode.Status406NotAcceptable);
    }
    Thread.Sleep(500);

    return Serialize(new ServiceStatusResponseEnvelope
    {
        StatusCode = StatusCode.Status200OK,
        Message = (x1+x2).ToString(),
        Version = _versionProvider.LongVersion(GetType()),
        Generated = DateTime.UtcNow.ToString("o"),
    });
}
```

## **СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

[https://habr.com/ru/company/epam\\_systems/blog/188240/](https://habr.com/ru/company/epam_systems/blog/188240/)

<https://habr.com/ru/company/otus/blog/466519/>

<https://habr.com/ru/post/457100/>

<https://habr.com/ru/post/261483/>