



Universitatea Tehnică a Moldovei

**Implementarea unei arhitecturi reziliente  
bazate pe microservicii**

**Создание устойчивой архитектуры,  
основанной на микросервисах**

**Implementation of a resilient microservice  
architecture**

**Masterand:**

**Pancratov Alexei**

**Conducător:**

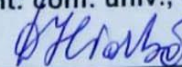
**conf.univ., dr. Ciorbă Dumitru**

**Chişinău — 2019**

MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII  
al REPUBLICII MOLDOVA  
Universitatea Tehnică a Moldovei  
Facultatea Calculatoare Informatică și Microelectronică  
Departamentul Ingineria Software și Automatică

Admis la susținere

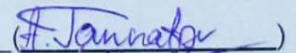
Șef de departament: conf. univ., dr. Ciorbă D.



„12” decembrie 2018

Создание устойчивой архитектуры, основанной  
на микросервисах  
Teză de master în  
Tehnologii Informaționale

Masterand: Pancratov Alexei



Conducător: Ciorbă Dumitru



## **Adnotarea**

Titlul lucrării – «Implementarea unei arhitecturi reziliente bazate pe microservicii», autorul Panctatov Alexei. Lucrarea consistă din: introducere, trei capitole, concluzie și bibliografie.

În aceasta lucrare se certetează metodele de atingere a rezilienței în sistemele distribuite bazate pe microservicii. Cercetare se începe cu o revizuire generală a microserviciilor. Principalele caracteristici și avantaje pe care le posedă în comparație cu sistemele monolitice sunt luate în considerare. În continuare, este luată în considerare reziliența sistemelor informaționale, precum și de ce asigurarea ei în sistemele de microservici reprezintă o sarcină dificilă, dar necesară. Asigurarea rezilienței este considerată mai mult din punctul de vedere al aplicării strategiilor general acceptate care reduc probabilitatea de eșec ("Retry"), previn eșecuri ("Timeouts", "Circuit Breaker", "Bulkheads") sau exclud apariția lor în sistemul ("Test Harness"). Se acordă atenție, de asemenea, a comunicării asincrone între microserviciile care utilizează cozile și brokerii de mesaje.

Scopul acestei lucrări a fost de a determina ce este un sistem distribuit rezilient și, de asemenea, de a studia modalitățile existente de asigurare a rezilienței și de a le pune în practică utilizând exemplul unui sistem de rezervări hoteliere.

Sistemul de microservici studiat este creat folosind tehnologia ASP.NET Core. Ea se compune din patru componente principale: un site web pentru utilizatorii finali, un microserviciu pentru procesarea rezervărilor, un microserviciu pentru căutarea hotelurilor și o bază de date. Comunicarea cu microservicii are loc prin intermediul protocolului HTTP. Fiecare dintre aceste componente este lansat în interiorul unui container Docker. Instrumentele pumba pentru Docker și mountebank au fost folosite pentru a testa rezistența și pentru a simula diferite scenarii.

Ca rezultat al studiului, sa constatat că, datorită alegerii corecte a strategiilor pentru asigurarea rezilienței, este posibil de obținut: un nivel înalt de SLA, reducerea numărului de eșecuri și distribuirea uniformă a sarcinilor pe resursele serverului. Aceste strategii pot fi aplicate pentru oricare sistem de microservici.

Cuvintele cheie: microservicii, reziliența, strategie de reziliența, arhitectură, comunicare, containere, testare, eșec.

## **Аннотация**

Название данной работы - «Создание устойчивой архитектуры, основанной на микросервисах», автор Панкратов Алесей. Она состоит из введения, трёх глав, вывода и библиографии.

В данной работе исследуются способы достижения устойчивости в распределённых системах, основанных на микросервисах. Исследование начинается с общего рассмотрения микросервисов. Рассматриваются их основные характеристики и преимущества, которыми они обладают по сравнению с монолитными системами. Далее рассматривается что такое устойчивость информационных систем, а также почему её обеспечение в микросервисных системах является сложной, но необходимой задачей. Обеспечение стабильности рассматривается в большей степени с точки зрения применения общепризнанных стратегий, которые снижают вероятность сбоя («Повторение»), предотвращают сбой («Таймауты», «Предохранитель», «Переборки») или исключают их появление в системе («Гостевая программа»). Также уделяется внимание рассмотрению асинхронному общению между микросервисами с использованием очередей и брокеров сообщений.

Целью данной работы было определить, что такое устойчивая распределённая система, а также изучить существующие способы обеспечения устойчивости и применить их на практике на примере системы бронирования номеров в отелях.

Рассматриваемая микросервисная система создана с использованием технологии ASP.NET Core. Она состоит из четырёх основных компонентов: веб-сайт для конечных пользователей, микросервис для обработки бронирований, микросервис для поиска отелей и базы данных. Общение с микросервисами происходит по протоколу HTTP. Каждый из этих компонентов развёрнут внутри Docker контейнера. Для тестирования устойчивости и симуляции различных сценариев были использованы инструменты rumba для Docker и mountebank.

В результате исследования было установлено, что благодаря правильному выбору стратегий обеспечения стабильности можно достичь высокого уровня SLA, снизить количество сбоев и равномерно распределить нагрузку на ресурсы сервера. Данные стратегии можно применить к любой микросервисной системе.

Ключевые слова: микросервисы, устойчивость, стратегия устойчивости, архитектура, общение, контейнеры, тестирование, сбой.

## **Abstract**

Title of thesis – “Implementation of a resilient microservice architecture”, author Pancratov Alexei. It consists of introduction, three chapters, conclusion and bibliography.

This paper describes ways to achieve resiliency in distributed systems based on microservices. Research starts with general overview of microservices. Their characteristics and advantages comparing to monolithic systems are examined. Then resiliency of information systems is examined, along with why its guarantee is hard to achieve in microservice systems, but is a necessary task. Resiliency attainment is considered more from the point of view of applying widely accepted strategies which reduce probability of failure (“Repeat”), prevent failures (“Timeouts”, “Circuit Breaker”, “Bulkheads”) or exclude them (“Test Harness”). In addition, asynchronous communication between microservices using queues and message brokers is considered.

The purpose of this study was to define what a resilient system means and to analyze existing practices of providing resiliency and to apply them on practice on a system for booking rooms in hotels.

The studied microservice system was created using ASP.NET Core technology. It consists of four main components: web-site for final users, microservice for bookings processing, microservice for hotels lookup and a database. Communication with microservices is done via the HTTP protocol. Each of these components is deployed inside a Docker container. For testing resiliency and simulation of different scenarios were used tools like pumba for Docker and mountebank.

As a result of the research it was established that thanks to the correct choice of resiliency strategy it is possible to achieve a high level of SLA, to reduce number of failures and to evenly distribute load on server. These strategies can be applied to any microservice system.

Keywords: microservices, resiliency, resiliency strategy, architecture, communication, containers, testing, failure.

## Содержание

Введение .....	10
1 Стабильность в системах, основанных на микросервисах .....	11
1.1 Общие сведения о микросервисной архитектуре .....	11
1.2 Устойчивость и шаги к её достижению .....	13
1.2.1 Определение требований к доступности системы .....	14
1.2.2 Разработка архитектуры системы .....	16
1.2.3 Стратегии для обеспечения стабильности .....	17
1.2.4 Тестирование имплементации, симулируя сбои .....	21
1.2.5 Развёртывание приложения используя надёжный и автоматизированный процесс .....	22
1.2.6 Мониторинг приложения на наличие сбоев на базе логов и графиков на их основе .....	23
1.2.7 Вмешательство технического персонала .....	23
1.3 Почему устойчивость тяжело обеспечить в микросервисной архитектуре? .....	23
1.4 Характеристики устойчивой микросервисной системы .....	25
2 Анализ паттернов и технологий обеспечения устойчивости .....	26
2.1 Архитектура системы и устойчивость .....	28
2.1.1 Синхронное общение посредством удалённого вызова процедуры .....	29
2.1.2 Асинхронное общение, используя сообщения .....	30
2.2 Контейнеры .....	33
2.3 Защита от перегрузки ресурсов .....	35
2.3.1 Проблема перегрузки ресурсов .....	35
2.3.2 Решение проблемы перегрузки ресурсов .....	35
2.4 Мониторинг состояния микросервисов .....	39
3 Результаты исследования и создание архитектуры .....	43
3.1 Имплементация микросервисов .....	43

3.2	Контейнеризация микросервисов .....	51
3.3	Имплементация устойчивого общения микросервисов в ASP.NET Core .....	56
3.3.1	Имплементация стратегии «Повторение» .....	60
3.3.2	Имплементация стратегии «Предохранитель» .....	61
3.3.3	Имплементация стратегии «Таймауты» .....	62
3.3.4	Имплементация стратегии «Переборки» .....	62
3.4	Тестирование устойчивости системы .....	63
3.4.1	Тестирование стратегии «Таймауты» .....	63
3.4.2	Тестирование стратегии «Предохранитель» .....	66
3.4.3	Тестирование стратегии «Повторение» .....	68
3.4.4	Тестирование стратегии «Переборки» .....	69
3.4.5	Имплементация стратегии «Тестовая программа» .....	72
3.5	Мониторинг системы .....	77
	Заключение .....	80
	Библиография .....	82