

**Министерство образования и науки Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)**

Кафедра автоматизации обработки информации (АОИ)

## **МИКРОСЕРВИСНАЯ АРХИТЕКТУРА**

Методические указания по выполнению лабораторных работ  
и организации самостоятельной работы  
для студентов направления «Программная инженерия»  
(уровень магистратуры)

**Владимиров Михаил Владимирович**

Микросервисная архитектура: Методические указания по выполнению лабораторных работ и организации самостоятельной работы для студентов направления «Программная инженерия» (уровень магистратуры) / М.В. Владимиров. – Томск, 2022. – 17 с.

Одобрено на заседании кафедры АОИ, протокол № 1 от 20.01.2022

## Оглавление

1 Введение .....	4
2 Методические указания к проведению лабораторных работ .....	5
2.1 Лабораторная работа «Проектирование системы на основе микросервисной архитектуры» .....	5
2.2 Лабораторная работа «Настройка Gateway» .....	6
2.3 Лабораторная работа «Очереди сообщений» .....	7
2.4 Лабораторная работа «Логирование в микросервисах» .....	8
2.5 Лабораторная работа «Работа с кластером» .....	9
2.6 Лабораторная работа «Авторизация в микросервисном приложении» .....	10
2.7 Лабораторная работа «Создание резервной копии базы данных» .....	11
2.8 Лабораторная работа «Проектирование микросервиса» .....	12
3 Методические указания для организации самостоятельной работы .....	13
3.1 Общие положения .....	13
3.2 Проработка лекционного материала .....	14
3.3 Подготовка к лабораторным работам .....	14
4. Рекомендуемая литература .....	17

## 1 Введение

Целью лабораторных работ по дисциплине «Микросервисная архитектура» является закрепление теоретических основ по разработке современных интернет-приложений на основе микросервисной архитектуры, а также формирование у студентов способности к самостоятельному или при помощи преподавателя анализу теоретического материала.

В результате проведения лабораторных работ студенты должны: получить знания об основных технологиях разработки, методах взаимодействия между элементами системы, подходах к проектированию ПО и разработке сложных систем; овладеть навыками разработки современных микросервисов, овладеть основными навыками разработки серверной части на основе современных фреймворков.

При изучении данной дисциплины необходимо знание студентами информатики в объеме первого, второго, третьего и четвертого курсов бакалавриата.

Пособие предназначено для студентов, обучающихся по направлению «Программная инженерия» всех форм обучения.

## **2 Методические указания к проведению**

### **лабораторных работ**

#### **2.1 Лабораторная работа «Проектирование системы на основе микросервисной архитектуры»**

##### **Цель занятия**

Изучить основные понятия, касающиеся проектирования систем на основе микросервисной архитектуры; получить навыки проектирования серверной части приложения; овладеть основами декомпозиции монолитных программных продуктов, изучить сопутствующие программные продукты при разработке микросервисов.

##### **Порядок выполнения лабораторной работы**

Данная лабораторная работа выполняется в текстовом редакторе MS Word, Google Docs и прочие.

Основная задача – выбрать тему для разработки сложной информационной системы, декомпонировать выбранную систему на микросервисы, продумать инфраструктурные составляющие и объяснить свой выбор в отчете.

Тему можно выбрать любую: маркетплейс (как Яндекс Маркет, например), внутренняя система предприятия (например, для предприятия по производству БПЛА), ПО для работы в ВУЗе и т.п. Выбрать можно любую тему.

Необходимо кратко описать особенности системы для понимания контекста при дальнейшем обсуждении. Данное описание должно тезисно отражать основные требования к системе.

Следующий этап – декомпозиция. Один из способов декомпозиции – на основе бизнес-задач, которые должна выполнять система. Например, каждая сложная информационная система требует наличия функций для авторизации и аутентификации пользователей. Для уведомления пользователей требуется сервис для рассылок.

Далее необходимо продумать наличие необходимых инфраструктурных элементов для работоспособности системы. Например, для получения бесплатных ssl-сертификатов требуется наличие контейнера для ПО certbot. Еще один пример: для отправки и выполнения асинхронных задач между разными сервисами необходимо наличие ПО RabbitMQ.

Необходимо весь получившийся результат отразить в отчете.

Сдача лабораторной работы проходит путем демонстрации отчета и обсуждения описанного проекта.

## 2.2 Лабораторная работа «Настройка Gateway»

### Цель занятия

Изучить основы организации единой точки входа в проект на основе веб-сервера, получить навыки настройки веб-сервера.

### Порядок выполнения работы

Для выполнения данной лабораторной работы необходимо создать два микросервиса для тестирования работоспособности создаваемого Gateway. Вся лабораторную работу необходимо выполнять, используя ПО Docker и Docker Compose.

Можно использовать различные подходы к организации Gateway. Предлагается в рамках данной лабораторной работы создать Gateway в виде прокси-сервера на основе NGINX. Документация по настройке NGINX, а также по проксированию запросов находится по следующей ссылке: [https://nginx.org/en/docs/beginners\\_guide.html](https://nginx.org/en/docs/beginners_guide.html)

Документация для использования официального образа NGINX для Docker находится по следующей ссылке: [https://hub.docker.com/\\_/nginx](https://hub.docker.com/_/nginx)

Пример конфигурации для настройки хоста с использованием префиксов можно посмотреть по следующей ссылке: <https://www.baeldung.com/linux/nginx-multiple-proxy-endpoints>

Запрос должен быть направлен в основной контейнер с NGINX. NGINX должен быть настроен на прием запросов по IP-адресу 127.0.0.1 и порту 80. В настройках данного хоста нужно указать параметры переадресации. При правильной разработке микросервисов каждый из них должен содержать уникальный префикс, который не повторяется в других микросервисах. В рамках файла конфигурации для данного хоста требуется указать префиксы, которые будут перенаправлять запрос в соответствующий микросервис по названию контейнера, который обрабатывает запросы соответствующего микросервиса.

Лабораторная работа считается выполненной, если отправка запросов на <http://127.0.0.1> с различными параметрами возвращает корректные ответы, реализованные в разных микросервисах.

В рамках данной лабораторной работы можно использовать результат, полученный при выполнении лабораторных работ по предмету «Разработка интернет-приложений», который изучался при обучении в бакалавриате по направлению «Программная инженерия» и «Бизнес-информатика».

Сдача лабораторной работы проходит путем демонстрации работоспособности разработанного функционала и написания отчета.

## 2.3 Лабораторная работа «Очереди сообщений»

### Цель занятия

Овладение навыками интеграции очереди сообщений в серверную часть проекта, изучение инструмента RabbitMQ, освоение основ докеризации обработчиков сообщений из очереди сообщений.

### Порядок выполнения работы

Для выполнения данной лабораторной работы понадобится результат выполнения второй лабораторной работы.

В рамках выполнения лабораторной работы предлагается использовать ПО RabbitMQ для организации очередей сообщений и проекты на основе Symfony для реализации основного функционала.

Ссылка на официальный образ RabbitMQ для Docker и его документацию: [https://hub.docker.com/\\_/rabbitmq](https://hub.docker.com/_/rabbitmq). Для работы с административной панелью очереди сообщений необходимо использовать версию RabbitMQ с постфиксом `-management`. При развертывании такого образа подключение ПО к очереди сообщений будет доступно по адресу `localhost:5672`, а административная панель будет доступна по адресу `localhost:15672`.

Для работы с очередями сообщений в рамках проекта на основе фреймворка Symfony необходимо добавить в проект бандл Messenger. Для его установки в проект потребуется выполнить команду `composer require messenger`.

Далее требуется настроить в `.env` файле взаимодействие с очередью сообщений путем указания соответствующих ссылок в новом блоке с настройками для установленного бандла.

Ссылка на документацию по работе с бандлом: <https://symfony.com/doc/current/messenger.html>.

В первую очередь после настройки бандла нужно создать класс, который будет являться структурой данных для сообщения. Делается это при помощи команды `bin/console make:message`. При выполнении данной команды будет создан файл со структурой сообщения, а также обработчик сообщения, которые нужно будет продублировать в микросервис, который далее будет обрабатывать сообщения.

Отправка сообщения в очередь выполняется следующим фрагментом кода: `$bus->dispatch(new SmsNotification('Look! I created a message!'));`

Данный код можно использовать в контроллере, консольных командах и т.д.

Обработчик сообщений должен быть реализован отдельным контейнером. Необходимо выводить в лог результат обработки сообщений. Для выполнения лабораторной работы будет достаточно в логах отобразить отправленное сообщение.

Лабораторная работа считается выполненной, если весь разработанный функционал реализован при помощи технологии Docker, а также есть возможность продемонстрировать процесс отправки сообщения в очередь сообщений и его последующей обработки другим микросервисом.

Сдача лабораторной работы проходит путем демонстрации работоспособности заданного функционала и написания отчета.

## **2.4 Лабораторная работа «Логирование в микросервисах»**

### **Цель занятия**

Овладение навыками организации логирования в распределенных системах, изучение инструментов для хранения и обработки логов, изучение подходов к логированию.

### **Порядок выполнения работы**

Для быстрого анализа работоспособности и выявления ошибок часто возникает необходимость для создания системы правильного логирования. Для распределенных систем это сложный процесс, требующий использования стороннего ПО.

Для выполнения данной лабораторной работы предлагается использовать ПО Logstash и результаты 2 лабораторной работы.

В первую очередь при выполнении лабораторной работы необходимо настроить микросервисы так, чтобы логи были в едином формате. Для этого используется бандл Monolog.

Ссылка на документацию по работе с данным бандлом и инструкции по его установке в проект: <https://symfony.com/doc/current/logging.html>.

Для отслеживания ошибок при вложенных запросах необходимо организовать процесс логирования таким образом, чтобы можно было однозначно идентифицировать самый первый запрос и все его последующие запросы. Для организации такого подхода при получении запроса



микросервисом или Gateway, или первый микросервис-обработчик запроса должен генерировать уникальный идентификатор запроса. Например, Uuid4. Данный идентификатор необходимо передавать всем последующим микросервисам. Каждый микросервис должен логировать входящий запрос и указывать полученный идентификатор.

Получившиеся логи будут находиться в соответствующих файлах в соответствующих микросервисах. Для удобства поиска неисправностей и анализа логов организуется централизованное хранение всех логов, которые необходимо передавать в отдельное ПО. В рамках данной лабораторной работы предлагается использовать программу Logstash. Ссылка на документацию для официального образа Docker: [https://hub.docker.com/\\_/logstash](https://hub.docker.com/_/logstash).

После разворачивания Logstash требуется настроить микросервисы так, чтобы логи после записи попадали в ПО для хранения логов.

Лабораторная работа считается выполненной, если в развернутом инструменте для централизованного хранения логов отобразятся логи при отправке нового запроса.

Сдача лабораторной работы проходит путем демонстрации работоспособности заданного функционала и написания отчета.

## **2.5 Лабораторная работа «Работа с кластером»**

### **Цель занятия**

Овладение навыками создания кластера, изучение инструментов для работы с кластерами, овладение навыками объединения нескольких серверов в один кластер.

### **Порядок проведения занятия**

Часто для горизонтального масштабирования требуется создать кластер. Кластер – это объединение ресурсов нескольких компьютеров для выполнения одной задачи.

Для выполнения данной лабораторной работы необходимо использовать ПО Docker. Для организации кластеров используется Docker Swarm, который уже предустановлен в Docker. Документация по Docker Swarm: <https://docs.docker.com/engine/swarm/>.

Чтобы продемонстрировать выполненную задачу, необходимо объединиться в пары, инициализировать кластер на одном устройстве и подключить к нему другое устройство.

Также необходимо развернуть в кластере сервис, который будет работать, используя вычислительные ресурсы кластера. Для этого можно использовать результат выполнения лабораторной работы №2.

Для выполнения данной лабораторной работы необходимо в терминале использовать команды *docker swarm init*, *docker swarm join* и *docker service create*.

Сдача лабораторной работы проходит путем демонстрации работоспособности заданного функционала и написания отчета.

## **2.6 Лабораторная работа «Авторизация в микросервисном приложении»**

### **Цель занятия**

Овладение навыками разработки микросервисов, изучение особенностей авторизации и аутентификации при работе с микросервисами.

### **Порядок выполнения работ**

Основной задачей данной лабораторной работы является внедрение авторизации в разработанные на лабораторной работе №2 микросервисы.

Как правило, в приложении на основе микросервисной архитектуры несколько различных микросервисов требуют наличие функций для авторизации пользователей.

В рамках выполнения данной лабораторной работы можно использовать любой из известных методов авторизации. Предлагается использовать способ авторизации, основанный на запросах в микросервис авторизации.

Для создания микросервиса авторизации необходимо создать новый микросервис и предусмотреть в нем возможности регистрации пользователя, авторизации пользователя и получения токена. Можно использовать результат соответствующей лабораторной работы, выполненной на предмете «Разработка интернет-приложений», который изучался в программе бакалавриата для направлений «Программная инженерия» и «Бизнес-информатика».

Далее, для внедрения ограничений в прочие микросервисы, требуется создать аутентификатор. Документация по созданию аутентификатора находится по ссылке: [https://symfony.com/doc/current/security/custom\\_authenticator.html](https://symfony.com/doc/current/security/custom_authenticator.html).

В данном аутентификаторе предлагается получать из заголовков запроса токен, который был выдан сервисом аутентификации, после чего его нужно будет отправить, используя функцию `curl` в сервис авторизации, чтобы получить информацию о пользователе.

Полученные данные о пользователе необходимо будет использовать при проверке прав доступа на то или иное действие в новом микросервисе.

Для сдачи лабораторной работы требуется продемонстрировать процесс авторизации и получения токена, отправку запроса с токеном и получением правильного ответа, а также отправку того же запроса без токена и отображение ошибки о нехватке прав для выполнения задачи.

Сдача лабораторной работы проходит путем демонстрации работоспособности заданного функционала и написания отчета.

## **2.7 Лабораторная работа «Создание резервной копии базы данных»**

### **Цель занятия**

Овладение навыками создания резервных копий базы данных, овладение навыком использования резервных копий для наполнения базы данных.

### **Порядок выполнения работ**

Обеспечение безопасности данных – это одна из самых важных задач разработчика сложных информационных систем. Основной способ сохранения данных – это создания резервных копий базы данных.

Для выполнения данной лабораторной работы требуется создать базу данных, наполнить ее информацией, создать резервную копию базы данных, удалить базу данных, создать новую базу данных на основе имеющейся резервной копии.

Для выполнения данной лабораторной работы предлагается использовать СУБД MySQL 5.7 и проект на основе фреймворка Symfony. Можно использовать базу данных, которая была создана на одной из лабораторных работ ранее.

Если проект в предыдущих лабораторных работах создавался на основе фреймворка Symfony, то для настроенного и готового к работе проекта создание базы данных выполняется командой `php bin/console doctrine:database:create`.

Далее требуется создать как минимум одну сущность и внести изменения в базу данных.

Добавить данные в базу данных можно несколькими способами: создать API и выполнить несколько запросов, добавить данные с помощью клиентского приложения для базы данных, использовать миграции и т.п.

После наполнения созданной базы данных данными требуется создать резервную копию, используя следующую команду: `mysqldump -u USER -pPASSWORD DATABASE > /path/to/file/dump.sql`

Данная команда создаст файл `dump.sql`, который будет содержать всю внесенную информацию.

Следующий шаг – удаление базы данных для проверки работоспособности резервной копии.

После удаления базы данных необходимо создать новую пустую базу данных. Чтобы применить данные из резервной копии к новой базе данных, нужно выполнить команду: `mysql -u USER -pPASSWORD DATABASE < /path/to/dump.sql`

Для сдачи лабораторной работы необходимо продемонстрировать процесс удаления базы данных, применения к новой базе данных информации из резервной копии. Также нужно продемонстрировать в новой базе данных наличие старых данных.

Сдача лабораторной работы проходит путем демонстрации работоспособности заданного функционала и написания отчета.

## **2.8 Лабораторная работа «Проектирование микросервиса»**

### **Цель занятия**

Получение навыков по проектированию микросервисов.

### **Порядок выполнения работ**

Процесс проектирования является одним из важнейших и базовых процессов при разработке современных микросервисов. Проектирование

– это необходимый процесс, т.к. он позволяет избежать ошибок, а также сделать работу микросервиса эффективнее, а его сопровождение проще.

Для выполнения данной лабораторной работы рекомендуется использовать любой удобный текстовый редактор.

Задача – выбрать любой микросервис и спроектировать его, указав, какие необходимы контейнеры для разработки и production-режима, какие нужно использовать технологии, какие инструменты для взаимодействия.

Сдача лабораторной работы проходит путем демонстрации работоспособности заданного функционала и написания отчета.

## **3 Методические указания для организации самостоятельной работы**

### **3.1 Общие положения**

Целями самостоятельной работы являются систематизация, расширение и закрепление теоретических и практических знаний, приобретение навыков разработки интернет-приложений.

Самостоятельная работа студента по дисциплине «Микросервисная архитектура» включает следующие виды его активности:

1. проработка лекционного материала;
2. подготовка к лабораторным работам;
3. подготовка к экзамену.

### **3.2 Проработка лекционного материала**

Данный вид самостоятельной работы направлен на получение навыков работы с конспектом, структурирования материала, а также умения выделить основные пункты и положения, изложенные на лекции. Кроме того, проработка лекционного материала способствует более глубокому пониманию и прочному запоминанию теоретической части дисциплины.

При проработке лекционного материала необходимо:

1. обработать прослушанную лекцию, то есть прочитать конспект, прочитать документацию и сопоставить его материал с конспектом; восполнить пробелы, если они остались после лекции в силу того, что студент что-то не понял или не успел записать;

2. перед каждой последующей лекцией прочитать предыдущую, чтобы не тратилось много времени для восстановления контекста изучения дисциплины при продолжающейся теме, а также чтобы максимально правильно ответить на вопросы теста, который проводится на каждой лекции.

Для наиболее эффективной работы с конспектом рекомендуется сначала просмотреть его целиком, чтобы выделить структуру лекции. Эту структуру полезно выписать в виде плана. Затем по каждому пункту нужно выделить основные положения, определения и формулы, если они есть. Формулы тоже полезно записывать, чтобы кроме зрительной, включалась еще и моторная память.

### **3.3 Подготовка к лабораторным работам**

Для подготовки к практическим занятиям необходимо изучить теоретические вопросы по теме работы, проработать основные понятия, необходимые для решения практических задач и выполнения индивидуально-го задания по практической работе.

**Лабораторная работа «Проектирование системы на основе микросервисной архитектуры»**

При подготовке к данной лабораторной работе необходимо повторить изученный материал по организации баз данных, основам работы с Docker, а также по основам декомпозиции. Данная информация понадобится в процессе выполнения лабораторной работы.

Лабораторная работа будет выполняться текстовом редакторе.

### **Лабораторная работа «Настройка Gateway»**

При подготовке к выполнению данной лабораторной работы требуется повторить изученный материал по предмету «Разработка интернет-приложений», а именно основы PHP и Symfony.

Для студентов, выполняющих лабораторную работу на личном ПК требуется развернуть окружение, а именно Docker или PHP, Composer и MySQL актуальной версии.

### **Лабораторная работа «Очереди сообщений»**

При подготовке к данной лабораторной работе необходимо изучить документацию для ПО RabbitMQ. Необходимо изучить процесс взаимодействия и дополнительные инструменты, предоставляемые Symfony.

### **Лабораторная работа «Логирование в микросервисах»**

Для успешной подготовки к лабораторной работе по теме «Логирование в микросервисах» необходимо повторить изученный материал о работе с Docker и Symfony. Основные команды, которые используются в стандартном shell-терминале, команды, которые предоставляет composer, symfony-cli и сопутствующих бандлов.

Необходимо также учитывать, что разработка будет вестись на языке PHP, в связи с чем стоит повторить пройденный материал и перечитать документацию по теме.

### **Лабораторная работа «Работа с кластером»**

В данном разделе рассматриваются только задачи по созданию кластера между двумя компьютерами. Создание кластера включает в себя основы работы с терминалом, основа организации серверов для обработки запросов, основы работы с Docker.

### **Лабораторная работа «Авторизация в микросервисном приложении»**

Для подготовки к данной работе необходимо учитывать рекомендации, приведенные в предыдущем пункте.

Кроме этого, для качественного выполнения лабораторной работы необходимо повторить пройденный материал из курса по разработке интернет-приложений.

### **Лабораторная работа «Создание резервной копии базы данных»**

При подготовке к данной лабораторной работе нужно изучить актуальную документацию по использованию СУБД MySQL, а также лекционный материал. Особенное внимание необходимо уделить консольным командам в контейнере.

Также требуется изучить возможности cli, которое предоставляет ПО Docker, а именно команды для запуска, остановки, проверки и проксирования команд.

### **Лабораторная работа «Проектирование микросервиса»**

При подготовке к лабораторной работе по теме «Проектирование микросервиса», следует обратить внимание на лекционный материал.



## 4. Рекомендуемая литература

1. Организация баз данных: Методические указания к лабораторным работам, курсовой работе и организации самостоятельной работы / П. В. Сенченко - 2018. 85 с.

2. Информатика и программирование: Учебное пособие / Н. В. Пермякова - 2016. 188 с.

3. Основы гипертекстового представления интернет-контента: учебное пособие / Ю. П. Ехлаков, Э. К. Ахтямов - 2017. 181 с.