

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

РАБОТА ПРОВЕРЕНА

Рецензент

_____ В.В. Батуев
«__» _____ 2022 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой ЭВМ

_____ Д.В. Топольский
«__» _____ 2022 г.

Реализация вычислительной платформы с организацией сбора, хранения,
анализа данных интернета вещей в туманной вычислительной среде
(стартап как диплом)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУРГУ-090401.2022.207 ПЗ МД

Консультант,

к.э.н., доцент каф.

Экономика и финансы

_____ Л.Ш. Морозова
«__» _____ 2022 г.

Руководитель работы,

заведующий кафедрой ЭВМ

_____ Д.В. Топольский
«__» _____ 2022 г.

Консультант,

директор ООО «Стратегия роста»

_____ С.В. Ведерников
«__» _____ 2022 г.

Автор работы,

студент группы КЭ-222

_____ А.А. Асташов
«__» _____ 2022 г.

Нормоконтролер,

ст. преп. каф. ЭВМ

_____ С.В. Сяськов
«__» _____ 2022 г.

Челябинск-2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

_____ Д.В. Топольский

«___» _____ 2022 г.

ЗАДАНИЕ

на выполнение магистерской диссертации

студенту группы КЭ-222

Асташов Андрей Алексеевич

обучающемуся по направлению

09.04.01 «Информатика и вычислительная техника»

1. **Тема работы:** «Реализация вычислительной платформы с организацией сбора, хранения, анализа данных интернета вещей в туманной вычислительной среде» утверждена приказом по университету от 25.04.2022 № 627-13/2.
2. **Срок сдачи студентом законченной работы:** 1 июня 2022 г.
3. **Исходные данные к работе:**
 - платформа оркестрации контейнеризованных приложений Kubernetes (не ниже 1.21.x версии);
 - платформа контейнеризации Docker (не ниже 20.10.x версии);
 - программа для управления жизненным циклом контейнеров Containerd;

- фреймворк для реализации распределённой обработки данных Apache Spark версии 3.2.1;
- распределенный программный брокер сообщений Apache Kafka версии 3.1.0;
- набор серверов (4 CPU, 8 RAM) с предустановленной операционной системой Ubuntu Server версии 20.04.4 LTS (Focal Fossa);
- операционная система Windows выпуск 10 Pro версии 20H2 сборки 19042.1586;
- операционная система Ubuntu Server версии 20.04.4 LTS (Focal Fossa);
- язык программирования Python версии 3.9.x;
- среда разработки PyCharm Professional версии 2021.3;
- пользовательские аккаунты в VK Cloud Solutions (Россия).

4. Перечень подлежащих разработке вопросов:

- введение и обзор литературы;
- определение требований;
- проектирование системы;
- разработка системы;
- проведение тестирования;
- оценка экономической эффективности;
- анализ результатов.

5. Дата выдачи задания: 1 декабря 2021 г.

Руководитель работы _____ /Д.В. Топольский/

Студент _____ /А.А. Астахов/

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	10.03.2022	
Определение требований. Проектирование платформы	21.03.2022	
Реализация платформы	04.04.2022	
Тестирование	25.04.2022	
Оценка экономической эффективности	05.05.2022	
Компоновка текста работы и сдача на нормоконтроль	16.05.2022	
Подготовка презентации и доклада	24.05.2022	

Руководитель работы _____ /Д.В. Топольский/

Студент _____ /А.А. Астахов/

Аннотация

А.А. Асташов. Реализация вычислительной платформы с организацией сбора, хранения, анализа данных интернета вещей в туманной вычислительной среде. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШ ЭКН; 2022, 58 с., 30 ил., библиогр. список – 28 наим.

В рамках выпускной квалификационной работы производится детальный анализ современных технологий облачных и туманных вычислений. Выполняется составление требований, проектирование и разработка вычислительной платформы для сбора, хранения и анализа данных интернета вещей в туманной вычислительной среде. Производится тестирование программных интерфейсов ядра платформы. Реализуется оценка экономической эффективности планируемого стартапа. Анализируются результаты и составляется заключение.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	8
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	11
1.1. ИНТЕРНЕТ ВЕЩЕЙ	11
1.2. ПРОМЫШЛЕННЫЙ ИНТЕРНЕТ ВЕЩЕЙ	12
1.3. ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ	13
1.4. ТУМАННЫЕ ВЫЧИСЛЕНИЯ.....	15
1.5. КОНТЕЙНЕРНАЯ ВИРТУАЛИЗАЦИЯ.....	15
1.6. КОНТЕЙНЕРНАЯ ОРКЕСТРАЦИЯ	17
1.7. KUBERNETES	17
1.8. DOCKER SWARM.....	18
2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ	20
2.1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ.....	20
2.2. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ.....	21
2.2.1. ИНСТРУМЕНТЫ РАЗРАБОТКИ.....	21
2.2.2. ИНФРАСТРУКТУРА РАЗРАБОТКИ	21
2.2.3. АРХИТЕКТУРА ПЛАТФОРМЫ.....	21
2.2.4 ВЫБОР СУБД	22
2.3. ВЫВОДЫ	22
3. ПРОЕКТИРОВАНИЕ.....	23
3.1. РАЗВЕРТЫВАНИЕ КЛАСТЕРА.....	23
3.2. РАЗВЕРТЫВАНИЕ ВЫЧИСЛИТЕЛЬНОЙ ПЛАТФОРМЫ.....	24
3.3. КОМПОНЕНТЫ ПЛАТФОРМЫ.....	26
3.4. ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ ПЛАТФОРМЫ	27
3.5. ОПИСАНИЕ КОНТЕЙНЕРОВ	29
3.6. ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ МИКРОСЕРВИСА СБОРА.....	30

3.7. ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ МИКРОСЕРВИСА ХРАНЕНИЯ	31
3.8. ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ МИКРОСЕРВИСА АНАЛИЗА	32
3.9. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ	33
3.10. ВЫВОДЫ	34
4. РЕАЛИЗАЦИЯ	35
4.1. БЛОК-СХЕМА АЛГОРИТМА ПРЕДОСТАВЛЕНИЯ РАСПРЕДЕЛЕННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	35
4.2. БЛОК-СХЕМА ЯДРА ПЛАТФОРМЫ	36
4.3. ВЫВОДЫ	37
5. ТЕСТИРОВАНИЕ	38
5.1. МЕТОДОЛОГИЯ ТЕСТИРОВАНИЯ	39
5.2. ПРОВЕДЕНИЕ ПРОЦЕДУРЫ ТЕСТИРОВАНИЯ	39
5.3. ВЫВОДЫ	44
6. ОЦЕНКА ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ	45
6.1. ОРГАНИЗАЦИОННЫЙ ПЛАН	45
6.2. ПОТРЕБНОСТЬ В ИНВЕСТИЦИЯХ	47
6.3. ОТЧЕТ О ДВИЖЕНИИ ДЕНЕЖНЫХ СРЕДСТВ	48
6.4. ОТЧЕТ О ФИНАНСОВЫХ РЕЗУЛЬТАТАХ	50
6.5. РАСЧЕТ ПОКАЗАТЕЛЕЙ ЭФФЕКТИВНОСТИ ИНВЕСТПРОЕКТА	52
6.6. ВЫВОДЫ	54
7. ЗАКЛЮЧЕНИЕ	55
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	56

ВВЕДЕНИЕ

На сегодняшний день невозможно представить жизнь без повсеместной информатизации. Информационные технологии настолько внедрены в жизнедеятельность современного человека, что, кажется, в мире практически не остается людей, которые хотя бы раз не воспользовались ими.

Однозначно, с развитием информационных технологий жизнь становится все более удобной и безопасной.

Появление электронной вычислительной техники позволило человеку выполнять миллиарды логических и математических операций за короткое время, что стало эволюционным прорывом для всего человечества. Внедрение современных компьютеров создало такую эволюцию в каждом доме. Организация единой сети передачи данных между компьютерами позволила людям делиться информацией на протяжении в сотни тысяч километров без длительного ожидания отправки и доставки сообщений, что обеспечило мгновенное получение знаний о всем, что только существует в нашей жизни. Непрерывное повышение вычислительной емкости компьютерной техники до сих пор создает новые возможности развития как для информационной сферы, так и для всего человечества.

Цель и задачи работы

В ходе выполнения выпускной квалификационной работы необходимо выполнить такие задачи, как:

- произвести детальный обзор предметной области;
- определить функциональные и нефункциональные требования;
- спроектировать вычислительную платформу;
- разработать вычислительную платформу;
- провести тестирование программных интерфейсов;

- выполнить расчет экономической эффективности инвестпроекта;
- проанализировать полученные результаты и сделать сопутствующие выводы.

Структура и объем работы

Выпускная квалификационная работа состоит из введения, 6 глав, заключения и библиографического списка.

Работа составляет 58 страниц, в библиографическом списке указано 28 источников.

Содержание работы

В первой главе производится анализ предметной области по тематике разрабатываемой вычислительной платформы.

Во второй главе описаны функциональные и нефункциональные требования к разрабатываемой вычислительной платформе.

В третьей главе описывается проектирование вычислительной платформы, ее составляющих, приведены диаграммы в нотации UML.

В четвертой главе продемонстрирована реализация основных алгоритмических составляющих ядра платформы.

В пятой главе выполняется тестирование программных интерфейсов платформы на предмет подтверждения ожидаемых результатов.

Шестая глава представляет вниманию расчет экономической эффективности инвестпроекта планируемого стартапа.

В заключении описаны результаты разработки и оценки экономической эффективности.

Актуальность

Актуальность разработки вычислительной платформы с организацией сбора, хранения и анализа данных интернета вещей в туманной вычислительной среде заключается в применении новейшей вычислительной концепции

туманных вычислений, которая позволит как сократить время передачи данных с устройств интернета вещей до вычислительной инфраструктуры, так и развертывать собственные инфраструктуры вычислительных мощностей для обеспечения возможностей отказа от сервисов облачных провайдеров и реализации цифровых решений на основе платформы с организацией взаимодействия с устройствами интернета вещей.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

На сегодняшний день, когда технологии интернета вещей и облачных вычислений уверенно находят свое применение, развиваются новые вычислительные концепции, в частности технологии туманных вычислений. Данный вид вычислений до сих пор является новейшей концепцией предоставления вычислительных ресурсов, которая только начинает набирать свою популярность [1].

1.1. ИНТЕРНЕТ ВЕЩЕЙ

Интернет вещей (Internet of things, IoT) – концепция взаимодействия сенсоров и исполнительных устройств как в сети интернет, так и с помощью специальных видов сетей.

К специальным видам сетей интернета вещей относятся:

- сети с низким энергопотреблением и малым диапазоном (Bluetooth, Z-Wave, NFC, ZigBee, WiFi/802.11) [2];
- сети с низким энергопотреблением и широкой зоной охвата (4G LTE, 5G, Cat-0, Cat-1, LoRaWan, LTE, CAT-M1, Sigfox).

Развитие технологий интернета вещей во многом повлияло на жизнь современного человека. С помощью «умного» взаимодействия датчиков и актуаторов в повседневной жизнедеятельности появилась, ранее привычная только для промышленных предприятий, автоматизация, которая позволила свести к минимуму обыденные и рутинные действия. Стоит добавить, что автоматизация повлияла не только на обыденность и рутину, но и существенно повысила уровень жизни современного человека с помощью использования сбора и анализа обрабатываемой информации.

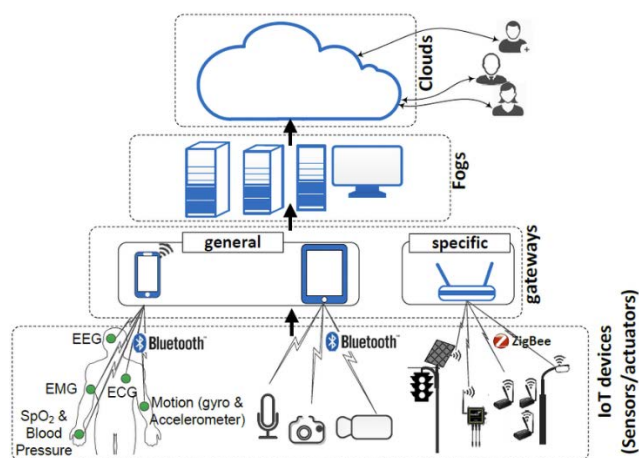


Рисунок 1 – Взаимодействие IoT устройств и облачных вычислений

1.2. ПРОМЫШЛЕННЫЙ ИНТЕРНЕТ ВЕЩЕЙ

Технологии интернета вещей нашли свое применение не только в повышении качества и безопасности жизнедеятельности современного человека, но также позволили привнести новшества в бизнес-процессы промышленного производства. Технологии и устройства традиционного интернета вещей, адаптированные под нужды промышленности, были названы промышленным интернетом вещей (Industrial internet of things, IIoT).

Использование технологий промышленного интернета вещей во многом позволило предприятиям:

- увеличить капитализацию;
- увеличить маржинальность товаров и услуг;
- сократить издержки;
- создать и протестировать новые бизнес-модели;
- повысить скорость адаптации ко внешним изменениям;
- повысить эффективность бизнес-процессов;
- сократить трудозатраты производства.

К основным функциональным возможностям концепции промышленного интернета вещей относится прогнозирование результата выпускаемой продукции, а также мониторинг и диспетчеризация состояния оборудования и инфраструктуры предприятия.

1.3. ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ

Облачные вычисления – модель предоставления услуг вычисления и хранения информации в удаленных центрах обработки данных с помощью использования сети интернет [3].

Модель предоставления вычислительных услуг подразделяется на 3 основных вида:

- инфраструктура как сервис (infrastructure as a service, IaaS);
- платформа как сервис (platform as a service, PaaS);
- программное обеспечение как сервис (software as a service, SaaS).

«Инфраструктура как сервис» подразумевает предоставление таких вычислительных ресурсов, как виртуальные машины, сети, кластеры и реестры контейнеров. Данный вид услуги направлен на взаимодействие с разработчиками и системными администраторами.

«Платформа как сервис» подразумевает предоставление удобных инструментов разработки программного обеспечения, которые не редко имеют тесную интеграцию с инфраструктурными облачными компонентами. Модель предоставления «платформа как сервис» направлена на взаимодействие с разработчиками, администраторами, архитекторами программных решений и другими, причастными к разработке программного обеспечения, лицами.

«Программное обеспечение как сервис» предоставляет функциональные возможности для прямого взаимодействия с пользователем, который может не

иметь знаний о разработке и администрировании программного обеспечения, но желает приобрести необходимые ему функции за определенную стоимость.

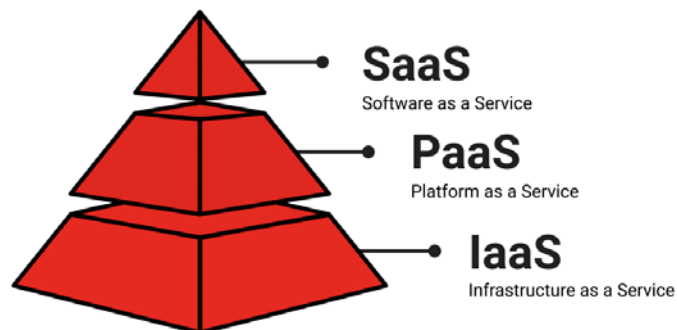


Рисунок 2 – Модели предоставления облачных услуг

Стоит отметить, что каждая из рассмотренных ранее моделей предоставления облачных вычислительных услуг может быть представлена в различной степени доступности. Таким образом, выделяют следующие виды доступности облаков:

- открытые;
- частные (приватные);
- гибридные.

Открытые облачные платформы ориентируются на публичное предоставление сервисов. Примером российских публичных облачных платформ являются сервисы «Яндекс Облако» и «VK Cloud Solution».

Частные (приватные) облака организуются для предоставления вычислительных ресурсов конкретному и ограниченному числу потребителей

Гибридное предоставление облачных сервисов подразумевает частичное развертывание вычислительной инфраструктуры в публичной области пользования и частичное в частной (приватной).

1.4. ТУМАННЫЕ ВЫЧИСЛЕНИЯ

Туманные вычисления – концепция предоставления вычислительных ресурсов в непосредственной близости к конечным пользователям [4]. Стоит отметить, что конечным пользователем может являться устройство интернета вещей.

Туманная вычислительная концепция предназначена для минимизации времени передачи информации от пользователя до вычислительного узла.

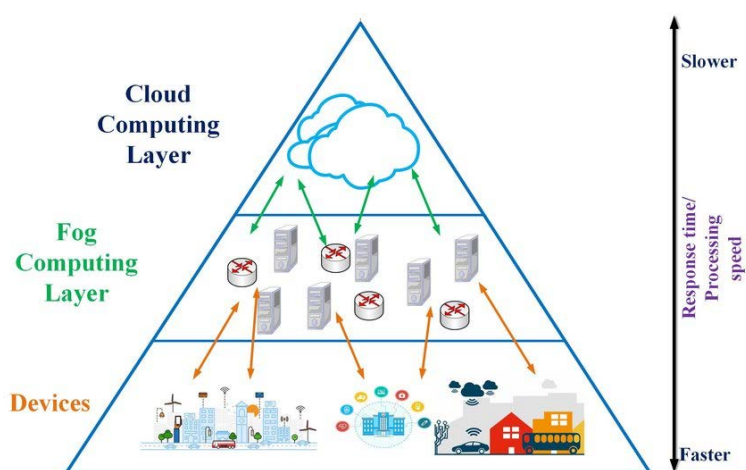


Рисунок 3 – Вычислительные слои облачных и туманных вычислений

Стоит понимать, что туманные вычисления дополняют облачные, выполняя предиктивный анализ и агрегацию данных перед отправкой для вычислений в облачный центр обработки данных.

1.5. КОНТЕЙНЕРНАЯ ВИРТУАЛИЗАЦИЯ

В отличие от аппаратной виртуализации, когда эмулируется аппаратное окружение и может быть запущено множество гостевых операционных систем, в контейнере запускается экземпляр операционной системы только с тем же ядром, что и у исходной (хостовой) операционной системы. Таким образом, все

контейнеры узла используют общее ядро. При этом отсутствуют дополнительные накладные расходы ресурсов на эмуляцию виртуального оборудования и запуск полноценного экземпляра операционной системы, что характерно аппаратной виртуализации.

Таким образом, контейнеры позволяют уместить гораздо больше приложений на одном физическом сервере, чем любая виртуальная машина, которая занимает гораздо больше системных ресурсов. В отличие от виртуальной машины, где эмулируется операционная система и необходимое виртуальное оборудование, в контейнере размещается только приложение и необходимый минимум системных библиотек. Благодаря этому можно запустить в 2-3 раза больше приложений на одном сервере. Также контейнеризация позволяет создавать портативное и целостное окружение для разработки, тестирования и последующего развертывания.

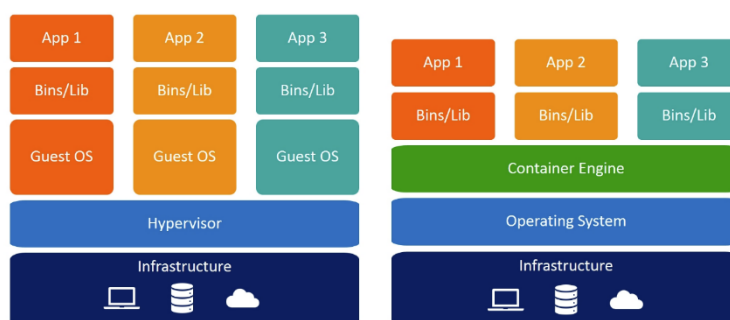


Рисунок 4 – Сравнение архитектуры виртуализации на базе гипервизора и архитектуры на базе контейнерной виртуализации

1.6. КОНТЕЙНЕРНАЯ ОРКЕСТРАЦИЯ

Контейнерная оркестрация – организация совместной работы множества сервисов в рамках одного приложения. На рисунке 3 можем видеть схему контейнеров, объединенных в кластер.

На данный момент, доступно множество решений для оркестрации контейнеров, которые могут управлять полным жизненным циклом контейнеров или группой контейнеров, масштабируя их вертикально или горизонтально, перемещая, самостоятельно восстанавливая, выполняя оркестрацию хранения и другие функции.

Лидерами решений контейнерной оркестрации являются Kubernetes и Docker Swarm.

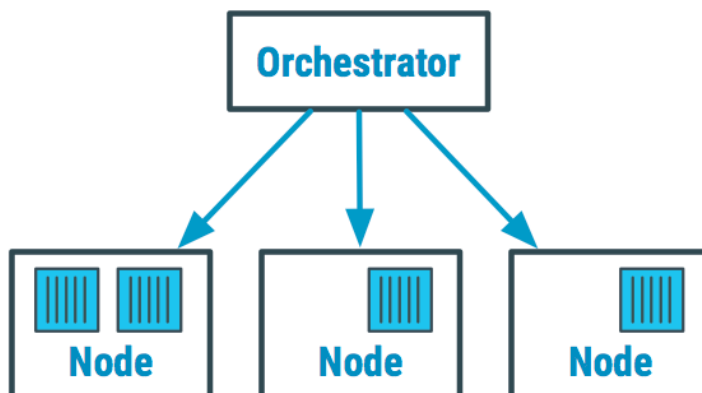


Рисунок 5 – Схема развертывание контейнеров в кластере

1.7. KUBERNETES

Kubernetes – инструмент оркестрации с открытым исходным кодом для автоматизации развертывания, масштабирования и управления приложениями на основе контейнеров. Можно запускать Kubernetes в локальной или публичной облачной инфраструктуре. Kubernetes устанавливает кластер, состоящий из

Kubernetes-master, Kubernetes-minions (рабочие узлы) и Pod (один или группа контейнеров для запуска одного приложения). Kubernetes-master наблюдает за одним или несколькими миньонами и управляет приложениями.

Kubernetes управляет и запускает контейнеры на большом количестве хостов, а также обеспечивает совместное размещение и репликацию большого количества контейнеризированных контейнеров. Kubernetes обеспечивает высокую надежность контейнеров, эффективно использует доступные ресурсы, координирует развертывание и реализует взаимодействие контейнеров между собой.

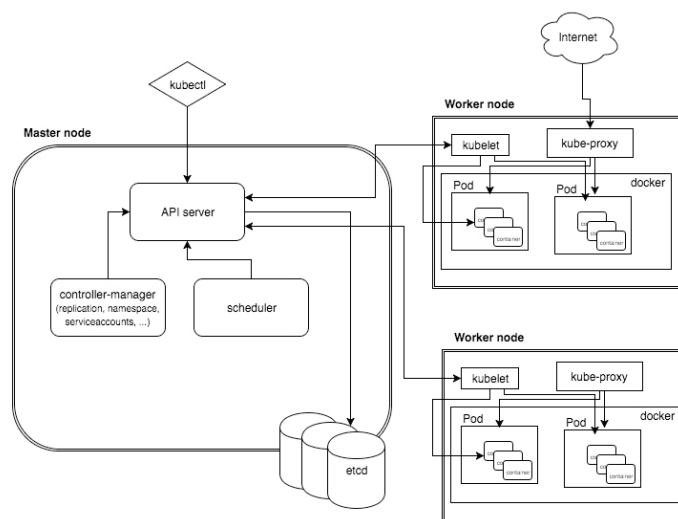


Рисунок 6 – Архитектура платформы оркестрации Kubernetes

1.8. DOCKER SWARM

Docker Swarm – кластерный режим платформы контейнерной виртуализации Docker, позволяющий беспрепятственно организовывать кластер вычислительных узлов для управления контейнеризованными приложениями. Является одним из наиболее простых по взаимодействию оркестраторов.

Архитектурно состоит из управляющих и рабочих узлов, каждый из выполняет свою функцию.

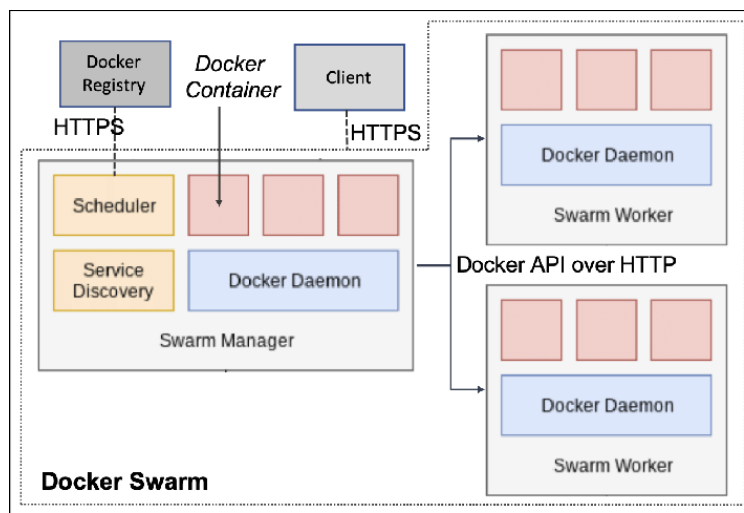


Рисунок 7 – Архитектура платформы оркестрации Docker Swarm

2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

Для реализации вычислительной платформы с организацией сбора, хранения, анализа данных интернета вещей в туманной вычислительной среде необходим следующий набор программного обеспечения и виртуального оборудования:

- фреймворк для реализации распределённой обработки данных Apache Spark версии 3.2.1;
- распределенный программный брокер сообщений Apache Kafka версии 3.1.0;
- набор серверов (4 CPU, 8 RAM) с предустановленной операционной системой Ubuntu Server версии 20.04.4 LTS (Focal Fossa);
- операционная система Ubuntu Server версии 20.04.4 LTS (Focal Fossa);
- среда разработки PyCharm Professional версии 2021.3.

2.1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

К перечню функциональных требований вычислительной платформы относятся:

- авторизация в платформе;
- добавление и инициализация кластера;
- создание необходимого ресурса;
- удаление необходимого ресурса;
- горизонтальное масштабирование ресурса;
- получение списка запущенных вычислительных ресурсов;
- получение информации о состоянии узлов кластера (загрузка CPU, RAM).

2.2. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

2.2.1. ИНСТРУМЕНТЫ РАЗРАБОТКИ

В качестве инструментов разработки были выбраны:

- язык программирования Python 3;
- веб-фреймворк языка Python Flask;
- платформа контейнеризации Docker;
- платформа оркестрации Kubernetes;
- платформа сборки и доставки программного обеспечения GitHub Actions;
- платформа облачного хранения кода GitHub;

2.2.2. ИНФРАСТРУКТУРА РАЗРАБОТКИ

Для реализации вычислительной платформы необходимо обеспечить доступ к личному кабинету провайдера облачных сервисов VK Cloud Solution с возможностью заказывать услугу предоставления кластера Kubernetes на требуемое количество вычислительных узлов.

2.2.3. АРХИТЕКТУРА ПЛАТФОРМЫ

Для реализации вычислительной платформы была выбрана микросервисная архитектура.

К основным необходимым преимуществам микросервисной архитектуры относятся:

- независимость инструментов и платформ разработки;
- возможность горизонтального масштабирования репликаций микросервисов;
- высокий уровень доступности разрабатываемого решения;

- упрощенная возможность поддержки и доработки отдельных микросервисов.

2.2.4 ВЫБОР СУБД

В качестве реляционной системы управления базой данных была выбрана PostgreSQL версии 14.3.

2.3. ВЫВОДЫ

В ходе выявления требований были определены функциональные требования и нефункциональные требования, в том числе:

- определены инструменты разработки;
- определена инфраструктура разработки;
- определена архитектура вычислительной платформы;
- выбрана система управления базой данных.

3. ПРОЕКТИРОВАНИЕ

Для описания развертывания кластера, развертывания вычислительной платформы, компонентов платформы, вариантов использования платформы и микросервисов, а также составляющих контейнеров был использован унифицированный язык моделирования UML, предназначенный для упрощенного проектирования и разработки программного обеспечения.

3.1. РАЗВЕРТЫВАНИЕ КЛАСТЕРА

Для описания развертывания кластера используем диаграмму развертывания UML.

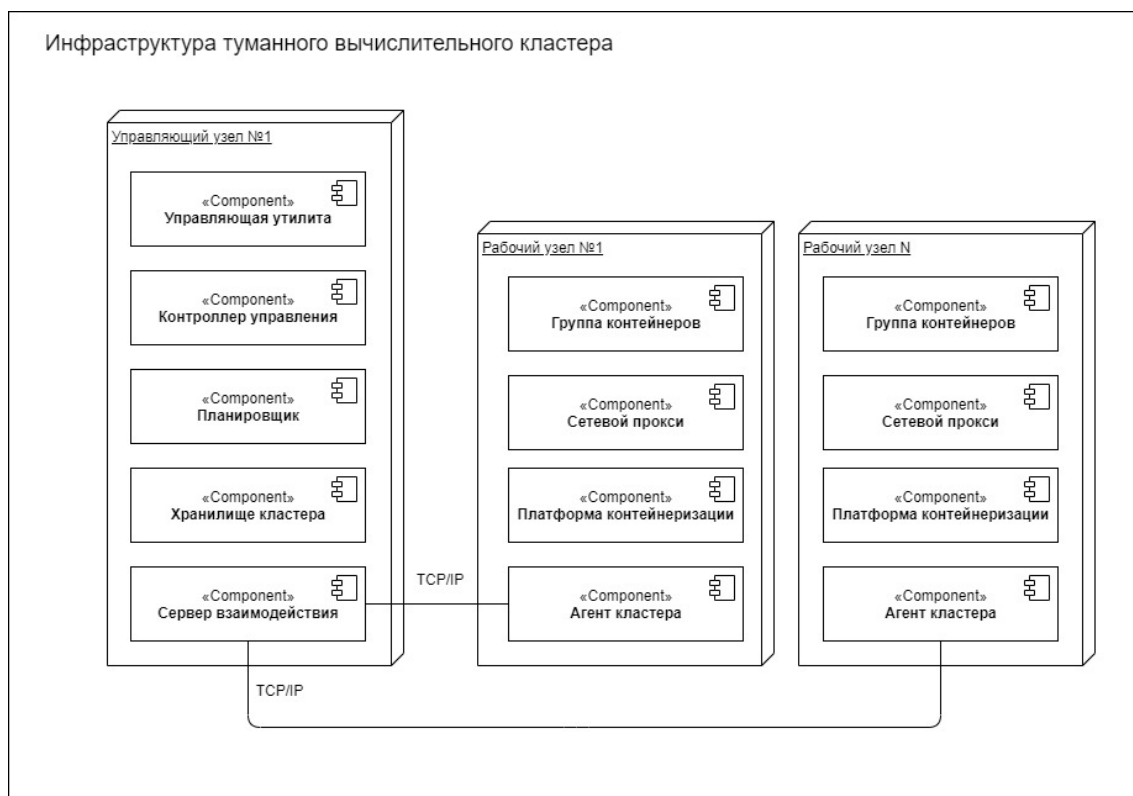


Рисунок 8 – Диаграмма развертывания туманного вычислительного кластера

Из рисунка 12 можем видеть, что кластер инфраструктуры развертывания платформы содержит управляющие и рабочие узлы.

Управляющие узлы кластера включают в себя:

- управляющую утилиту;
- контроллер управления;
- планировщик ресурсов;
- системное хранилище;
- сервер взаимодействия через API.

Рабочие узлы кластера включают в себя:

- агент кластера;
- платформу контейнеризации;
- сетевой прокси сервер;
- группу контейнеров (группа может состоять как из множества, так и из одного контейнера).

3.2. РАЗВЕРТЫВАНИЕ ВЫЧИСЛИТЕЛЬНОЙ ПЛАТФОРМЫ

Для описания развертывания вычислительной платформы воспользуемся унифицированным языком моделирования UML, который предоставляет нотацию по описанию диаграммы развертывания. Диаграмма развертывания вычислительной платформы продемонстрирована на рисунке 13.



Рисунок 9 – Диаграмма развертывания вычислительной платформы

Из рисунка 13 можем видеть, что в минимальной конфигурации вычислительная платформа состоит из:

- системы (микросервиса) предоставления распределенного программного обеспечения для сбора данных;
- системы (микросервиса) предоставления распределенного программного обеспечения для хранения данных;
- системы (микросервиса) предоставления распределенного программного обеспечения для проведения интеллектуального анализа данных.

3.3. КОМПОНЕНТЫ ПЛАТФОРМЫ

Для описания компонентов вычислительной платформы используем диаграмму компонентов в нотации языка UML. Диаграмма компонентов вычислительной платформы продемонстрирована на рисунке 14.

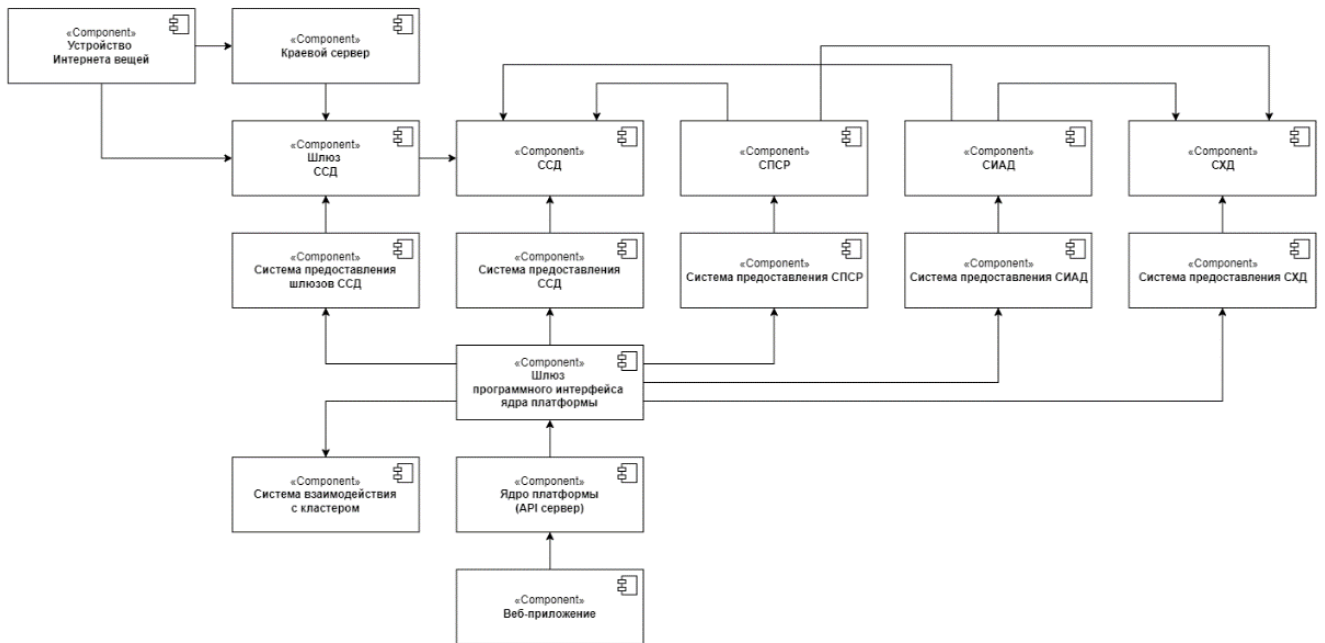


Рисунок 10 – Диаграмма компонентов вычислительной платформы

Из диаграммы компонентов видим, что платформа состоит из:

- ядра платформы;
- шлюза программного интерфейса маршрутизации запросов;
- утилиты взаимодействия с кластером;
- системы (микросервиса) предоставления распределенного программного обеспечения для сбора данных;
- системы (микросервиса) предоставления распределенного программного обеспечения для хранения данных;

- системы (микросервиса) предоставления распределенного программного обеспечения для проведения интеллектуального анализа данных;
- системы сбора данных;
- системы интеллектуального анализа данных;
- системы хранения данных;
- системы предоставления шлюза взаимодействия с устройствами интернета вещей и краевыми серверами;
- шлюза взаимодействия с устройства интернета вещей и краевыми серверами;
- устройства интернета вещей;
- краевые сервера.

3.4. ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ ПЛАТФОРМЫ

Для описания вариантов использования вычислительной платформой воспользуемся диаграммой прецедентов UML. Диаграмма прецедентов вычислительной платформы изображена на рисунке 15.

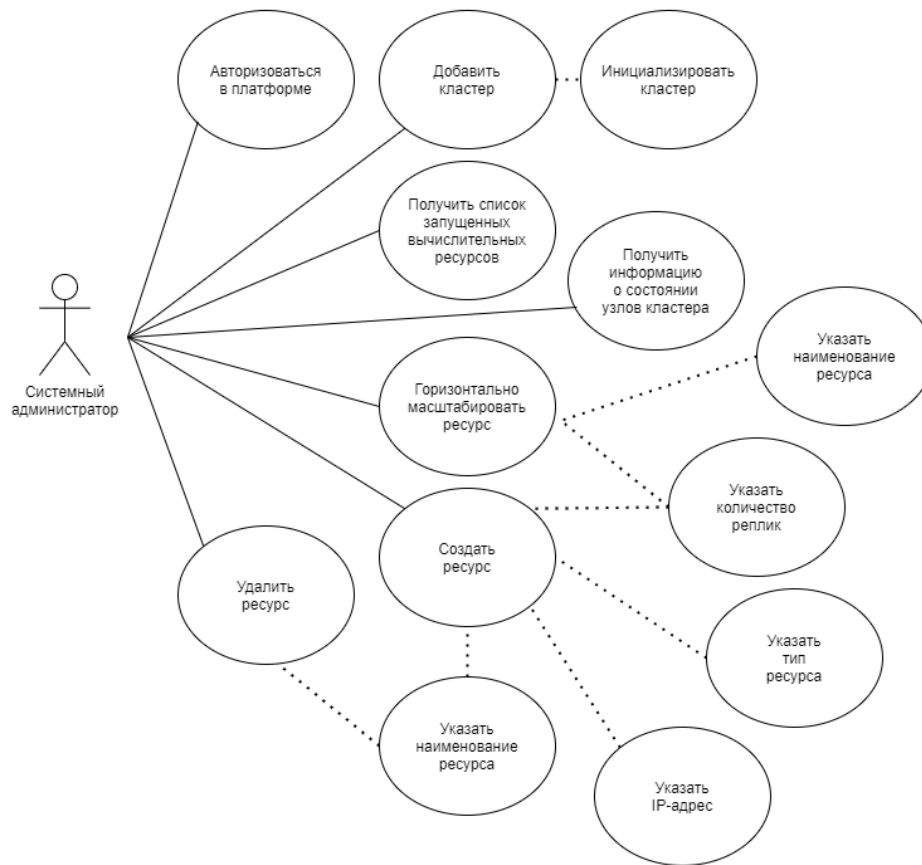


Рисунок 11 – Диаграмма прецедентов вычислительной платформы

Из рисунка 15 видно, что платформа предоставляет такие возможности, как:

- авторизация в платформе;
- добавление и инициализация кластера;
- создание необходимого ресурса;
- удаление необходимого ресурса;
- горизонтальное масштабирование ресурса;
- получение списка запущенных вычислительных ресурсов;
- получение информации о состоянии узлов кластера (загрузка CPU, RAM).

3.5. ОПИСАНИЕ КОНТЕЙНЕРОВ

Для описания составляющих компонентов каждого микросервиса, работающего внутри контейнера, воспользуемся унифицированным языком моделирования программного обеспечения UML. На рисунке 16 представлена диаграмма пакетов описания составляющих контейнеров микросервисов.

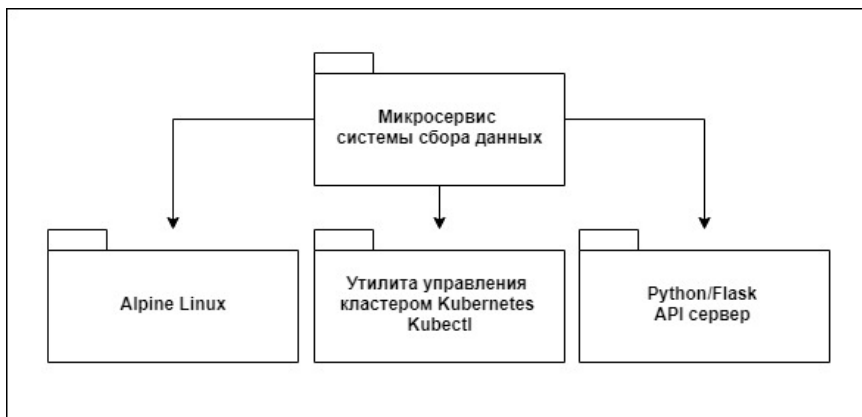


Рисунок 12 – Диаграмма пакетов описания составляющих контейнеров микросервисов

Из рисунка 16 видим, что диаграмма пакетов описывает такие компоненты, как:

- Alpine Linux – дистрибутив Linux, ориентированный на безопасность, легковесность и нетребовательность к ресурсам;
- утилиту управления кластером kubectl, с помощью которой происходит обработка запросов на микросервис;
- python/flask api сервер, предназначенный для обработки запросов.

3.6. ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ МИКРОСЕРВИСА СБОРА

Для описания предоставляемых возможностей микросервисов, воспользуемся нотацией UML.

На рисунке 17 представлена диаграмма прецедентов микросервиса сбора данных.

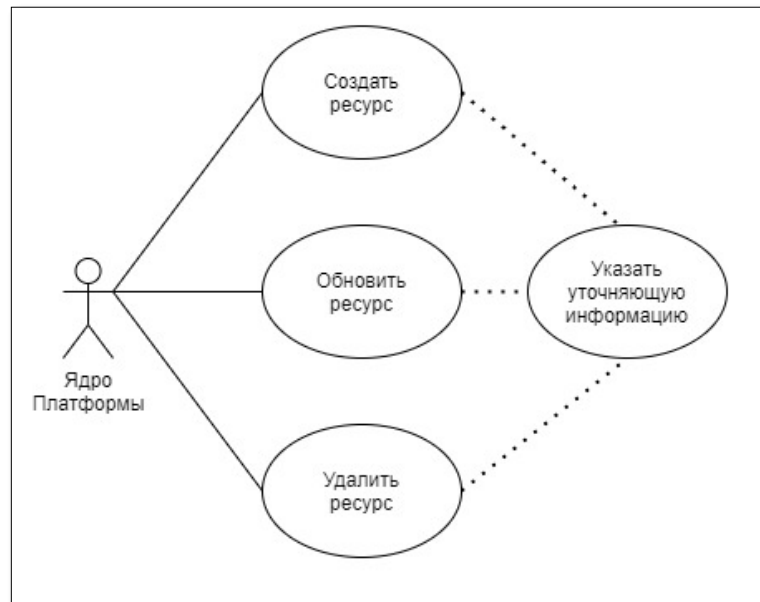


Рисунок 13 – Диаграмма прецедентов микросервиса сбора данных

Из рисунка 17, на котором изображены диаграммы прецедентов микросервиса сбора данных видим, что микросервис обеспечивает возможности:

- создание ресурса;
- обновление ресурса;
- удаления ресурса.

3.7. ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ МИКРОСЕРВИСА ХРАНЕНИЯ

Для описания функциональных возможностей микросервиса хранения данных используем язык моделирования UML.

На рисунке 18 представлена диаграмма прецедентов микросервиса хранения данных.

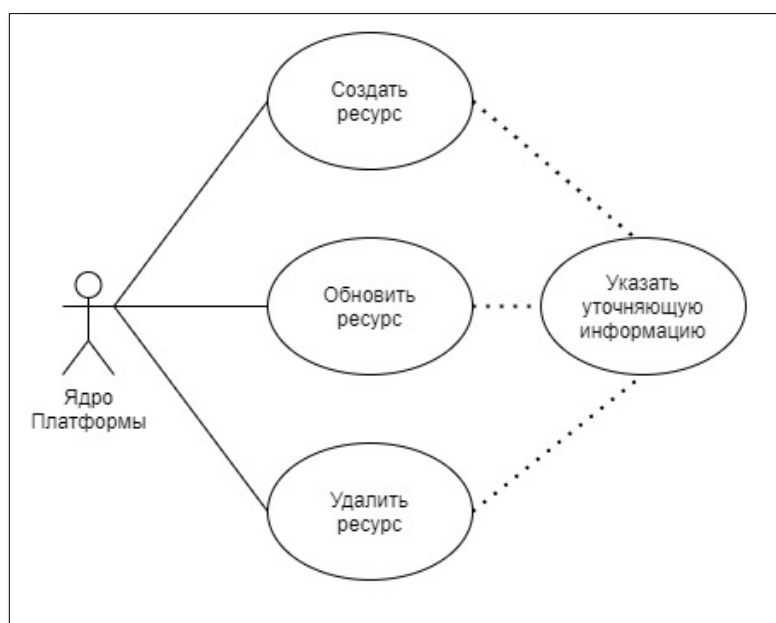


Рисунок 14 – Диаграмма прецедентов микросервиса хранения данных

Из диаграммы прецедентов видим (рисунок 18), что микросервис предоставляет такие возможности, как:

- создание ресурса;
- обновление ресурса;
- удаления ресурса.

3.8. ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ МИКРОСЕРВИСА АНАЛИЗА

Для описания использования микросервиса интеллектуального анализа данных воспользуемся унифицированным языком моделирования UML.

На рисунке 19 представлена диаграмма прецедентов микросервиса интеллектуального анализа данных.

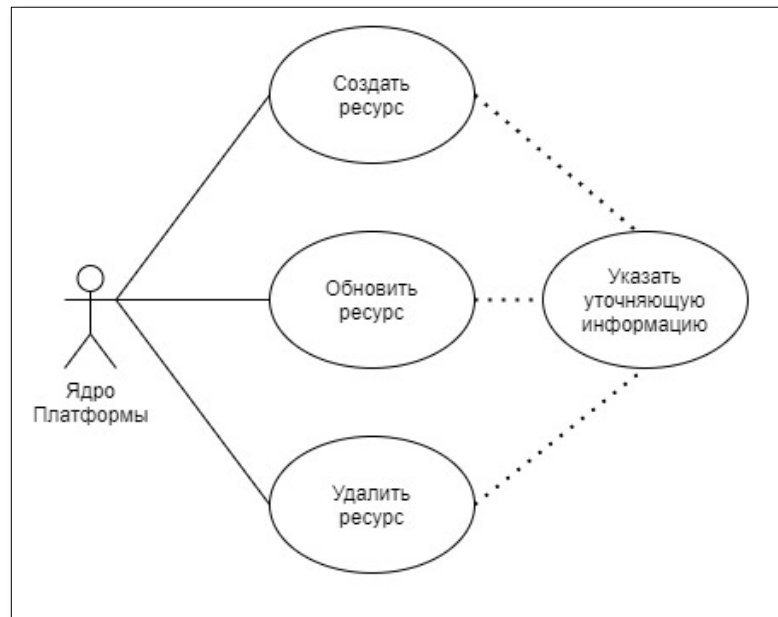


Рисунок 15 – Диаграмма прецедентов микросервиса анализа данных

Из рисунка 19 видим, что диаграмма прецедентов описывает следующие возможности микросервиса:

- создание ресурса;
- обновление ресурса;
- удаления ресурса.

3.9. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

Для проектирования базы данных составим схему. На рисунке 20 изображена схема базы данных.

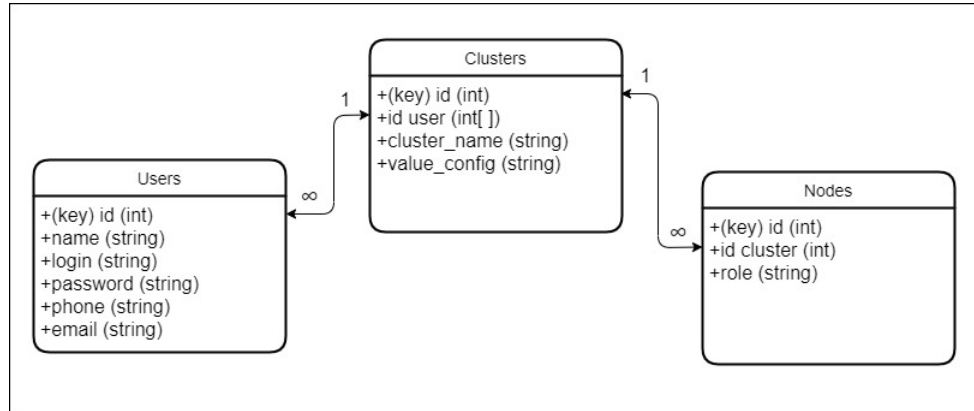


Рисунок 16 – Схема данных

Для описания данных предоставим перечень таблиц базы данных (таблицы 1-4).

Таблица 1 – Список таблиц базы данных.

Название таблицы	Назначение
Users	Данные о пользователях
Clusters	Данные о кластерах
Nodes	Данные об узлах

В таблицах 2-4 описаны атрибуты полей таблиц.

Таблица 2 – Атрибуты таблицы «Users».

Наименование	Тип данных	Назначение
Id	int(64)	Уникальный идентификатор
Name	Text(неогр.)	Имя (ФИО)
Login	Text(неогр.)	Логин

Password	Text(неогр.)	Пароль
Phone	Text(неогр.)	Контактный телефон
Email	Text(неогр.)	Электронная почта

Таблица 3 – Атрибуты таблицы «Clusters»

Наименование	Тип данных	Назначение
Id	Int(64)	Уникальный идентификатор
Id user	Int(64)	Идентификатор пользователя
Cluster_name	Text(неогр.)	Имя кластера
Value_config	Text(неогр.)	Конфигурационные данные авторизации

Таблица 4 – Атрибуты таблицы «Nodes».

Наименование	Тип данных	Назначение
Id	Int(64)	Уникальный идентификатор
Id_cluster	Int(64)	Идентификатор кластера
role	Text(неогр.)	Роль узла

3.10. ВЫВОДЫ

В ходе проектирования были описаны такие детали вычислительной платформы, как развертывание кластера и вычислительной платформы, компоненты и варианты использования платформы, контейнеры и их прецеденты. Также была спроектирована схема базы данных с описанием данных.

4. РЕАЛИЗАЦИЯ

Реализация вычислительной платформы подразумевает использование микросервисной архитектуры. Из диаграммы компонентов, представленной на рисунке 14, следует, что основными составляющими компонентами платформы являются ядро и микросервисы предоставления распределенного программного обеспечения.

4.1. БЛОК-СХЕМА АЛГОРИТМА ПРЕДОСТАВЛЕНИЯ РАСПРЕДЕЛЕННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

На рисунке 21 представлена блок-схема фрагмента алгоритма развертывания программного обеспечения микросервисом.

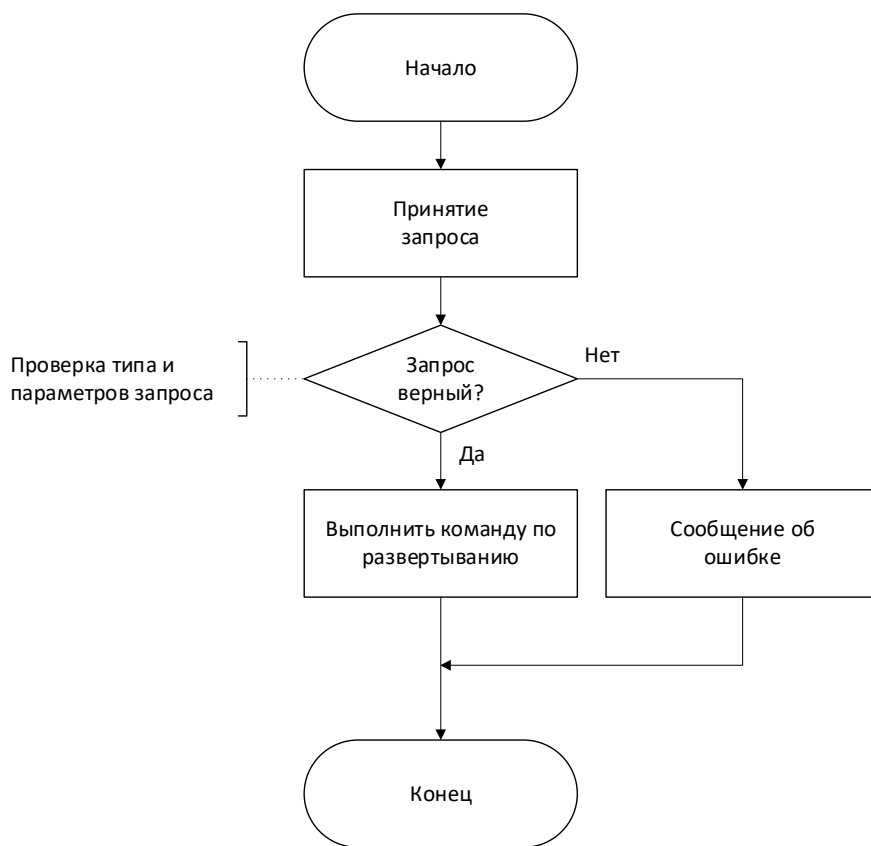


Рисунок 17 – Блок-схема фрагмента алгоритма микросервиса

4.2. БЛОК-СХЕМА ЯДРА ПЛАТФОРМЫ

На рисунке 22 представлена блок-схема фрагмента алгоритма функции выполнения запроса к микросервисам от ядра платформы.

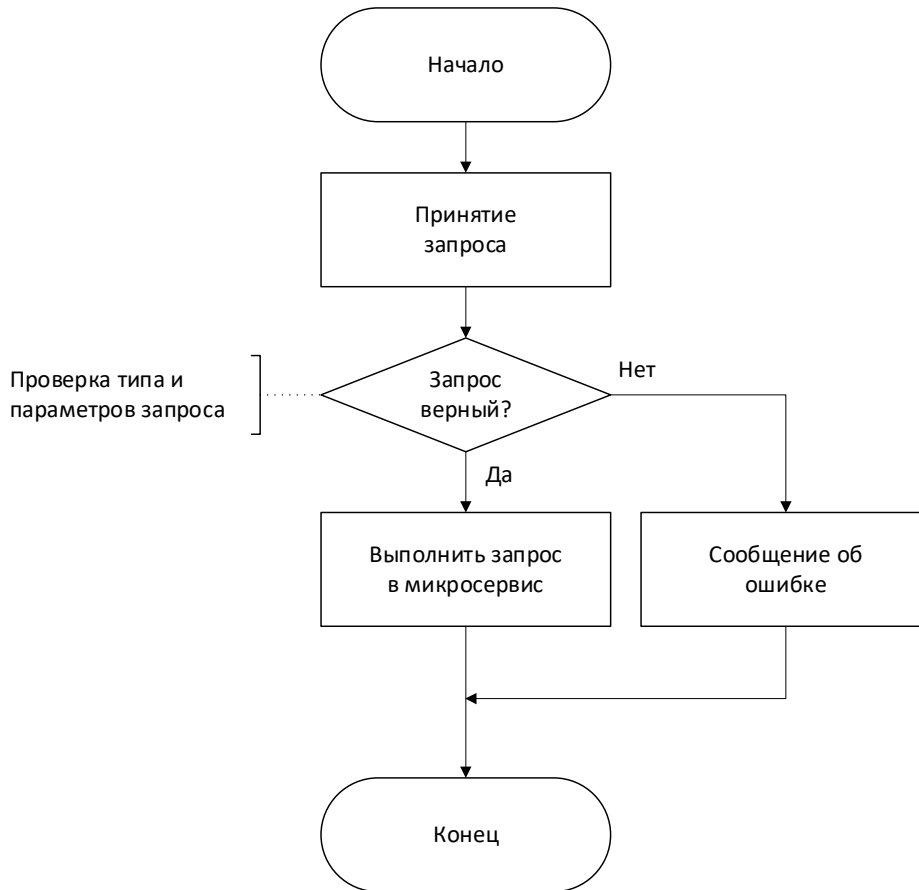


Рисунок 18 – Фрагмент блок-схемы ядра платформы

4.3. ВЫВОДЫ

В ходе реализации вычислительной платформы был разработан набор микросервисов предоставления распределенного программного обеспечения для сбора, хранения и анализа данных.

Также было разработано ядро платформы, выполняющее роль публичного API-сервера взаимодействия пользователей и платформы в туманной вычислительной среде.

5. ТЕСТИРОВАНИЕ

Тестирование является значимой стадией жизненного цикла разработки программного обеспечения. Правильно спроектированная процедура тестирования программных систем позволит увидеть ошибки разработки и вовремя предпринять решения для их исправления.

К видам тестирования программного обеспечения относятся следующие:

- функциональное;
- интеграционное;
- системное;
- тестирование производительности;
- регрессионное;
- модульное;
- тестирование безопасности.

Наиболее важным для реализации программных систем является функциональное тестирование. В частности, при использовании в разрабатываемой программной системе функций сетевого взаимодействия, функциональное тестирование расширяется до тестирования функций и программных интерфейсов.

Уровень программных интерфейсов API является одним из наиболее важных компонентов любой системы, которая имеет сетевое взаимодействие. Программный интерфейс выполняет роль канала соединения между клиентом и сервером или одним микросервисом с другим. Тестирование API позволит выявить неверные или некорректные ответы, отследить работоспособное состояние как программного клиента, так и программного сервера.

5.1. МЕТОДОЛОГИЯ ТЕСТИРОВАНИЯ

Для выполнения тестирования воспользуемся платформой Postman, которая позволяет разработчикам программного обеспечения проектировать, создавать и тестировать выполнение запросов к программным интерфейсам разрабатываемой системы. Платформы тестирования API Postman позволяет вводить необходимый адрес, указывать вид запроса, отслеживать получаемый ответ и его статус. Для удобства проведения тестирования Postman позволяет сохранять и группировать часто повторяющиеся запросы.

5.2. ПРОВЕДЕНИЕ ПРОЦЕДУРЫ ТЕСТИРОВАНИЯ

Для проведения тестирования программных интерфейсов выполним следующие запросы:

- POST запрос к /api/v0.1/dcs/;
- DELETE запрос к /api/v0.1/dcs/;
- POST запрос к /api/v0.1/dss/;
- DELETE запрос к /api/v0.1/dss/;
- PUT запрос к /api/v0.1/dss/;
- POST запрос к /api/v0.1/dms/;
- PUT запрос к /api/v0.1/dms/;
- DELETE запрос к /api/v0.1/dms/.

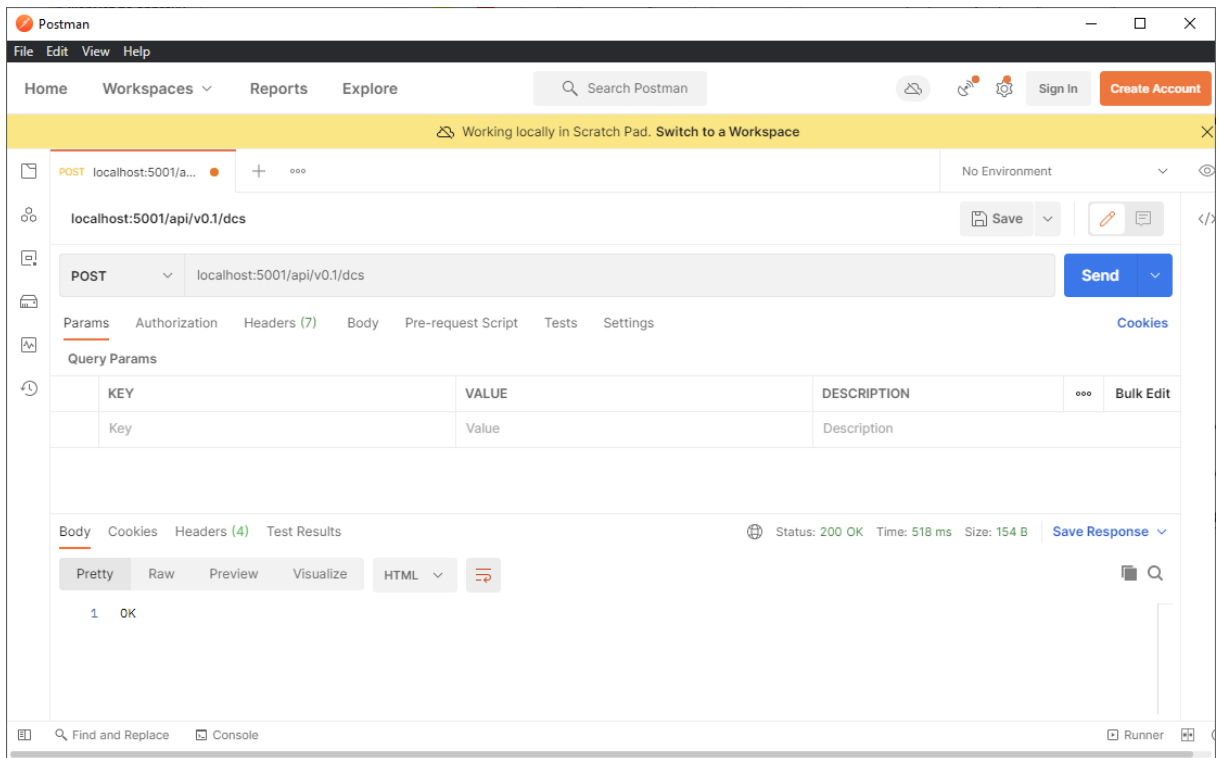


Рисунок 19 – POST запрос к `/api/v0.1/dcs/`

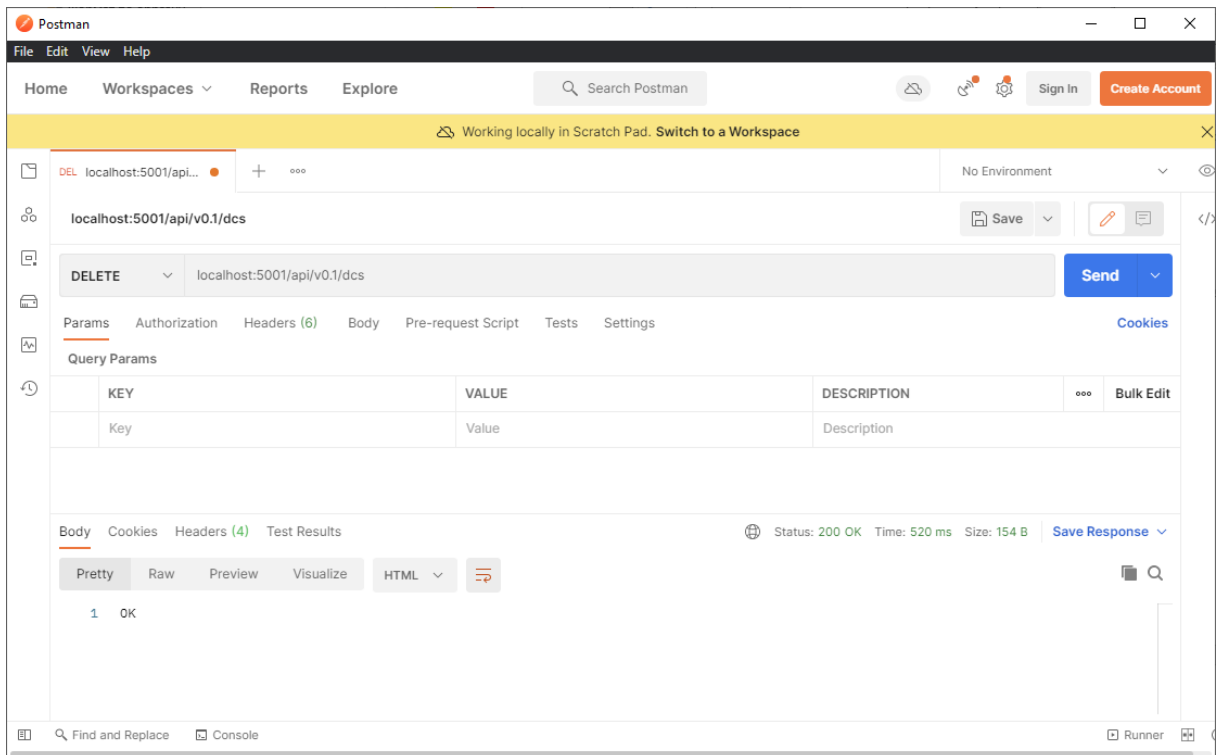


Рисунок 20 – DELETE запрос к `/api/v0.1/dcs/`

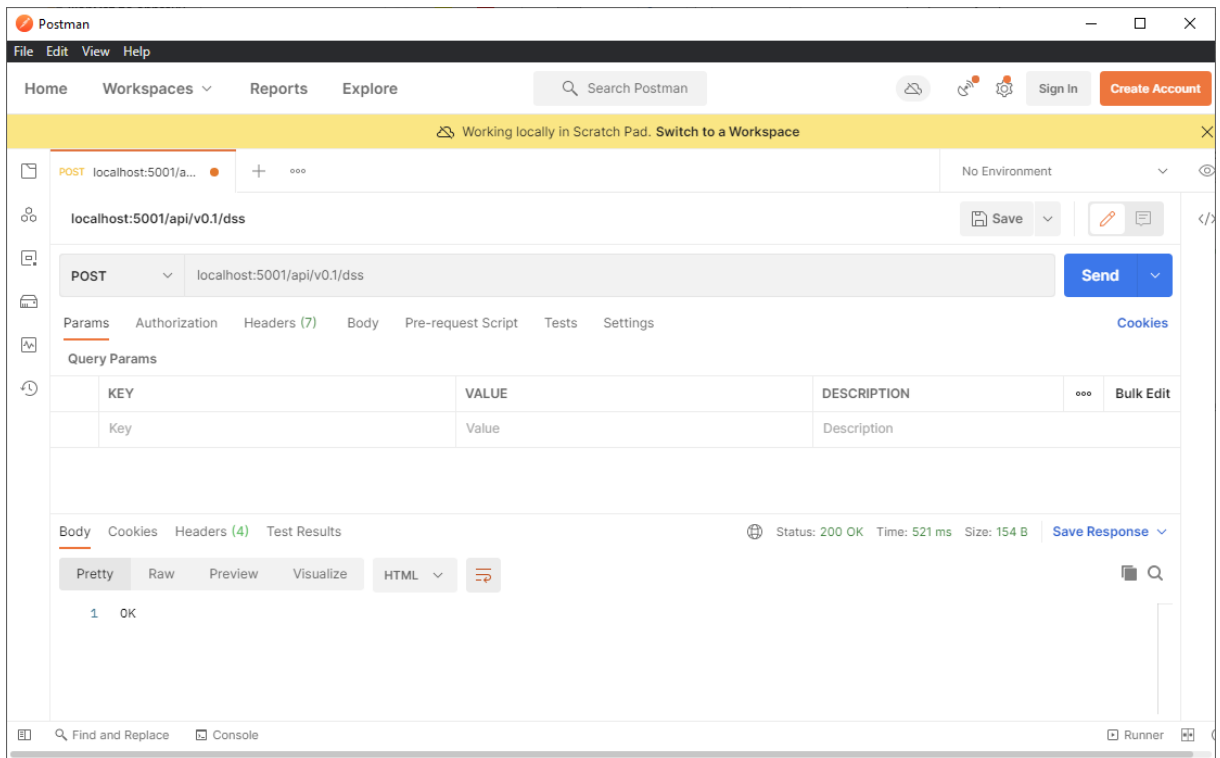


Рисунок 21 – POST запрос к `/api/v0.1/dss/`

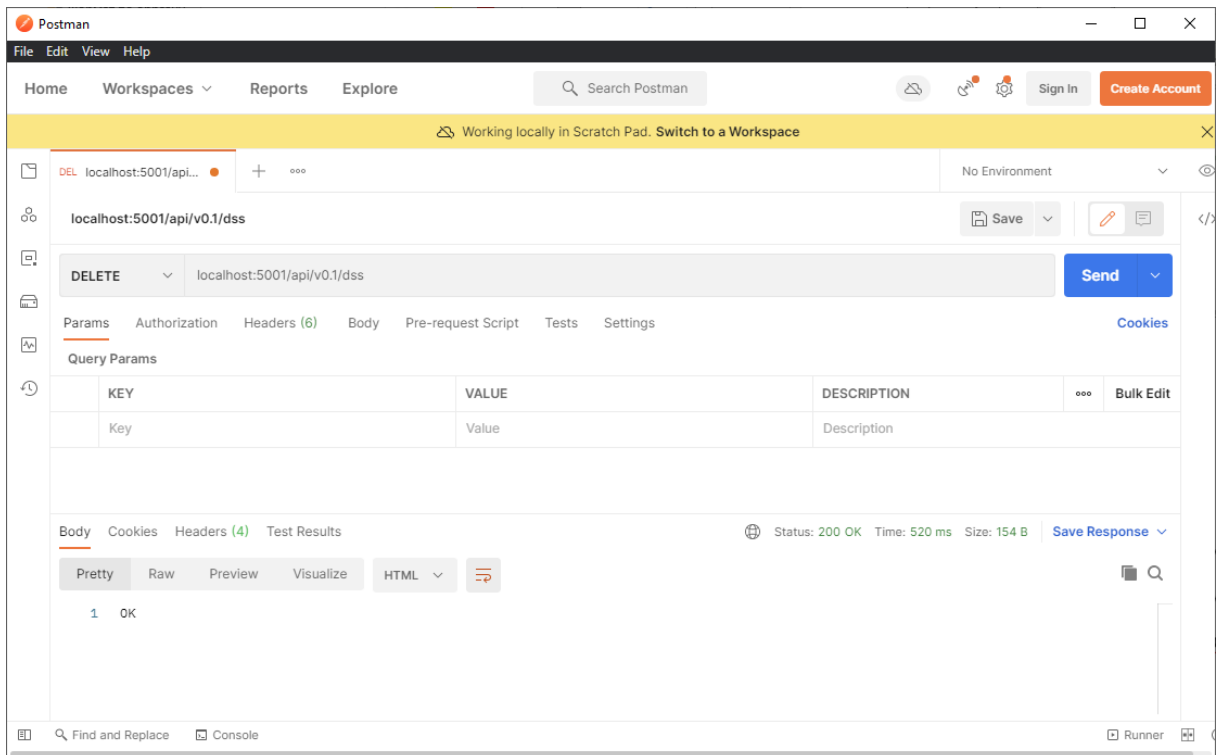


Рисунок 22 – DELETE запрос к `/api/v0.1/dss/`

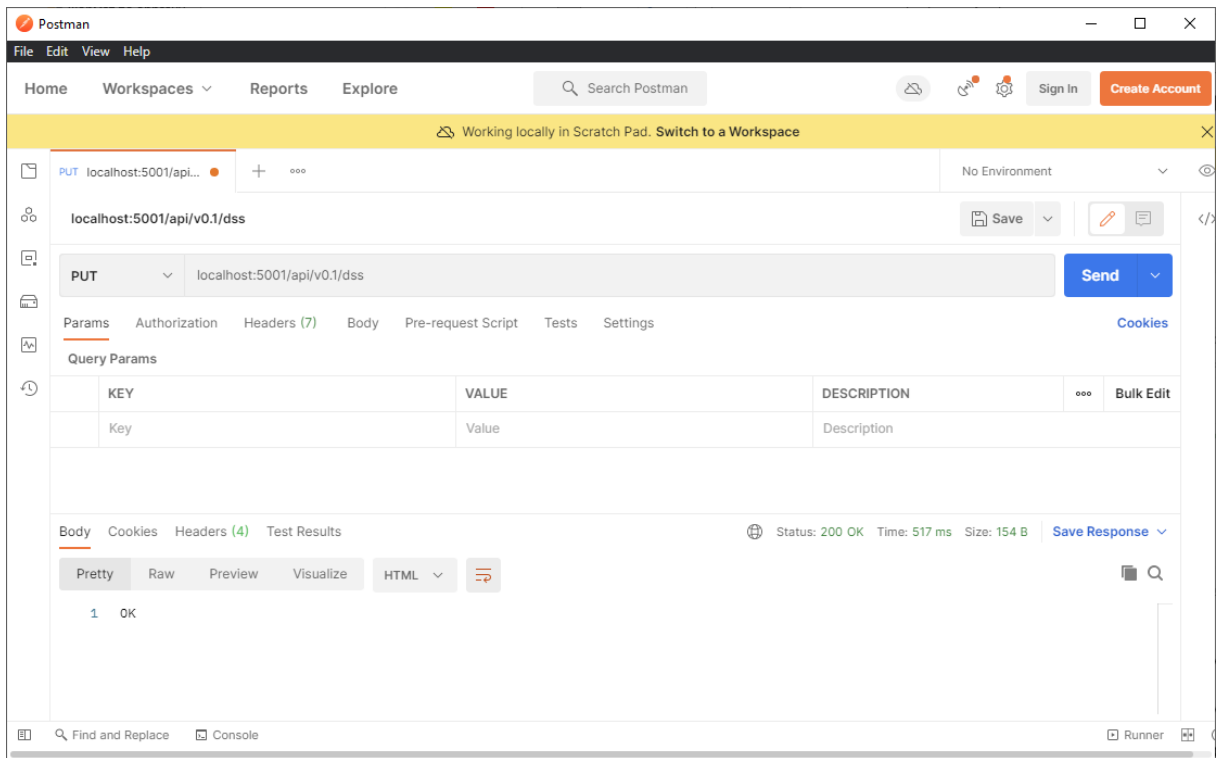


Рисунок 23 – PUT запрос к /api/v0.1/dss/

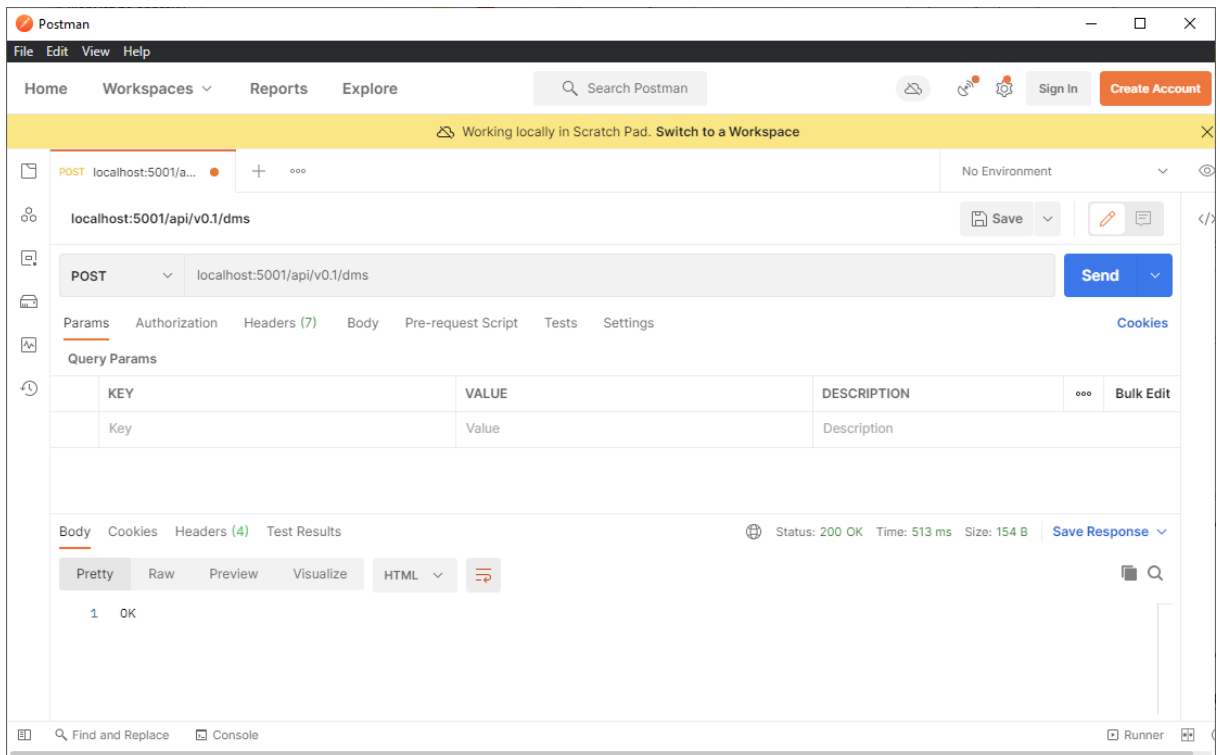


Рисунок 24 – POST запрос к /api/v0.1/dms/

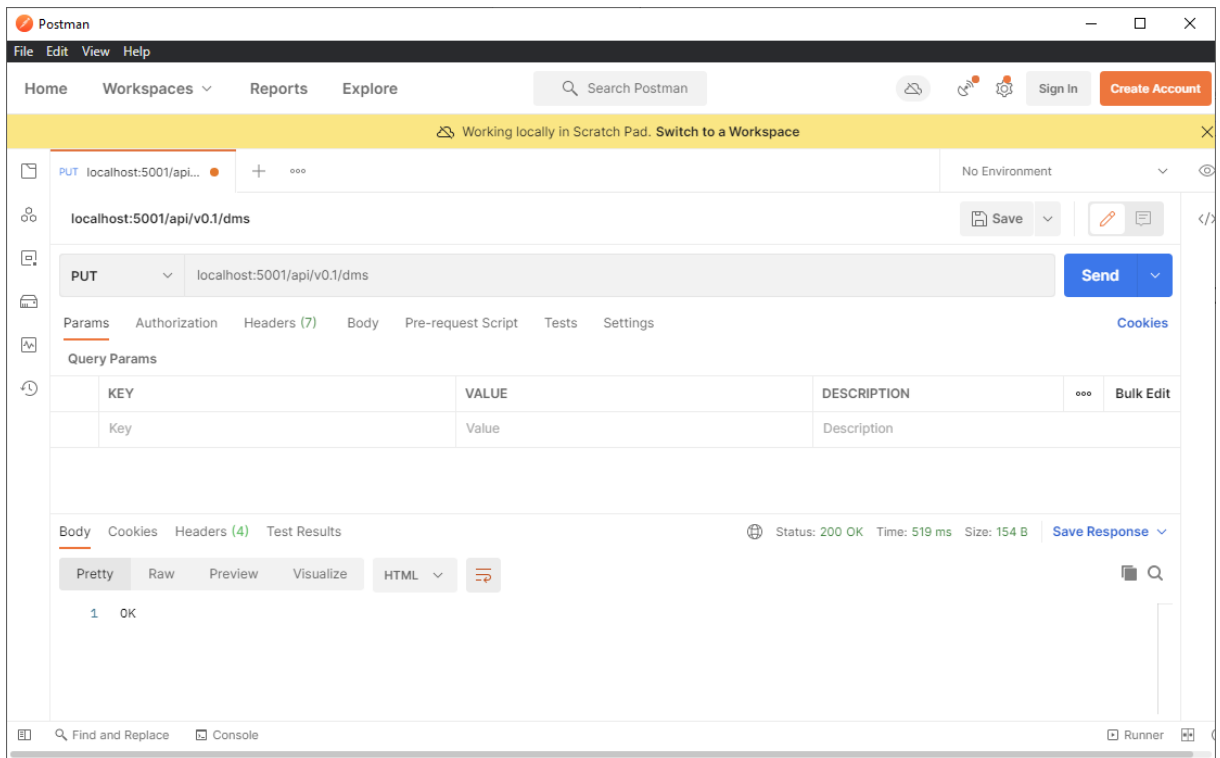


Рисунок 25 – PUT запрос к `/api/v0.1/dms/`

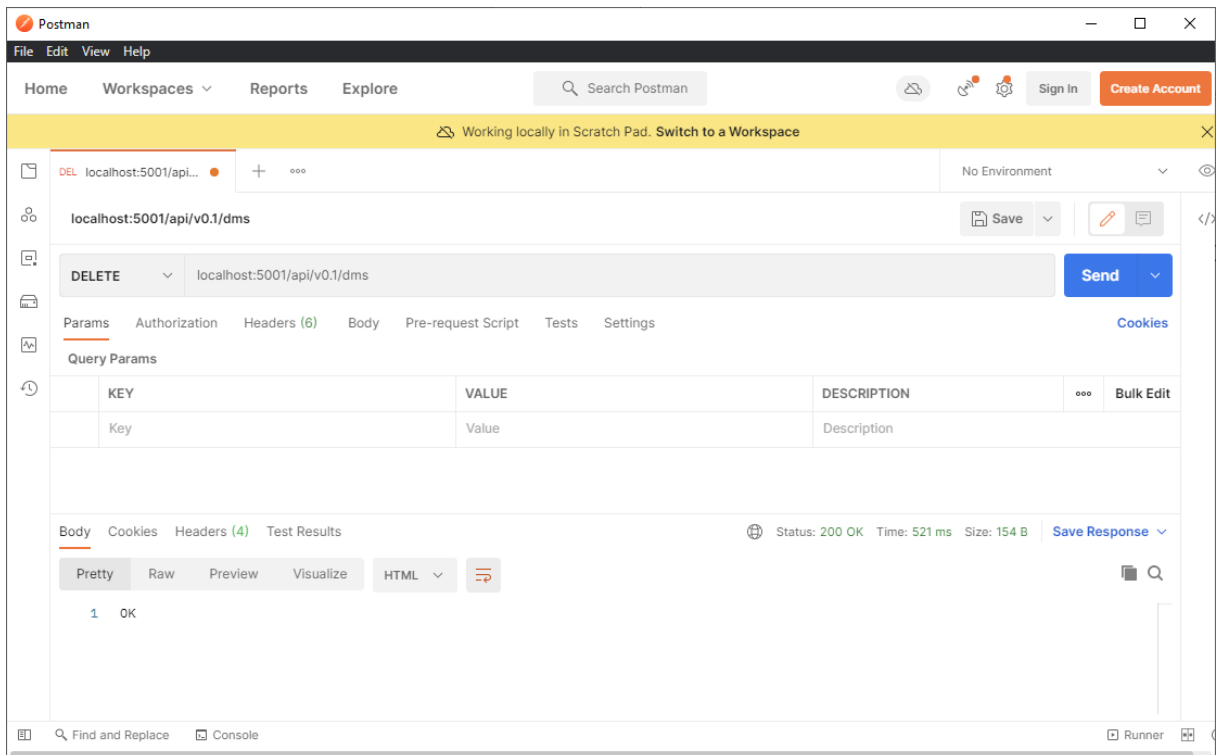


Рисунок 26 – DELETE запрос к `/api/v0.1/dms/`

Как можем видеть из рисунков 23-30, выполненные запросы вернули ожидаемый ответ и статус от API сервера ядра платформы.

5.3. ВЫВОДЫ

В ходе проведения тестирования были выполнены запросы к ядру платформы. Все запросы вернули ожидаемый результат и статус. Тестирование прошло успешно.

6. ОЦЕНКА ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ

Для проведения оценки экономической эффективности инвестпроекта необходимо:

- составить организационный план предприятия;
- провести расчет потребности в инвестициях;
- составить прогнозный отчет о движении денежных средств;
- сделать прогноз финансовых результатов проекта;
- провести расчет показателей эффективности инвестиционного проекта.

6.1. ОРГАНИЗАЦИОННЫЙ ПЛАН

Организационный план – это элемент бизнес-плана, в котором отображается состав участников, партнеров, нанятых на постоянной основе и временных специалистов с перечислением предполагаемого оклада и необходимого количества человек. Организационная структура создаваемой в рамках инвестиционного проекта компании носит функциональный характер. Это позволит рационально распределить все задачи, которые необходимо решить в рамках проекта, по соответствующим специалистам. Целью такого распределения будет повышения эффективности использования человеческого капитала. Также отметим, что команда разработчиков будет работать по проектному принципу. Занятость сотрудников на начальном этапе выполнения будет не выше среднего уровня. Именно поэтому мы имеем потенциал роста количества задач и проектов в рамках приглашаемого на данном этапе количества специалистов. Заработная плата будет повременной в виде оклада. Премияльный фонд будет формироваться в будущем на основе реальных результатов реализации проекта. На данном этапе работы премияльный фонд не предусмотрен.

Отообразим потребность в специалистах в таблице 5.

Таблица 5 – Организационный план.

№	Наименование должности	Оклад в месяц, тыс. руб.	Количество, чел.	Сумма в месяц, тыс. руб.
1	Генеральный директор	100	1	100
2	Исполнительный директор	80	1	80
3	Менеджер по продажам	80	1	80
4	Менеджер по подбору персонала	30	1	30
5	Бухгалтер	30	1	30
6	Юрист	30	1	30
7	Маркетолог	30	1	30
8	Разработчик платформы ТВП	120	1	120
9	Разработчик платформы ЦД	120	1	120
10	Руководитель команды разработки	120	1	120
11	Разработчик визуального моделирования	80	1	80
12	Fullstack разработчик	150	1	150
13	Мобильный разработчик	100	1	100
14	Администратор платформы ТВП	80	1	80
15	Инженер по внедрению	80	1	80
			Итого	1230

На основе данных таблицы, мы видим, что потребность в персонале составляет 15 человек: из них 4 человека являются руководителями высшего или среднего звена, 5 человек – специалисты, которые обеспечивают работу компании, 6 человек – специалисты, занимающиеся основным видом

деятельности компании. Общий фонд оплаты труда составляет 1 230 000 руб. в месяц.

6.2. ПОТРЕБНОСТЬ В ИНВЕСТИЦИЯХ

Потребность в инвестициях, которая может быть определена как спрос, выраженный в денежной форме, в приобретении товаров и услуг инвестиционного назначения.

Для запуска проекта необходимо будет арендовать помещение и осуществить покупку мебели и оборудования. Стоимость закупаемой мебели и компьютерного оборудования будет учтена при расчете потребности в инвестициях. Арендная плата не включена в расчет инвестиций в связи с тем, что для запуска проекта не нужно много времени. Поэтому целесообразнее арендные платежи включить в себестоимость текущей деятельности и учитывать при расчете финансовых результатов.

Текущая инвестиционная потребность определена и представлена в таблице 6.

Таблица 6 – Потребность в инвестициях.

№	Наименование	Цена, тыс. руб.	Количество, шт.	Сумма, тыс. руб.
1	Вычислительный кластер	500	2	1000
2	Компьютерный стол	10	15	150
3	Компьютерный стул	10	15	150
4	Персональный компьютер	105	15	1575
			Итого	2875

На основе данных таблицы, мы видим, что базовые инвестиционные затраты составляют 2 875 000 руб. Предполагается использование привлеченных

источников финансирования проекта. Велика вероятность получения гранта или приглашение инвесторов под эмиссию акций или облигаций. В работе рассчитан вариант использования гранта в качестве источника финансирования.

6.3. ОТЧЕТ О ДВИЖЕНИИ ДЕНЕЖНЫХ СРЕДСТВ

Отчет денежных средств – отчет компании об источниках денежных средств и их использовании за отчетный период.

Составление прогнозного отчета о движении денежных средств позволяет сделать вывод о финансовой реализуемости данного проекта. Проект является финансово реализуемым, если в каждом периоде времени сальдо накопительным итогом является положительным. Отметим также, что юридические лица, сдающие традиционную бухгалтерскую отчетность в налоговые органы по итогам своей деятельности, оформляют отчет о движении денежных средств в виде формы 4. В ходе составления формы 4 анализ идет по трем видам деятельности:

- операционная деятельность предприятия;
- инвестиционная деятельность;
- финансовая деятельность.

Операционная деятельность предприятия учитывает денежные потоки от основной деятельности компании согласно ОКВЭД. Включает в себя притоки в виде выручки (поступления от продаж), а также оттоки в виде затрат (зарплата, аренда, страховые взносы, коммунальные платежи, реклама и т.д.). Инвестиционная деятельность включает в себя оттоки по инвестиционной деятельности и, как правило, равна сальдо от инвестиционной деятельности и соответствует потребности в инвестициях. Финансовая деятельность отражает источники финансирования проекта.

Составим прогнозный отчет о движении денежных средств (таблица 7):

Таблица 7 – Отчет о планируемом движении денежных средств.

Наименование	Период 0	Период 1	Период 2	Период 3
Поступление от продаж		30000,00	60000,00	90000,00
ФОТ		19188,00	22066,20	25376,13
Интернет		120,00	138,00	158,70
Аренда		780,00	897,00	1031,55
Операционная деятельность		9912,00	36898,80	63433,62
Инвестиции	-2875,00			
Инвестиционная деятельность	-2875,00	0,00	0,00	0,00
Грант	5000,00			
Финансовая деятельность	5000,00	0,00	0,00	0,00
Сальдо на конец периода	2125,00	9912,00	36898,80	63433,62
Сальдо накопленным итогом	2125,00	12037,00	48935,80	112369,42

На основе прогнозного отчета о движении денежных средств видим, что проект является финансово реализуемым, так как накопленное сальдо на каждом этапе реализации проекта является положительным.

6.4. ОТЧЕТ О ФИНАНСОВЫХ РЕЗУЛЬТАТАХ

Отчет о финансовых результатах (форма №2 бухгалтерской отчетности в Российской Федерации) характеризует финансовые результаты деятельности организации за отчетный период.

Для расчета прогноза финансовых результатов проекта необходимо рассчитать прогнозные показатели выручки и себестоимости. Прогноз выручки основан на предварительных договоренностях с потенциальными клиентами. Прогноз себестоимости является следствием расчета затрат осуществления текущей основной деятельности. Фонд оплаты труда рассчитан на основе запланированных показателей по заработной плате за месяц по всем сотрудникам, а также страховых взносов. Амортизация рассчитывается на основе первоначальной стоимости основных средств, а также срока полезного использования оборудования. Арендные платежи также включаются в себестоимость по основной деятельности. Коммерческие и управленческие расходы включают в себя ФОТ и страховые взносы соответствующих специалистов.

Страховые взносы составляет 30% от фонда оплаты труда и являются обязательными. Также стоит помнить, что компания является налоговым агентом по НДФЛ и обязана взимать и перечислять его налоговым органам.

IT-компаниям в существующих условиях получают поддержку от государства, а значит, освобождены от уплаты налога на прибыль.

Валовая прибыль рассчитывается как разница между выручкой и себестоимостью. Если из валовой прибыли вычесть коммерческие и управленческие расходы, то мы получим прибыль от продаж.

В таблице 8 определены и представлены прогнозируемые финансовые результаты.

Таблица 8 – Отчет о финансовых результатах.

Наименование	Период 0	Период 1	Период 2	Период 3
Выручка		30000,00	60000,00	90000,00
Себестоимость		15018,00	17142,00	19585,60
ФОТ		13260,00	15249,00	17536,35
Амортизация вычислительного кластера		333,00	333,00	334,00
Амортизация компьютерной техники		525,00	525,00	525,00
Аренда		780,00	897,00	1031,55
Интернет		120,00	138,00	158,70
Валовая прибыль (убыток)	0,00	14982,00	42858,00	70414,40
Коммерческие расходы, включая страховые взносы		1716,00	1973,40	2269,41
Управленческие расходы, включая страховые взносы		4212,00	4843,80	5570,37
Прибыль (убыток) от продаж	0,00	9054,00	36040,80	62574,62
Налог на прибыль	0,00	0,00	0,00	0,00
Чистая прибыль	0,00	9054,00	36040,80	62574,62

На основе данных таблицы видно, что прогноз по финансовым результатам положительный.

6.5. РАСЧЕТ ПОКАЗАТЕЛЕЙ ЭФФЕКТИВНОСТИ ИНВЕСТПРОЕКТА

Чтобы провести оценку эффективности инвестиционного проекта, нужно рассчитать следующие показатели:

- чистый денежный поток от операционной деятельности;
- чистый денежный поток от инвестиционной деятельности;
- чистый денежный поток;
- дисконтированный чистый денежный поток;
- дисконтированный поток нарастающим итогом;
- чистую приведенную стоимость;
- внутреннюю норму доходности;
- индекс рентабельности.

Первые три показателя уже рассчитаны в таблице 7.

NPV или чистая (текущая) стоимость – характеризует изменение потоков денежных средств.

IRR или внутренняя норма прибыли (доходности) – представляет собой максимально допустимую стоимость вложений, при которой показатель чистой стоимости равен нулю.

PI или индекс рентабельности инвестиций – показатель метода чистой приведенной стоимости.

DPP или дисконтированный срок окупаемости – срок окупаемости, учитывающий разную ценность финансовых средств при поступлении и выплат во времени.

Расчет показателей эффективности проекта представлен в таблице 9.

Таблица 9 – Расчет показателей эффективности инвестпроекта.

Наименование	Период 0	Период 1	Период 2	Период 3
Чистый денежный поток от операционной деятельности	0,00	9912,00	36898,80	63433,62
Чистый денежный поток от инвестиционной деятельности	-2875,00	0,00	0,00	0,00
Чистый денежный поток	-2875,00	9912,00	36898,80	63433,62
Дисконтированный чистый денежный поток	-2875,00	7080,00	18825,91	23117,20
Дисконтированный поток нарастающим итогом	-2875,00	4205,00	23030,91	46148,12

Для расчета чистой приведенной стоимости используются показатели чистого денежного потока от операционной и инвестиционной деятельности. Рассчитанный чистый денежный поток дисконтируется. Таким образом, мы получаем показатель дисконтированного потока нарастающим итогом.

Ставка дисконтирования была рассчитана на основе кумулятивного метода и равна 40 % и складывается из минимальной безрисковой ставки доходности 10% (ставка сбербанка на 2022 год), уровне инфляции 20% (ставка на 2022 год) и рискованной премии 10% (научная прикладная разработка).

В таблице 10 представлены рассчитанные в работе показатели эффективности инвестиционного проекта.

Таблица 10 – Показатели эффективности инвестпроекта.

Наименование	Показатель
Ставка дисконтирования	40,00
NPV	46148,12
IRR, %	338,00
PI	17,05
DPP	0,41

Проведенный в работе анализ эффективности проекта показал, что проект является экономически выгодным и эффективным. Показатель NPV больше нуля, IRR больше ставки дисконтирования, PI больше единицы, а дисконтированный срок окупаемости проекта меньше срока его реализации. Такой большой запас в показателях эффективности инвестиционного проекта является необходимостью в условиях функционирования повышенных рисков, связанных с существующими экономическими условиями.

6.6. ВЫВОДЫ

Оценка экономической эффективности инвестпроекта показала, что планируемый стартап является экономически выгодным и финансово реализуемым. Показатели эффективности, представленные в таблице 10, доказывают достаточный уровень внутренней нормы доходности, что является весомым аргументом к устойчивости планируемого стартапа.

7. ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы была спроектирована и реализована вычислительная платформы для сбора, хранения, анализа данных интернета вещей в туманной вычислительной среде.

Вычислительная платформа архитектурно составляет набор взаимосвязанных микросервисов, реализованных на языке программирования Python 3 и веб-фреймворке Flask с применением технологий контейнерной виртуализации Containerd и контейнерной оркестрации Kubernetes.

Оценка экономической эффективности инвестпроекта показала, что организация планируемого стартапа является экономически прибыльной. Точка безубыточности достигается в первый год.

Для достижения цели были выполнены следующие задачи:

- выполнен обзор литературы;
- определены требования;
- спроектирована и реализована вычислительная платформа;
- проведено тестирование API;
- выполнен расчет экономической эффективности инвестпроекта;
- проведен анализ результатов.

В результате выполнения выпускной квалификационной работы была разработана программная система, которая, организуя сбор, хранение и анализ данных с устройств интернета вещей, позволит организовывать цифровые двойники промышленных предприятий.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Асташов, А.А. Выпускная квалификационная работа бакалавра на тему «Исследование эффективности платформ управления вычислительными сервисами при организации туманных вычислений» / А.А. Асташов. – Россия: Челябинск, Южно-Уральский Государственный Университет 2020. – 45 с.
2. Ли, П. Архитектура интернета вещей / П. Ли; пер. с англ. М. А. Райтмана. – М.: ДМК Пресс, 2019. – 454 с.
3. IoT, туман и облака: поговорим про технологии. [Электронный ресурс] URL: <https://habr.com/ru/company/cloud4y/blog/467711/> (дата обращения: 01.05.2022)
4. Зачем облака спускаются на землю. [Электронный ресурс] URL: <https://mcs.mail.ru/blog/why-clouds-get-closer-to-the-ground> (дата обращения: 01.05.2022)
5. Довгаль, В.А. Роль туманных вычислений в Интернете Вещей / Ежеквартальный рецензируемый, реферируемый научный журнал «Вестник АГУ». Выпуск 4(231), 2018. – С. 205–209.
6. Evaluating Container Platforms at Scale. [Электронный ресурс] URL: <https://medium.com/on-docker/evaluating-container-platforms-at-scale-5e7b44d93f2c/> (дата обращения: 01.05.2022)
7. Hoque, S. Towards Container Orchestration in Fog Computing Infrastructures / S. Hoque, M.S. de Brito, T. Magedanz, A. Willner, O. Keil // IEEE 41st Annual Computer Software and Applications Conference. – 2017. – P. 294–299.
8. Virtualization Technology & Virtual Machine Software: What is Virtualization? [Электронный ресурс] URL:

- <https://www.vmware.com/ru/solutions/virtualization.html> (дата обращения: 01.05.2022)
9. Ажиотаж вокруг контейнеров. [Электронный ресурс] URL: <https://www.osp.ru/lan/2014/10/13043208/> (дата обращения: 01.05.2022)
 10. Гордеев, А.В. Сравнительное тестирование контейнерной и гипервизорной виртуализации / А.В. Гордеев, Д.В. Горелик, СПбГУАП, Информационно-управляющие системы №2, 2018. – С. 60–66.
 11. Упакуем все – зачем нужны контейнеры и как с ними работать в Big Data. [Электронный ресурс] URL: <https://medium.com/@bigdataschool/упакуем-все-зачем-нужны-контейнеры-и-как-с-ними-работать-в-big-data-256acd15fd8f> (дата обращения: 01.05.2022)
 12. Зачем и как использовать контейнеры – разбираемся с Docker, Kubernetes и другими инструментами. [Электронный ресурс] URL: <https://tproger.ru/articles/containers-explained/amp/> (дата обращения: 01.05.2022)
 13. Моуэт, Э. Использование Docker / Э. Моуэт; пер. с англ. А.В. Снастина; науч. ред. А. А. Маркелов. – М.: ДМК Пресс, 2017. – 354 с.
 14. de Brito, M. S. A Service Orchestration Architecture for Fog-enabled Infrastructures / M. S. de Brito, S. Hoque, T. Magedanz R. Steinke, A. Willner, D. Nehls, O. Keils, F. Schreiner // 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC). – 2017. – P. 127–132.
 15. Khan A. Key Characteristics of a Container Orchestration Platform to Enable a Modern Application / IEEE Cloud computing. – 2017. – P. 42–48.
 16. Электронная документация Docker. [Электронный ресурс] URL: <https://docs.docker.com/> (дата обращения: 01.05.2022)
 17. Электронная документация Docker Swarm. [Электронный ресурс] URL: <https://docs.docker.com/engine/swarm/> (дата обращения: 01.05.2022)

18. Электронная документация Apache Kafka. [Электронный ресурс] URL: <https://kafka.apache.org/documentation/> (дата обращения: 01.05.2022)
19. Электронная документация Kubernetes. [Электронный ресурс] URL: <https://kubernetes.io/ru/docs/home/> (дата обращения: 01.05.2022)
20. Электронная документация RabbitMQ. [Электронный ресурс] URL: <https://www.rabbitmq.com/documentation.html> (дата обращения: 01.05.2022)
21. Алексеева, М.М. Планирование деятельности фирмы. - М.: Финансы и статистика, 2011. – 412 с.
22. Басовский Л.Е. Прогнозирование и планирование в условиях рынка. Учебное пособие. – М.: ИНФРА – М, 2011.
23. Багиев, Г.Л., Тарасевич, В.М., Анн, Х. Маркетинг: Учебник для вузов 3-е изд. – СПб.: Питер, 2011. – 736 с.
24. Бизнес-план. Книга менеджера. - М.: Финансы и статистика, 2012. – 362 с.
25. Бизнес-планирование: Учебник / Под ред. И. Попова. - М.: Финансы и статистика, 2013. – 672 с.
26. Бизнес-план. Методические материалы / Под ред. Р.Г. Маниловского. – М.: Финансы и статистика, 2012. – 212 с.
27. Бизнес-план компании / Под ред. Уткина Э.А. – М.: Тандем, 2013. – 100 с.
28. Бизнес-план. Методические материалы под ред. проф. Р.Г. Маниловского - М. Финансы и статистика, 2012. – 156 с.