

Министерство науки и высшего образования РФ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ О.В. Непомнящий  
« \_\_\_ » \_\_\_\_\_ 2024 г.

## МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Инфраструктура студенческого конструкторского бюро.  
Развертывание сервера хранения документации

090401 Информатика и вычислительная техника

09.04.01.11 «Вычислительные системы и сети»

Руководитель	_____	_____	доцент, канд. физ.-мат. наук _____	К.В. Коршун
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	
Выпускник	_____	_____	_____	А.А. Погосян
	<i>подпись</i>	<i>дата</i>		
Рецензент	_____	_____	доцент _____	П.В. Кочкин
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	
Нормоконтролер	_____	_____	доцент, канд. физ.-мат. наук _____	К.В. Коршун
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	

Красноярск 2024

## СОДЕРЖАНИЕ

Введение.....	3
1 Исследование существующих подходов к организации инфраструктуры хранения.....	5
1.1 Постановка задачи.....	5
1.2 Инструменты для реализации решения.....	8
1.2.1 Организация учетных записей пользователей.....	9
1.2.2 Решения для совместной разработки.....	14
1.2.3 Почтовый сервис.....	16
1.3 Выводы по главе.....	19
2 Разработка инфраструктуры студенческого конструкторского бюро.....	21
2.1 Техническое задание.....	21
2.2 Общая структура.....	21
2.3 Разработка подсистемы централизованной аутентификации.....	28
2.4 GitLab.....	34
2.5 Разработка почтового сервиса.....	37
2.6 Выводы по главе.....	44
3 Обоснование и выбор аппаратного обеспечения.....	45
3.1 Структурная схема системы разработанной модели.....	45
3.2 Реализация серверной инфраструктуры.....	46
3.3 Выводы по главе.....	47
4 Описание методики тестирования.....	48
4.1 Результаты тестирования LDAP.....	48
4.2 Результаты тестирования GitLab.....	50
4.3 Результаты тестирования iRedMail.....	56
4.4 Выводы по главе.....	59
Заключение.....	60
Список сокращений.....	62
Список используемых источников.....	63

## ВВЕДЕНИЕ

**Актуальность проблемы:** студенческое конструкторское бюро – это организация, объединяющая студентов, занимающихся научной и проектной деятельностью в качестве факультативных занятий. Участие в подобной организации может дать студентам навыки работы над большими наукоёмкими проектами в составе команды под руководством преподавателей и работодателей. Также при подобной деятельности у студентов появляется опыт работы с практическими задачами.

Для формирования рабочей среды, приближенной к реальному предприятию или компании, поставлена задача создать инфраструктуру студенческого конструкторского бюро. Подобные компании оперируют мультидисциплинарными проектами, где участвуют специалисты различных профилей – от экономистов и планировщиков до алгоритмистов и разработчиков радиоэлектронной аппаратуры [1].

Многие решения не позволяют в полной мере реализовать весь спектр задач для каждого предприятия, поскольку они являются уникальными.

Анализ проблем в данной области показал отсутствие Российских систем управления ИТ-предприятием, гибко настраиваемых на требуемый класс решаемых задач [2]. В большинстве своем они являются платными и закрытыми и обладают избыточностью, ориентированы на крупное промышленное предприятие и слабо подходят для управления небольшими группами разработчиков и конструкторских бюро.

Таким образом можно сделать обоснованное предположение о том, что результаты могут быть получены в интеграции известных средств, которые необходимы для совместной работы разработчиков:

- система контроля версий;
- средство коммуникации в виде почтового сервиса и другие решения.

Организация такой инфраструктуры в студенческом конструкторском бюро позволит приблизить процесс работы студентов к таковому на рабочих

местах, что даст студентам полезные навыки и практический опыт для последующего трудоустройства. Таким образом, тема магистерской диссертации является актуальной.

**Цель диссертационной работы:** создание и разработка инфраструктуры студенческого конструкторского бюро. Для достижения назначенной цели сформулированы следующие основные задачи исследования:

- выполнить анализ известных программных инструментов;
- разработать инфраструктуру студенческого конструкторского бюро;
- обосновать и выбрать аппаратное обеспечение;
- выполнить тестирование режимов работы серверной, аппаратной и программной части.

**Предполагаемая научная новизна исследования** заключается в предложенной архитектуре инфраструктуры студенческого конструкторского бюро, базирующейся на выбранном программном обеспечении для сервера, сервисов для аутентификации, хранения кода и системы контроля версий, а также для почтового сервера. Предложенная архитектура позволит организовать рабочий процесс участников студенческого конструкторского бюро, позволяя им эффективно работать над проектами, получая практический опыт в разработке программного обеспечения и решении сложных задач. Все это создает условия для совместной работы и обмена знаниями между участниками конструкторского бюро, способствуя развитию коллективного мышления и повышению эффективности работы студентов.

# **1 Исследование существующих подходов к организации инфраструктуры хранения**

В современном мире организации сталкиваются с необходимостью эффективного хранения, управления и обмена документацией. Организации любого масштаба и профиля деятельности стремятся к созданию инфраструктуры, которая позволит им эффективно работать с документами, обеспечивая доступность, безопасность и удобство использования.

Исследование подходов к организации инфраструктуры хранения документации позволяет выявить различные методы и инструменты, применяемые в современном бизнесе. Различные организации могут использовать разнообразные системы управления документами, облачные хранилища, сетевые файловые системы и другие технологии для обеспечения требуемого уровня хранения и доступа к документам.

В рамках данного исследования будет проведен анализ существующих подходов к организации инфраструктуры хранения документации в организациях различного типа и масштаба. Будут рассмотрены преимущества и недостатки различных методов, а также выявлены основные требования, которые предъявляются к инфраструктуре хранения документов в современном бизнесе.

## **1.1 Постановка задачи**

В современных компаниях существует необходимость в улучшении координации работы различных отделов, особенно в области информационных технологий, где проекты требуют участия специалистов разных профилей - от экономистов и планировщиков до разработчиков программного обеспечения и инженеров по разработке электроники.

Существующие решения не всегда отвечают всем потребностям компаний из-за их уникальных характеристик и требований. Проведенный анализ показал,

что на рынке отсутствуют гибкие и настраиваемые под разные классы задач российские системы управления ИТ-предприятием. Большинство существующих решений ориентировано на крупные промышленные предприятия и не подходит для управления небольшими группами разработчиков и конструкторских бюро. Кроме того, многие из них являются платными и закрытыми, что создает препятствия для доступа к ним.

Для совместной работы над проектами необходим комплекс программ, который имеет возможность не только отслеживать процесс в комплексе, но реализовывать весь функционал, требуемый компании. Комплекс должен состоять из:

- протокол, используемый для аутентификации и авторизации пользователей в сетях и веб-приложениях;
- система коммуникаций;
- система контроля версий.

Перед разработкой схемы инфраструктуры студенческого конструкторского бюро, а также перед анализом программного обеспечения, необходимо определить, на основе какой операционной системы будет работать сервер. Рассмотрим две наиболее распространенные ОС: Windows [3] и Linux [4].

Windows – это операционная система, широко используемая во всем мире, которая предлагает пользовательский интерфейс с графической оболочкой. Windows поддерживает множество программ и приложений, включая офисные пакеты, игры и различные утилиты. Она также обладает широкой совместимостью с аппаратным обеспечением и поддерживает большое количество драйверов для различных устройств.

Linux – это операционная система с открытым исходным кодом, которая разрабатывается сообществом разработчиков по всему миру. Она предлагает гибкость и настраиваемость, что делает ее популярной среди опытных пользователей и системных администраторов. Linux также имеет различные дистрибутивы, такие как Ubuntu, Astra и CentOS. Они предлагают разные наборы

программ и настроек, чтобы соответствовать потребностям различных пользователей. Linux широко используется в серверных средах, веб-хостинге, настройке сетей и других областях, где требуется надежность и безопасность.

В рамках данной работы выбор операционной системы остановился на Linux. Ниже описаны причины данного выбора:

- свобода и открытый исходный код: Linux основан на принципах свободного программного обеспечения и имеет открытый исходный код. Это означает, что вы можете свободно использовать, изменять и распространять операционную систему без ограничений. Это также позволяет сообществу разработчиков создавать и поддерживать различные дистрибутивы, такие как Ubuntu, которые предлагают разные функции и настройки;

- гибкость и настраиваемость: Linux предлагает гибкость и настраиваемость, что позволяет пользователям настроить операционную систему под свои потребности. Вы можете выбрать различные дистрибутивы, установить только необходимые программы и настроить систему по своему усмотрению. Это особенно полезно для системных администраторов и опытных пользователей.

- стабильность и производительность: Linux известен своей стабильностью и производительностью. Он может работать без перезагрузки в течение длительного времени и обеспечивает эффективное использование ресурсов компьютера. Это делает его хорошим выбором для серверов и других задач, требующих надежности и производительности.

- большое сообщество и поддержка: Linux имеет активное сообщество разработчиков и пользователей, которые готовы помочь и поддержать друг друга. Вы можете найти множество ресурсов, форумов и документации, которые помогут вам разобраться с Linux и решить возникающие проблемы.

## 1.2 Инструменты для реализации решения

На рисунке 1 приведена разработанная схема инфраструктуры, которая будет использоваться для системы управления конструкторским бюро. Инфраструктура состоит из локальной сети конструкторского бюро и сервера, необходимого для работы программного обеспечения.

Ранее, была выбрана операционная система для сервера – Linux, но у данной ОС есть множество различных дистрибутивов. В ходе исследований различных дистрибутивов, изучении документации, а также на основе опыта работы выбор был сделан в пользу Ubuntu [5]. Ubuntu предлагает несколько преимуществ, которые делают его лучшим выбором в качестве ОС на сервере:

- широкая поддержка и активное сообщество: Ubuntu имеет одно из самых активных сообществ в мире Linux. Это означает, что вы можете легко найти руководства, документацию и помощь от опытных пользователей и разработчиков;

- простота установки и использования: Ubuntu предлагает достаточно простой процесс установки и настройки. Он также имеет интуитивно понятный интерфейс и удобные инструменты управления, что делает его доступным даже для новичков;

- обширный репозиторий программного обеспечения: Ubuntu имеет обширный репозиторий с большим количеством программного обеспечения, чего нельзя сказать об остальных дистрибутивах;

- широкая документация и руководства: Ubuntu имеет обширную документацию и руководства, которые могут помочь в установке большого количества ПО. Найти подробные инструкции и советы можно в официальной документации Ubuntu и на различных форумах.



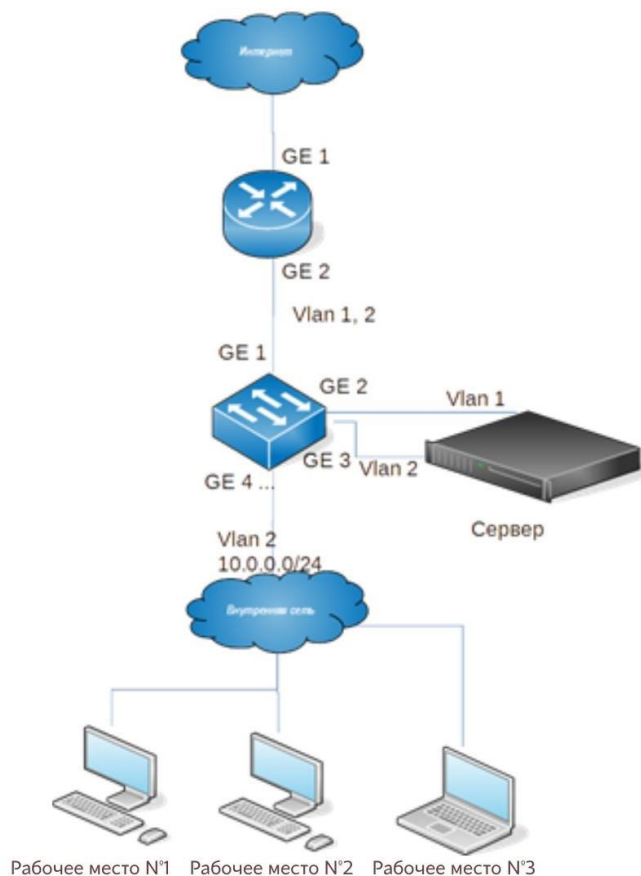


Рисунок 1 – Общая схема инфраструктуры конструкторского бюро

### 1.2.1 Организация учетных записей пользователей

Существует несколько решений в виде протоколов, используемых для аутентификации и авторизации пользователей в компьютерных сетях и веб-приложениях. Рассмотрим варианты протоколов:

1. LDAP (Lightweight Directory Access Protocol);
2. Kerberos;
3. OAuth (Open Authorization);
4. OpenID;
5. SAML (Security Assertion Markup Language);
6. JWT (JSON Web Token).

Эти протоколы предоставляют различные механизмы аутентификации и авторизации, которые могут быть использованы разработчиками для

обеспечения безопасности и удобства доступа к ресурсам в сетевых и веб-приложениях. Теперь рассмотрим каждый из них подробнее.

1. LDAP (Lightweight Directory Access Protocol) – протокол, предназначенный для доступа к каталогам, который широко используется для управления пользователями и ресурсами в сетевой среде. LDAP отличается открытостью, простотой в реализации и масштабируемостью. Он бесплатен и поддерживается широким спектром устройств и программного обеспечения [6].

Характеристики:

- протокол доступа к каталогу для поиска и аутентификации пользователей;
- обычно используется в централизованных каталогах, таких как Active Directory;
- предоставляет структурированное хранилище информации о пользователях и ресурсах.

Сложность обслуживания/установки: установка и настройка могут быть сложными, особенно для крупных организаций с множеством пользователей.

Бесплатность: LDAP протокол бесплатен, но могут возникнуть затраты на поддержку и обслуживание серверов каталогов.

Простота в работе: после настройки LDAP может быть относительно прост в использовании для аутентификации и авторизации пользователей.

2. Kerberos – протокол аутентификации, который обеспечивает безопасное взаимодействие между клиентами и серверами в распределенной среде. Он обладает высоким уровнем безопасности, но требует сложной настройки и установки [7].

Характеристики:

- протокол аутентификации, используемый для безопасной аутентификации клиентов и серверов в сетях Windows и Unix;
- использует симметричное шифрование для защиты учетных данных;

- предоставляет одноразовые билеты для аутентификации пользователей.

Сложность обслуживания/установки: установка и настройка Kerberos может вызвать сложности при управлении ключами для неопытного человека, на клиенте и сервере должно быть одинаковое время, иначе не будут работать цифровые сертификаты, используемые протоколом.

Бесплатность: Kerberos является открытым протоколом и бесплатно доступен для использования.

Простота в работе: Kerberos обеспечивает высокий уровень безопасности, но может потребоваться определенный уровень экспертизы для правильной настройки и использования.

3. OAuth (Open Authorization) – протокол авторизации, который позволяет делегировать доступ к ресурсам от имени пользователя без предоставления пароля [8]. Он широко используется в социальных сетях и сервисах авторизации, но может быть сложен в реализации из-за необходимости настройки разрешений.

Характеристики:

- протокол авторизации, который позволяет пользователям предоставлять доступ к своим ресурсам третьим сторонам без передачи своих учетных данных;

- широко используется в веб-приложениях и API для предоставления доступа к данным сторонним приложениям.

Сложность обслуживания/установки: Реализация OAuth может быть сложной из-за необходимости обеспечить безопасность и авторизацию веб-приложений.

Бесплатность: OAuth является открытым стандартом и бесплатно доступен для использования.

Простота в работе: OAuth может быть относительно прост в использовании для разработчиков веб-приложений, но требует правильной настройки и безопасности.

4. OpenID – стандарт аутентификации, который позволяет пользователям использовать один аккаунт для доступа ко множеству ресурсов [9]. Он удобен в использовании, но может иметь некоторые ограничения в функциональности.

Характеристики:

- протокол аутентификации, который позволяет пользователям использовать одни и те же учетные данные для доступа к нескольким веб-сайтам;

- предоставляет средства для однократной аутентификации пользователей без необходимости создания новых учетных записей.

Сложность обслуживания/установки: реализация OpenID может быть сложной из-за необходимости интеграции с различными веб-сервисами и учетными записями пользователей.

Бесплатность: OpenID является открытым стандартом и бесплатно доступен для использования.

Простота в работе: OpenID может быть удобен для пользователей, так как позволяет им использовать одни и те же учетные данные на различных сайтах.

5. SAML (Security Assertion Markup Language) – протокол обмена данными для безопасной аутентификации и авторизации между веб-сервисами [10]. Он обеспечивает высокий уровень безопасности, но требует специализированных знаний для настройки и работы.

Характеристики:

- протокол одностороннего обмена утверждениями, используемый для аутентификации и авторизации пользователей в веб-сервисах и приложениях;

- обычно используется для однократной аутентификации и передачи утверждений между идентификационными провайдерами и службами.

Сложность обслуживания/установки: реализация SAML может быть сложной из-за необходимости настройки и согласования между различными системами.

Бесплатность: SAML является открытым стандартом и бесплатно доступен для использования.

Простота в работе: SAML обеспечивает высокий уровень безопасности, но может потребоваться определенный уровень экспертизы для правильной настройки и использования.

6. JWT (JSON Web Token) – компактный и самодостаточный способ представления информации об аутентификации и авторизации в виде JSON-объекта, который может быть передан между двумя сторонами [11].

Характеристики:

- компактный и самодостаточный способ представления информации об аутентификации и авторизации в виде JSON-объекта;
- обычно используется для передачи утверждений между клиентами и серверами в веб-приложениях и API.

Сложность обслуживания/установки: использование JWT может быть относительно простым для разработчиков, но требует правильной обработки и безопасности использования при передаче конфиденциальной информации.

Бесплатность: JWT доступен бесплатно и является открытым стандартом.

Благодаря своей открытому коду, LDAP легко интегрируется с различными устройствами и программами, что обеспечивает единый и удобный доступ к данным с любого устройства или приложения.

Кроме того, LDAP позволяет управлять доступом к ресурсам, устанавливать права и ограничения для пользователей и групп, что способствует безопасности и эффективности работы в сетевой среде. Благодаря масштабируемости протокола, его легко адаптировать под различные потребности организации, обеспечивая гибкость и удобство в настройке и использовании.

Использование LDAP упрощает администрирование сети и повышает производительность за счет централизованного хранения данных и управления учетными записями. Кроме того, LDAP позволяет создавать отказоустойчивые

системы, обеспечивая доступ к данным даже в случае сбоев в одном из серверов. В целом, протокол LDAP является надежным и эффективным инструментом для организации управления пользователями и ресурсами в сетевой среде.

### 1.2.2 Решения для совместной разработки

На данный момент существует несколько решений для управления репозиториями кода и совместной разработки в среде Linux. Некоторые из них включают:

**GitHub.** Один из самых популярных и широко используемых сервисов для хостинга и управления репозиториями Git. Хотя GitHub предоставляет отличные инструменты для совместной работы над проектами, он обычно используется как облачный сервис и не всегда подходит для организаций, предпочитающих хранить данные внутри собственной инфраструктуры [12].

Преимущества: широкое распространение, крупное сообщество, множество интеграций и поддержка для открытых проектов.

Недостатки: требует оплаты для использования в коммерческих проектах.

**Bitbucket.** Еще один популярный сервис хостинга Git-репозитория, который предоставляет возможности совместной работы над кодом и интеграцию с другими инструментами разработки. Bitbucket также может быть использован как облачный сервис или установлен на собственном сервере [13].

Преимущества: интеграция с другими сервисами Atlassian, бесплатное использование для небольших команд.

Недостатки: менее популярен и имеет более ограниченное сообщество, чем GitHub.

**GitLab.** Полноценная платформа для управления Git-репозиториями с открытым исходным кодом, управление проектами, GitLab предоставляет широкий спектр функциональных возможностей, включая управление кодом, непрерывную интеграцию и доставку (CI/CD), управление задачами и релизами,

систему отслеживания ошибок, wiki и многое другое [14]. GitLab также предлагает возможность самостоятельной установки на собственных серверах.

Преимущества: широкий набор инструментов, возможность самостоятельной установки, бесплатное использование для открытых проектов и коммерческих проектов с ограниченным количеством пользователей, встроенная система CI/CD.

Недостатки: менее популярен, чем GitHub, для больших коммерческих проектов потребуется платная подписка.

Проанализировав информацию выше, можно сделать вывод, что именно GitLab обладает наиболее широким функционалом, обеспечивая полный цикл разработки и доставки программного обеспечения. Он предлагает широкий набор инструментов для управления проектами, CI/CD, управления задачами и других задач разработки. Более того, GitLab обеспечивает возможность самостоятельной установки на собственных серверах, что дает большую гибкость и контроль над процессами разработки.

Существенным минусом является то, что GitHub не обеспечивает возможность самостоятельной установки на собственных серверах, как GitLab.

GitHub в основном предоставляет облачный хостинг для Git-репозиторий и инструменты совместной работы над проектами, а также управления версиями кода. Это платформа, предлагаемая GitHub, Inc., и не предоставляет пользователям возможность развертывания собственных экземпляров GitHub на собственных серверах.

GitLab, напротив, предоставляет две версии: GitLab Community Edition (CE) с открытым исходным кодом, которая может быть установлена и использована бесплатно, и GitLab Enterprise Edition (EE), которая предлагает расширенные функции и поддержку, но платная. Обе версии позволяют пользователям развернуть GitLab на своих серверах, обеспечивая большую гибкость и контроль над окружением разработки.

### 1.2.3 Почтовый сервис

Почтовая система, на части серверного ПО разбивается на несколько компонентов, которые выполняют различные функции. Каждый компонент отвечает за определенные аспекты работы почтовой системы, такие как прием и отправка сообщений, хранение и обработка почтовых данных. Рассмотрим обязательные компоненты почтовой системы:

MTA (Mail Transfer Agent) – Агент передачи почты (разбирается с исходящей почтой)

MTA – это компонента, отвечающая за отправку и маршрутизацию почтовых сообщений между различными почтовыми серверами. Он обрабатывает задачи по отправке электронной почты от отправителя к получателю, используя протоколы, такие как SMTP (Simple Mail Transfer Protocol) [15]. MTA является ключевым элементом в почтовой инфраструктуре, обеспечивая доставку сообщений между почтовыми серверами.

MTA принимает электронные письма от почтовых клиентов или других MTA и затем маршрутизирует их к соответствующим почтовым серверам, где получатели имеют свои почтовые ящики. Он использует протоколы, такие как SMTP (Simple Mail Transfer Protocol), для передачи сообщений между серверами.

При отправке письма, почтовый клиент или MTA отправителя обращается к DNS-серверу, чтобы найти запись MX (Mail Exchanger) для домена получателя. Затем MTA отправителя устанавливает соединение с MTA получателя и передает ему письмо. MTA получателя принимает письмо и доставляет его в почтовый ящик получателя.

MX (Mail Exchanger) (принимает почту с Интернета)

Компонента MX (Mail Exchanger) является частью системы DNS (Domain Name System) [16], которая отвечает за маршрутизацию электронной почты. Она определяет, на какие серверы должны быть отправлены электронные сообщения для определенного домена. Когда пользователь отправляет электронное письмо,



почтовый клиент или почтовый сервер обращается к DNS-серверу, чтобы найти запись MX для адреса электронной почты получателя. Запись MX указывает на серверы, которые отвечают за прием и доставку почты для этого домена. Каждая запись MX имеет приоритет (число), которое указывает на порядок предпочтения серверов. Компонента MX играет важную роль в обеспечении надежной доставки электронной почты, позволяя маршрутизировать письма на соответствующие почтовые серверы.

MDA (Mail Delivery Agent) – агент доставки почты

MDA отвечает за доставку почтовых сообщений на конкретные почтовые ящики пользователя. MDA обрабатывает протоколы, такие как POP3 (Post Office Protocol 3) и IMAP (Internet Message Access Protocol), чтобы пользователи могли получать и управлять своей почтой [15].

Когда электронное письмо достигает почтового сервера, MDA берет на себя задачу доставки этого письма в соответствующий почтовый ящик получателя. MDA сохраняет письма в ящике и обеспечивает доступ к ним через протоколы POP3 или IMAP. Пользователи могут использовать почтовые клиенты или веб-интерфейсы, чтобы получать доступ к своей почте, читать письма, отправлять новые сообщения и управлять папками и метками.

MUA (Mail User Agent) – почтовый клиент.

MUA представляет собой программу, используемую конечным пользователем для доступа к своим электронным письмам: для чтения, отправки, получения и управления своей электронной почтой. Это может быть веб-интерфейс, как в случае RainLoop, или приложение, например, Outlook. MUA взаимодействует с пользователем и передает сообщения MTA для отправки [17].

Выше были рассмотренные компоненты почтового сервера, которые используются при работе с почтой. Рассмотрим почтовые серверы, состоящих из компонентов, указанных выше.

**Mail-in-a-Box** – это программное обеспечение, которое позволяет создать собственный почтовый сервер. Оно включает в себя почтовый сервер, веб-

интерфейс для электронной почты, а также средства для синхронизации контактов и календарей. Mail-in-a-Vox предоставляет возможность полного контроля над вашей электронной почтой и защиты вашей конфиденциальности, может быть использован для создания собственного почтового сервера для личного использования или для малых организаций [18].

Плюсы: прост в установке и обслуживании, имеет встроенные функции безопасности и автоматического резервного копирования, бесплатный и с открытым исходным кодом.

Минусы: может быть не такой мощной и расширяемой, имеет ограниченный выбор интегрированных функций.

**Zimbra** – платформа для совместной работы, которая включает в себя почтовый сервер, календарь, контакты и другие инструменты для совместной работы. Она предоставляет возможность обмена сообщениями, планирования встреч и совместной работы над документами [19]. Zimbra может быть использована как веб-интерфейс для электронной почты или как сервер для хранения и управления электронной почтой и другими данными.

Плюсы: это платформа для совместной работы, включающая почту, календарь, контакты и другие функции.

Минусы: требует больше ресурсов и усилий для установки и обслуживания, сложен в настройке и поддержке для неопытных пользователей.

**Microsoft Exchange** – программное обеспечение для обмена сообщениями и совместной работы, разработанное компанией Microsoft [20]. Предоставляет возможность отправки и получения электронных писем, планирования встреч, управления контактами и других функций, связанных с электронной почтой и совместной работой. Microsoft Exchange Server широко используется в предприятиях, которые ищут решение через инфраструктуру Microsoft.

Плюсы: мощная и широко используемая корпоративная почтовая система, включает интеграцию с другими продуктами Microsoft.

Минусы: платное решение с высокими затратами на лицензии и обслуживание, ограничивает выбор по операционной системе, т.к. устанавливается на серверах с операционной системой Windows Server.

**iRedMail** – это полноценное решение для почтового сервера с открытым исходным кодом. Оно включает в себя почтовый сервер, веб-интерфейс для электронной почты, антиспам-фильтры и другие компоненты, необходимые для работы почтового сервера [21]. iRedMail может быть использован для создания почтового сервера для бизнеса или личного использования.

Плюсы: несложен в установке и использовании, хорошо интегрируется с Linux Ubuntu и поддерживает LDAP для аутентификации, бесплатный и с открытым исходным кодом, наличие веб-интерфейса, нет ограничения в количестве аккаунтов.

Минусы: может оказаться менее расширяемым или функциональным по сравнению с некоторыми коммерческими решениями.

При условии, что сервер работает на Linux Ubuntu и требуется интеграция с LDAP для аутентификации, можно сделать вывод, что iRedMail является оптимальным выбором благодаря своей простоте установки, поддержке LDAP и открытому исходному коду. Это позволяет гибко управлять почтовым сервером.

### **1.3 Выводы по главе**

Выполнено исследование существующих решений ПО. Для определения подходящих вариантов был выполнен обзор и сравнение ПО в следующих аспектах:

- авторизация пользователей в СКБ, а именно был выбран протокол LDAP, который удовлетворяет всем требованиям;
- система коммуникаций, а именно были проанализированы решения и выбрано наиболее подходящее - iRedMail;
- были исследованы решения систем контроля версий, из которых было выбрано оптимальное ПО в виде решения GitLab.

Разработанная структура СКБ для взаимодействия пользователей, а также выбор основных инструментов для практической реализации позволили перейти к разработке и изготовлению серверной части.

## **2 Разработка инфраструктуры студенческого конструкторского бюро**

### **2.1 Техническое задание**

Студенческое конструкторское бюро имеет несколько рабочих мест, на которых установлен стандартный набор программ. Этот набор включает в себя:

- офисный пакет для оформления и чтения документации;
- оболочки для работы с языками программирования C/C++;
- MatLab для выполнения математических расчетов;
- утилиты для работы с языком программирования Python;
- утилиты для работы с языком программирования Assembler.

Также на схеме имеются сервера с доступом в интернет, на которых работают следующие сервисы:

- протокол LDAP, который отвечает за аутентификацию пользователей.
- GitLab, который выполняет роль сервиса для хранения кода и системы контроля версий при совместной разработке программного обеспечения.
- iRedMail, который выполняет роль почтового сервера.

Основные задачи данной инфраструктуры включают:

1. Аутентификация пользователей через протокол LDAP для обеспечения безопасности и контроля доступа к ресурсам.
2. Хранение кода и система контроля версий через GitLab для совместной разработки программного обеспечения.
3. Предоставление почтового сервера iRedMail для коммуникации и обмена информацией между участниками конструкторского бюро.

### **2.2 Общая структура**

На рисунке 2 изображена разработанная схема инфраструктура студенческого конструкторского бюро.

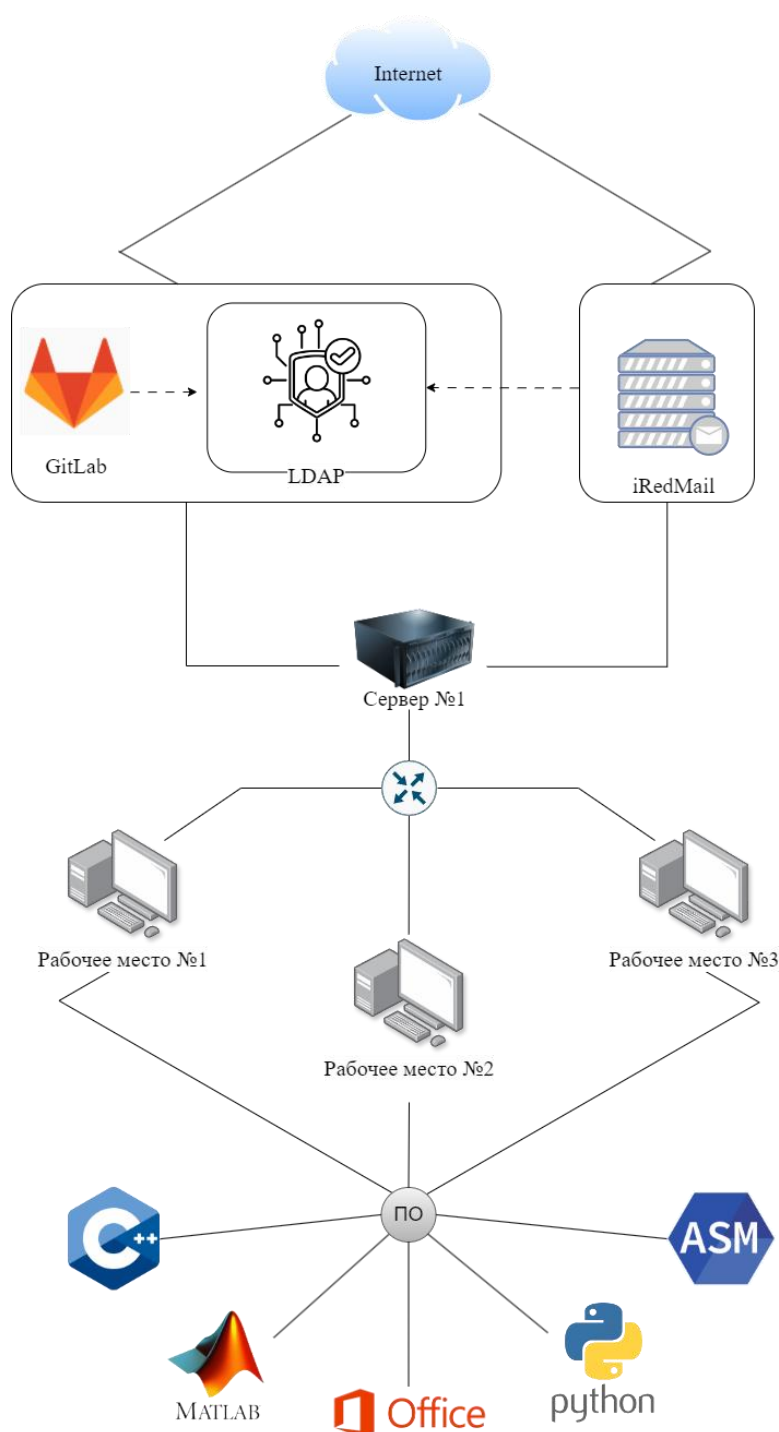


Рисунок 2 – Схема инфраструктуры студенческого конструкторского бюро

Имеется несколько рабочих мест, на которых установлен стандартный набор программ: офисный пакет для оформления и чтения документации, оболочки для работы с C/C++ языками, MatLab, утилиты для работы с Python, а также утилиты для работы с Assembler. Также на схеме имеется сервер с доступом в интернет, на котором развернуто две виртуальные машины:

- на виртуальной машине №1 работает протокол LDAP, отвечающий за аутентификацию и GitLab, который выполняет роль сервиса для хранения кода и системы контроля версий при совместной разработке программного обеспечения;

- на виртуальной машине №2 работает iRedMail, выполняющий роль почтового сервера.

Для реализации данной схемы был разработан виртуальный стенд, который представляет собой две виртуальные машины с установленной ОС Linux Ubuntu 22.04.4 LTS. Гипервизором является ноутбук HP OMEN Laptop 15-en0xxx (Рисунок 3) с установленной программой виртуализации VirtualBox 7.0 (Рисунок 4) [22]. Виртуальные машины имеют по 2 Гб оперативной памяти (Рисунок 5) по одному процессору (Рисунок 6).

#### Характеристики устройства

##### OMEN Laptop 15-en0xxx

Имя устройства	LAPTOP-DCCO4G6E
Процессор	AMD Ryzen 5 4600H with Radeon Graphics 3.00 GHz
Оперативная память	16,0 Гб (доступно: 15,4 Гб)
Код устройства	527A82D0- B2CE-4B7F-804D-742C2995F831
Код продукта	00325-81987-29802-AAOEM
Тип системы	64-разрядная операционная система, процессор x64
Перо и сенсорный ввод	Для этого монитора недоступен ввод с помощью пера и сенсорный ввод

Копировать

Переименовать этот ПК

#### Характеристики Windows

Выпуск	Windows 10 Домашняя
Версия	22H2
Дата установки	03.02.2021
Сборка ОС	19045.4291
Взаимодействие	Windows Feature Experience Pack 1000.19056.1000.0

Копировать

Рисунок 3 – Характеристики ноутбука HP OMEN Laptop 15-en0xxx

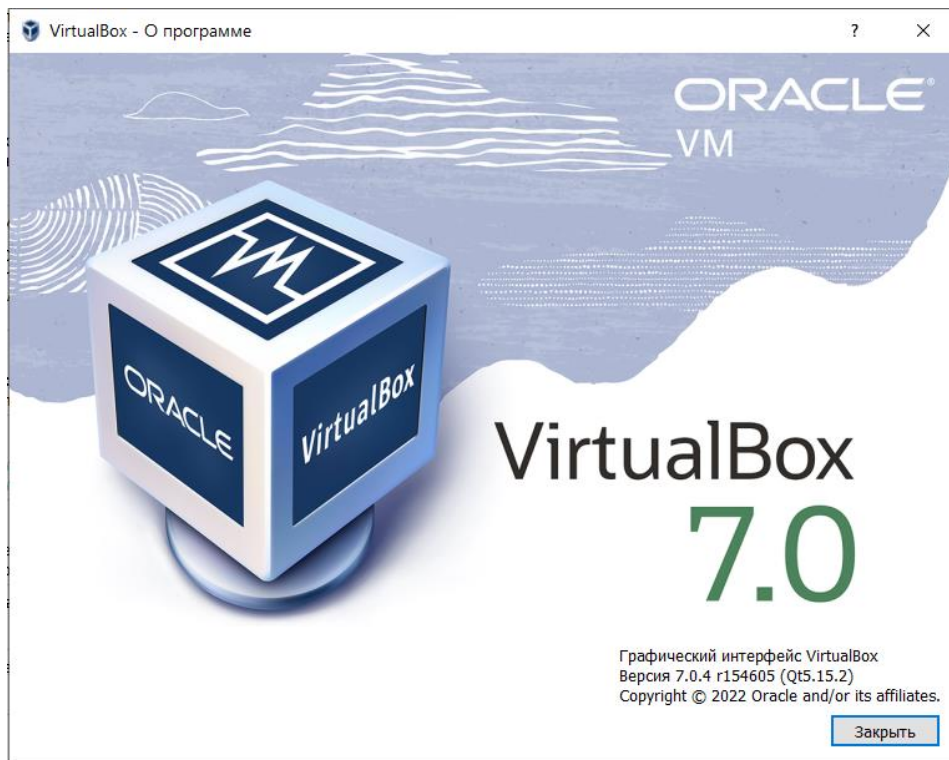


Рисунок 4 – Программа виртуализации

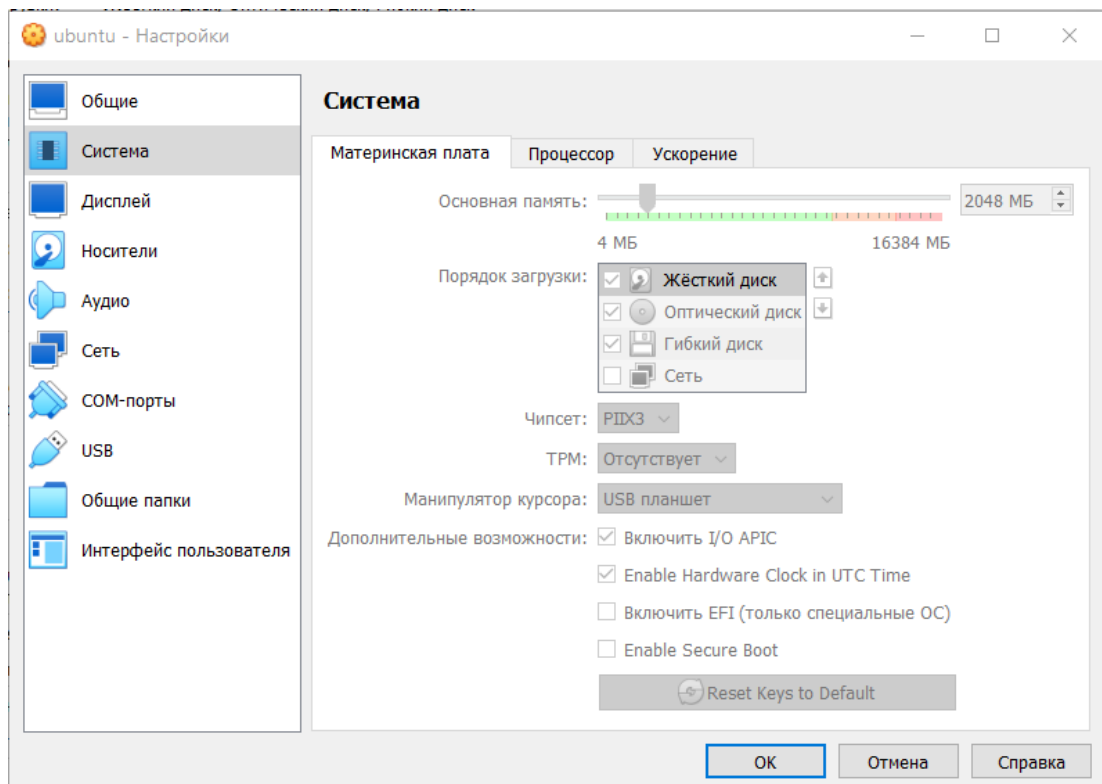


Рисунок 5 – Оперативная память виртуальной машины



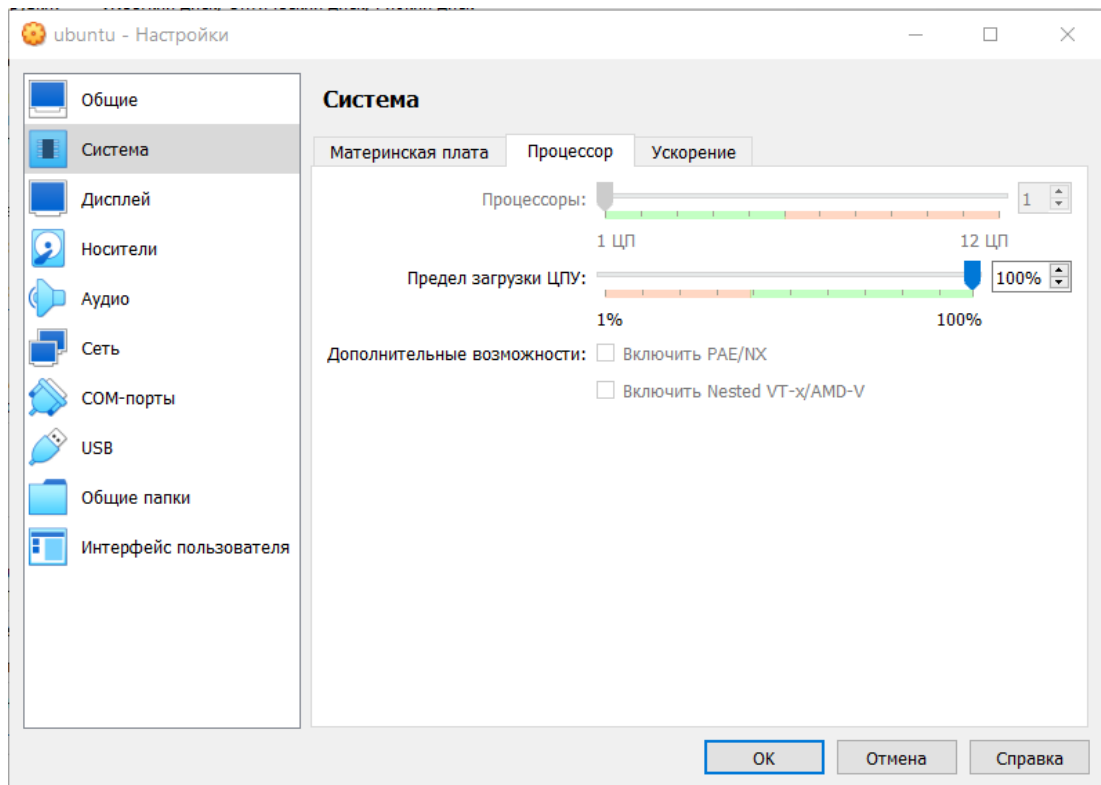


Рисунок 6 – Количество процессоров виртуальных машин

Виртуальная машина в VirtualBox может иметь различные настройки сети (Рисунок 7). В данном случае указывается тип подключения «Сетевой мост» (Рисунок 8).

Таким образом, данная настройка адаптера позволит находиться виртуальной машине и ноутбуку в одной сети.

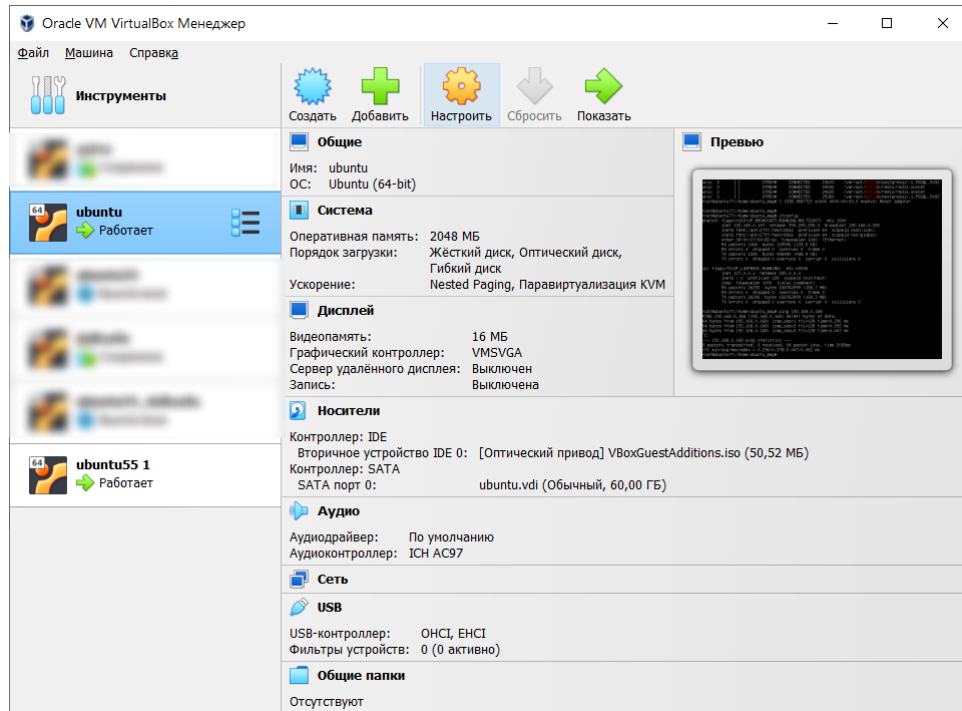


Рисунок 7 – Раздел «Настроить»

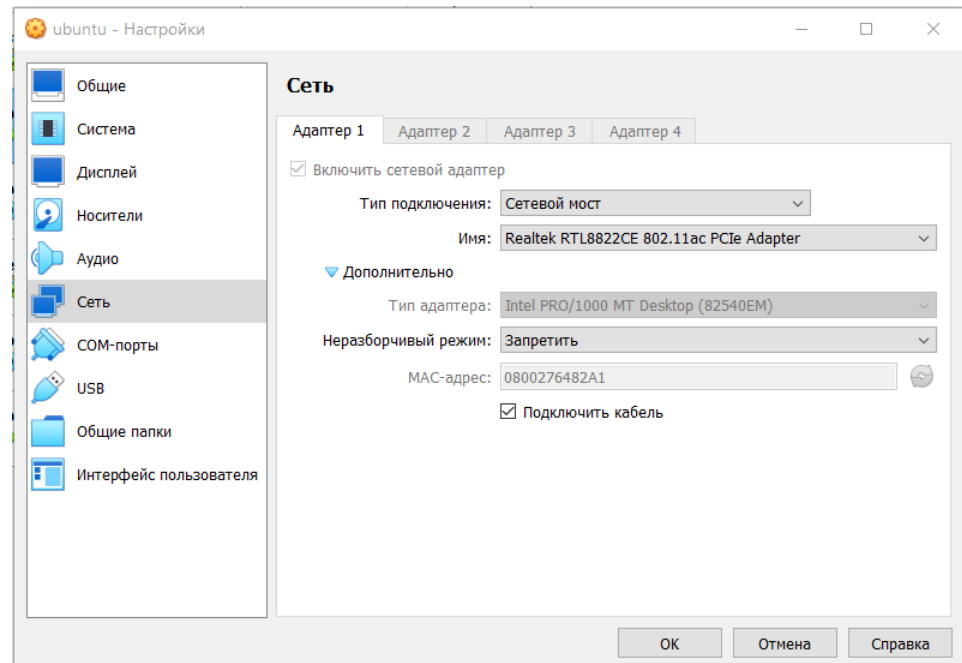


Рисунок 8 – Настройка сети виртуальных машин



```
ubuntu55 1 [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Смотрите https://ubuntu.com/esm или выполните: sudo pro status

Last login: Thu Jun 20 18:32:44 UTC 2024 on tty1
mag_ubuntu@myguest7:~$ [ 31.869151] cloud-init[2425]: Cloud-init v. 23.2.1-0ubuntu0~22.04.1 running
g 'modules:final' at Thu, 20 Jun 2024 18:34:10 +0000. Up 31.70 seconds.
[ 32.047930] cloud-init[2425]: Cloud-init v. 23.2.1-0ubuntu0~22.04.1 finished at Thu, 20 Jun 2024
18:34:10 +0000. Datasource DataSourceNone. Up 32.04 seconds
[ 32.050941] cloud-init[2425]: 2024-06-20 18:34:10,936 - cc_final_message.py[WARNING]: Used fallback
ck datasource

i: command not found
mag_ubuntu@myguest7:~$
mag_ubuntu@myguest7:~$
mag_ubuntu@myguest7:~$
mag_ubuntu@myguest7:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.0.134  netmask 255.255.255.0  broadcast 192.168.0.255
    inet6 fe80::a00:27ff:fe6b:f832  prefixlen 64  scopeid 0x20<link>
    inet6 fd01::a00:27ff:fe6b:f832  prefixlen 64  scopeid 0x0<global>
    ether 08:00:27:6b:f8:32  txqueuelen 1000  (Ethernet)
    RX packets 80  bytes 12064 (12.0 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 75  bytes 7742 (7.7 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Local Loopback)
    RX packets 2060  bytes 230154 (230.1 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 2060  bytes 230154 (230.1 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

mag_ubuntu@myguest7:~$ _
```

Рисунок 11 – Сеть на установленной виртуальной машине №2

### 2.3 Разработка подсистемы централизованной аутентификации

В качестве протокола, отвечающего за централизованную аутентификацию, в ходе исследований был выбран LDAP. Перейдем к его установке.

1. Первым шагом необходимо выполнить обновление пакетов. После выполнения команды «*sudo apt-get update*» будет получен обновленный список пакетов, который можно использовать для установки или обновления ПО в системе.

2. После того, как пакеты обновлены, можно приступать к установке LDAP сервера и набора утилит (ldap-utils), которые позволяют управлять и работать с каталогом LDAP на сервере Ubuntu. Выполняем установку, используя команду «*sudo apt-get install slapd ldap-utils*». После выполнения команды открывается консоль для настройки администратора LDAP (Рисунок 12).

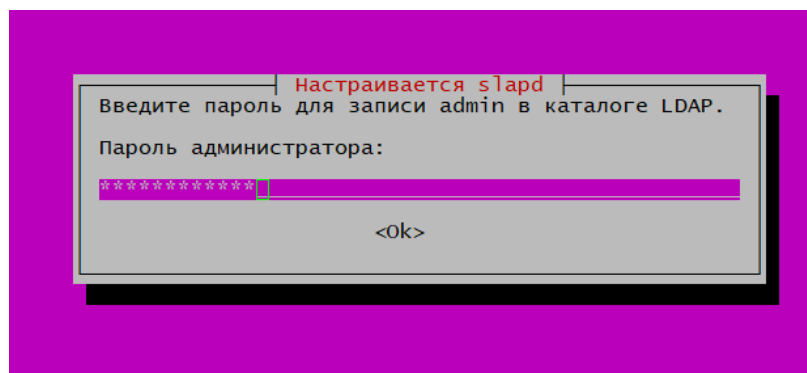


Рисунок 12 – Настройка администратора LDAP

После ввода пароля администратора LDAP, на следующем шаге нужно будет его повторить.

3. Теперь в терминале выполним команду *«sudo slapcat»*. Slapcat - утилита генерации LDIF из базы данных SLAPD. Выполняется данная команда для генерации файла формата LDIF (LDAP Directory Interchange Format) на основе содержимого базы данных slapd(8) LDAP.

4. После генерации LDIF файла необходимо выполнить *«dpkg-reconfigure slapd»*. Данная команда используется для повторной настройки пакета slapd. Она предоставляет возможность изменить конфигурационные параметры slapd без использования интерактивного пользовательского интерфейса. При выполнении команды *«dpkg-reconfigure slapd»* открывается интерактивное окно, в котором можно ввести настройки или сделать выбор. На скриншотах ниже показан процесс настройки (Рисунки 13-17).

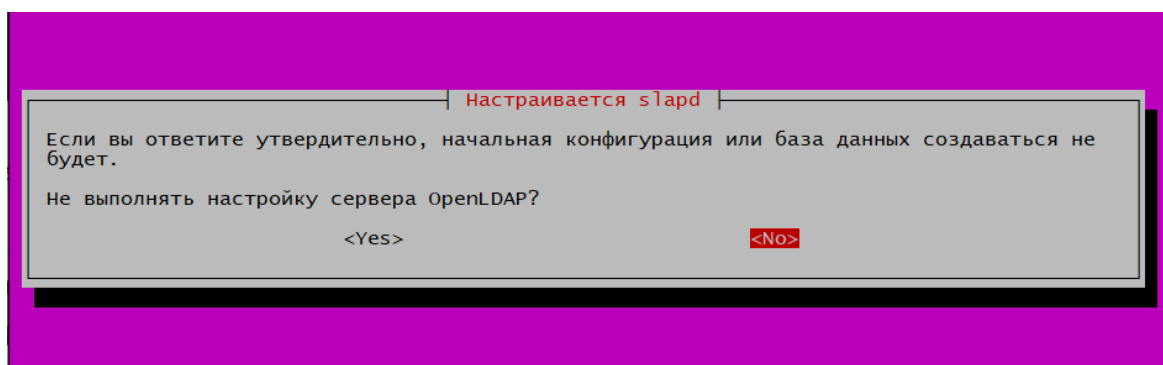


Рисунок 13 – Настройка LDAP

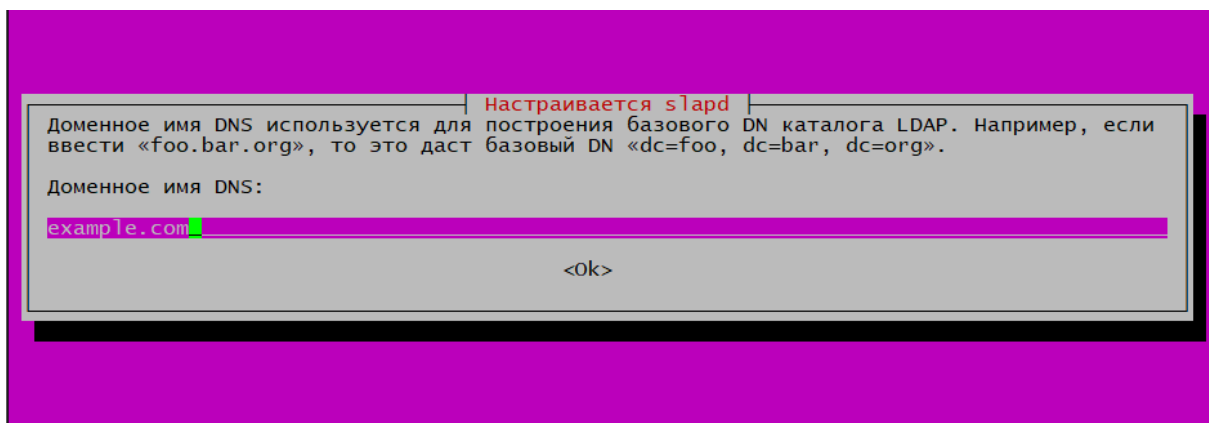


Рисунок 14 – Настройка LDAP, указываем доменное имя

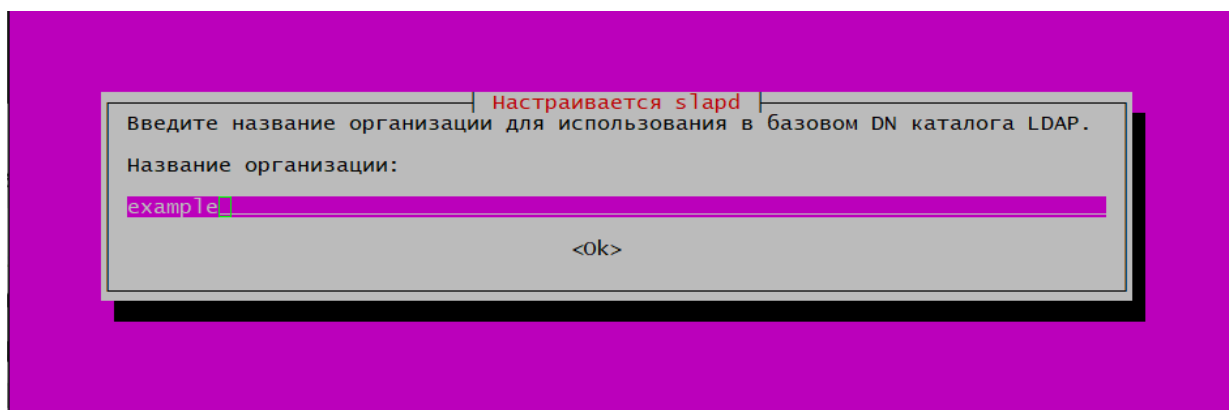


Рисунок 15 – Настройка LDAP, задаем название организации

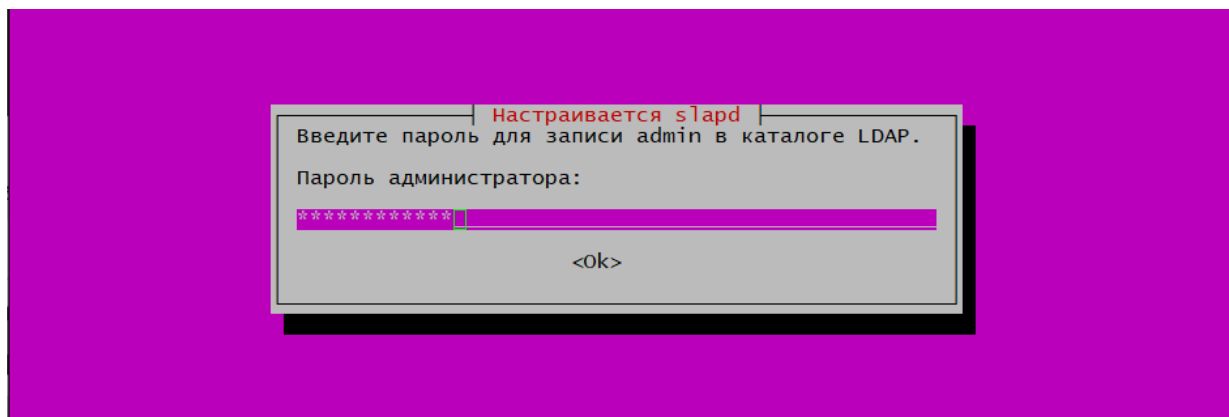


Рисунок 16 – Указываем пароль администратора LDAP

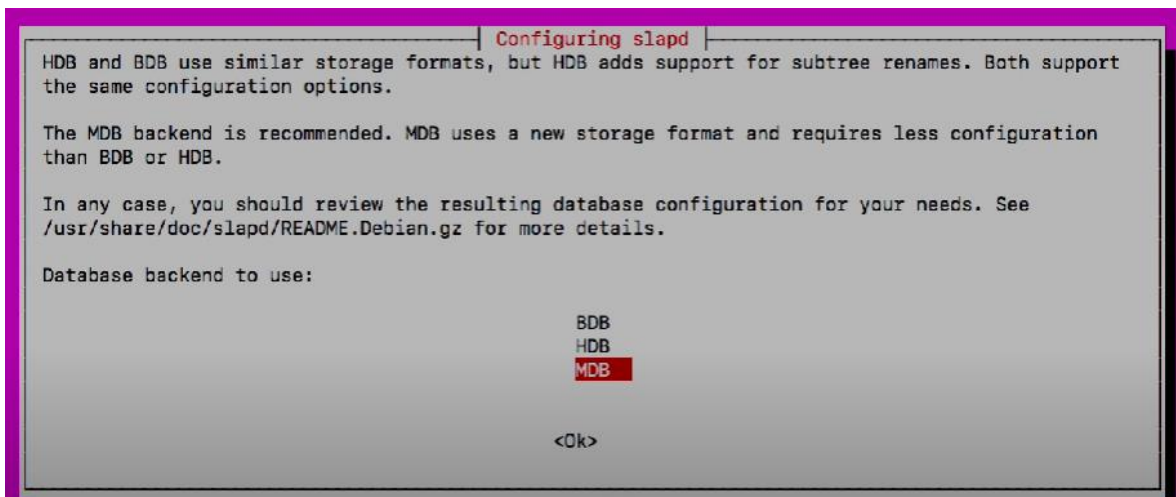


Рисунок 17 – Выбираем вариант базы данных

5. После того, как настройка выполнена, проверяем статус службы с помощью команды «*service slapd status*» в результате которой можем видеть, что служба запущена и действительно работает (Рисунок 18).

```
root@ubuntu22: /home/ubuntu_mag
Scanning linux images...
Restarting services...
systemctl restart packagekit.service
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ubuntu22: /home/ubuntu_mag# sudo apt-get install slapd ldap-utils
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Уже установлен пакет ldap-utils самой новой версии (2.5.17+dfsg-0ubuntu0.22.04.1).
Уже установлен пакет slapd самой новой версии (2.5.17+dfsg-0ubuntu0.22.04.1).
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и 26 п
ов не обновлено.
root@ubuntu22: /home/ubuntu_mag# service slapd status
● slapd.service - LSB: OpenLDAP standalone server (Lightweight Directory Access Protocol)
   Loaded: loaded (/etc/init.d/slapd; generated)
   Drop-In: /usr/lib/systemd/system/slapd.service.d
            └─slapd-remain-after-exit.conf
   Active: active (running) since Tue 2024-04-30 14:35:20 UTC; 18h ago
     Docs: man:systemd-sysv-generator(8)
   Process: 22483 ExecStart=/etc/init.d/slapd start (code=exited, status=0/SUCCESS)
    Tasks: 3 (limit: 2221)
   Memory: 3.0M
     CPU: 24ms
   CGroup: /system.slice/slapd.service
           └─22490 /usr/sbin/slapd -h "ldap:/// ldapi:///" -g openldap -u openldap -F /
apn 30 14:35:20 ubuntu22 systemd[1]: Starting LSB: OpenLDAP standalone server (Lightweigh
apn 30 14:35:20 ubuntu22 slapd[22483]: * Starting OpenLDAP slapd
apn 30 14:35:20 ubuntu22 slapd[22489]: @(#) $OpenLDAP: slapd 2.5.17+dfsg-0ubuntu0.22.04.1
Ubuntu Developers <ubuntu-devel-discuss
apn 30 14:35:20 ubuntu22 slapd[22490]: slapd starting
apn 30 14:35:20 ubuntu22 slapd[22483]: ...done.
apn 30 14:35:20 ubuntu22 systemd[1]: Started LSB: OpenLDAP standalone server (Lightweight
lines 1-20/20 (END)
root@ubuntu22: /home/ubuntu_mag#
```

Рисунок 18 – Статус службы LDAP

6. Также необходимо настроить брандмауэр, так как при подключении со стороны LDAP будут блокироваться входящие соединения. Сделаем это путем использования команды `«sudo ufw allow ldap»`, которая используется для разрешения входящих соединений по протоколу LDAP. После выполнения всех действий необходимо перезагрузить сервер.

7. Попробуем добавить новый каталог в LDAP. Чтобы создать новую ветку создаем файл с нужной структурой, используя команду `«touch /etc/ldap/gitlab.idif»`. После этого необходимо с помощью команды `«ldapadd -x -D cn=admin,dc=example,dc=com -W -c -f /etc/ldap/gitlab.idif»` добавить эту запись в базу данных. Разберем данную команду:

`ldapadd`: это утилита командной строки, которая позволяет добавлять данные в LDAP-сервер.

`-x`: ключ, указывающий на использование простой аутентификации (simple authentication).

`-D cn=admin,dc=example,dc=com`: эта часть указывает DN (Distinguished Name) пользователя, который будет использоваться для аутентификации. В данном случае, `cn=admin,dc=example,dc=com` означает, что пользователь с Common Name (CN) "admin" и находящийся в домене "example.com" будет использоваться для аутентификации.

`-W`: данный ключ отвечает за запрос пароля пользователя во время выполнения команды.

`-c`: это опция, указывающая, что при возникновении ошибок выполнение команды будет продолжено

`-f /etc/ldap/gitlab.idif`: это опция, указывающая путь к файлу, содержащему данные, которые нужно добавить в LDAP-сервер. В данном случае, файл `/etc/ldap/gitlab.idif` содержит данные, которые будут добавлены.

При выполнении система потребует ввести пароль администратора LDAP. После верного ввода система выдаст сообщение о том, что новая запись добавлена (Рисунок 19).



```
root@ubuntu77:/etc/ldap# touch /etc/ldap/gitlab.idif
root@ubuntu77:/etc/ldap#
root@ubuntu77:/home/ubuntu_mag# ldapadd -x -D cn=admin,dc=example,dc=com -W -c -f /etc/ldap/gitlab.idif
Enter LDAP Password:
adding new entry "ou=GITLAB,dc=example,dc=com"
root@ubuntu77:/home/ubuntu_mag#
```

Рисунок 19 – Добавление каталога GITLAB в БД LDAP

После создания каталога, добавим пользователя. Для этого создадим необходимый файл с содержимым (Рисунок 20). Разберем каждую строку из файла:

`dn: cn=Pogosyan Agnessa Arturovna,ou=GITLAB,dc=example,dc=com:`  
данная строка определяет уникальное имя (Distinguished Name) записи и указывает на ее положение в древовидной структуре каталога. В данном случае, запись находится в организационной единице (OU) с именем "GITLAB" в домене "example.com". Имя записи (cn) равно "Pogosyan Agnessa Arturovna".

`objectClass: inetOrgPerson:` Эта строка указывает на класс объекта, к которому относится данная запись. В данном случае, запись относится к классу "inetOrgPerson", который представляет собой стандартный класс для представления информации о человеке в LDAP.

`cn: Pogosyan Agnessa Arturovna:` Эта строка указывает на значение атрибута "cn" (Common Name), который представляет собой полное имя человека.

`sn: Pogosyan:` Эта строка указывает на значение атрибута "sn" (Surname), который представляет собой фамилию человека.

`givenName: Agnessa:` Эта строка указывает на значение атрибута "givenName", который представляет собой имя человека.

`uid: aaarpogosyan:` Эта строка указывает на значение атрибута "uid" (User ID), который представляет собой уникальный идентификатор пользователя.

`mail: pogosyan.agnessa@mail.ru:` Эта строка указывает на значение атрибута "mail", который представляет собой адрес электронной почты пользователя.

`userPassword: password777:` Эта строка указывает на значение атрибута "userPassword", который представляет собой пароль пользователя.

```
mc [root@ubuntu77]:/etc/ldap/slapd.d
/etc/ldap/slapd.d/aaapogosyan.ldif
dn: cn=Pogosyan Agnessa Arturovna,ou=GITLAB,dc=example,dc=com
objectClass: inetOrgPerson
cn: Pogosyan Agnessa Arturovna
sn: Pogosyan
givenName: Agnessa
uid: aaapogosyan
mail: pogosyan.agnessa@mail.ru
userPassword: password777
```

Рисунок 20 – Содержимое файла с пользователем

Выполняем «`ldapadd -x -D cn=admin,dc=example,dc=com -W -c -f /etc/ldap/slapd.d/aaapogosyan.ldif`» и видим сообщение об успешном добавлении записи в базу данных LDAP (Рисунок 21)

```
root@ubuntu77:/etc/ldap/slapd.d# ldapadd -x -D cn=admin,dc=example,dc=com -W -c -f /etc/ldap/slapd.d/aaapogosyan.ldif
Enter LDAP Password:
adding new entry "cn=Pogosyan Agnessa Arturovna,ou=GITLAB,dc=example,dc=com"
```

Рисунок 21 – Добавление пользователя в БД LDAP

Таким образом, мы установили и настроили службу LDAP на Linux Ubuntu 22.04.4 LTS.

## 2.4 GitLab

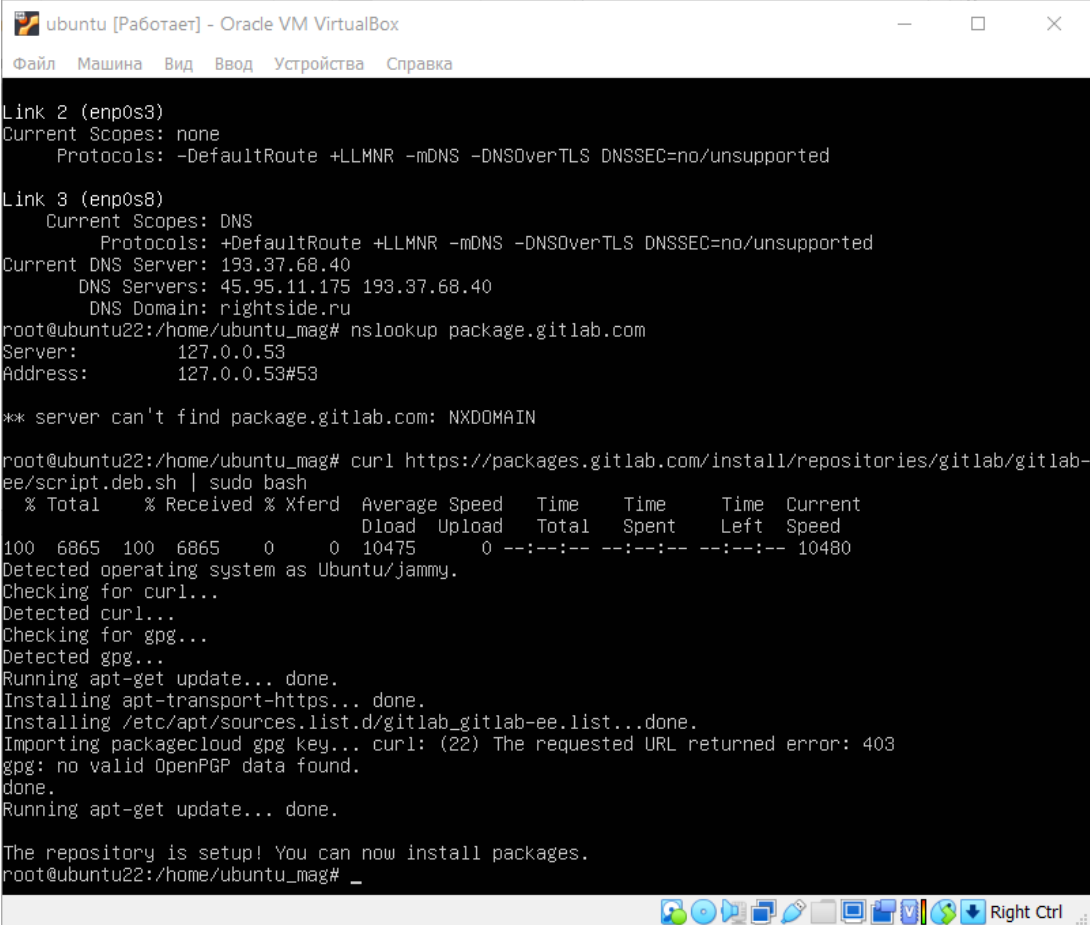
В ходе качества решения для совместной разработки, в ходе анализа был выбран GitLab. Перейдем к его установке на Ubuntu 22.04.4 LTS.

1. Первым шагом нам необходимо выполнить обновление пакетов. После выполнения команды «`sudo apt-get update`» будет получен обновленный список пакетов, который можно использовать для установки или обновления ПО в системе.

2. После того, как пакеты обновлены, можем приступать к установке GitLab. Установим необходимые зависимости командой «*sudo apt-get install -y curl openssh-server ca-certificates tzdata perl*».

GitLab имеет две версии — Community Edition (CE) и Enterprise Edition (EE). У первой полностью открытый исходный код, а вторая построена на базе первой, но имеет дополнительные функции, код которых, увы, не открыт для всех желающих и является платной версией. В рамках данного задания будем устанавливать GitLab Community Edition (CE).

Для установки необходимо добавить репозиторий пакетов GitLab, затем установить пакет. Загружаем установочный скрипт, выполнив команду «*curl https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.deb.sh | sudo bash*» (Рисунок 22).



```
ubuntu [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка

Link 2 (enp0s3)
Current Scopes: none
  Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported

Link 3 (enp0s8)
  Current Scopes: DNS
    Protocols: +DefaultRoute +LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
Current DNS Server: 193.37.68.40
  DNS Servers: 45.95.11.175 193.37.68.40
  DNS Domain: rightside.ru
root@ubuntu22:/home/ubuntu_mag# nslookup package.gitlab.com
Server:         127.0.0.53
Address:        127.0.0.53#53

** server can't find package.gitlab.com: NXDOMAIN

root@ubuntu22:/home/ubuntu_mag# curl https://packages.gitlab.com/install/repositories/gitlab/gitlab-
ee/script.deb.sh | sudo bash
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 6865    100 6865    0     0  10475      0  --:--:--  --:--:--  --:--:-- 10480
Detected operating system as Ubuntu/jammy.
Checking for curl...
Detected curl...
Checking for gpg...
Detected gpg...
Running apt-get update... done.
Installing apt-transport-https... done.
Installing /etc/apt/sources.list.d/gitlab_gitlab-ee.list...done.
Importing packagecloud gpg key... curl: (22) The requested URL returned error: 403
gpg: no valid OpenPGP data found.
done.
Running apt-get update... done.

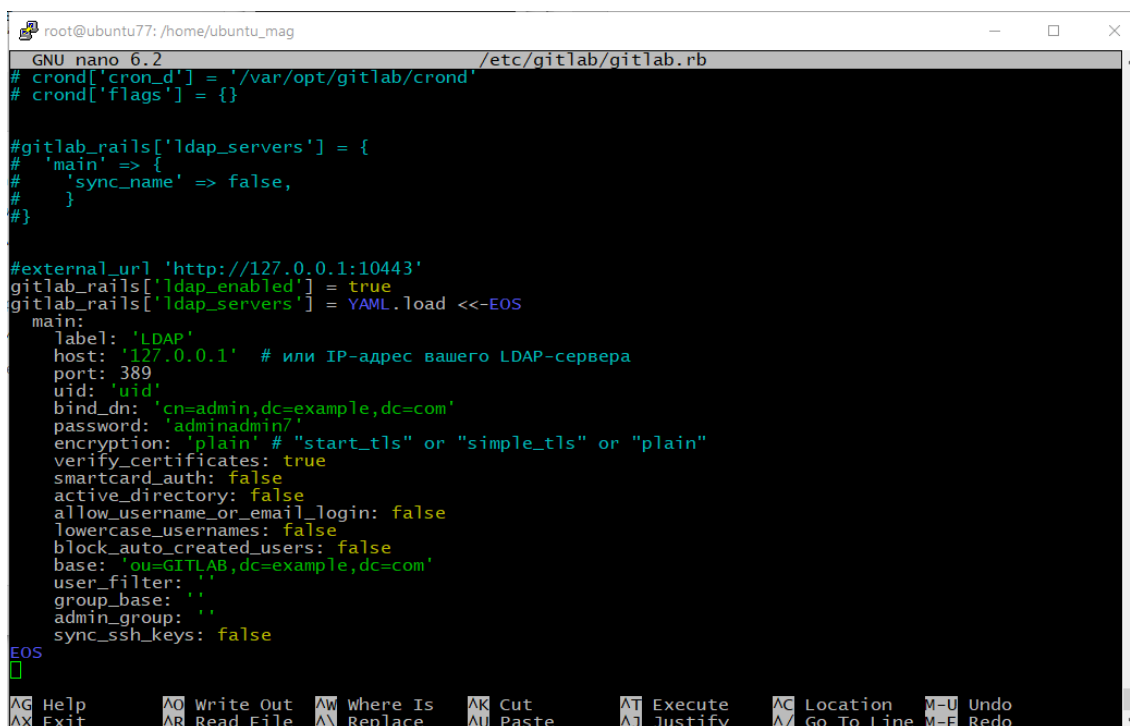
The repository is setup! You can now install packages.
root@ubuntu22:/home/ubuntu_mag# _
```

Рисунок 22 – Добавление установочного скрипта



Необходимо скорректировать в файле `/etc/gitlab/gitlab.rb` параметр «`external_url 'http://192.168.0.147:8888'`», указав корректный адрес, по которому будут подключаться и порт. Также в файле `/var/opt/gitlab/gitlab-rails/etc/gitlab.yml` необходимо найти параметр `port`, в котором указывается порт подключения. В итоге строка выглядит следующим образом: «`port: 8888`».

После завершения установки и настройки GitLab, переходим к интеграции LDAP в GitLab. Для настройки аутентификации в GitLab через LDAP, необходимо внести изменения в файл GitLab `/etc/gitlab/gitlab.rb`. Измененный конфигурационный файл GitLab можно увидеть на рисунке 25.



```
root@ubuntu77: /home/ubuntu_mag
GNU nano 6.2 /etc/gitlab/gitlab.rb
# crond['cron_d'] = '/var/opt/gitlab/cron'
# crond['flags'] = {}

#gitlab_rails['ldap_servers'] = {
#  'main' => {
#    'sync_name' => false,
#  }
#}

#external_url 'http://127.0.0.1:10443'
gitlab_rails['ldap_enabled'] = true
gitlab_rails['ldap_servers'] = YAML.load <<-EOS
main:
  label: 'LDAP'
  host: '127.0.0.1' # или IP-адрес вашего LDAP-сервера
  port: 389
  uid: 'uid'
  bind_dn: 'cn=admin,dc=example,dc=com'
  password: 'adminadmin7'
  encryption: 'plain' # "start_tls" or "simple_tls" or "plain"
  verify_certificates: true
  smartcard_auth: false
  active_directory: false
  allow_username_or_email_login: false
  lowercase_usernames: false
  block_auto_created_users: false
  base: 'ou=GITLAB,dc=example,dc=com'
  user_filter: ''
  group_base: ''
  admin_group: ''
  sync_ssh_keys: false
EOS
[ ]
⌘ Help      ⌘ Write Out  ⌘ Where Is   ⌘ Cut        ⌘ Execute   ⌘ Location  ⌘ Undo
⌘ Exit      ⌘ Read File  ⌘ Replace    ⌘ Paste      ⌘ Justify   ⌘ Go To Line ⌘ Redo
```

Рисунок 25 – Измененный файл `gitlab.rb`

После выполнения настроек, необходимо применить изменения командой «`gitlab-ctl reconfigure`».

## 2.5 Разработка почтового сервиса

1. Первым шагом нам необходимо выполнить обновление пакетов. После выполнения команды «`sudo apt-get update`» будет получен обновленный список

пакетов, который можно использовать для установки или обновления ПО в системе.

2. После того, как пакеты обновлены, можем приступить к установке почтового сервера. Скачиваем с GitHub самый последний релиз iRedMail, в моем случае 1.6.8, который также и является самым стабильным, выполнив команду «`wget https://github.com/iredmail/iRedMail/archive/refs/tags/1.6.8.tar.gz`» . Далее разархивируем архивный файл командой «`tar -xzf 1.6.8.tar.gz`», перейдем в него: «`cd iRedMail-1.6.8/`». Затем необходимо выдать права на исполнение скрипта командой «`chmod +x iRedMail.sh`» и запускаем скрипт командой «`./iRedMail.sh`». После выполнения этих шагов откроется мастер настройки iRedMail (Рисунок 26).

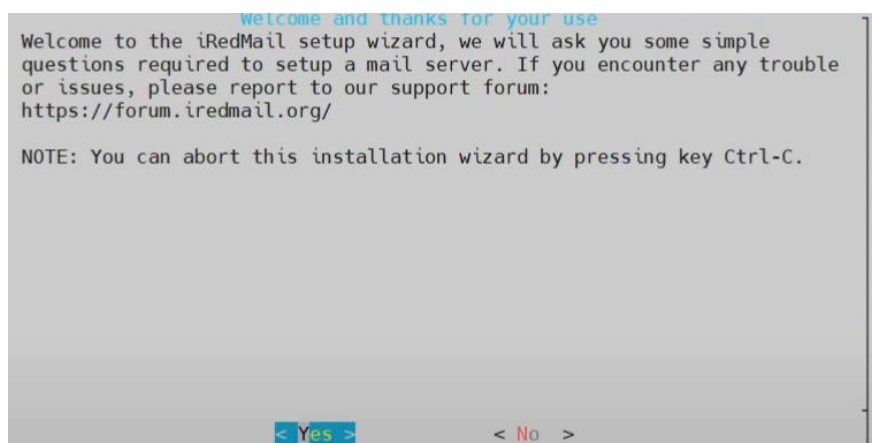


Рисунок 26 – Мастер настройки iRedMail

Указываем «Yes» и переходим далее.

3. На рисунке 27 показана настройка расположения почтовых ящиков. Для хранения почтовых ящиков пользователей укажем каталог по умолчанию: `/var/vmail`.

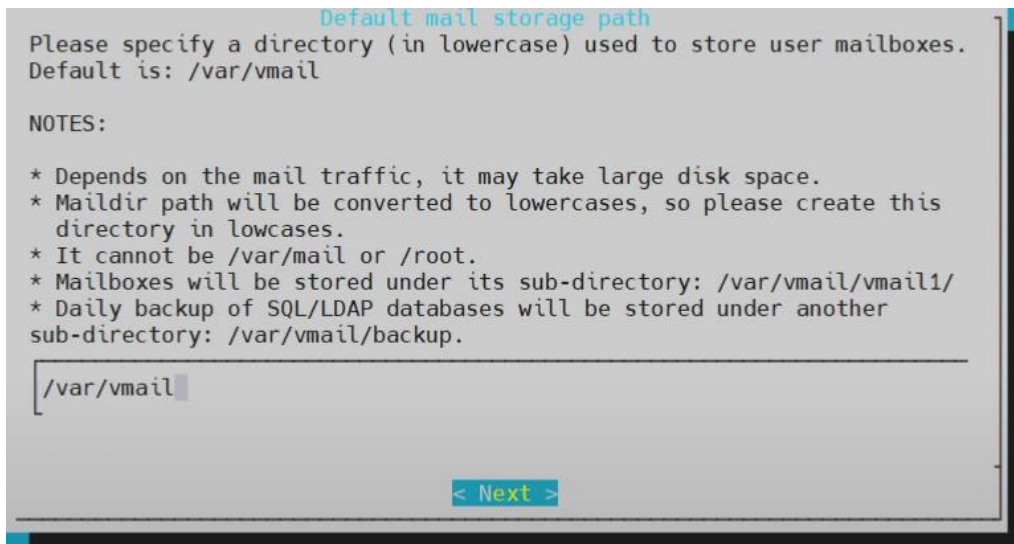


Рисунок 27 – Каталог для почтовых ящиков пользователей

4. Далее необходимо выбрать веб-сервер, который будет использоваться. Укажем Nginx.
5. Укажем базу данных из предложенных вариантов (Рисунок 28).

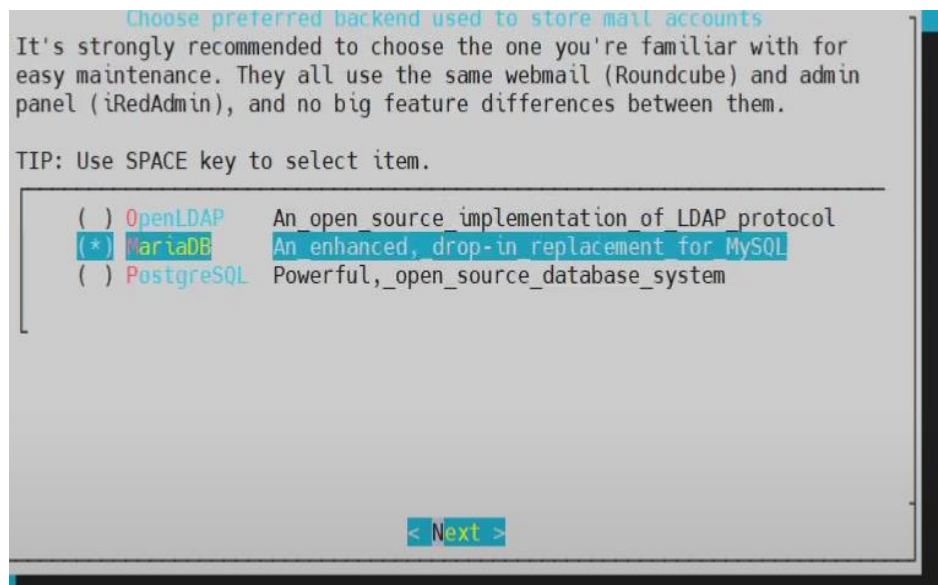


Рисунок 28 – Выбор базы данных

6. На следующем шаге необходимо задать пароль для администратора базы данных, также подтвердить его.

7. Затем нужно задать доменное имя. Следует обратить внимание, что доменное имя не может совпадать с hostname сервера. В моем случае доменное имя virtualbox57.org.

8. На данном этапе нужно задать пароль для администратора postmaster@virtualbox57.org.

9. На следующем шаге укажем компоненты в соответствии с рисунком 29: Roundcubemail – это веб-интерфейс электронной почты, который предоставляет пользователю возможность отправлять, получать и управлять своей почтой через веб-браузер. Он является одним из компонентов iRedMail и предоставляет удобный интерфейс для работы с почтовыми сообщениями.

Netdata – это система мониторинга производительности в реальном времени. Она предоставляет детальную информацию о нагрузке на сервер, использовании ресурсов и других параметрах производительности. Netdata включен в состав iRedMail для обеспечения мониторинга и анализа работы почтового сервера.

iRedAdmin – это веб-панель администратора для управления почтовым сервером. Он предоставляет администратору возможность управлять пользователями, доменами, почтовыми ящиками и другими аспектами почтового сервера. iRedAdmin облегчает администрирование и настройку iRedMail.

Fail2ban – это инструмент для защиты сервера от атак на основе перебора паролей или других подозрительных действий. Он анализирует журналы системы и блокирует IP-адреса, с которых происходят подозрительные активности. Fail2ban включен в iRedMail для повышения безопасности почтового сервера.



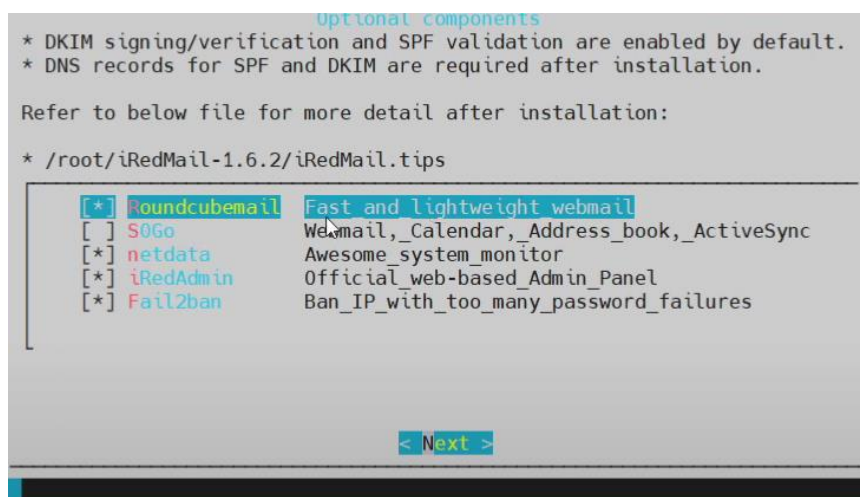


Рисунок 29 – Выбор устанавливаемых компонентов почтового сервера

После выполнения всех действий в мастере настроек, нажимаем Enter, после чего продолжается установка. В конце система спросит: «Использовать ли правила firewall, предоставляемые iRedMail по портам SSH?», соглашаемся, указав «у», затем нажав Enter (Рисунок 30).

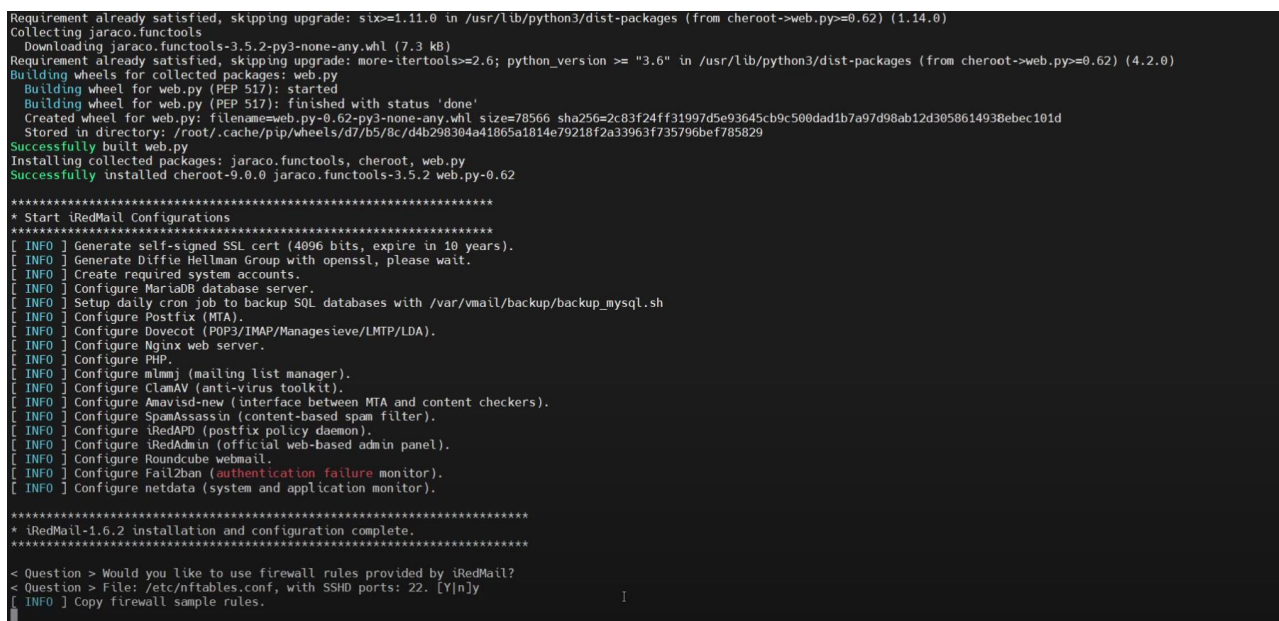


Рисунок 30 – Установка iRedMail

После завершения выполнения команды необходимо перезагрузить сервер.

Для того, чтобы войти в почту, необходимо перейти по адресу `https://ip_address/mail/`, но предварительно необходимо выпустить SSL сертификаты.

Для этого переходим в каталог, куда был установлен iRedMail. В нашем случае это `/root/iRedMail-1.6.8/tools` и запустим скрипт `generate_ssl_keys.sh`, который сгенерирует необходимый сертификат, а также закрытый ключ, необходимый для работы Nginx. Выполним данное действие командой «`sudo bash generate_ssl_keys.sh`». После успешного выполнения будет сообщение:

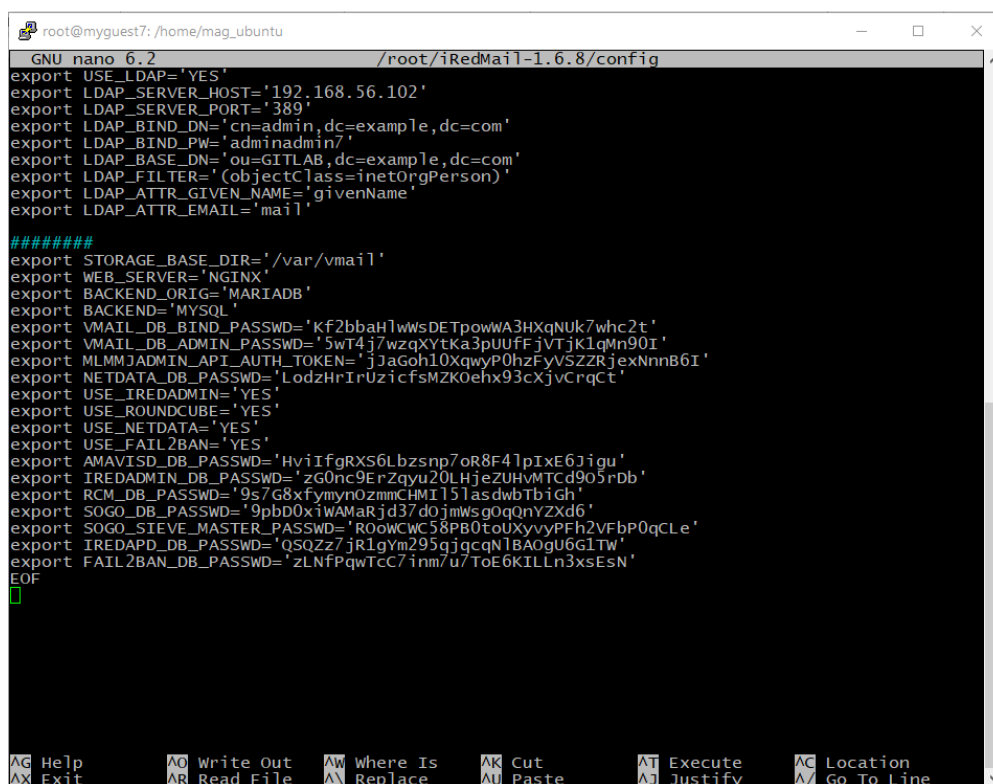
SSL keys were generated:

- `/root/iRedMail-1.6.8/tools/certs/iRedMail.crt`;
- `/root/iRedMail-1.6.8/tools/private/iRedMail.key`.

Выпуск сертификатов выполнен успешно.

Настройка интеграции LDAP. Для этого нужно выполнить следующие шаги:

1. Перейти в конфигурационный файл iRedMail для редактирования с помощью команды «`sudo nano /root/iRedMail-1.6.8/config`».



```
root@myguest7: /home/mag_ubuntu
GNU nano 6.2 /root/iRedMail-1.6.8/config
export USE_LDAP='YES'
export LDAP_SERVER_HOST='192.168.56.102'
export LDAP_SERVER_PORT='389'
export LDAP_BIND_DN='cn=admin,dc=example,dc=com'
export LDAP_BIND_PW='adminadmin7'
export LDAP_BASE_DN='ou=GITLAB,dc=example,dc=com'
export LDAP_FILTER='(objectClass=inetOrgPerson)'
export LDAP_ATTR_GIVEN_NAME='givenName'
export LDAP_ATTR_EMAIL='mail'

#####
export STORAGE_BASE_DIR='/var/vmail'
export WEB_SERVER='NGINX'
export BACKEND_ORIG='MARIADB'
export BACKEND='MYSQL'
export VMAIL_DB_BIND_PASSWORD='Kf2bbaH1wWsdETpowWA3HXqNuk7whc2t'
export VMAIL_DB_ADMIN_PASSWORD='5wT4j7wzqXytKa3pUUFjvTjK1qMn90I'
export MLMMJADMIN_API_AUTH_TOKEN='jJaGoh10XqwyP0hzFyVSZZRjexNnnB6I'
export NETDATA_DB_PASSWORD='LodzHrIrUzicfsMZK0ehx93cXjvCrqCt'
export USE_IREDADMIN='YES'
export USE_ROUNDROBIN='YES'
export USE_NETDATA='YES'
export USE_FAIL2BAN='YES'
export AMAVISD_DB_PASSWORD='HviIfgRXS6LbzsnP7oR8F41pIxE6Jigu'
export IREDADMIN_DB_PASSWORD='zG0nc9ErZqyu20LHjeZUHvMTCd9o5rDb'
export RCM_DB_PASSWORD='9s7G8xfymyn0zmmCHMI151asdwbTbiGh'
export SOGO_DB_PASSWORD='9pbD0xiWAMARjd37d0jmwsg0qQnYZXd6'
export SOGO_SIEVE_MASTER_PASSWORD='Ro0WCWC58PB0toUxYvyPFh2VFbP0qCle'
export IREDAPD_DB_PASSWORD='QSQZz7jR1gYm295gjqcQn1BA0gu6G1TW'
export FAIL2BAN_DB_PASSWORD='zLNfPqwTcC7inm7u/ToE6KILLn3xsEsN'
EOF
^
# Help      ^O Write Out  ^W Where Is   ^K Cut        ^J Execute    ^C Location
# Exit      ^R Read File  ^X Replace    ^U Paste     ^_ Justify    ^/ Go To Line
```

Рисунок 31 – Настройки LDAP в файле config

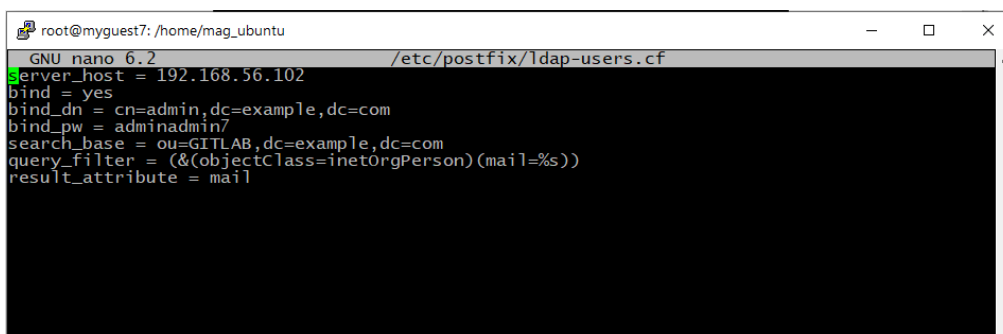
2. Перейти в файл «*sudo nano /etc/dovecot/dovecot-ldap.conf.ext*» и указать настройки (Рисунок 32).



```
root@myguest7: /home/mag_ubuntu
GNU nano 6.2 /etc/dovecot/dovecot-ldap.conf.ext
hosts = 192.168.56.102
dn = cn=admin,dc=example,dc=com
dnpass = adminadmin7
base = ou=GITLAB,dc=example,dc=com
ldap_version = 3
user_attrs = homeDirectory=home,uidNumber=uid,gidNumber=gid
user_filter = (&(objectClass=inetOrgPerson)(uid=%u))
pass_filter = (&(objectClass=inetOrgPerson)(uid=%u))
```

Рисунок 32 – Настройки LDAP в файле dovecot-ldap.conf.ext

3. Перейти в конфигурационный файл iRedMail для редактирования с помощью команды «*sudo nano /etc/postfix/ldap-users.cf*» и указать соответствующие настройки (Рисунок 33).



```
root@myguest7: /home/mag_ubuntu
GNU nano 6.2 /etc/postfix/ldap-users.cf
server_host = 192.168.56.102
bind = yes
bind_dn = cn=admin,dc=example,dc=com
bind_pw = adminadmin7
search_base = ou=GITLAB,dc=example,dc=com
query_filter = (&(objectClass=inetOrgPerson)(mail=%s))
result_attribute = mail
```

Рисунок 33 – Настройки в файле ldap-users.cf

4. Перезапускаем службы командами «*sudo systemctl restart dovecot*» и «*sudo systemctl restart postfix*».

Если все настроено правильно, пользователи, существующие в LDAP, должны иметь возможность входить в iRedMail, и их почтовые ящики будут создаваться автоматически при первом входе.

## **2.6 Выводы по главе**

Во второй главе исследования была разработана схема студенческого конструкторского бюро. Эта схема предусматривает наличие нескольких компьютеров, которые будут подключены к центральному серверу.

В рамках главы были установлены и настроены три важные составляющие на серверах для работы сотрудников: LDAP, GitLab и iRedMile. Эти программы являются неотъемлемой частью функциональности бюро и обеспечивают его эффективную работу. Была продемонстрирована установка и настройка ПО.

В результате исследований можно сделать вывод, что разработанная схема студенческого конструкторского бюро и установленные программы LDAP, GitLab и iRedMile обеспечивают гибкость и эффективность работы бюро.

## **3 Обоснование и выбор аппаратного обеспечения**

### **3.1 Структурная схема системы разработанной модели**

Система состоит из одного сервера и двух виртуальных машин, развернутых на нем, которые работают под управлением операционной системы Linux Ubuntu версии 22.04.4. Основные компоненты системы включают:

**GitLab:** Платформа для DevOps, включающая функции репозитория исходного кода, CI/CD, управления проектами и отслеживания задач. Взаимодействует с пользователями через веб-интерфейс и CLI (Command Line Interface - командный интерфейс), использует хранилище данных для проектов.

**LDAP (Lightweight Directory Access Protocol):** Служба каталогов для управления пользователями и аутентификацией. Взаимодействует с GitLab и iRedMail для аутентификации пользователей.

**iRedMail:** Комплексное решение для электронной почты, включающее Postfix (SMTP-сервер), Dovecot (IMAP/POP3-сервер) и другие компоненты. Взаимодействует с пользователями через почтовые клиенты, обрабатывает SMTP, IMAP и POP3 запросы.

Принципы функционирования системы:

- GitLab предоставляет интерфейс для разработчиков, которые могут копировать репозиторий, вносить изменения и отправлять их обратно. Система также обрабатывает запросы на объединение изменений и выполняет автоматические сборки и тестирование.

- LDAP управляет пользователями и их аутентификацией, позволяя централизованно хранить учетные данные и предоставлять доступ к различным системам.

- iRedMail обрабатывает отправку и получение электронной почты, обеспечивает доступ к почтовым ящикам и использует LDAP для аутентификации пользователей.

## 3.2 Реализация серверной инфраструктуры

Для обеспечения эффективной работы серверной инфраструктуры, необходимо выбрать подходящее аппаратное обеспечение для сервера, учитывая рабочую нагрузку и будущий рост.

Нагрузка на GitLab: GitLab требует значительных ресурсов, обработке запросов на слияние и хранении большого количества репозитория. Минимальные рекомендации по процессору (6 ядер) и оперативной памяти (16 GB) должны обеспечить базовую производительность для 15 пользователей.

Нагрузка на LDAP: LDAP в данном случае используется для аутентификации и авторизации пользователей, что обычно не требует больших ресурсов. Минимальные рекомендации (2 ядра процессора, 4 GB оперативной памяти) для сервера должны быть достаточными для работы с LDAP.

Нагрузка на iRedMail: iRedMail обрабатывает почтовые сообщения и взаимодействует с LDAP для аутентификации пользователей. Минимальные рекомендации (4 ядра процессора, 16 GB RAM) для сервера, на котором будет работать iRedMail, должны быть достаточными для обслуживания 15 пользователей.

Сервер, на котором будет работать разработанная схема инфраструктуры студенческого конструкторского бюро, должен быть оборудован высокопроизводительными процессорами и достаточным объемом оперативной памяти для поддержки обоих сервисов. Основные требования к оборудованию включают:

- процессор: минимум 6 ядер (рекомендуется 6-ядерный процессор Intel Xeon или AMD EPYC);
- оперативная память: минимум 16 GB ECC RAM;
- хранилище: SSD объемом 512 GB для обеспечения скорости доступа к данным;
- сетевая карта: 1 Gbps Ethernet (минимум);
- операционная система: Ubuntu Server 20.04 LTS.

### **3.3 Выводы по главе**

Сервер, на котором будет работать разработанная схема инфраструктуры студенческого конструкторского бюро, должен быть оборудован высокопроизводительными и надежными компонентами, чтобы обеспечить стабильную и эффективную работу систем. Сервер требует большей вычислительной мощности и объема оперативной памяти. Использование SSD для хранения данных на обоих серверах обеспечит высокую скорость доступа и обработки данных, что является критически важным для производительности системы.

Рассчитанные требования к ресурсам и рекомендованное аппаратное обеспечение помогут обеспечить стабильную работу системы, учитывая максимальную нагрузку в 15 пользователей. Данный сервер должны быть готовы к работе с текущими требованиями и иметь возможность для дальнейшего масштабирования при необходимости.

## 4 Описание методики тестирования

В данной главе будет выполнено тестирование установленных ранее ПО: LDAP, GitLab и iRedMail.

В таблице 1 указаны критерии, по выполнению которых будет проверяться работоспособность установленных программ.

Таблица 1 – Критерии проверки

Имя	Критерии
LDAP	Установка соединения в LDAP Browser
GitLab	Установка веб-соединения
	Авторизация
	Создание проекта
	Клонирование репозитория
	Создание новой ветки
	Внесение изменений в файле из ветки
	Выполнить commit
	Перенести в репозиторий изменения, выполненные ранее
	Проверить доступность к проекту из другого аккаунта
iRedMail	Проверка доступности
	Возможность авторизации
	Отправка писем

### 4.1 Результаты тестирования LDAP

После выполненной установки и настройки в п. 2.3, выполним проверку чтобы убедиться в работоспособности.

Для проверки работоспособности протокола извне, будем использовать программу LDAPSoft Ldap Browser, которая позволит просмотреть LDAP каталоги и его содержимое в удобном виде.

Подключаемся к через LDAPSoft Ldap Browser, введя IP-адрес и указав порт, по которому происходит соединение (Рисунок 34).



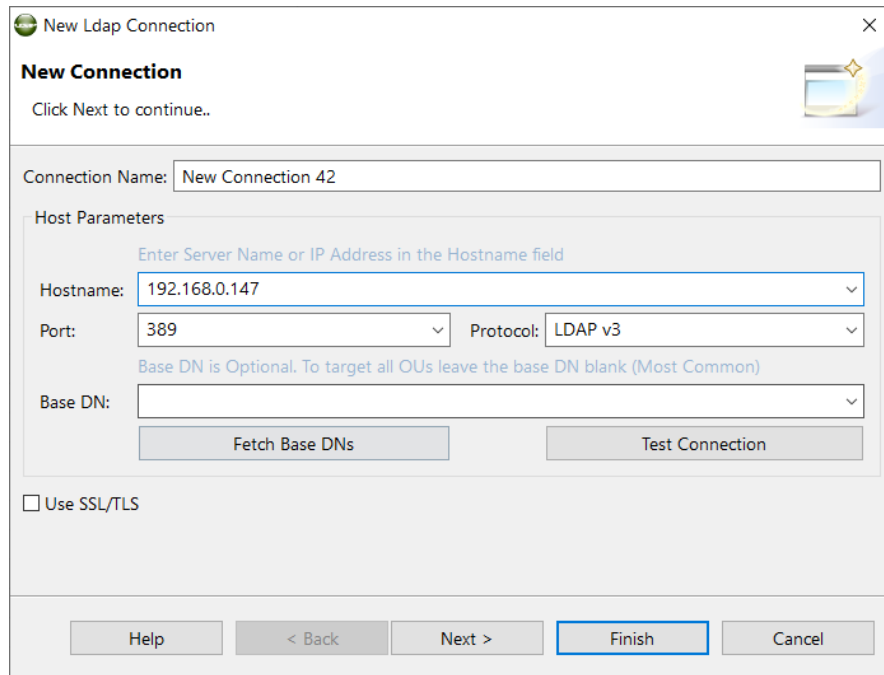


Рисунок 34 – Настройки подключения

При успешном соединении мы увидим каталог GITLAB, который создавали ранее и существующих пользователей в данном каталоге.

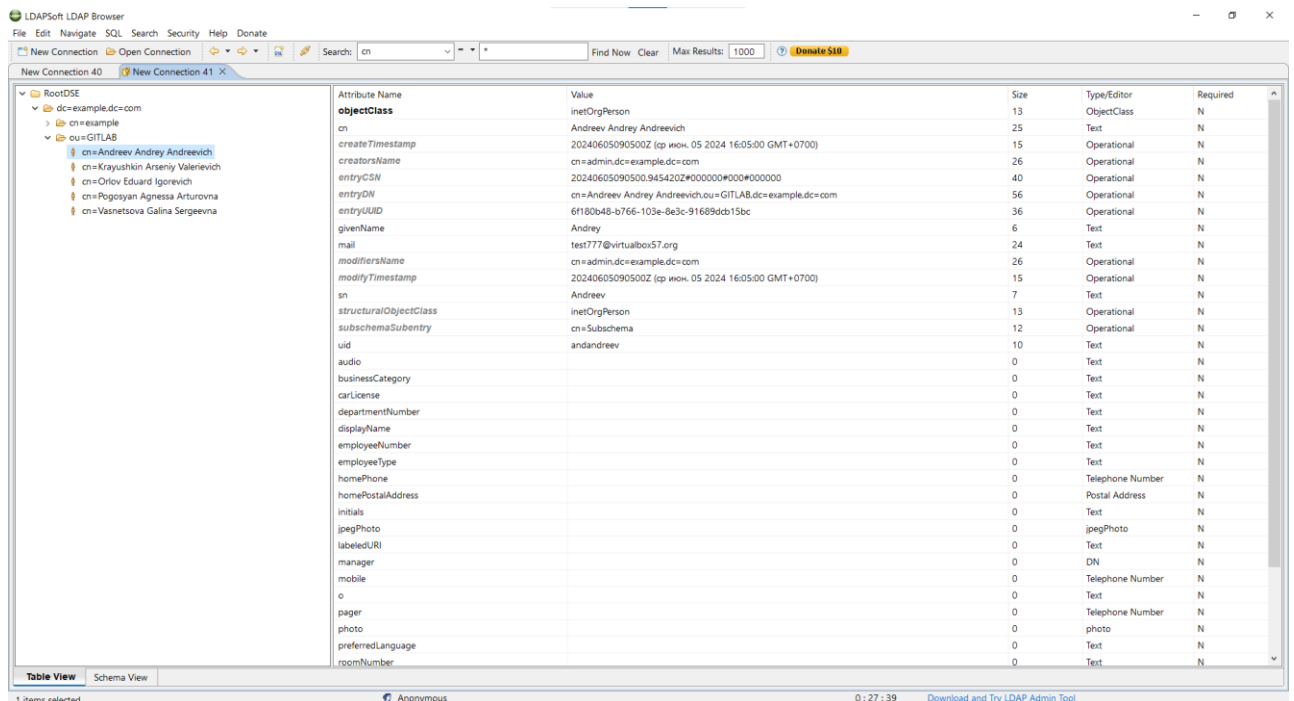


Рисунок 35 – Демонстрация содержимого LDAP

## 4.2 Результаты тестирования GitLab

Перейдем в Google Chrome по адресу `http://192.168.0.147:8888/` для подключения к GitLab. Наблюдаем, что GitLab доступен и пользователь может авторизоваться (Рисунок 36).

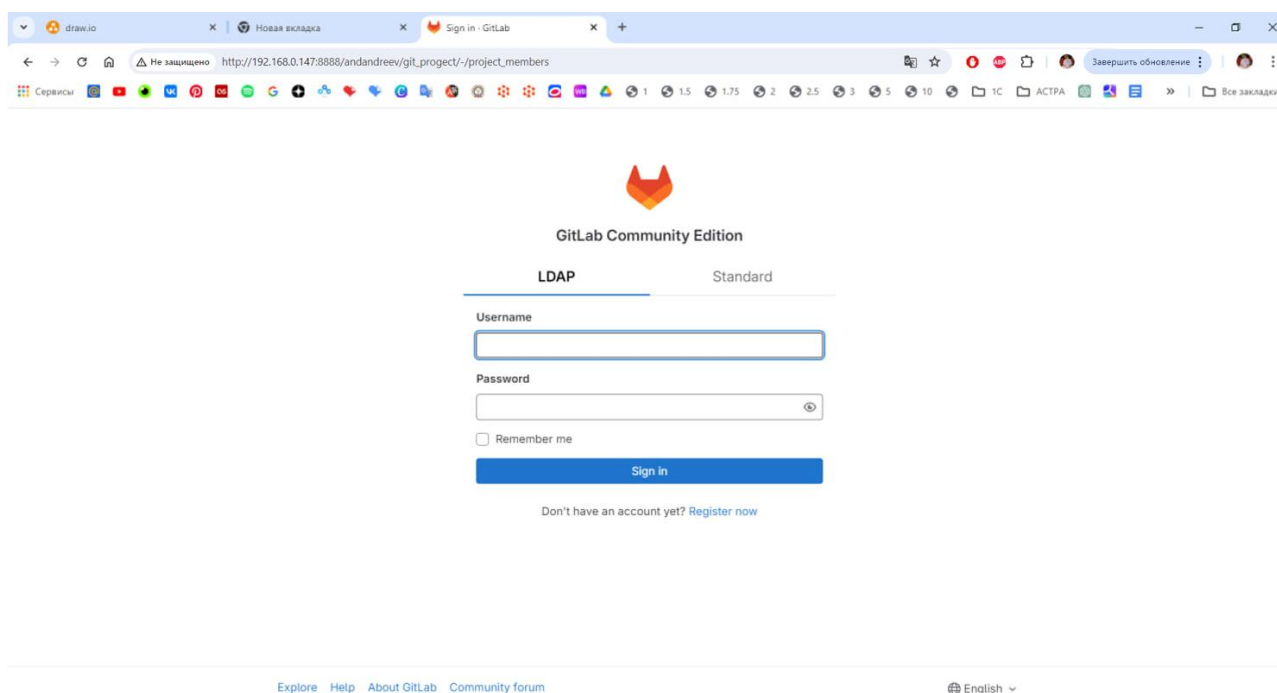


Рисунок 36 – Страница авторизации в GitLab

После успешной авторизации создадим новый проект. Проект будем создавать в личном кабинете пользователя Andreev Andrey Andreevich.

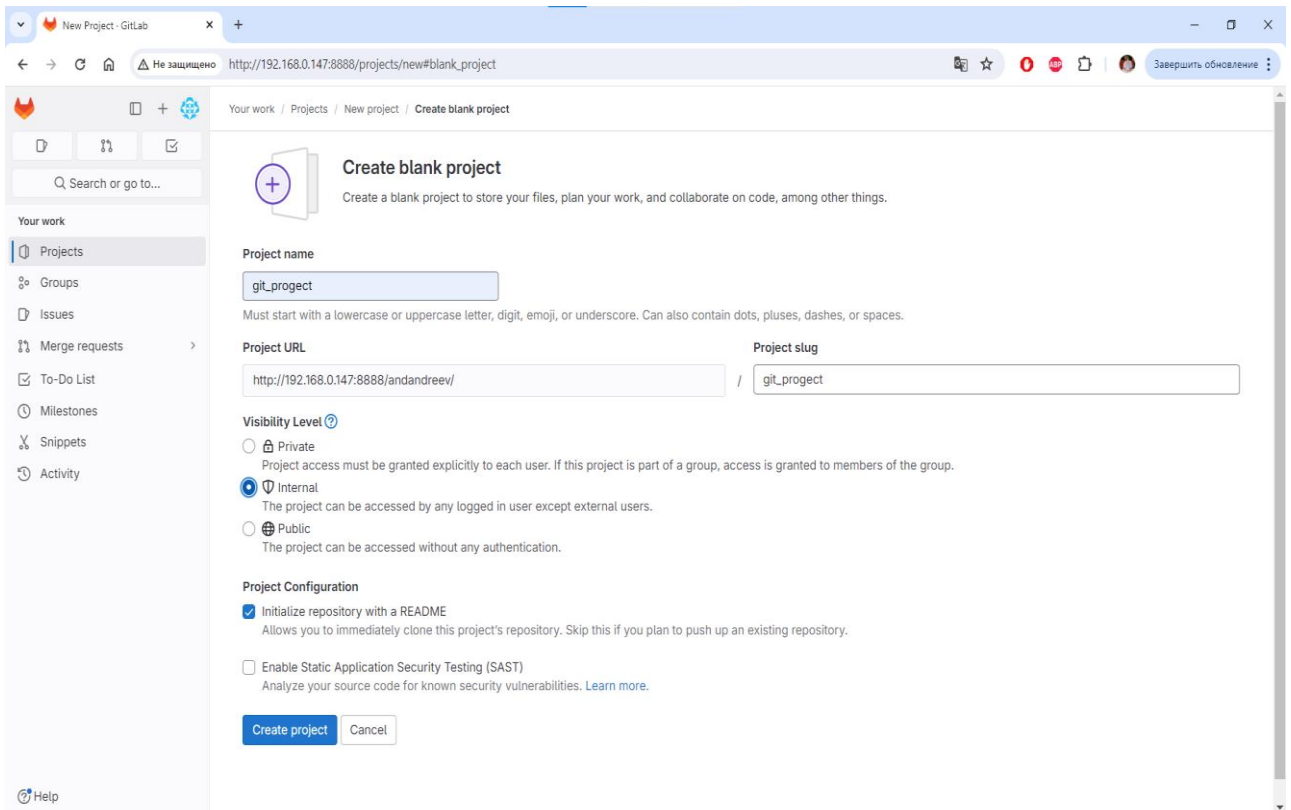


Рисунок 37 – Создание нового проекта

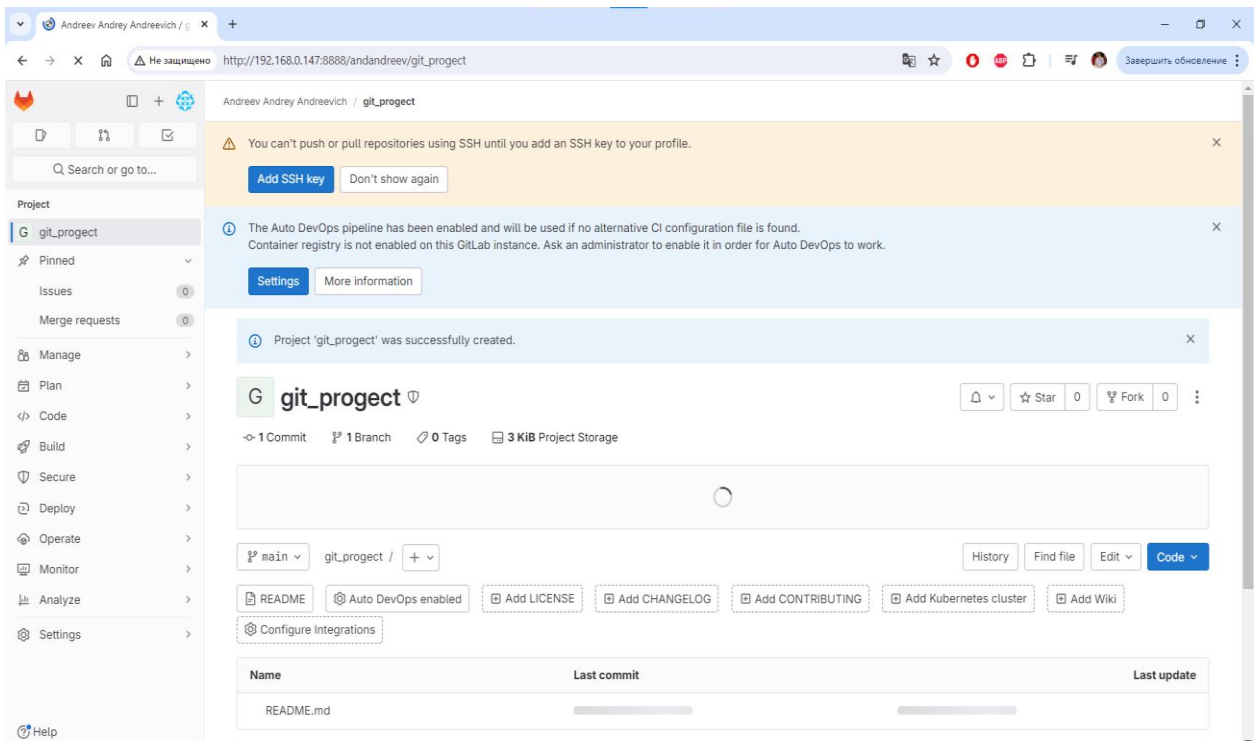


Рисунок 38 – Успешное создание проекта

Теперь в терминале выполним следующие действия:

1. Скопировать ссылку, раскрыв меню кнопки «Code» (Рисунок 39).

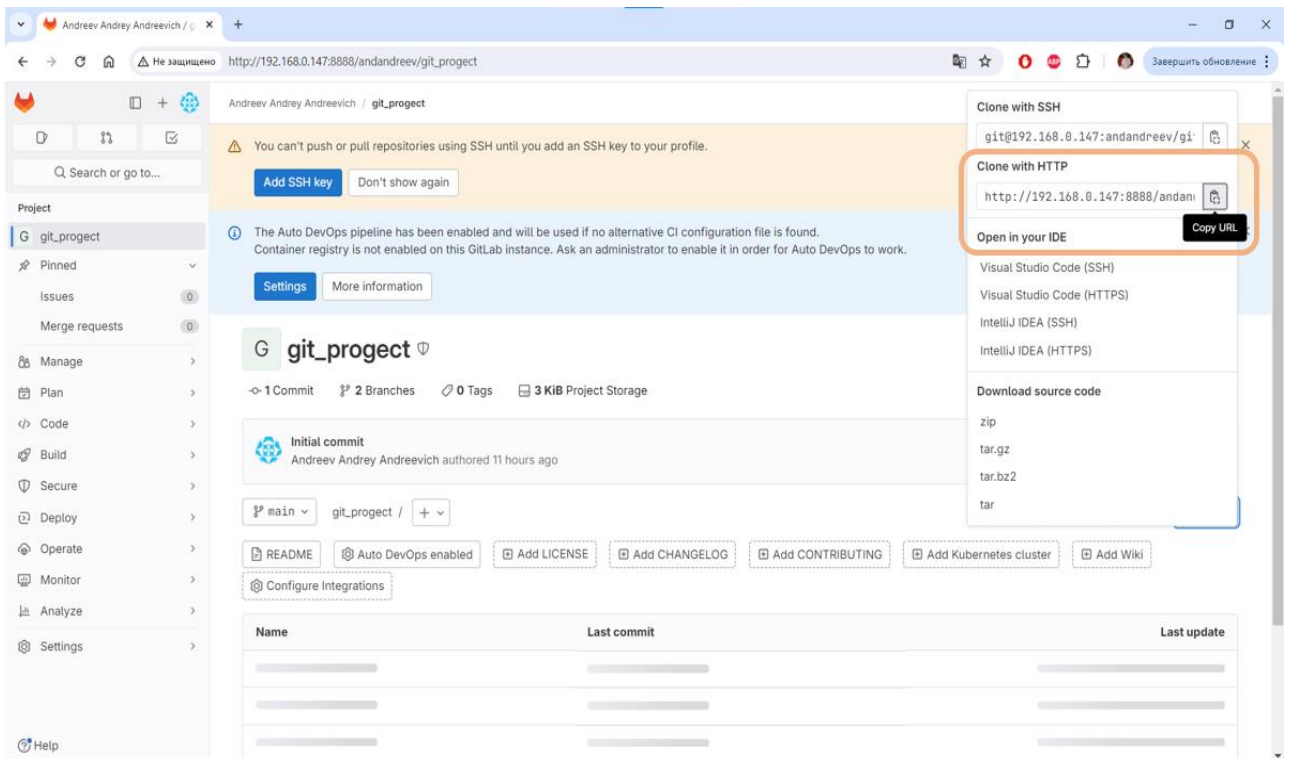


Рисунок 39 – Ссылка для клонирования репозитория

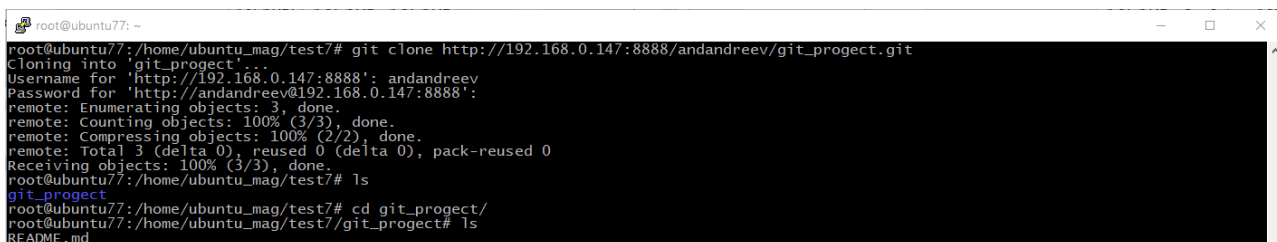
2. Создадим каталог `test7` командой «`mkdir test7`», в который будем клонировать созданный проект.

3. Создать клон репозитория командой «`git clone http://192.168.0.147:8888/andandreev/git_project.git`» (Рисунок 40).

4. Командой «`ls`» убедимся, что предыдущий шаг был успешно выполнен. Перейдем в каталог «`git_project`» командой «`cd git_project`». Используя «`ls`» увидим наличие одного файла `README.md` (Рисунок 40). Создадим файл `file.txt` командой «`touch file.txt`»

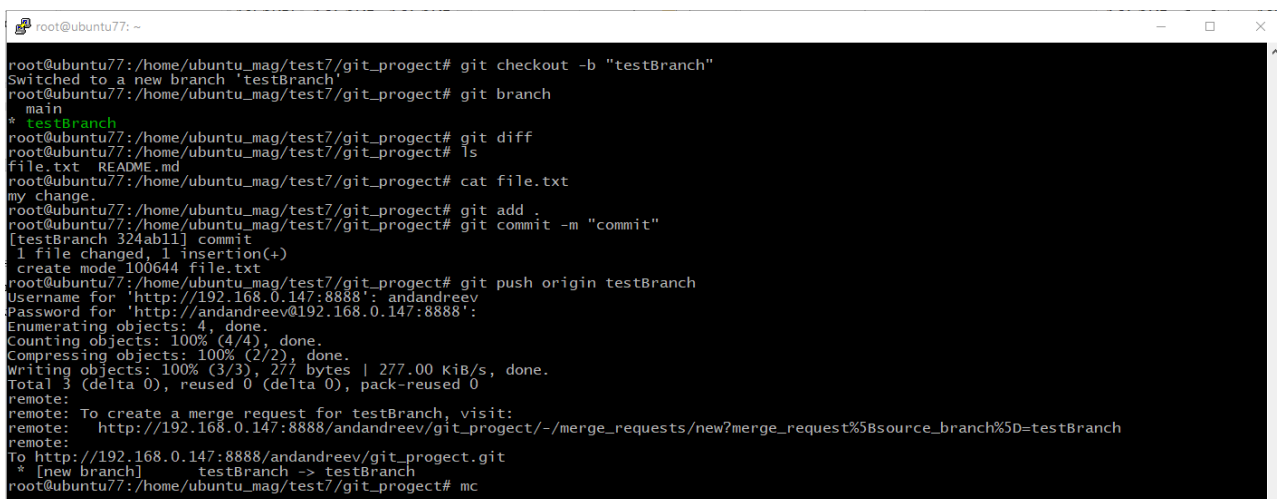
5. Далее выполним «`git checkout -b "testBranch"`» для создания и перемещения в новую ветку 'testBranch'. При успешном выполнении команды увидим результат «Switched to a new branch 'testBranch'». Используя команду «`git branch`» увидим все существующие ветки и в какой именно мы находимся (выделено зеленым). В нашем случае это ветки «`main`» и «`testBranch`», и находимся в «`testBranch`» (Рисунок 41).

6. Внесем изменения в файл file.txt.
7. Команда «git add .» добавляет все файлы, которые были созданы или изменены. Для добавления конкретного файла стоит использовать команду «git add наименование\_файла».
8. Пробуем выполнить commit.
9. Затем выполним «git push origin testBranch». Команда git push выполняет отправку недавних коммитов с компьютера локального репозитория на сервер с удаленным репозиторием. Система попросит ввести логин и пароль для выполнения данного действия.



```
root@ubuntu77: ~
root@ubuntu77:/home/ubuntu_mag/test7# git clone http://192.168.0.147:8888/andandreev/git_project.git
Cloning into 'git_project'...
Username for 'http://192.168.0.147:8888': andandreev
Password for 'http://andandreev@192.168.0.147:8888':
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
root@ubuntu77:/home/ubuntu_mag/test7# ls
git_project
root@ubuntu77:/home/ubuntu_mag/test7# cd git_project/
root@ubuntu77:/home/ubuntu_mag/test7/git_project# ls
README.md
```

Рисунок 40 – Клонирование репозитория



```
root@ubuntu77: ~
root@ubuntu77:/home/ubuntu_mag/test7/git_project# git checkout -b "testBranch"
Switched to a new branch 'testBranch'
root@ubuntu77:/home/ubuntu_mag/test7/git_project# git branch
* main
* testBranch
root@ubuntu77:/home/ubuntu_mag/test7/git_project# git diff
root@ubuntu77:/home/ubuntu_mag/test7/git_project# ls
file.txt README.md
root@ubuntu77:/home/ubuntu_mag/test7/git_project# cat file.txt
my change.
root@ubuntu77:/home/ubuntu_mag/test7/git_project# git add .
root@ubuntu77:/home/ubuntu_mag/test7/git_project# git commit -m "commit"
[testBranch 324ab11] commit
1 file changed, 1 insertion(+)
 create mode 100644 file.txt
root@ubuntu77:/home/ubuntu_mag/test7/git_project# git push origin testBranch
Username for 'http://192.168.0.147:8888': andandreev
Password for 'http://andandreev@192.168.0.147:8888':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 277 bytes | 277.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: To create a merge request for testBranch, visit:
remote: http://192.168.0.147:8888/andandreev/git_project/-/merge_requests/new?merge_request%5Bsource_branch%5D=testBranch
remote:
To http://192.168.0.147:8888/andandreev/git_project.git
 * [new branch] testBranch -> testBranch
root@ubuntu77:/home/ubuntu_mag/test7/git_project# mc
```

Рисунок 41 – Выполнение команд

10. Перейдя в веб интерфейс, можем наблюдать наши ветки репозитория (Рисунок 42).

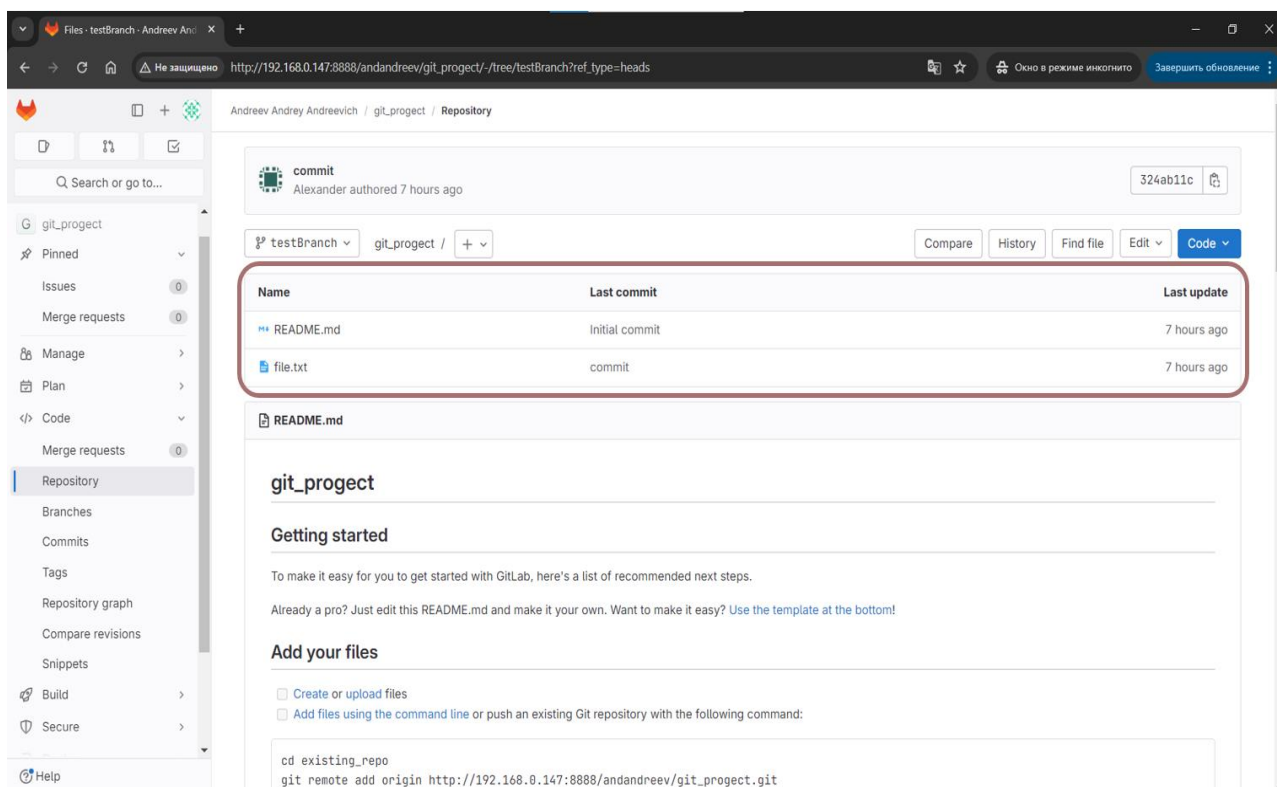


Рисунок 42 – Содержимое ветки testBranch

11. Чтобы был доступ к данному проекту у другого пользователя, нужно пригласить участника. Для этого необходимо перейти на страницу проекта, воспользоваться кнопкой «Invite members», после чего откроется окно, в котором надо заполнить имя пользователя, которого хотим пригласить и его роль (Рисунок 43). В данном случае добавляем пользователя Orlov Eduard с ролью Developer. После заполнения нажимаем «Invite» и наблюдаем результат (Рисунок 44). В личном кабинете пользователя Orlov Eduard Igorevich видим доступный репозиторий, в котором перейдем в ветку «testBranch» и увидим ранее измененный файл (Рисунок 45).

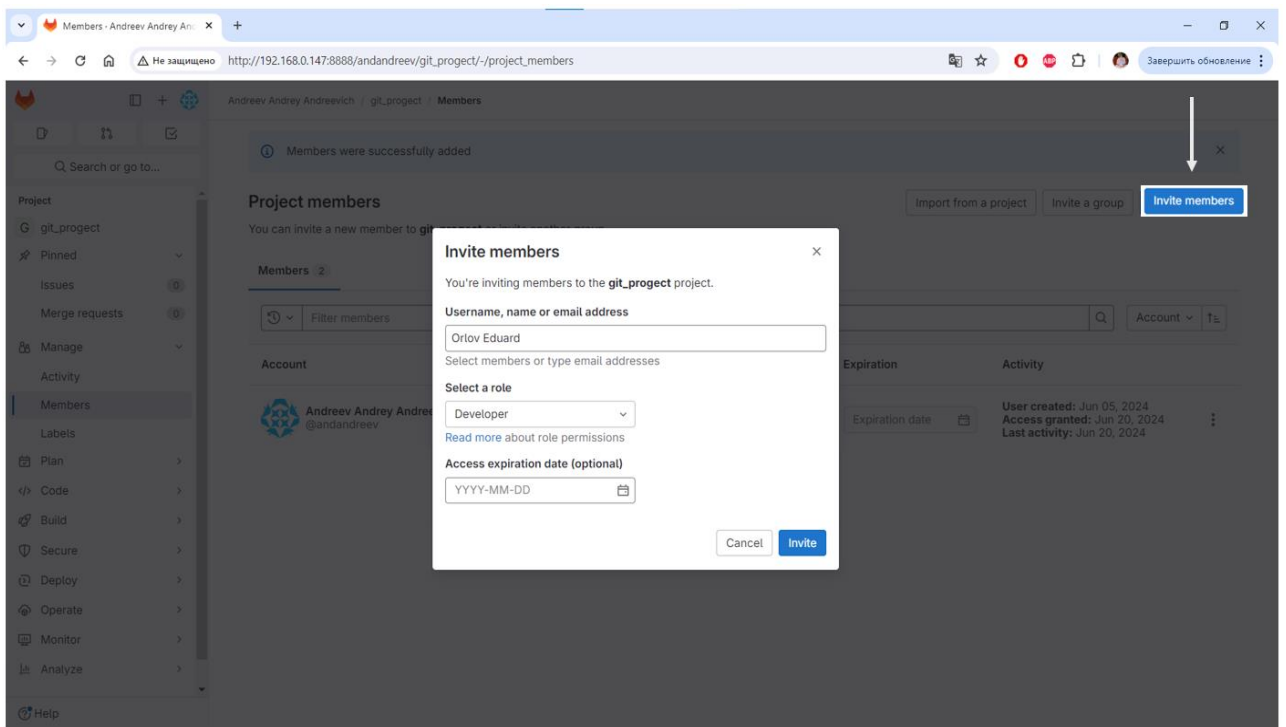


Рисунок 43 – Приглашение участника Orlov Eduard Igorevich

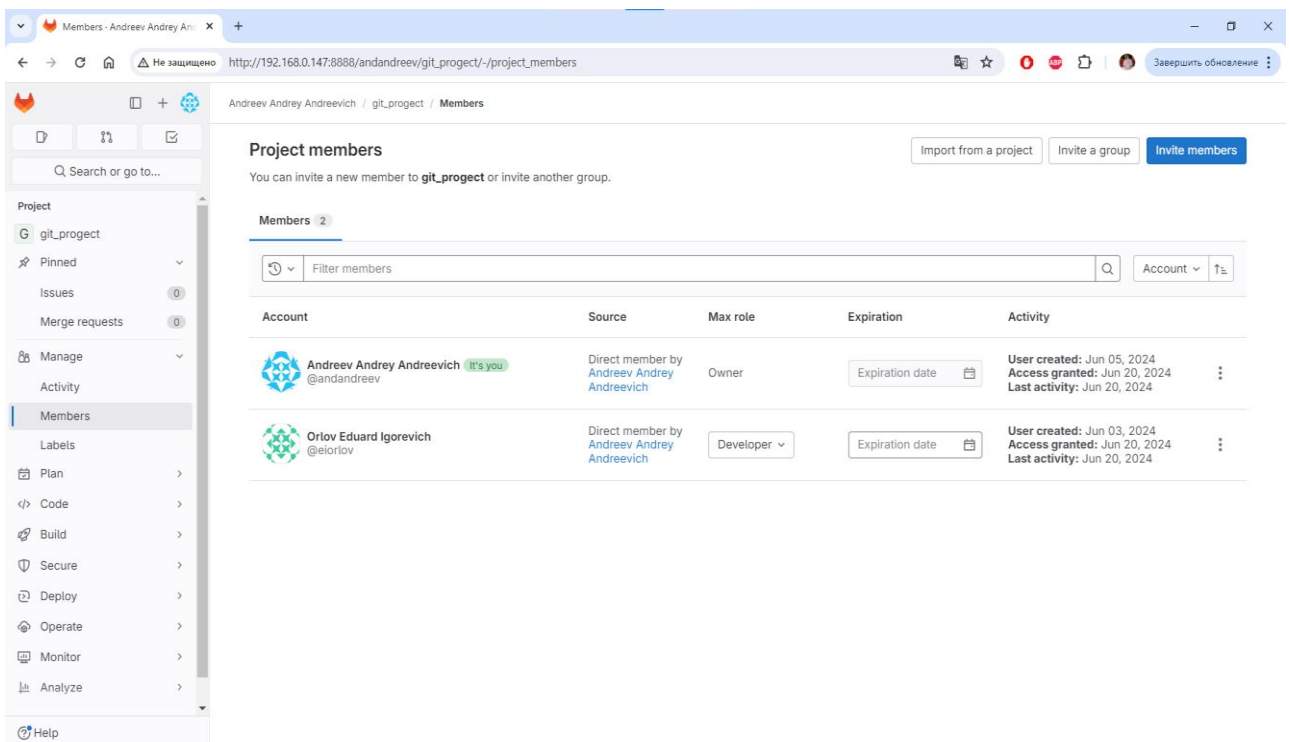


Рисунок 44 – Результат добавления участника в проект

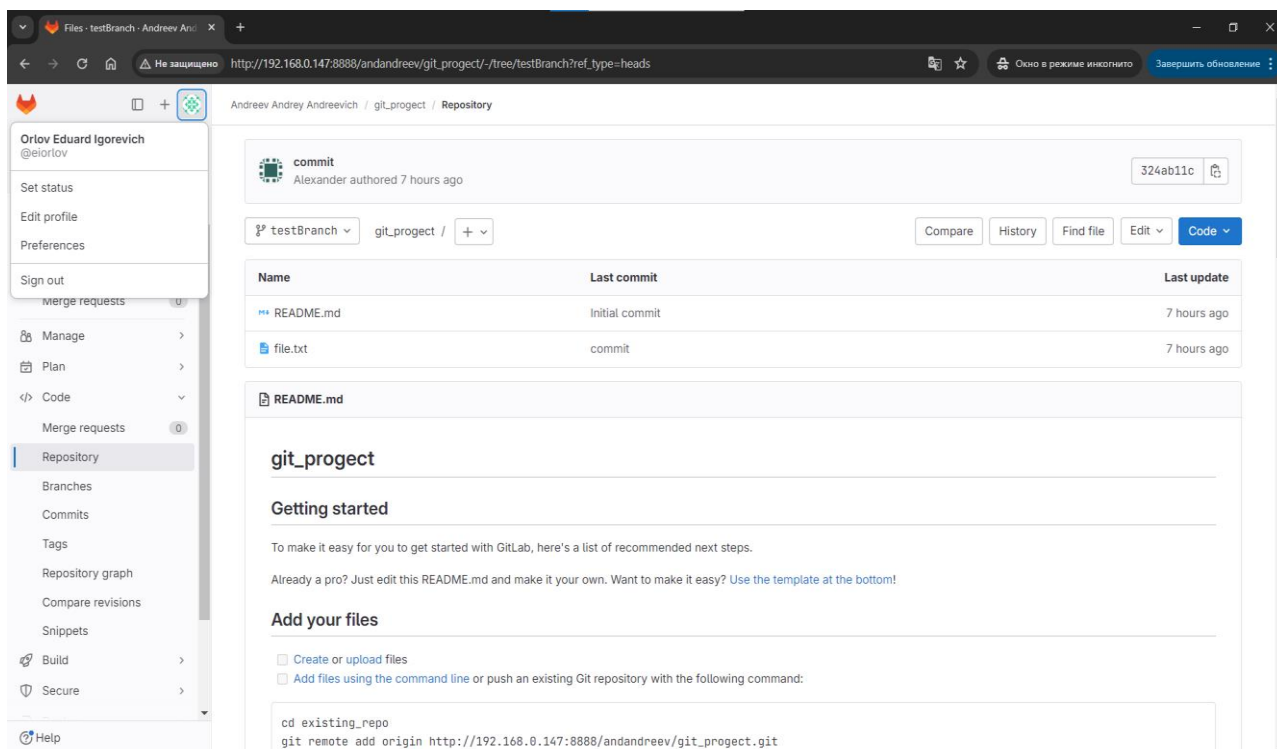


Рисунок 45 – Отображение проекта у приглашенного участника

### 4.3 Результаты тестирования iRedMail

В п. 2.5 был установлен и настроен iRedMail. Проверяем в браузере доступность почтового сервера из веб-браузера.

1. Для этого в адресной строке браузера Google Chrome укажем путь до iRedMail в виде `https://ip_address/iredadmin/`. На рисунке 46 можем наблюдать доступность почтового сервера iRedMail.

2. Пробуем осуществить вход в панель администратора, где Username - `postmaster@virtualbox57.org`, а Password – значение, которое мы задавали на шаге 8 п. 2.5.



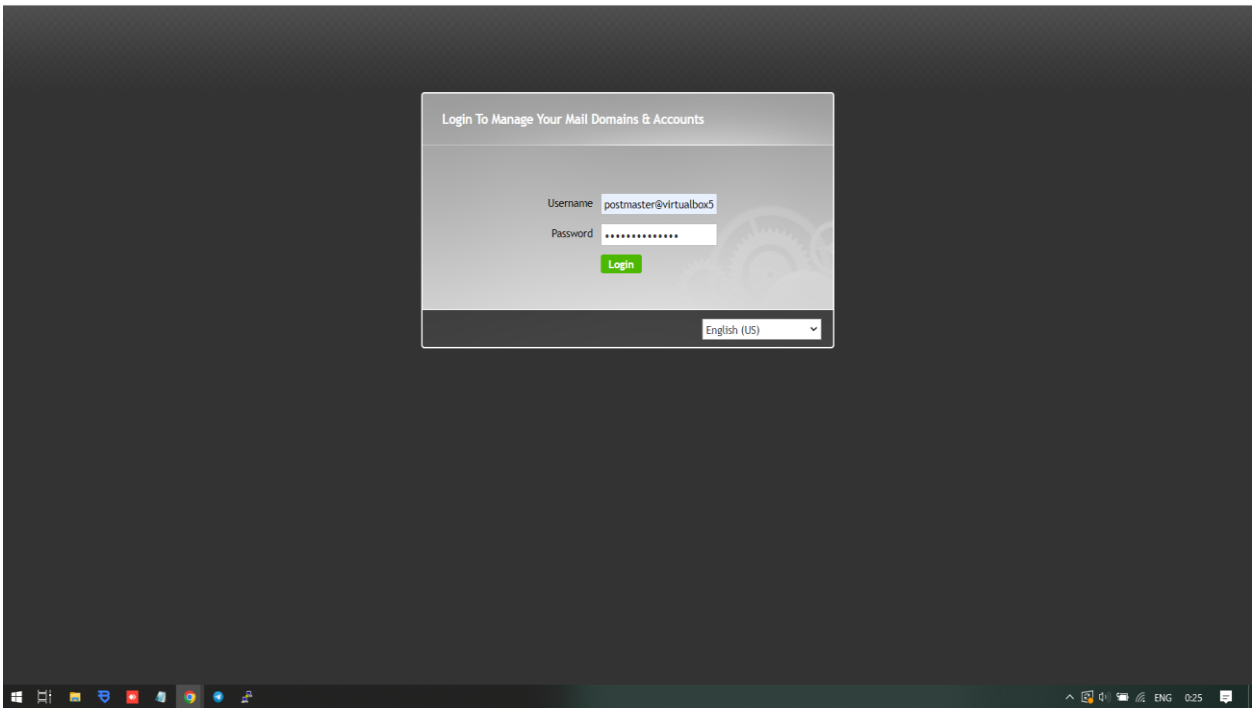


Рисунок 46 – Подключение к почте из браузера

После успешной авторизации можем наблюдать панель администратора.

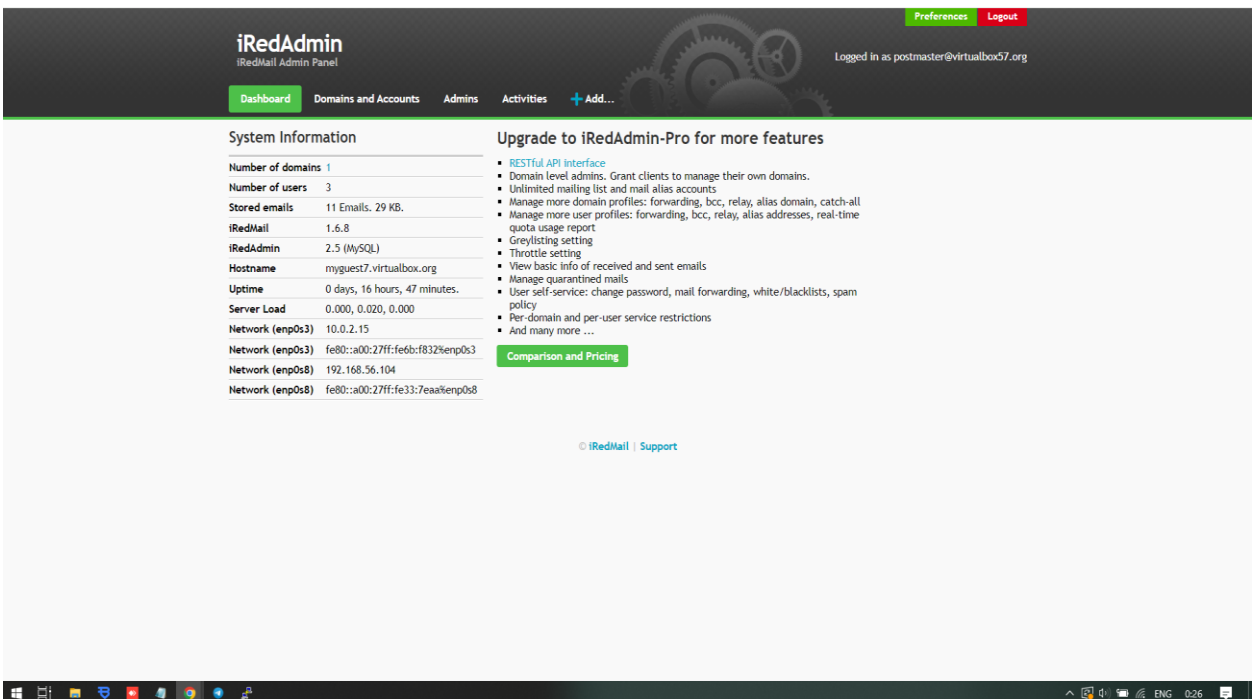


Рисунок 47 – Панель администратора

Чтобы войти в почту, необходимо перейти по адресу [https://ip\\_address/mail/](https://ip_address/mail/). При входе можно увидеть страницу со входящими письмами (Рисунок 48).

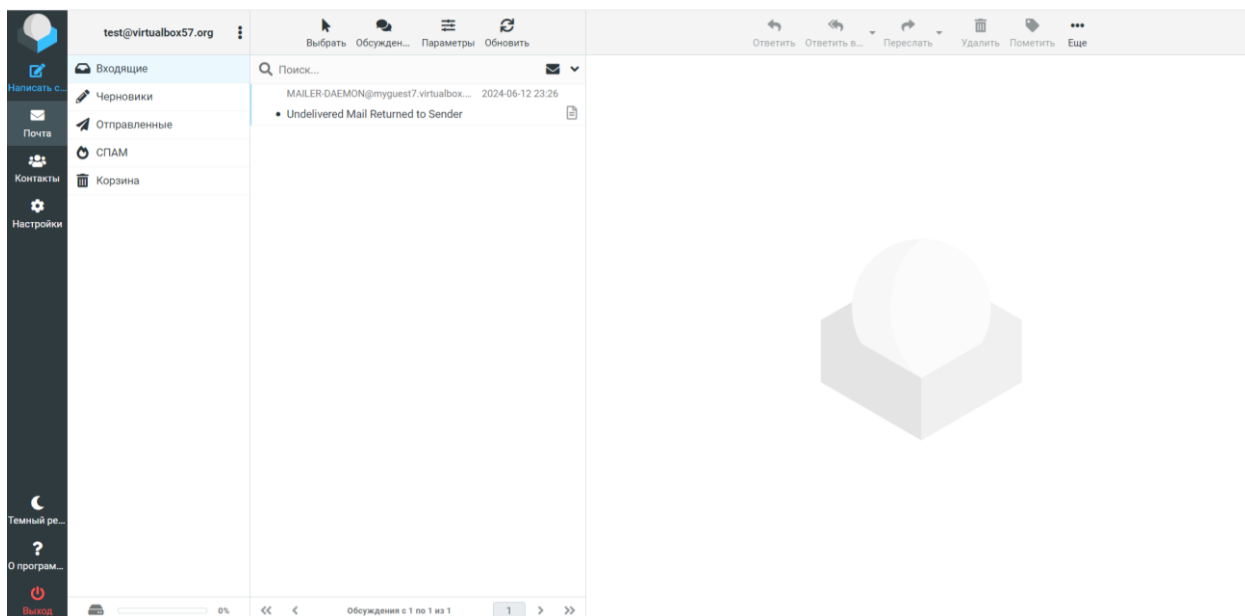


Рисунок 48 – Страница «Входящие»

Сформируем сообщение для отправки двум пользователям.

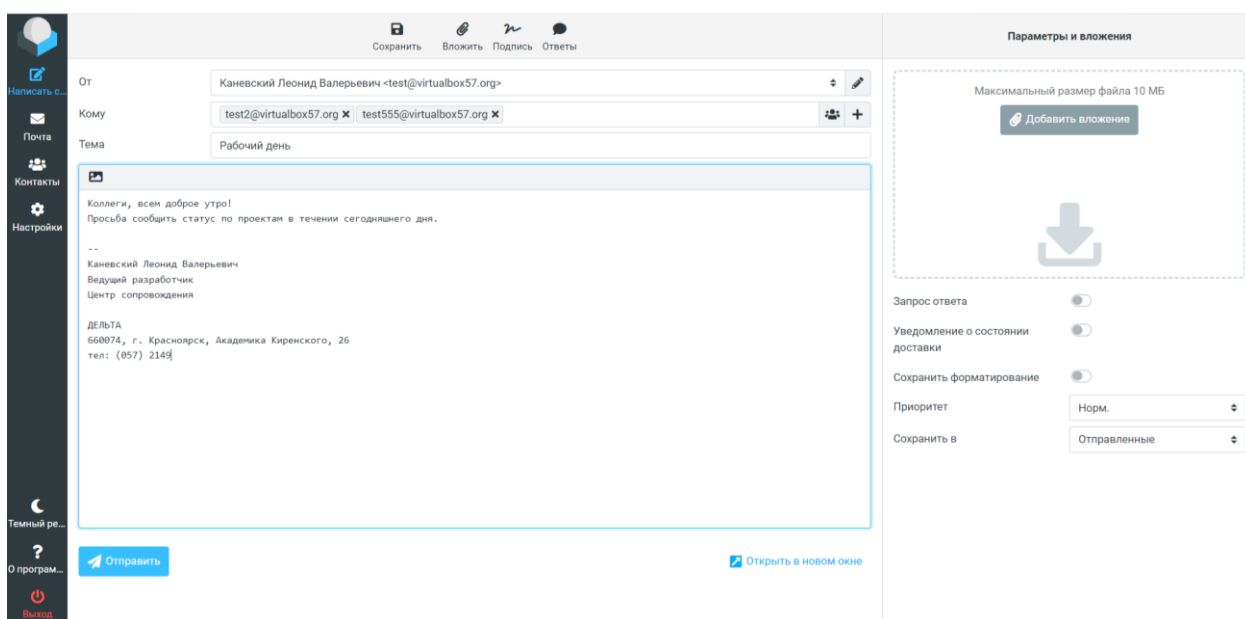


Рисунок 49 – Формирование письма

После отправки письма, перейдем в личный кабинет получателя test555@virtualbox57.org для того, чтобы убедиться, что письмо получено.

Ниже на скриншоте можно увидеть входящее письмо, которые ранее отправлено из почты test@virtualbox57.org (Рисунок 50).

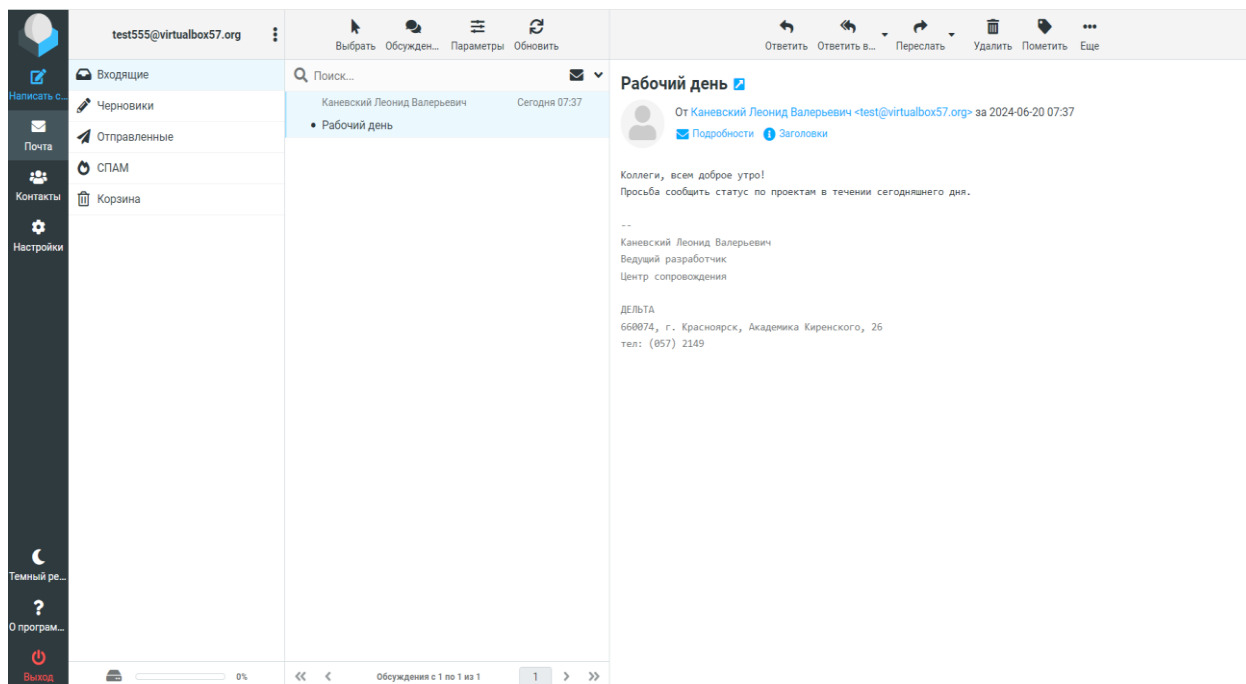


Рисунок 50 – Полученное письмо

#### 4.4 Выводы по главе

В данной главе было проведено тестирование установленных программных решений. Для этого был составлен набор критериев проверки, при выполнении которых, в случае отсутствия ошибок и нарушений, тестирование считается успешным.

В ходе тестирования ошибок не было обнаружено. Исходя из всех продемонстрированных действий, каждый критерий проверки из таблицы 1 выполнен успешно.

## ЗАКЛЮЧЕНИЕ

В процессе выполнения задания было реализовано исследование существующих решений ПО для различных целей. Для определения подходящих вариантов был выполнен обзор и сравнение ПО в следующих аспектах:

- авторизация пользователей в СКБ, а именно был выбран протокол LDAP, который удовлетворяет всем требованиям;
- система коммуникаций, а именно были проанализированы решения и выбрано наиболее подходящее – iRedMail;
- были исследованы решения систем контроля версий, из которых было выбрано оптимальное ПО в виде решения GitLab.

Также в ходе выполнения была разработана структура СКБ для взаимодействия пользователей, а также выбор основных инструментов для практической реализации позволили перейти к разработке и изготовлению серверной части.

Во второй главе моего исследования была разработана схема студенческого конструкторского бюро. Эта схема предусматривает наличие нескольких компьютеров, которые будут подключены к центральному серверу. Были установлены и настроены важные составляющие инфраструктуры на сервере для работы сотрудников: LDAP, GitLab и iRedMile. Эти программы являются неотъемлемой частью функциональности бюро и обеспечивают его эффективную работу.

В третьей главе было продемонстрировано тестирование данных программ, в процессе которого ошибок и нарушений не выявлено.

Таким образом, все задачи, поставленные в первом этапе ВКР решены, что позволяет сделать вывод о достижении цели работы.

Предполагаемой научной новизной работы является предложенная архитектура инфраструктуры студенческого конструкторского бюро, базирующейся на выбранном программном обеспечении для сервера, сервисов

для аутентификации, хранения кода и системы контроля версий, а также для почтового сервера. Предложенная архитектура позволит организовать рабочий процесс участников студенческого конструкторского бюро, позволяя им эффективно работать над проектами, получая практический опыт в разработке программного обеспечения и решении сложных задач.

## **СПИСОК СОКРАЩЕНИЙ**

ПО – программное обеспечение.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. RB.RU :\_сайт. – URL: <https://rb.ru/opinion/otdely/> (дата обращения 15.03.2023)
2. Iee :\_сайт. – URL: <http://www.iee.unn.ru/wpcontent/uploads/sites/9/2014/09/Книга-slajdy> (дата обращения 15.03.2023)
3. Microsoft :\_\_сайт. – URL: <https://www.microsoft.com/ru-ru> (дата обращения 15.02.2023)
4. Linux.org :\_сайт. – URL: <https://www.linux.org/> (дата обращения 15.02.2023)
5. Ubuntu :\_сайт. – URL: <https://ubuntu.com/> (дата обращения 15.02.2023)
6. LDAP Lightweight Directory Access Protocol :\_\_сайт. – URL: <https://ldap.com/> (дата обращения 13.09.2023)
7. Fortinet : сайт. – URL: <https://www.fortinet.com/resources/cyberglossary/kerberos-authentication#> (дата обращения 13.09.2023)
8. OAuth :\_сайт. – URL: <https://oauth.net/2/> (дата обращения 13.09.2023)
9. OpenID :\_сайт. – URL: <https://openid.net/developers/how-connect-works/> (дата обращения 14.09.2023)
10. SAML :\_сайт. – URL: <https://saml.xml.org/> (дата обращения 14.09.2023)
11. CodeX :\_сайт. – URL: <https://codex.so/jwt> (дата обращения 14.09.2023)
12. GitHub :\_сайт. – URL: <https://github.com/> (дата обращения 07.10.2023)
13. Bitbucket :\_сайт. – URL: <https://bitbucket.org/product/ru/guides/getting-started/overview#a-brief-overview-of-bitbucket> (дата обращения 17.10.2023)
14. GitLab :\_\_сайт. – URL: <https://about.gitlab.com/> (дата обращения 17.10.2023)
15. Weril :\_сайт. – URL: <https://weril.me/mail/> (дата обращения 17.10.2023)
16. RUSH ANALYTICS. – URL: <https://www.rush-analytics.ru/blog/chto-takoe-mx-zapis-domena?clckid=0e78884f#:~> (дата обращения 21.10.2023)
17. Хабр :\_сайт. – URL: <https://qna.habr.com/q/466518?clckid=fd16a58c#:~> (дата обращения 05.11.2023)

18. Mail-in-a-Box : сайт. – URL: <https://mailinabox.email/> (дата обращения 05.11.2023)
19. Zimbra : сайт. – URL: <https://www.zimbra.com/> (дата обращения 05.11.2023)
20. Microsoft : сайт. – URL: <https://www.microsoft.com/ru-ru/microsoft-365/exchange/email> (дата обращения 05.11.2023)
21. iRedMail : сайт. – URL: <https://www.iredmail.org/> (дата обращения 18.11.2023)
22. VirtualBox : сайт. – URL: <https://www.virtualbox.org/> (дата обращения 10.09.2023)
23. LDAPSft : сайт . – URL: <https://www.ldapsft.com/ldapbrowserfree.html> (дата обращения 23.01.2024)



Министерство науки и высшего образования РФ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

О.В. Непомнящий

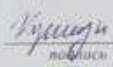



«19» 06 2024 г.

### МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Инфраструктура студенческого конструкторского бюро.  
Развертывание сервера хранения документации

090401 Информатика и вычислительная техника

09.04.01.11 «Вычислительные системы и сети»

Руководитель	 подпись	19.06.24 дата	доцент, канд. физ.-мат. наук должность, ученая степень	К.В. Коршун
Выпускник	 подпись	19.06.24 дата		А.А. Погосян
Рецензент	 подпись	19.06.24 дата	доцент должность, ученая степень	П.В. Кочкин
Нормоконтролер	 подпись	19.06.24 дата	доцент, канд. физ.-мат. наук должность, ученая степень	К.В. Коршун

Красноярск 2024