

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего
образования
«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа 2

Группа: Р34102

Выполнил:

Лапин А.А.

Юнусов Р.Э.

Проверил:

к.т.н. преподаватель Белозубов А.В.

Санкт-Петербург

2024г.

ОГЛАВЛЕНИЕ

Введение	5
Цели и задачи	5
Цели:	5
Задачи:	5
Настройка виртуального облака на гостевой ОС - WS_ubuntu.....	7
Текст задания.....	7
Настройка NextCloud с помощью веб-сервера nginx и MySQL	8
Шаг 1: переключить сетевое подключение на всех виртуальных машинах – Сетевой мост.....	8
Шаг 2: определить IP-адрес гостевой ОС	8
Шаг 3: Установка веб-сервера nginx	9
Шаг 4: Установка MySQL	10
Шаг 5: Установка необходимых PHP модулей	12
Шаг 6: Настройка конфигурационных файлов MySQL	14
Шаг 7. Настройка конфигурационных файлов PHP	15
Шаг 8: Создание базы данных и пользователя.....	17
Шаг 9: Установка Nextcloud	18
Шаг 10: Настройка PHP-FPM.....	19
Шаг 11: Генерация самоподписанных SSL-сертификатов.....	21
Шаг 12: Настройка Nginx	22
Шаг 13: Завершение установки Nextcloud	30
Шаг 14: Оптимизация и устранение предупреждений Nextcloud	33
Шаг 15: Добавление пользователей.....	38
Шаг 16: Проверка работы	39
Результаты выполнения задания	41
Установка и настройка системы документооборота на гостевой ОС	43
Текст задания.....	43

Установка и настройка Moodle	43
Шаг 1: Установка необходимых пакетов	43
Шаг 2: Скачивание и установка Moodle.....	44
Шаг 3: Настройка прав доступа	46
Шаг 4: Конфигурация MySQL	46
Шаг 5: Создание базы данных Moodle	47
Шаг 6. Конфигурация PHP	49
Шаг 7. Генерация самоподписанных SSL-сертификатов.....	50
Шаг 8. Установка Moodle	53
Шаг 9. Настройка Moodle	57
Шаг 10. Увеличим максимальный размер загружаемого файла.....	59
Шаг 11. Добавление пользователей и создание курса.....	61
Результаты выполнения задания	63
Установить и настроить пакет phpvirtualbox на гостевой ОС - WS_ubuntu	65
Текст задания.....	65
Выполнение задания.....	65
Шаг 1. Установка необходимых зависимостей.....	65
Шаг 2. Установка phpvirtualbox	66
Шаг 3. Настройка Nginx под phpvirtualbox	66
Шаг 4. Настройка php.....	68
Шаг 5. Настройка phpvirtualbox	69
Шаг 6. Настройка веб-службы VirtualBox	69
Шаг 7. Доступ к phpVirtualBox	70
Шаг 8. Исправление ошибки "Method 'ns1:IMachine_getAudioAdapter' not implemented"	71
Шаг 9. Исправление ошибки "Choose a Virtual optical Disk File Not Load files..."	72
Итог.....	73
Результаты выполнения задания	76

Установка приложения Webmin на гостевой ОС - WS_ubuntu.....	78
Текст задания.....	78
Выполнение задания.....	78
Шаг 1. Обновление системы	78
Шаг 2: Установка необходимых зависимостей	78
Шаг 3: Добавление репозитория Webmin	78
Шаг 4: Установка Webmin.....	78
Шаг 5: Доступ к Webmin	79
Шаг 6: Вход в Webmin	79
Шаг 7: Изменение системного времени	79
Шаг 8: Настройка языка интерфейса.....	80
Шаг 9: Добавление нового пользователя	81
Шаг 10: Изучение функционала Webmin.....	81
Результаты выполнения задания	82
Установка сервера видео конференций TrueConf Server Free на гостевой ОС - WS_ubuntu	83
Текст задания.....	83
Выполнение задания.....	83
Шаг 1. Установка из репозитория.....	83
Шаг 2. Добавление пользователя для доступа к панели управления.....	83
Шаг 3. Проверка установки	84
Шаг 4. Вход в панель управления.....	84
Шаг 5. Добавление пользователей.....	85
Шаг 6. Проведение конференции	87
Результаты выполнения задания	88
Заключение.....	89
Литература.....	91

Введение

В современном мире информационных технологий эффективное управление данными и коммуникациями становится критически важным для любой организации. Данная работа посвящена установке и настройке различных серверных приложений на базе операционной системы Ubuntu, которые решают широкий спектр задач по организации корпоративной инфраструктуры.

Цели и задачи

Цели:

1. Получение практических навыков установки и настройки облачных систем хранения данных
2. Освоение методов развертывания систем электронного документооборота
3. Изучение инструментов удаленного управления виртуальными машинами
4. Приобретение опыта работы с системами администрирования через веб-интерфейс
5. Научиться разворачивать системы видеоконференцсвязи

Задачи:

1. Настройка виртуального облака NextCloud:
 - a. Установка и конфигурация веб-сервера, базы данных и PHP
 - b. Развертывание системы облачного хранения
 - c. Настройка безопасности и оптимизация производительности
 - d. Тестирование доступа с различных устройств
2. Установка системы документооборота:
 - a. Развертывание выбранной системы (Moodle)
 - b. Создание пользователей и назначение ролей
 - c. Настройка проекта и сроков выполнения
 - d. Тестирование совместной работы
3. Настройка веб-интерфейса управления виртуализацией:
 - a. Установка и настройка phpvirtualbox
 - b. Интеграция с VirtualBox
 - c. Тестирование создания и управления виртуальными машинами
4. Установка системы удаленного администрирования:
 - a. Развертывание Webmin

- b. Настройка локализации и системного времени
 - c. Управление пользователями
 - d. Изучение функционала системы
5. Настройка сервера видеоконференций:
 - a. Установка TrueConf Server Free
 - b. Создание пользователей
 - c. Тестирование видеоконференций

Настройка виртуального облака на гостевой ОС - WS_ubuntu

Текст задания

1. Переключить сетевое подключение на всех виртуальных машинах – Сетевой мост
2. Определить IP-адрес гостевой ОС
3. Установить и настроить веб сервер Apache или nginx
4. Установить и настроить MySQL или MariaDB
5. Установить необходимые PHP модули
6. Установить и настроить пакет NextCloud-XXXX.tar.bz2 с сайта <https://nextcloud.com>.
7. Добавить пользователей (себя и Белозубов А.В.) в систему
8. Проверить соединение
9. Установить на WS_win Desktop приложение NextCloud и проверить подключение.
10. По возможности проверить подключение через мобильное приложение Nextcloud

Настройка NextCloud с помощью веб-сервера nginx и MySQL

Шаг 1: переключить сетевое подключение на всех виртуальных машинах – Сетевой мост

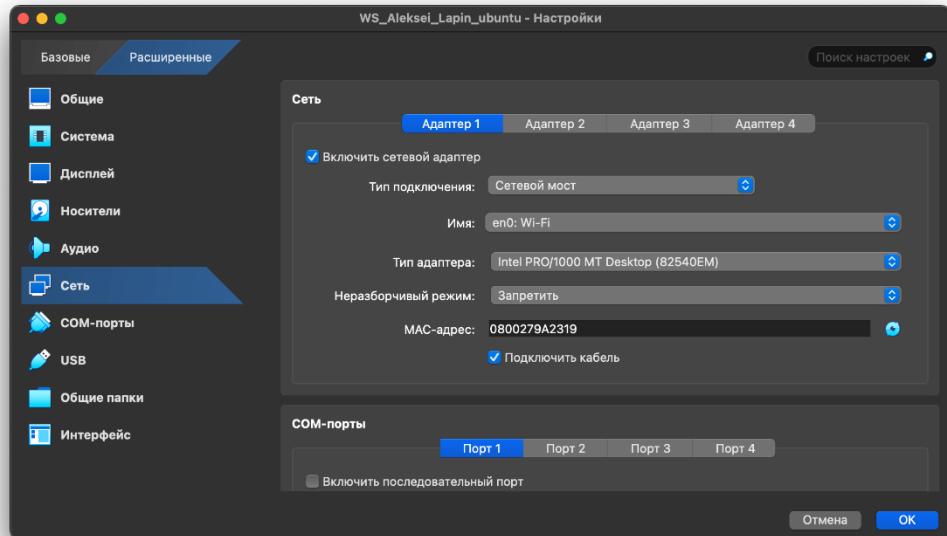


Рисунок 1. Настройка сетевого моста в VirtualBox для виртуальной машины Ubuntu

Шаг 2: определить IP-адрес гостевой ОС

A screenshot of a macOS terminal window titled 'aleksei@alekseis-MacBook-Pro:~'. The user has run the command 'networksetup -getinfo Wi-Fi'. The output shows the following details:

```
Last login: Fri Nov 15 16:14:42 on ttys002
networksetup -getinfo Wi-Fi
DHCP Configuration
IP address: 192.168.0.196
Subnet mask: 255.255.255.0
Router: 192.168.0.1
Client ID:
IPv6: Off
Wi-Fi ID: f0:18:98:21:51:f4
```

The terminal window has a dark theme and shows the current date and time at the top right.

Рисунок 2. Вывод команды `networksetup -getinfo Wi-Fi`, показывающий IP-адрес и

сетевые настройки гостевой ОС.

Шаг 3: Установка веб-сервера nginx

Для установки веб-сервера nginx были выполнены следующие команды:

1. Установка nginx:

```
sudo apt update  
sudo apt install nginx
```

2. Запуск и включение автозапуска nginx:

```
sudo systemctl start nginx # Запуск веб-сервера  
sudo systemctl enable nginx # Включение автозапуска
```

3. Проверка статуса nginx:

```
sudo systemctl status nginx
```

4. Проверка работоспособности через curl:

```
curl -I localhost # Проверка ответа веб-сервера
```

Результаты выполнения команд показали, что:

- Nginx успешно установлен и запущен
- Сервер работает как системный сервис
- Все рабочие процессы (workers) запущены
- Сервер отвечает кодом 200 OK на запросы
- Версия установленного nginx: 1.24.0

```

WS_Aleksei_Lapin_ubuntu (before-lab2) [Работает]
alex@WS-Lapin-ubuntu:~$ sudo systemctl start nginx
alex@WS-Lapin-ubuntu:~$ sudo systemctl enable nginx
alex@WS-Lapin-ubuntu:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-11-15 13:16:35 UTC; 1min 18s ago
     Docs: man:nginx(8)
     Main PID: 1389 (nginx)
        Tasks: 5 (limit: 4614)
       Memory: 5.0M (peak: 5.2M)
          CPU: 44ms
         CGroup: /system.slice/nginx.service
                 ├─1389 nginx: master process /usr/sbin/nginx -g daemon on; master process
                 ├─1391 "nginx: worker process"
                 ├─1392 "nginx: worker process"
                 ├─1394 "nginx: worker process"
                 └─1395 "nginx: worker process"

Nov 15 13:16:34 WS-Lapin-ubuntu systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Nov 15 13:16:35 WS-Lapin-ubuntu systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
alex@WS-Lapin-ubuntu:~$ curl -I localhost
HTTP/1.1 200 OK
Server: nginx/1.24.0 (Ubuntu)
Date: Fri, 15 Nov 2024 13:18:10 GMT
Content-Type: text/html
Content-Length: 615
Last-Modified: Fri, 15 Nov 2024 12:33:58 GMT
Connection: keep-alive
ETag: "67373fae-267"
Accept-Ranges: bytes

alex@WS-Lapin-ubuntu:~$ 

```

Рисунок 3. Результаты установки и настройки веб-сервера nginx, включая вывод команд проверки статуса и тестового запроса

Шаг 4: Установка MySQL

Для установки и настройки MySQL были выполнены следующие команды:

- Установка MySQL сервера:

```

sudo apt update
sudo apt install mysql-server

```

- Запуск и включение автозапуска MySQL:

```

sudo systemctl start mysql
sudo systemctl enable mysql

```

```

alex@WS-Lapin-ubuntu:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
   Active: inactive (dead)

```

Рисунок 4. Проверка статуса MySQL сервера

- Настройка безопасности MySQL с помощью скрипта

mysql_secure_installation:

```
alex@WS-Lapin-ubuntu: $ sudo mysql_secure_installation

Securing the MySQL server deployment.

Enter password for user root:
Error: Can't connect to local MySQL server through socket '/var/run/mysqld/mysqld.sock' (2)
alex@WS-Lapin-ubuntu: $ sudo systemctl start mysql
alex@WS-Lapin-ubuntu: $ sudo systemctl enable mysql
Synchronizing state of mysql.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable mysql
alex@WS-Lapin-ubuntu: $ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y

There are three levels of password validation policy:

LOW    Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary
          file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0

Skipping password set for root as authentication with auth_socket is used by default
.
If you would like to use password authentication instead, this can be done with the
"ALTER_USER" command.
See https://dev.mysql.com/doc/refman/8.0/en/alter-user.html#alter-user-password-management for more information.

By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.

Normally, root should only be allowed to connect from
```

Рисунок 5. Процесс настройки безопасности MySQL сервера с помощью скрипта
mysql_secure_installation начальный экран

```
Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) :n^

... skipping.
Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
alex@WS-Lapin-ubuntu:~$
```

Рисунок 6. Процесс настройки безопасности MySQL сервера с помощью скрипта

mysql_secure_installation конечный экран

В процессе настройки безопасности были выполнены следующие действия:

- Включена валидация паролей с низким уровнем сложности (длина ≥ 8 символов)
- Удалены анонимные пользователи
- Отключена возможность удаленного подключения для пользователя root
- Перезагружены таблицы привилегий для применения изменений

Результаты выполнения команд показали, что:

- MySQL сервер успешно установлен и запущен
- Настроена базовая безопасность
- Сервис работает в автоматическом режиме
- Версия установленного MySQL: 8.0

Шаг 5: Установка необходимых PHP модулей

Для установки и настройки необходимых PHP модулей были выполнены следующие команды:

1. Установим необходимые пакеты:

```
sudo apt install -y php php-curl php-gd php-json php-mbstring php-zip php-mysql php-bz2  
php-intl php-ldap php-bcmath php-gmp php-imagick libreoffice php-smbclient php-imap  
sudo apt install ffmpeg
```

- php-curl: Поддержка cURL для HTTP-запросов
- php-gd: Поддержка графических операций
- php-json: Поддержка JSON-данных
- php-mbstring: Поддержка многобайтовых строк
- php-zip: Поддержка ZIP-архивов
- php-mysql: Поддержка MySQL
- php-bz2: Поддержка BZIP2
- php-intl: Поддержка международных форматов
- php-ldap: Поддержка LDAP
- php-bcmath: Поддержка математических операций с большими числами
- php-gmp: Поддержка математических операций с большими числами
- php-imagick: Поддержка графических операций с использованием ImageMagick
- libreoffice: Поддержка LibreOffice для работы с документами
- php-smbclient: Поддержка SMB-клиента для работы с файлами на серверах SMB
- php-imap: Поддержка IMAP для работы с почтовыми серверами
- ffmpeg: Поддержка видео и аудио-конвертации

Эти пакеты обеспечивают полную функциональность Nextcloud, включая:

- Обработку различных типов файлов
- Безопасное хранение и передачу данных
- Интеграцию с внешними сервисами
- Поддержку многоязычности
- Работу с медиафайлами и документами

2. Убедимся, что PHP установлен и работает:

```
php -v
```

```
alex@WS-Lapin-ubuntu:~$ php -v
PHP 8.3.6 (cli) (built: Sep 30 2024 15:17:17) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.6, Copyright (c) Zend Technologies
    with Zend OPcache v8.3.6, Copyright (c), by Zend Technologies
alex@WS-Lapin-ubuntu:~$
```

Рисунок 7. Вывод команды `php -v`, показывающий версию PHP.

Шаг 6: Настройка конфигурационных файлов MySQL

1. Внесем изменения в конфигурационный файл MySQL:

Листинг 1. Конфигурационного файла `/etc/mysql/my.cnf`

```
[server]
skip_name_resolve = 1      # Отключает DNS-резолвинг для повышения производительности
innodb_buffer_pool_size = 128M # Размер буфера InnoDB для кэширования данных и индексов
innodb_buffer_pool_instances = 1      # Устанавливает количество экземпляров буферного пула;
innodb_flush_log_at_trx_commit = 2
innodb_log_buffer_size = 32M # Размер буфера журнала транзакций
innodb_max_dirty_pages_pct = 90 # Максимальный процент "грязных" страниц до принудительной записи
tmp_table_size = 64M # Максимальный размер временных таблиц в памяти
max_heap_table_size = 64M # Максимальный размер MEMORY таблиц
slow_query_log = 1 # Включение логирования медленных запросов для анализа и оптимизации запросов Nextcloud
slow_query_log_file = /var/log/mysql/slow.log # Путь к файлу лога медленных запросов
long_query_time = 1 # Порог времени (в секундах) для определения медленных запросов

[client-server]
!includedir /etc/mysql/conf.d/ # Подключение дополнительных конфигурационных файлов
!includedir /etc/mysql/mysql.conf.d/ # Подключение стандартных конфигурационных файлов

[client]
default-character-set = utf8mb4 # Кодировка по умолчанию для клиентских подключений,
                                обеспечивает поддержку Unicode
```

```
[mysqld]
character_set_server = utf8mb4    # Кодировка сервера для поддержки Unicode
collation_server = utf8mb4_general_ci    # Правила сортировки для UTF-8
transaction_isolation = READ-COMMITTED # Уровень изоляции транзакций для безопасной работы
binlog_format = ROW    # Формат бинарного лога для репликации
innodb_file_per_table = 1    # Хранение каждой таблицы в отдельном файле
```

Эти настройки оптимизируют MySQL для работы с Nextcloud, обеспечивая:

- **Повышенную производительность:** Увеличение размеров буферов и отключение DNS-рэзолвинга сокращают задержки и ускоряют обработку запросов, что особенно важно при большом количестве пользователей и операций в Nextcloud.
 - **Надежность и устойчивость:** Баланс между производительностью и надежностью при записи транзакций обеспечивает стабильную работу базы данных, уменьшая риск потери данных.
 - **Эффективное использование ресурсов:** Оптимизация памяти и буферов позволяет MySQL эффективно работать на сервере с ограниченными ресурсами, что важно для средних и малых установок Nextcloud.
 - **Поддержка многоязычности и правильной кодировки:** Использование `utf8mb4` гарантирует корректное хранение и отображение символов различных языков, что необходимо для пользователей Nextcloud из разных регионов.
 - **Мониторинг и оптимизация:** Включение логирования медленных запросов помогает выявлять и оптимизировать самые ресурсоемкие операции, улучшая общую производительность системы.
 - **Безопасная работа с транзакциями:** Уровень изоляции транзакций READ-COMMITTED предотвращает возможные проблемы при одновременном доступе нескольких пользователей, что часто происходит в среде Nextcloud.
- Эти настройки обеспечивают стабильную, быструю и безопасную работу Nextcloud, позволяя эффективно управлять большими объемами данных и многочисленными пользователями.

Шаг 7. Настройка конфигурационных файлов PHP

1. Внесем изменения в конфигурационный файл PHP:

```

extension pdo_mysql.so # Подключает расширение PDO для взаимодействия PHP с MySQL

[mysql]
mysql.allow_local_infile=On # Разрешает использование локальных файлов для загрузки данных
mysql.allow_persistent=On # Позволяет использовать постоянные соединения с базой данных
mysql.cache_size=2000 # Размер кэша для оптимизации запросов к базе данных
mysql.max_persistent=-1 # Не ограничивает количество постоянных соединений
mysql.max_links=-1 # Не ограничивает количество одновременных соединений
mysql.default_port=3306 # Устанавливает стандартный порт для подключения к MySQL
mysql.default_socket=/var/run/mysql/mysqld.sock# Указывает путь к сокету MySQL
mysql.default_host=localhost # Устанавливает хост MySQL на локальный
mysql.default_user=
mysql.default_password=
mysql.connect_timeout=60 # Устанавливает тайм-аут соединения
mysql.trace_mode=off # Отключает режим трассировки для повышения безопасности и производительности Nextcloud

```

Эти настройки обеспечивают эффективное соединение PHP с MySQL, улучшая производительность и стабильность работы Nextcloud.

Вот описание некоторых из настроек:

- **extension=pdo_mysql.so:** Позволяет PHP взаимодействовать с MySQL через PDO, что необходимо для работы Nextcloud.
- **mysql.allow_local_infile=On:** Позволяет загрузку локальных файлов, что может быть необходимо для импорта данных в Nextcloud.
- **mysql.allow_persistent=On:** Использование постоянных соединений снижает накладные расходы на установку новых соединений для каждого запроса, что улучшает производительность Nextcloud.
- **mysql.cache_size=2000:** Размер кэша используется для быстрого доступа к часто запрашиваемым данным, ускоряя работу Nextcloud.
- **mysql.max_persistent=-1:** Отсутствие ограничения на количество постоянных соединений позволяет Nextcloud эффективно масштабироваться при высокой нагрузке.
- **mysql.max_links=-1:** Отсутствие ограничения на количество соединений обеспечивает гибкость при одновременной работе множества пользователей Nextcloud.
- **mysql.default_port=3306:** Устанавливает стандартный порт для подключения к MySQL, упрощая конфигурацию Nextcloud.

Эти настройки оптимизируют взаимодействие между PHP и MySQL, обеспечивая

стабильную и эффективную работу Nextcloud. Они позволяют базе данных работать быстрее, безопаснее и надежнее, что особенно важно для обеспечения высокой производительности и масштабируемости системы при большом количестве пользователей и операций.

Шаг 8: Создание базы данных и пользователя

1. Войдем в MySQL под root:

```
mysql -uroot -p # Подключение к MySQL серверу:  
# -u указывает имя пользователя (root)  
# -p запрашивает пароль при подключении
```

2. Создадим базу данных и пользователя для Nextcloud:

```
CREATE USER 'nextclouduser'@'localhost' IDENTIFIED BY 'password';  
CREATE DATABASE IF NOT EXISTS nextcloud CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;  
GRANT ALL PRIVILEGES on nextcloud.* to 'nextclouduser'@'localhost';  
FLUSH PRIVILEGES;
```

Пояснения:

- **CREATE USER 'nextclouduser'@'localhost' IDENTIFIED BY 'password';**

Создает нового пользователя `nextclouduser` с паролем `password`, который сможет подключаться к MySQL только с локального хоста (`localhost`). Это обеспечивает безопасность, ограничивая доступ только с сервера, на котором установлен MySQL.

- **CREATE DATABASE IF NOT EXISTS nextcloud CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;**

Создает базу данных `nextcloud`, если она еще не существует. Устанавливает кодировку `utf8mb4` для поддержки широкого спектра символов Unicode и `utf8mb4_general_ci` для общего сопоставления символов, что важно для многоязычных систем.

- **GRANT ALL PRIVILEGES on nextcloud.* to 'nextclouduser'@'localhost';**

Предоставляет пользователю `nextclouduser` полные привилегии на все таблицы и объекты внутри базы данных `nextcloud`. Это необходимо для полноценной работы Nextcloud, позволяя пользователю выполнять любые необходимые операции с базой данных.

- **FLUSH PRIVILEGES;**

Применяет все изменения привилегий, сделанные выше, без необходимости перезапуска сервера MySQL. Это гарантирует, что новые настройки доступа вступят в силу немедленно.

```
mysql> CREATE USER 'nextclouduser'@'localhost' IDENTIFIED BY 'alex333221';
Query OK, 0 rows affected (0.19 sec)

mysql> CREATE DATABASE nextcloud;
Query OK, 1 row affected (0.04 sec)

mysql> CREATE DATABASE IF NOT EXISTS nextcloud CHARACTER SET utf8mb4 COLLATE
-> utf8mb4_general_ci;
Query OK, 1 row affected, 1 warning (0.01 sec)

mysql> GRANT ALL PRIVILEGES on nextcloud.* to 'nextclouduser'@'localhost';
Query OK, 0 rows affected (0.02 sec)
```

Рисунок 8. Создание базы данных и пользователя для Nextcloud

Шаг 9: Установка Nextcloud

- Скачаем последнюю версию Nextcloud:

```
wget https://download.nextcloud.com/server/releases/latest.zip
```

- Распакуем архив в директорию веб-сервера:

```
sudo unzip latest.zip -d /var/www/
```

- Установим права доступа для веб-сервера:

```
sudo chown -R www-data:www-data /var/www/nextcloud
sudo chmod -R 755 /var/www/nextcloud
```

Пояснения:

- `sudo chown -R www-data:www-data /var/www/nextcloud`

Эта команда изменяет владельца и группу всех файлов и каталогов внутри директории `/var/www/nextcloud` на `www-data`.

Зачем:

- `www-data` — это стандартный пользователь и группа, под которыми работает веб-сервер (например, Nginx или Apache).
- Обеспечивает веб-серверу необходимые права для чтения, записи и выполнения файлов Nextcloud, что критично для корректной работы приложения.

- `sudo chmod -R 755 /var/www/nextcloud`

Эта команда устанавливает права доступа для всех файлов и каталогов внутри директории `/var/www/nextcloud` на `755`.

Зачем:

- `755` означает, что владелец (`www-data`) имеет права на чтение, запись и выполнение, а группа и остальные пользователи имеют права на чтение и выполнение только.
- Обеспечивает безопасность, позволяя веб-серверу выполнять необходимые скрипты и читать файлы, одновременно предотвращая несанкционированный доступ к критическим файлам системы.

Совместное использование этих команд обеспечивает безопасную и функциональную среду для работы Nextcloud, позволяя веб-серверу выполнять необходимые задачи без риска компрометации системы.

Шаг 10: Настройка PHP-FPM

1. Установка необходимых пакетов:

```
sudo apt install -y php-fpm php-ctype php-curl php-gd php-iconv php-json php-mbstring php-xml php-zip
```

Пояснения к установленным пакетам:

- **php-fpm**: PHP FastCGI Process Manager, необходимый для обработки PHP-скриптов. Позволяет PHP работать как отдельный процесс, улучшая производительность и управляемость.
- **php-ctype**: Расширение PHP для работы с символьными классами. Используется Nextcloud для обработки текстовых данных и валидации ввода.
- **php-curl**: Расширение PHP для выполнения HTTP-запросов. Необходимо для интеграции Nextcloud с внешними сервисами и API.
- **php-gd**: Расширение PHP для обработки графических изображений. Используется Nextcloud для управления аватарами пользователей и другими графическими элементами.
- **php-iconv**: Расширение PHP для преобразования символов между различными кодировками. Обеспечивает корректную обработку многоязычных данных в Nextcloud.
- **php-json**: Расширение PHP для работы с JSON-данными. Необходимо для обработки конфигурационных файлов и взаимодействия с API Nextcloud.
- **php-mbstring**: Расширение PHP для работы с многобайтовыми строками. Обеспечивает поддержку Unicode и многоязычности в Nextcloud.
- **php-xml**: Расширение PHP для обработки XML-данных. Используется Nextcloud для обработки данных конфигурации и взаимодействия с

некоторыми сервисами.

- **php-zip:** Расширение PHP для работы с ZIP-архивами. Необходимо для загрузки и распаковки файлов в Nextcloud.

2. Конфигурация PHP-FPM:

```
sudo vim /etc/php/8.3/fpm/php.ini
```

Листинг 2. Конфигурационный файл /etc/php/8.3/fpm/php.ini

```
memory_limit = 512M
upload_max_filesize = 1024M
post_max_size = 1024M
max_execution_time = 360
```

Пояснения:

- **`memory_limit = 512M`**

Зачем: устанавливает максимальное количество памяти, которое скрипты PHP могут использовать. Для Nextcloud рекомендуется увеличивать этот лимит, чтобы обеспечить достаточный объем памяти для обработки больших файлов и сложных операций. Это помогает предотвратить ошибки памяти при выполнении ресурсоемких задач.

- **`upload_max_filesize = 1024M`**

Зачем: определяет максимальный размер файла, который может быть загружен через PHP. Установка этого значения на 1024М позволяет пользователям загружать файлы до 1 ГБ, что важно для облачного хранилища Nextcloud, где пользователи могут работать с крупными файлами.

- **`post_max_size = 1024M`**

Зачем: определяет максимальный размер данных, которые могут быть отправлены методом POST. Должен быть не меньше, чем `upload_max_filesize`, чтобы поддерживать загрузку больших файлов. Это гарантирует, что запросы с большими данными не будут отклонены сервером.

- **`max_execution_time = 360`**

Зачем: устанавливает максимальное время выполнения скриптов PHP в секундах. Увеличение этого значения позволяет скриптам Nextcloud выполнять дольше, что полезно при выполнении длительных операций, таких как обработка больших файлов, индексация данных или выполнение сложных вычислений.

Настройка этих параметров в PHP-FPM обеспечивает стабильную и эффективную работу Nextcloud, позволяя приложению обрабатывать большие объемы данных без сбоев и

задержек.

3. Перезапусти сервис PHP-FPM

```
sudo systemctl restart php8.3-fpm
```

Шаг 11: Генерация самоподписанных SSL-сертификатов

```
sudo mkdir -p /etc/ssl/nginx
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
    -keyout /etc/ssl/nginx/nextcloud.key \
    -out /etc/ssl/nginx/nextcloud.crt
```

Пояснения:

- Зачем нужны SSL-сертификаты

Безопасность данных: Шифрование всего трафика между клиентом и сервером, защищая передаваемые файлы и личную информацию от перехвата

- Команды и их назначение:

- sudo mkdir -p /etc/ssl/nginx : создает директорию для хранения SSL-сертификатов
- openssl req -x509: создает самоподписанный сертификат X.509
- -nodes: пропускает шифрование закрытого ключа (для автоматической загрузки)
- -days 365: устанавливает срок действия сертификата на 1 год
- -newkey rsa:2048: создает новый RSA-ключ длиной 2048 бит
- -keyout: указывает путь для сохранения закрытого ключа
- -out: указывает путь для сохранения сертификата

Примечание: В производственной среде рекомендуется использовать сертификаты от доверенных центров сертификации, но для тестовой среды самоподписанный сертификат является приемлемым решением.

```

alex@WS-Lapin-ubuntu:~$ sudo systemctl restart php8.3-fpm
alex@WS-Lapin-ubuntu:~$ sudo mkdir -p /etc/ssl/nginx
alex@WS-Lapin-ubuntu:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
    -keyout /etc/ssl/nginx/nextcloud.key \
    -out /etc/ssl/nginx/nextcloud.crt
...
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:RU
State or Province Name (full name) [Some-State]:Saint-Petersburg
Locality Name (eg, city) []:Saint-Petersburg
Organization Name (eg, company) [Internet Widgits Pty Ltd]:mycompany
Organizational Unit Name (eg, section) []:unitname
Common Name (e.g. server FQDN or YOUR name) []:MC
Email Address []:

```

Рисунок 9. Генерация самоподписанного SSL-сертификата

Шаг 12: Настройка Nginx

- Узнаем IP-адрес виртуальной машины:

```
ip addr show
```

```

alex@WS-Lapin-ubuntu:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
            inet6 ::1/128 scope host noprefixroute
                valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:9a:23:19 brd ff:ff:ff:ff:ff:ff
        inet 192.168.0.137/24 brd 192.168.0.255 scope global dynamic noprefixroute enp0s3
            valid_lft 5872sec preferred_lft 5872sec
            inet6 fe80::a00:27ff:fe9a:2319/64 scope link
                valid_lft forever preferred_lft forever

```

Рисунок 10. Узнаем IP-адрес виртуальной машины

- Создание конфигурационного файла для Nginx для Nextcloud

```
sudo vim /etc/nginx/sites-available/nextcloud
```

2.1. Определение upstream для PHP-FPM nginx

```

upstream php-handler {
    # Используйте UNIX-сокет для PHP-FPM, если он настроен
    server unix:/run/php/php8.3-fpm.sock;
    # или используйте IP и порт, если предпочитаете TCP-соединение
    # server 127.0.0.1:9000;
}

```

Пояснения:

- Назначение: определяет обработчик PHP-запросов. Использование UNIX-сокета обеспечивает более быструю и эффективную коммуникацию между

Nginx и PHP-FPM по сравнению с TCP-соединением.

2.2. Кэширование неизменяемых ресурсов

```
# Set the `immutable` cache control options only for assets with a cache busting `v` argument
map $arg_v $asset_immutable {
    "" "";
    default ", immutable";
}
```

Пояснения:

- Назначение: позволяет устанавливать заголовок `Cache-Control` с параметром `immutable` для ресурсов с параметром версии (`v`).
- Почему для Nextcloud: улучшает кэширование статических файлов (CSS, JS, и т.д.), снижая задержки и увеличивая скорость загрузки страниц.

2.3. Перенаправление HTTP на HTTPS

```
server {
    listen 80;
    listen [::]:80;
    server_name 192.168.0.137; # Замените на IP вашей виртуальной машины

    # Prevent nginx HTTP Server Detection
    server_tokens off;

    # Enforce HTTPS
    return 301 https://$server_name$request_uri;
}
```

Пояснения:

- Назначение: перенаправляет все HTTP-запросы на HTTPS.
- Почему для Nextcloud: обеспечивает защищенное соединение, предотвращая перехват данных и соответствующая требованиям безопасности Nextcloud.

2.4. Основной блок сервера HTTPS

Листинг 3. Основной блок сервера HTTPS

```
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    # With NGinx >= 1.25.1 you should use this instead:
    # listen 443      ssl;
    # listen [::]:443 ssl;
    # http2 on;
    server_name 192.168.0.137; # IP виртуальной машины
```

```

# Path to the root of your installation
root /var/www/nextcloud;

# Use Mozilla's guidelines for SSL/TLS settings
# https://mozilla.github.io/server-side-tls/ssl-config-generator/
ssl_certificate /etc/ssl/nginx/nextcloud.crt;
ssl_certificate_key /etc/ssl/nginx/nextcloud.key;

# Prevent nginx HTTP Server Detection
server_tokens off;

# HSTS settings
# WARNING: Only add the preload option once you read about
# the consequences in https://hstspreload.org/. This option
# will add the domain to a hardcoded list that is shipped
# in all major browsers and getting removed from this list
# could take several months.
#add_header Strict-Transport-Security "max-age=15768000; includeSubDomains; preload"
always;

# set max upload size and increase upload timeout:
client_max_body_size 512M;
client_body_timeout 300s;
fastcgi_buffers 64 4K;

# Enable gzip but do not remove ETag headers
gzip on;
gzip_vary on;
gzip_comp_level 4;
gzip_min_length 256;
gzip_proxied expired no-cache no-store private no_last_modified no_etag auth;
gzip_types application/atom+xml text/javascript application/javascript
application/json application/ld+json application/manifest+json application/rss+xml
application/vnd.geo+json application/vnd.ms-fontobject application/wasm application/x-
font-ttf application/x-web-app-manifest+json application/xhtml+xml application/xml
font/opentype image/bmp image/svg+xml image/x-icon text/cache-manifest text/css text/plain
text/vcard text/vnd.rim.location.xloc text/vtt text/x-component text/x-cross-domain-
policy;

# Pagespeed is not supported by Nextcloud, so if your server is built
# with the `ngx_pagespeed` module, uncomment this line to disable it.
#pagespeed off;

# The settings allows you to optimize the HTTP2 bandwidth.
# See https://blog.cloudflare.com/delivering-http-2-upload-speed-improvements/
# for tuning hints
client_body_buffer_size 512k;

# HTTP response headers borrowed from Nextcloud ` .htaccess `
add_header Referrer-Policy "no-referrer" always;
add_header X-Content-Type-Options "nosniff" always;
add_header X-Frame-Options "SAMEORIGIN" always;

```

```

add_header X-Permitted-Cross-Domain-Policies "none" always;
add_header X-Robots-Tag "noindex,nofollow" always;
add_header X-XSS-Protection "1; mode=block" always;
add_header Strict-Transport-Security "max-age=15552000; includeSubDomains" always;
# Remove X-Powered-By, which is an information leak
fastcgi_hide_header X-Powered-By;

# Set .mjs and .wasm MIME types
# Either include it in the default mime.types list
# and include that list explicitly or add the file extension
# only for Nextcloud like below:
include mime.types;
types {
    text/javascript mjs;
    application/wasm wasm;
}

# Specify how to handle directories -- specifying `/index.php$request_uri`
# here as the fallback means that Nginx always exhibits the desired behaviour
# when a client requests a path that corresponds to a directory that exists
# on the server. In particular, if that directory contains an index.php file,
# that file is correctly served; if it doesn't, then the request is passed to
# the front-end controller. This consistent behaviour means that we don't need
# to specify custom rules for certain paths (e.g. images and other assets,
# `/updater`, `/ocs-provider`), and thus
# `try_files $uri $uri/ /index.php$request_uri`
# always provides the desired behaviour.
index index.php index.html /index.php$request_uri;

# Rule borrowed from `.htaccess` to handle Microsoft DAV clients
location = / {
    if ( $http_user_agent ~ ^DavClnt ) {
        return 302 /remote.php/webdav/$is_args$args;
    }
}

location = /robots.txt {
    allow all;
    log_not_found off;
    access_log off;
}

# Make a regex exception for `/.well-known` so that clients can still
# access it despite the existence of the regex rule
# `location ~ /(.|autotest|...)` which would otherwise handle requests
# for `/.well-known`.
location ^~ /.well-known {
    # The rules in this block are an adaptation of the rules
    # in `.htaccess` that concern `/.well-known`.

    location = /.well-known/carddav { return 301 /remote.php/dav/; }
    location = /.well-known/caldav { return 301 /remote.php/dav/; }
}

```

```

location /.well-known/acme-challenge { try_files $uri $uri/ =404; }
location /.well-known/pki-validation { try_files $uri $uri/ =404; }

# Let Nextcloud's API for `/.well-known` URIs handle all other
# requests by passing them to the front-end controller.
return 301 /index.php$request_uri;
}

# Rules borrowed from `/.htaccess` to hide certain paths from clients
location ~ ^/(?:build|tests|config|lib|3rdparty|templates|data)(?:$|/) { return 404;
}
location ~ ^/(?:\.|autotest|occ|issue|indie|db_|console) { return 404;
}

# Ensure this block, which passes PHP files to the PHP process, is above the blocks
# which handle static assets (as seen below). If this block is not declared first,
# then Nginx will encounter an infinite rewriting loop when it prepends `/index.php`
# to the URI, resulting in a HTTP 500 error response.
location ~ \.php(?:$|/) {
    # Required for legacy support
    rewrite
        ^/(!index|remote|public|cron|core|ajax|update|status|ocs|v[12]|updater|.+|ocs-
provider|/.+|.+|richdocumentscode(_arm64)?|proxy) /index.php$request_uri;

    fastcgi_split_path_info ^(.+?\.\php)(/.*)$;
    set $path_info $fastcgi_path_info;

    try_files $fastcgi_script_name =404;

    include fastcgi_params;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    fastcgi_param PATH_INFO $path_info;
    fastcgi_param HTTPS on;

    fastcgi_param modHeadersAvailable true;           # Avoid sending the security
headers twice
    fastcgi_param front_controller_active true;       # Enable pretty urls
    fastcgi_pass php-handler;

    fastcgi_intercept_errors on;
    fastcgi_request_buffering off;

    fastcgi_max_temp_file_size 0;
}

# Serve static files
location ~ \.(?:css|js|mjs|svg|gif|ico|jpg|png|webp|wasm|tflite|map|ogg|flac)$ {
    try_files $uri /index.php$request_uri;
    # HTTP response headers borrowed from Nextcloud `/.htaccess`
    add_header Cache-Control "public, max-
age=15778463$asset_immutable";
}

```

```

    add_header Referrer-Policy           "no-referrer"      always;
    add_header X-Content-Type-Options    "nosniff"         always;
    add_header X-Frame-Options          "SAMEORIGIN"       always;
    add_header X-Permitted-Cross-Domain-Policies "none"        always;
    add_header X-Robots-Tag             "noindex, nofollow" always;
    add_header X-XSS-Protection        "1; mode=block"     always;
    access_log off;      # Optional: Don't log access to assets
}

location ~ \.(otf|woff2?)$ {
    try_files $uri /index.php$request_uri;
    expires 7d;          # Cache-Control policy borrowed from `htaccess`
    access_log off;      # Optional: Don't log access to assets
}

# Rule borrowed from `htaccess`
location /remote {
    return 301 /remote.php$request_uri;
}

location / {
    try_files $uri $uri/ /index.php$request_uri;
}
}

```

Пояснения:

1. Настройка SSL/TLS:

- a. `ssl_certificate & ssl_certificate_key`: Указывают на пути к SSL-сертификату и закрытому ключу. Обеспечивают шифрование данных между клиентом и сервером.
- b. `server_tokens off`: Скрывает информацию о версии Nginx, повышая безопасность.
- c. Strict-Transport-Security (HSTS): Настройка заголовка для принудительного использования HTTPS. Рекомендуется для повышения безопасности.

2. Параметры загрузки и таймауты:

- a. `client_max_body_size 512M`: Устанавливает максимальный размер загружаемого файла. Требуется для поддержки больших файлов в Nextcloud.
- b. `client_body_timeout 300s`: Увеличивает время ожидания загрузки тела запроса, предотвращая разрывы соединения при медленной загрузке.

3. Кэширование и сжатие:

- a. `gzip` настройки: Включение сжатия данных для уменьшения объема передаваемых данных и ускорения загрузки страниц.

- b. Cache-Control: Управляет кэшированием статических ресурсов, улучшая производительность.

4. Заголовки безопасности:

- a. Referrer-Policy, X-Content-Type-Options, X-Frame-Options, X-Permitted-Cross-Domain-Policies, X-Robots-Tag, X-XSS-Protection: Защищают сайт от различных типов атак, таких как XSS, Clickjacking и других.
- b. fastcgi_hide_header X-Powered-By: Скрывает информацию о сервере, повышая безопасность.

5. Обработка URI и маршрутизация:

- a. index index.php index.html /index.php\$request_uri: Определяет порядок обработки файлов по умолчанию. В случае отсутствия статических файлов перенаправляет запросы на `index.php`.
- b. location ~ \.php(?:\$|/): Обработка PHP-файлов через PHP-FPM. Важно для корректной работы Nextcloud, который базируется на PHP.

6. Исключения и перенаправления:

- a. location ^~ /.well-known: Обеспечивает корректную обработку специальных URI, необходимых для Let's Encrypt и других сервисов.
- b. location ~ ^/(?:build|tests|config|lib|3rdparty|templates|data)(?:\$|/): Скрывает конфиденциальные директории от доступа извне, повышая безопасность.
- c. location /remote: Специальное перенаправление для DAV клиентов, обеспечивающее корректную работу клиентских приложений Nextcloud.

7. Обслуживание статических файлов:

- a. location ~ \.(?:css|js|mjs|svg|gif|ico|jpg|png|webp|wasm|tflite|map|ogg|flac)\$: Оптимизирует обслуживание статических ресурсов, устанавливая соответствующие заголовки и отключая логирование для улучшения производительности.
- b. location ~ \.(otf|woff2?): Настройка кэширования для шрифтов, обеспечивая быструю загрузку и снижение нагрузки на сервер.

8. Безопасность взаимодействия:

- a. Добавление `no-referrer`, `nosniff`, `SAMEORIGIN`, и других заголовков: Гарантирует, что браузеры обрабатывают контент безопасно, предотвращая утечку информации и атаки.

9. Оптимизация HTTP/2:

- a. Использование HTTP/2 с поддержкой SSL повышает производительность за

счет мультиплексирования запросов и улучшенного управления соединениями.

10. Дополнительные параметры:

- a. fastcgi_buffers 64 4K: Оптимизирует буферизацию FastCGI для улучшения производительности при обработке PHP-запросов.
- b. fastcgi_request_buffering off: Отключает буферизацию запросов FastCGI, что может повысить производительность при обработке больших файлов.

Итог:

Данная конфигурация Nginx для Nextcloud нацелена на обеспечение высокой безопасности, производительности и стабильности работы приложения. Она включает в себя оптимизации для обработки PHP-запросов, кэширования статических ресурсов, обеспечения защищенного соединения через HTTPS и настройки заголовков безопасности. Следование этим настройкам позволяет Nextcloud эффективно обслуживать пользователей, обеспечивая быстрый доступ к данным и защиту от потенциальных угроз.

3. Создадим символическую ссылку для конфигурационного файла:

```
sudo ln -s /etc/nginx/sites-available/nextcloud /etc/nginx/sites-enabled/
```

Пояснения:

- Назначение: Создание символьской ссылки между директориями `sites-available` и `sites-enabled` является стандартной практикой в конфигурации Nginx.
- Как это работает:
 - o sites-available/ содержит все доступные конфигурационные файлы виртуальных хостов
 - o sites-enabled/ содержит только активные конфигурации
 - o Nginx читает только конфигурации из директории sites-enabled/
- Преимущества такого подхода:
 - o Легко включать/отключать сайты без удаления конфигурации
 - o Лучшая организация и управление несколькими виртуальными хостами

4. Проверим конфигурацию Nginx:

```
sudo nginx -t
```

```

alex@WS-Lapin-ubuntu:~$ sudo vim /etc/nginx/sites-available/nextcloud
alex@WS-Lapin-ubuntu:~$ sudo ln -s /etc/nginx/sites-available/nextcloud /etc/nginx/sites-enabled/
alex@WS-Lapin-ubuntu:~$ sudo nginx -t
2024/11/15 18:37:26 [warn] 27386#27386: duplicate extension "wasm", content type: "application/wasm", previous content type: "application/wasm" in /etc/nginx/sites-enabled/nextcloud:94
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
alex@WS-Lapin-ubuntu:~$ 

```

Рисунок 11. Проверка конфигурации Nginx

Пояснения:

- Назначение: Проверка синтаксиса конфигурационных файлов Nginx перед применением изменений
- Почему это важно:
 - Предотвращает запуск Nginx с некорректной конфигурацией
 - Помогает обнаружить ошибки в синтаксисе конфигурационных файлов
 - Показывает подробные сообщения об ошибках, если они есть

5. Перезапустим Nginx:

```
sudo systemctl restart nginx
```

Шаг 13: Завершение установки Nextcloud

1. Откроем браузер и перейдем по адресу Вирт. машины <http://192.168.0.137>
2. Следуем инструкциям на экране, чтобы завершить установку, указав данные root пользователя и базы данных (имя базы данных, имя пользователя и пароль)

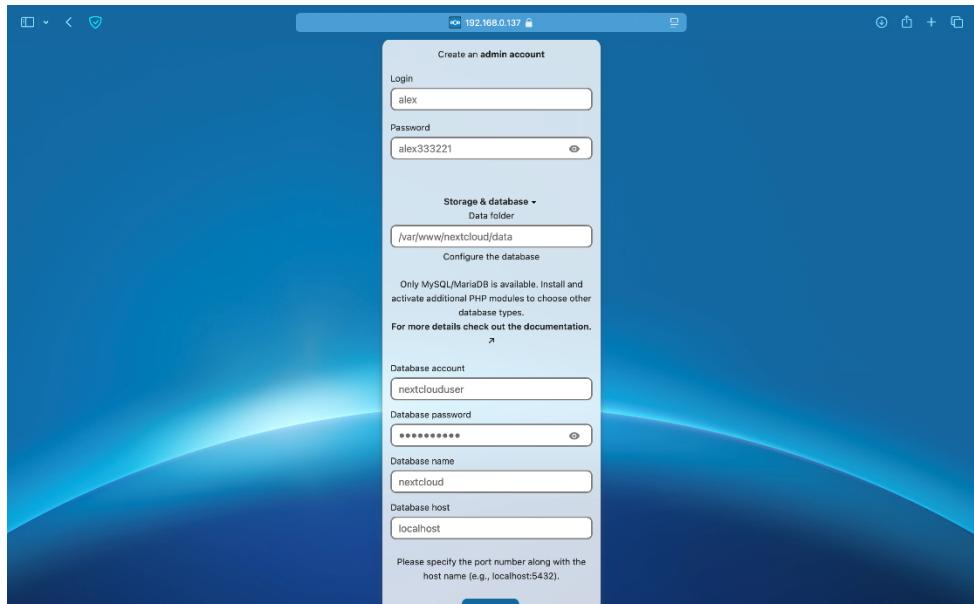


Рисунок 12. Настройка root пользователя и базы данных

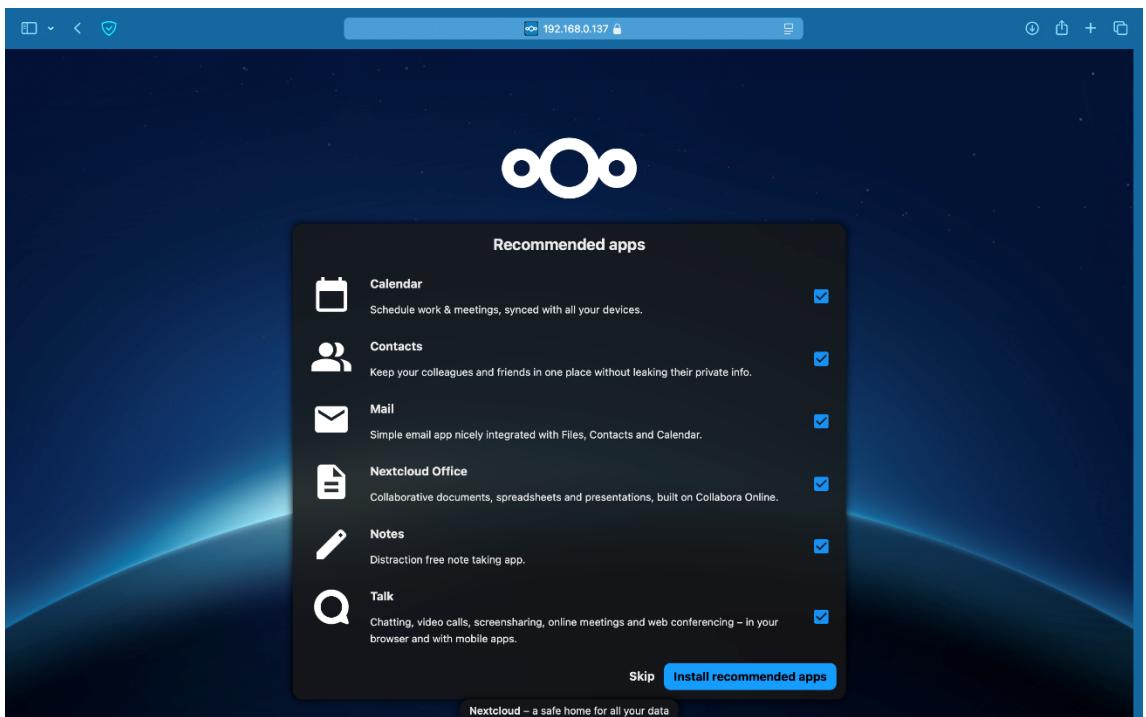


Рисунок 13. Начальный экран Nextcloud

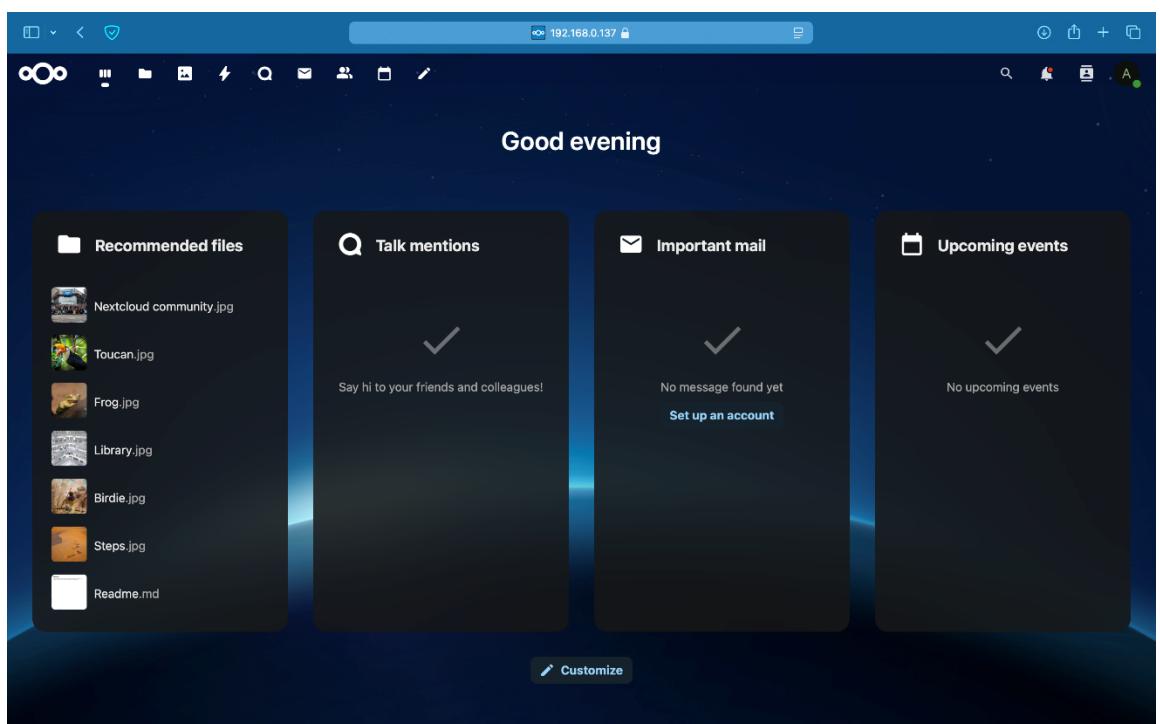


Рисунок 14. Дашборд Nextcloud

3. После завершения установки настроим cron для автоматического выполнения задач Nextcloud

```
sudo crontab -u www-data -e
*/15 * * * * php -f /var/www/nextcloud/cron.php
sudo -u www-data crontab -l
```

Пояснения по настройке cron:

- Назначение: Настройка регулярного выполнения фоновых задач Nextcloud
- Почему это важно:
 - Автоматическое обновление файловых кэшей
 - Проверка наличия обновлений системы
 - Отправка уведомлений пользователям
 - Очистка временных файлов и устаревших данных
 - Синхронизация внешних хранилищ
 - Создание превью для медиафайлов
- Как работает:
 - `*/15 * * * *` - запуск каждые 15 минут
 - `-u www-data` - выполнение от имени пользователя веб-сервера
 - `cron.php` - скрипт, выполняющий все запланированные задачи Nextcloud
- Преимущества перед альтернативными методами:
 - По сравнению с AJAX (веб-интерфейс):
 - Не зависит от активности пользователей в веб-интерфейсе
 - Меньше нагрузка на веб-сервер
 - Гарантированное выполнение задач даже при отсутствии активных сессий
 - По сравнению с Webcron:
 - Не требует внешних сервисов или дополнительной настройки
 - Более надежное выполнение (не зависит от доступности внешних сервисов)
 - Лучший контроль над временем выполнения задач
 - Более безопасно (все операции выполняются локально)

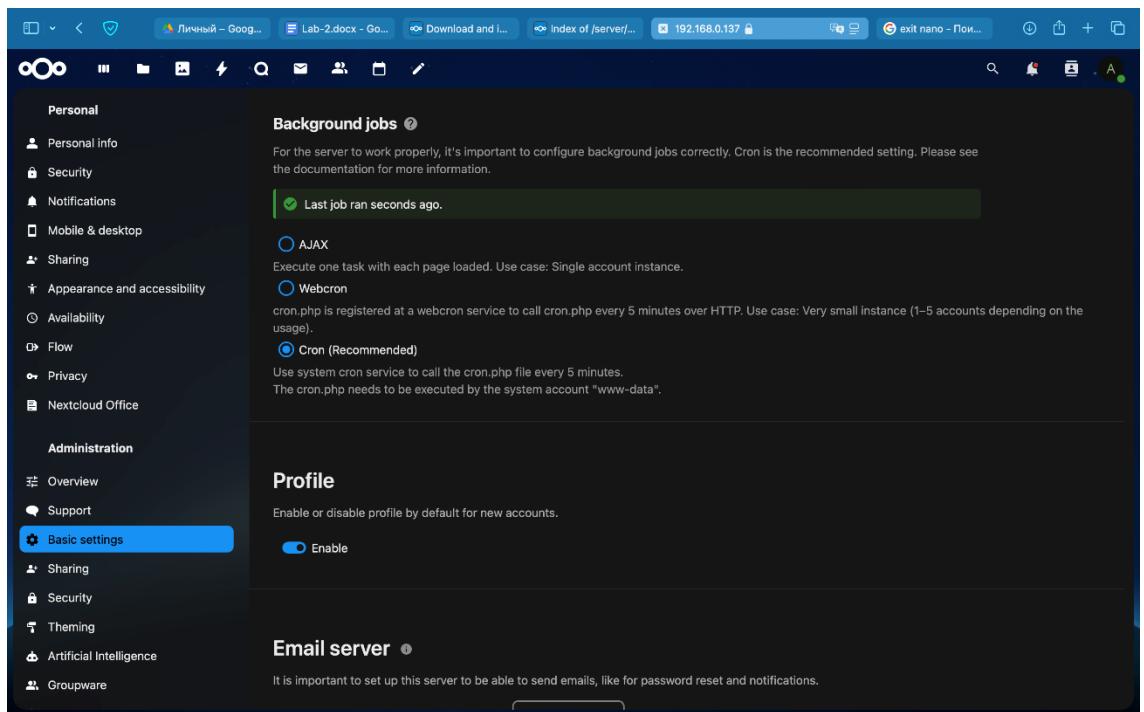


Рисунок 15. Настройка Background Jobs для Nextcloud

Шаг 14: Оптимизация и устранение предупреждений Nextcloud

Перед выполнением команд осс необходимо перейти в директорию установки Nextcloud:

```
cd /var/www/nextcloud
```

Пояснения:

- Назначение команды `cd`:
 - Переход в директорию установки Nextcloud необходим для корректного выполнения команд осс
 - OCC (OwnCloud Console) — это инструмент командной строки для администрирования Nextcloud

1. Настройка окна обслуживания:

```
sudo -u www-data php occ config:system:set maintenance_window_start --value="1"
```

Пояснения:

- Назначение: Установка времени начала окна обслуживания на 1 час ночи.
- Почему это важно:
 - Позволяет выполнять тяжелые задачи обслуживания в период минимальной активности

- Снижает влияние на производительность системы в рабочее время
- Автоматизирует выполнение задач обслуживания в оптимальное время

2. Полное обслуживание и восстановление системы:

```
sudo -u www-data php occ maintenance:repair --include-expensive
```

Пояснения:

- Назначение: Выполнение полной проверки и исправления системы, включая ресурсоемкие операции
- Что делает:
 - Обновляет и исправляет MIME-типы файлов
 - Проверяет целостность базы данных
 - Восстанавливает отсутствующие индексы
 - Исправляет несоответствия в файловой системе
- MIME-типы (Multipurpose Internet Mail Extensions) - это стандарт, определяющий формат файлов в интернете

3. Добавление индексов БД:

```
sudo -u www-data php occ db:add-missing-indices
```

Пояснения:

- Назначение: Создание отсутствующих индексов в базе данных Nextcloud
- Почему это важно:
 - Значительно ускоряет поиск и сортировку данных
 - Улучшает производительность запросов к базе данных
 - Оптимизирует работу с большими объемами данных
 - Снижает нагрузку на сервер при выполнении сложных запросов

4. Настройка PHP OPcache.

Отредактируем файл php.ini:

```
sudo vim /etc/php/8.3/fpm/php.ini
```

Листинг 4. Конфигурационный файл /etc/php/8.3/fpm/php.ini

```
opcache.interned_strings_buffer=16
```

Пояснения:

- Что такое OPcache:
 - Это система кэширования байт-кода PHP
 - Хранит скомпилированный код PHP в памяти
- Зачем нужно увеличение буфера:
 - Nextcloud использует много строковых операций
 - Значение 16 оптимально для кэширования строк Nextcloud
 - Предотвращает переполнение буфера при интенсивной работе

5. Установка и настройка memcache.

Установим Redis:

```
sudo apt install redis-server
```

Добавим в config.php Nextcloud:

```
'memcache.local' => '\OC\Memcache\Redis',
'memcache.locking' => '\OC\Memcache\Redis',
'redis' => [
    'host' => 'localhost',
    'port' => 6379,
],
```

Пояснения:

- Что такое Redis:
 - Высокопроизводительное хранилище данных в памяти
 - Работает как распределенный кэш и система блокировок
- Зачем нужен для Nextcloud:
 - Улучшение производительности:
 - Кэширование часто запрашиваемых данных
 - Снижение нагрузки на базу данных
 - Ускорение доступа к файлам и метаданным
 - Управление блокировками:
 - Предотвращение конфликтов при одновременном доступе
 - Обеспечение целостности данных при параллельной работе
 - Координация доступа между разными серверами
 - Масштабируемость:
 - Поддержка распределенных систем
 - Эффективная работа с большим количеством пользователей
 - Оптимизация использования системных ресурсов

- Настройки в config.php:
 - `memcache.local` - использование Redis для локального кэширования
 - `memcache.locking` - использование Redis для управления блокировками
 - Параметры подключения настроены на локальный сервер Redis

6. Настройка региона телефонных номеров.

Добавим в config.php:

```
'default_phone_region' => 'RU',
```

Пояснения:

- Назначение: Установка региона по умолчанию для форматирования телефонных номеров
- Почему это важно:
 - Корректное отображение и валидация телефонных номеров
 - Правильное форматирование номеров при отображении контактов
 - Поддержка локальных форматов номеров для России

7. Установка поддержки SVG для imagick:

```
sudo apt install imagemagick
sudo apt install php-imagick
```

Пояснения:

- Назначение: Добавление поддержки векторной графики SVG через ImageMagick
- Почему это важно:
 - Корректное отображение SVG-изображений в интерфейсе Nextcloud
 - Необходимо для корректной работы иконок и логотипов в интерфейсе

8. Настройка HSTS в Nginx. Отредактируем конфигурацию:

```
sudo vim /etc/nginx/sites-available/nextcloud
```

Добавим в секцию server:

```
add_header Strict-Transport-Security "max-age=15552000; includeSubDomains" always;
```

Пояснения:

- Что такое HSTS (HTTP Strict Transport Security):

- Механизм безопасности, который принуждает браузеры использовать только HTTPS
- Предотвращает атаки с понижением протокола и перехватом данных
- Параметры настройки:
 - `max-age=15552000` - срок действия политики HSTS (180 дней)
 - `includeSubDomains` - применяет политику ко всем поддоменам
 - `always` - применяет заголовок ко всем ответам сервера
- Почему это важно для Nextcloud:
 - Усиливает безопасность передачи данных
 - Защищает от атак типа man-in-the-middle
 - Соответствует современным требованиям безопасности

9. Настройка переменных окружения PHP.

Отредактируем php-fpm конфигурацию:

```
sudo vim /etc/php/8.3/fpm/pool.d/www.conf
```

Добавим или раскомментируем:

```
env[PATH] = /usr/local/bin:/usr/bin:/bin
clear_env = no
```

Пояснения:

- Настройка доступа PHP-FPM к системным переменным окружения
- Определение путей для выполнения внешних команд

10. Перезапустим необходимые сервисы:

```
sudo systemctl restart php8.3-fpm
sudo systemctl restart nginx
sudo systemctl restart redis
```

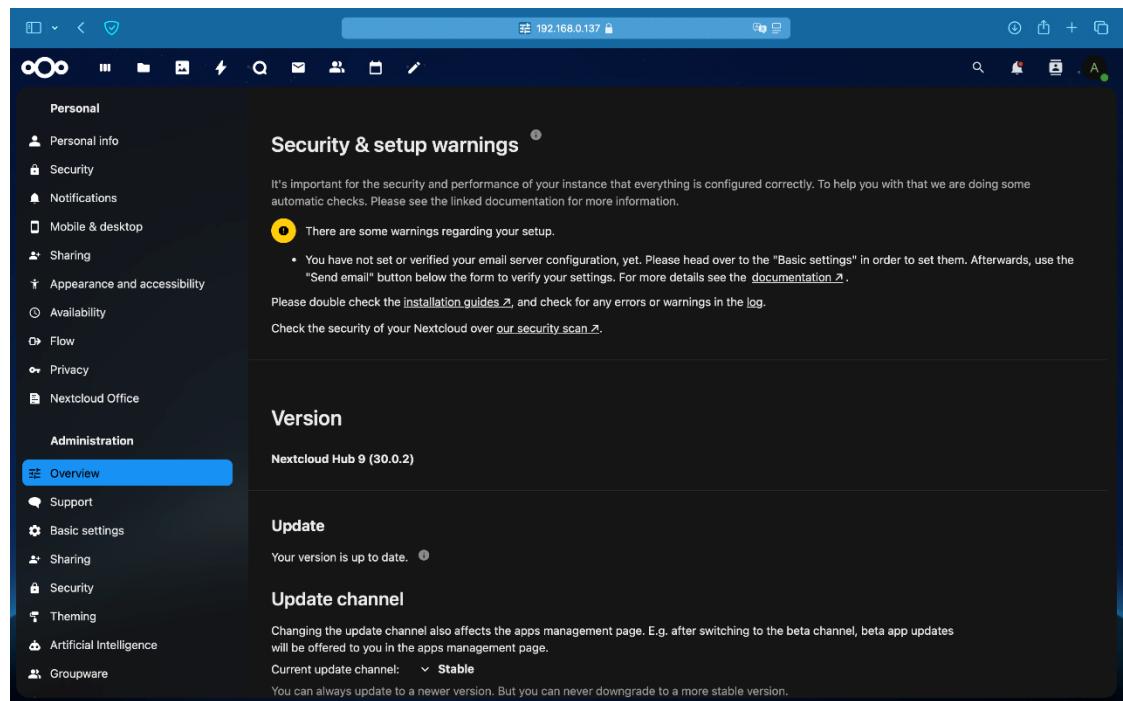


Рисунок 16. Security and setup warmings

Осталось только подключить почтовый сервер к Nextcloud. Но для этого нужен домен.

Шаг 15: Добавление пользователей

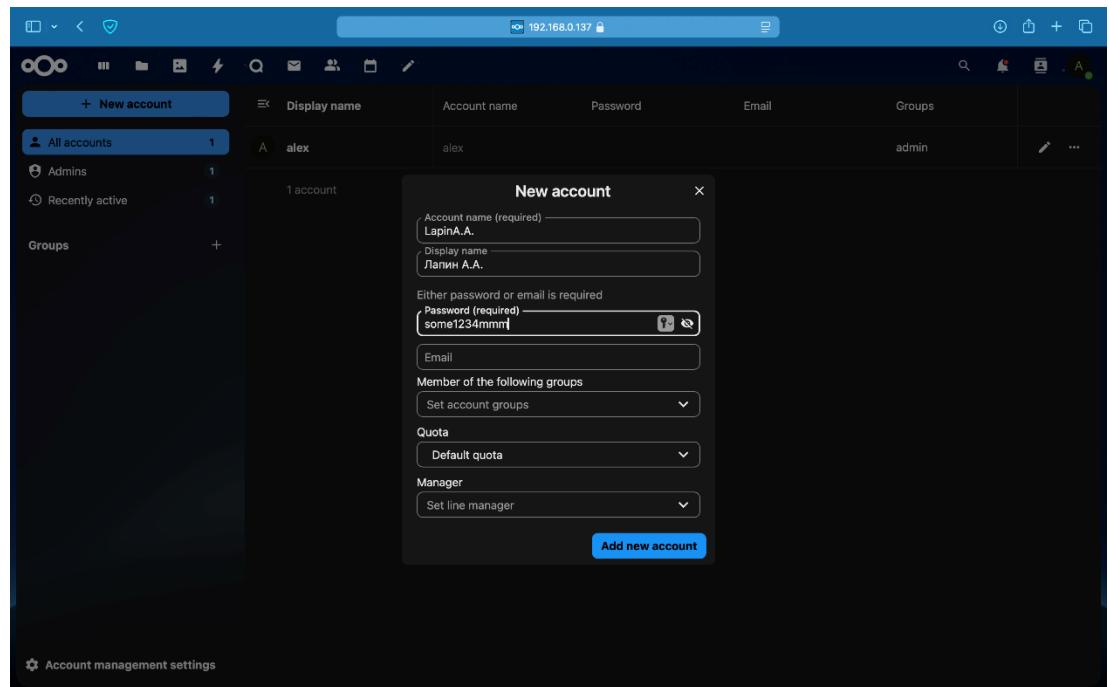


Рисунок 17. Добавление пользователя Лапина А.А. в систему

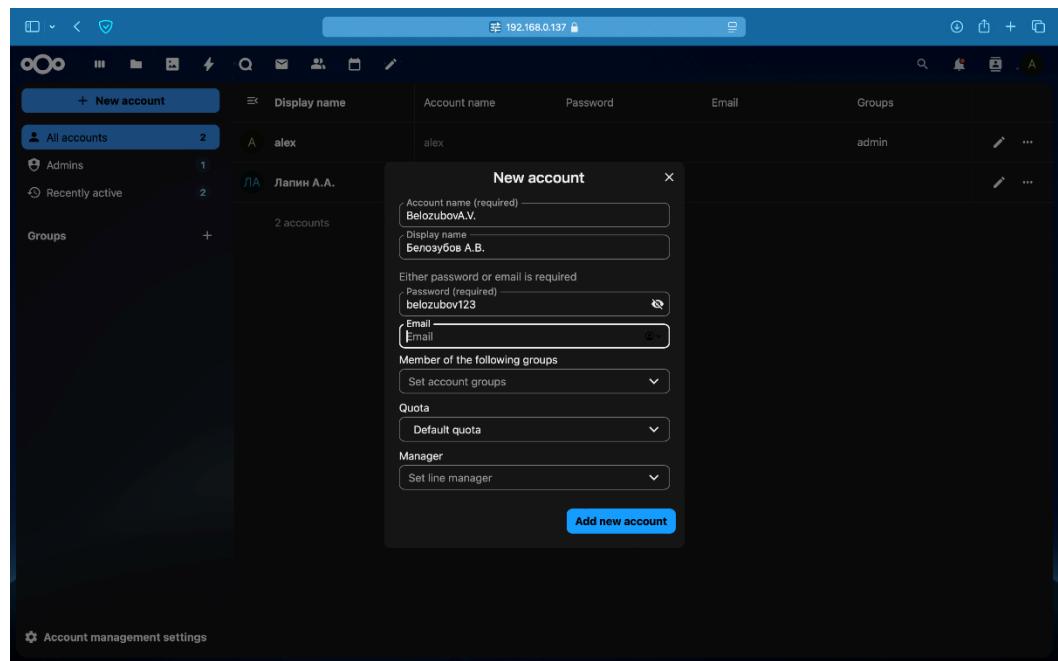


Рисунок 18. Добавление пользователя Белозубова А.В. в систему

Шаг 16: Проверка работы

Через Desktop приложение Nextcloud на WS_win:

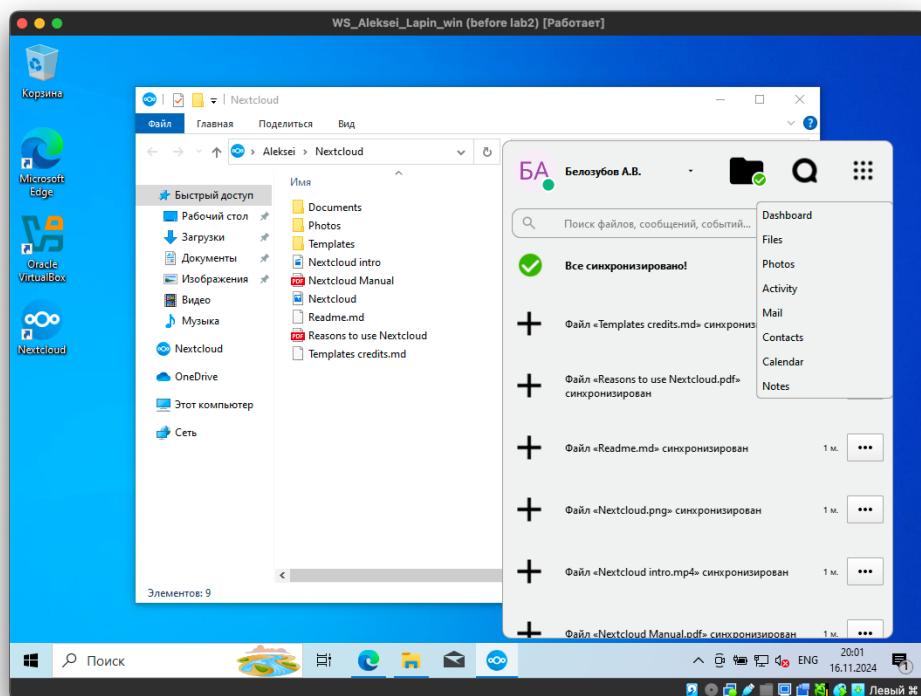


Рисунок 19. Проверка работы через Desktop приложение Nextcloud на WS_win

Через Desktop приложение Nextcloud на хост-машине:

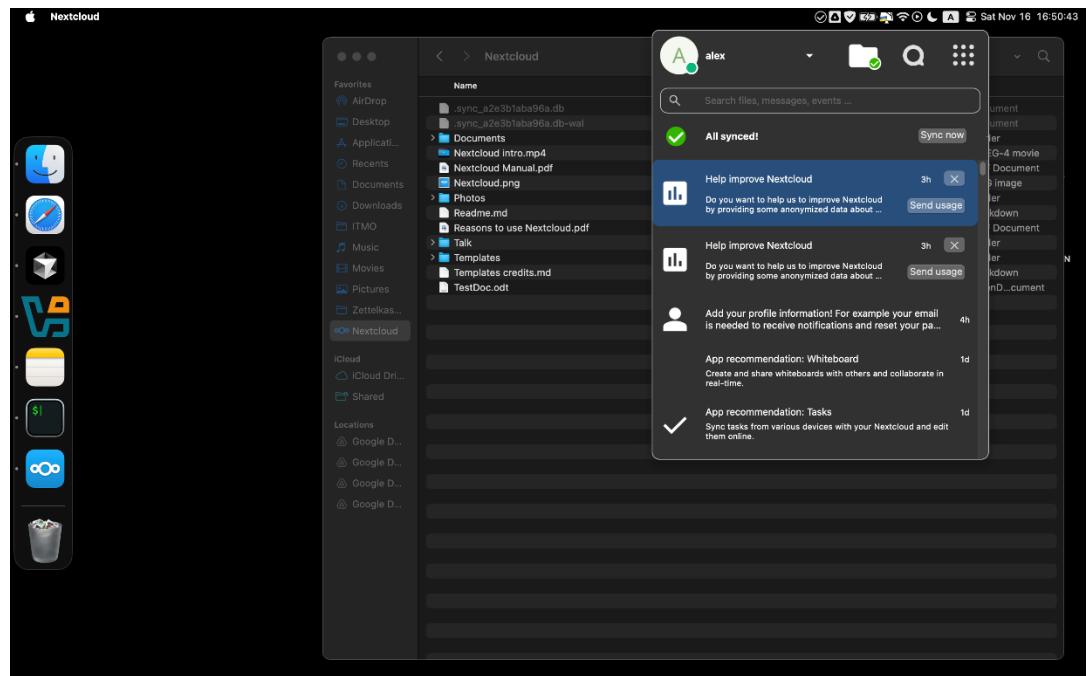


Рисунок 20. Проверка работы через Desktop приложение Nextcloud на хост-машине

Проверка работы мобильного приложения Nextcloud на Android

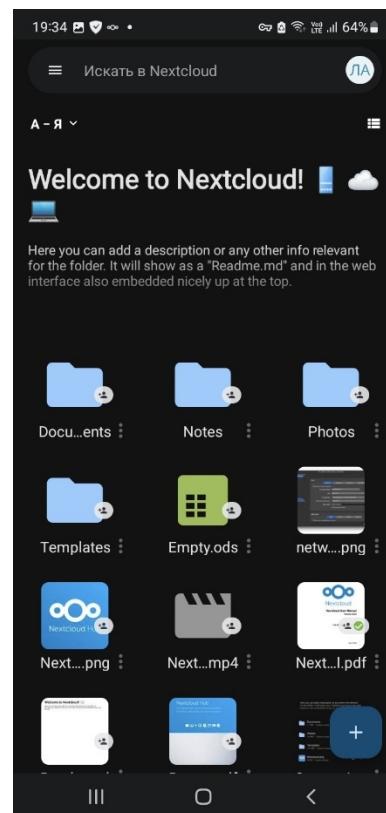


Рисунок 21. Приветственный экран мобильного приложения Nextcloud на Android

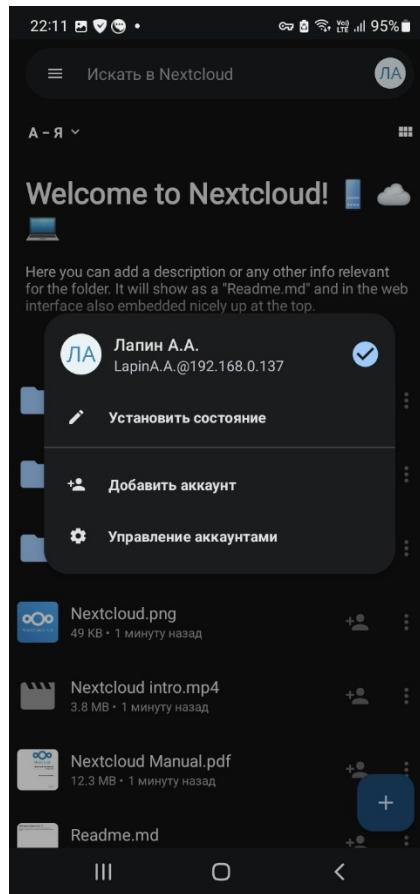


Рисунок 22. Аккаунт мобильного приложения Nextcloud на Android

Результаты выполнения задания

В результате выполнения задания было успешно развернуто частное облачное хранилище Nextcloud со следующими характеристиками:

1. Сетевая конфигурация:
 - a. Настроен сетевой мост для всех виртуальных машин
 - b. Обеспечена прямая доступность сервера Nextcloud из локальной сети
 - c. IP-адрес сервера: 192.168.0.137
2. Серверная инфраструктура:
 - a. Веб-сервер: Nginx с оптимизированной конфигурацией
 - b. База данных: MySQL 8.0 с настроенной безопасностью
 - c. PHP 8.3 с необходимыми модулями и оптимизацией производительности
 - d. Redis для кэширования и управления блокировками
3. Безопасность:
 - a. Настроено SSL-шифрование (самоподписанный сертификат)
 - b. Включен HSTS для принудительного HTTPS
 - c. Настроены заголовки безопасности в Nginx

- d. Реализована защита критических директорий
 - e. Отключено отображение версии Nginx
4. Оптимизация производительности:
- a. Настроен PHP OPcache
 - b. Реализовано кэширование через Redis
 - c. Оптимизированы настройки MySQL
 - d. Настроено сжатие данных (gzip)
 - e. Оптимизирована обработка статических файлов
5. Функциональность:
- a. Успешно добавлены пользователи (Лапин А.А. и Белозубов А.В.)
 - b. Настроены фоновые задачи через cron
 - c. Реализована поддержка SVG через ImageMagick
 - d. Настроена локализация
6. Проверка доступности:
- a. Веб-интерфейс доступен и работает корректно
 - b. Desktop-клиент успешно подключается с WS_win
 - c. Desktop-клиент успешно подключается с хост-машины
 - d. Мобильное приложение на Android успешно подключается и работает

Установка и настройка системы документооборота на гостевой ОС

Текст задания

Электронный документооборот на linux с использованием

- OPENCOURSE — пакета Group-Office <http://xdeb.ru>
- Redmine <http://www.redmine.org>
- Dolibarr ERP - CRM <https://sourceforge.net/projects/dolibarr/>
- Apache OFBiz <https://ofbiz.apache.org>
- ODOO (Trial) <https://www.odoo.com>
- Vtiger CRM 7.1.0 <https://www.vtiger.com>
- Jira <https://www.atlassian.com/ru/software/jira>
- Moodle <https://moodle.org>

1. Изучить функционал системы
2. Добавить двух пользователей в систему документооборота.
3. Назначить одного из пользователей руководителем проекта.
4. Создать совместный проект и установить сроку выполнения.
5. Подключить второго пользователя к проекту.

Установка и настройка Moodle

Шаг 1: Установка необходимых пакетов

```
# Установка инструмента для управления репозиториями
sudo apt-get install software-properties-common

# Добавление репозитория с актуальными версиями PHP
sudo add-apt-repository ppa:ondrej/php

# Установка основных компонентов:
# - apache2: веб-сервер для обработки HTTP-запросов
# - mysql-client и mysql-server: система управления базами данных для хранения данных Moodle
# - php8.3: интерпретатор PHP для выполнения кода Moodle
# - libapache2-mod-php8.3: модуль PHP для Apache
sudo apt install -y apache2 mysql-client mysql-server php8.3 libapache2-mod-php8.3
```

```
# Установка дополнительных компонентов и расширений PHP для Moodle:  
sudo apt install -y graphviz aspell ghostscript clamav php8.3-pspell php8.3-curl php8.3-gd  
php8.3-intl php8.3-mysql php8.3-xml php8.3-xmlrpc php8.3-ldap php8.3-zip php8.3-soap  
php8.3-mbstring git
```

Пояснения к установленным пакетам:

1. Утилиты для работы с контентом:
 - a. graphviz: Создание диаграмм и графических представлений данных
 - b. aspell: Проверка орфографии в текстовых материалах
 - c. ghostscript: Обработка PDF-файлов и изображений
 - d. clamav: Антивирусная защита загружаемых файлов
2. Расширения PHP для функциональности Moodle:
 - a. php8.3-pspell: Проверка орфографии в PHP
 - b. php8.3-curl: Выполнение HTTP-запросов (интеграции, обновления)
 - c. php8.3-gd: Обработка изображений (аватары, превью)
 - d. php8.3-intl: Поддержка интернационализации
 - e. php8.3-mysql: Работа с MySQL базами данных
 - f. php8.3-xml: Обработка XML-данных
 - g. php8.3-xmlrpc: Поддержка XML-RPC протокола
 - h. php8.3-ldap: Интеграция с LDAP-каталогами
 - i. php8.3-zip: Работа с ZIP-архивами
 - j. php8.3-soap: Поддержка SOAP протокола
 - k. php8.3-mbstring: Поддержка многобайтовых строк (Unicode)
3. Система контроля версий:
 - a. git: Клонирование Moodle из репозитория

Проверим версию PHP:

```
php -v
```

Перезапустим Apache:

```
sudo systemctl restart apache2
```

Шаг 2: Скачивание и установка Moodle

```
# Переходим в директорию /opt для установки Moodle  
cd /opt
```

```
# Клонируем официальный репозиторий Moodle
git clone git://git.moodle.org/moodle.git

# Переходим в директорию проекта
cd moodle

# Просматриваем доступные ветки репозитория
git branch -a

# Создаем локальную ветку, отслеживающую стабильную версию Moodle 4.0.5
sudo git branch --track MOODLE_405_STABLE origin/MOODLE_405_STABLE

# Переключаемся на стабильную версию
sudo git checkout MOODLE_405_STABLE
```

Пояснения к процессу установки:

1. Выбор директории `/opt`:
 - a. Стандартное место для установки дополнительного программного обеспечения
2. Использование Git для установки:
 - a. Получение кода напрямую из официального репозитория
 - b. Легкость установки из командной строки
 - c. Выбор из широкого спектра версий
3. Выбор стабильной версии (MOODLE_405_STABLE):
 - a. Использование последней стабильной версии
 - b. Совместимость с последними версиями PHP и MySQL

Рисунок 23. Выбор стабильной версии Moodle

Копируем Moodle в директорию веб-сервера

```
cp -R /opt/moodle /var/www/html/
```

Создаем директорию для данных Moodle

```
mkdir /var/moodledata
```

Шаг 3: Настройка прав доступа

```
sudo chown -R www-data /var/moodledata
sudo chmod -R 777 /var/moodledata
sudo chmod -R 0755 /var/www/html/moodle
```

Шаг 4: Конфигурация MySQL

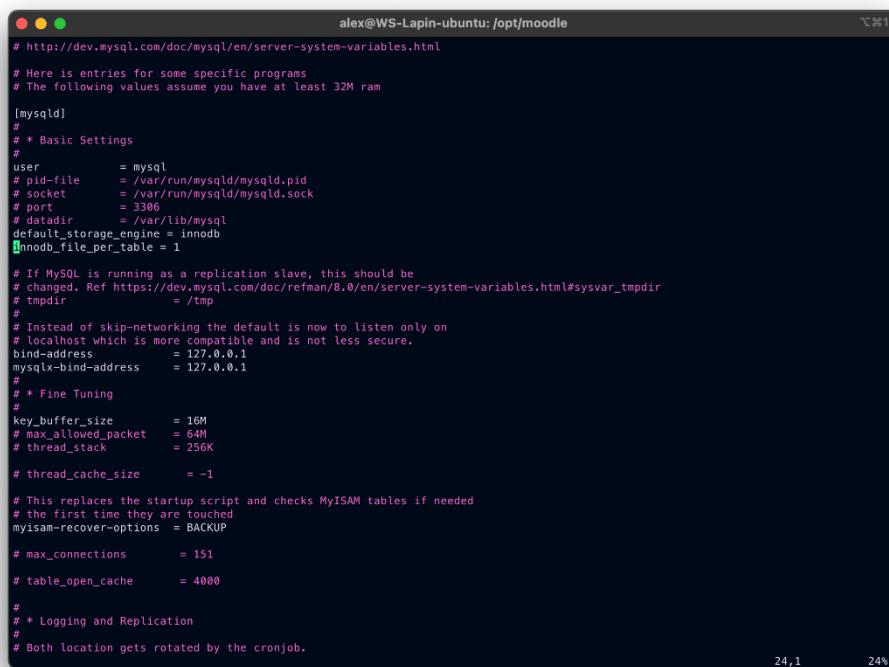
Откроем конфигурационный файл MySQL:

```
sudo vim /etc/mysql/mysql.conf.d/mysqld.cnf
```

Добавим следующие строки:

```
# Устанавливаем InnoDB как движок по умолчанию
default_storage_engine = innodb

# Включаем режим отдельных файлов для каждой таблицы
innodb_file_per_table = 1
```



```
alex@WS-Lapin-ubuntu:/opt/moodle
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user          = mysql
# pid-file     = /var/run/mysqld/mysqld.pid
# socket      = /var/run/mysqld/mysqld.sock
# port         = 3306
# datadir      = /var/lib/mysql
default_storage_engine = innodb
innodb_file_per_table = 1

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir        = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address    = 127.0.0.1
mysqlx-bind-address = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size      = 16M
# max_allowed_packet = 64M
# thread_stack        = 256K
# thread_cache_size   = -1

# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam-recover-options = BACKUP

# max_connections     = 151
# table_open_cache    = 4000
#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
#
```

Рисунок 24. Конфигурационный файл mysqld.cnf

Перезапустим MySQL:

```
service mysql restart
```

Шаг 5: Создание базы данных Moodle

Создадим базу данных и пользователя:

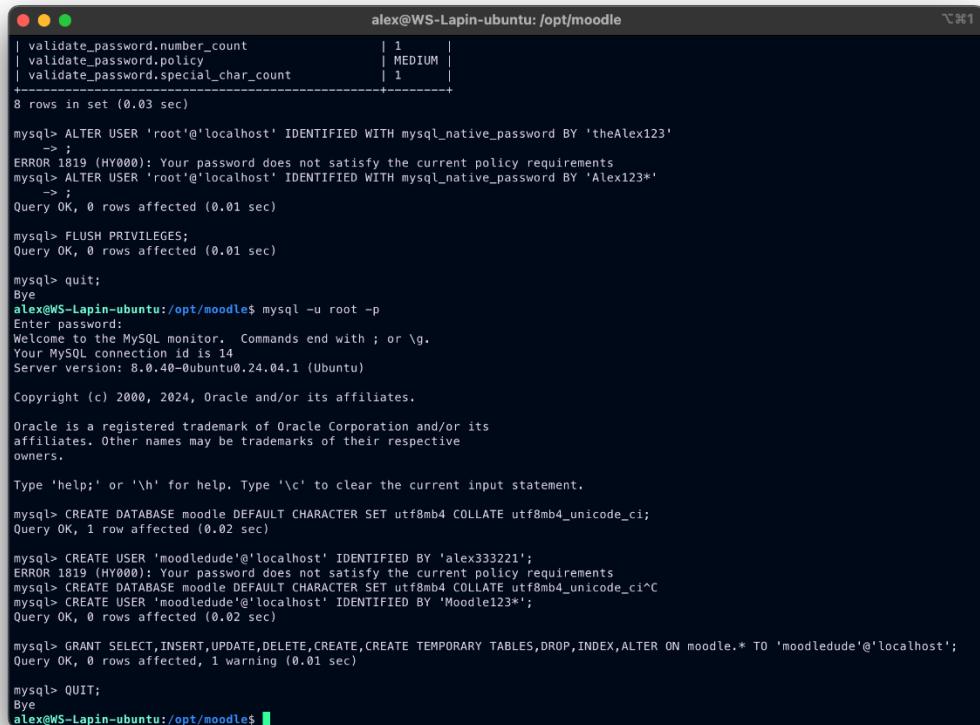
```
# Подключаемся к MySQL как root
mysql -u root -p

# Создаем базу данных с поддержкой Unicode
CREATE DATABASE moodle DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

# Создаем пользователя для Moodle
CREATE USER 'moodledude'@'localhost' IDENTIFIED BY 'YOUR_PASSWORD';

# Назначаем необходимые права пользователю
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,CREATE TEMPORARY TABLES,DROP,INDEX,ALTER ON
moodle.* TO 'moodledude'@'localhost';
```

```
# ВЫХОДИМ ИЗ MySQL  
QUIT;
```



```
alex@WS-Lapin-ubuntu: /opt/moodle
+-----+
| validate_password.number_count | 1      |
| validate_password.policy       | MEDIUM |
| validate_password.special_char_count | 1      |
+-----+
8 rows in set (0.03 sec)

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'theAlex123'
      ->;
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'Alex123*'
      ->;
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql> quit;
Bye
alex@WS-Lapin-ubuntu:/opt/moodle$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.40-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE moodle DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
Query OK, 1 row affected (0.02 sec)

mysql> CREATE USER 'moodledude'@'localhost' IDENTIFIED BY 'alex333221';
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
mysql> CREATE DATABASE moodle DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci^C
mysql> CREATE USER 'moodledude'@'localhost' IDENTIFIED BY 'Moodle123*';
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,CREATE TEMPORARY TABLES,DROP,INDEX,ALTER ON moodle.* TO 'moodledude'@'localhost';
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> QUIT;
Bye
alex@WS-Lapin-ubuntu:/opt/moodle$
```

Рисунок 25. Создание базы данных Moodle

Пояснения к командам:

1. Создание базы данных:

```
CREATE DATABASE moodle DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;
```

Назначение:

- Создание отдельной базы данных для Moodle

Параметры:

- `utf8mb4`: Поддержка полного набора символов Unicode
- `utf8mb4_unicode_ci`: Правила сортировки с учетом различных языков

Почему важно:

- Корректное хранение многоязычного контента
- Поддержка эмодзи и специальных символов
- Совместимость с различными языками интерфейса

2. Создание пользователя:

```
CREATE USER 'moodledude'@'localhost' IDENTIFIED BY
```

'YOUR_PASSWORD';

Назначение:

- Создание отдельного пользователя для Moodle

Безопасность:

- Изоляция доступа к базе данных
- Минимизация рисков при компрометации
- Упрощение аудита безопасности

3. Назначение прав:

```
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,CREATE TEMPORARY  
TABLES,DROP,INDEX,ALTER ON moodle.* TO 'moodledude'@'localhost';
```

Предоставляемые права:

- 'SELECT': Чтение данных
- 'INSERT': Добавление новых записей
- 'UPDATE': Изменение существующих записей
- 'DELETE': Удаление записей
- 'CREATE': Создание новых таблиц
- 'CREATE TEMPORARY TABLES': Создание временных таблиц
- 'DROP': Удаление таблиц
- 'INDEX': Управление индексами
- 'ALTER': Изменение структуры таблиц

Почему эти права необходимы:

- Установка и обновление Moodle
- Создание новых курсов и активностей
- Управление пользовательскими данными
- Оптимизация производительности базы данных

Принцип наименьших привилегий:

- Пользователь получает только необходимые права
- Нет доступа к другим базам данных
- Ограничение действий только базой данных Moodle
- Повышение безопасности системы

Шаг 6. Конфигурация PHP

1. Найдем и отредактируем файл php.ini:

```
sudo vim /etc/php/8.3/apache2/php.ini
```

2. Найдем строку с max_input_vars (или добавим её, если отсутствует):

```
max_input_vars = 5000
```

3. Перезапустим Apache:

```
sudo systemctl restart apache2
```

Пояснения:

Параметр max_input_vars:

- Определяет максимальное количество входных переменных, которые PHP может обработать в одном запросе
- По умолчанию в PHP: 1000
- Почему нужно увеличить:
 - Moodle использует сложные формы с большим количеством полей
 - Необходимо для работы с большими наборами данных
 - Массовое редактированием элементов

Шаг 7. Генерация самоподписанных SSL-сертификатов

1. Создадим директорию для сертификатов и сертификаты:

```
sudo mkdir /etc/apache2/ssl  
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \  
-keyout /etc/apache2/ssl/moodle.key \  
-out /etc/apache2/ssl/moodle.crt
```

2. Включаем необходимые модули Apache:

```
sudo a2enmod ssl  
sudo a2enmod rewrite
```

Рисунок 26. Генерация самоподписанного SSL-сертификата

3. Создаем конфигурацию для HTTPS:

```
sudo nano /etc/apache2/sites-available/moodle-ssl.conf
<virtualHost *:443>
    ServerName 192.168.0.137
    DocumentRoot /var/www/html

    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/moodle.crt
    SSLCertificateKeyFile /etc/apache2/ssl/moodle.key

    <Directory /var/www/html>
        Options Indexes FollowSymLinks Multiviews
        AllowOverride All
        Require all granted

        RewriteEngine On
        RewriteBase /
        RewriteCond %{REQUEST_URI} !^/install\.php
        RewriteCond %{REQUEST_URI} !^/favicon\.ico
        RewriteCond %{REQUEST_URI} !^/robots\.txt
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/moodle_error.log
    CustomLog ${APACHE_LOG_DIR}/moodle_access.log combined
</VirtualHost>
```

4. Создаем редирект с HTTP на HTTPS:

```
sudo nano /etc/apache2/sites-available/moodle-redirect.conf  
<virtualHost *:80>
```

```
ServerName 192.168.0.137
Redirect permanent /moodle https://192.168.0.137/moodle
</VirtualHost>
```

5. Включаем сайты и отключаем дефолтный:

```
sudo a2ensite moodle-ssl
sudo a2ensite moodle-redirect
sudo a2dissite 000-default
```

6. Обновляем config.php Moodle:

```
sudo vim /var/www/html/moodle/config.php
<?php // Moodle configuration file

unset($CFG);
global $CFG;
$CFG = new stdClass();

$CFG->dbtype      = 'mysqli';
$CFG->dblibrary   = 'native';
$CFG->dbhost      = 'localhost';
$CFG->dbname      = 'moodle';
$CFG->dbuser      = 'moodledude';
$CFG->dbpass      = 'Moodle123*';
$CFG->prefix      = 'mdl_';
$CFG->dboptions = array (
    'dbpersist' => 0,
    'dbport' => '',
    'dbsocket' => '',
    'dbcollation' => 'utf8mb4_unicode_ci',
);
$CFG->wwwroot     = 'https://192.168.0.137/moodle';
$CFG->dataroot    = '/var/moodledata';
$CFG->admin        = 'admin';
$CFG->sslproxy    = false;
$CFG->cookiesecure = true;
$CFG->directorypermissions = 0777;

require_once(__DIR__ . '/lib/setup.php');
```

7. Перезапускаем Apache:

```
sudo systemctl restart apache2
```

Пояснения к настройке SSL:

1. Включение модулей Apache:

```
sudo a2enmod ssl  
sudo a2enmod rewrite
```

- ssl модуль:
 - Обеспечивает поддержку HTTPS
 - Управляет SSL/TLS соединениями
- rewrite модуль:
 - Позволяет переписывать URL-адреса

2. Конфигурация HTTPS (moodle-ssl.conf):

a. Основные настройки:

- i. Определение порта 443 для HTTPS
- ii. Указание путей к SSL-сертификатам
- iii. Настройка прав доступа к директориям

3. Редирект HTTP на HTTPS:

- Назначение: Автоматическое перенаправление всех HTTP-запросов на HTTPS

4. Настройка config.php:

a. Важные параметры:

- i. `wwwroot`: Использование HTTPS в основном URL
- ii. `cookiesecure`: Защита cookies через HTTPS
- iii. `sslproxy`: Настройки для работы за прокси

Шаг 8. Установка Moodle

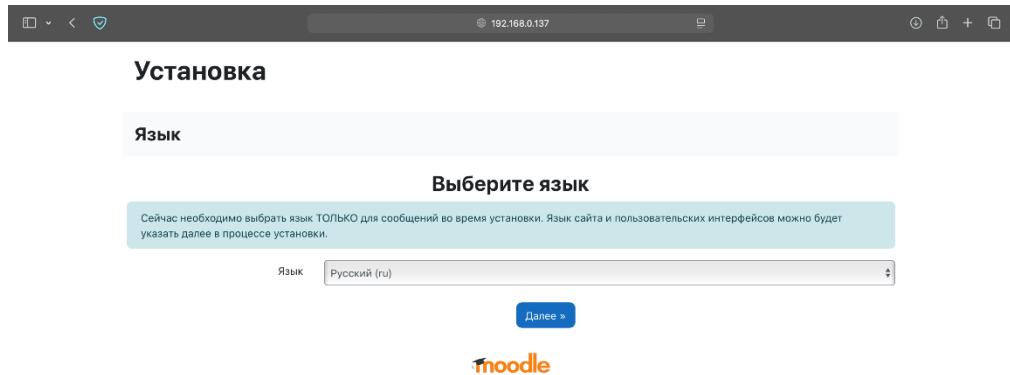


Рисунок 27. Выбор языка Moodle

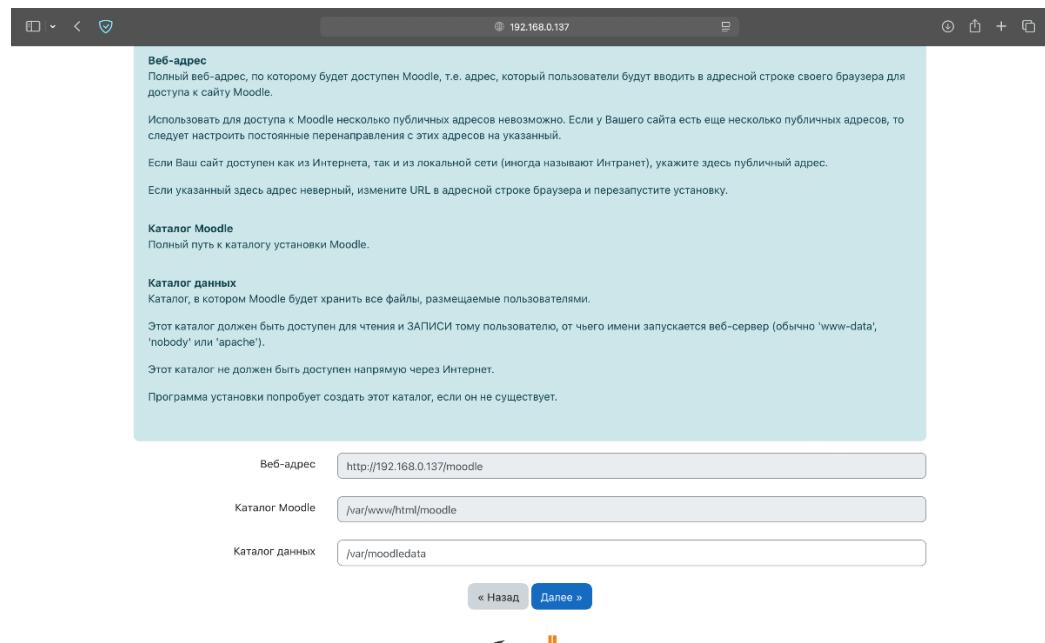


Рисунок 28. Подтвердите пути

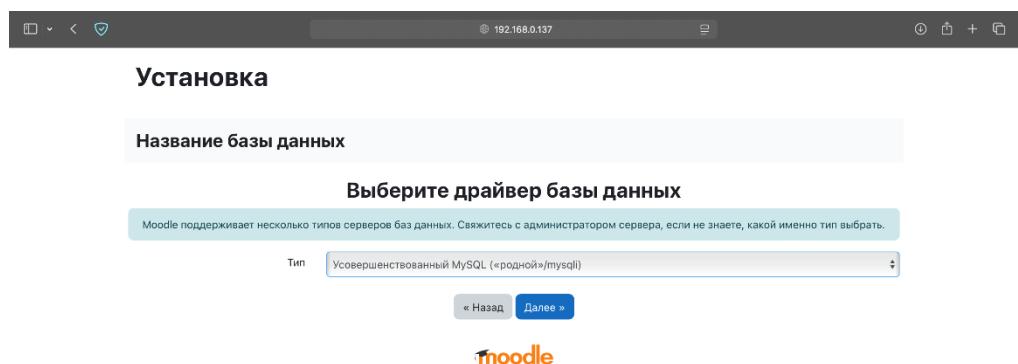


Рисунок 29. Выберите драйвер базы данных

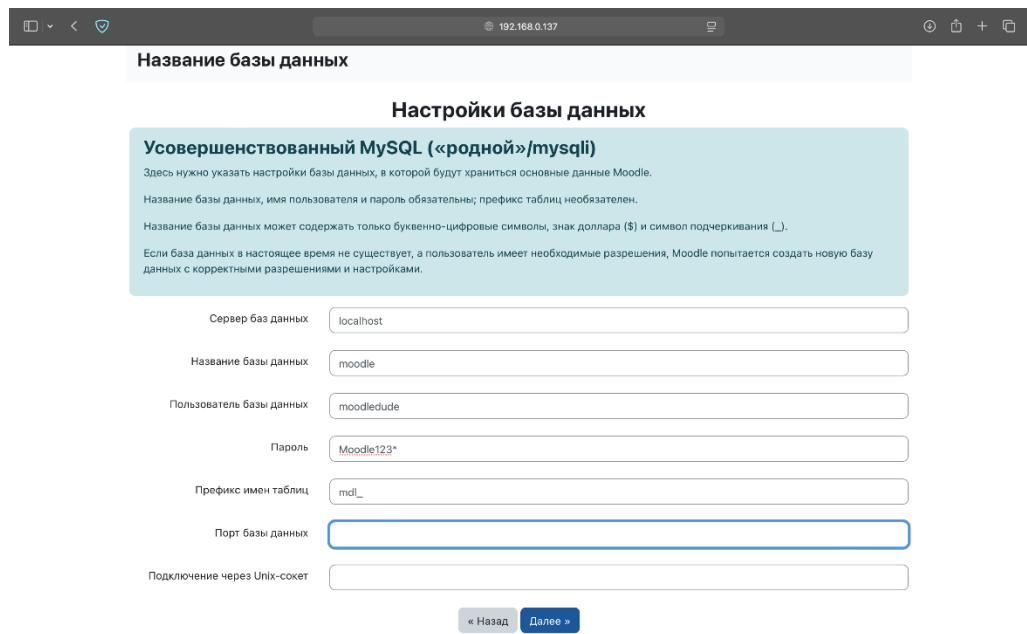


Рисунок 30. Настройки базы данных

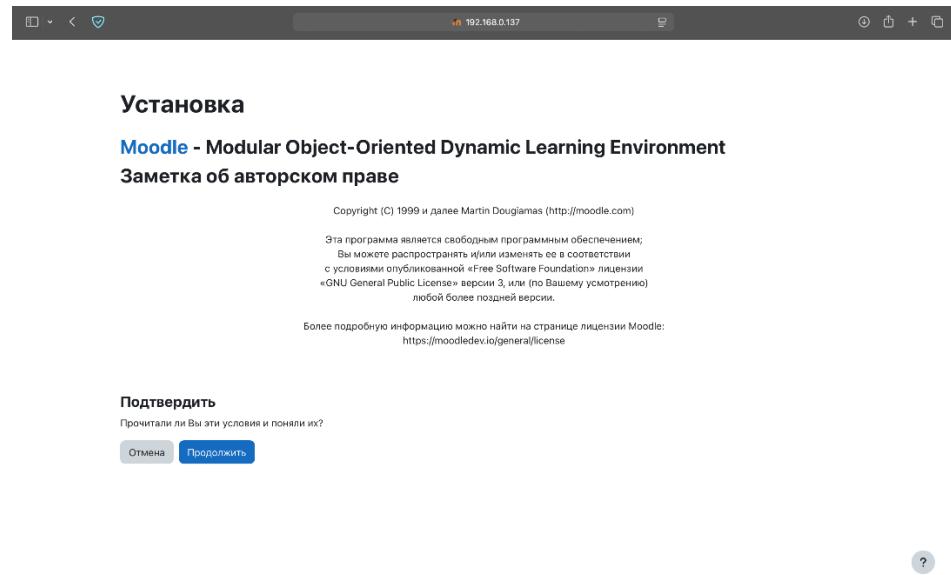


Рисунок 31. Подтверждение согласия с лицензией.

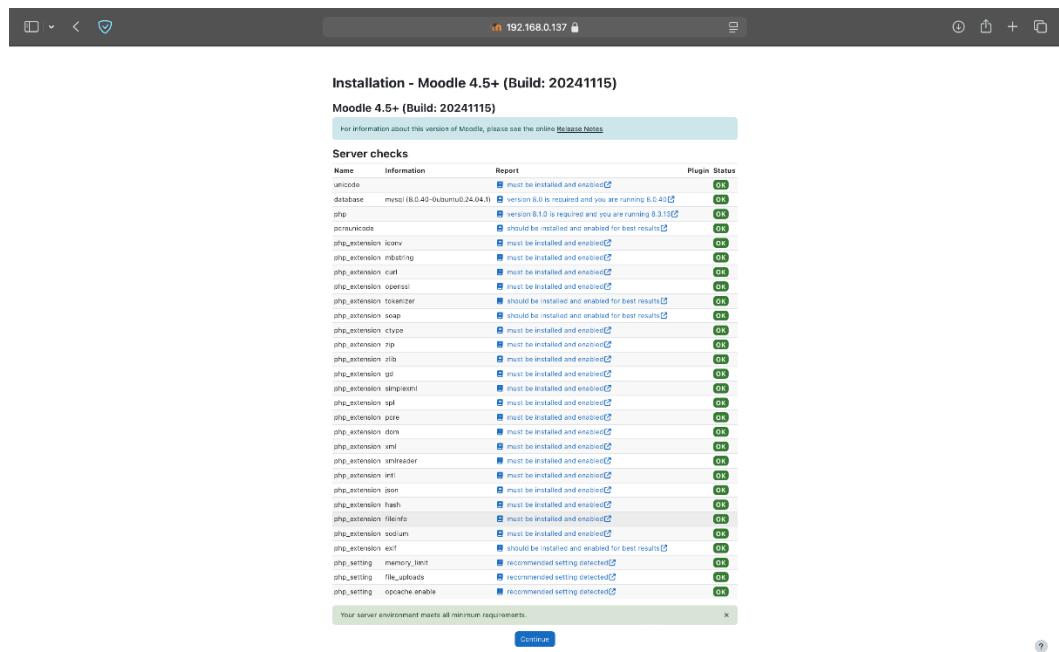


Рисунок 32. Moodle Server Check

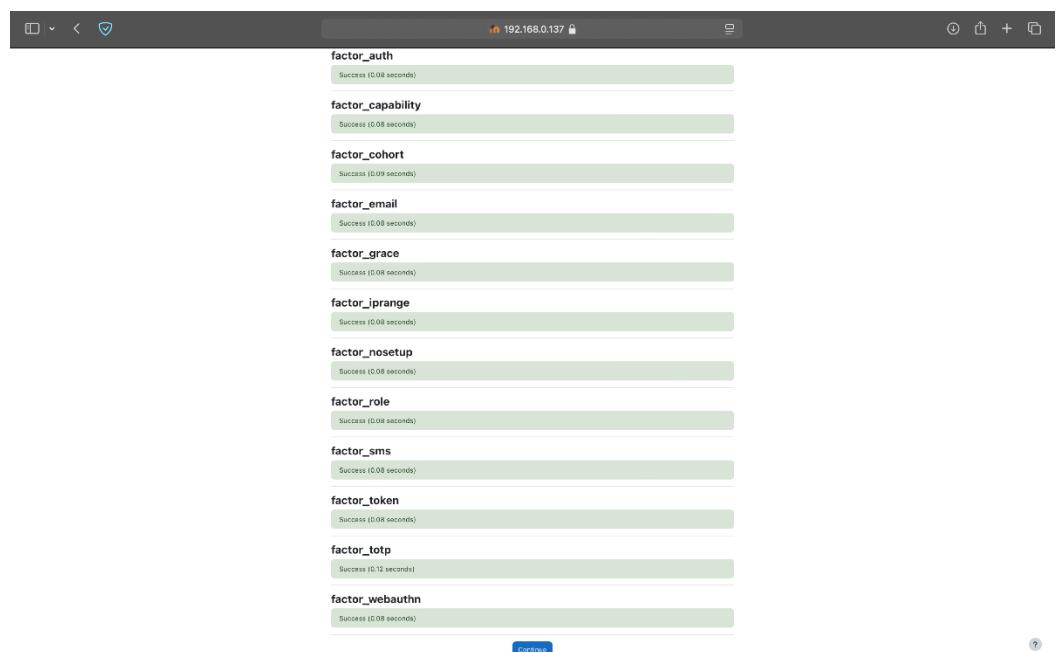


Рисунок 33. Moodle Install Ready

Шаг 9. Настройка Moodle

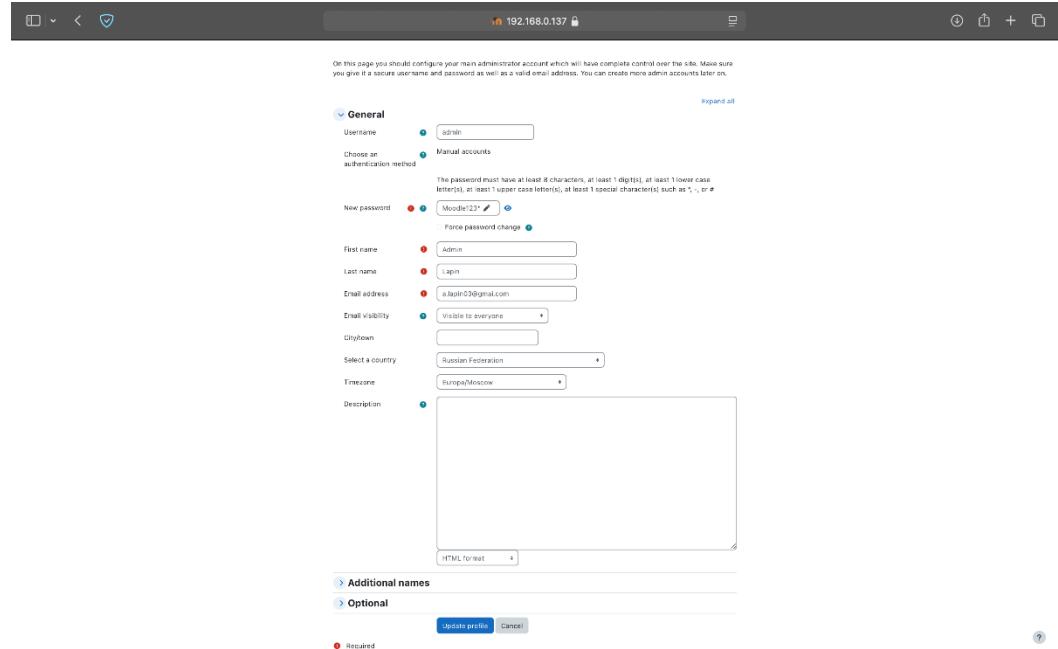


Рисунок 34. Настройка администратора

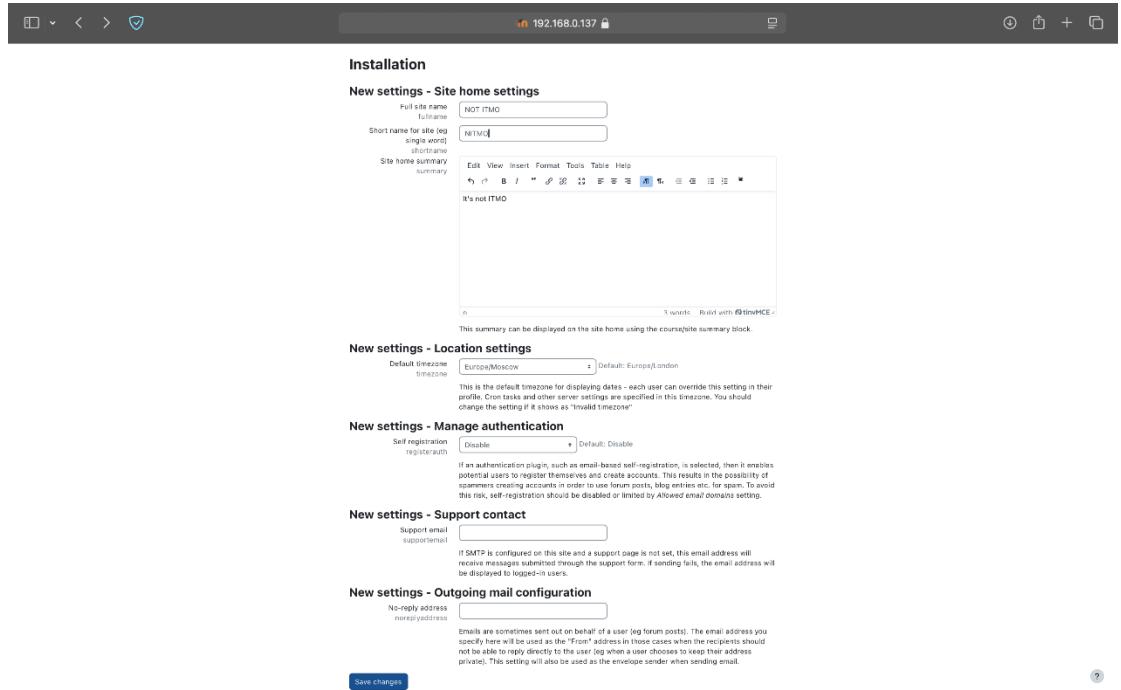


Рисунок 35. Site home settings

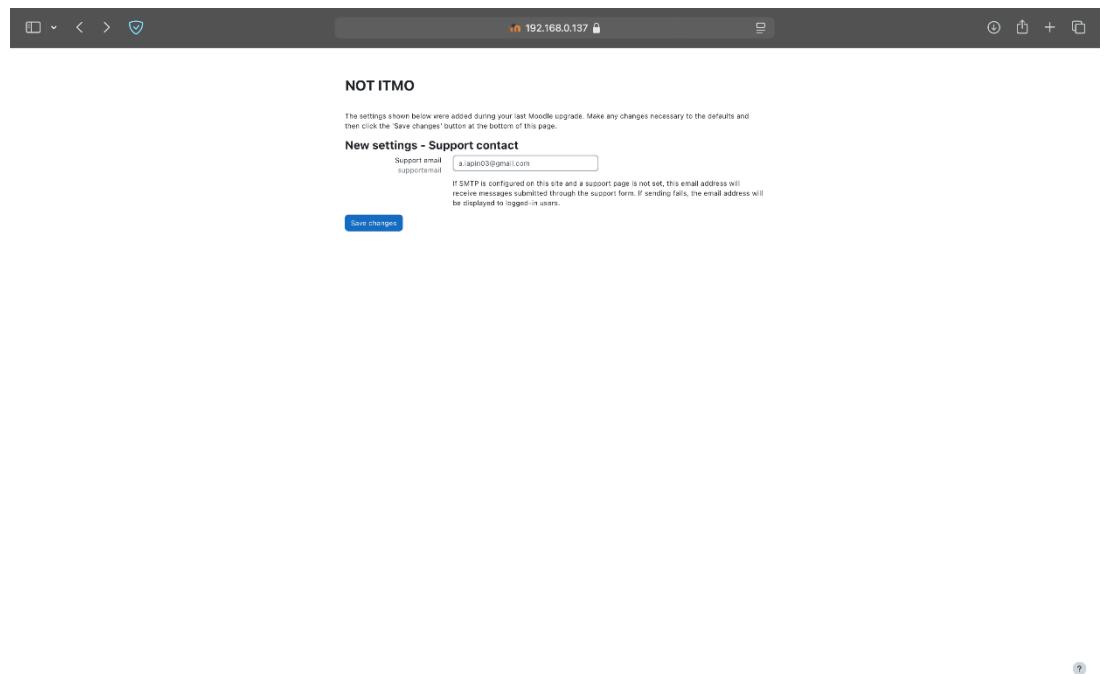


Рисунок 36. Support contact

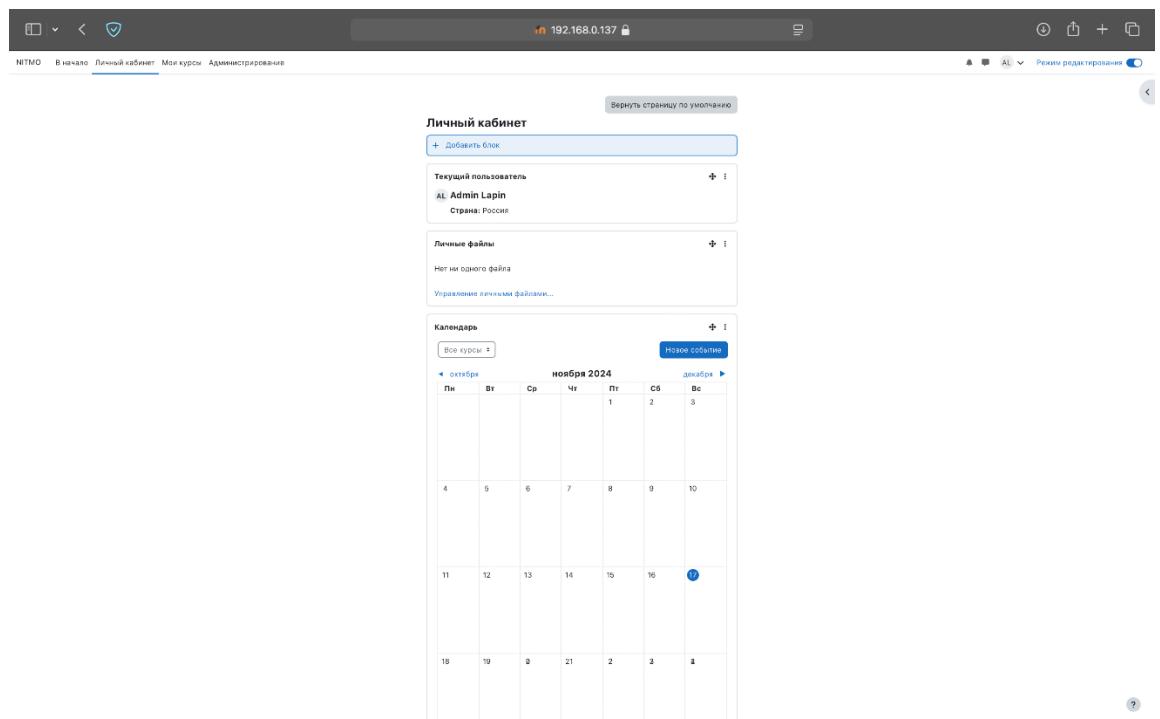


Рисунок 37. Welcome screen

Настроим системные пути к du, aspell, dot:

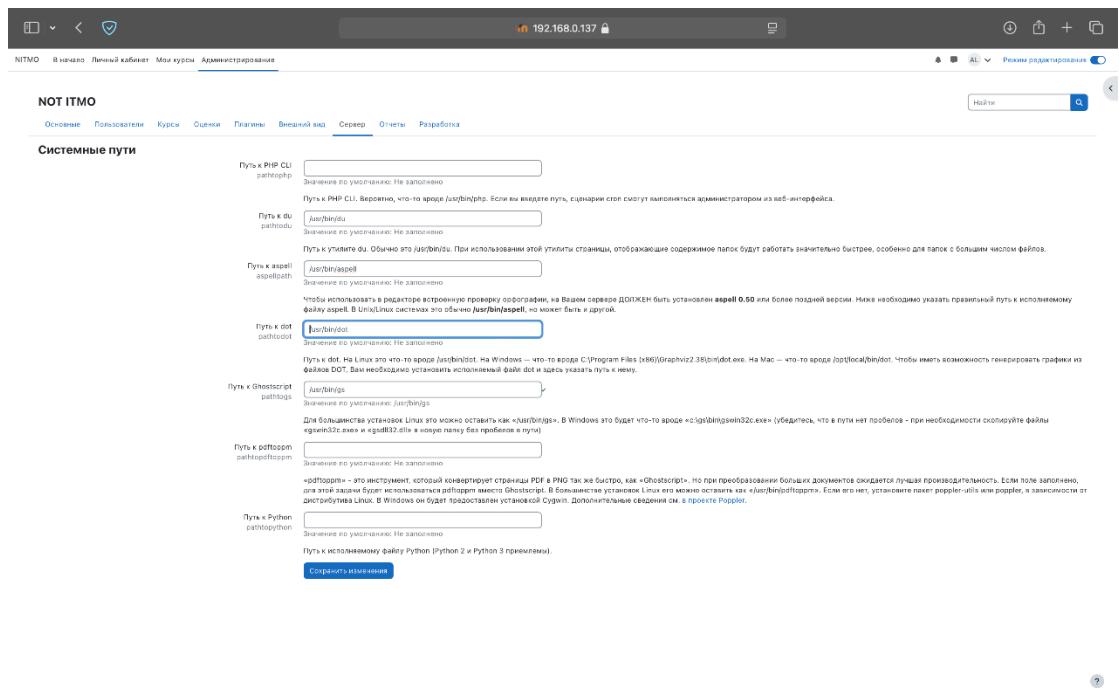


Рисунок 38. System paths

Шаг 10. Увеличим максимальный размер загружаемого файла

1. Изменим настройки PHP:

```
sudo vim /etc/php/8.3/apache2/php.ini
```

Внесем следующие изменения:

```
; Увеличиваем лимиты загрузки файлов
upload_max_filesize = 10G
post_max_size = 10G
max_execution_time = 3600
max_input_time = 3600
memory_limit = 512M
```

2. Настроим Apache:

```
sudo vim /etc/apache2/apache2.conf
```

Добавим секцию Directory:

```
<Directory /var/www/html/moodle>
    LimitRequestBody 10737418240 # 10GB в байтах (10 * 1024 * 1024 * 1024)
</Directory>
ServerName 192.168.0.137
```

3. Отредактируем конфигурацию PHP-FPM:

```
sudo vim /etc/php/8.3/fpm/php.ini
```

Внесем те же изменения, что и в п.1

```
upload_max_filesize = 10G  
post_max_size = 10G  
max_execution_time = 3600  
max_input_time = 3600  
memory_limit = 512M
```

4. Перезапустим Apache и PHP-FPM:

```
sudo systemctl restart apache2  
sudo systemctl restart php8.3-fpm
```

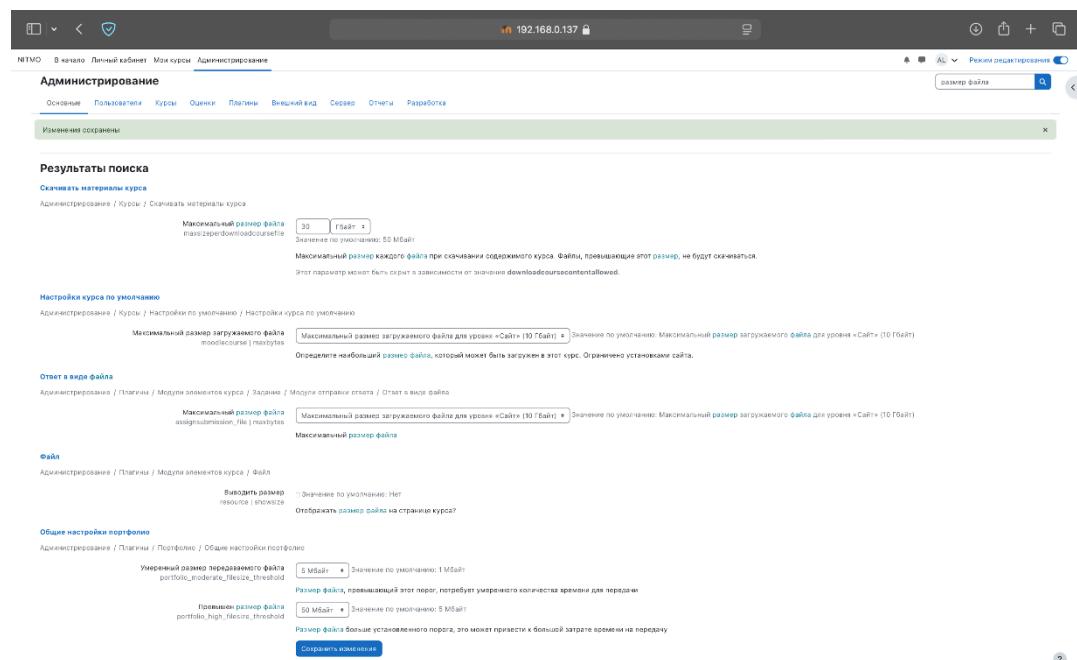
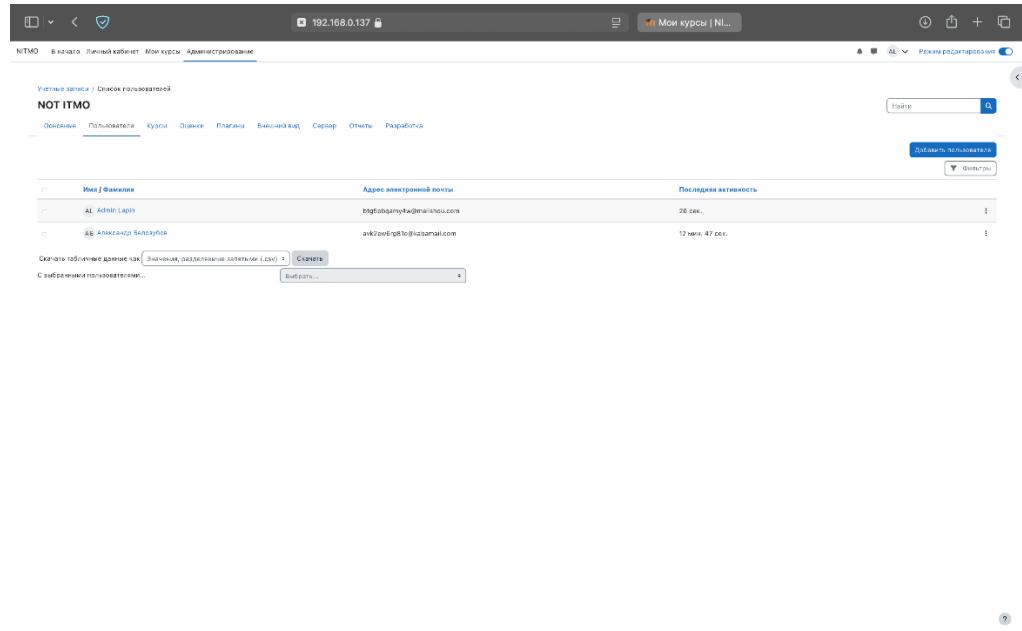


Рисунок 39. Настройки загрузки файлов

Шаг 11. Добавление пользователей и создание курса



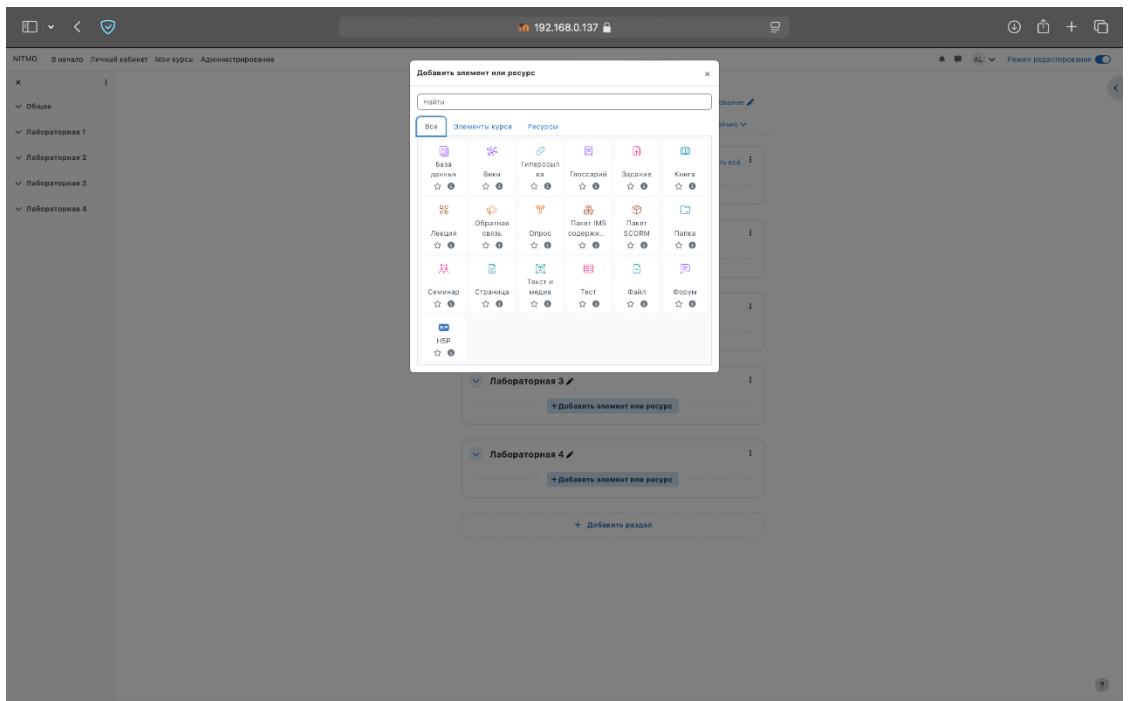
The screenshot shows the 'User list' page in the NITMO administration interface. At the top, there are tabs for 'Основные' (Main), 'Пользователи' (Users), 'Курсы' (Courses), 'Оценки' (Assessments), 'Планы' (Plans), 'Библиотека' (Library), 'Сервер' (Server), 'Отчеты' (Reports), and 'Разработка' (Development). The 'Пользователи' tab is selected. The main area displays a table with two rows of user data:

Имя / Фамилия	Адрес электронной почты	Последняя активность
AI Admin Lapis	bsg@vkontakte.ru@mail.ru	28 сек.
Александр Белозубов	ak721eng8@mail.ru	12 мин. 47 сек.

Below the table, there are buttons for 'Import user data from CSV' and 'Select...'.

Рисунок 40. Список пользователей

Добавили пользователя Белозубов А.В. и назначили ему роль студента.



The screenshot shows the 'Добавить элемент или ресурс' (Add element or resource) dialog box in the NITMO administration interface. The dialog has three tabs: 'Все' (All), 'Элементы курса' (Course elements), and 'Ресурсы' (Resources). The 'Все' tab is selected. It contains a grid of icons representing different course elements:

- База данных
- Блок
- Гиперссылка
- Голосовой
- Задание
- Книга
- Лекция
- Обратная связь
- Опрос
- Пакет IMS
- Пакет SCORM
- Папка
- Семинар
- Страница
- Текст и медиа
- Тест
- Файл
- Форум
- НПР

Below the grid, there are two course structures listed:

- Лабораторная 3
- Лабораторная 4

At the bottom of the dialog, there are buttons for '+ Добавить элемент или ресурс' (Add element or resource) and '+ Добавить раздел' (Add section).

Рисунок 41. Добавление элемента курса

Рисунок 42. Создание курса

В рамках демонстрации работы Moodle был создан курс "Технологии виртуализации" со следующей структурой:

1. Лабораторная 1
 - a. Лекция 1: Введение (с презентацией в формате PPTX)
 - b. Лабораторная работа 1 с установленными сроками:
 - i. Дата открытия: 17 ноября 2024
 - ii. Срок сдачи: 24 января 2025
2. Лабораторная 2
 - a. Опрос "Nginx или apache?"
 - b. Лабораторная работа 2:
 - i. Открыта с 17 ноября 2024
 - ii. Срок сдачи: 1 декабря 2024
 - c. Важное задание (тест):
 - i. Открыто с 17 ноября 2024
 - ii. Закрывается 20 ноября 2024
3. Лабораторная 3
 - a. Раздел скрыт от студентов (находится в разработке)

Курс структурирован по лабораторным работам.

В курс был зачислен студент Белозубов А.В.

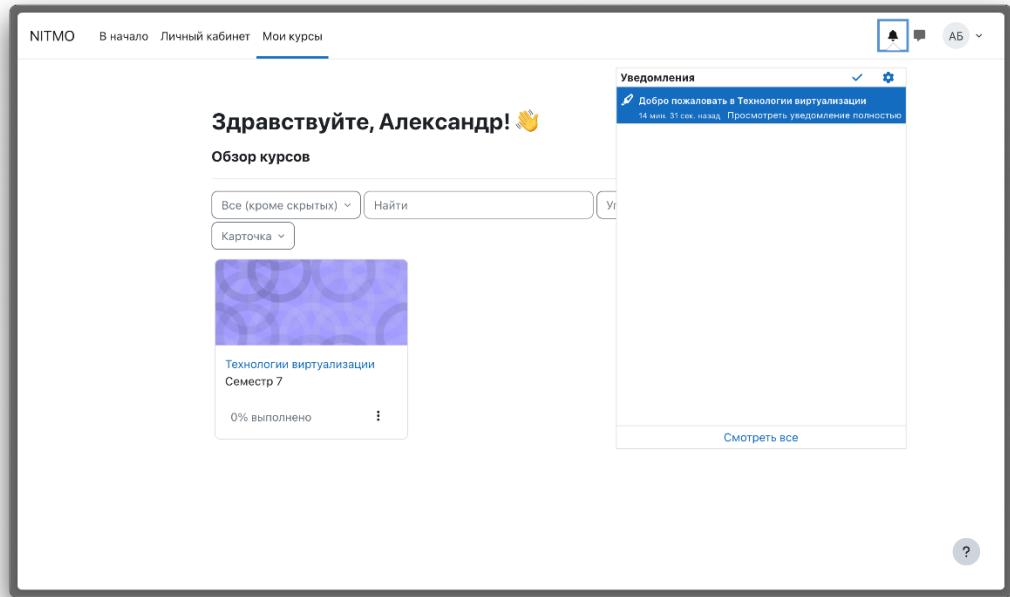


Рисунок 43. Список курсов студента Белозубов А.В.

This screenshot shows the detailed view of the 'Технологии виртуализации' course. The left sidebar has a 'Общее' tab selected, showing course structure like 'Лабораторная 1' and 'Лабораторная 2'. The main content area is titled 'Технологии виртуализации'. It includes tabs for 'Курс', 'Участники', 'Оценки', and 'Компетенции'. Under 'Курс', there are sections for 'Общее', 'Лабораторная 1', and 'Лабораторная 2'. Each section contains various course components such as lectures, assignments, and lab reports. For example, 'Лабораторная 1' includes a lecture titled 'Лекция 1. Введение' and a report titled 'Лабораторная 1 РПТХ'. Both sections have a 'Надо сделать' (To do) button.

Рисунок 44. Просмотр курса от студента Белозубов А.В.

Результаты выполнения задания

В ходе выполнения задания была успешно развернута и настроена система электронного документооборота Moodle. Рассмотрим основные достигнутые результаты:

1. Установка и базовая настройка системы:

- a. Успешно установлены все необходимые компоненты (Apache, MySQL, PHP)
 - b. Оптимизированы настройки PHP и MySQL для работы Moodle
 - c. Система полностью функциональна и доступна через веб-интерфейс
2. Выполнение требований задания:
- a. Изучен функционал системы Moodle как платформы для организации обучения и документооборота
 - b. Добавлены два пользователя:
 - i. Администратор-преподаватель
 - ii. Студент (Белозубов А.В.)
 - c. Создан учебный проект (курс "Технологии виртуализации")
 - d. Установлены сроки выполнения для различных элементов курса
 - e. Подключен пользователь к проекту в роли студента
3. Дополнительные достижения:
- a. Настроена поддержка больших файлов (до 10GB)
 - b. Настроена система безопасности с SSL-сертификатами
4. Проверка работоспособности:
- a. Система успешно запускается
 - b. Пользователи могут авторизоваться
 - c. Курс доступен для просмотра
 - d. Работает загрузка материалов
 - e. Корректно отображаются сроки выполнения заданий
 - f. Оценки ставятся
 - g. Тесты сдаются, лабораторные принимаются

Система полностью готова к использованию в образовательном процессе и соответствует всем поставленным в задании требованиям.

Установить и настроить пакет phpvirtualbox на гостевой ОС

- WS_ubuntu

Текст задания

1. Установить и настроить пакет phpvirtualbox
2. Запустить через браузер виртуальную машину с ОС Windows

Выполнение задания

Шаг 1. Установка необходимых зависимостей

1. Установка PHP8.0, nginx, git

```
sudo apt install -y software-properties-common
sudo add-apt-repository ppa:ondrej/php
sudo apt update
sudo apt install -y php8.0 \
    php8.0-fpm \
    php8.0-common \
    php8.0-xml \
    php8.0-mbstring \
    php8.0-curl \
    php8.0-zip \
    php8.0-gd \
    php8.0-ldap \
    php8.0-mysql \
    nginx \
    git
```

Установка необходимых пакетов:

- php8.0 и php8.0-fpm - основной интерпретатор PHP и FastCGI Process Manager
- php8.0-common - общие библиотеки PHP
- php8.0-xml - поддержка XML, необходимая для обработки конфигурационных файлов
- php8.0-mbstring - поддержка многобайтовых строк для работы с различными кодировками
- php8.0-curl - библиотека для выполнения HTTP-запросов к VirtualBox API
- php8.0-zip - поддержка работы с ZIP-архивами
- php8.0-gd - библиотека для работы с изображениями
- php8.0-ldap - поддержка LDAP для возможной интеграции с системами

- аутентификации
 - php8.0-mysql - драйвер MySQL (может использоваться для хранения настроек)
 - nginx - веб-сервер для хостинга phpVirtualBox
 - git - система контроля версий для загрузки исходного кода phpVirtualBox
2. Проверим установку PHP:

```
php8.0 -v
```

3. Стартуем и проверяем статус службы PHP:

```
sudo systemctl start php8.0-fpm
sudo systemctl enable php8.0-fpm
sudo systemctl status php8.0-fpm
```

4. Установка virtualbox7.0

```
sudo apt install -y virtualbox-7.0
```

Это самая старая версия virtualbox, которая поддерживает Ubuntu 24.04.

Шаг 2. Установка phpvirtualbox

```
cd /var/www/
sudo git clone https://github.com/phpvirtualbox/phpvirtualbox.git
sudo git branch -a
sudo git switch -c local-vbox7.0-php8.0 remotes/origin/vbox7.0-php8.0
```

Пояснения:

- Устанавливаем phpvirtualbox в директорию `/var/www/`, где обычно размещаются веб-приложения
- Используем Git для загрузки нужной нам версии phpvirtualbox
- Важно выбрать правильную ветку репозитория, совместимую с установленными версиями VirtualBox (7.0) и PHP (8.0)
- Создание локальной ветки позволяет изолировать наши изменения от основного репозитория и при необходимости получать обновления

Шаг 3. Настройка Nginx под phpvirtualbox

```
sudo vim /etc/nginx/sites-available/phpvirtualbox
server {
    listen 80;
    server_name 192.168.0.137;
```

```

# Set root to /var/www
root /var/www;
index index.php index.html index.htm;

# Handle phpvirtualbox location specifically
location /phpvirtualbox {
    alias /var/www/phpvirtualbox;
    index index.php index.html;

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php8.0-fpm.sock;
        # Important: use $request_filename instead of
$document_root$fastcgi_script_name
        fastcgi_param SCRIPT_FILENAME $request_filename;
        include fastcgi_params;
    }

    try_files $uri $uri/ /phpvirtualbox/index.php?$args;
}

# Serve static files directly
location ~* \.(jpg|jpeg|gif|css|png|js|ico|html)$ {
    access_log off;
    expires max;
}

# Security
location ~ /\.ht {
    deny all;
}
}

```

Пояснения к настройке Nginx:

- Базовая конфигурация сервера:

```

server {
    listen 80;
    server_name 192.168.0.137;
}

```

- Настройка прослушивания HTTP-порта 80
- Указание IP-адреса сервера

- Корневая директория и индексные файлы:

```

root /var/www;
index index.php index.html index.htm;

```

- Указание приоритета индексных файлов

- Обработка phpVirtualBox:

```
location /phpvirtualbox {  
    alias /var/www/phpvirtualbox;  
    index index.php index.html;
```

- Создание отдельного местоположения для phpVirtualBox
- Указание реальной директории приложения

4. Обработка PHP-файлов:

```
location ~ \.php$ {  
    include snippets/fastcgi-php.conf;  
    fastcgi_pass unix:/run/php/php8.0-fpm.sock;  
    fastcgi_param SCRIPT_FILENAME $request_filename;
```

- Настройка обработки PHP через PHP-FPM
- Использование Unix-сокета для взаимодействия с PHP

5. Обработка статических файлов:

```
location ~* \.(jpg|jpeg|gif|css|png|js|ico|html)$ {  
    access_log off;  
    expires max;
```

- Оптимизация доставки статического контента
- Настройка кэширования в браузере

6. Безопасность:

```
location ~ /\.ht {  
    deny all;  
}
```

- Запрет доступа к .htaccess файлам

Выдача прав на файл:

```
sudo chown www-data:www-data /etc/nginx/sites-available/phpvirtualbox
```

Создание символьической ссылки:

```
sudo ln -s /etc/nginx/sites-available/phpvirtualbox /etc/nginx/sites-enabled/  
sudo nginx -t  
sudo systemctl restart nginx
```

Шаг 4. Настройка php

```
sudo vim /etc/php/8.0/fpm/php.ini  
# увеличиваем лимит памяти для PHP-процессов  
memory_limit = 256M
```

```
# Увеличиваем максимальный размер загружаемых файлов  
upload_max_filesize = 50M  
  
# Увеличиваем максимальный размер POST-запросов  
post_max_size = 50M  
  
# Увеличиваем максимальное время выполнения скриптов  
max_execution_time = 300
```

Перезапускаем php-fpm:

```
sudo systemctl restart php8.0-fpm
```

Шаг 5. Настройка phpvirtualbox

1. Выдача прав

```
cd /var/www/phpvirtualbox  
sudo cp config.php-example config.php  
sudo chown -R www-data:www-data /var/www/phpvirtualbox  
sudo chmod -R 755 /var/www/phpvirtualbo
```

2. Отредактируем config.php с правильными настройками

```
sudo vim /var/www/phpvirtualbox/config.php  
# Адрес и порт веб-службы VirtualBox  
var $location = 'http://192.168.0.137:18083';  
# Учетные данные для доступа к VirtualBox  
var $username = 'alex';  
var $password = 'alex';  
# Используются для:  
# - Аутентификации в VirtualBox API  
# - Должны соответствовать пользователю, от имени которого запущен VirtualBox  
# Язык интерфейса  
var $language = 'en';  
# Отключение VRDE  
var $vrde = false;  
# Настройки аутентификации  
var $noAuth = true; # Отключение аутентификации в веб-интерфейсе  
var $authMaster = true; # Предоставление прав администратора всем пользователям  
# Важно для:  
# - Упрощения доступа в тестовой среде  
# - В производственной среде рекомендуется установить false  
# IP-адрес хоста  
var $host = '192.168.0.137';
```

Шаг 6. Настройка веб-службы VirtualBox

Отредактируем файл /etc/default/virtualbox:

```
sudo vim /etc/default/virtualbox
VBOXWEB_USER=alex          # имя пользователя, от которого запущен VirtualBox
VBOXWEB_HOST=192.168.0.137 # адрес, на котором запущена веб-служба VirtualBox
VBOXWEB_PORT=18083 # порт, на котором запущена веб-служба VirtualBox
```

Перезапустим модули ядра VirtualBox:

```
sudo modprobe vboxdrv
sudo modprobe vboxnetadp
sudo modprobe vboxnetflt
```

Запустим VirtualBox system service:

```
sudo vboxconfig
sudo systemctl start vboxdrv
sudo systemctl status vboxdrv
```

Запустим веб-службу VirtualBox:

```
sudo systemctl start vboxweb-service
sudo systemctl enable vboxweb-service
```

Проверим занят порт 18083:

```
sudo ss -tu1np | grep 18083
```

Проверим работу веб-службы VirtualBox:

```
curl -v http://192.168.0.137:18083/
```

Если служба не запускается, то запустим её вручную:

```
sudo /usr/lib/virtualbox/vboxwebsrv --background -H 192.168.0.137 -p 18083
```

Если все еще не работает, то установим и настроим необходимые модули ядра:

```
sudo /sbin/vboxconfig
```

Шаг 7. Доступ к phpVirtualBox

Откройте браузер и перейдите по адресу: <http://192.168.0.137/phpvirtualbox/>

Шаг 8. Исправление ошибки "Method 'ns1:IMachine_getAudioAdapter' not implemented"

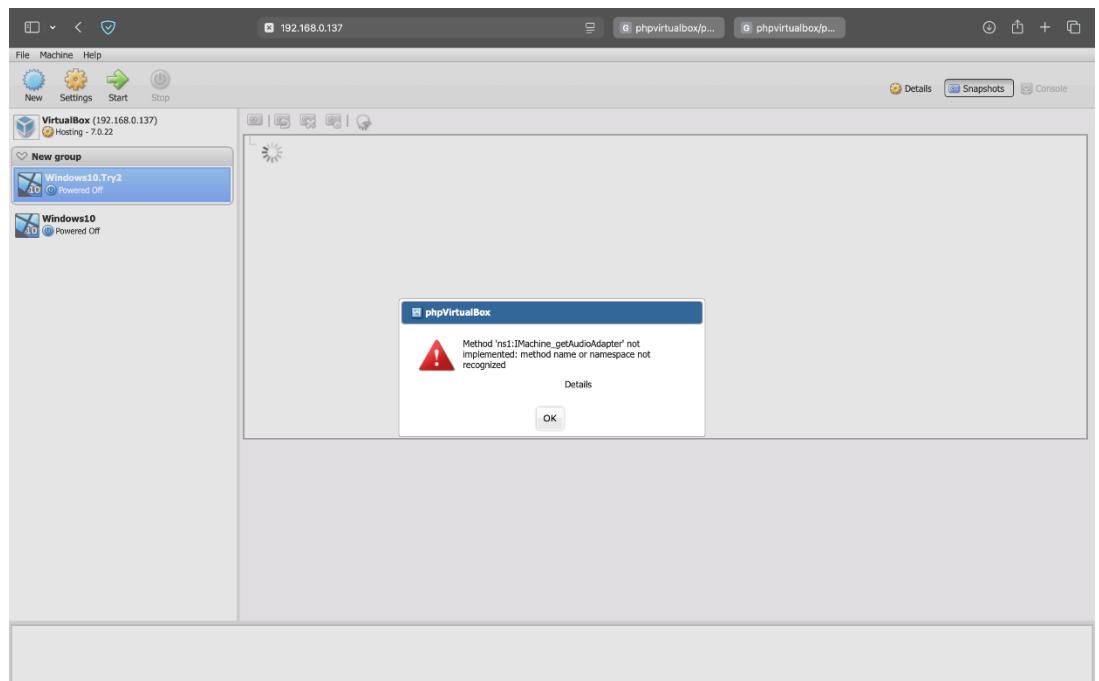


Рисунок 45. Ошибка "Method 'ns1:IMachine_getAudioAdapter' not implemented"

1. Скачиваем vboxwebService-7.0.wsdl, vboxServiceWrappers.php, vboxweb-7.0.wsdl
2. Внести исправления в файл vboxwebService-7.0.wsdl

```
- import location="vboxweb.wsdl" namespace="http://www.virtualbox.org//"
+ import location="vboxweb-7.0.wsdl" namespace="http://www.virtualbox.org//"
```

3. Изменить версию в файле /var/www/phpvirtualbox/endpoints/lib/config.php

```
- "6.1-0"
+ "7.0-0"
```

4. Внести изменения в файл /var/www/phpvirtualbox/endpoints/lib/vboxconnector.php

```
line: 4257
$m->audioSettings->Adapter->enabled = ($args['audioAdapter']['enabled'] ? 1 : 0);
$m->audioSettings->Adapter->audioController = $args['audioAdapter']['audioController'];
$m->audioSettings->Adapter->audioDriver = $args['audioAdapter']['audioDriver'];
lines: 1943 - 1945
'audioAdapter' => array(
'enabled' => $m->audioSettings->Adapter->enabled,
'audioController' => (string)$m->audioSettings->Adapter->audioController,
'audioDriver' => (string)$m->audioSettings->Adapter->audioDriver,
),
```

Ссылка на открытое issue: <https://github.com/phpvirtualbox/phpvirtualbox/issues/313>

Шаг 9. Исправление ошибки "Choose a Virtual optical Disk File Not Load files..."

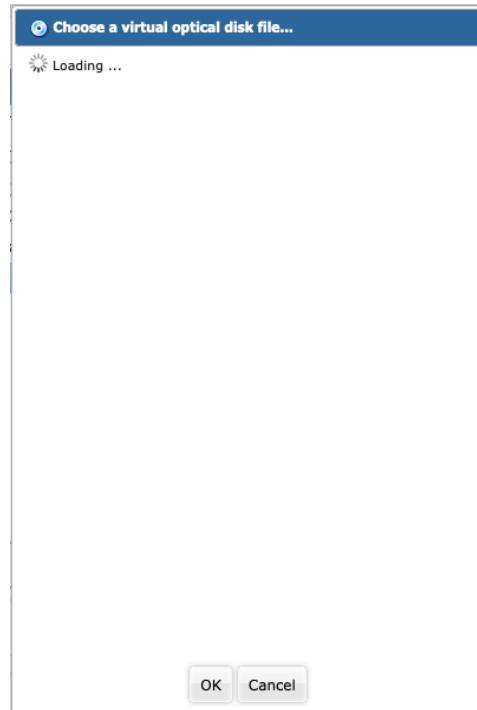


Рисунок 46. Ошибка "Choose a Virtual optical Disk File Not Load files..."

1. Исправим ошибку в файле /var/www/phpvirtualbox/endpoints/jqueryFileTree.php:

```
sudo vim /var/www/phpvirtualbox/endpoints/jqueryFileTree.php
*/
function getdir($dir, $dirsOnly=false, $recurse=array()) {

+    global $allowed_exts;
+
    if(!$dir) $dir = DSEP;

    $entries = getDirEntries($dir, $dirsOnly);
@@ -251,9 +253,9 @@
        // Push file on to stack
    } else {

-
        $ext = strtolower(preg_replace('/^.*\.\./', '', $file));
+
        $ext = strtolower(preg_replace('/^.*\.\./', '', $path));

-
        if(count($allowed) && !$allowed['.'.$ext]) continue;
+
        if(count($allowed_exts) && !$allowed_exts['.'.$ext]) continue;

        array_push($dirents, file_entry($path));
    }
}
```

Ссылка на открытое issue: <https://github.com/phpvirtualbox/phpvirtualbox/issues/273>

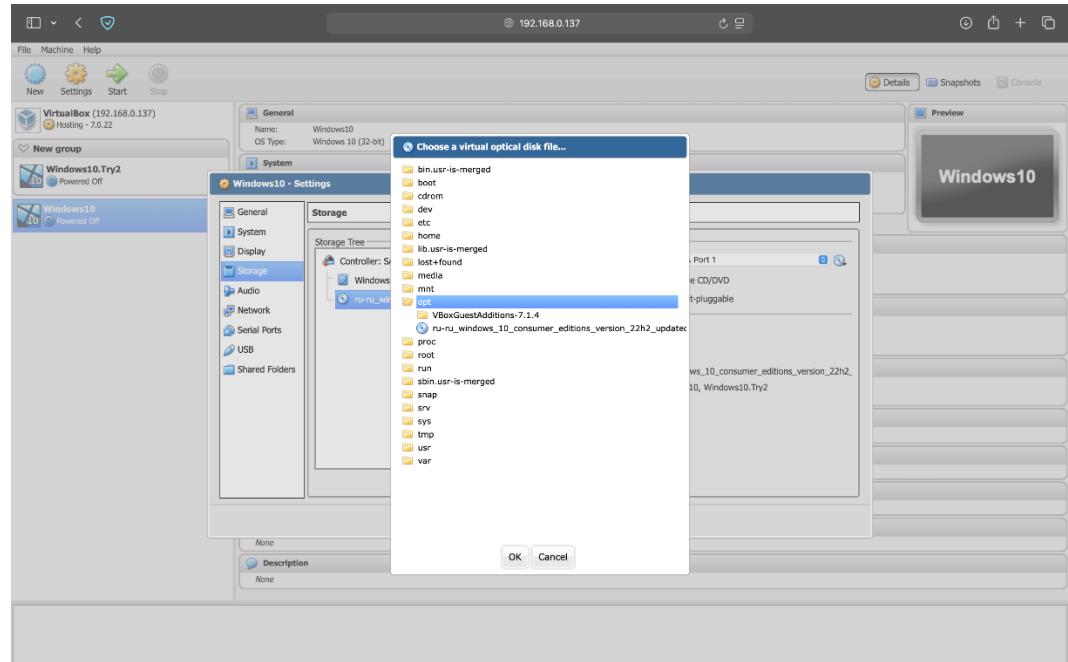


Рисунок 47. После исправления ошибки "Choose a Virtual optical Disk File Not Load files..."

Итог

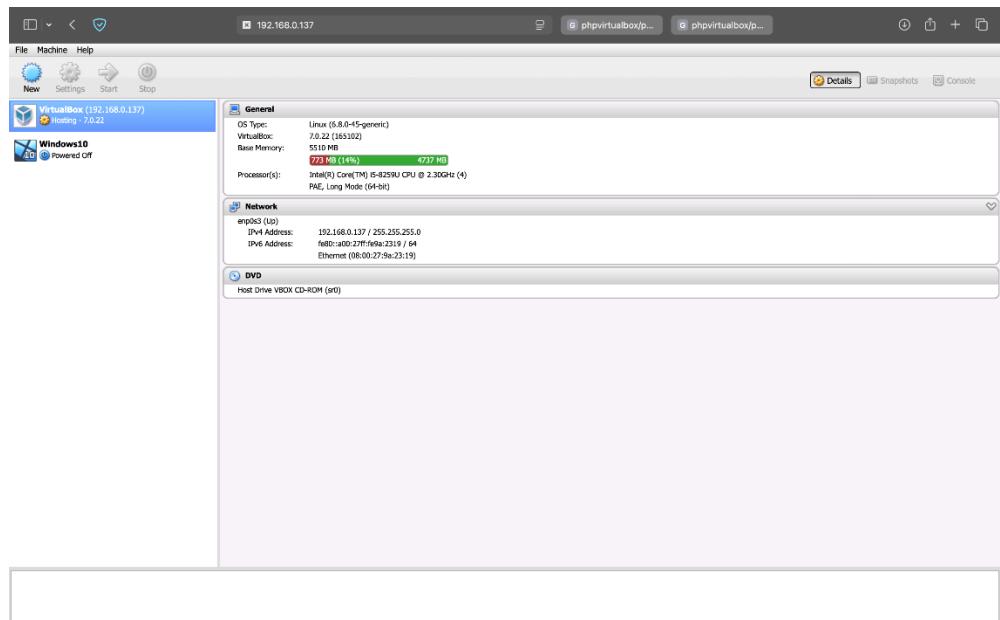


Рисунок 48. Экран приветствия

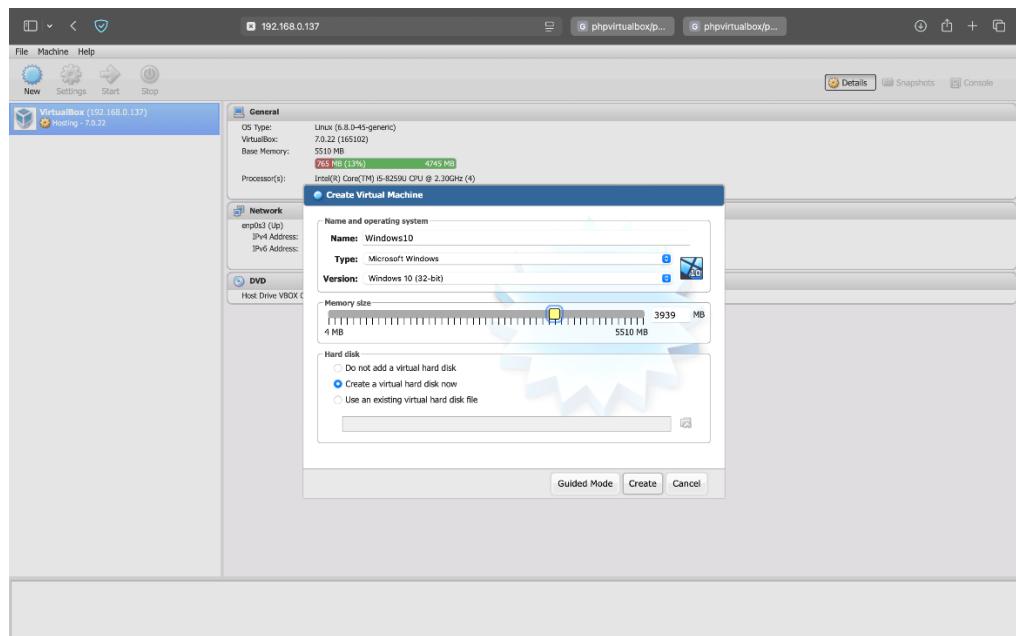


Рисунок 49. Создание виртуальной машины

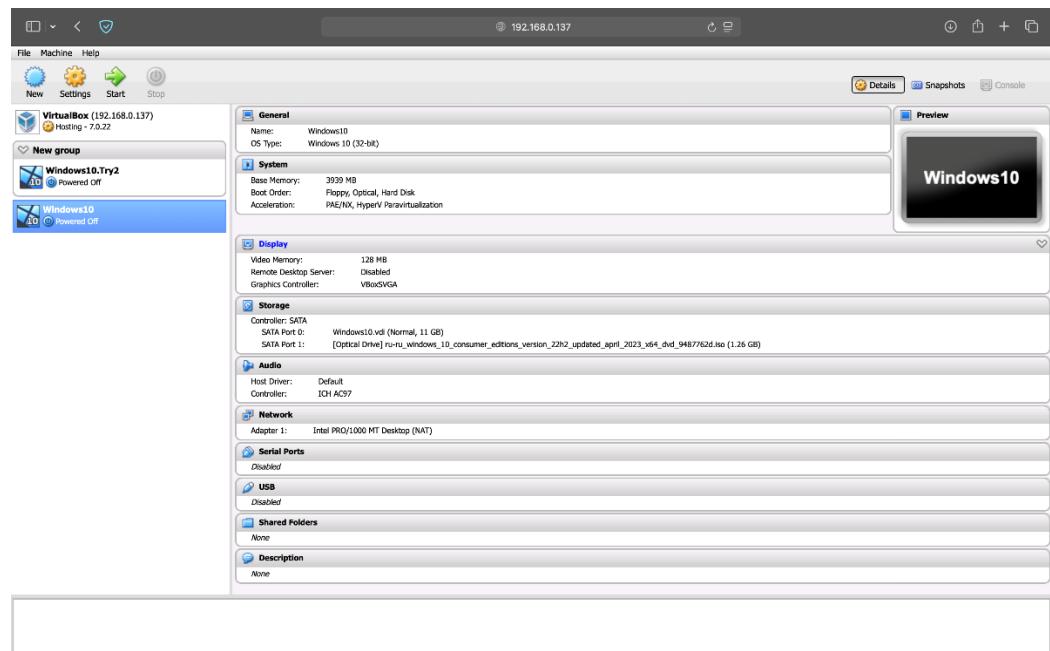


Рисунок 50. Список виртуальных машин

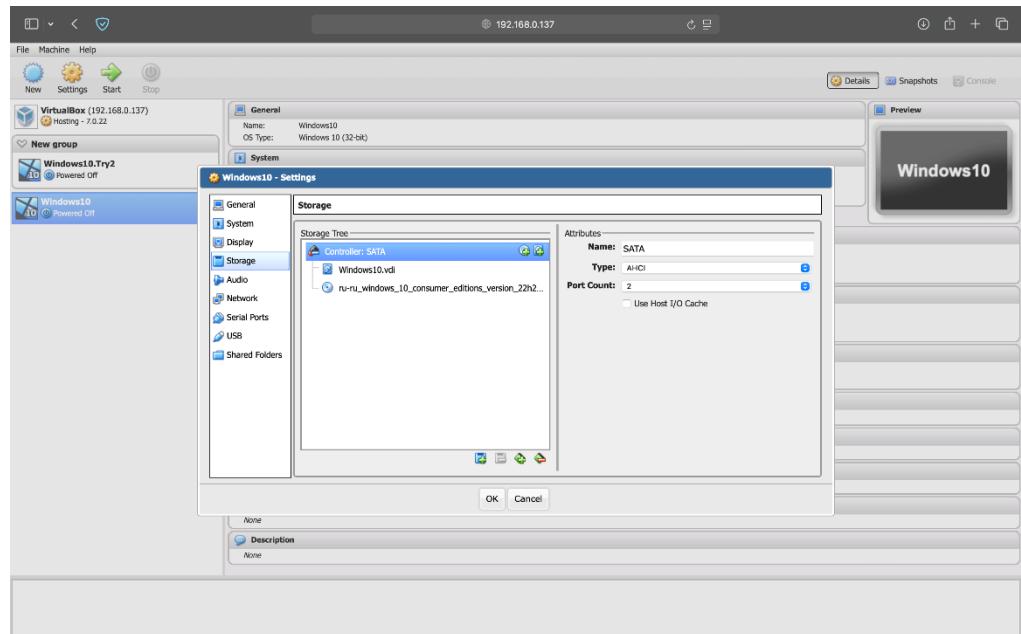


Рисунок 51. Выбор виртуального оптического диска

Получилось создать виртуальную машину, но запустить её не удалось. При загрузке падает с ошибкой.

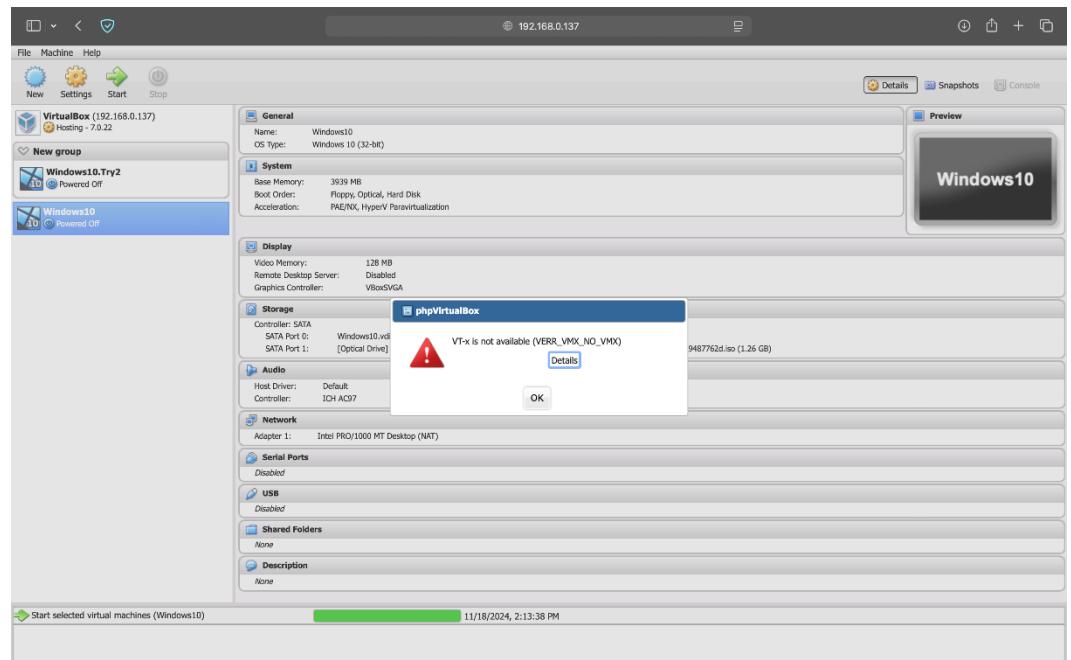


Рисунок 52. Ошибка 1 при загрузке виртуальной машины

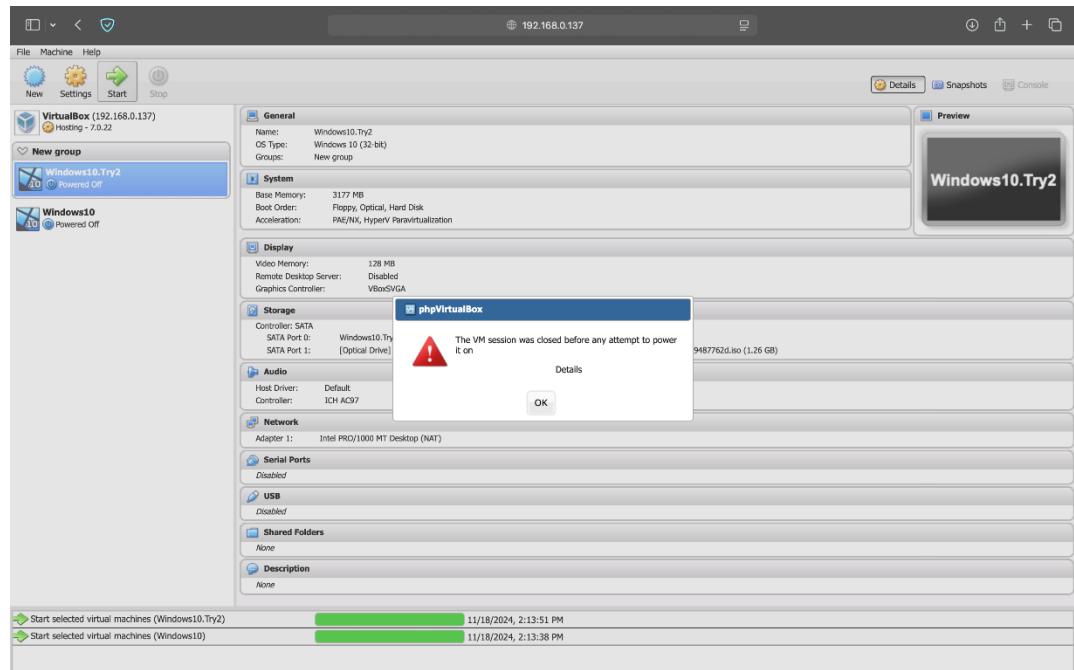


Рисунок 53. Ошибка 2 при загрузке виртуальной машины

Разница между двумя виртуальными машинами в динамическом или фиксированном размере диска.

Результаты выполнения задания

1. Достигнутые цели:
 - a. Успешно установлен и настроен пакет `phpvirtualbox`
 - b. Настроена интеграция с `VirtualBox 7.0`
 - c. Исправлены некоторые ошибки в работе `phpvirtualbox`
2. Частично решенные задачи:
 - a. Создание виртуальных машин работает корректно
 - b. Интерфейс управления функционирует
 - c. Возможность выбора ISO-образов восстановлена после исправления ошибки
3. Нерешенные проблемы:
 - a. Не удалось запустить виртуальные машины через веб-интерфейс
 - b. Возникают ошибки при попытке загрузки ВМ (как с динамическим, так и с фиксированным размером диска)
4. Возможные причины проблем:
 - a. Проблемы с совместимостью версий `VirtualBox` и `phpvirtualbox`
5. Рекомендации по дальнейшей отладке:

- a. Рассмотреть возможность использования более старой версии VirtualBox и Ubuntu
6. Общий вывод:
 - a. Задача выполнена частично из-за несовместимости новой версии VirtualBox и phpvirtualbox. Основная функциональность установлена и настроена, но для обеспечения полноценной работы виртуальных машин через веб-интерфейс требуется использовать более старые версии Ubuntu и VirtualBox.

Установка приложения Webmin на гостевой ОС - WS_ubuntu

Текст задания

1. Установить и настроить пакет https://www.webmin.com
2. Изменить системное время
3. Настроить в интерфейсе пользователя русский язык.
4. Добавить нового пользователя в систему
5. Изучить функционал системы

Выполнение задания

Шаг 1. Обновление системы

Перед началом установки рекомендуется обновить список пакетов и установить последние обновления.

```
sudo apt update && sudo apt upgrade -y
```

Шаг 2: Установка необходимых зависимостей

Webmin требует некоторых зависимостей, которые необходимо установить перед началом установки.

```
sudo apt install -y software-properties-common apt-transport-https wget
```

Шаг 3: Добавление репозитория Webmin

Webmin не доступен в стандартных репозиториях Ubuntu, поэтому необходимо добавить официальный репозиторий Webmin.

1. Импортируем ключ GPG Webmin:

```
wget -q - http://www.webmin.com/jcameron-key.asc | sudo apt-key add -o-
```

2. Добавляем репозиторий Webmin в список источников:

```
sudo add-apt-repository "deb http://download.webmin.com/download/repository sarge contrib"
```

Шаг 4: Установка Webmin

Теперь, когда репозиторий добавлен, можно установить Webmin.

```
sudo apt update  
sudo apt install webmin -y
```

Шаг 5: Доступ к Webmin

После установки Webmin можно получить доступ к нему через веб-браузер.

Откройте браузер и перейдите по следующему URL: <http://192.168.0.137:10000>

Шаг 6: Вход в Webmin

Введите свои учетные данные:

Используем имя пользователя и пароль нашей системы Ubuntu для входа.

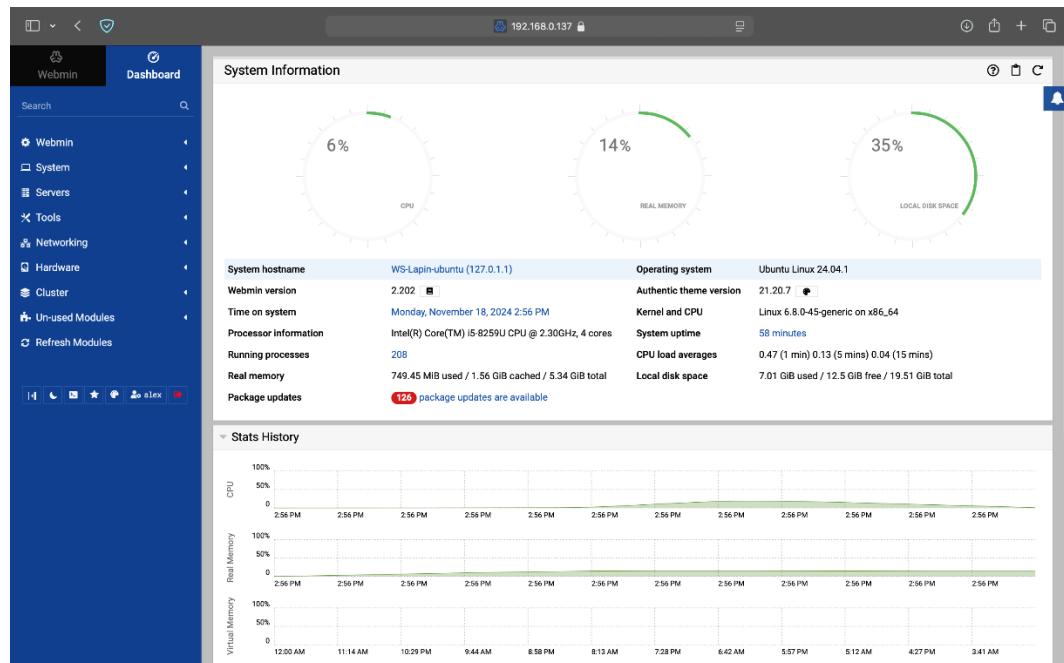


Рисунок 54. Даcборд Webmin

Шаг 7: Изменение системного времени

Чтобы изменить системное время в интерфейсе Webmin:

- Перейдите в Hardware > System Time
- Выберите нужную дату и время
- Нажмите "Apply"

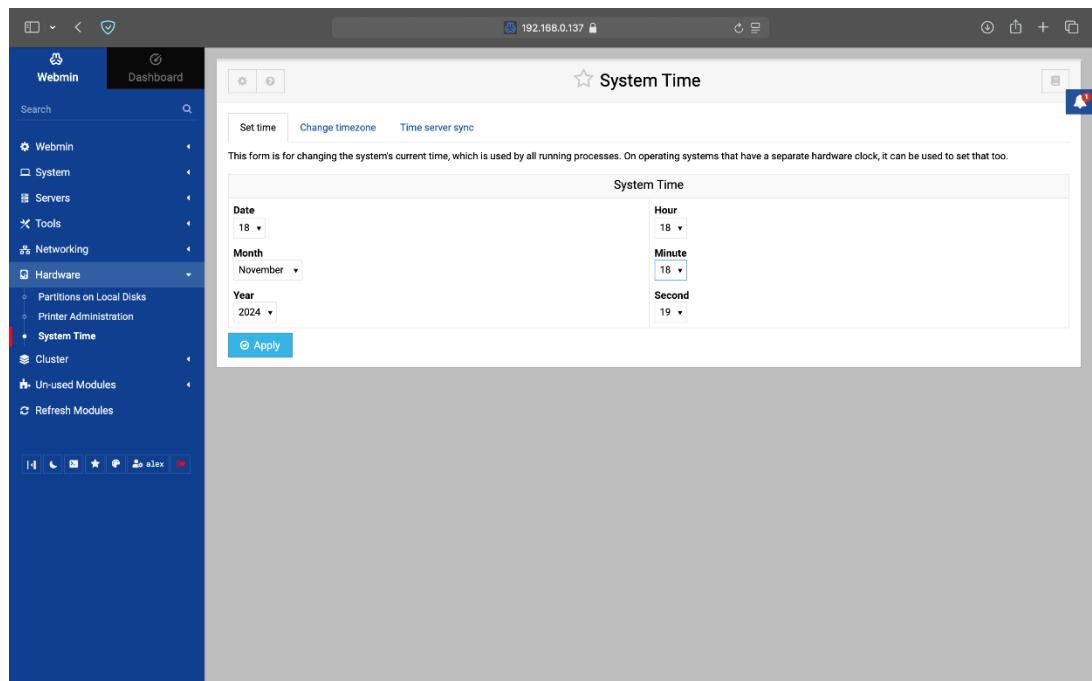


Рисунок 55. Изменение системного времени

Шаг 8: Настройка языка интерфейса

Чтобы настроить русский язык в интерфейсе Webmin:

- Перейдите в Webmin > Change Language and Theme
- Выберите тему и настройте язык на русский.
- Нажмите "Make changes"

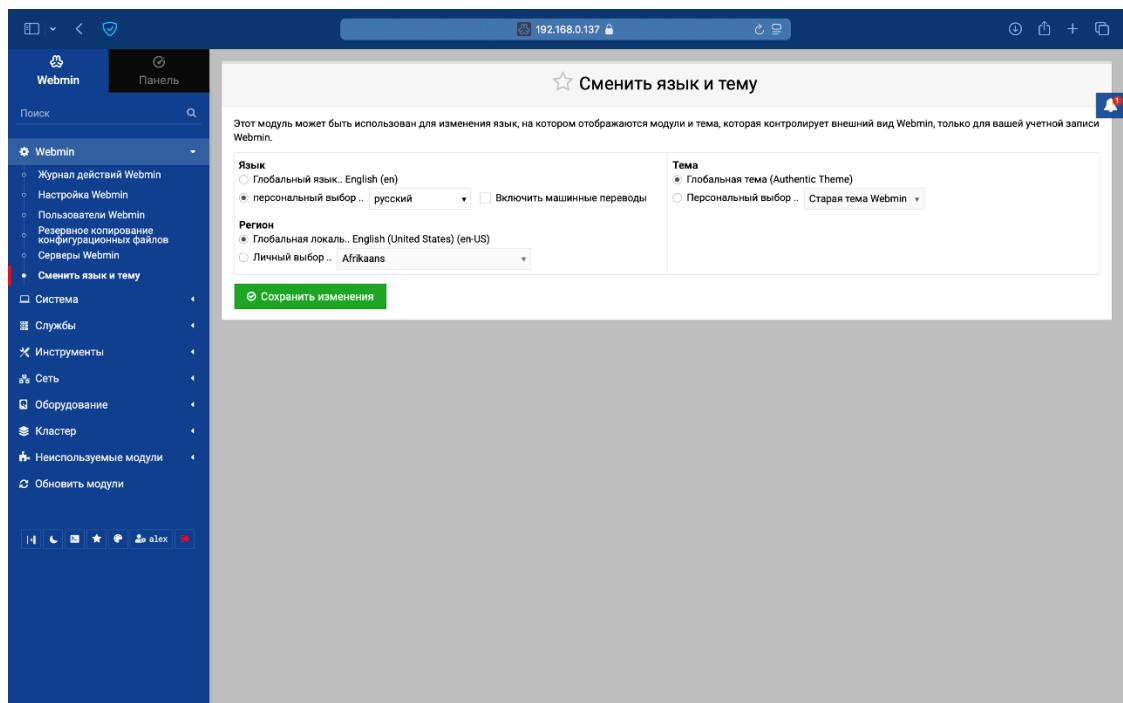


Рисунок 56. Настройка языка интерфейса

Шаг 9: Добавление нового пользователя

1. Перейдите в Webmin > Пользователи Webmin > Добавить пользователя Webmin
2. Введите имя пользователя и пароль
3. Нажмите "Создать"

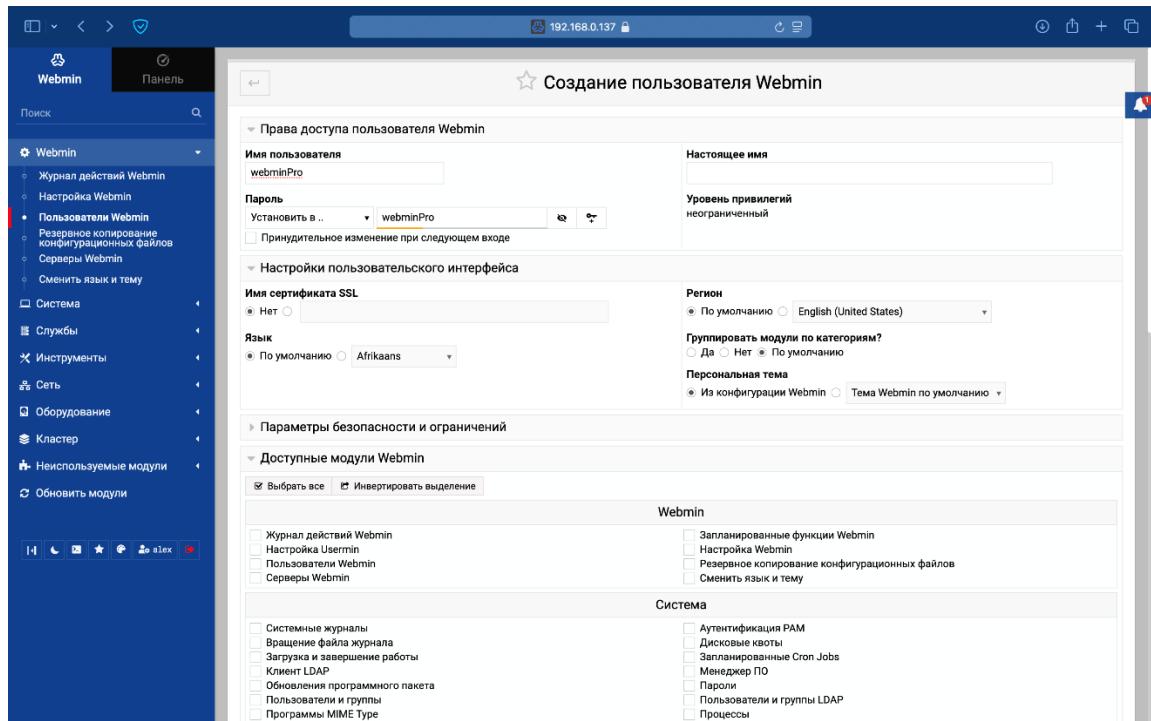


Рисунок 57. Добавление нового пользователя

Шаг 10: Изучение функционала Webmin

Webmin предлагает множество модулей для управления различными аспектами системы, включая:

1. Управление пользователями и группами: создание, удаление, изменение прав доступа и паролей пользователей и групп.
2. Управление файловой системой: просмотр, редактирование, создание и удаление файлов и папок на сервере.
3. Настройка сети: управление сетевыми интерфейсами, настройка IP-адресов, маршрутизации, DNS-серверов и других сетевых параметров.
4. Управление базами данных: создание и удаление баз данных, управление правами доступа и настройками баз данных.
5. Настройка веб-серверов: управление Apache, Nginx и другими веб-серверами, настройка виртуальных хостов, SSL-сертификатов и других параметров.
6. Управление DNS-серверами: создание и удаление DNS-записей, настройка

DNS-серверов.

7. Управление почтовыми серверами: настройка SMTP, POP3, IMAP и других почтовых сервисов, создание и удаление почтовых ящиков.
8. Мониторинг системы: просмотр текущей загрузки сервера, использования ресурсов (памяти, дискового пространства), журналов событий.

Результаты выполнения задания

В результате выполнения задания были достигнуты следующие результаты:

1. Установка Webmin
 - a. Успешно установлен и настроен веб-интерфейс для управления системой
 - b. Установлены все необходимые зависимости
 - c. Официальный репозиторий Webmin добавлен в список источников, что позволяет легко обновлять пакет
2. Настройка системы
 - a. Выполнена локализация интерфейса на русский язык
 - b. Настроено корректное системное время
 - c. Создан дополнительный пользователь
3. Практическая значимость
 - a. Получен опыт работы с системой централизованного управления
 - b. Изучены основные возможности веб-интерфейса администрирования
 - c. Освоены базовые принципы управления Linux-системой через GUI

Установка сервера видео конференций TrueConf Server Free на гостевой ОС - WS_ubuntu

Текст задания

1. Установить и настроить пакет <https://trueconf.ru/products/tcsf/besplatniy-server-videoconferenciy.html>
2. Добавить нового пользователя в систему
3. Изучить функционал системы
4. Провести конференцию

Выполнение задания

Шаг 1. Установка из репозитория

1. Создадим файл /etc/apt/sources.list.d/trueconf.list с записью внутри:

```
printf "deb [arch=$(dpkg --print-architecture)] https://mirror.trueconf.ru/debian $(lsb_release -cs) non-free" > /etc/apt/sources.list.d/trueconf.list
```

2. С помощью учетной записи администратора последовательно выполним команды:

```
wget -O- https://mirror.trueconf.ru/deb.trueconf.pub | gpg --no-default-keyring --keyring gnupg-ring:/etc/apt/trusted.gpg.d/trueconf.gpg --import  
chmod 644 /etc/apt/trusted.gpg.d/trueconf.gpg  
apt-get update  
apt-get -y install trueconf-server
```

Если в ОС нет предустановленной криптографической утилиты gnupg, то появится ошибка вида:

```
E: gnupg, gnupg2 and gnupg1 do not seem to be installed, but one of them is required for this operation
```

В этом случае установите недостающий пакет с помощью команды:

```
sudo apt install gnupg2
```

Шаг 2. Добавление пользователя для доступа к панели управления

В процессе установки появится поле для ввода имён пользователей ОС, которые будут иметь административный доступ к панели управления сервером. Укажите имя пользователя ОС или созданного ранее пользователя.



Рисунок 58. Добавление пользователя для доступа к панели управления

Шаг 3. Проверка установки

В ОС добавится несколько служб TrueConf (в терминологии Linux демонов). При этом две из них — веб-сервер trueconf-web и менеджер работы с БД и файлами настроек trueconf-manager — должны запуститься автоматически после установки. Проверим их статус командами:

```
systemctl status trueconf-web  
systemctl status trueconf-manager
```

При необходимости запустить вручную их можно командами:

```
systemctl start trueconf-web  
systemctl start trueconf-manager
```

Основная служба сервера trueconf не может быть запущена, пока вы его не зарегистрируете. Для этого перейдите к следующему шагу.

Шаг 4. Вход в панель управления

1. Узнать IP-адрес сервера можно командой:

```
ip addr show
```

2. Откройте в браузере адрес сервера в формате <http://192.168.0.156> и перейдите на страницу входа.

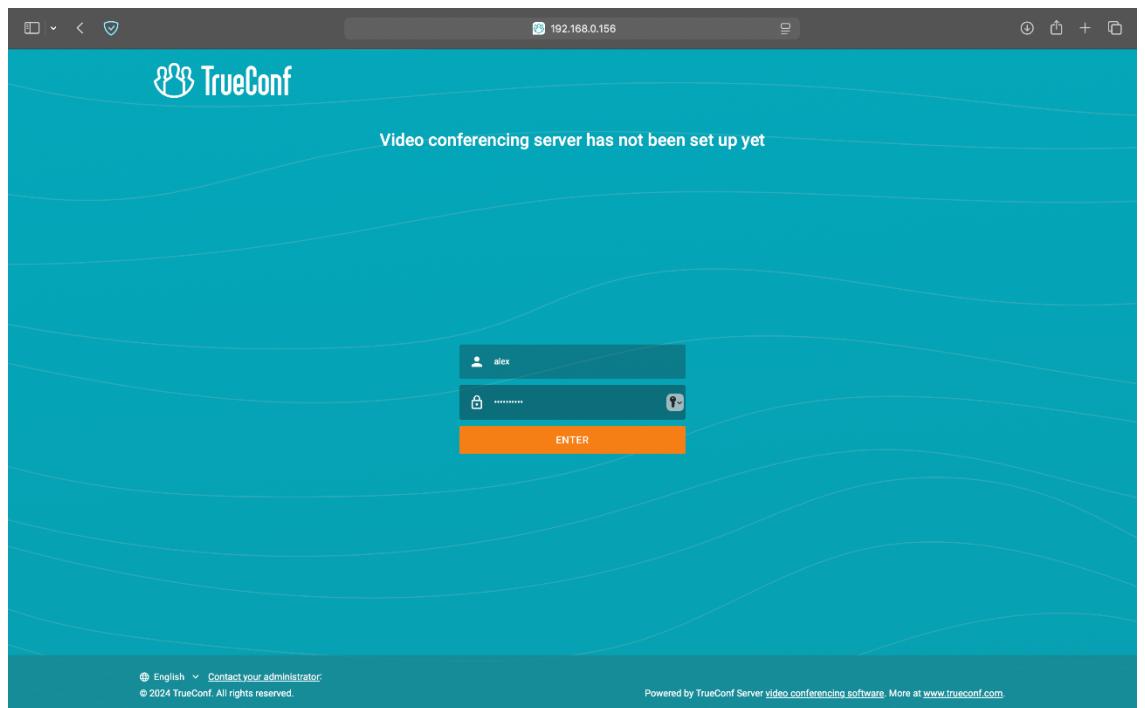


Рисунок 59. Вход в панель управления

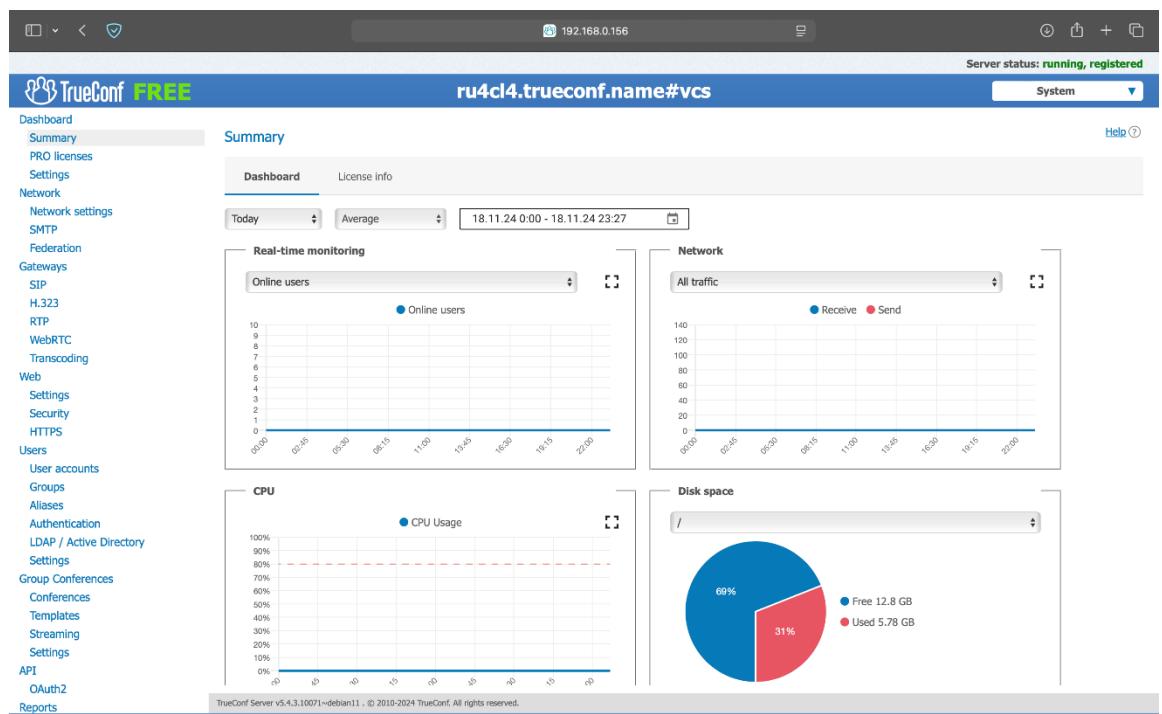


Рисунок 60. Панель управления

Шаг 5. Добавление пользователей

Добавим пользователя в систему. Для этого в панели управления перейдите в раздел Users > User accounts и нажмите кнопку Add user.

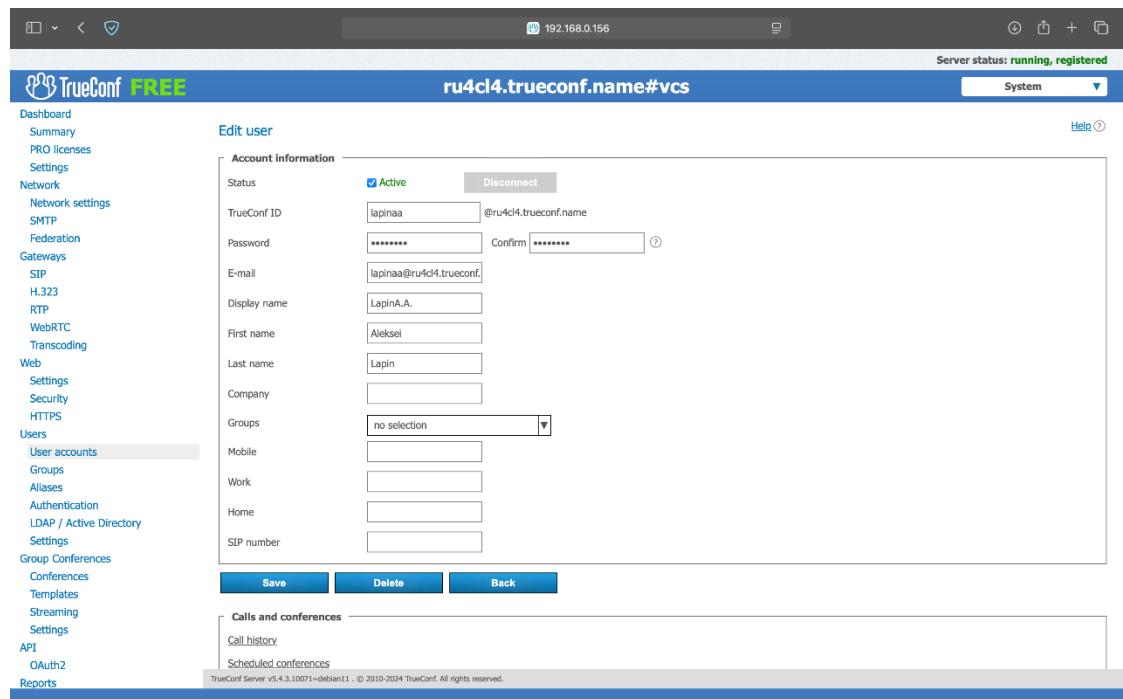


Рисунок 61. Добавление пользователя LapinA.A.

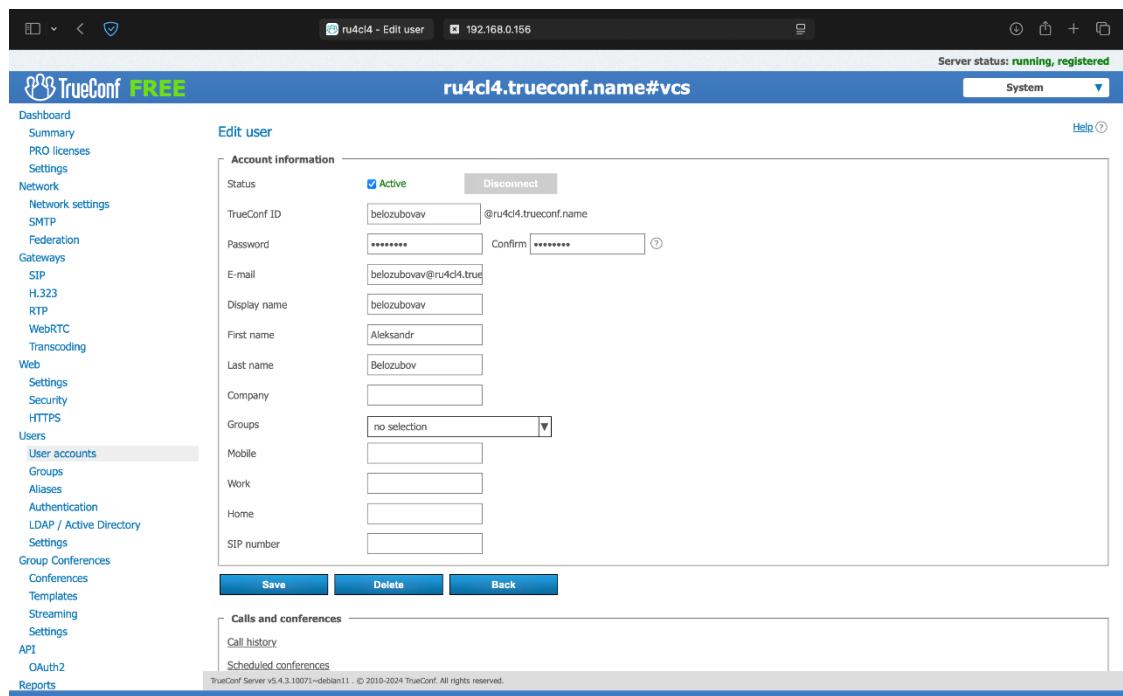


Рисунок 62. Добавление пользователя BelozubovA.V.

Шаг 6. Проведение конференции

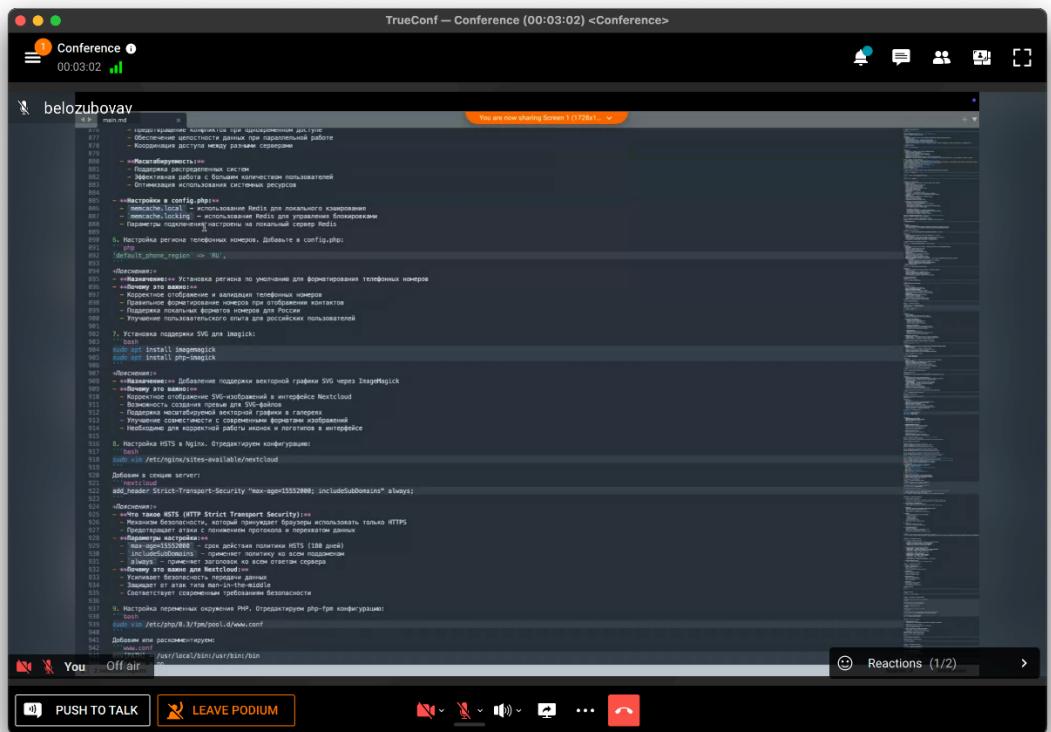


Рисунок 63. Проведение конференции

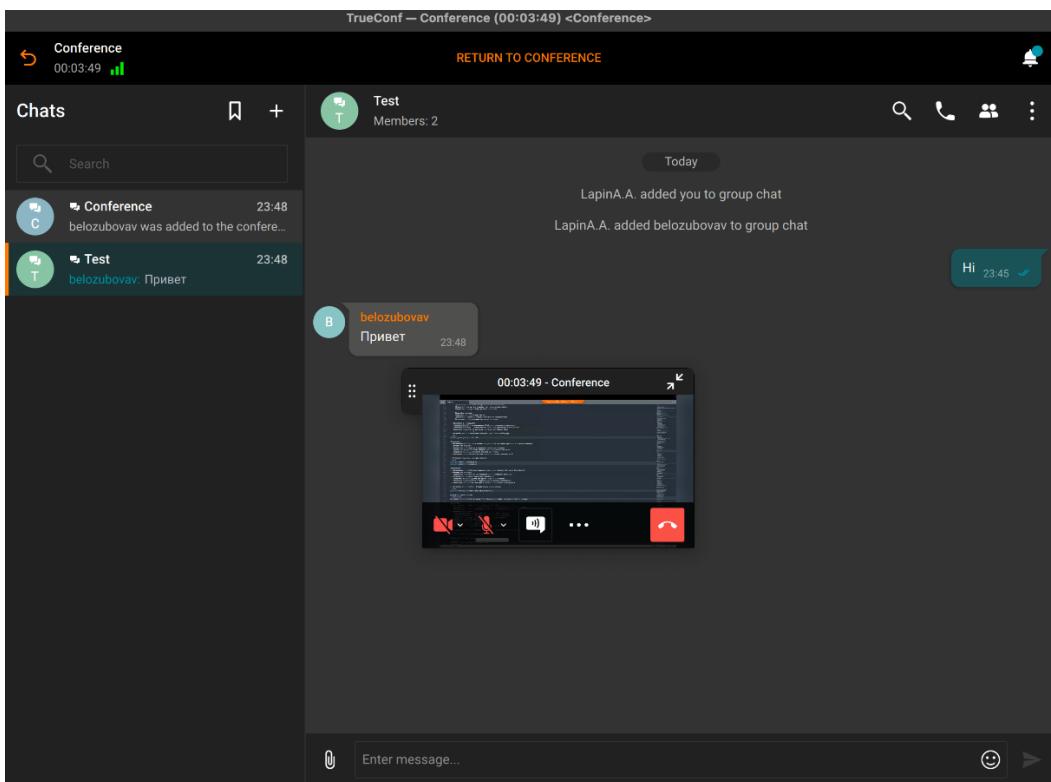


Рисунок 64. Чат в trueconf

Результаты выполнения задания

В ходе выполнения задания были достигнуты следующие результаты:

1. Успешно установлен и настроен TrueConf Server Free на Debian 11:
 - a. Корректно добавлен официальный репозиторий
 - b. Установлены все необходимые компоненты
 - c. Службы успешно запущены и работают
2. Выполнена базовая настройка системы:
 - a. Настроен доступ администратора к панели управления
 - b. Добавлены пользователи для проведения конференций
 - c. Проверена работоспособность веб-интерфейса
3. Протестирован основной функционал системы:
 - a. Проведена тестовая видеоконференция между пользователями
 - b. Проверена работа чата
 - c. Подтверждена стабильность видео, аудио связи и демонстрации экрана

Заключение

В ходе выполнения лабораторной работы были успешно реализованы все поставленные задачи.

Достигнутые результаты:

1. Настройка облачного хранилища NextCloud
 - Успешно развернута полнофункциональная система
 - Настроена безопасная конфигурация с SSL и HSTS
 - Реализован доступ через веб, десктоп и мобильные клиенты
 - Вывод: все задачи выполнены и система готова к эксплуатации
2. Развёртывание Moodle
 - Установлена и настроена система управления обучением
 - Создан тестовый курс с различными типами материалов
 - Настроены роли пользователей и доступы
 - Вывод: все задачи выполнены и платформа полностью готова для организации учебного процесса
3. Установка phpVirtualBox
 - Настроена интеграция с VirtualBox 7.0
 - Настроен веб-интерфейс для управления виртуальными машинами
 - Исправлены известные ошибки в работе phpvirtualbox
 - Создание виртуальных машин работает корректно
 - Не удалось запустить виртуальные машины через веб-интерфейс
 - Вывод: требуется использовать более старые версии Ubuntu и VirtualBox, так как новые несовместимы с phpvirtualbox
4. Настройка Webmin
 - Успешная установка и базовая настройка
 - Локализация и настройка системного времени
 - Добавлен новый пользователь
 - Вывод: все задачи выполнены
5. Развёртывание TrueConf Server
 - Установка и настройка сервера выполнены успешно
 - Проведено тестирование видеоконференций

- Проверена работа всех основных функций
- Вывод: все задачи выполнены, система готова к проведению видеоконференций

Приобретенные навыки:

- Установка и настройка веб-серверов, СУБД и PHP в среде Linux
- Администрирование облачных сервисов
- Работа с системами управления проектами и документооборотом
- Управление пользователями и правами доступа
- Настройка сетевых служб и сервисов

Общее заключение:

- Получен практический опыт по установке и настройке различных серверных систем
- Освоены базовые принципы работы с системами облачного хранения, документооборота и видеоконференций
- Приобретены навыки удаленного администрирования через веб-интерфейс

Литература

1. Xakep.ru. Установка и настройка Nextcloud [Электронный ресурс]. URL: <https://xakep.ru/2020/12/18/nextcloud/> (дата обращения: 19.11.2024).
2. Nextcloud Documentation. Source Installation [Электронный ресурс]. URL: https://docs.nextcloud.com/server/20/admin_manual/installation/source_installation.html (дата обращения: 19.11.2024).
3. Nextcloud Documentation. Nginx Configuration [Электронный ресурс]. URL: https://docs.nextcloud.com/server/20/admin_manual/installation/nginx.html (дата обращения: 19.11.2024).
4. Nextcloud Documentation. Latest Nginx Configuration [Электронный ресурс]. URL: https://docs.nextcloud.com/server/latest/admin_manual/installation/nginx.html (дата обращения: 19.11.2024).
5. YouTube. How to install Moodle on Ubuntu Server [Электронный ресурс]. URL: <https://www.youtube.com/watch?v=xOPTMnB117A> (дата обращения: 19.11.2024).
6. Ubuntu Wiki. Установка и настройка phpvirtualbox [Электронный ресурс]. URL: https://help.ubuntu.ru/wiki/phpvirtualbox#установка_и_настройка_phpvirtualbox (дата обращения: 19.11.2024).
7. GitHub. Официальный репозиторий phpvirtualbox [Электронный ресурс]. URL: <https://github.com/phpvirtualbox/phpvirtualbox> (дата обращения: 19.11.2024).
8. YouTube. Easy Webmin Setup on Ubuntu 22.04: Step-by-Step Guide! [Электронный ресурс]. URL: <https://www.youtube.com/watch?v=v-l7QGuDSac> (дата обращения: 19.11.2024).
9. PQ.Hosting. Зачем нужен Webmin? [Электронный ресурс]. URL: <https://pq.hosting/help/zachem-nuzhen-webmin> (дата обращения: 19.11.2024).
10. TrueConf. Как за 15 минут развернуть систему видеоконференций на базе ОС Linux [Электронный ресурс]. URL: <https://trueconf.ru/blog/baza-znaniy/kak-za-15-minut-razvernut-sistemu-videokonferencij-na-baze-os-linux> (дата обращения: 19.11.2024).