


**Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»  
(ВлГУ)**

УТВЕРЖДАЮ

Заведующий кафедрой ИСПИ

  
И.Е. Жигалов

«20» марта 2025 г.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ  
К ЛАБОРАТОРНЫМ РАБОТАМ  
МЕЖДИСЦИПЛИНАРНОГО КУРСА**

**«НАСТРОЙКА И СОПРОВОЖДЕНИЕ ИНФОРМАЦИОННЫХ РЕСУРСОВ»**

**В РАМКАХ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**

**«ТЕХНИЧЕСКАЯ ПОДДЕРЖКА И АДМИНИСТРИРОВАНИЕ  
ИНФОРМАЦИОННЫХ РЕСУРСОВ»**

09.02.09 Веб-разработка  
Разработчик веб приложений

**Владимир, 2025**

Методические указания к лабораторным работам междисциплинарного курса «Настройка и сопровождение информационных ресурсов» разработал старший преподаватель кафедры ИСПИ Шамышев А.А.


Методические указания к лабораторным работам рассмотрены и одобрены на заседании УМК специальности 09.02.09 Веб-разработка протокол № 1 от «10» марта 2025 г.

Председатель УМК специальности  И.Е. Жигалов

Методические указания к лабораторным работам рассмотрены и одобрены на заседании кафедры ИСПИ протокол № 7а от «12» марта 2025 г.

Рецензент от работодателя:  
руководитель группы обеспечения  
качества программного обеспечения  
ООО «БСЦ МСК»



 С.С. Смирнова

## Введение

Современные информационные системы требуют надежной инфраструктуры, включающей серверное оборудование, программное обеспечение и системы управления. Настройка и сопровождение таких систем является важной задачей специалистов в области информационных технологий.

Дисциплина "Настройка и сопровождение информационных ресурсов" направлена на формирование знаний и практических навыков, необходимых для проектирования, настройки, управления и сопровождения серверных систем. Эти навыки включают анализ оборудования, установку и настройку программного обеспечения, организацию систем резервного копирования, а также работу с инструментами для обработки пользовательских запросов.

Методические рекомендации включают в себя пошаговые инструкции, задания и контрольные вопросы, которые помогут студентам освоить ключевые аспекты дисциплины. Каждая тема сопровождается теоретическими материалами и практическими заданиями, направленными на закрепление изученных концепций.

Эти рекомендации охватывают следующие ключевые области:

- Настройка серверного оборудования и программного обеспечения.
- Организация систем резервного копирования данных.
- Работа с системами управления запросами пользователей.

Данный материал предназначен для студентов технических направлений и помогает развить навыки, необходимые для работы в реальных IT-проектах и инфраструктурных решениях.

## Оглавление

1. Лабораторная работа № 1. Анализ характеристик серверного оборудования .....	5
2. Лабораторная работа № 2. Установка серверного ПО.....	9
3. Лабораторная работа № 3. Настройка системы резервного копирования .....	13
4. Лабораторная работа № 4. Работа с системой обработки запросов пользователей .....	17

**Цель работы:**

Научиться анализировать аппаратные характеристики серверного оборудования и выбирать оптимальную конфигурацию для развертывания корпоративных приложений и информационных ресурсов. Работа нацелена на получение навыков по подбору оборудования для различных сценариев использования.

**Теория.**

Серверное оборудование отличается от настольных ПК более высокой производительностью и устойчивостью к нагрузкам. Физические серверы представляют собой оборудование, которое размещается в локальных дата-центрах или серверных комнатах компании. Они позволяют компаниям полностью контролировать все аспекты настройки и эксплуатации оборудования. Облачные серверы предоставляют инфраструктуру в виде виртуальных машин, развернутых на серверах облачных провайдеров (AWS, Google Cloud, Azure, Yandex Cloud).

Основные компоненты серверов включают:

- **Процессоры (CPU):**
  - Частота и количество ядер процессора напрямую влияют на производительность сервера. Чем больше ядер и выше частота, тем больше одновременных задач сервер может обрабатывать без задержек.
  - Например, процессоры Intel Xeon и AMD EPYC обеспечивают высокую производительность и поддержку виртуализации, что важно для облачных серверов.
- **Оперативная память (ОЗУ):**
  - Объем и скорость памяти критически важны для многозадачных сред. Больше ОЗУ позволяет обрабатывать большее количество запросов одновременно, что повышает стабильность и скорость работы приложений.
  - Для серверов рекомендуется использовать ECC-память (Error-Correcting Code), которая предотвращает ошибки и повышает надежность.
- **Дисковые накопители:**
  - Для облачных серверов используются SSD и HDD накопители. SSD обеспечивают высокую скорость чтения и записи, что особенно важно для

приложений с интенсивным использованием базы данных и высокой загрузкой.

- HDD лучше подходят для долгосрочного хранения больших объемов данных, где скорость доступа не является критически важной.

- **Сетевые интерфейсы:**

- Высокая пропускная способность сетевых интерфейсов (1 Gbps и выше) важна для обеспечения быстрой передачи данных и стабильного подключения.
- Для облачных серверов также важна поддержка VPN и шифрования для безопасного соединения с корпоративной сетью.

- **Операционные системы:**

- **Linux (Ubuntu, CentOS, Debian):** Идеальны для серверов благодаря стабильности, безопасности и гибкости настройки. Чаще всего используются для веб-серверов и баз данных.
- **Windows Server:** Подходит для приложений, разработанных на базе технологий Microsoft (например, ASP.NET, MSSQL). Имеет удобные инструменты для администрирования и управления.

Порядок выполнения работы:

### Этап 1: Определение критериев для выбора сервера

#### 1. Определите ключевые критерии для развертывания серверов:

- Производительность (процессор, ОЗУ).
- Отказоустойчивость (резервное копирование, аварийное восстановление).
- Безопасность (шифрование данных, VPN).
- Стоимость (капитальные затраты для физических серверов, операционные — для облачных).

### Этап 2: Составление списка требований

#### 2. Составление список минимальных и рекомендуемых характеристик:

- Для физических серверов:
  - 4 ядра процессора, 16 ГБ ОЗУ, 2 ТБ HDD (или 512 ГБ SSD).
- Для облачных серверов:
  - 4 ядра процессора, 8-16 ГБ ОЗУ, 100 ГБ SSD.

- Необходимые параметры сетевых интерфейсов:
  - Пропускная способность 1 Gbps и выше.
  - Поддержка VPN для безопасного подключения.

### Этап 3: Подбор оборудования

#### 3. Исследуйте предложения для физических серверов и облачных сервисов:

- Физические серверы: Dell PowerEdge, HPE ProLiant, Lenovo ThinkSystem.
- Облачные платформы: AWS EC2, Google Cloud Platform, Microsoft Azure, Yandex Cloud.

Параметр	Физические серверы	Облачные серверы
Контроль	Полный контроль над оборудованием	Зависимость от провайдера
Гибкость и масштабируемость	Ограниченная	Высокая
Первоначальные затраты	Высокие	Низкие
Эксплуатационные затраты	Высокие (обслуживание и ремонт)	Оплата по использованию
Время развертывания	Долгое	Быстрое
Отказоустойчивость	Зависит от резервирования	Встроенные функции провайдеров
Безопасность	Высокая, если внутренний доступ	Высокая, но зависит от провайдера

таблица 1 - Сравнение физических и облачных серверов

### Этап 4: Сравнение серверов по характеристикам

#### 4. Сравните предложения по следующим параметрам:

- Производительность: процессор, ОЗУ, тип накопителей.
- Стоимость: единовременные и операционные затраты.
- Отказоустойчивость и безопасность.

Платформа/ Сервер	Тип	Ядра	ОЗУ	Накопитель	Цена (в месяц)	Отказоустойчивость	Безопасность
Dell PowerEdge R650	Физический	8	32 ГБ	2 ТБ HDD	\$2500 (единоразово)	Зависит от настройки	Внутренняя
AWS EC2 T3.large	Облачный	2	8 ГБ	100 ГБ SSD	\$75	Автоснапшоты	Шифрование
Google Compute Engine e2- standard-4	Облачный	4	16 ГБ	120 ГБ SSD	\$95	Автоматические копии	Защищенные сети
Yandex Cloud s2.medium	Облачный	2	8 ГБ	100 ГБ SSD	\$70	Автоснапшоты	Шифрование

таблица 2 - Сравнение серверов по характеристикам

На основе проведенного анализа можно сделать следующие выводы:

- Физические серверы лучше подходят для компаний с высокими требованиями к безопасности и полным контролем над данными.
- Облачные серверы обеспечивают гибкость, масштабируемость и низкие начальные затраты, что делает их идеальными для стартапов и проектов с переменными нагрузками.
- Yandex Cloud и AWS EC2 являются отличным выбором для компаний, работающих в России, благодаря низкой стоимости и высокому уровню безопасности.

### Контрольные вопросы

1. В чем преимущества использования физических серверов по сравнению с облачными?
2. Какие критерии наиболее важны при выборе облачного сервера?
3. Как обеспечить безопасность данных на облачных серверах?
4. Чем отличаются IaaS, PaaS и SaaS модели облачных вычислений?
5. Какие факторы влияют на выбор между физическим и облачным сервером?



## Лабораторная работа №2: Установка серверного ПО

### Цель работы

Ознакомление с процессом установки и настройки серверного программного обеспечения на базе операционной системы Ubuntu Server 22.04 LTS. Научиться устанавливать и конфигурировать веб-серверы, системы управления базами данных (СУБД) и обеспечивать их корректную работу.

### Теория

**Ubuntu Server 22.04 LTS** — популярная операционная система Linux, которая используется для управления серверными ресурсами. Она предлагает стабильность, высокий уровень безопасности и широкий выбор инструментов для настройки серверного программного обеспечения.

### Основные компоненты серверного ПО:

Операционная система:

- Ubuntu Server — оптимизирована для серверной работы. Поддерживает долгосрочные обновления (LTS).
- Использует менеджер пакетов **apt** для установки программного обеспечения.

Веб-сервер Apache:

- Один из самых популярных серверов HTTP, предоставляющий стабильную платформу для размещения веб-приложений.

СУБД MySQL:

- Реляционная база данных, широко используемая для веб-приложений. Обеспечивает управление данными через язык SQL.

### Основные команды для работы в Linux:

- **apt**: менеджер пакетов, используемый для установки программного обеспечения.
- **systemctl**: команда для управления службами в системе.
- **nano**: текстовый редактор для изменения конфигурационных файлов.

## Порядок выполнения работы

### Этап 1: Подготовка операционной системы

1. Установка Ubuntu Server:
  - Скачайте ISO-образ Ubuntu Server 22.04 LTS с официального сайта: <https://ubuntu.com/download/server>.
  - Создайте виртуальную машину в VirtualBox или другой платформе виртуализации. Укажите параметры:
    - 2 ядра процессора.
    - 4 ГБ оперативной памяти.
    - 20 ГБ дискового пространства.
2. Установите Ubuntu Server, следуя шагам установщика:
  - Выберите язык системы и раскладку клавиатуры.
  - Настройте сетевые параметры (по умолчанию DHCP).
  - Создайте учетную запись администратора.
3. Обновление системы: После установки войдите в систему под учетной записью администратора и выполните команду для обновления пакетов:

```
sudo apt update && sudo apt upgrade
```

### Этап 2: Установка веб-сервера Apache

1. Установите Apache: Выполните команду:

```
sudo apt install apache2
```

2. Проверьте состояние службы Apache:

```
sudo systemctl status apache2
```

Статус должен быть "active (running)".

3. Проверьте работу сервера, введя IP-адрес сервера в адресной строке браузера.
4. Настройте виртуальные хосты:
  - Создайте новый файл конфигурации в директории `/etc/apache2/sites-available/`.
  - Добавьте информацию о домене, директории и портах.

- Активируйте конфигурацию:  
`sudo systemctl reload apache2`

### Этап 3: Установка СУБД MySQL

1. Установите MySQL Server:

```
sudo apt install mysql-server
```

2. Настройте MySQL:

Запустите интерактивную настройку MySQL:

```
sudo mysql_secure_installation
```

Следуйте инструкциям для установки пароля администратора и отключения анонимного доступа.

3. Создайте базу данных и пользователя:

Подключитесь к MySQL

```
sudo mysql
```

Выполните следующие команды:

```
CREATE DATABASE testdb;  
CREATE USER 'testuser'@'localhost' IDENTIFIED BY 'password';  
GRANT ALL PRIVILEGES ON testdb.* TO 'testuser'@'localhost';  
FLUSH PRIVILEGES;
```

### Варианты заданий

1. Настроить Apache.
2. Настроить HTTPS с использованием самоподписанного сертификата.
3. Установить MySQL.
4. Настроить удаленное подключение к базе данных через SSH.

## Содержание отчета

1. Цель работы.
  2. Теоретическая часть: описание компонентов серверного ПО.
  3. Пошаговая инструкция с командами.
  4. Скриншоты, подтверждающие успешное выполнение задач.
  5. Выводы.
- 

## Контрольные вопросы

1. Какие команды используются для установки Apache в Ubuntu?
  2. Как настроить виртуальный хост в Apache?
  3. Какие шаги необходимо выполнить для защиты MySQL?
  4. Как проверить доступность веб-сервера?
  5. Какие преимущества использования Ubuntu Server?
- 

## Список литературы

1. Документация Ubuntu Server 22.04 LTS:  
<https://ubuntu.com/server/docs>.
2. Документация Apache HTTP Server:  
<https://httpd.apache.org/docs/>.
3. Документация MySQL:  
<https://dev.mysql.com/doc/>.
4. "Операционные системы. Основы UNIX": Вавренюк А.Б., Курышева О.К. — М.: ИНФРА-М, 2022.
5. "Администрирование серверов и сетевых устройств": Буренин П.В., Девянин П.Н. — М.: Знаниум, 2023.

## Лабораторная работа №3: Настройка системы резервного копирования

### Цель работы

Научиться настраивать систему резервного копирования данных, включая автоматизацию создания резервных копий и восстановление данных. Освоить работу с утилитами командной строки Linux для резервного копирования, такими как **rsync**, **tar**, **pg\_dump**, а также с планировщиком задач **cron**.

### Теоретический материал

**Резервное копирование** — это процесс создания копий данных для их защиты от потерь из-за сбоев оборудования, ошибок пользователей или кибератак.

### Виды резервного копирования:

1. **Полное (Full Backup):**
  - Копируется весь объем данных.
  - Используется для базового резервирования, но занимает больше времени и места.
2. **Дифференциальное (Differential Backup):**
  - Копируются только изменения, произошедшие с момента последнего полного резервного копирования.
3. **Инкрементное (Incremental Backup):**
  - Копируются изменения, сделанные с момента последнего любого (полного или инкрементного) резервирования.

### Инструменты резервного копирования в Linux:

- **rsync:** синхронизация данных между директориями и устройствами.
- **tar:** архиватор для создания резервных копий.
- **pg\_dump:** создание резервных копий баз данных PostgreSQL.
- **cron:** планировщик задач для автоматического выполнения заданий по расписанию.

## **Принципы настройки резервного копирования:**

- Выбор места хранения (локальный диск, удаленный сервер, облако).
- Настройка расписания резервного копирования.
- Проверка возможности восстановления данных.

## **Порядок выполнения работы**

### **Этап 1: Подготовка окружения**

1. Создайте рабочую директорию для хранения данных и резервных копий:

```
mkdir ~/data ~/backups
```

2. Заполните директорию ~/data тестовыми файлами:

```
echo "Пример данных 1" > ~/data/file1.txt
```

```
echo "Пример данных 2" > ~/data/file2.txt
```

### **Этап 2: Настройка резервного копирования с помощью rsync**

1. Создайте резервную копию данных с помощью rsync:

```
rsync -av --delete ~/data/ ~/backups/
```

- Параметр -a обеспечивает сохранение прав доступа.
- Параметр -v включает подробный вывод.
- Параметр --delete удаляет файлы в резервной копии, которых больше нет в исходной директории.

2. Проверьте содержимое директории резервных копий:

```
ls ~/backups
```

### **Этап 3: Создание архивной копии с помощью tar**

1. Создайте архив текущей директории data:

```
tar -czvf ~/backups/data_backup.tar.gz ~/data
```

- Параметр -c создаёт архив.
- Параметр -z сжимает данные с использованием gzip.

- Параметр -v включает подробный вывод.
- Параметр -f указывает имя файла архива.

## 2. Проверьте содержимое архива:

```
tar -tzvf ~/backups/data_backup.tar.gz
```

## Этап 4: Резервное копирование базы данных с помощью pg\_dump

### 1. Создайте тестовую базу данных:

```
sudo -u postgres psql -c "CREATE DATABASE testdb;"
```

```
sudo -u postgres psql -c "CREATE TABLE testdb.users (id SERIAL PRIMARY KEY,  
name VARCHAR(50));"
```

```
sudo -u postgres psql -c "INSERT INTO testdb.users (name) VALUES ('John Doe');"
```

### 2. Создайте резервную копию базы данных с помощью pg\_dump:

```
sudo -u postgres pg_dump testdb > ~/backups/testdb_backup.sql
```

### 3. Проверьте содержимое резервной копии:

```
cat ~/backups/testdb_backup.sql
```

## Этап 5: Настройка автоматизации с помощью cron

### 1. Откройте редактор crontab:

```
crontab -e
```

### 2. Добавьте задачу для ежедневного резервного копирования в 2:00 ночи:

```
0 2 * * * rsync -av --delete ~/data/ ~/backups/
```

### 3. Проверьте текущие задачи cron:

```
crontab -l
```

## Варианты заданий

1. Настроить резервное копирование с помощью scp для сохранения данных на удаленном сервере.
2. Использовать rsync для инкрементного резервного копирования с использованием флага --link-dest.
3. Настроить полное резервное копирование базы данных с последующим дифференциальным резервированием.
4. Настроить отправку уведомлений об успешности резервного копирования по email.

## Содержание отчета

1. Цель работы.
2. Теоретическая часть: описание видов резервного копирования и используемых инструментов.
3. Пошаговая инструкция с командами и конфигурациями.
4. Скриншоты, подтверждающие выполнение задач (резервное копирование, восстановление данных).
5. Ответы на контрольные вопросы.
6. Выводы.

## Контрольные вопросы

1. Какие преимущества использования rsync для резервного копирования?
2. Чем отличается полное резервное копирование от инкрементного?
3. Как создать резервную копию базы данных PostgreSQL?
4. Как настроить автоматическое резервное копирование с помощью cron?
5. Какие параметры tar используются для создания архивов?

## Список литературы

1. **Документация rsync:**  
<https://linux.die.net/man/1/rsync>.
2. **Документация tar:**  
<https://www.gnu.org/software/tar/manual/>.
3. **Документация PostgreSQL:**  
<https://www.postgresql.org/docs/>.
4. **Операционные системы. Основы UNIX:** Вавренюк А.Б., Курьшева О.К. — М.: ИНФРА-М, 2022.  
**Администрирование серверов и сетевых устройств:** Буренин П.В., Девянин П.Н. — М.: Знаниум, 2023.



## **Цель работы**

Освоение работы с системами обработки запросов пользователей (Service Desk) для управления заявками, мониторинга задач и предоставления обратной связи. Изучение основных функций, таких как создание, маршрутизация и закрытие запросов.

## **Теоретический материал**

**Системы обработки запросов пользователей (Service Desk)** обеспечивают управление запросами на техническую поддержку, сервисное обслуживание или выполнение задач. Они используются для упрощения взаимодействия между пользователями и IT-подразделением.

### **Ключевые функции Service Desk:**

1. **Создание и регистрация запросов:** Пользователи создают заявки, указывая проблему или задачу.
2. **Маршрутизация запросов:** Система автоматически или вручную направляет запросы к соответствующим специалистам.
3. **Мониторинг выполнения:** Сотрудники и пользователи могут отслеживать статус заявки.
4. **Закрытие запросов:** После выполнения задачи заявка закрывается с указанием результата.
5. **Отчётность:** Формирование отчетов для анализа эффективности работы.

Примеры популярных систем Service Desk:

- **Jira Service Management**
- **GLPI**
- **Freshdesk**
- **OTRS**

## Порядок выполнения работы

### Этап 1: Установка и настройка Service Desk

#### 1. Выбор системы для работы:

- 1) Для данной работы используем бесплатную систему **GLPI**.

#### 2. Установка GLPI:

- 1) Установите веб-сервер Apache, PHP и СУБД MySQL на Ubuntu Server:

```
sudo apt update
```

```
sudo apt install apache2 php libapache2-mod-php php-mysql mysql-server
```

- 2) Скачайте последнюю версию GLPI с официального сайта:

<https://glpi-project.org/>.

- 3) Разместите файлы GLPI в директории /var/www/html/glpi:

```
sudo mv glpi /var/www/html/glpi
```

- 4) Настройте права доступа:

```
sudo chown -R www-data:www-data /var/www/html/glpi
```

- 5) Перейдите в браузере по адресу <http://<IP-адрес-сервера>/glpi> и следуйте инструкциям установщика.

#### 3. Создание базового пользователя:

- 1) В панели администратора создайте учётные записи для администратора и пользователя.

### Этап 2: Создание и регистрация запросов

1. Войдите в систему под учетной записью пользователя.

2. Создайте новый запрос:

- Укажите тему (например, "Проблемы с подключением к интернету").
- Добавьте описание.
- Укажите приоритет (низкий, средний, высокий).

3. Сохраните запрос и перейдите к его просмотру.

### Этап 3: Обработка запросов

1. Войдите под учетной записью администратора.

2. Найдите созданный запрос и назначьте его соответствующему исполнителю.

3. Добавьте комментарий к запросу с пояснениями.
4. Измените статус заявки на **"В работе"**.

#### **Этап 4: Закрытие запросов**

1. После выполнения задачи выполните следующие действия:
  - 1) Укажите, какие действия были предприняты для решения проблемы.
  - 2) Закройте запрос, изменив его статус на **"Завершён"**.
2. Проверьте отчеты в системе, чтобы убедиться, что запрос был закрыт.

#### **Варианты заданий**

1. Настроить автоматическую маршрутизацию заявок на основе ключевых слов в их описании.
2. Добавить новый тип запросов (например, "Обновление ПО").
3. Настроить оповещения для пользователей при изменении статуса их заявки.
4. Составить отчёт по выполненным запросам за неделю.

#### **Содержание отчета**

1. Цель работы.
2. Теоретическая часть: описание системы Service Desk и её функций.
3. Пошаговая инструкция с командами и настройками.
4. Скриншоты, подтверждающие выполнение заданий (создание заявки, назначение, закрытие).
5. Ответы на контрольные вопросы.
6. Выводы.

#### **Контрольные вопросы**

1. Какие основные функции выполняют системы Service Desk?
2. Как создается заявка в GLPI?
3. Как назначить исполнителя для обработки запроса?
4. Какие типы отчетов можно сформировать в системе Service Desk?
5. Какие параметры можно настроить для автоматической маршрутизации запросов?

#### **Список литературы**

1. **Официальная документация GLPI:**  
<https://glpi-project.org/documentation/>.
2. **"Администрирование серверов и сетевых устройств":** Буренин П.В., Девянин П.Н. — М.: Знаниум, 2023.
3. **"Операционные системы. Основы UNIX":** Вавренюк А.Б., Курышева О.К. — М.: ИНФРА-М, 2022.
4. **Документация Apache HTTP Server:**  
<https://httpd.apache.org/docs/>.
5. **Документация MySQL:**  
<https://dev.mysql.com/doc/>.

## СПИСОК ЛИТЕРАТУРЫ

1. Ковган, Н. М. Компьютерные сети учеб. пособие / Н. М. Ковган - Минск : РИПО. URL : <https://www.studentlibrary.ru/book/ISBN9789855033746.html> (дата обращения: 10.03.2025).
2. Чурбанова, О. В. Базы данных и знаний. Проектирование баз данных в Microsoft Access / О. В. Чурбанова, А. Л. Чурбанов - Архангельск : ИД САФУ. URL : <https://www.studentlibrary.ru/book/ISBN9785261010296.html> (дата обращения: 10.03.2025).
3. Администрирование серверов и сетевых устройств: учебное пособие / Буренин П.В., Девянин П.Н. - М.: Знаниум. URL : <https://znanium.com/catalog/product/1910111> (дата обращения: 10.03.2025).
4. Операционные системы. Основы UNIX: Учебное пособие / Вавренюк А.Б., Курышева О.К. - М.: ИНФРА-М. URL : <https://znanium.com/catalog/product/1862859> (дата обращения: 10.03.2025).