

Министерство образования и науки Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО

Кафедра дискретной  
математики и информационных  
технологий

**Настройка серверов в корпоративной сети**  
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студента 4 курса 421 группы  
направления 09.03.01 – Информатика и вычислительная техника  
факультета КНиИТ  
Буравкова Иосифа Сергеевича

Научный руководитель  
доцент, к.ф.-м.н., доцент

\_\_\_\_\_

В.А Поздняков

Заведующий кафедрой  
доцент, к.ф.-м.н., доцент

\_\_\_\_\_

Л.Б. Тяпаев

Саратов 2017

## ВВЕДЕНИЕ

Понятие «сервер» можно интерпретировать по-разному, иногда под ним подразумевают оборудование, а иногда ПО. В обоих случаях основой всегда будет являться соответствующие ПО. Вообще говоря, слово «сервер» имеет тот же корень, что и «сервис». Таким образом, с точки зрения аппаратных средств сервер — это компьютер, который способен оказывать некоторые услуги другим, подсоединенным к нему компьютерам. Подразумевается, что компьютеры каким-то образом связаны с сервером и друг с другом

Так как каждый день создаются по тысячи новых организаций, которым так или иначе взаимодействуют с ИТ-сферой, данные компания будут нуждаться в соответствующем ПО и оборудовании, и как следствие и их настройки, поэтому данная тема является весьма актуальной на данный момент.

Целью бакалаврской работы является изучение и настройка различных серверов, использующихся в корпоративной сети. Для достижения этой цели были сформулированы следующие задачи:

1. Изучить типы корпоративных серверов;
2. Применить возможности языка JAVA для реализации web-сервера;
3. Реализовать основные корпоративные сервера с помощью соответствующих пакетов на базе ОС Linux.

Работа состоит из введения, пяти глав, заключения и списка использованных источников.

Во введении приводятся общие сведения, актуальность темы, цели и задачи работы. В первой главе дается основное понятие Web-сервера, реализован web-сервер на языке JAVA, и сделана настройка связки apache + PHP. Во второй главе описана технология SQL, а также настройка СУБД PostgreSQL. В третьей главе описан принцип работы SSH и показано как его настроить. В четвертой главе осуществлен подробный разбор технологии DNS и произведена настройка DNS-сервера на пакетной базе BIND9. Пятая глава содержит основные принципы работы протокола FTP и производится

настройка FTP-сервера. В заключении сделаны выводы о проделанной работе. Список использованной литературы содержит источники, на которые приводятся ссылки в работе.

## Основное содержание работы

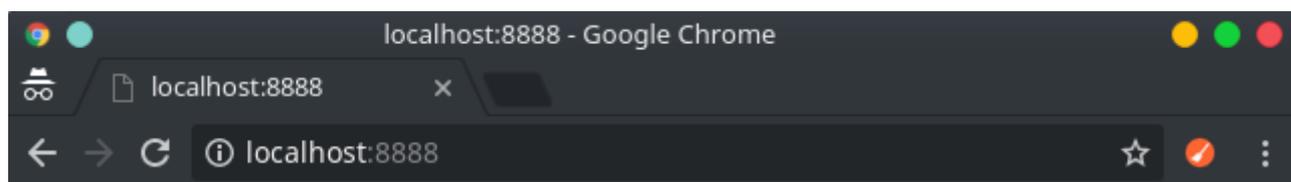
### 1 Web-сервера

Современный сайт представляет собой не просто набор HTML-документов, но и включает в себя множество технологий, в том числе серверных, таких как: SSI (Server-Side Includes, включения на стороне сервера), PHP (PHP: Hypertext Preprocessor, PHP: препроцессор гипертекста), базы данных и многое другое. Для ознакомления и изучения этих технологий ошибкой будет использовать посещаемый рабочий сайт в Интернете, поэтому зачастую необходим установить локальный web сервер и тестировать все на нем.

Так что же называется web-сервером. Веб-сервер – это программа, которая принимает входящие HTTP-запросы, обрабатывает эти запросы, генерирует HTTP-ответ и отправляет его клиенту.

На сегодняшний день существует большое количество различных реализаций веб-серверов. Одним из наиболее популярных и универсальных веб-серверов является веб-сервер с открытым исходным кодом Apache.

Так как одной из задач является реализация web-сервера, то в бакалаврской работе, в приложение А он реализован скриншот работы показан на рисунке 1. Но так как требуется обеспечить безопасность, производительность, то данный сервер не подходит.



# Hello WORLD



Рисунок 1 – HTML страница

Процесс установки и настройки apache будет содержать в себе следующие шаги:

**1. Установка пакета apache и необходимых ему компонентов.**

```
sudo apt update
sudo apt upgrade
sudo apt install apache2 libapache2-mod-fastcgi php7.0
php7.0-fpm phpmyadmin mysql-client mysql-server
```

**2. Настройка основных конфигурационных файлов.**

```
<VirtualHost *:80>
    ServerName 192.168.0.101
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/localhost-error.log
    CustomLog ${APACHE_LOG_DIR}/localhost-access.log
    combined
    # Регулярное выражения для кастомизации php
    ProxyPassMatch ^/(.*\.php(/.*)?)$ unix:/run/php/php7.0-
    fpm.sock|fcgi://localhost/var/www/localhost/html
    # Настройка фаервола защита от xss атак
    <IfModule mod_header.c>
        Header always append X-Frame-Options SAMEORIGIN
        Header set X-Content-Type-Options nosniff
        Header set X-XSS-Protection "1; mode=block"
    </ifModule>
    <Directory /var/www/localhost/html>
        AllowOverride none
        Options +indexes +ExecCGI
        Order deny,allow
        Allow from all
    </Directory>
</VirtualHost>
```

**3. Проверка состояния сервера показана на рисунке 2.**

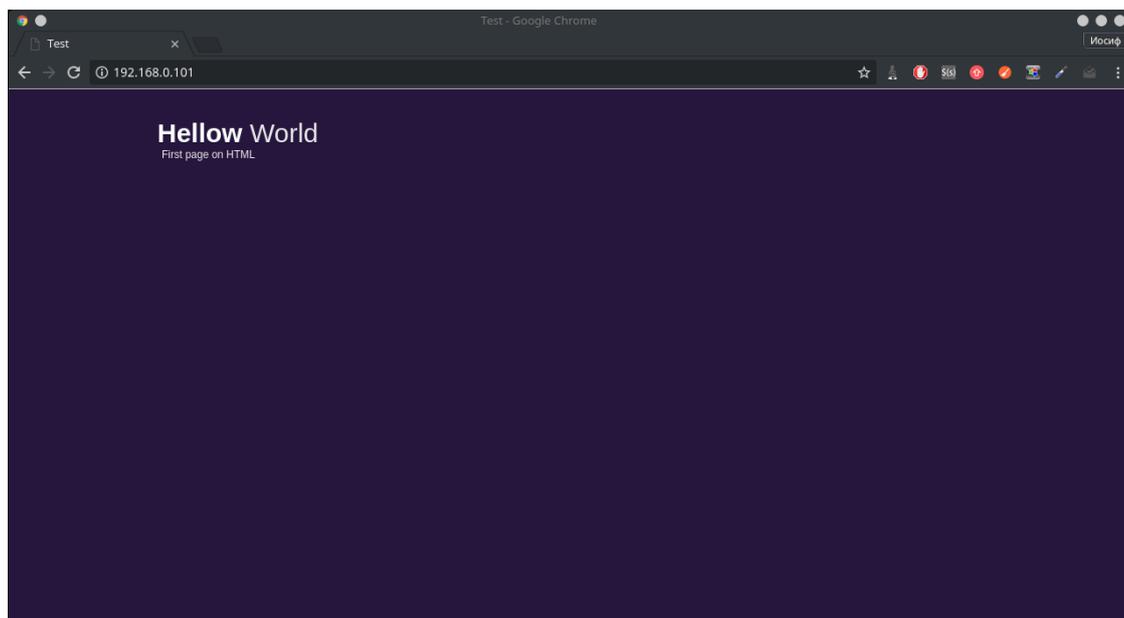


Рисунок 2 – HTML- страница на сервере Apache

## 2 Технология SQL

Взаимодействие с сервером СУБД происходит на языке структурированных запросов SQL (Structured Query Language). Базовый набор языка стандартизован ANSI. Действующая редакция ANSI SQL92. Это не процедурный язык. Он предназначен именно для построения запросов и манипуляции данными и структурами данных. У него нет ни переменных, ни меток, ни циклов, ни всего прочего, с чем привык работать нормальный программист. Надо четко представлять, что SQL оговаривает способ передачи данных в клиентскую программу, но никак не оговаривает то, как эти данные должны в клиентской программе обрабатываться и представляться пользователю.

Существуют специальные процедурные расширения SQL-диалектов. Они похожи на обычные процедурные языки, т.е. у них есть и нормальные переменные, и метки, и циклы, и все прочее, а также полностью поддерживается синтаксис SQL. Жесткого стандарта на процедурные расширения нет, поэтому фирмы-изготовители СУБД определяют синтаксис, так как считают нужным. Опять же существует большое количество

фирменных расширений, в частности Postgress, который позволяет реализовать синхронизацию между серверами.

Практическая часть данной главы будет заключаться, в выше упомянутой синхронизации, для ее реализации потребуется настроить основные конфигурационные файлы среды Postgress, и проверить работоспособность.

Получаем следующие, если создать любой элемент на сервере main он будет моментально скопирован на сервер backup это показано на рисунке 3 и 4.

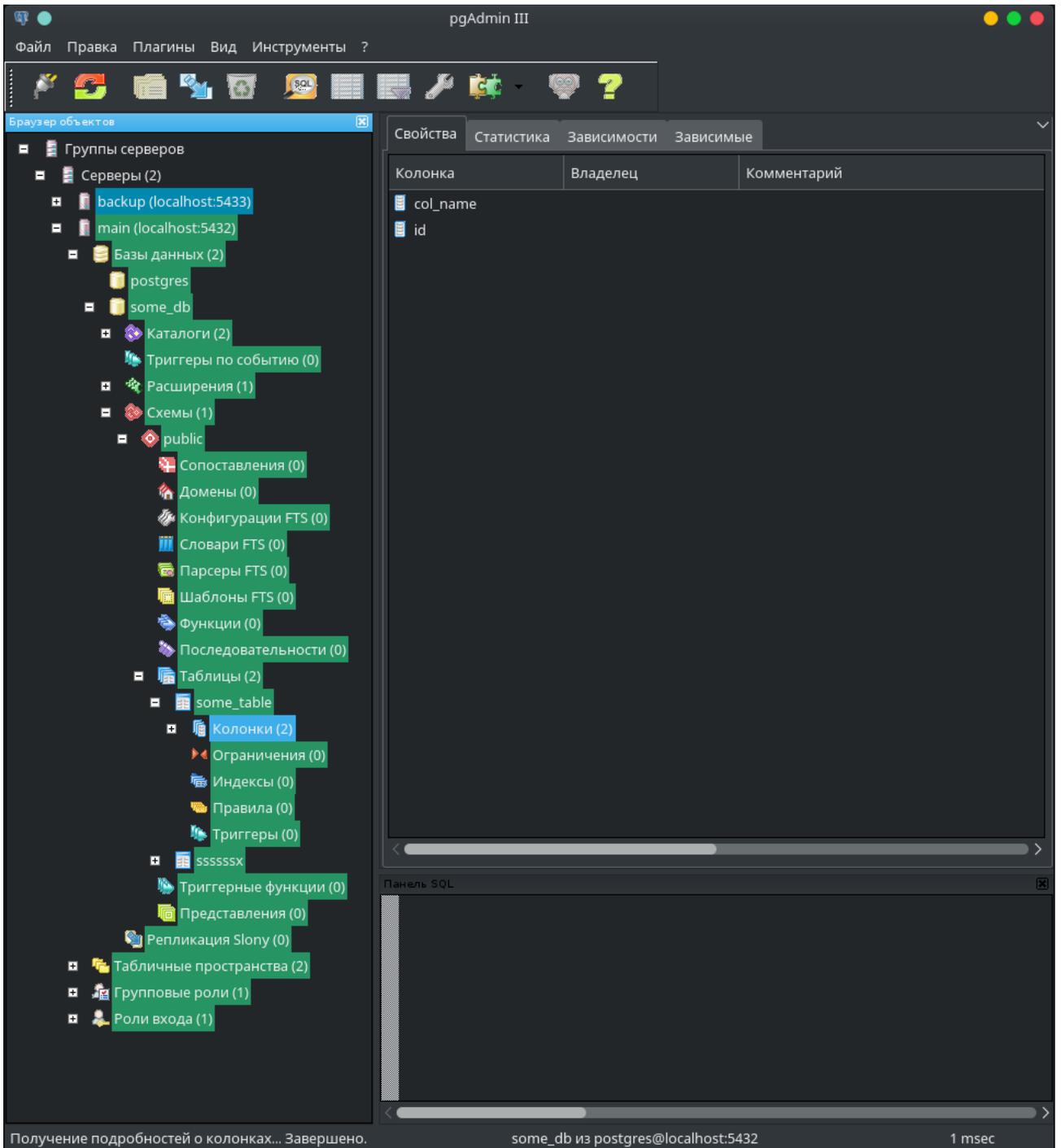


Рисунок 3 – Main сервер

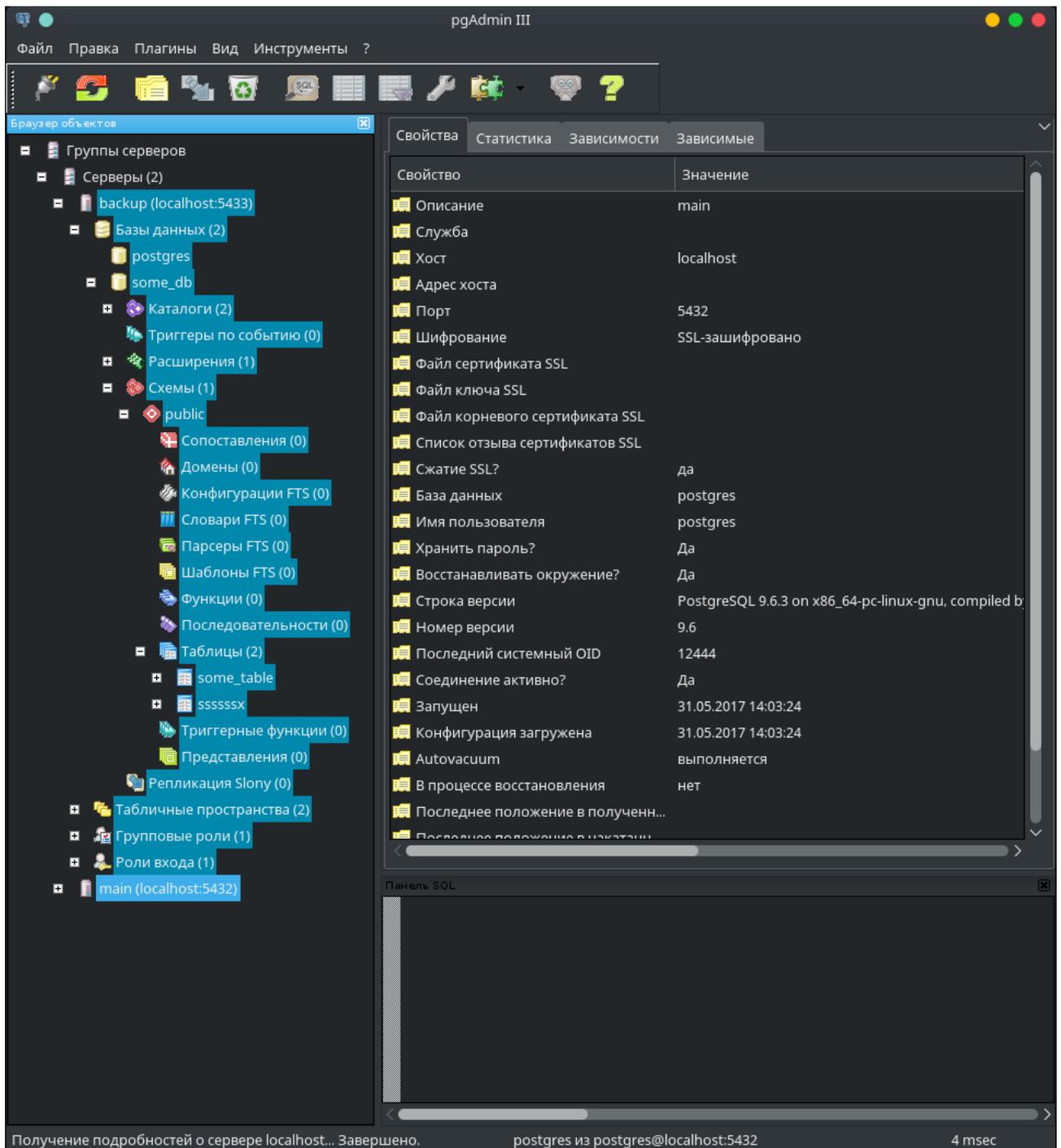


Рисунок 4 – Backup сервер

### 3 Протокол SSH

Протокол Secure Shell (SSH) был разработан для максимальной защиты сетевых взаимодействий с удаленными узлами сети. Этот протокол шифрует сетевой трафик при помощи улучшенных возможностей аутентификации, а также функций, направленной на защиту других незащищенных протоколов – таких как защищенное копирование (Secure Copy, SCP), защищенный

протокол передачи файлов (Secure File Transfer Protocol, SFTP), перенаправление X-сеансов и перенаправление портов. Поддерживается несколько типов шифрования (начиная от 512-битного и заканчивая 32768-битным) с использованием различных криптографических алгоритмов – Blowfish, Triple DES, CAST-128, Advanced Encryption Scheme (AES) и ARCFOUR. Чем выше разрядность шифрования, тем больший объем трафика генерируется в сети.

Независимо от используемой операционной системы SSH предоставляет множество преимуществ при повседневной работе с компьютером. Этот протокол не только надежен, безопасен и гибок, но и прост в установке, настройке и использовании, не говоря уже о его многочисленных возможностях.

Установка SSH сервера, так же, как и любого другого сервера, будет состоять из установки нужного пакета и конфигурирования его настроек под нужды пользователя.

Настроенный конфигурационный файл `sshd_config`, будет выглядеть следующим образом:

```
#/etc/ssh/sshd_config место хранения
Port 22
PermitEmptyPasswords no
ChallengeResponseAuthentication no
UsePAM yes
X11Forwarding yes
PrintMotd no
AcceptEnv LANG LC_*
Subsystem sftp /usr/lib/openssh/sftp-server
```

В качестве результата и проверки работоспособности сервера, удаленно подключимся к серверу и узнаем параметры ПК, которые показаны на рисунке 5.

```
joseph@PC:~$ neofetch
      ,-/+oossssoo+/-,
      `:+ssssssssssssssssss+:`
      -+ssssssssssssssssssyyssss+-
      .osssssssssssssssssdMMMMNyssso.
      /ssssssssssshdmmNNmyNMMMMhssssss/
      +ssssssssshmydMMMMMMNdddyssssssss+
      /ssssssssshNMMMyhhyyyhmNMMNhsssssss/
      .ssssssssdMMMMNhssssssssshNMMMdssssss.
      +ssshhhyNMMNysssssssssyNMMMyssssss+
      ossyNMMNyMMhssssssssshmmhssssssso
      ossyNMMNyMMhssssssssshmmhssssssso
      +ssshhhyNMMNysssssssssyNMMMyssssss+
      .ssssssssdMMMMNhssssssssshNMMMdssssss.
      /ssssssssshNMMMyhhyyyhdNMMNhsssssss/
      +ssssssssdmydMMMMMMNdddyssssssss+
      /ssssssssshdmNNNmyNMMMMhssssss/
      .osssssssssssssssssdMMMMNyssso.
      -+ssssssssssssssssyyssss+-
      `:+ssssssssssssssss+:`
      ,-/+oossssoo+/-,

joseph@PC
-----
OS: Ubuntu 17.04 x8
Kernel: 4.10.0-21-g
Uptime: 6 mins
Packages: 2278
Shell: bash 4.4.7
CPU: Intel i7-3770
GPU: NVIDIA GeForce
Memory: 2102MiB / 7

joseph@PC:~$ █
```

ESC / | - HOME ↑ END PGUP FN  
TAB CTRL ALT ← ↓ → PGDN

Рисунок 5 – Удаленное подключение по SSH

#### 4 DNS - Доменная система имен

Доменная система имен (Domain Name System, DNS) — это распределенная база данных, которая содержит информацию о компьютерах, включенных в сеть Internet. Чаще всего информация включает имя машины и IP-адрес.

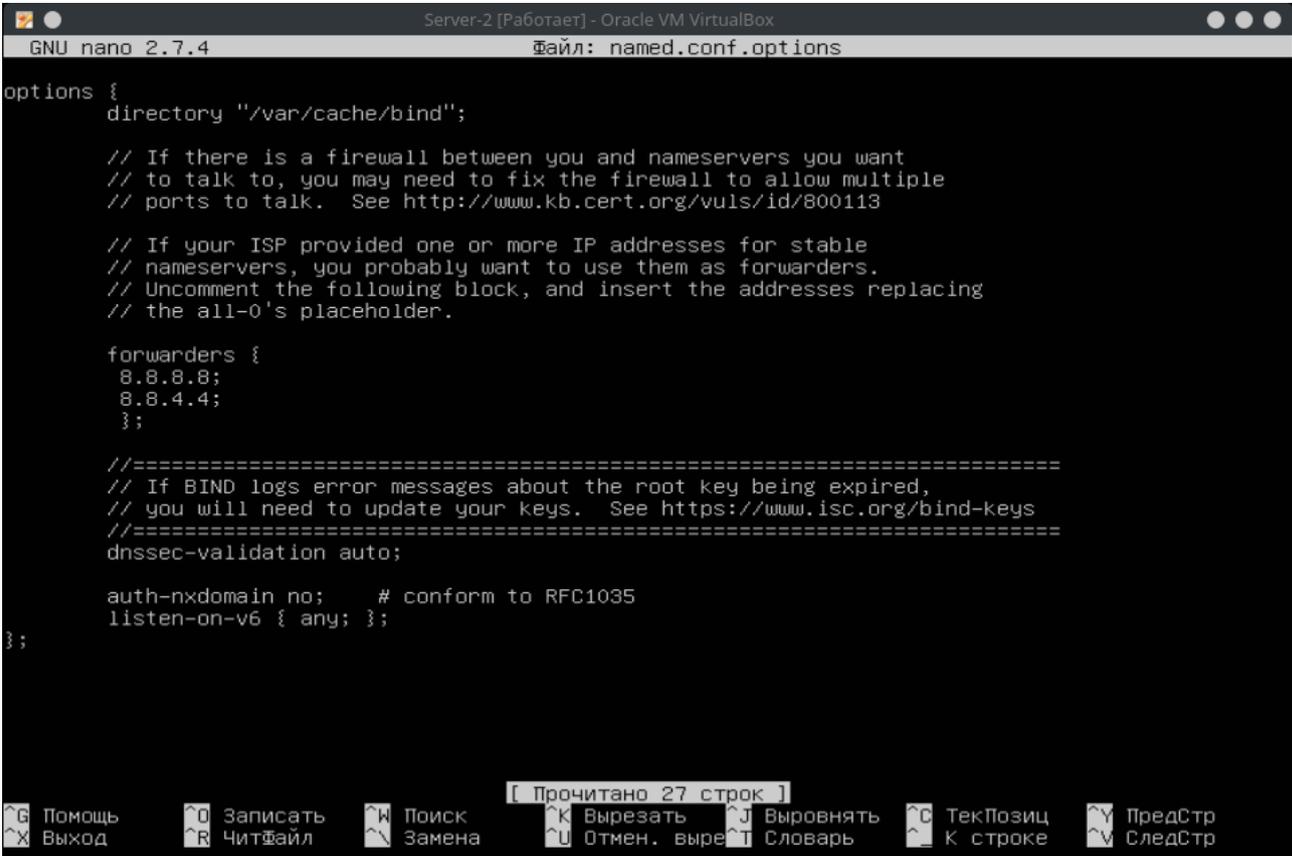
Обращения к хостам в сети Internet используются 32-разрядные IP-адреса, однозначно идентифицирующие любой сетевой компьютер в этой сети. Однако для пользователей применение IP-адресов при обращении к хостам не удобно. Поэтому была создана система преобразования имен, позволяющая компьютеру в случае отсутствия у него информации о соответствии имен и IP-адресов получить необходимые сведения от DNS-сервера, ip-адрес которого хранится в настройках подключения к Internet.

Основная задача DNS — преобразование имен компьютеров в IP-адреса и наоборот.

Для реализации системы DNS был создан специальный сетевой протокол DNS. В сети имеются специальные выделенные информационно-поисковые серверы - DNS-серверы.

Развертка сервера будет производиться на ОС Ubuntu server, то можно воспользоваться встроенным в нее пакетом bind9 для развертывания DNS.

Сначала требуется настроить dns-forwarding, если домашний dns сервер не будет знать по какому ip адресу находится домен, он обратится к другим dns серверам, как на рисунке 6.



```
Server-2 [Работает] - Oracle VM VirtualBox
GNU nano 2.7.4                               Файл: named.conf.options
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    forwarders {
        8.8.8.8;
        8.8.4.4;
    };

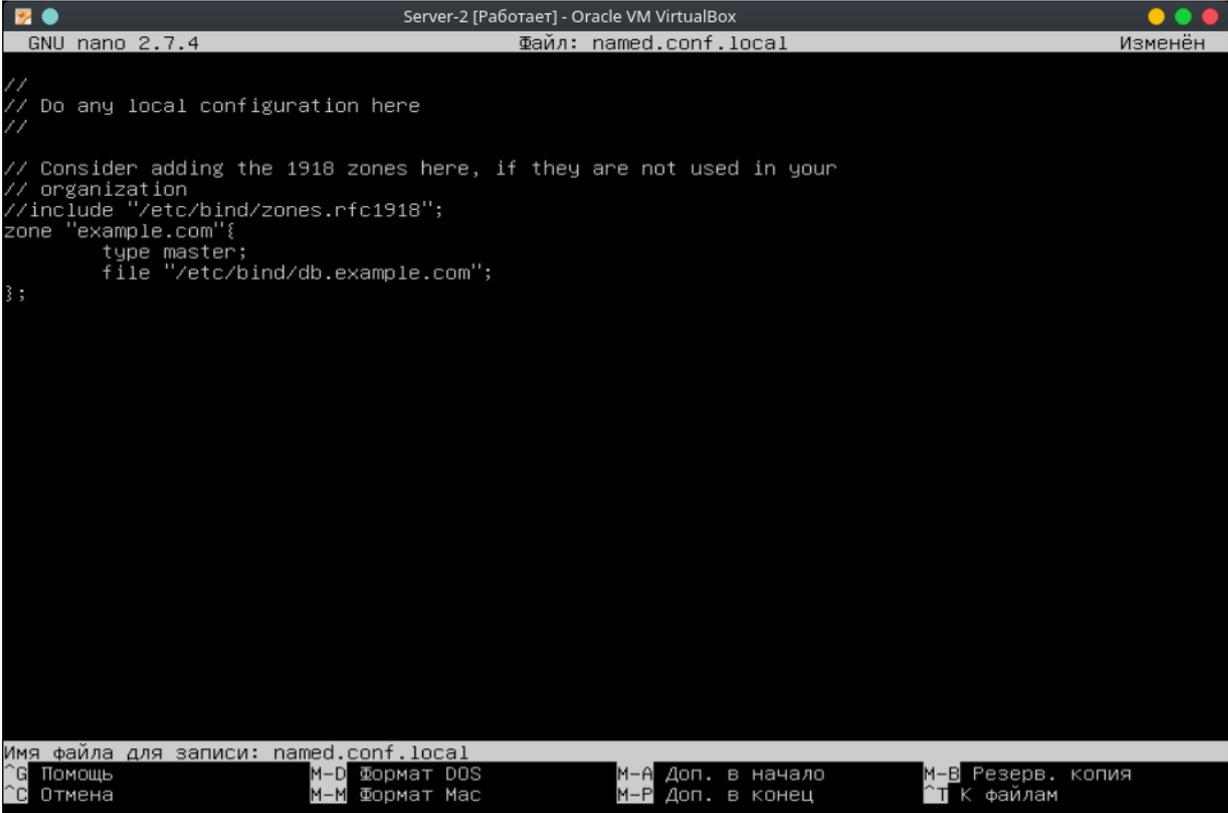
    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.  See https://www.isc.org/bind-keys
    //=====
    dnssec-validation auto;

    auth-nxdomain no;      # conform to RFC1035
    listen-on-v6 { any; };
};

[ Прочитано 27 строк ]
^G Помощь      ^O Записать   ^W Поиск      ^K Вырезать   ^J Выровнять  ^C ТекПозиц   ^Y ПредСтр
^X Выход       ^R ЧитФайл   ^M Замена     ^U Отмен. выр  ^T Словарь   ^G К строке  ^V СледСтр
```

Рисунок 6 – Конфигурационный файл name.conf.options

Далее требуется приписать DNS-зону



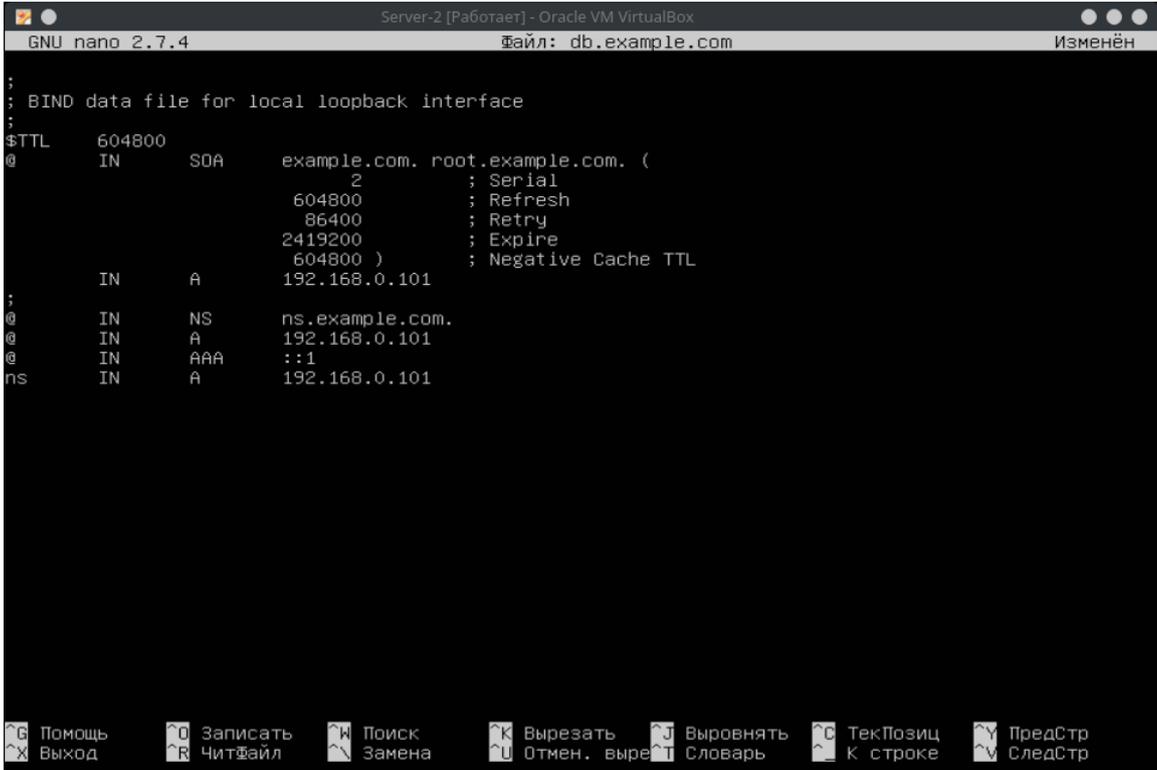
The screenshot shows a terminal window titled "Server-2 [Работает] - Oracle VM VirtualBox" running GNU nano 2.7.4. The file being edited is named.conf.local. The content of the file is as follows:

```
//  
// Do any local configuration here  
//  
  
// Consider adding the 1918 zones here, if they are not used in your  
// organization  
//include "/etc/bind/zones.rfc1918";  
zone "example.com"{  
    type master;  
    file "/etc/bind/db.example.com";  
};
```

The bottom of the window shows the nano editor's command palette with options like "Имя файла для записи: named.conf.local", "Помощь", "Отмена", "Формат DOS", "Формат Mac", "Доп. в начало", "Доп. в конец", "Резерв. копия", and "К файлам".

Рисунок 7 – Конфигурационный файл name.conf.local

Создать запись в конфигурационном файле содержащий в себе информацию о домене, к которому будет идти обращение.



The screenshot shows a terminal window titled "Server-2 [Работает] - Oracle VM VirtualBox" running GNU nano 2.7.4. The file being edited is db.example.com. The content of the file is as follows:

```
;  
; BIND data file for local loopback interface  
;  
$TTL 604800  
@ IN SOA example.com. root.example.com. (  
    2 ; Serial  
    604800 ; Refresh  
    86400 ; Retry  
    2419200 ; Expire  
    604800 ) ; Negative Cache TTL  
    IN A 192.168.0.101  
;  
@ IN NS ns.example.com.  
@ IN A 192.168.0.101  
@ IN AAAA ::1  
ns IN A 192.168.0.101
```

The bottom of the window shows the nano editor's command palette with options like "Помощь", "Выход", "Записать", "ЧитФайл", "Поиск", "Замена", "Вырезать", "Отмен. выре", "Выводить", "Словарь", "ТекПозиц", "К строке", "ПредСтр", and "СледСтр".

## Рисунок 8 – Конфигурационный файл db.example.com

В качестве результата обратившись к домену example.com должен вернуться его адрес, что и видно на рисунке 9.

```
server1@pc:/etc/bind$ sudo nslookup
> server 192.168.0.101
Default server: 192.168.0.101
Address: 192.168.0.101#53
> example.com
Server:          192.168.0.101
Address:         192.168.0.101#53

Name:   example.com
Address: 192.168.0.101
> exit
```

Рисунок 9 – проверка работоспособности DNS

## 5 FTP

FTP (с англ. File Transfer Protocol - «протокол передачи файлов») - протокол, с помощью которого осуществляется передача данных в интернете.

Протокол FTP был разработан для следующих целей:

- для поддержки совместного использования файлов (компьютерные программы и/или данные);
- для поддержки неявного (используя программы) использования удаленных компьютеров;
- для актуализации версий программ на серверах файлов;
- для точной и эффективной передачи файлов.

Устройство FTP, представлено на рисунке 10.

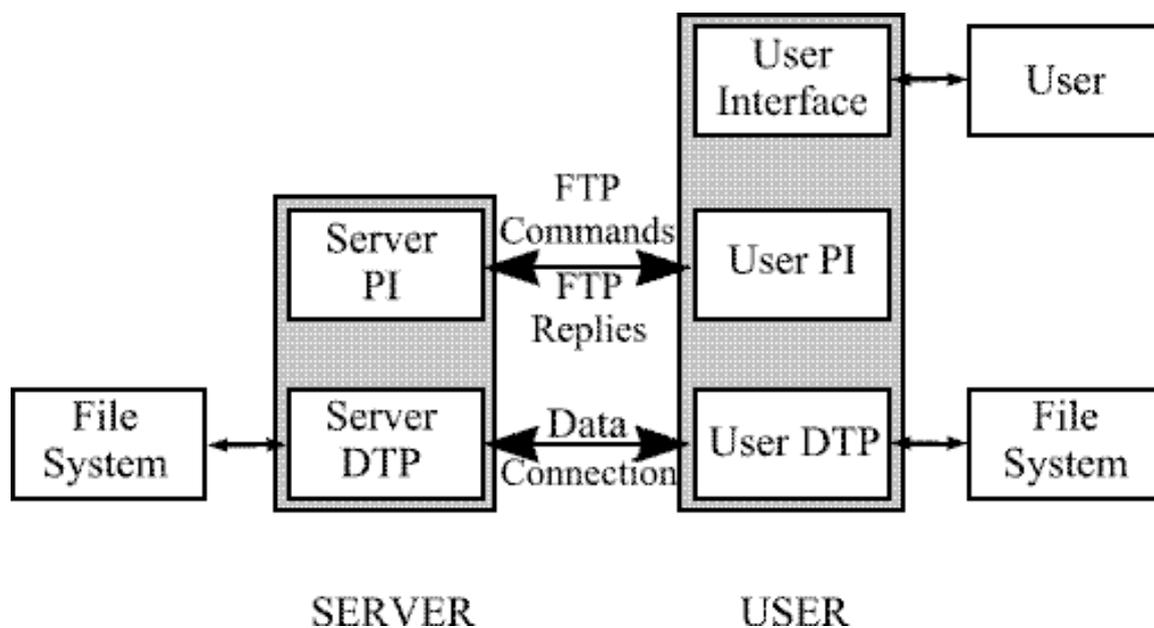


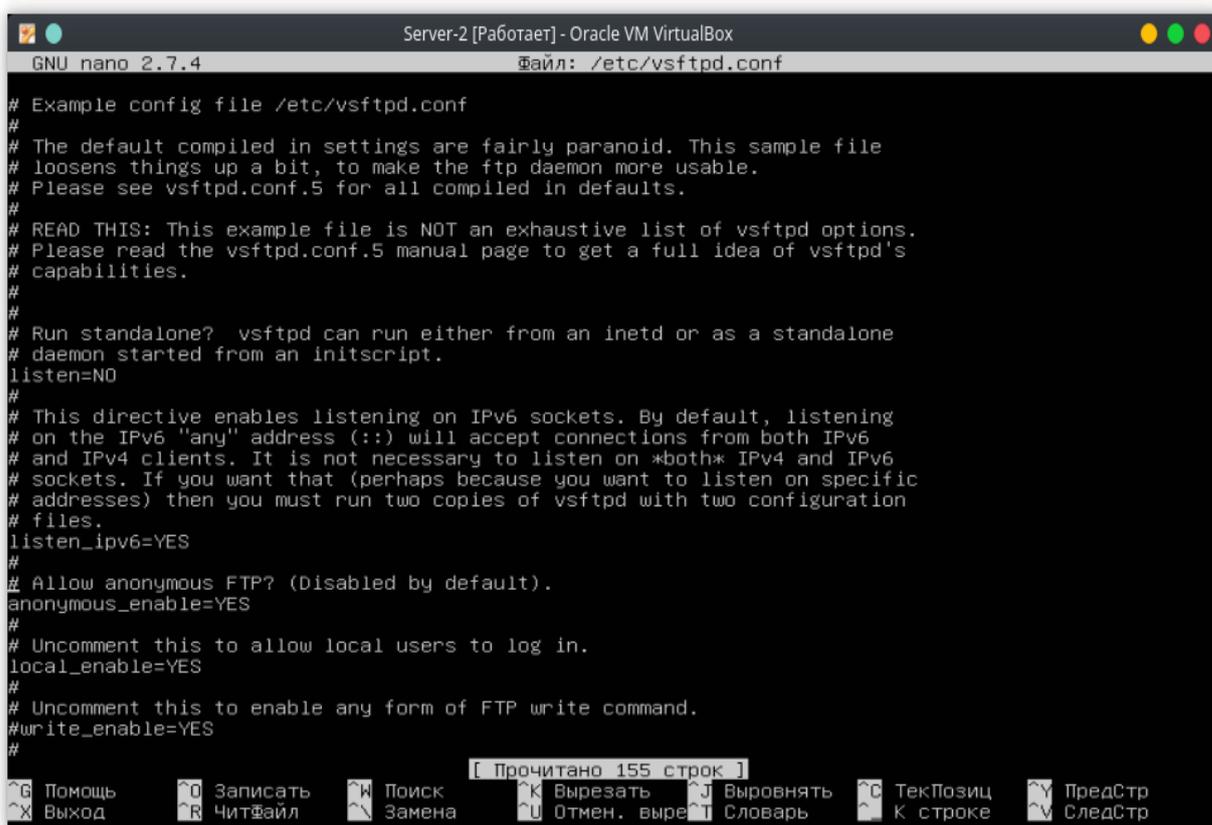
Рисунок 10 – Устройство FTP

- PI - интерпретатор протокола (Protocol Interpreter). Протокол со сторон пользователя и сервера реализован по-разному в User-PI и Server-PI.
- Server-PI - протокол сервера ожидает на порту X соединения с User-PI и устанавливает соединение. Он принимает стандартные FTP команды от User-PI, посылает ответы и управляет Server-DTP-ом.
- User-PI - интерпретатор протокола пользователя является инициатором установления соединения со своего порта Y с процессом Server-FTP, высылает FTP-команды и управляет User-DTP, если этот процесс появится в процессе работы.
- Server-DTP - процесс передачи данных (Data Transfer Process), в своем нормальном "активном" состоянии устанавливает соединение "слушая" порт данных. Он устанавливает параметры для передачи и приема и передает данные с командами из Server-PI.
- User-DTP - процесс передачи данных "слушает" порт данных для соединения с процессом Server-FTP. Когда два сервера передают данные между собой, User-DTP находится в неактивном состоянии.

Команды FTP определяют параметры для DATA-соединения (порт данных, вид передачи, тип представления данных и структуру) и сущность операции (сохранение, поиск, добавление, удаление и т.д.). User-DTP должен

"слушать" определенный порт данных, и сервер определит data connection и data-передатчик (data-transfier) в соответствии с определенными параметрами. Следует отметить, что порт данных (data port) не обязательно должен находиться на том самом хосте, что передает команды через управляющее соединение (control connection), однако пользователь или процесс FTP-пользователя должен обеспечивать "слушание" на определенном порте данных. Необходимо также отметить, что data соединение (data connection) может использоваться для одновременной передачи и приема.

Что бы развернуть FTP-сервер потребуется, поставить и настроить необходимое для его работы пакеты. В бакалаврской работе описано и показана установка и настройка одного из популярных и часто используемых решений vsftpd.



```
Server-2 [Работает] - Oracle VM VirtualBox
GNU nano 2.7.4                               файл: /etc/vsftpd.conf
# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
[ Прочитано 155 строк ]
^G Помощь      ^O Записать    ^W Поиск      ^K Вырезать   ^J Выровнять
^X Выход       ^R ЧитФайл    ^M Замена     ^U Отмен. выр ^I Словарь
^C ТекПозиц   ^N ПредСтр
^K К строке  ^V СледСтр
```

Рисунок 11 – конфигурационный файл FTP

Теперь можно посмотреть результат настройки в виде рабочего FTP-сервера  
на рисунке 12.

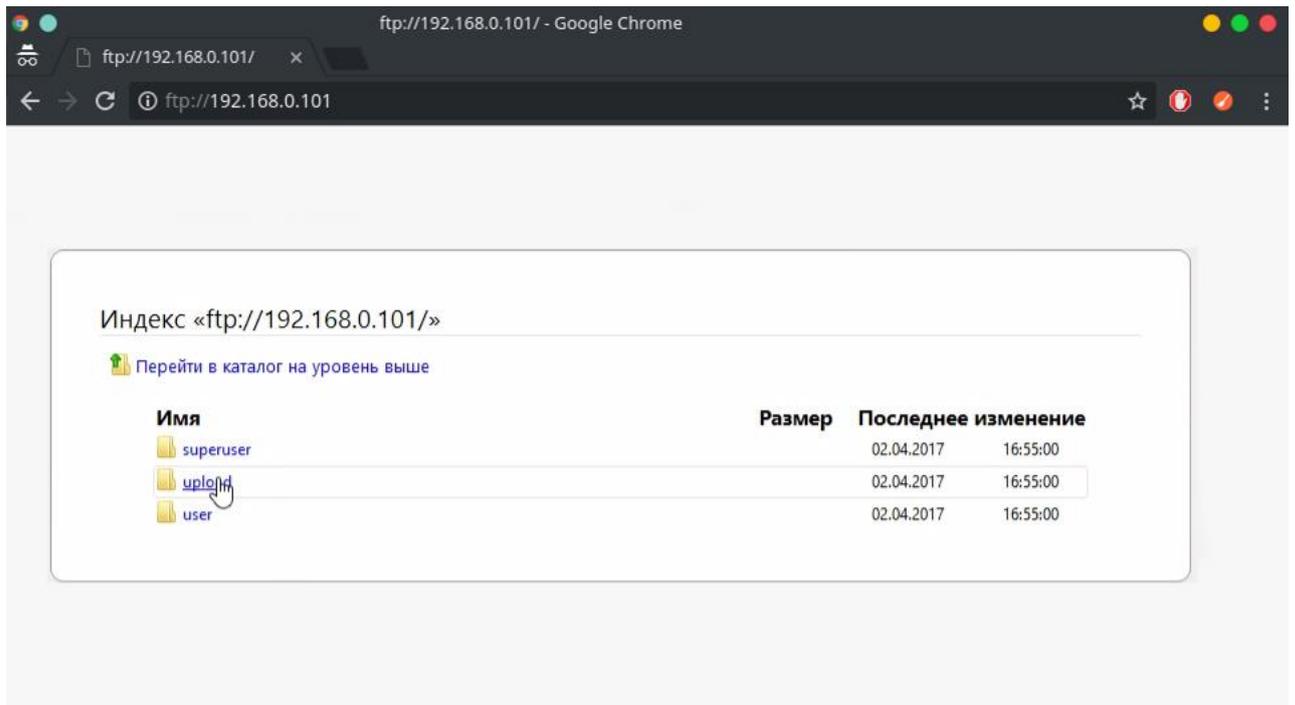


Рисунок 12 – рабочий FTP

## **ЗАКЛЮЧЕНИЕ**

В ходе бакалаврской работы была изучена специальная литература, описаны основные понятия, использующиеся в данной работе. В соответствии с планом, разработана и написана программа в среде IntelliJ IDEA Community Edition 2017.1, на языке JAVA, в которой реализован web сервер. Также был настроен отдельный web-сервер, базирующийся на пакете apache, установлены и настроены SQL, SSH, FTP, DNS сервера. Таким образом, все поставленные задачи выполнены, цель практики достигнута.