**Лабораторная работа 4.**

**Хостинг.**

**Цель работы:** Знакомство с возможностями размещения собственного сайта на сторонних ресурсах – хостингах. Настройка подключения и переноса данных на удаленную систему.

**Теоретический материал:**

Хостинг (hosting) — это услуга, предоставляемая компаниями хостинг-провайдерами (хостерами), которая заключается в размещении информационных ресурсов пользователей на их (компаний) серверах. Функциональность сервера зависит от установленного программного обеспечения, которое, в свою очередь, зависит от операционной системы. Хостинги отличаются наличием тех или иных служб, возможностью поддержки различных технологий, языков программирования и скриптов. Также различаются по количественным и качественным ограничениям (дисковое пространство, количество сайтов, пропускная способность каналов и прочее). Доступ пользователя к своим данным реализован через личный кабинет. Обработка запросов к информационным ресурсам заказчика ведётся в режиме реального времени 24/7. В услугу хостинга входит техподдержка, обеспечение безопасности и резервное копирование данных.

Информационные ресурсы (здесь) — это сайты различного функционала: интернет-магазины, корпоративные сайты, интернет-порталы, информационно-новостные сайты, веб-сервисы. Они состоят из обилия цифровых данных:

* текстовые, графические, фото, видео файлы
* файлы кода
* системы управления базами данных
* CRM-системы (программное обеспечение для управления взаимоотношениями с клиентами, поставщиками и работниками внутри компании)
* CMS-системы, предназначенные для управления контентом сайта (WordPress, Joomla, Drupal, MODX, OpenСart)
* Приложения

**Что такое выделенный сервер**

Начать стоит именно с него, потому что остальные хост-услуги являются виртуальными аналогами выделенного сервера.

Выделенный сервер (dedicated server) — это реальный физический сервер (bare-metal server), который хостинг-компания предоставляет клиенту в аренду. Важно то, что хостер выделяет сервер клиенту целиком в нераздельное пользование, то есть все ресурсы сервера, всё его «железо» используются только одним арендатором. Есть возможность гибкой настройки системы и оптимизации её «под себя», включая установку ПО. По причине полного владения выделенный сервер — самая дорогая хостинг-услуга.

Это мощный компьютер, установленный в стойку с другими подобными серверами и коммутатором. Серверные стойки локализуются в хостерских ЦОДах или индивидуальных дата-центрах (то есть принадлежащих самой компании для её бизнес-целей; в этом случае речь не идёт о выделенных серверах, поскольку их никто не выделяет в аренду — машины являются собственностью компании и называются просто серверами).

Выделенный сервер необходим для размещения проектов, которые либо требуют мощных вычислительных ресурсов, либо не могут соседствовать на одной машине с другими проектами или их компонентами (например, сложное бизнес-приложение, или база данных, или площадка сетевой игры), либо требуют прямого доступа к оборудованию.

**Что такое виртуальный хостинг**

Виртуальный хостинг (shared hosting) — это предоставление в аренду одного физического сервера сразу нескольким арендаторам. Его аппаратные мощности распределяются между всеми пользователями. Во избежание неравномерности распределения мощностей машины (процессорной мощности, дисковой и оперативной памяти) и их дефицита во время пиковых нагрузок некоторые хостеры ограничивают ресурсы сервера для скриптов отдельного пользователя (script — файл или программа со сценарием автоматизации некоторых процессов сайта). Настройка «под себя» отсутствует — все пользуются одним и тем же ПО, которое устанавливает провайдер. Подходит для небольших и несложных проектов. Это самый дешёвый вид хостинга.

**Что такое VPS и VDS**

VPS (Virtual Private Server) и VDS (Virtual Dedicated Server) — по сути одно и то же. Это виртуальные серверы, эмулирующие работу физического сервера и предоставляемые хостером в аренду клиенту с максимальными привилегиями: установка операционной системы и программного обеспечения, root-доступ (права главного администратора). Клиент получает в пользование выделенные аппаратные ресурсы сервера с полным управлением и контролем, но, возможно, не все, имеющиеся на машине. То есть на одном физическом сервере запущено несколько виртуальных серверов под разных клиентов, но на каждом из них администратор (заказчик) может работать так же, как на отдельной физической машине, при этом арендная плата существенно ниже, чем за выделенный сервер.

**Разница между VPS и VDS**

*VPS.* Виртуализация такого сервера осуществляется в рамках операционной системы (контейнеризация) с помощью специальных технологий (FreeBSD Jail, Parallels Virtuozzo Containers, iCore Virtual Accounts, OpenVZ и других). Клиент может работать в персональном изолированном программном окружении с правами суперпользователя, но не может поменять саму операционную систему или её ядро. Также существуют разные лимитирования, основанные на политике хостера (например, непрерывная доступность ресурсов сервера).

*VDS.* Виртуализация этого вида сервера осуществляется на аппаратном уровне. Это физическое выделение ресурсов машины, точнее эмуляция её физических элементов при помощи специальных технологий (VMWare, LDoms, KVM, Microsoft Hyper-V, XEN). Клиент не только получает в постоянное пользование ресурсы CPU, дискового пространства и памяти, но также может устанавливать иную операционную систему и совершать её модификации вплоть до смены настроек BIOS.

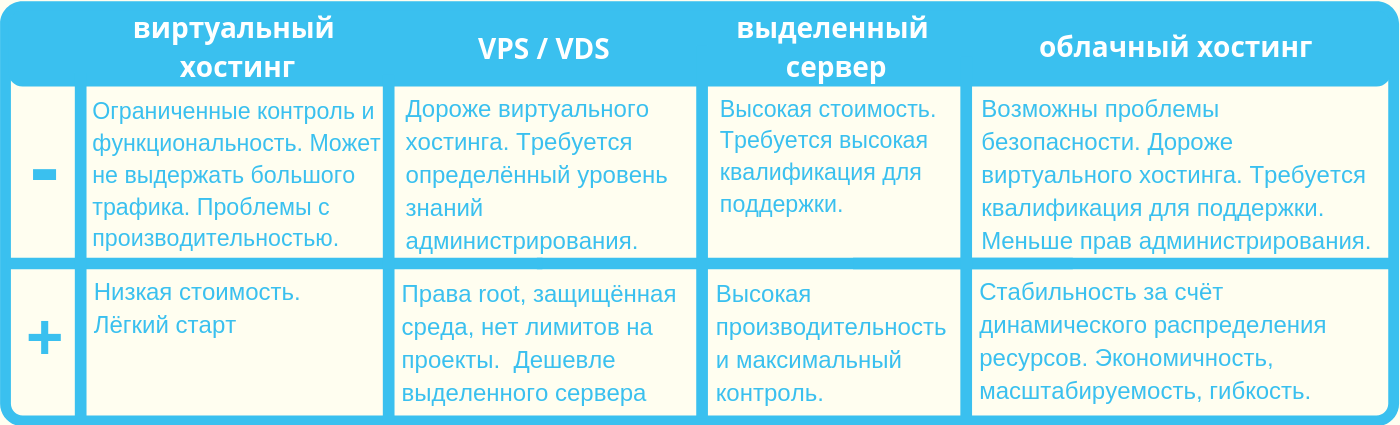
**Что такое облачный хостинг**

Облачный хостинг использует облачные ресурсы для развёртывания сайтов, мобильных приложений и даже целых «офисов в облаке» (виртуальный офис). В отличие от других видов хостинга это происходит не на одном локальном сервере, а на нескольких связанных в кластер физических и виртуальных серверов к которым добавляются связанные в сеть системы хранения данных (SAN). При аренде пользователь платит только за используемые по факту ресурсы, которые могут быстро масштабироваться или отключаться в зависимости от нужд. Причём оплата почасовая, то есть можно настроить потребление ресурсов в дневное и ночное время. Настройки могут регулироваться в автоматическом режиме через приложение. Эта гибкость — основное преимущество облачного хостинга.

Облачная модель набирает популярность во многом благодаря разработкам BaaS (Backend-as-a-Service — бэкенд как услуга) и IaaS (Infrastructure-as-a-Service — инфраструктура как услуга). Не будем углубляться в её описание, так как это выходит за рамки статьи. Скажем только, что разработчиков привлекает в ней универсальность кроссплатформенного бэкенда для любого проекта, а владельцев бизнеса — надёжность резервного копирования и удобство коллективной работы над проектами. Подходит для крупных и быстрорастущих проектов, компаний с сезонными колебаниями спроса.

В облаке сайты размещаются в обособленных средах — контейнерах. В одном контейнере, например, могут располагаться рабочие версии сайтов, в другом — тестовые. Процессы и возможные неполадки в работе сайтов как внутри контейнеров, так и между контейнерами никак не отражаются на остальных сайтах. Внутри отдельного контейнера можно размещать неограниченное количество проектов.

**Сравнение выделенного сервера, виртуального сервера, виртуального хостинга и облачного хостинга**



**SSH**

SSH — это Secure Shell (безопасная оболочка); это специальный протокол для передачи данных в безопасном режиме. Он очень часто применяется для удаленного управления компьютерами и устройствами по сети.

Протокол SSH-1 начал свою историю в 1995 году. Изначально он имел достаточное количество брешей и проблем, поэтому уже в 1996 году была презентована его следующая версия. SSH-1 и SSH-2 — это два разных протокола, которые не являются совместимыми между собой. На сегодняшний день, когда речь идет о SSH, подразумевается SSH-2. Вторая версия протокола до сих пор работает и за это время претерпела небольшие изменения.

Чистый SSH — это коммерческий продукт, который можно использовать только на платной основе. В бесплатном распространении есть одна из версий SSH — это OpenSSH. Именно протокол OpenSSH наиболее популярен среди разработчиков, так как он бесплатен, очень безопасен и распространяется с открытым исходным кодом.

Протокол SSH — это то, что дает следующие возможности:

* можно подключаться к удаленному компьютеру и работать с ним через командную строку;
* возможно осуществить шифрование данных, применяя различные алгоритмы;
* он передает любые данные по зашифрованному каналу, в том числе и аудио- или видеофайлы;
* может сжать файлы для их дальнейшего шифрования и транспортировки по сети;
* обеспечивает надежную транспортировку данных и предотвращает несанкционированное подключение к каналу передачи информации с целью ее перехвата.

Для того чтобы настроить функционирование протокола SSH, нужны:

* SSH-сервер. Именно сервер отвечает за коммуникацию и аутентификацию удаленных компьютеров. Аутентификация на сервере может происходить тремя путями: по IP-адресу клиента, по публичному ключу, по паролю клиента. В качестве SSH-сервера могут выступать следующие программы: OpenSSH, freeSSHd, lsh-server, WinSSHD, MobaSSH и др.
* SSH-клиент. Это программное обеспечение, которое нужно для интеграции с SSH-сервером и для выполнения на нем различных действий. Например: взаимодействие с различными файлами, редактирование файлов, контроль процесса функционирования SSH, архивирование файлов, взаимодействие с базами данных. В качестве SSH-клиентов могут выступать: Vinagre, ZOC, lsh-client, SecureCRT, SSHWindows, OpenSSH, Xshell, i-SSH, PuTTY и мн. др.

**Настройка FTP на Linux**

FTP (протокол передачи файлов) – это стандартный сетевой протокол, используемый для передачи файлов в удаленную сеть и из нее. Для более безопасной и быстрой передачи данных используйте SCP.

Для Linux доступно множество FTP-серверов с открытым исходным кодом. Самыми популярными и широко используемыми являются PureFTPd, ProFTPD и vsftpd.

Убедитесь, что вы вошли в систему как пользователь с привилегиями sudo.

Пакет vsftpd доступен в репозиториях Ubuntu. Чтобы установить его, просто выполните следующие команды:

*sudo apt update*

*sudo apt install vsftpd*

Служба vsftpd автоматически запустится после завершения процесса установки. Проверьте его, выведя статус службы:

*sudo systemctl status vsftpd*

Результат будет выглядеть примерно так, показывая, что служба vsftpd активна и работает:

*\* vsftpd.service - vsftpd FTP server*

*Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset: enabled)*

*Active: active (running) since Mon 2018-10-15 03:38:52 PDT; 10min ago*

*Main PID: 2616 (vsftpd)*

*Tasks: 1 (limit: 2319)*

*CGroup: /system.slice/vsftpd.service*

*`-2616 /usr/sbin/vsftpd /etc/vsftpd.conf*

**Настройка vsftpd**

Сервер vsftpd можно настроить, отредактировав файл /etc/vsftpd.conf. Большинство настроек задокументированы внутри файла конфигурации. Для всех доступных вариантов посетите официальную страницу vsftpd.

Начните с открытия файла конфигурации vsftpd:

*sudo nano /etc/vsftpd.conf*

1. Доступ к FTP

Мы разрешаем доступ к FTP-серверу только локальным пользователям, находим директивы anonymous\_enable и local\_enable и проверяем соответствие вашей конфигурации следующим строкам:

/etc/vsftpd.conf

*anonymous\_enable=NO*

*local\_enable=YES*

1. Включение загрузки

Раскомментируйте настройку write\_enable, чтобы разрешить изменения в файловой системе, такие как загрузка и удаление файлов.

/etc/vsftpd.conf

*write\_enable=YES*

1. Chroot

Чтобы пользователи FTP не могли получить доступ к каким-либо файлам вне своих домашних каталогов, раскомментируйте настройку chroot.

/etc/vsftpd.conf

*chroot\_local\_user=YES*

По умолчанию для предотвращения уязвимости безопасности, когда chroot включен, vsftp откажется загружать файлы, если каталог, в котором пользователи заблокированы, доступен для записи.

* Метод 1. – Рекомендуемый способ разрешить загрузку – включить chroot и настроить FTP-каталоги. В этой статье мы создадим каталог ftp внутри home пользователя, который будет служить chroot и записываем каталог uploads для загрузки файлов.

/etc/vsftpd.conf

*user\_sub\_token=$USER*

*local\_root=/home/$USER/ftp*

* Метод 2. Еще один вариант – добавить следующую директиву в файл конфигурации vsftpd. Используйте этот параметр, если вы должны предоставить доступ к записи для своего пользователя в свой домашний каталог.

/etc/vsftpd.conf

*allow\_writeable\_chroot=YES*

1. Пассивные FTP-соединения

vsftpd может использовать любой порт для пассивных FTP-соединений. Мы укажем минимальный и максимальный диапазон портов, а затем откроем диапазон в нашем брандмауэре.

Добавьте в файл конфигурации следующие строки:

/etc/vsftpd.conf

*pasv\_min\_port=30000*

*pasv\_max\_port=31000*

1. Ограничение входа пользователя

Чтобы разрешить только определенным пользователям входить на FTP-сервер, добавьте следующие строки в конец файла:

/etc/vsftpd.conf

*userlist\_enable=YES*

*userlist\_file=/etc/vsftpd.user\_list*

*userlist\_deny=NO*

Когда эта опция включена, вам необходимо явно указать, какие пользователи могут войти, добавив имена пользователей в файл /etc/vsftpd.user\_list (по одному пользователю в строке).

1. (Дополнительно) Обеспечение передачи с помощью SSL/TLS

Чтобы шифровать FTP-передачи с помощью SSL/TLS, вам необходимо иметь сертификат SSL и настроить FTP-сервер для его использования.

Вы можете использовать существующий сертификат SSL, подписанный доверенным центром сертификации, или создать самоподписанный сертификат.

Если у вас есть домен или поддомен, указывающий на IP-адрес FTP-сервера, вы можете легко создать бесплатный сертификат SSL для шифрования.

Сгенерируем самоподписанный сертификат SSL с помощью команды openssl.

Следующая команда создаст 2048-битный закрытый ключ и самоподписанный сертификат, действительный в течение 10 лет. Как закрытый ключ, так и сертификат будут сохранены в одном файле:

*sudo openssl req -x509 -nodes -days 3650 -newkey rsa:2048 -keyout /etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem*

Теперь, когда создан SSL-сертификат, откройте файл конфигурации vsftpd:

*sudo nano /etc/vsftpd.conf*

Найдите директивы rsa\_cert\_fileи rsa\_private\_key\_file, измените их значение pam в пути к файлу и установите директиву ssl\_enable на YES:

/etc/vsftpd.conf

rsa\_cert\_file=/etc/ssl/private/vsftpd.pem

rsa\_private\_key\_file=/etc/ssl/private/vsftpd.pem

ssl\_enable=YES

Если не указано иное, FTP-сервер будет использовать только TLS для обеспечения безопасных подключений.

Сохраните файл и перезапустите службу vsftpd, чтобы изменения вступили в силу:

*sudo systemctl restart vsftpd*

**Открытие брандмауэра**

Если вы используете брандмауэр UFW, вам необходимо разрешить FTP-трафик.

Чтобы открыть порт 21(порт команд FTP), порт 20(порт данных FTP) и 30000-31000(диапазон пассивных портов), выполните следующие команды:

*sudo ufw allow 20:21/tcp*

*sudo ufw allow 30000:31000/tcp*

Чтобы избежать блокировки, мы откроем порт 22:

*sudo ufw allow OpenSSH*

Перезагрузите правила UFW, отключив и снова включив UFW:

*sudo ufw disablesudo ufw enable*

Чтобы проверить выполнение изменений:

*sudo ufw status*

*Status: active*

*To Action From*

*-- ------ ----*

*20:21/tcp ALLOW Anywhere*

*30000:31000/tcp ALLOW Anywhere*

*OpenSSH ALLOW Anywhere*

*20:21/tcp (v6) ALLOW Anywhere (v6)*

*30000:31000/tcp (v6) ALLOW Anywhere (v6)*

*OpenSSH (v6) ALLOW Anywhere (v6)*

**Создание пользователя FTP**

Чтобы протестировать наш FTP-сервер, создадим нового пользователя.

* Если у вас уже есть пользователь, которому вы хотите предоставить FTP-доступ, пропустите 1-й шаг.
* Если вы установили allow\_writeable\_chroot=YES в своем файле конфигурации, пропустите 3-й шаг.

Создайте нового пользователя с именем newftpuser:

*sudo adduser newftpuser*

Добавьте пользователя в список разрешенных пользователей FTP:

*echo "newftpuser" | sudo tee -a /etc/vsftpd.user\_list*

Создайте дерево каталогов FTP и установите правильные разрешения:

*sudo mkdir -p /home/newftpuser/ftp/uploadsudo chmod 550 /home/newftpuser/ftpsudo chmod 750 /home/newftpuser/ftp/uploadsudo chown -R newftpuser: /home/newftpuser/ftp*

Пользователь сможет загружать свои файлы в каталог ftp/upload.

На данный момент FTP-сервер полностью работоспособен, и должна быть возможность подключаться к вашему серверу с любым FTP-клиентом, который может быть настроен на использование шифрования TLS, например, FileZilla.

**Установка LAMP в Linux**

1. Установка пакетов lamp с помощью утилиты tasksel

Утилита tasksel позволяет устанавливать наборы пакетов различных программ буквально в одну команду. Она разработана для Debian, но может использоваться и в Ubuntu. Для её установки выполните в терминале команду:

*sudo apt -y install tasksel*

Далее, чтобы установить Lamp на Ubuntu, достаточно выполнить команду:

*sudo tasksel install lamp-server*

Установка займёт определенное время, так как утилите необходимо скачать все необходимые пакеты. После завершения установки вы получите полностью работающий стек LAMP.

1. Установка пакетов lamp вручную

Если вы не хотите использовать tasksel, вы можете установить все пакеты по отдельности из стандартных репозиториев и настроить их для совместной работы.

Подробная инструкция по настройке представлена по [ссылке](https://losst.ru/ustanovka-lamp-v-ubuntu-20-04" \l "1_Установка_пакетов_LAMP_с_помощью_утилиты_tasksel).

1. Настройка брандмауэра

По умолчанию брандмауэр закрывает все неразрешённые входящие подключения. Поэтому, чтобы к вашему веб-серверу можно было получить доступ извне, необходимо добавить его порт в исключения:

*sudo ufw allow in 80/tcp*

1. Проверка работы lamp

Для начала необходимо убедиться, что веб-сервер Apache и СУБД MySQL успешно запустились и при необходимости запустить их. Смотрим статус apache, выполнив команду в терминале:

*sudo systemctl status apache2*

В разделе Active должно быть active (running). Те же самые действия необходимо сделать для проверки MySQL:

*sudo systemctl status mysql*

Если в статусах вместо active будет inactive (dead), то сервисы необходимо запустить вручную:

*sudo systemctl start apache2*

*sudo systemctl status mysql*

Для проверки наберите в адресной строке браузера IP-адрес сервера, куда вы устанавливали LAMP, в качестве примера localhost

Если при открытии вы увидели страницу Apache 2 это означает что веб-север apache успешно запущен и работает. Теперь необходимо проверить правильно ли работает PHP. Создайте файл в директории /var/www/html при помощи следующей команды:

*sudo vi /var/www/html/phpinfo.php*

Вместо *vi* можно использовать любой другой текстовый редактор на ваш выбор (nano, emacs, gedit и др.).

В файл phpinfo.php впишите следующую команду:

*<?php phpinfo(); ?>*

Далее откройте адрес localhost/phpinfo.php в браузере. Если всё работает, вы должны увидеть страничку PHP.

1. Настройка apache

Для обработки запросов будет использоваться модуль mpm\_prefork. Данный модуль запускает по отдельному процессу на каждый запрос. Иначе говоря, каждый процесс одновременно обрабатывает только 1 поток на одно соединение. Для начала необходимо выключить mpm\_event и mpm\_worker и только потом включить модуль mpm\_prefork:

*sudo a2dismod mpm\_event*

*sudo a2dismod mpm\_worker*

*sudo a2enmod mpm\_prefork*

Включаем модуль mod\_rewrite. Это модуль, предназначенный для преобразования URL адресов. Модуль использует в своей работе правила, которые могут быть описаны как в конфигурации сервера (httpd.conf), так и в файлах .htaccess непосредственно в файловой структуре Вашего сайта.

*sudo a2enmod rewrite*

Если вы хотите использовать файлы httaccess для настроек, то необходимо в файле /etc/apache2/apache2.conf изменить значение AllowOverride с None на All для нужных местоположений, например, для /var/www.

Откройте файл apache2.conf при помощи любого текстового редактора:

*sudo vi /etc/apache2/apache2.conf*

Найдите следующий блок:

*<Directory /var/www/>*

*Options Indexes FollowSymLinks*

*AllowOverride None*

*Require all granted*

*</Directory>*

И замените AllowOverride None на AllowOverride All, затем сохраните изменения и закройте файл. Перезапустите apache:

*sudo systemctl restart apache2*

1. Настройка php

По умолчанию короткие теги, обрамляющие скрипты PHP, отключены. Это может привести к тому, что некоторые скрипты, использующие эти теги, могут не выполнятся. Для включения этой возможности откройте файл php.ini при помощи любого текстового редактора:

*sudo vi /etc/php/7.4/apache2/php.ini*

Найдите short\_open\_tag = Off и замените на:

*short\_open\_tag = On*

Вывод ошибок в PHP по умолчанию отключён. Если вы хотите разрабатывать программы на этом сервере, то вам необходимо включить ошибки, иначе вместо ошибок вы будете видеть пустой экран. Для этого найдите эти строки и приведите их к такому виду:

*error\_reporting = E\_ALL*

*display\_errors = On*

После внесения всех изменений необходимо перезапустить Apache:

*sudo systemctl restart apache2*

**Порядок выполнения работы.**

Подготовка существующего сайта или создание нового сайта с использованием HTML+CSS+JavaScript и PHP (или иных средств разработки) для дальнейшего размещения на хостинге.

**Контрольное задание.**

1. Выбор хостинга для размещения сайта. В качестве примера, можно использовать следующие сервисы: [Yandex.Cloud](https://cloud.yandex.ru), [Firebase](https://firebase.google.com) и другие.
2. Настройка подключения к хостингу. Использование SSH или иного подключения к удаленному серверу.
3. Настройка [FTP подключения](https://andreyex.ru/ubuntu/kak-ustanovit-ftp-server-s-vsftpd-na-ubuntu-18-04/) к хостингу.
4. Перенос файлов сайта на уделенную машину с использованием FTP или Git.
5. Настройка сети на хостинге. Определение внешнего адреса удаленного сервера.
6. Подключение к «развернутому» сайта из сети Интернет.
7. Проверка работоспособности сайта.