|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

**ПРАКТИЧЕСКАЯ РАБОТА №3**

по дисциплине «Разработка серверных частей интернет-ресурсов»

**Студент группы** ИКБО-24-20 Краев И. А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Руководитель практической работы**  Благирев М.М.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Москва 2022

# **Цель работы:**

Создать сложную серверную конфигурацию, состоящую из связки apache + nginx + php + База данных.

# **Постановка задачи:**

В задании предлагается создать сложную серверную конфигурацию, состоящую из связки apache+nginx+php+База данных. Возможно использование связки apache+php как единый компонент. В данной конфигурации предполагается создание как минимум 3 элементов(контейнеров) или использование как основы серверной конфигурации, созданной в практической работе №1. В этой конфигурации предполагается акселерированное проксирование без кэширования.

# **Ход работы:**

Для тестирования конфигурации было создано простое тестовое веб-приложение на тему «Магазин настольных игр». Содержимое приложения представлено на рисунках 1-4. Две статические страницы развернуты на сервере nginx, две динамические на apache.

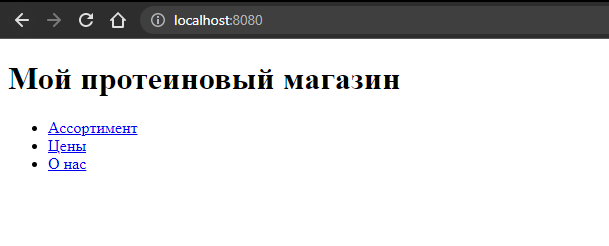


Рисунок 1 – страница index.html.

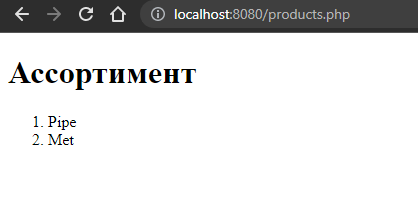


Рисунок 2 – страница products.php.

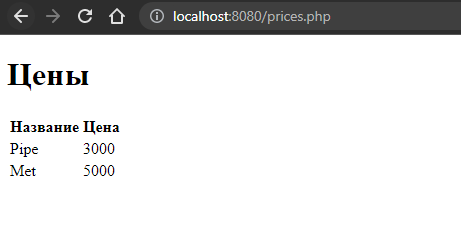


Рисунок 3 – страница prices.php.

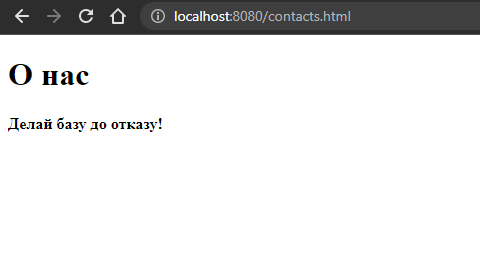


Рисунок 4 – страница contacts.php.

Содержимое файла docker-compose.yml представлено на рисунке 5.

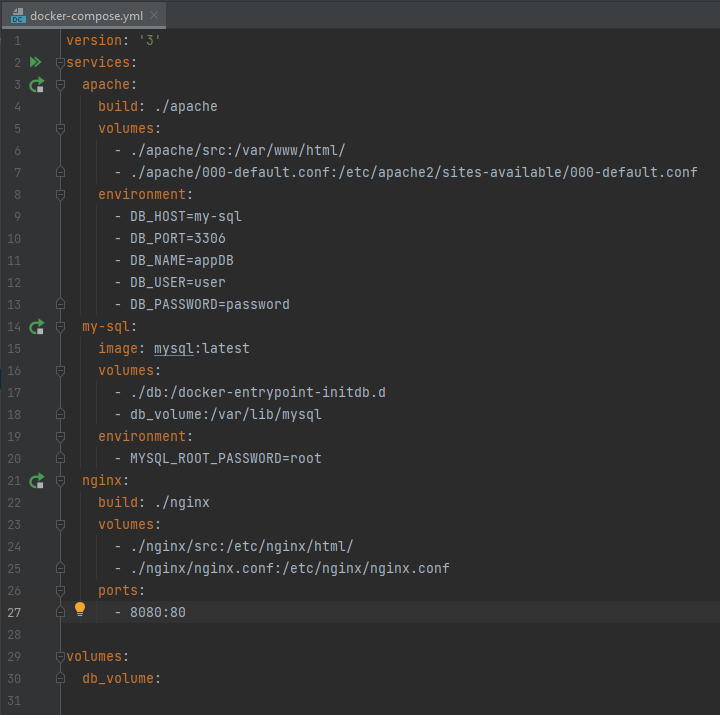


Рисунок 5 – Файл docker-compose.yml

Чтобы наша конфигурация, состоящая из двух серверов и базы данных работала, необходимы конфигурационные файлы. Конфигурационные файлы nginx и apache серверов представлены на рисунках 6 и 7 соответственно.

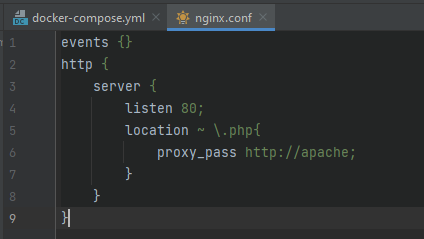


Рисунок 6 – Файл nginx.conf

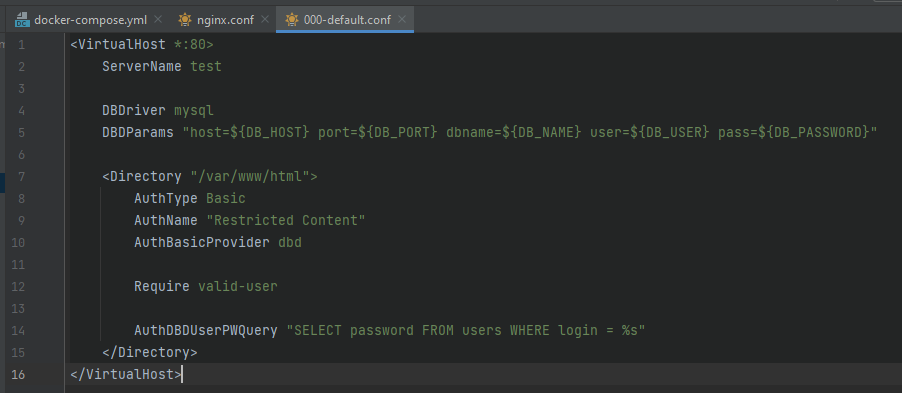


Рисунок 7 – Файл 000-default.conf

Dockerfile apache и nginx представлены на рисунках 8, 9.

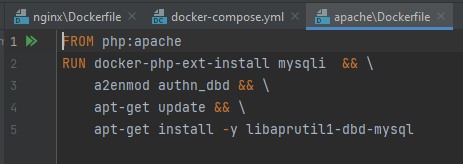


Рисунок 8 – Dockerfile apache

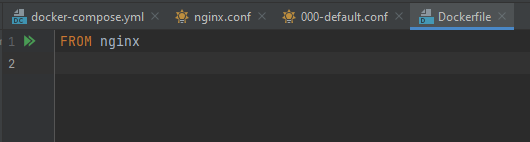


Рисунок 9 – Dockerfile nginx

# **Выводы:**

В ходе выполнения практической работы была создана серверная конфигурация, состоящая из apache, nginx, php, db

# **Ссылка на репозиторий:**

https://github.com/Grigalgol/PHP\_3

# **Ответы на вопросы:**

1. **Что такое веб-сервер?**

Веб-сервер-это компьютерное программное обеспечение и базовое оборудование, которое принимает запросы через HTTP (сетевой протокол, созданный для распространения веб-контента) или его безопасный вариант HTTPS.

1. **Что такое сервер приложения и чем он отличается от веб-сервера?**

Веб-сервер — компьютерная программа, нон-стоп обрабатывающая запросы пользователей и показывающая им HTML-страницы. Проще говоря, это любое устройство, на базе которого работает сайт. Сервер приложения — программа, поддерживающая определенную сетевую логику в полноценном приложении.

1. **Кратко опишите протокол HTTP**

Протокол передачи гипертекста (HTTP) - это протокол прикладного уровня в модели Internet protocol suite для распределенных, коллаборативных гипермедиа-информационных систем.

1. **Опишите технологию SSI.**

Технология SSI (Server Side Includes) – это директивы, которые помещаются в HTML-страницы и оцениваются на сервере, когда обращаются к страницам. Она позволяет вам добавлять динамически сгенерированное содержание в существующую HTML-страницу без необходимости обработки всей страницы через CGI-программу или другую динамическую технологию.

1. **Что такое система управления контентом?**

Система управления контентом (CMS) - это компьютерное программное обеспечение, используемое для управления созданием и модификацией цифрового контента (content management).

1. **Верно ли, что сервер приложения умеет работать с протоколом HTTP?**

Нет. Сервер приложений не умеет работать с протоколом HTTP и обрабатывать пользовательские запросы, так как это задача веб-сервера.

1. **Что такое CGI?**

Стандарт интерфейса, используемого внешней программой для связи с веб-сервером. Программу, которая работает по такому интерфейсу совместно с веб-сервером, принято называть шлюзом, хотя многие предпочитают названия «скрипт» или «CGI-программа».

1. **Назовите достоинства и недостатки CGI.**

CGI не налагает особых условий на платформу и web - сервер, поэтому работает на всех популярных платформах и web - серверах. Также технология не привязана к конкретному языку программирования и может быть использована на любом языке, работающем со стандартными потоками ввода/вывода.

· Производительность CGI - программ не высока. Основной причиной этого является то, что при очередном обращении к серверу для работы CGI – программы создается отдельный процесс, что требует большого количества системных ресурсов.

· Встроенных средств масштабируемости технология не предусматривает.

· CGI - программа представляет из себя готовый к исполнению файл, что препятствует легкому расширению системы.

1. **Что такое FastCGI?**

Интерфейс FastCGI - клиент-серверный протокол взаимодействия веб-сервера и приложения, дальнейшее развитие технологии CGI. По сравнению с CGI является более производительным и безопасным.

1. **Назовите основные отличия CGI от FastCGI.**

FastCGI — развитие технологии CGI, ее более производительная и современная версия. В отличие от CGI, в данном режиме для каждого скрипта не запускается отдельный процесс, благодаря чему меньше расходуются ресурсы.

1. **Что такое менеджер процессов?**

Менеджер процессов отвечает за создание новых процессов в системе и за управление основными ресурсами, связанными с процессом. Все эти услуги предоставляются посредством сообщений.

1. **Условная конструкция в php.**

Особенности режима:

В разы производительнее других режимов работы

Настройки PHP можно задать индивидуально для каждого пользователя, и пользователь может менять их самостоятельно

Для каждого домена можно выбрать версию PHP

Быстрая обработка статики

Не поддерживается .htaccess. Правила необходимо настраивать в конфигурационном файле Nginx

1. **Что такое Spawn-fcgi?**

spawn-fcgi используется для запуска удаленных и локальных [FastCGI](http://vds-admin.ru/) процессов. Почему лучше использовать spawn-fcgi:

Разделение привилегий без необходимости *suid*-исполняемого файла или запуска сервера с привилегиями root.

Возможность отдельного перезапуска как *fastcgi* приложения, так и сервера

Возможность запуска в изолированном окружении ( *chroot* ).

Ваше *FastCGI* приложение не зависит от используемого веб сервера что дает возможность, использовать различные веб серверы.

1. **Что такое Lighttpd?**

lighttpd — [веб-сервер](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80), разрабатываемый с расчётом на скорость и защищённость, а также соответствие стандартам.

1. **Что такое chroot окружение?**

chroot — операция изменения [корневого каталога](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D0%BD%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BA%D0%B0%D1%82%D0%B0%D0%BB%D0%BE%D0%B3) в [Unix-подобных](https://ru.wikipedia.org/wiki/Unix-%D0%BF%D0%BE%D0%B4%D0%BE%D0%B1%D0%BD%D0%B0%D1%8F_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) [операционных системах](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0).

1. **Опишите механизм взаимодействия серверов с использованием FastCGI.**

 FastCGI-процессы используют [Unix Domain Sockets](https://ru.wikipedia.org/wiki/Unix_domain_socket) или [TCP/IP](https://ru.wikipedia.org/wiki/TCP/IP) для связи с сервером.

1. **Что такое SCGI?**

SCGI-это протокол, который определяет связь между веб - сервером и сервером приложений.

1. **Что такое PCGI?**

PCGI (Perl Common Gateway Interface) — библиотека к языку программирования Perl для работы с интерфейсом CGI (Common Gateway Interface). Библиотека позволяет с высокой скоростью обрабатывать входящий поток данных. Основное достоинство заключается в том, что библиотека позволяет совершенно безопасно принимать сколь угодно крупные объёмы данных, при этом очень экономично потребляя оперативную память.

1. **Что такое PSGI?**

PSGI или Perl Web Server Gateway Interface - спецификация, предназначенная для отделения среды веб-сервера от кода веб-фреймворка. PSGI не является программным интерфейсом (API) для веб-приложений.

1. **Что такое WSGI?**

WSGI ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Web Server Gateway Interface) — стандарт взаимодействия между [Python](https://ru.wikipedia.org/wiki/Python)-программой, выполняющейся на стороне сервера, и самим [веб-сервером](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80).

1. **Опишите механизм взаимодействия серверов Apache и PHP.**

Запрос от клиента поступает на сервер, если запрашиваемый документ статический и находится на сервере, то сервер просто отдает его. Если же документ(тип документа) является динамическим, то будет поднят новый процесс, который через интерфейс CGI запустит php-скрипт, а результат его работы вернет клиенту.

1. **Опишите преимущества веб-сервера Apache.**

Бесплатное программное обеспечение как для личного использования, так и для коммерческого;

гибкость программы. Открытый исходный код позволяет подстроить программу под специфические нужды пользователя;

поддержка многих языков программирования (PHP, Python, Ruby, Perl, ASP, Tcl), что дает доступ к различным серверным ПО;

удобная и надежная конфигурация системы. Приложение легко настраивается через текстовые конфигурационные файлы и может быстро масштабироваться;

Apache хорошо работает со многими системами управления контентом ([CMS](https://help.reg.ru/hc/ru/articles/4408054695313)), такими как WordPress, Joomla, Drupal;

поддерживает технологии FastCGI и CGI;

может использовать СУБД для аутентификации пользователей;

большая база инструкций. Так как продукт уже давно на рынке, разработчики и пользователи уже написали множество инструкций под разные ситуации.

1. **Опишите недостатки веб-сервера Apache.**

При высоком трафике снижается производительность. На высокопосещаемых сайтах может значительно снижаться скорость работы программы. Это связано с особенностью работы сервиса. Каждый пользовательский запрос обрабатывается отдельным потоком. Когда потоки заканчиваются, остальные операции ждут своей очереди. Если планируется большое количество посещений сайта, то лучше использовать [Nginx](https://help.reg.ru/hc/ru/articles/4408047693329);

много параметров настройки. Если допустить ошибку при настройке конфигурации, это может привести к появлению уязвимостей во всей системе;

риск заражения вирусом. Создавать модули могут не только добросовестные разработчики, но и злоумышленники. В модуль может быть внедрён вредоносный код. Устанавливать модули стоит только от проверенных разработчиков.

1. **Опишите архитектуру веб-сервера Apache.**

Apache состоит из: ядра, системы конфигурации, многопроцессорной модели, системы модулей, механизм виртуальных хостов.

1. **Опишите функции ядра веб-сервера Apache.**

Apache состоит из ядра и модулей. Ядро выполняет основные функции: обработка конфигурационных файлов, работа с протоколом HTTP, система загрузки модулей.

1. **Что такое URI, URL и чем они различаются.**

URI является либо URL, либо URN, либо одновременно обоими.

URL — это URI, который, помимо идентификации ресурса, предоставляет ещё и информацию о местонахождении этого ресурса. А URN — это URI, который только идентифицирует ресурс в определённом пространстве имён (и, соответственно, в определённом контексте), но не указывает его местонахождение.

# **Список используемой литературы**

1. Дергачев А. М. Проблемы эффективного использования сетевых сервисов / Научно-технический вестник СПбГУ ИТМО. 2011. № 1 (71). С. 83-87.  
2. Розенфельд Л., Морвиль П. Информационная архитектура в Интернете, 2 е издание. – Пер. с англ. – СПб: Символ Плюс, 2005 – 544 с., ил.  
3. Спинеллис Д., Гусиос Г. Идеальная архитектура. Ведущие специалисты о красоте программных архитектур. – Пер. с англ. – СПб.: Символ Плюс, 2010 – 528 с., ил.  
4. Фаулер, Мартин. Ф28 Архитектура корпоративных программных приложений.: Пер. с англ. — М.: Издательский дом "Вильяме", 2006 — 544 с.: ил. — Парал. тит. англ.  
5. Голицына О.Л., Максимов Н.В., Партыка Т.Л., Попов И.И. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ. - 2-е изд. - Москва: ФОРУМ - ИНФРА-М, 2008. - 395 с.  
6. Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. ван Стеен. — СПб.: Питер, 2003. — с. 83-93 — (Серия «Классик computer science»). ISBN 5-272-00053-6-.