

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

МАТЕРИАЛЫ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

|  |  |
| --- | --- |
| **Технологии хранения в системах кибербезопасности** | |
|  | *(наименование дисциплины (модуля) в соответствии с учебным планом)* |
| Уровень | специалитет |
|  | *(бакалавриат, магистратура, специалитет)* |
| Форма обучения | очная |
|  | *(очная, очно-заочная, заочная)* |
| Направление(-я) подготовки | 10.05.04 Информационно-аналитические системы безопасности |
|  | *(код(-ы) и наименование(-я))* |
| Институт | Кибербезопасности и цифровых технологий (ИКБ) |
|  | *(полное и краткое наименование)* |
| Кафедра | КБ-2 «Прикладные информационные технологии» |
|  | *(полное и краткое наименование кафедры, реализующей дисциплину (модуль))* |
| Лектор | к.т.н., Селин Андрей Александрович, Бугаев Александр Александрович |
|  | *(сокращенно – ученая степень, ученое звание; полностью – ФИО)* |

|  |  |  |
| --- | --- | --- |
| Используются в данной редакции с учебного года |  | 2024/2025 |
|  | *(уче* | *бный год цифрами)* |
| Проверено и согласовано « » 2024 г. |  | А.А. Бакаев |
|  | *(подпись ди* | *ректора Института/Филиала*  *с расшифровкой)* |

Москва 2024 г.

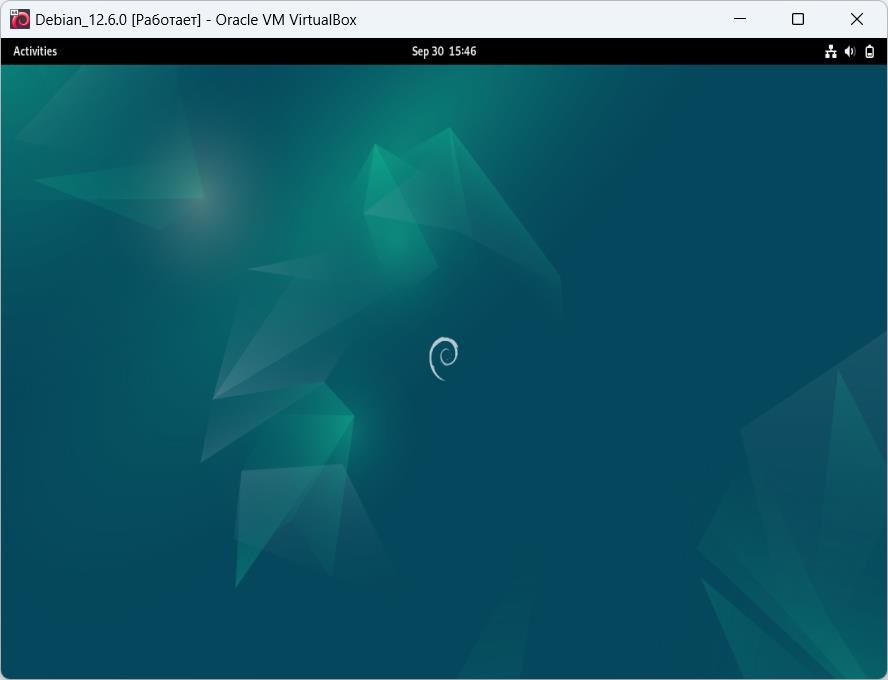
**ПРАКТИЧЕСКАЯ РАБОТА № 3**

**«Изучение основ работы с Docker»**

**Цель работы** – получение навыков развертывания приложений с использованием Docker.

**Задание:**

1. Запустите Unix-подобную систему (в данном примере используется Debian 12.6.0 64-bit[1](#_bookmark0)). Также можно использовать WSL.



1. Создайте пользователя с именем формата **fio\_nn**, где f – первая буква фамилии на латинице;

i – первая буква имени на латинице;

o – первая буква отчества на латинице (при наличии), nn – двузначный номер по списку в группе.

Добавьте его в группу sudo. **Все дальнейшие действия необходимо выполнять от имени созданного пользователя.**



1 Можно скачать готовый образ виртуальной машины по ссылке <https://sourceforge.net/projects/osboxes/files/v/vb/14-D-b/12.6.0/64bit.7z/download>

1. Запустите терминал и установите Docker:
   1. Обновите систему.



* 1. Установите необходимые пакеты, чтобы программа apt могла использовать репозиторий по протоколу HTTPS.
  2. Добавьте официальный GPG-ключ Docker.

**sudo install -m 0755 -d /etc/apt/keyrings**

**sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o**

**/etc/apt/keyrings/docker.asc**

**sudo chmod a+r /etc/apt/keyrings/docker.asc**

* 1. Добавьте репозиторий Docker в APT sources:

**echo \**

**"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/debian \**

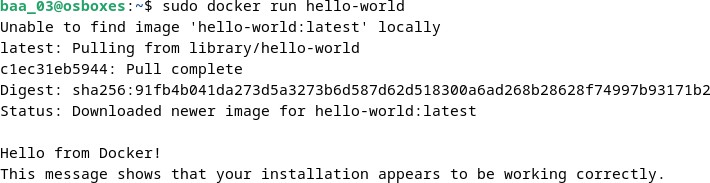
**$(. /etc/os-release && echo "$VERSION\_CODENAME") stable" | \ sudo tee /etc/apt/sources.list.d/docker.list > /dev/null**

* 1. Обновите индекс пакета и установите необходимые пакеты.

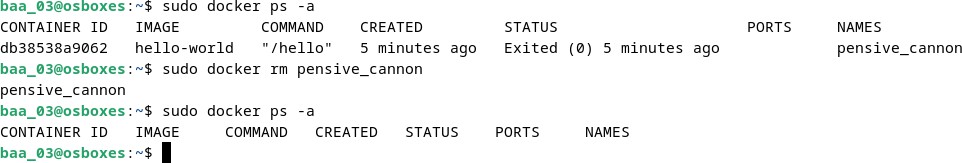
**sudo apt-get update**

**sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin**

* 1. Проверьте работоспособность Docker, используя образ «hello-world».



* 1. Посмотрите список контейнеров и удалите запущенный на предыдущем шаге по сгенерированному имени контейнера.



1. Повторите (изучите):

* основные понятия Docker: образы, контейнеры, volumes, сети и др.;
* основные команды Docker.

Официальная документация: [https://docs.docker.com](https://docs.docker.com/).

1. Изучите основную библиотеку образов для Docker ([https://hub.docker.com](https://hub.docker.com/)). Найдите официальные (или от проверенного издателя) образы PostgreSQL и pgAdmin. Скачайте оба образа с помощью команды **docker pull**. Проверьте список образов с помощью команды **docker images**.
2. Запустите pgAdmin с помощью команды **docker run**.

**Требования к запускаемым сервисам (для всех последующих заданий):**

* последние 2 цифры номера порта, на котором будет развернут сервис, должны соответствовать номеру по списку в группе (например, для 3 – 12303, 8003, 9903 и т.п.);
* имя контейнера должно заканчиваться на символ подчеркивания и инициалы ФИО (например, для Иванова Петра Дмитриевича – postgres\_ipd, pgadmin\_ipd).

Убедитесь в работоспособности сервиса с помощью браузера. Остановите и удалите запущенный контейнер.

1. Повторите/изучите основные команды **Docker Compose (up, stop, rm)** и структуру конфигурационного файла (**docker-compose.yml**).
2. Сформируйте конфигурационный файл **docker-compose.yml** для запуска сервисов PostgreSQL и pgAdmin (**без использования volumes**).
3. Разверните PostgreSQL и pgAdmin с помощью Docker Compose по сформированному конфигурационному файлу.
4. Подключитесь к PostgreSQL из pgAdmin и создайте БД с именем формата test\_fio (например, test\_ipd).
5. Остановите и удалите запущенные контейнеры. Запустите их повторно и проверьте наличие созданной БД.
6. Добавьте в конфигурационный файл **volumes** (любой вариант), чтобы пользовательские данные после удаления контейнеров сохранялись.

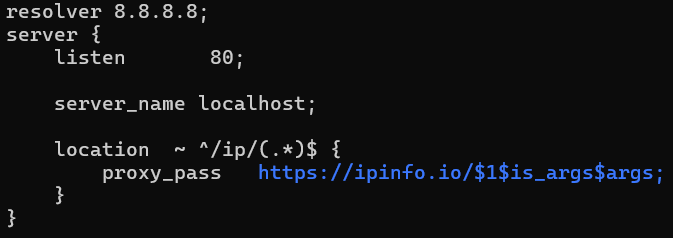
Каталоги данных:

* для PostgreSQL – /var/lib/postgresql/data;
* для pgAdmin – /var/lib/pgadmin.

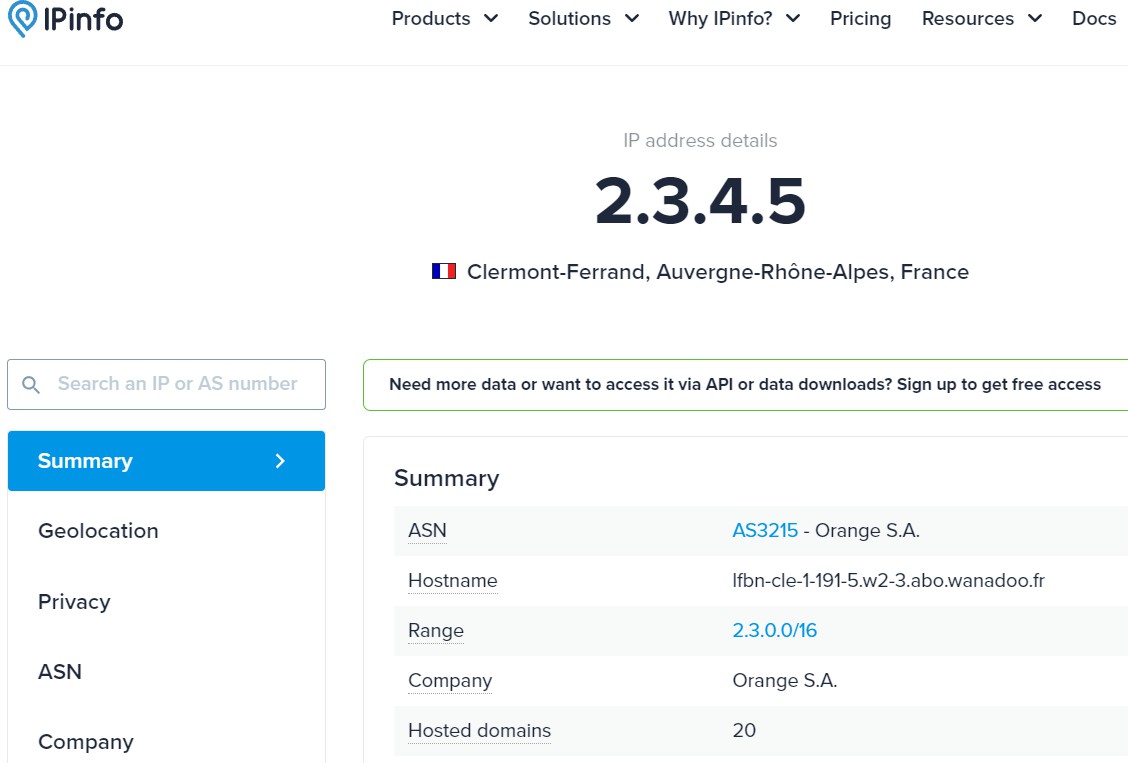
1. Повторите 8-10 пункты с использованием volumes.
2. Создайте пользовательский образ с помощью **Dockerfile**. В образе должен быть развернут веб-сервер (например, nginx). С помощью GET-запроса по протоколу HTTP необходимо получать информацию об IP-адресе (например, перенаправление на сайт [https://ipinfo.io](https://ipinfo.io/)). Развернуть контейнер. Проверить работоспособность.

***Пример для nginx***

В каталоге /etc/nginx/templates создаем файл default.conf.template со следующим содержимым:



При развертывании контейнера настраиваем проброс портов 8080:80. Результат обращения на страницу http://localhost:8080/ip/2.3.4.5:



1. Сохранить образ в архив с помощью команды **docker save**. Пример: **docker save myimage:latest | gzip > myimage\_latest.tar.gz**
2. Загрузить сохраненный образ с помощью команды **docker load** под другим именем (например, с постфиксом «\_new»). Проверить список образов.
3. **Дополнительно:** разверните те же сервисы PostgreSQL и pgAdmin с использованием 2 отдельных конфигурационных файлов (docker-compose.yml). Свяжите их между собой по сети с использованием драйверов bridge и external. Проверьте работоспособность.