**Практическая работа №22**

**Установка SQL сервера**

**Цель**

Научиться устанавливать разные СУБД, используя docker.

**Литература**

1. <https://itproblog.ru/%D1%83%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B0-ubuntu-server-22-04-lts/>
2. <https://docs.docker.com/engine/install/ubuntu/>
3. <https://docs.docker.com/engine/install/linux-postinstall/>

**Задание**

1 Создать в VirtualBox виртуальную машину, в которой будут устанавливаться СУБД.

1.1 Указать следующие параметры при создании:

Имя: ubuntuXXNN (XX - номер группы, NN - номер ПК от 01 до 14)

Папка: c:/temp/isppXX (XX - номер группы)

Образ ISO: сетевой диск с ISO, ubuntu-24.04-live-server-amd64.iso

Имя пользователя: root

Пароль: 1

Доменное имя: dbmsXXNN.local

Дополнения гостевой ОС: да (VBoxGuestAdditions.iso)

Основная память: 6144

Процессоры: 4

Размер диска: 15

Сеть: сетевой мост

1.2 Указать в терминале параметры ОС при установке:

Type of installation: Ubuntu Server (minimized)

Your name: username

Your servers name: ubuntuXXNN (XX - номер группы, NN - номер ПК от 01 до 14)

Pick a username: user

Choose a password: 1

SSH configuration: Install OpenSSH Server

1.3 Для удобства написания команд можно использовать MobaXterm (терминал для удаленного администрирования компьютеров и серверов).

Для подключения в MobaXterm выбрать:

Session - SSH

Remote host: ip-адрес (для того, чтобы узнать ip-адрес, написать в терминале виртуальной машины **ip a**)

Specify user name: user

2 Установить docker

Docker – ПО для автоматизации развёртывания приложений в средах с поддержкой контейнеризации, Позволяет «упаковать» приложение со всем своим окружением и зависимостями в контейнер.

2.1 Актуализировать доступные пакеты в репозиториях и установить доступные обновления для системы:

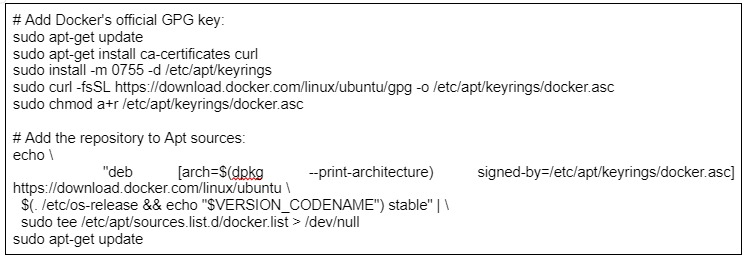
sudo apt update

sudo apt upgrade

2.2 Выполнить набор команд для установки docker и плагинов

Для установки выполнить шаги раздела “Install using the apt repository” со страницы <https://docs.docker.com/engine/install/ubuntu/>

- выполнить набор команд для установки docker:



- выполнить команду для установки плагинов (нажать y при запросе подтверждения):



- проверить, что удается запустить тестовый проект hello-world:

sudo docker run hello-world

|  |
| --- |
| Если тестовый проект не запустился:  *Исправить настройки в netplan (требуется из-за особенностей сети колледжа):*  *- установить текстовый редактор vim и открыть конфигурацию netplan*  *sudo apt install vim*  *sudo vim /etc/netplan/что-то тут (что-то тут заполнится при нажатии клавиши Tab)*  *Дописать в конфигурацию сетевого интерфейса параметр mtu со значением 1400. Для перехода в режим редактирования нажать Insert, для выхода – Esc.*    *После внесения изменения в файл сохранить его (:wq) и применить конфигурацию*  *sudo netplan apply*  *поплакать, если не работает*  *sudo docker run hello-world* |

2.3 Добавить пользователя в группу docker (создается автоматически при установке docker)

Настройка прав: <https://docs.docker.com/engine/install/linux-postinstall/>

- проверить работу docker без sudo:

docker run hello-world

- добавить текущего пользователя в группу docker:

sudo usermod -aG docker $USER

newgrp docker

- проверить работу docker без sudo

docker run hello-world

3 Установка adminer (утилита для управления СУБД в браузере)

В папке /home/user/ создать папку dbms. В ней будут все контейнеры для работы СУБД.

3.1 Добавить в dbms папку adminer.

3.2 Создать в ней файл docker-compose.yaml со следующим содержимым:

|  |
| --- |
| services:  adminer:  image: adminer  restart: always  ports:  - "8080:8080" |

3.3 Для запуска контейнера:

- перейти в папку настроек контейнера adminer

cd ./dbms/adminer

- запустить контейнер

docker compose up -d

3.4 Отобразить запущенные контейнеры

docker ps

3.5 Проверить, что adminer развернулся, открыв его страницу в браузере:

ip-адрес виртуальной машины:8080

4 Добавить контейнеры для работы СУБД

4.1 Создать контейнер для развертывания СУБД MySQL 9:

- добавить в dbms папку mysql9,

- добавить в нее файл docker-compose.yaml со следующим содержимым:

|  |
| --- |
| services:  db:  image: mysql:latest #имя образа  container\_name: mysql9\_dbms #имя контейнера  hostname: mysql9  restart: unless-stopped  ports:  - "3307:3306" # портХоста:портКонтейнера  environment:  - MYSQL\_ROOT\_PASSWORD=Passw0rd! |

*- портКонтейнера: порт сервиса в контейнере (например, для СУБД MySQL по умолчанию – 3306)*

*- портХоста: порт, по которому подключаться к сервису в контейнере (например, для СУБД MySQL – 3306, но в этом примере изменен на 3307)*

- проверить настройки контейнера:

- перейти в папку настроек контейнера СУБД

cd /путь/к/папке/ mysql9

- вывести конфигурацию docker compose

docker compose config

- запустить контейнер

- отобразить запущенные контейнеры

4.2 Настройка переменных окружения

Добавить в dbms файл dbms.env (для хранения переменных окружения контейнеров).

Указать в файле dbms.env список переменных:

|  |
| --- |
| MYSQL\_ROOT\_PASSWORD=Passw0rd!  MSSQL\_SA\_PASSWORD=Passw0rd!  ACCEPT\_EULA=YES  POSTGRES\_PASSWORD=Passw0rd! |

4.3 Создание файлов docker-compose.yaml

Добавить в dbms папки mssql, mysql, pgsql (для контейнера каждой СУБД).

В каждую из папок для контейнеров (mssql, mysql, pgsql) добавить файл docker-compose.yaml со следующим содержимым (данные указаны в таблице далее):

|  |
| --- |
| services:  db:  image: имя образа  container\_name: имя контейнера  restart: unless-stopped  volumes:  #- "./data:путь к данным в контейнере" # вариант для монтирования папки data  - data:путь к данным в контейнере # вариант для работы с томом data  ports:  - "портХоста:портКонтейнера"  env\_file:  - ../dbms.env  networks:  - backend  # healthcheck – опциональный раздел  healthcheck:  test: команда для проверки работы СУБД  interval: 10s  timeout: 20s  start\_period: 40s  start\_interval: 4s  retries: 5    volumes:  data:    networks:  backend:  external: false  driver: bridge |

После создания файлов docker-compose.yaml:

- проверить настройки контейнеров

- запустить контейнеры

- отобразить запущенные контейнеры

Данные для конкретной СУБД:

|  |  |  |  |
| --- | --- | --- | --- |
| СУБД | MSSQL | MySQL | Postgres |
| имя образа | mcr.microsoft.com/mssql/server:2022-latest | mysql:8.3 | postgres:latest |
| имя контейнера | mssql\_dbms | mysql\_dbms | pgsql\_dbms |
| путь к данным в контейнере | /var/opt/mssql | /var/lib/mysql | /var/lib/postgresql/data |
| порт контейнера | 1433 | 3306 | 5432 |
| команда для проверки работы СУБД | /opt/mssql-tools18/bin/sqlcmd -C -S localhost -U sa -P "$${MSSQL\_SA\_PASSWORD}" -Q "SELECT 1" | mysqladmin ping -h localhost  или  mysql -u root -p"$${MYSQL\_ROOT\_PASSWORD}" -e "SHOW DATABASES;" | pg\_isready -U postgres |

5 Подключение к СУБД

5.1 Проверить работу СУБД, подключившись к ней через adminer, DBeaver, аналоги

|  |  |  |  |
| --- | --- | --- | --- |
| СУБД | MSSQL | MySQL | Postgres |
| Сервер | ip-адрес ВМ:1433 | ip-адрес ВМ:3306 | ip-адрес ВМ:5432 |
| Пользователь | sa | root | postgres |
| Пароль | из .env | из .env | из .env |

5.2 Создать БД, таблицу в каждой БД и заполнить их данными

5.3 Остановить и перезапустить контейнер одной из БД, чтобы убедиться, что данные сохраняются:

Остановка контейнера без удаления данных из volume

docker compose down

5.4 Остановить и перезапустить контейнер одной из БД, чтобы убедиться, что данные не сохраняются.

Остановка контейнера без удаления данных из volume

docker compose down -v

**Содержание отчета**

1 Титульный лист

2 Цель работы

3 Ответы на контрольные вопросы

4 Вывод

**Контрольные вопросы**

1 Какие стандартные порты для подключения к СУБД MSSQL, MySQL, Postgres?

2 Какие стандартные папки для хранения данных СУБД MSSQL, MySQL, Postgres?

3 Для чего используется Docker?