МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеративное государственное бюджетное образовательное учреждение

высшего профессионального образования

**“УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”**

Кафедра «Измерительно-вычислительные комплексы»

**Отчёт**

к лабораторной работе №3

по дисциплине “Облачные вычисления и системы”

Выполнил:

студент группы ЦИСТбв-51

Нгуен Х. А.

Принял:

преподаватель

Сазонов С. Н.

Ульяновск

УлГТУ

2023

# Цель лабораторной работы

Создать на базе «Яндекс.Облако» БД под управлениемPostgreSQL ирассмотреть основные операции по работе с ней.

# Выполнение работы

В консоли облачный платформы, выбрав облако «cloud-an-nguen», создаём каталог для этой лабораторной работы нажав на кнопку «Создать каталог» в верхнем правом углу (рис. 1).

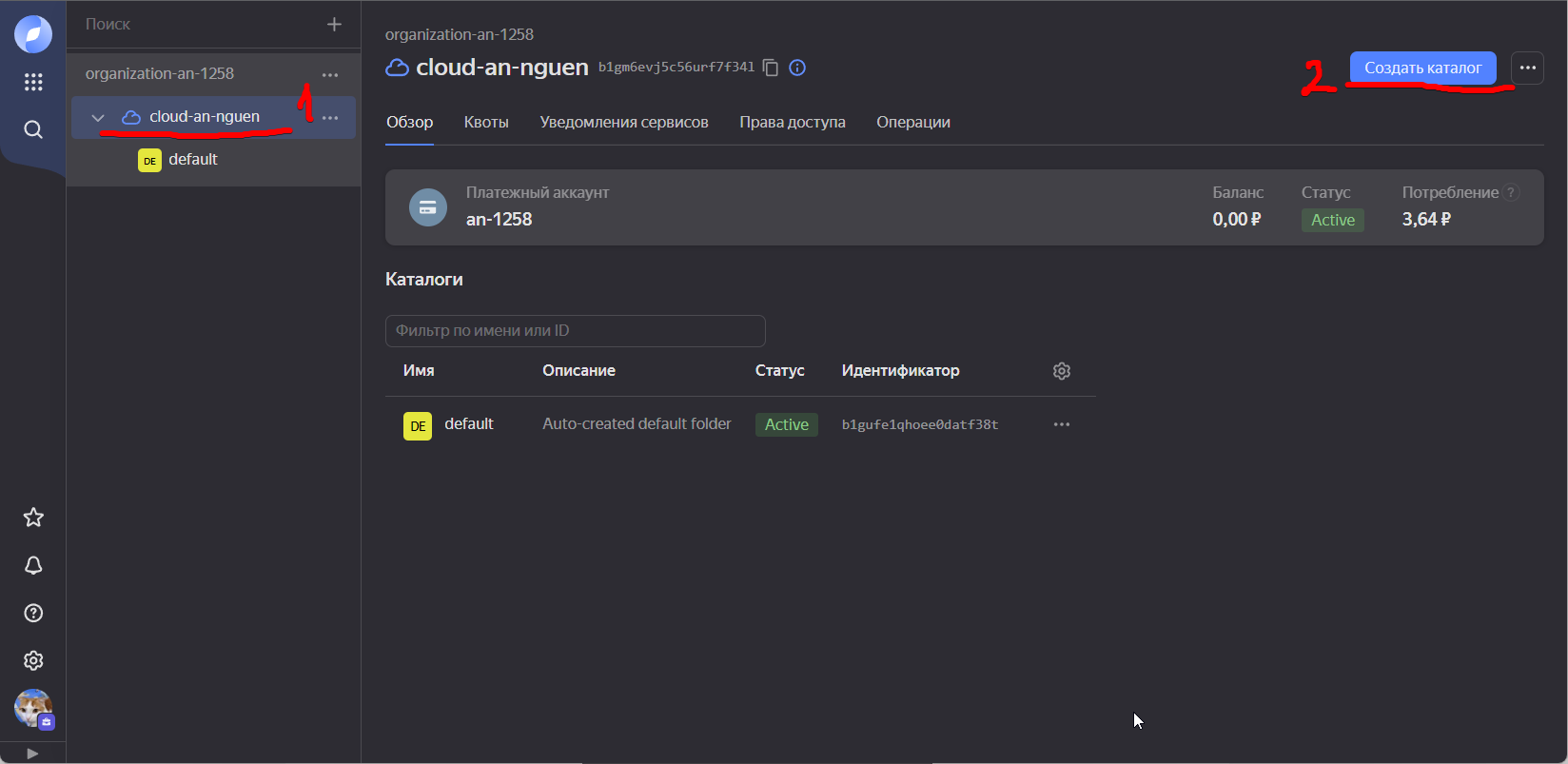


Рисунок 1 – Страница консоли

Далее открывается диалоговое окно создания каталога. В поле «Имя» вводим lab<номер лабораторной работы>-<ФИО> (рис. 2).

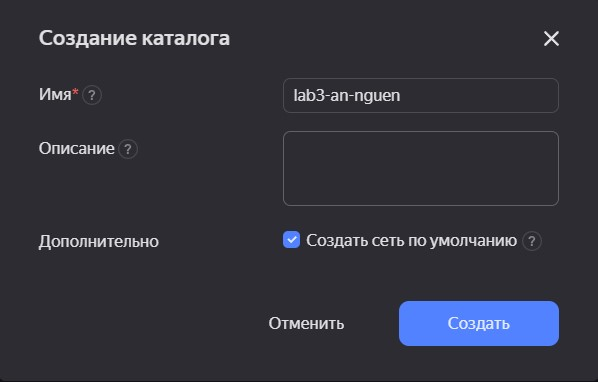


Рисунок 2 – Диалоговое окно создания каталога

Затем откроется страница каталога с выбранной вкладкой «Дашборд каталога». Здесь нам нужно нажать «Создать ресурс» в верхнем правом углу на рис. 3.

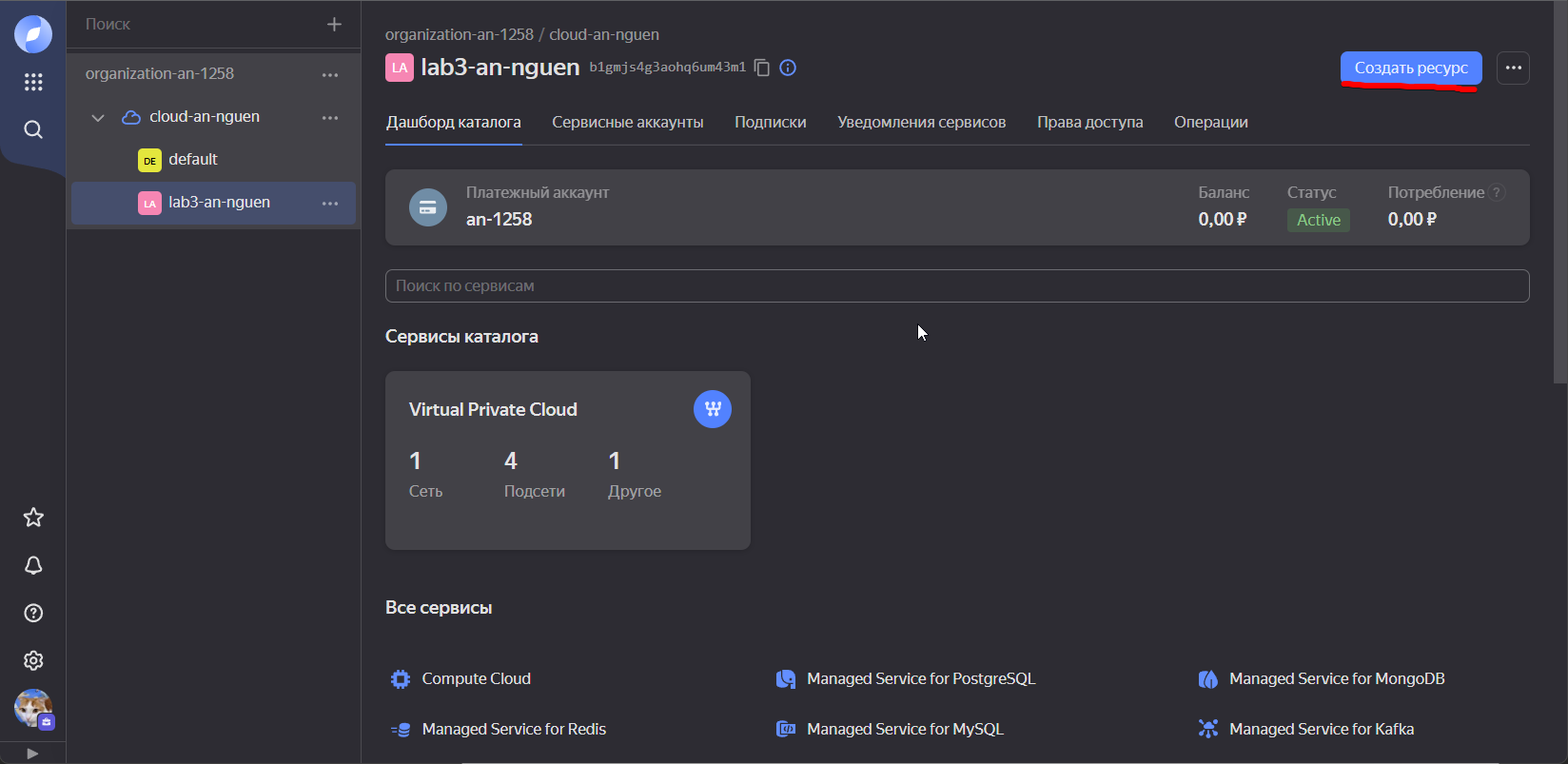


Рисунок 3 – Дашборд каталога

Нажав на кнопку, открывается меню, в котором мы можем выбрать какой ресурс нам нужно создать. Здесь нужно выбрать «Кластер PostgreSQL» (рис. 4).

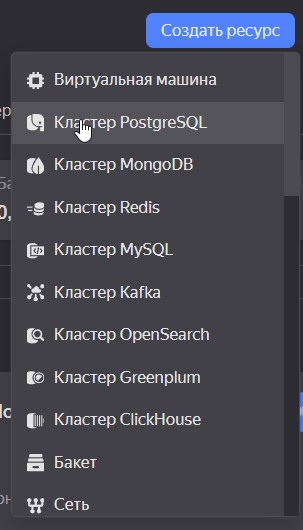


Рисунок 4 – Меню создания ресурса

Далее открывается страница создания кластера PostgreSQL. Вводим имя кластера и хоста (рис. 5).

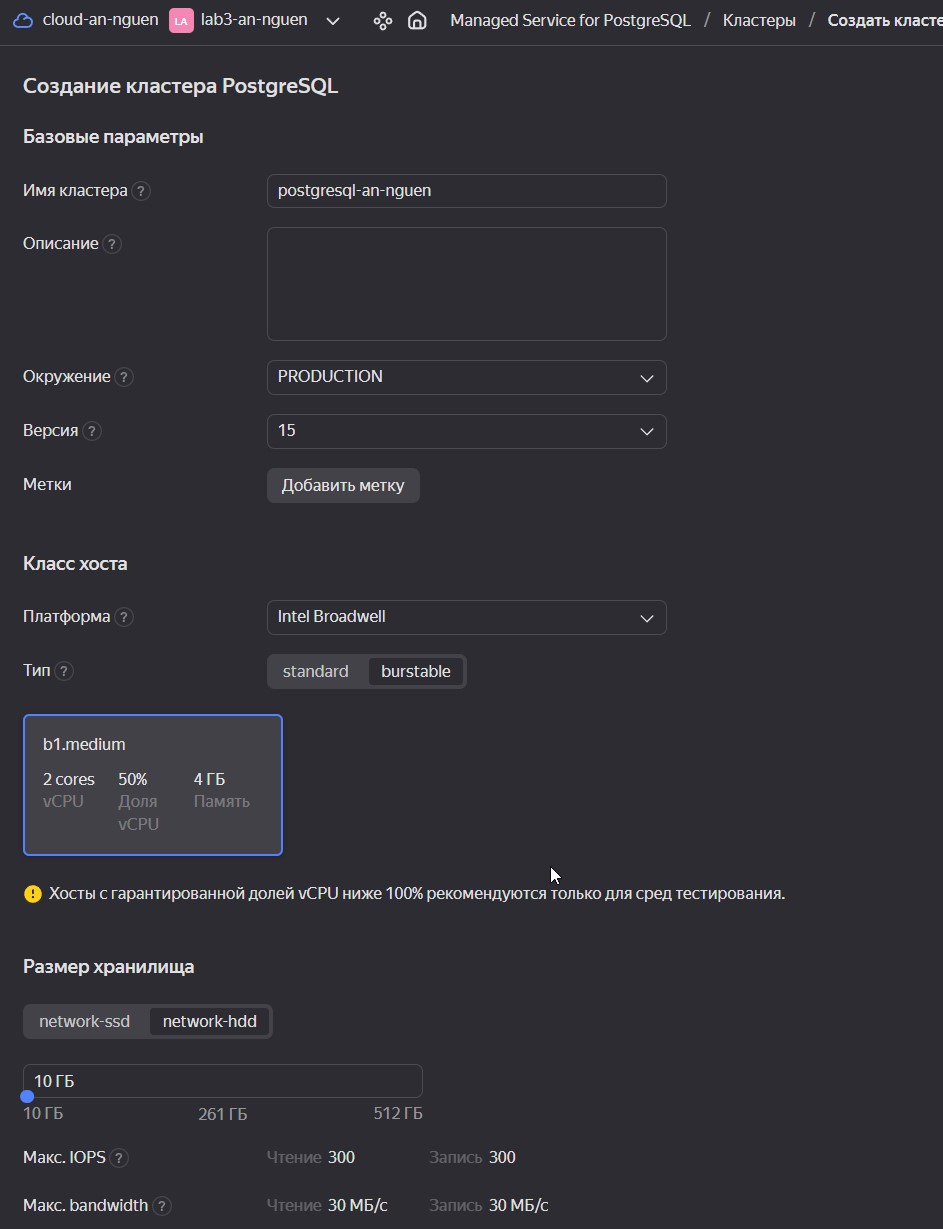


Рисунок 5 – Имя кластера и класс хоста

В блоке «Базы данных» вводим имя создаваемой базы данных, своё имя пользователя и пароль (рис. 6).

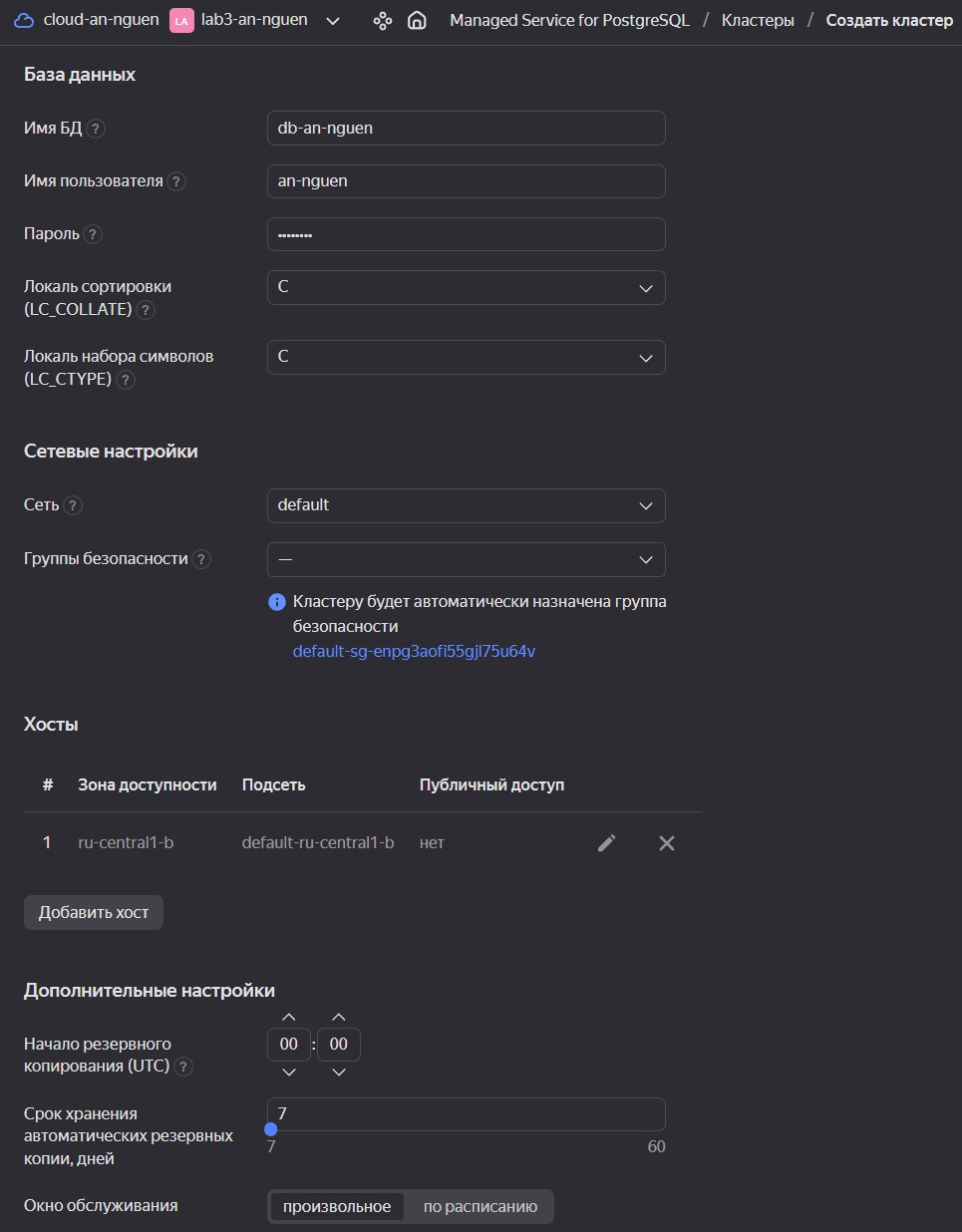


Рисунок 6 – Секция «Базы данных»

Далее в секции «Дополнительные настройки» ставим галки в «Доступ из DataLens» и «Доступ из консоли управления». Затем нажимаем на кнопку создать «Создать кластер» (рис. 7).

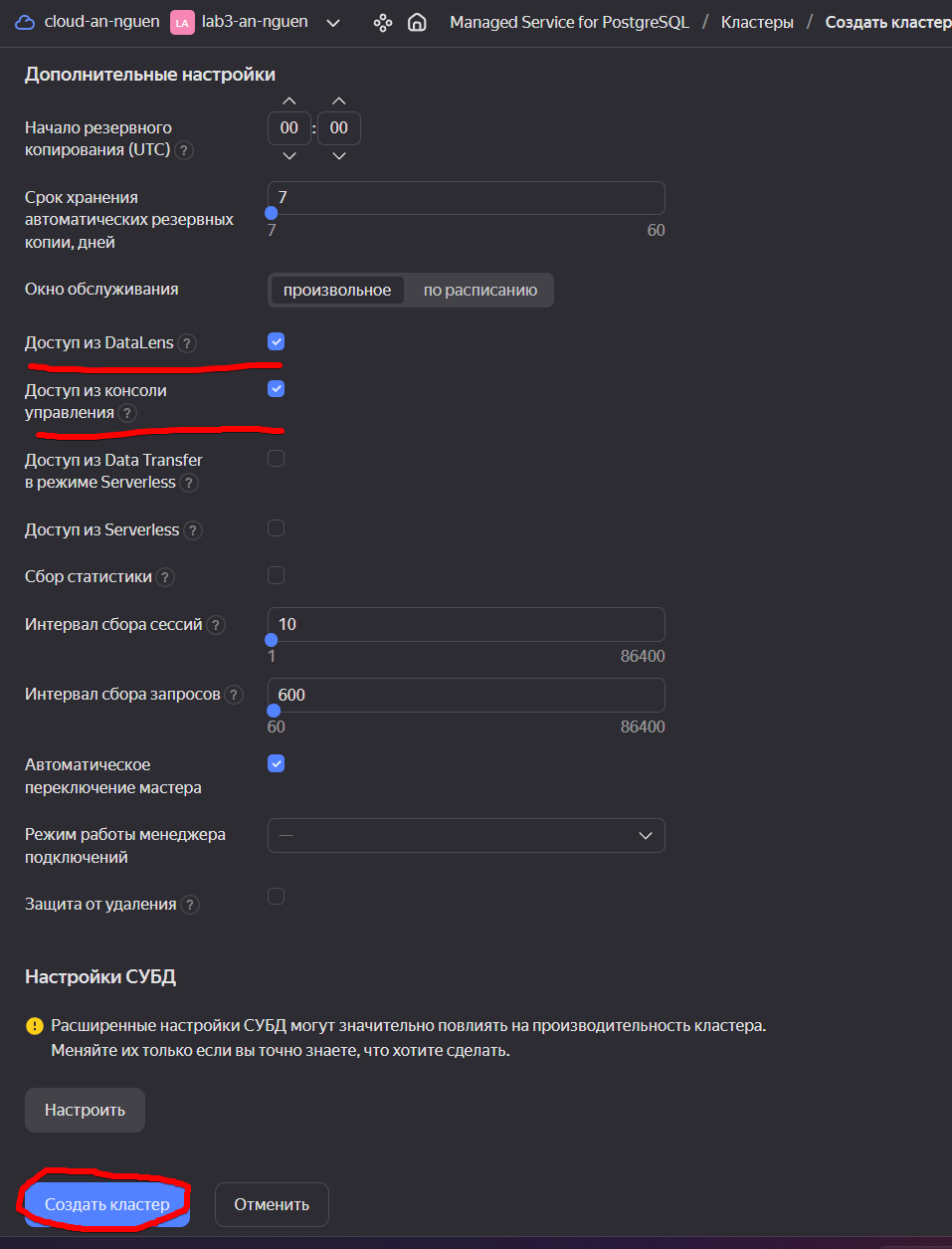


Рисунок 7 – Дополнительные настройки

Далее откроется страница таблица кластеров. Создание кластера занимает некоторое время до тех пока в колонке «Доступность» наша запись кластера в таблице примет значение «Alive» (рис. 8).

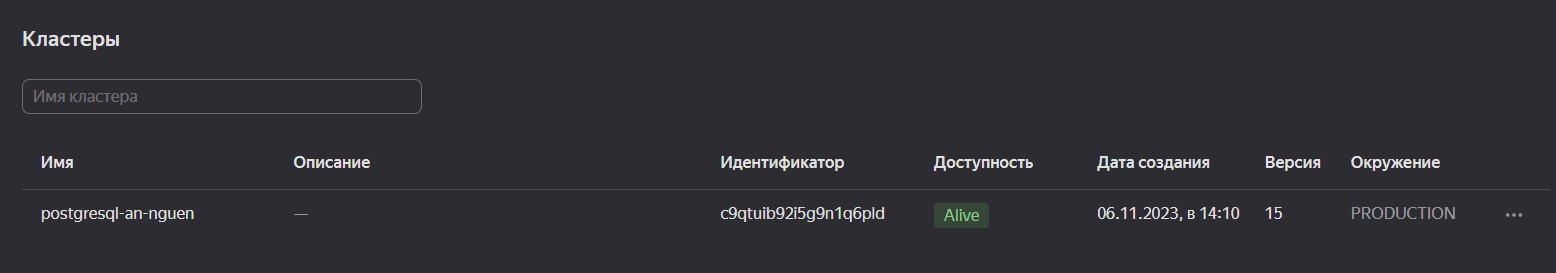


Рисунок 8 – Таблица кластеров

Нажав на запись кластера откроется страница с обзором кластера (рис. 9 и рис. 10).

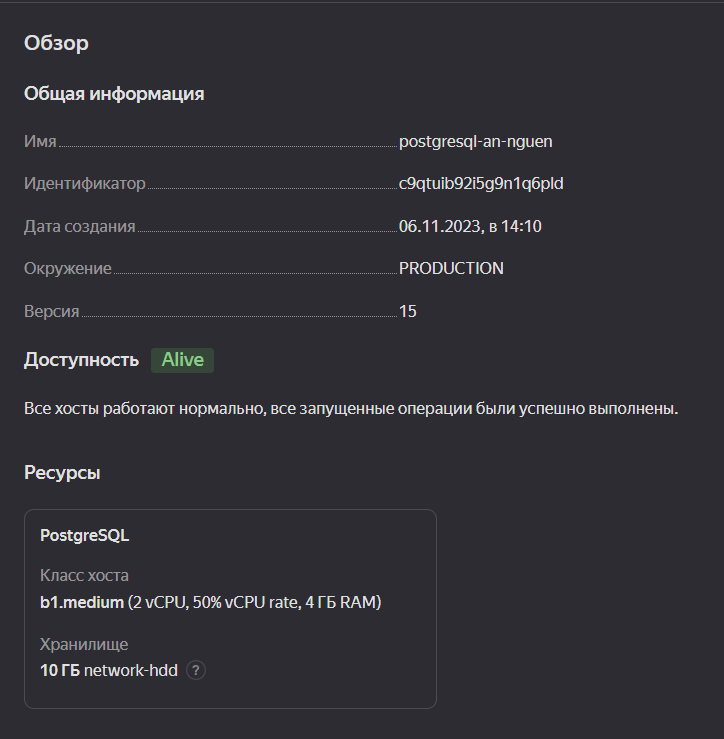


Рисунок 9 – Обзор (часть 1)

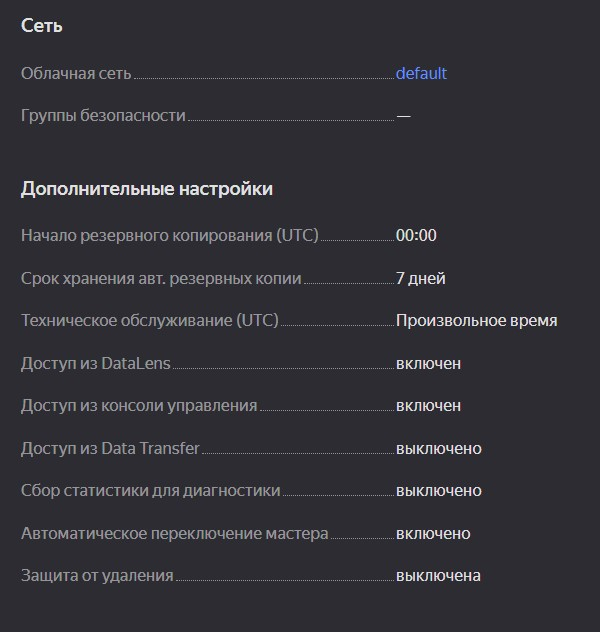


Рисунок 10 – Обзор (часть 2)

Открываем пункт «Базы данных» в левой панели навигации кластера (рис 11).

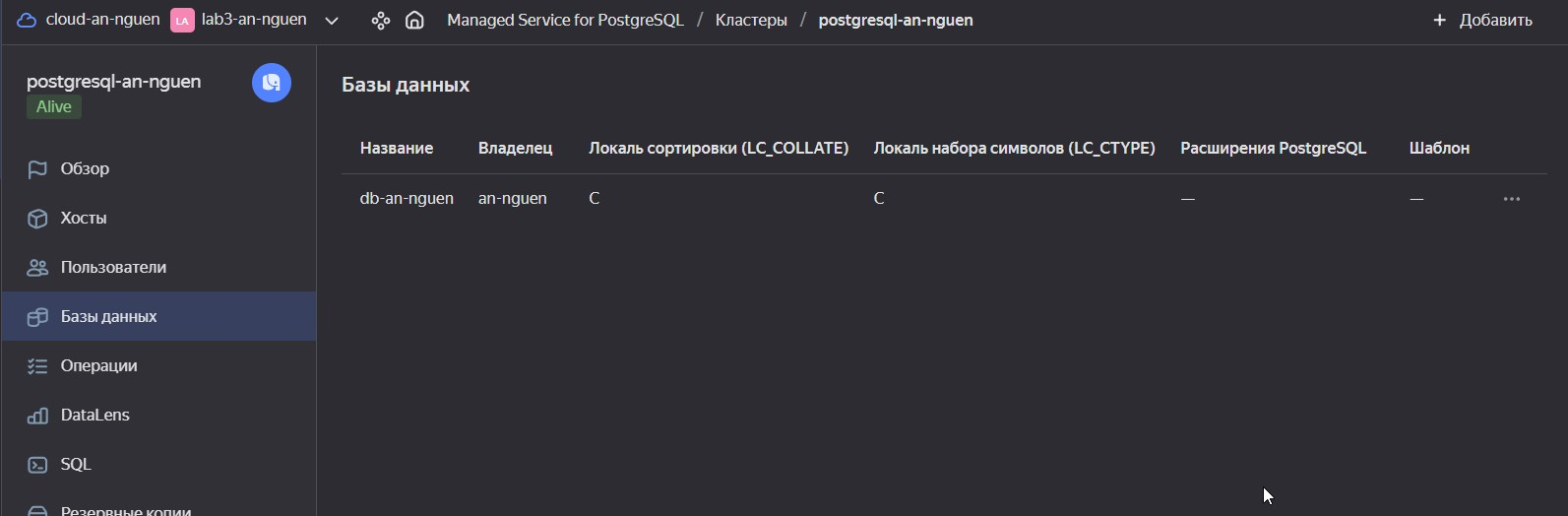


Рисунок 11 – Базы данных

Далее создаём виртуальную машину с Debian в каталоге с кластером СУБД PostgreSQL (рис. 12).

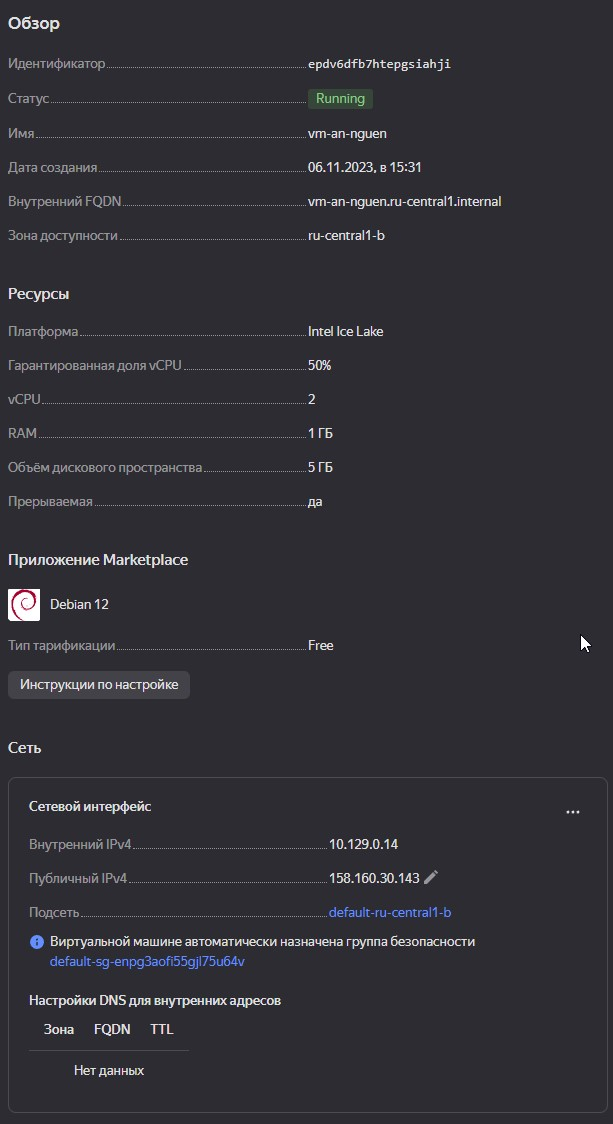


Рисунок 12 – Обзор ВМ

Далее подключаемся по SSH к нашей машине, набрав команду

ssh an-nguen@158.160.30.143

Затем нам нужно обновить список пакетов из репозитория и затем обновить все наши пакеты до последней версии

sudo apt update

sudo apt upgrade

Потом нам нужно установить postgresql клиент, чтобы мы смогли подключиться к СУБД. В репозитории Debian-а она называется «postgresql-client-15». Вводим следующую команду для установки

sudo apt install postgresql-client-15

Далее создаём группу безопасности в Virtual Private Cloud, и открывается страница обзора нашей группы (рис. 13).

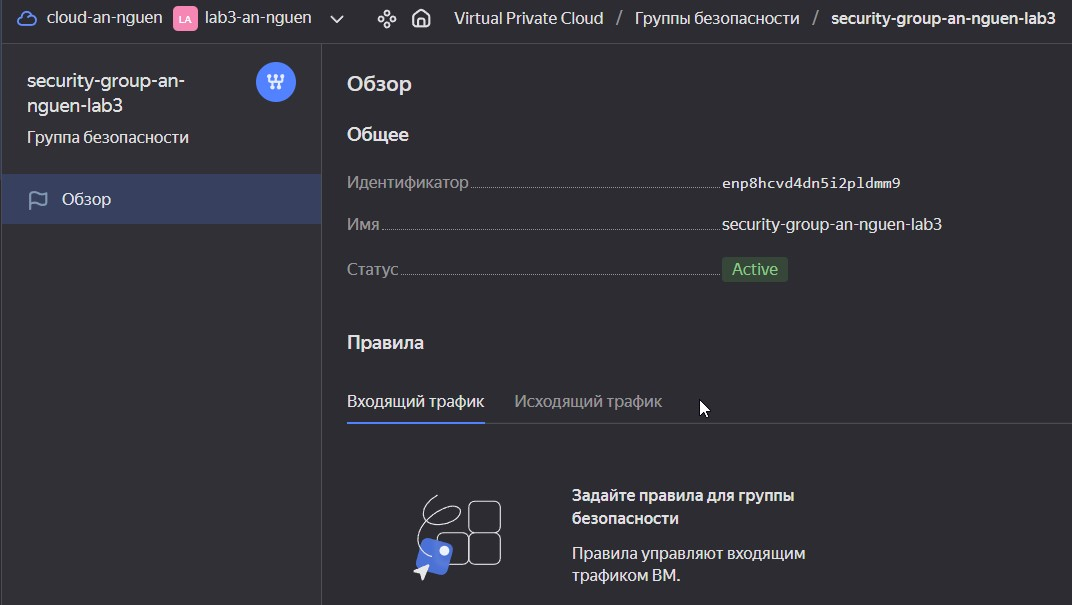


Рисунок 13

Чтобы добавить правило нам нужно открыть страницу редактирования. Её можно открыть в правом верхнем углу нужно нажать на кнопку «Редактировать». Далее в блоке «Правила» во вкладке «Входящий трафик» нажимаем добавить правила.

Нам нужно открыть порт 6432 для работы pgbouncer, который является пулер-ом соединений (connection pooler). Эта служба управляет соединениями к СУБД, может создавать новые соединения или использовать существующие. Любой postgresql клиент может к нему подключиться. На рисунке 14 выведено окно создания правила для pgbouncer.

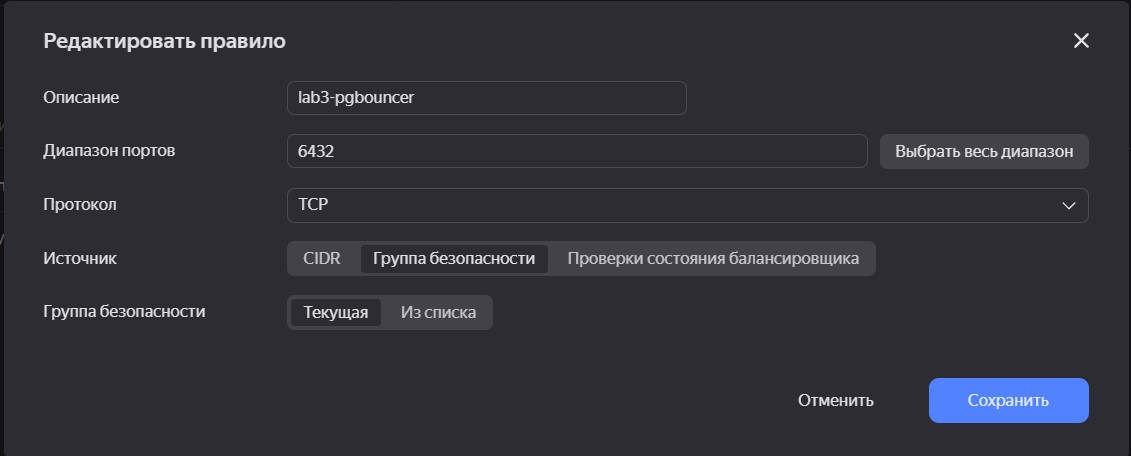


Рисунок 14

Далее нам разрешить соединения по протоколу SSH. Она обычно имеет порт 22 (рис. 15).

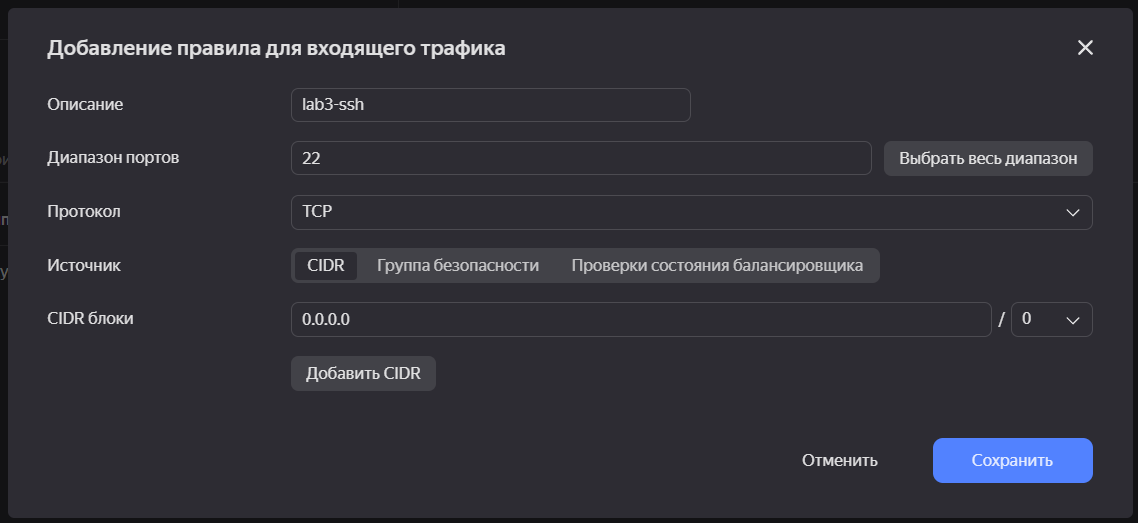


Рисунок 15

Нам нужно разрешить весь исходящий трафик. Во вкладке «Исходящий трафик» создаём такое правило (рис. 16).

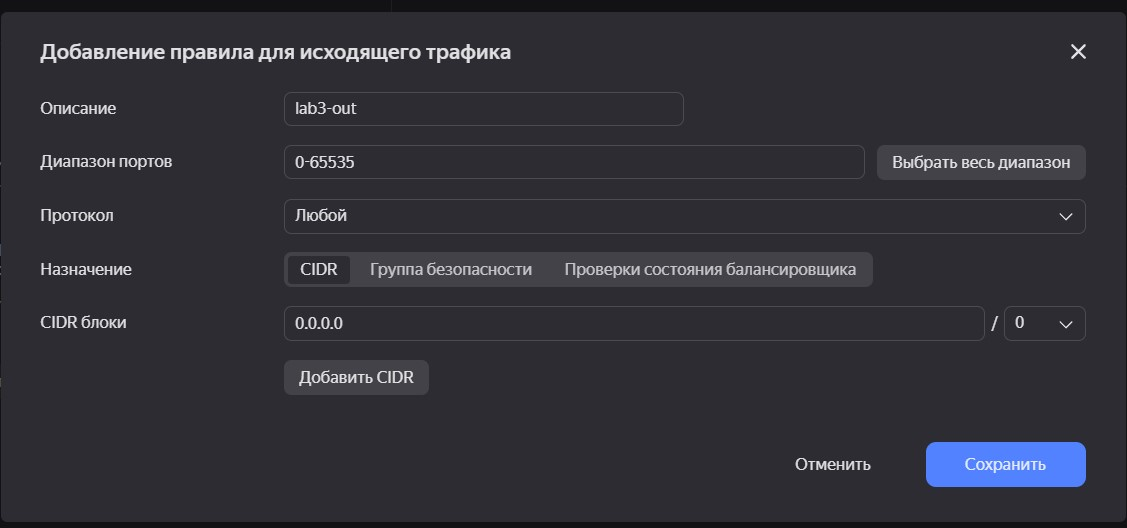


Рисунок 16

Далее нужно не забыть сохранить правила на странице редактирования.

После этого нужно проверить созданные правила. Для этого пробуем подключиться к ВМ (рис. 17).

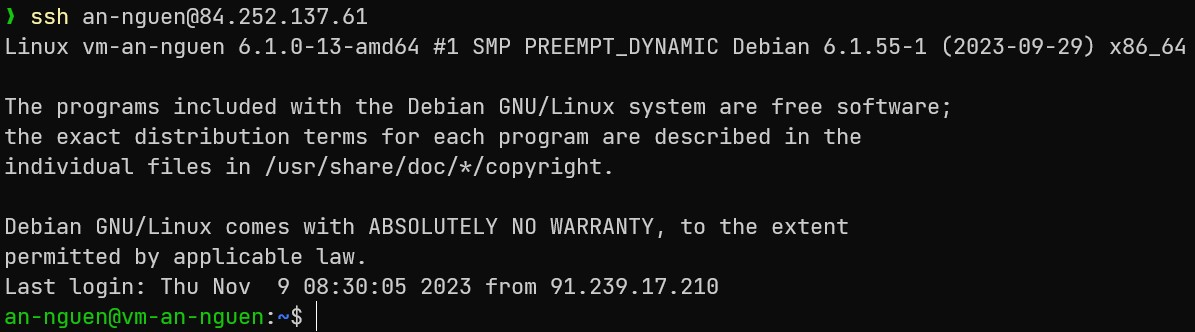


Рисунок 17

Далее нам нужно соединиться к СУБД используя postgresql клиент в ВМ. Параметры соединения можно передать в этот клиент через аргументы командной строки, либо использовать строку соединения (connection string). Перед подключением к СУБД нам нужно установить сертификаты центра авторизации для безопасного подключения к СУБД. В облачной платформе есть заготовленная команда для установки сертификата. В списке кластеров нужно нажать на кнопку «три точки» в строке с созданным кластером, далее открывается меню с действиями и нужно выбрать пункт «Подключиться» (рис. 18).

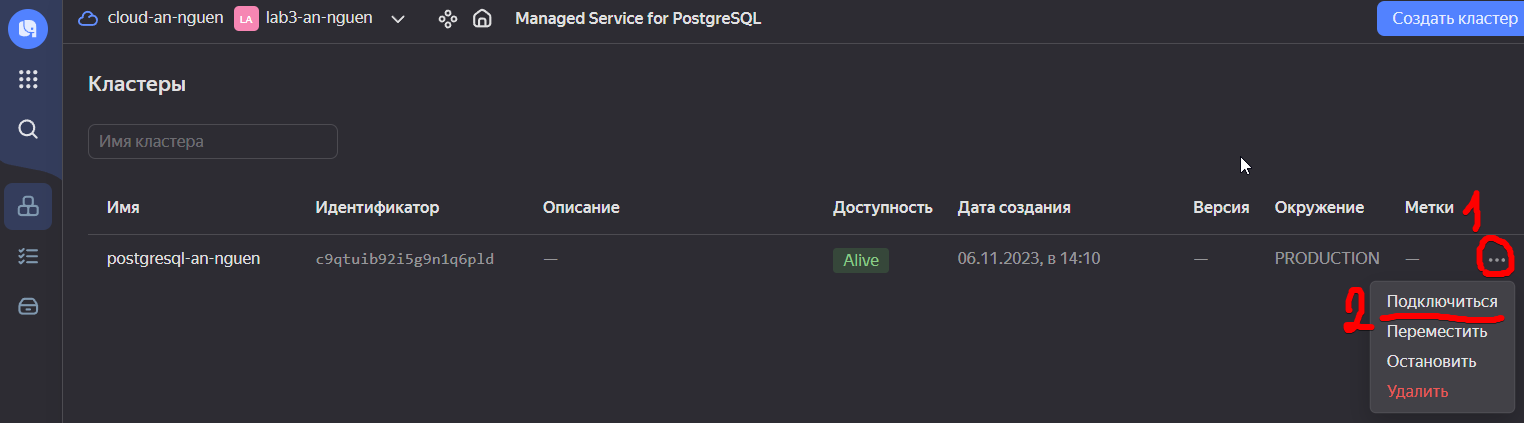


Рисунок 18

Далее открывается диалоговое окно изображенная на рисунке 19.

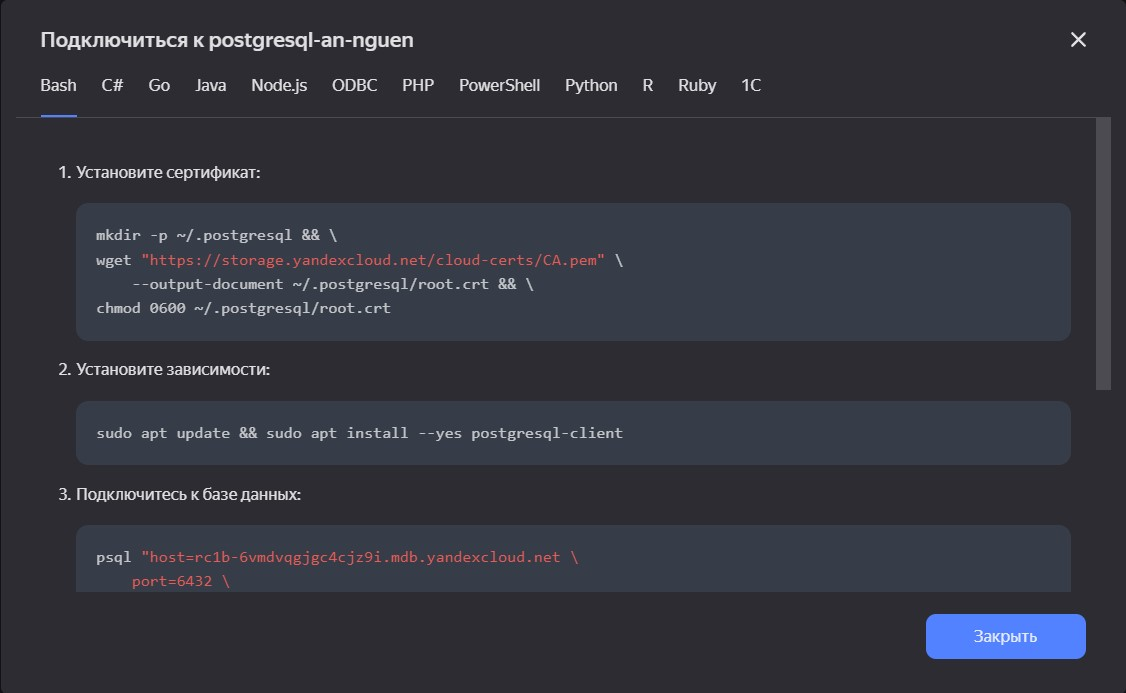


Рисунок 19

Копируем команду установки сертификата из блока 1, вставляем в командную строку и запускаем его в ВМ. Далее выполняем команду подключения из блока 3 (рис. 20).

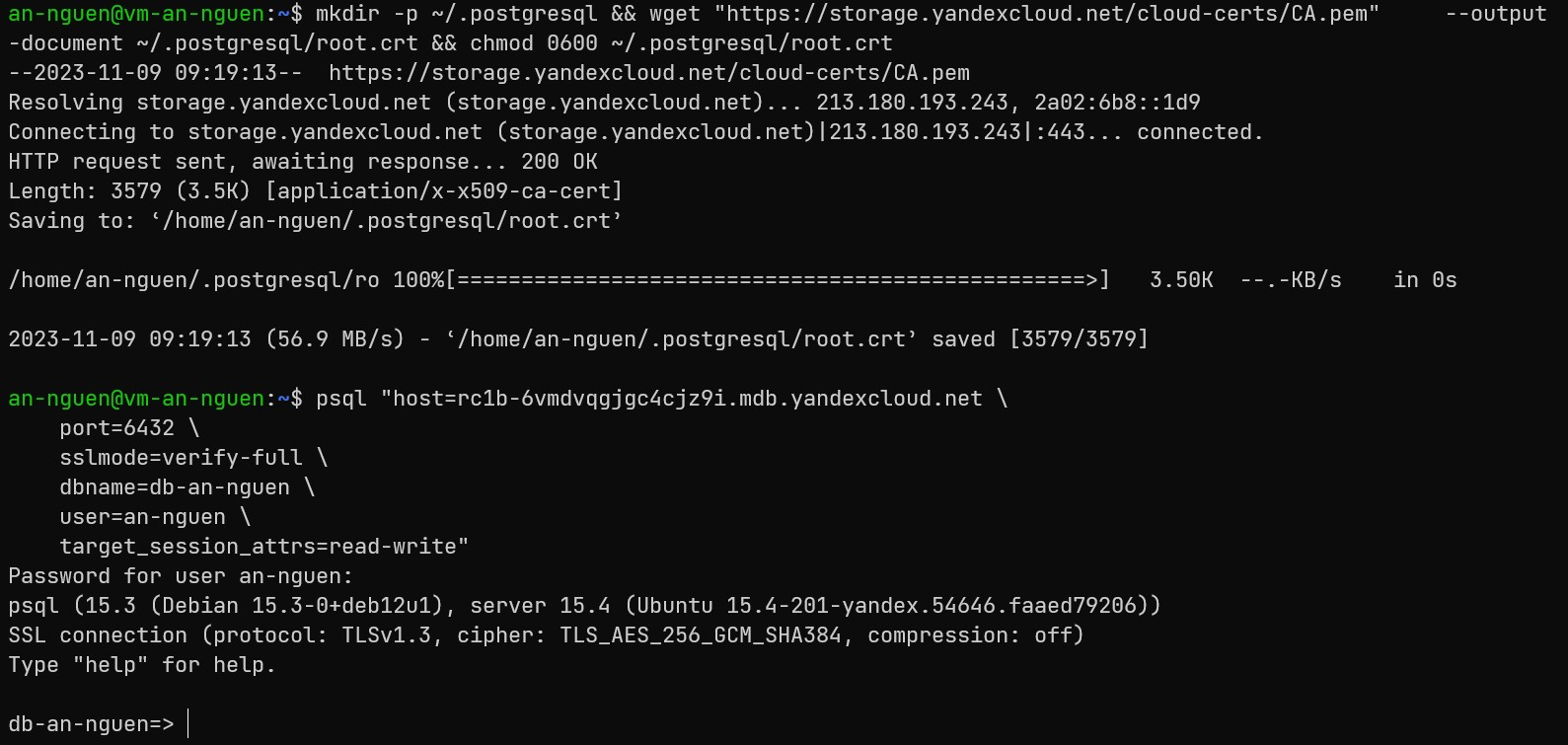


Рисунок 19

Здесь открывается терминальный клиент psql подключенная к СУБД. Значит, что правила группы безопасности работают корректно.

Далее нам нужно создать таблицу со студентами в базе данных. Для этого вводим следующий запрос

CREATE TABLE students (

id int PRIMARY KEY GENERATED ALWAYS AS IDENTITY,

firstname text NOT NULL,

surname text NOT NULL,

midname text,

birthdate date,

student\_group text

);

Каждый запрос должен заканчиваться на точке-запятой.

Результат запроса на рисунке 20.

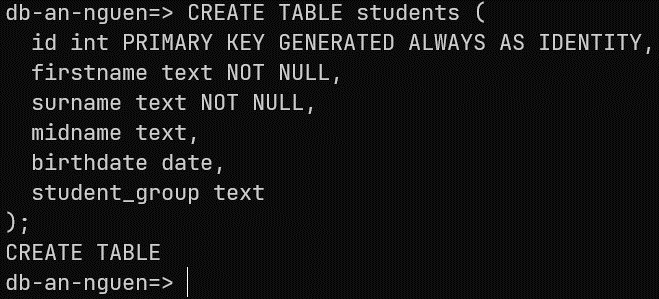


Рисунок 20

Далее создаём таблицу с оценками по дисциплинам, следующим запросом

CREATE TABLE grade (

id int PRIMARY KEY GENERATED ALWAYS AS IDENTITY,

semester int NOT NULL,

educator text NOT NULL,

subject text NOT NULL,

value int NOT NULL,

student\_id int references students(id)

);

Под колонкой *value* подразумевается значение оценки от 2 до 5. И здесь мы используем внешний ключ, которая ссылается на первичный ключ таблицы со студентами. Результат запроса на рисунке 21.

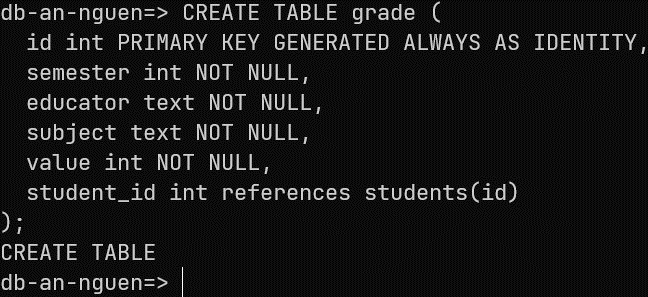


Рисунок 21

После создания двух таблиц их нужно заполнить. Для этого воспользуемся SQL командой INSERT. Примерный синтаксис этой команды

INSERT INTO *<имя\_таблицы> (<колонка\_1>, <колонка\_2>, …, <колонка\_n>)* VALUES (*<значения\_для\_колонки\_1>, <значения\_для\_колонки\_2>, … <значения\_для\_колонки\_n>*), [(*<значения\_для\_колонки\_1>, <значения\_для\_колонки\_2>, … <значения\_для\_колонки\_n>*)].

Для таблицы **students** запрос будет следующим:

INSERT INTO students (firstname, surname, midname, birthdate, student\_group) VALUES ('Хыу Ан', 'Нгуен', NULL, '21.12.1999', 'ЦИСТбв-51'), ('Иван', 'Петров', 'Иванович', '12.09.2001', 'ЦИСТбв-41'), ('Эльдар', 'Филиппов', 'Парфеньевич', '01.02.2001', 'ЦИСТбв-41');

Результат запроса на рисунке 22.

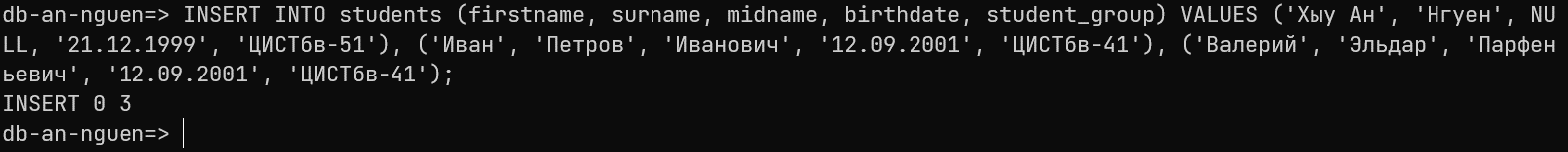


Рисунок 22

Теперь для таблицы grade запрос со вложенным запросом

INSERT INTO grade (semester, subject, educator, value, student\_id) VALUES

(3, 'А и СД', 'Армер', 4, (SELECT id FROM students WHERE surname='Нгуен')),

(3, 'ОИВ', 'Докторова', 5, (SELECT id FROM students WHERE surname='Нгуен')),

(3, 'БД', 'Родионов', 3, (SELECT id FROM students WHERE surname='Нгуен')),

(3, 'ММТ', 'Докторова', 5, (SELECT id FROM students WHERE surname='Нгуен')),

(4, 'Организация ЭВМ и систем', 'Кандаулов', 4, (SELECT id FROM students WHERE surname='Нгуен')),

(4, 'ПТИ', '', 3, (SELECT id FROM students WHERE surname='Нгуен')),

(4, 'ТП', 'Родионов', 3, (SELECT id FROM students WHERE surname='Нгуен')),

(4, 'Ин.яз.', 'Тарасова', 5, (SELECT id FROM students WHERE surname='Нгуен')),

(4, 'БД', 'Родионов', 3, (SELECT id FROM students WHERE surname='Нгуен')),

(4, 'БД', 'Родионов', 4, (SELECT id FROM students WHERE surname='Петров')),

(6, 'ОС', 'Бармина', 4, (SELECT id FROM students WHERE surname='Петров')),

(6, 'ИКСС', 'Ефимов', 4, (SELECT id FROM students WHERE surname='Филиппов'));

Результат запроса представлена на рисунке 23.

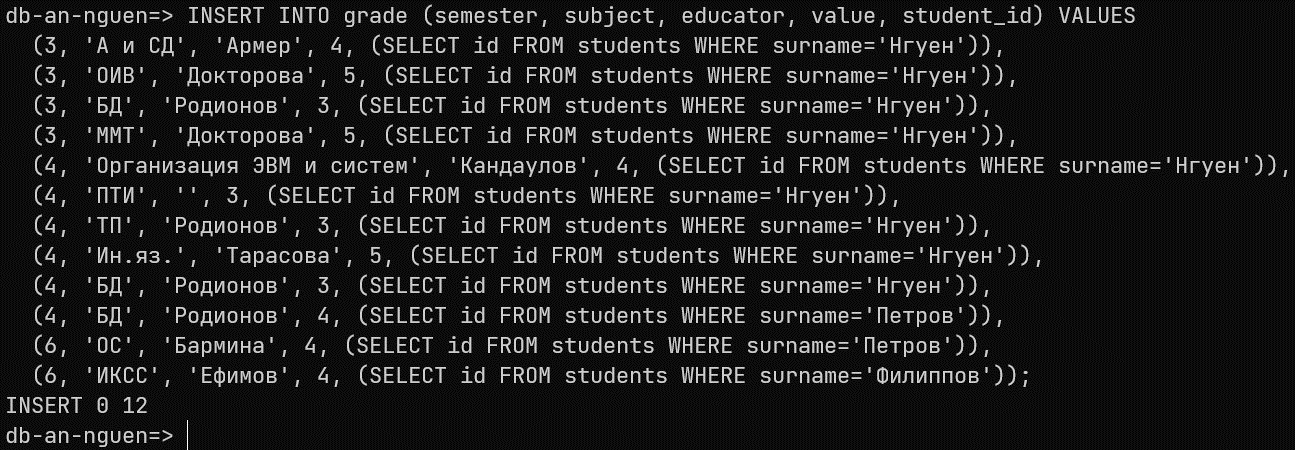


Рисунок 23

Далее нужно сделать запрос выборки собственных оценок. Для этого нужно использовать команду SELECT с операторами LEFT JOIN … ON …, которая берёт все записи из 1 таблицы (эта таблица указывается после оператора FROM), и совпадающие записи из 2 таблицы (указывается после LEFT JOIN) по конкретному условию (совпадение по внешнему ключу). Выполняем запрос выборки

SELECT g.semester, g.subject, g.educator, g.value FROM grade g LEFT JOIN students s ON s.id = g.student\_id WHERE s.surname = 'Нгуен';

Результат выборки изображена на рисунке

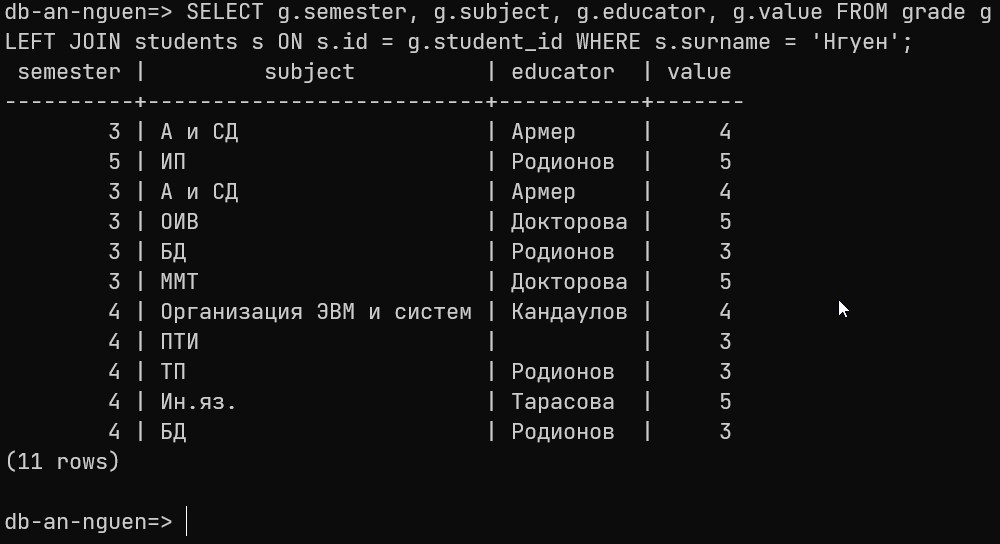


Рисунок 24

Вывод: в этой лабораторной работе создали кластер СУБД PostgreSQL, настроили сетевые правила, подключились к СУБД через консольную программу psql на виртуальной машине с ОС на базе Linux, и затем выполнили базовые SQL запросы, такие как создание таблиц, добавление в него данных и выборку таблиц.