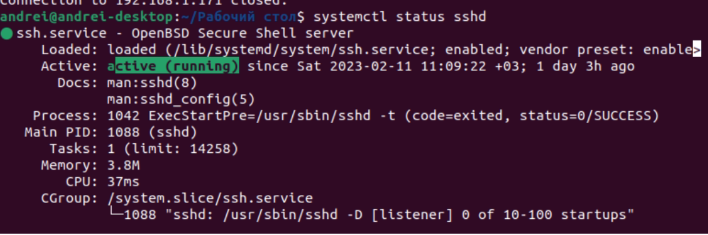


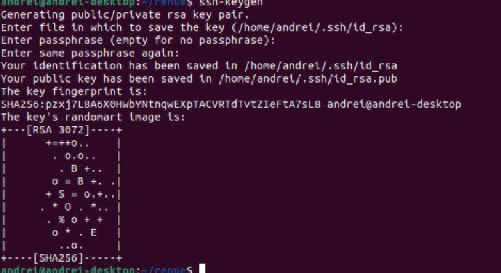
Вначале необходимо установить ssh сервер на СЕРВЕР А командой: sudo apt update | sudo apt-get install openssh-server и проверить установку командой: systemctl status sshd



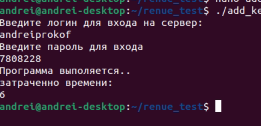
Также для взаимодействия с командой строкой удаленных серверов необходимо установить «expect» командой: sudo apt-get install expect.

После установки пакетов выполним на сервере А команду: ssh-keygen

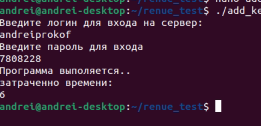
Подтвердим предложенные окна нажав enter и успешно сформируем ключи ssh.



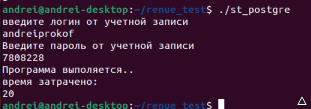
Затем необходимо запустить скрипт add\_key командой: **./add\_key**. Он добавит наш сформированный ssh ключ на сервер "B" и перезапустит службу SSH, после чего нам уже не потребуется указывать логин пароль при подключении. При открытии скрипта будет запрошен логин пароль по которому подключаться к удаленному серверу, и он же будет использован для пользователя sudo. Также по завершению выполнения в терминал выйдет информация по затраченному времени на этот скрипт. В последующих скриптах применен тот же принцип, а именно запрос пароля для sudo пользователя и логина для подключения по ssh. Также каждому исполняемому файлу были добавлены права доступа командой chmod 744 <namefile>.



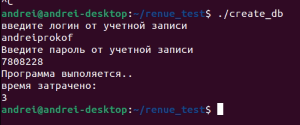
Затем выполняем скрипт add\_user командой: **./add\_user**. Он добавит пользователя DevOps и сделает возможным использование sudo без пароля на удаленном сервере B.

,

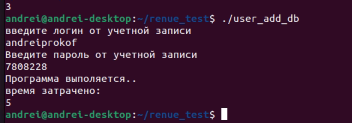
После чего устанавливается postgresql, запускается скрипт командой: **./st\_postgre**.



После выполнения скрипта по установке postgresql создадим базу данных myapp и myauth скриптом: **./create\_db**



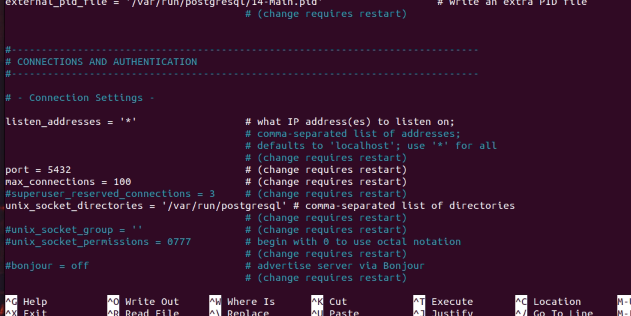
После чего добавим в базу данных пользователя develooper и дадим ему права на чтение и запись к myapp , и на чтение к myauth, запустим скрипт командой: **./user\_add\_db**



Чтобы выполнить задачу по настройке доступа пользователя developer только с сервера "С" необходимо:

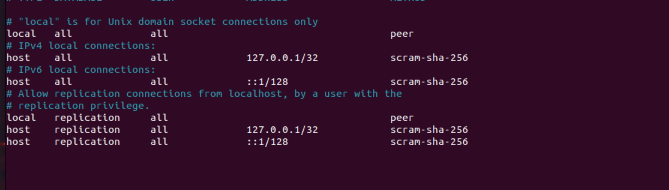
1. в терминале после выполнения предыдущего шага написать andreiprokof@192.168.1.171, что соответствует конструкции ssh username@host\_ip.

2. после чего вызовем файл /etc/postgresql/14/main/postgresql.conf в текстовом редакторе nano или vim. sudo nano /etc/postgresql/14/main/postgresql.conf. В разделе Connection Settings нужно раскомментировать строку listen\_addresses и привести ее к виду: listen\_addresses = '\*', после чего сохрнать файл и выйти (для nano ctrl + s , ctrl + x)

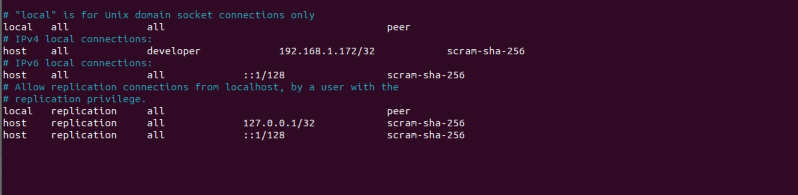


3. После сохранения предыдущего файла откроем файл /etc/postgresql/14/main/pg\_hba.conf

Найдем строки с скриншота



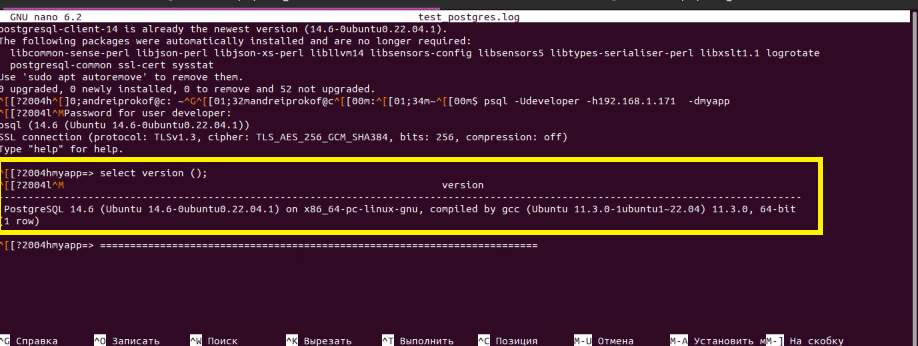
и изменим ip-адрес на сервер "С" в разделе Ipv4 , в данном случае 192.168.1.172/32 и раздел пользователя. Сохраним наш файл.



После выполненных действий перезагрузим postgresql Командой: sudo service postgresql restart. Теперь возможно подключиться к postgresql только с сервера "С" и только под пользователем developer. После выполненных действий выйдем с удаленного сервера командой: exit.

Для выполнения задания по доступности удаленного входа по SSH только с сервера "А" будет настроен iptables на сервере "С" и "B" и заблокирован доступ 22 порта для всех ip адресов кроме сервера А. Вначале запустим файл **./access\_blocking** затем файл **./iptables\_for\_c**. После запуска этих файлов будет установлен iptables добавлены правила входящих подключений для сервера "B" и "С".

Для проверки выполнения запроса с сервера "С" был установлен клиент postgresql и выполнен запрос. Для выполнения нужно запустить **./test\_postgresql**. После чего в логах (файл test\_postgresql.log) можно посмотреть выполнения нашего запроса и успешного ответа на него.



Таким образом были выполнены все 6 поставленных задач.

Порядок запуска:

1. add\_key

2. add\_user

3. st\_postgre

4. create\_db

5. user\_add\_db

6. Ручная настройка блокировки postgresql

7 access\_blocking

8. iptables\_for\_c

9. test\_postgresql