

Telegram-бот для удалённого управления серверами с использованием SSH и базы данных PostgreSQL

В.А. Васькин, С.А. Ямашкин

*Национальный исследовательский Мордовский государственный университет
им. Н.П. Огарева, Саранск*

Аннотация: Статья посвящена разработке telegram-бота для удаленного управления серверами. Исследование было направлено на реализацию приложения, которое было написано на языке Python с использованием Telegram API и PostgreSQL для хранения и манипуляции данными. Система направлена на повышение эффективности процессов в области работы системных администраторов, обеспечивая пользователям простой способ управления серверами через Telegram. Созданный бот позволит пользователям контролировать сервера, выполнять нужные команды удаленно, получать актуальную информацию и мониторинг состояния серверов в реальном времени.

Ключевые слова: python, ssh, it-сервис, telegram-api, postgresql, psycopg2, aiogram, sql, бот, администрирование, удаленное управление.

Введение

Современные требования к администрированию серверов нередко предполагают возможность удаленного управления, что подразумевает не только эффективное, но и безопасное взаимодействие с серверными ресурсами. Программное обеспечение для удаленного управления серверами – это система, позволяющая эффективно управлять виртуальными машинами. Использование мессенджера Telegram, скорость его работы и богатство API делают его оптимальной платформой для интеграции бизнес-процессов. Пользователям предлагается продукт, реализующий кроссплатформенный удаленный доступ к серверам вне зависимости от местоположения или удалённости от рабочего места. Кроме того, удаленное управление серверами востребовано в организационных системах, деятельность которых распределена на значительной территории.

Проектирование базы данных

В основе разработки систем с большим количеством данных ключевым этапом является проектирование базы данных. За хранение и обработку данных о серверах, группах удаленного доступа и пользователях, взаимодействующих с системой, будет отвечать PostgreSQL – реляционная база данных [1].

Проектирование базы данных напрямую связано с требованиями проекта. Таблица «users» хранит в себе данные о пользователях, их уникальный идентификатор в мессенджере Telegram, «servers» включает в себя информацию о серверах. За распределения прав пользователей к серверам отвечают таблицы `remote_access_group` и `host_group`. Такая иерархия таблиц обеспечивает логическую организацию данных и обеспечивает гибкость для будущего расширения функционала. Структура БД, ключевые сущности и связи между ними представлены на ER-диаграмме.

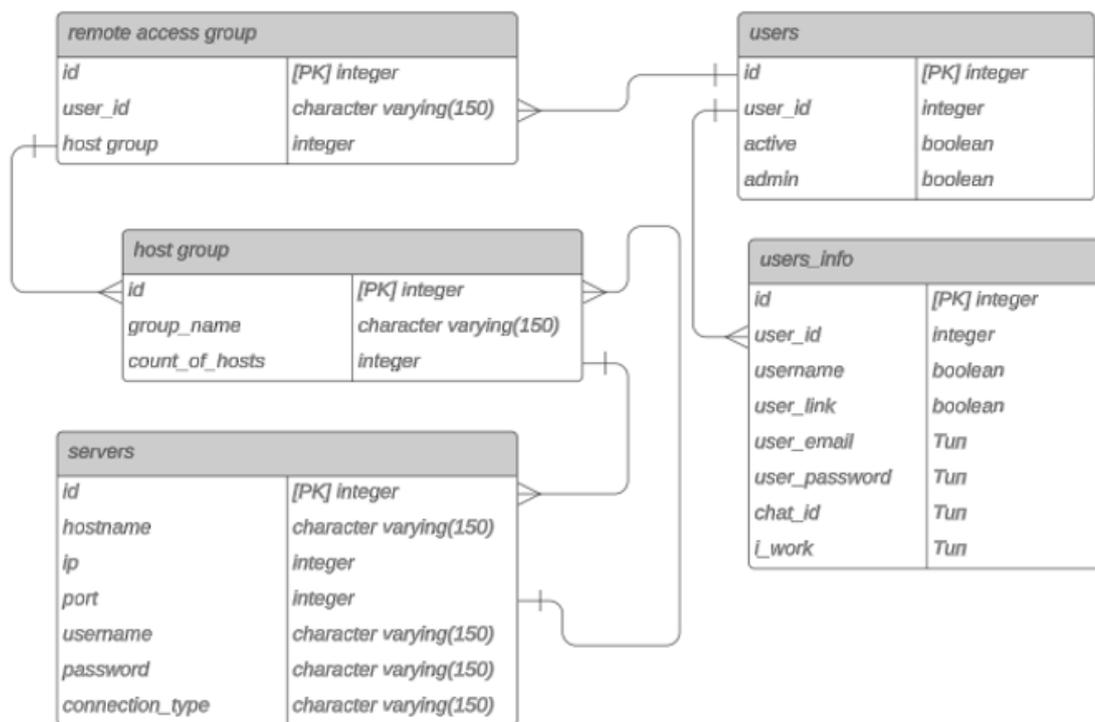


Рис. 1. – ER-диаграмма

Внесение данных в таблицы реализуется через Telegram API. Администратор может добавлять новых пользователей, предоставляя доступ посредством внесения telegram-id, а также управлять серверами, группами серверов и доступами к ним.

Для взаимодействия с базой данных PostgreSQL было достигнуто посредством библиотеки psycopg2, предоставляющей набор инструментов для выполнения SQL-запросов, управления данными и обеспечения безопасное взаимодействие с базой [2].

Например, запрос на добавление нового сервера в таблицу servers, для осуществления дальнейшей работы с ним. Перед добавлением бот проверит, способен ли он подключиться к этому серверу посредством приватного SSH-ключа. Если сервер недоступен – пользователь увидит сообщение об ошибке.

Основные функции системы

Функциональные возможности системы, предусматриваемые работой приложения:

- Добавление и удаление пользователей;
- Изменение групп удаленного доступа для подключения;
- Управление сеансами пользователя во время подключения по SSH.

Ключевым компонентом системы управления задачами является функция подключения пользователя к серверу, доступный только авторизованным пользователям, имеющим доступ к виртуальной машине.

Для реализации метода удаленного подключения пользователя к серверу необходима инициализация SSH-соединения [3,4]. Подключение к серверу происходит по данным сервера и пользователя. В БД должны храниться корректные IP-адрес и порт выбранного сервера. От пользователя необходим доступ к серверу посредством имени пользователя и паролю или приватному ключу [5].

В случае, когда аутентификация на уровне SSH проходит успешно, пользователю отправляется уведомление о успешном подключении [6]. В случае ошибки аутентификации или другой ошибки выводится соответствующее сообщение.

При вводе пользовательских команд производится тщательное анализирование их содержания. В случае обнаружения команды «exit», происходит корректное завершение SSH-соединения с последующим уведомлением об этом факте. При наличии команд, начинающихся с «cd», происходит определение новой директории и обновление текущего местоположения пользовательских файлов.

После входных данных и обработки команд происходит выполнение этих команд в текущей директории на сервере с использованием SSH [7]. Обработка вывода пользовательского запроса проводится в зависимости от общей длины вывода. При обнаружении информации, превышающей 3000 символов, происходит отправка данных в виде вложенного файла с уведомлением. В случае краткого вывода - информация передается, как отдельное сообщение.

Этот подход играет важную роль в эффективной коммуникации между пользователями и удаленным сервером, обеспечивая отправку файлов и сообщений в ответ на команды пользователей. Этот функционал призван улучшить процесс взаимодействия и предоставления результатов выполненных операций на сервере.

Такой механизм взаимодействия с удаленным сервером через Telegram бота позволяет пользователям управлять директориями, выполнять команды и получать качественный результат их выполнения [8]. Совокупное воздействие предоставляет пользователям гибкость и наглядность в управлении серверными операциями, в отличие от традиционных способов управления удаленными серверами.

Демонстрация работы приложения

После запуска бота пользователю, вошедшему под учетной записью администратора, предоставляется два типа управления: управление пользователями и управление серверами. Администратор может добавлять, удалять и просматривать пользователей системы. Для добавления на вход боту подается ссылка на аккаунт нового пользователя, серверная часть приложения, посредством Telegram API [9,10], записывает username и telegram-id в таблицу users. Данные таблицы позволят в дальнейшем пользователям получать доступ к серверам.

Посредством команды /add_user в базу данных добавляется новый клиент приложения, а /list выводит всех пользователей системы, предоставляя ссылки на аккаунты.

Управление серверами включает в себя добавление новых серверов и удалённое управление ими посредством имитации командной строки в чате Telegram. Процесс добавления нового сервера предполагает использование команды /add_server, ввод администратором информации о виртуальной машине, а именно IP-адрес, порт и название сервера для его дальнейшего отображения в системе. В процессе добавления нового сервера бот проверит его доступность с использованием приватного ключа. Если попытка подключения не удастся, бот выведет сообщение об ошибке. Это поможет оперативно выявить и устранить возможные проблемы с доступом к серверу по приватному ключу.

Подключение к серверу происходит посредством команды /start. Бот предоставит список из inline кнопок, каждую из которых можно выбрать для удалённого подключения к серверу. Это позволит легко и быстро установить соединение с нужным сервером без необходимости вводить дополнительные данные или выполнять сложные настройки.

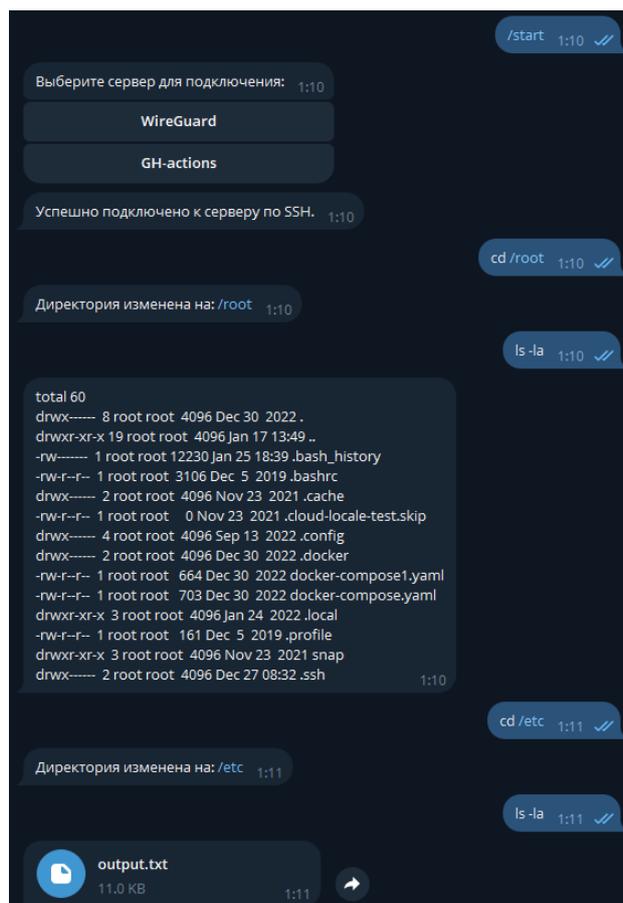


Рис. 2 – Подключение к удалённому серверу

После сообщения об успешном подключении, строка ввода сообщения Telegram становится терминалом виртуальной машины. Она позволяет удалённо пользоваться функционалом перехода по директориям, выполнения удаленных запросов к командной строке, отображая вывод команды ответным сообщением. На случай, когда команды используют интерактивный режим, в программное обеспечение зашит триггер на проверку введенных команд. Таким образом пользователь не сможет внутри терминала открыть файл для редактирования или в реальном времени посмотреть статистику сервера. В случае, если вывод выполненной команды превышает предел одного сообщения в Telegram, бот отправит файл, в котором результат выполнения команды на удалённом сервере будет приведен полностью.

```
output (5).txt - Блокнот
Файл  Правка  Формат  Вид  Справка
-rw-r--r-- 1 root root 2584 Feb 1 2020 gai.conf
drwxr-xr-x 2 root root 4096 Aug 24 2021 groff
-rw-r--r-- 1 root root 766 Apr 22 2022 group
-rw-r--r-- 1 root root 752 Jan 24 2022 group-
drwxr-xr-x 2 root root 4096 Nov 23 2021 grub.d
-rw-r----- 1 root shadow 638 Apr 22 2022 gshadow
-rw-r----- 1 root shadow 627 Jan 24 2022 gshadow-
drwxr-xr-x 3 root root 4096 Aug 24 2021 gss
-rw-r--r-- 1 root root 5060 Aug 21 2019 hdparm.conf
-rw-r--r-- 1 root root 109 Nov 23 2021 hetzner-build
-rw-r--r-- 1 root root 92 Dec 5 2019 host.conf
-rw-r--r-- 1 root root 10 Jan 24 2022 hostname
-rw-r--r-- 1 root root 545 Jan 24 2022 hosts
-rw-r--r-- 1 root root 411 Nov 23 2021 hosts.allow
-rw-r--r-- 1 root root 711 Nov 23 2021 hosts.deny
drwxr-xr-x 2 root root 4096 Jan 24 2022 init
drwxr-xr-x 2 root root 4096 Jan 10 06:32 init.d
drwxr-xr-x 5 root root 4096 Aug 24 2021 initramfs-tools
-rw-r--r-- 1 root root 1748 Feb 25 2020 inputrc
drwxr-xr-x 4 root root 4096 Aug 24 2021 iproute2
drwxr-xr-x 2 root root 4096 Jul 28 2023 iscsi
-rw-r--r-- 1 root root 26 Aug 4 2021 issue
-rw-r--r-- 1 root root 19 Aug 4 2021 issue.net
drwxr-xr-x 5 root root 4096 Nov 23 2021 kernel
```

Рис. 3 – Содержимое файла output

Работа на удалённом сервере через бот создает гибкую и эффективную среду для выполнения операций, мониторинга и обработки данных, что способствует оптимизации процессов управления виртуальными машинами и повышению производительности в рабочей среде.

Заключение

Статья представляет собой описание процесса разработки системы для управления серверами, рассматривает ключевые этапы реализации программного обеспечения.

Удалённое управление серверами играет важную роль в организационных системах, деятельность которых распределена на значительной территории. Следует отметить, что разработанная система обладает функциональными возможностями, которые необходимы в сфере системного администрирования. Гибкость разработанного приложения позволяет упростить способы работы с удалёнными виртуальными машинами посредством использования кроссплатформенного Telegram, а пользователи приложения смогут иметь доступ к серверу с любого устройства вне зависимости от удалённости рабочего места.



Литература

1. Таланов В. М., Федосин С. А. Проектирование информационных систем и баз данных / – Саранск: Издательство СВМО, 2006. 72 с. ISBN 5-7103-0590-1.
2. Лазарев Ю.Г., Талипова Л.В., Черкашин А.В., Гребенюк Е.А., Огурцов Г.Л. Методика создания базы данных ТСОДД на базе PostgreSQL и IndorTrafficPlan //Путевой навигатор. 2022. №. 52. С. 26-33.
3. Коптуров Д. А., Альхимович М. В. Централизованная система управления доступом к linux-серверам по протоколу ssh //Межвузовская научно-техническая конференция студентов, аспирантов и молодых специалистов им. ЕВ Арменского. 2019. С. 109-110.
4. Ракчеев А. Ю. Работа протокола ssh на практике //Научный журнал. 2020. №. 3 (48). С. 28-31.
5. Анзина А. В., Медведева А. Д., Лапина М. А. Исследование аутентификации в протоколе SSH //Студенческая наука для развития информационного общества. 2019. С. 224-232.
6. Бондаренко Е. С. Сравнительный анализ безопасной передачи данных с помощью протоколов telnet и ssh //Аллея науки. 2017. №. 5. С. 526-529.
7. Thomas L., Bhat S. A. Comprehensive Overview of Telegram Services-A Case Study //International Journal of Case Studies in Business, IT and Education (IJCSBE). 2022. V. 6. №. 1. pp. 288-301.
8. George A.S., Sagayarajan S. Securing cloud application infrastructure: understanding the penetration testing challenges of IaaS, PaaS, and SaaS environments // Partners Universal International Research Journal. 2023. V. 2. №. 1. pp. 24-34.

9. Gunawan T. S., Babiker A.B.F., Effendi M.R., Ismail N. Development of intelligent telegram chatbot using Natural Language Processing // 2021. 7th International Conference on Wireless and Telematics (ICWT). IEEE, 2021. pp. 1-5.

10. Khaund T., Hussain M.N., Shaik M., Agarwal N. Telegram: Data collection, opportunities and challenges // Annual international conference on information management and big data. Cham : Springer International Publishing, 2020. pp. 513-526.

References

1. Talanov V. M., Fedosin S. A. Proektirovanie informacionnyx sistem i baz dannyx [Design of information systems and databases]. Saransk : Izdatelstvo SVMO, 2006. 72 p. ISBN 5-7103-0590-1.

2. Lazarev Yu.G., Talipova L.V., Cherkashin A.V., Grebenyuk E.A., Ogurczov G.L. Putevoj navigator. 2022. №. 52. pp. 26-33.

3. Kopturov D. A., Alximovich M. V. Centralizovannaya sistema upravleniya dostupom k linux-serveram po protokolu ssh [Centralized access control system for Linux servers via the SSH protocol]. Mezhvuzovskaya nauchno-texnicheskaya konferenciya studentov, aspirantov i molodyx specialistov im. EV Armenskogo. 2019. pp. 109-110.

4. Rakcheev A. Yu. Nauchnyj zhurnal. 2020. №. 3 (48). pp. 28-31.

5. Anzina A. V., Medvedeva A. D., Lapina M. A. Issledovanie autentifikacii v protokole SSH [Research of authentication in the SS protocol]. Studencheskaya nauka dlya razvitiya informacionnogo obshhestva. 2019. pp. 224-232.

6. Bondarenko E. S. Alleya nauki. 2017. №. 5. pp. 526-529.

7. Thomas L., Bhat S. International Journal of Case Studies in Business, IT and Education (IJCSBE). 2022. V. 6. №. 1. pp. 288-301.

8. George A.S., Sagayarajan S. Partners Universal International Research Journal. 2023. V. 2. №. 1. pp. 24-34.



9. Gunawan T. S., Babiker A.B.F., Effendi M.R., Ismail N. Development of intelligent telegram chatbot using Natural Language Processing. 2021 7th International Conference on Wireless and Telematics (ICWT). IEEE, 2021. pp. 1-5.

10. Khaund T., Hussain M.N., Shaik M., Agarwal N. Telegram: Data collection, opportunities and challenges. Annual international conference on information management and big data. Cham : Springer International Publishing, 2020. pp. 513-526.

Дата поступления: 17.11.2024

Дата публикации: 26.01.2025