

Домашнее задание к занятию Система хранения данных - кеши, очереди, файловая система»

В качестве результата пришлите ответы на вопросы в личном кабинете студента на сайте netology.ru.

Описание

Эти домашние задания — лабораторные работы, в которых вы по инструкциям выполните действия.

Задание «Redis ACL»

Вы разберёте новый механизм ACL, появившийся в Redis 6.0.

В качестве целевой конфигурации используйте файл `docker-compose.yml` содержания:

```
version: '3.7'
services:
  cache:
    image: redis:6
    ports:
      - 6379:6379
    volumes:
      - ./redis.conf:/usr/local/etc/redis/redis.conf:ro
    command: ["redis-server", "/usr/local/etc/redis/redis.conf"]
```

Описание настроек:

1. `volumes` позволяет файл конфигурации `redis.conf`, расположенный рядом с вашим `docker-compose.yml`, подключить в контейнер по пути `/usr/local/etc/redis/redis.conf` в режиме только для чтения (`ro`).
2. `command` позволяет переопределить команду запуска, запустив сервер Redis с конфигурационным файлом, который мы подключили на предыдущем шаге.

Важно. Все файлы есть в каталоге `assets`:

1. `docker-compose.yml`;
2. `redis.conf`.

Вам нужно скачать их на то устройство, на которой будете запускать `Docker`.

О Command.

Инструкция `command` (`CMD` в Dockerfile) может записываться в трёх формах:

1. `[executable, options...]`: exec form — аналог того, что мы делали с вызовом `Command` в Go: shell не используется, запускается только исполняемый файл с переданными параметрами, которые никак не интерпретируются. Вспомните разницу между `exec` и `sh -c exec`.
2. `[options...]`: params to ENTRYPOINT — в большинстве кастомизируемых образов пишется специальный Shell script (см. PostgreSQL, Redis), который запускается при старте контейнера и умеет обрабатывать передаваемые аргументы, фактически, получается: `docker-entrypoint.sh options...`.
3. `command options...`: shell form — противоположность первой форме. Используется shell, все поля интерпретируются, то есть обрабатывает: `/bin/sh -c command options...`.

Предпочтительной в `Dockerfile` является первая форма, в Docker Compose всё зависит от того, определена ли в `Dockerfile` инструкция `ENTRYPOINT`. Если определена, то инструкция `command` будет преобразована во вторую форму, если не определена — то в первую форму.

Этапы выполнения

Часть 1. Настройки по умолчанию.

1. Запустите четыре терминала в каталоге, в котором у вас расположен файл `docker-compose.yml`, либо воспользуйтесь tmux.
2. Запустите контейнер Redis в первом терминале с настройками по умолчанию: `docker-compose up`.
3. Во втором терминале подключитесь к нему с помощью команды `docker-compose exec cache redis-cli`.
4. Проверьте, что вы подключены от лица пользователя `default`: `ACL WHOAMI`. `default` — это пользователь по умолчанию, который оставлен для совместимости с предыдущими версиями Redis, когда понятия пользователей не было.
5. Проверьте, что никаких ограничений для вас не установлено: `ACL LIST`.

Вы увидите вывод:

```
1) "user default on nopass ~* +@all"
```

Q. Что он означает?

A. По полям слева направо:

1. `user default` — описывается пользователь с именем `default`.

2. `on` — пользователь включён (`off` — выключен). Режим «включён/выключен» определяет, возможны ли новые подключения, которые попытаются аутентифицироваться под указанным пользователем. Выключение пользователя не обрывает текущие соединения.
3. `nopass` — для аутентификации под данным пользователем не требуется пароля. Будет подходить любой пароль.
4. `~*` — какие ключи доступны данному пользователю. `~` указывает шаблон, `*` — означает, что все.
5. `+@all` — `+` указывает на то, какие команды доступны пользователю, `-` — какие недоступны. `@` указывает категорию, `all` означает, что все категории.

5. Проверьте, что можете устанавливать значения ключей командой: `SET users:admin@localhost id:1|name:vasya`. Это общепринятая стратегия именования ключей для Redis: `object-type:name` или `object-type:name:field`, где вместо `name` может использоваться какое-то поле, например, `login` или `id`.

6. Завершите работу контейнера, нажав `Ctrl + C` в первом терминале, где был запущен `docker-compose up`, и выполните команду `docker-compose rm` для удаления данных контейнера.

Часть 2. Отключение `default`.

1. Отредактируйте файл `redis.conf`: перейдите на 759 строку и добавьте туда запись `user default off`, после сохраните файл.
2. Запустите контейнер Redis в первом терминале: `docker-compose up`.
3. Во втором терминале подключитесь к нему с помощью команды `docker-compose exec cache redis-cli`.
4. Выполните команду `ACL WHOAMI`, запишите вывод. Его нужно будет прислать в результатах выполнения домашнего задания.
5. Выполните команду `ACL LIST`, запишите вывод. Его нужно будет прислать в результатах выполнения домашнего задания.
6. Выполните команду `AUTH default`, запишите вывод. Его нужно будет прислать в результатах выполнения домашнего задания.
7. Завершите работу контейнера, нажав `Ctrl + C` в первом терминале, где был запущен `docker-compose up`, и выполните команду `docker-compose rm` для удаления данных контейнера.

Часть 3. Добавление пользователя `app`.

1. Отредактируйте файл `redis.conf`: перейдите на 760 строку и добавьте туда запись: `user app on #2bb80d537b1da3e38bd30361aa855686bde0eacd7162fef6a25fe97bf527a25b`

`~* +@read +@write`, после сохраните файл.

Q. Что эта запись означает?

A. По полям слева направо:

1. `user app` — описывается пользователь с именем `app`.
 2. `on` — пользователь включён (`off` — выключен).
 3. `#hash` — для аутентификации под данным пользователем требуется пароль, в конфигурационном файле Redis будет храниться хеш-функция SHA-256.
 4. `~*` — какие ключи доступны данному пользователю. `~` указывает шаблон, `*` — означает, что все.
 5. `+@read +@write` — `+` указывает на то, какие команды доступны пользователю, `@` указывает категорию (см. `ACL CAT read` / `ACL CAT write` для отображения конкретного списка команд в этих категориях).
2. Запустите контейнер Redis в первом терминале: `docker-compose up`.
3. Во втором терминале подключитесь к нему с помощью команды `docker-compose exec cache redis-cli`.
4. Аутентифицируйтесь с помощью команды `AUTH app password`, где вместо `password` подставьте пароль, полученный из хеша. Вам нужно будет вспомнить, как доставать из хеша пароли.
5. Выполните команду `ACL WHOAMI`, запишите вывод. Его нужно будет прислать в результатах выполнения домашнего задания.
6. Выполните команду `ACL LIST`, запишите вывод. Его нужно будет прислать в результатах выполнения домашнего задания.
7. Проверьте, что вы можете устанавливать значения ключей командой: `SET users:admin@localhost id:1|name:vasya`. Это общепринятая стратегия именования ключей для Redis: `object-type:name` или `object-type:name:field`, где вместо `name` может использоваться какое-то поле, например, `login` или `id`.
8. Завершите работу контейнера, нажав `Ctrl + C` в первом терминале, где был запущен `docker-compose up`, и выполните команду `docker-compose rm` для удаления данных контейнера.

Часть 4. Включение пользователя `default`.

1. Отредактируйте файл `redis.conf`: перейдите на 759 строку и удалите запись: `user default off`, после сохраните файл.
2. Запустите контейнер Redis в первом терминале: `docker-compose up`.

3. Во втором терминале подключитесь к нему с помощью команды `docker-compose exec cache redis-cli`.
4. Выполните команду `ACL WHOAMI`, запишите вывод. Его нужно будет прислать в результатах выполнения домашнего задания.
5. Выполните команду `ACL LIST`, запишите вывод. Его нужно будет прислать в результатах выполнения домашнего задания.
6. Проверьте, что можете устанавливать значения ключей командой: `SET users:admin@localhost id:1|name:vasya`. Это общепринятая стратегия именования ключей для Redis: `object-type:name` или `object-type:name:field`, где вместо `name` может использоваться какое-то поле, например, `login` или `id`.
7. Завершите работу контейнера, нажав `Ctrl + C` в первом терминале, где был запущен `docker-compose up`, и выполните команду `docker-compose rm` для удаления данных контейнера.

Результаты выполнения задания

Пришлите в личном кабинете студента ответы на вопросы, указанные в разделе «Выполнение»:

1. По части 2: выводы команд `ACL WHOAMI`, `ACL LIST`, `AUTH default`.
2. По части 3: выводы команд `ACL WHOAMI`, `ACL LIST`.
3. По части 4: выводы команд `ACL WHOAMI`, `ACL LIST`.

Примечания

1. Чаще всего в реальной конфигурации настройки безопасности выносят в файл отдельный файл и подключают с помощью инструкции `aclfile /path/to/file/users.acl`. Мы для упрощения сделали всё в `redis.conf`.
2. Детальное описание работы с ACL в Redis есть на странице <https://redis.io/topics/acl>.

Задание «Redis ACL LOGS»*

Важно. Это необязательная задача. Её невыполнение не влияет на получение зачёта по домашнему заданию.

Используя команду `ACL LOG`, пришлите отображение событий безопасности:

- попытка входа с неверным паролем;
- попытка доступа к запрещённой команде.

Важно. Вам нужно вводить команду `ACL LOG`. По умолчанию будет выведено до 10 событий. Если вы введёте `ACL LOG RESET`, то лог будет очищен, и придётся повторять

всю работу заново.

Выполнение задания

Самостоятельно соберите конфигурацию в двух вариантах.

1. У пользователя `default` выставлен пароль `UTJPE6WStAtg="6q` в виде хеш-функции SHA-256, доступны все ключи и все команды, пользователь включен.
2. У пользователя `app` выставлен пароль в виде хеш-функции SHA-256 `#2bb80d537b1da3e38bd30361aa855686bde0eacd7162fef6a25fe97bf527a25b`, доступны все ключи, команды доступны только в категориях `@read`, `@write`, пользователь включен.

Результаты выполнения задания

Пришлите в личном кабинете студента записи из лога событий ACL для событий:

- попытка входа с неверным паролем;
- попытка доступа к неразрешённой команде.