**Домашняя работа №3. Redis**

**Цель:**

В результате выполнения ДЗ вы научитесь разворачивать Redis в кластере, заполнять данными и делать запросы.

Описание/ инструкция выполнения домашнего задания:

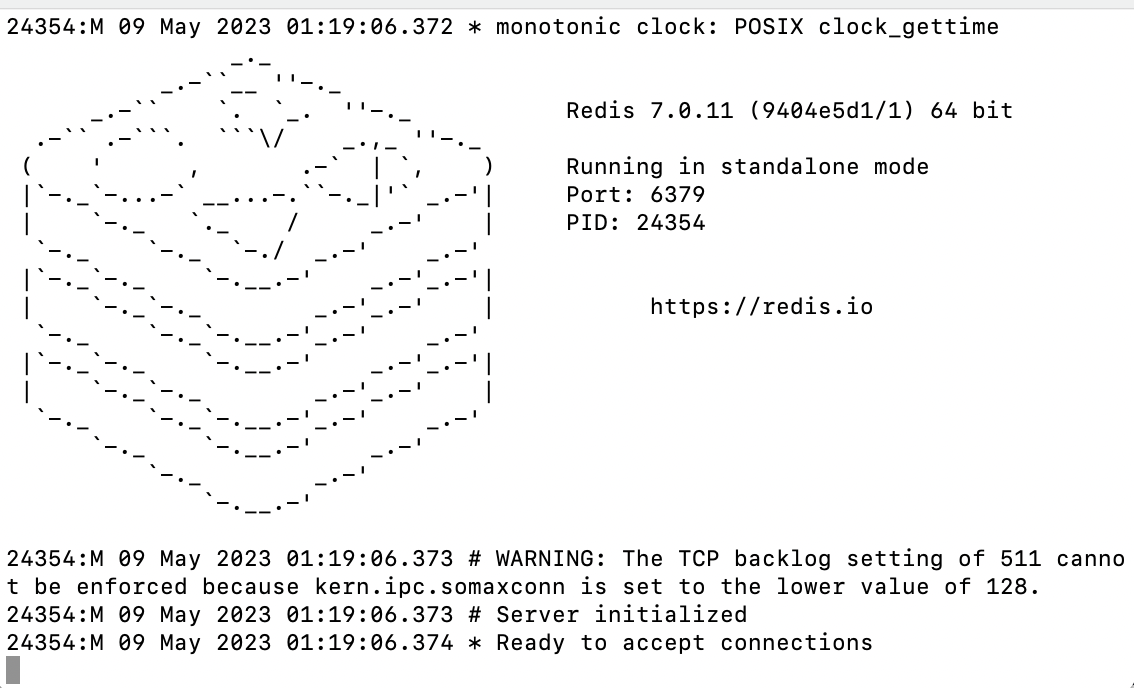
1. Сохранить большой JSON (~20МБ) в виде разных структур - строка, hset, zset, list;

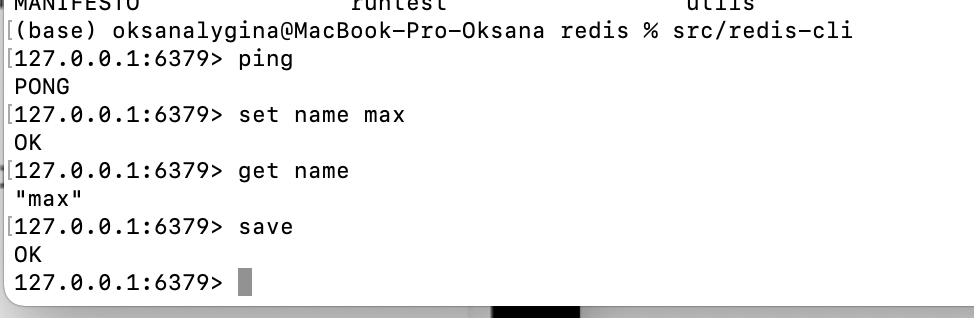
2. Протестировать скорость сохранения и чтения;

3. Настроить редис кластер на 3х нодах с отказоустойчивостью, затюнить таймауты.

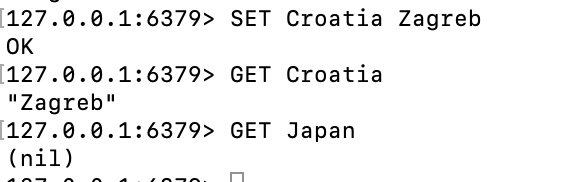
4. Предоставить отчет.

1. Устанавливаем redis





1. Используем команду SET для установки пар ключ-значение:



Redis будет возвращать nil вместо ошибки в случае, если ключ не будет найден

Приведем аналогию с Python:

capitals = {}

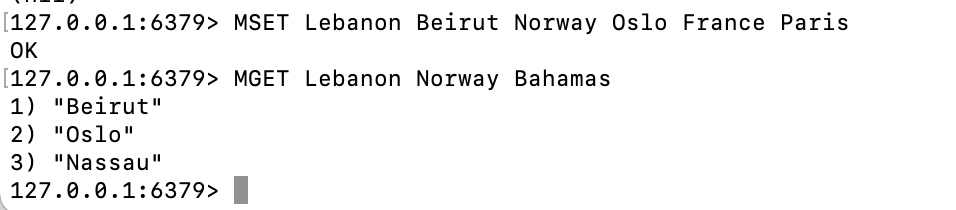
capitals["Bahamas"] = "Nassau"

capitals["Croatia"] = "Zagreb"

print(capitals.get("Croatia"))  *# 'Zagreb'*

print(capitals.get("Japan"))  *# None*

1. Используем команду MSET и MGET для получения множественных пар ключ-значение:



Аналог Python:

capitals.update({

...     "Lebanon": "Beirut",

...     "Norway": "Oslo",

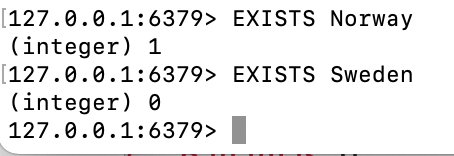
...     "France": "Paris",

... })

>>> [capitals.get(k) **for** k **in** ("Lebanon", "Norway", "Bahamas")]

['Beirut', 'Oslo', 'Nassau']

1. Проверяем, существует ли ключ



Аналог Python:

>>> "Norway" **in** capitals

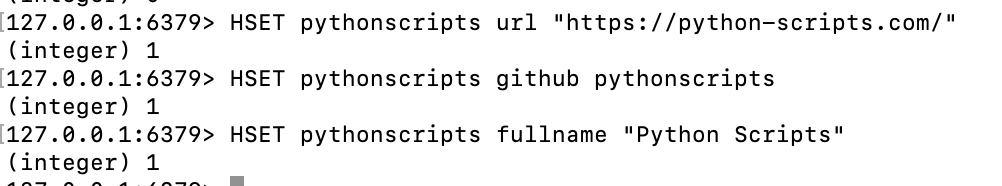
**True**

>>> "Sweden" **in** capitals

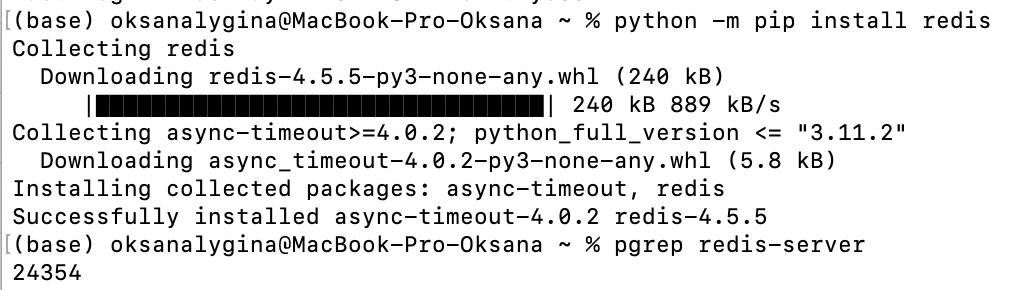
**False**

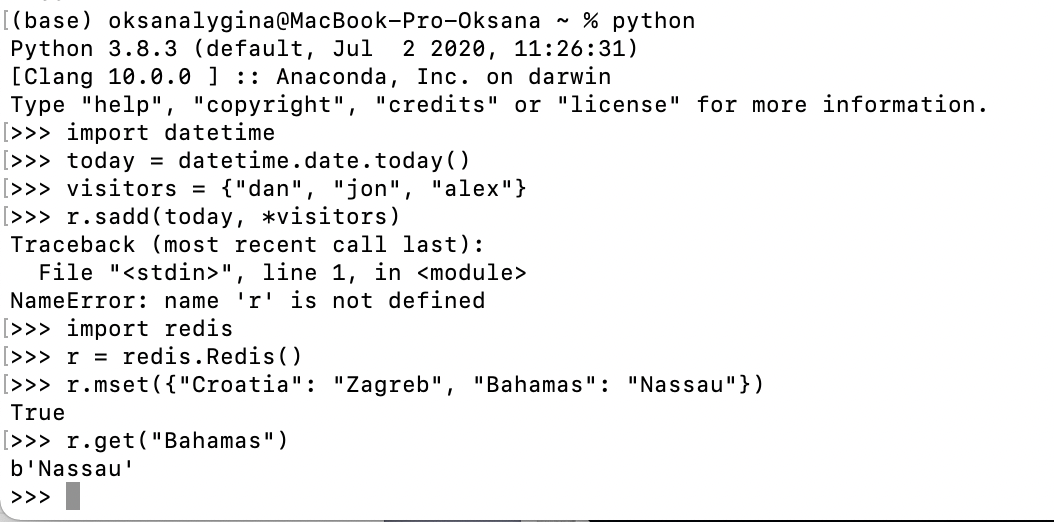
1. Хеш является отображением пары string:string, называемой парами поле-значение, и находится под одним ключом:

Настроем три пары поле-значение на один ключ python-scripts

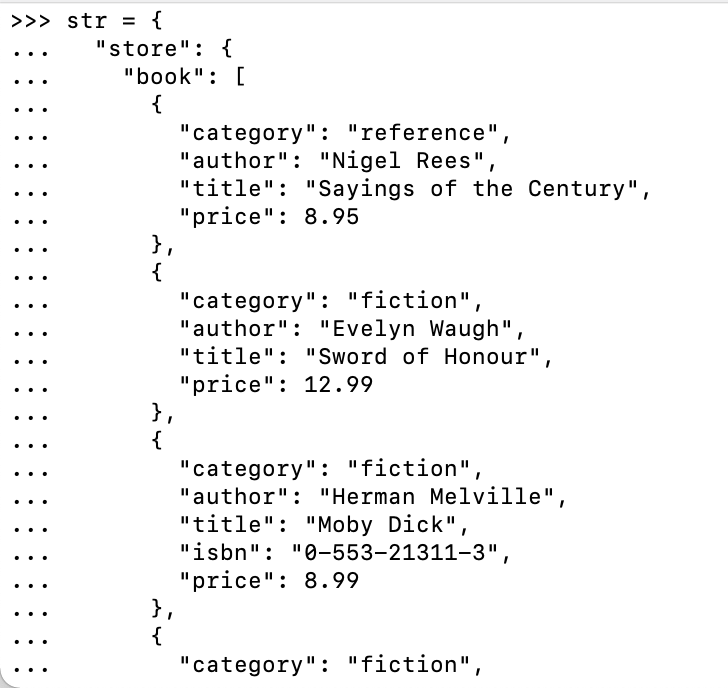
****

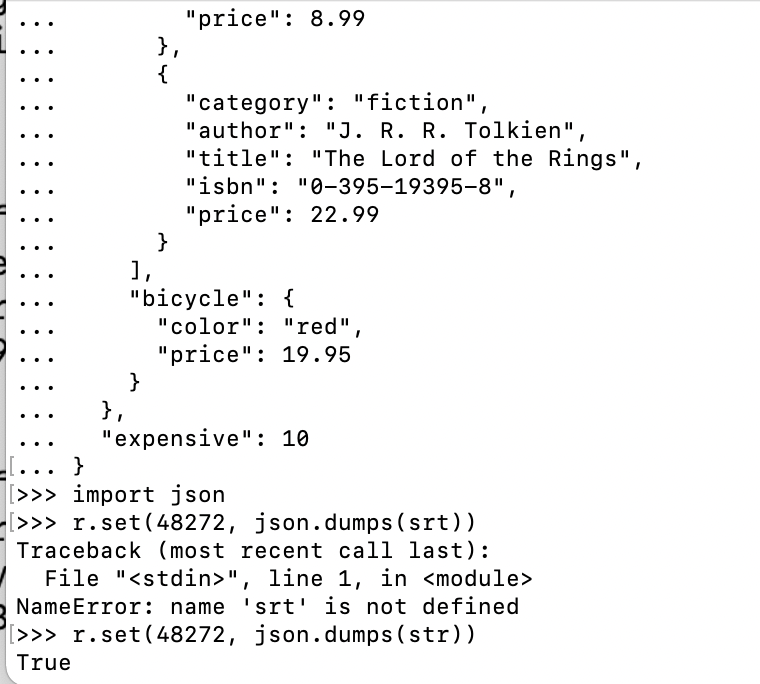
1. **Теперь перейдем в python оболочку:**

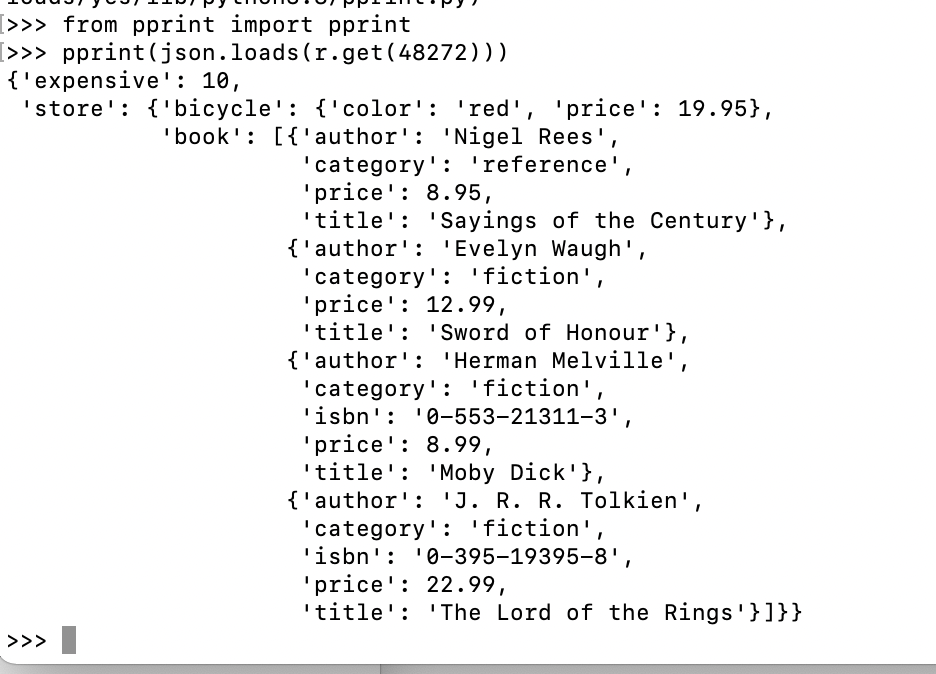
****

****

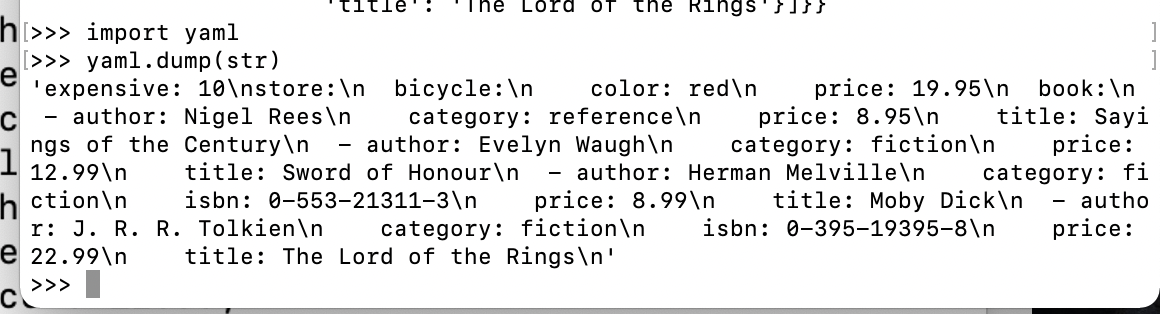
1. **Наш json:**

****

****

****

**В другом виде:**

****