

Beyond the Cache with Redis + Node.js



redis





Guy Royse

Developer Advocate



 @guyroyse

 github.com/guyroyse

 guy.dev

Getting Started



Assumptions

I am assuming that...

...you know modern JavaScript.

...you have a basic understanding of Express.

...you know absolutely nothing about Redis.

What Will We Cover?

We will cover...

- ...installing Redis and RedisInsight.
- ...using Redis commands.
- ...using Redis from Node.js.
- ...using RedisJSON & RediSearch.



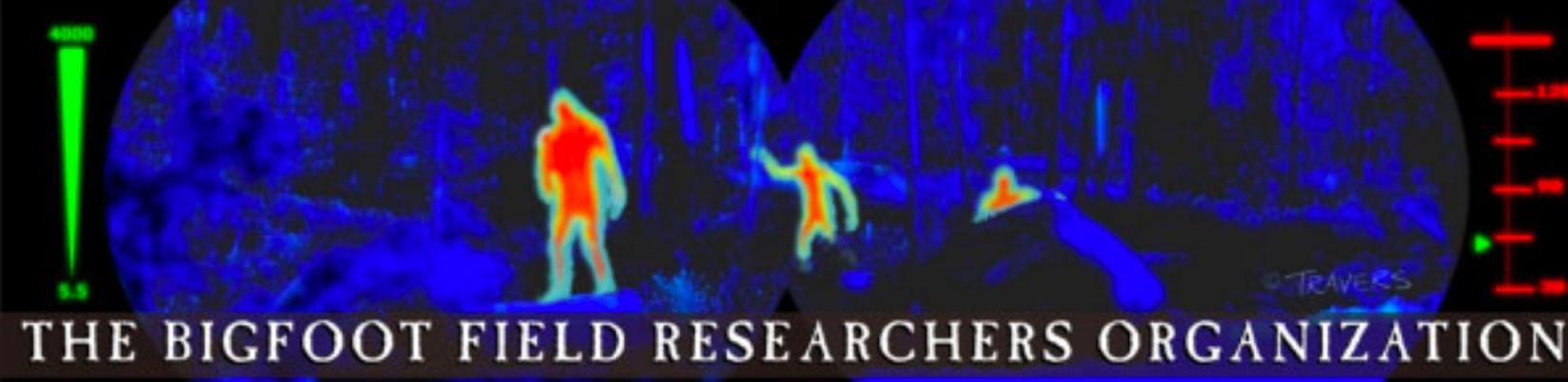


What Are We Doing Here?

We will build a Bigfoot Tracker API using...

- ...Redis Stack.
- ...Node Redis.
- ...Express.

This is a workshop, not a talkshop.



THE BIGFOOT FIELD RESEARCHERS ORGANIZATION

Founded in 1995 -- The only scientific research organization exploring the bigfoot/sasquatch mystery.

Contact us at ContactUs@BFRO.NET or Phone (408) 634-BFRO [408-634-2376]

The Comprehensive Sightings Database



Bigfoot Town Hall
Meetings in USA & Canada



Tim Renner
@timothyrenner

Following

FOLLOWERS FOLLOWING LIBRARY
443 **8** **5**

I do geo data science at HomeAway. Autistic.
Obsessed with weird data.

HomeAway
 [http://timothyrenner.github.io](https://timothyrenner.github.io)

1-5 of 5

Any ▾ Sort ▾



Haunted Places

OPEN

Haunted places in the United States pulled from the Shadowlands Haunted Places Index.

Dataset • Updated Jan 28 • Public Domain License

[paranormal, geography](#)

4 Comment



Bigfoot Sightings

OPEN

Full text and geocoded sighting reports from the Bigfoot Field Researchers Organization (BFRO).

Dataset • Updated Dec 2, 2017 • Public Domain License

[geography, clustering, sasquatch](#)

59 Comment



UFO Sightings

OPEN

Full text and geocoded UFO sighting reports from the National UFO Research Center (NUFORC).

Dataset • Updated Oct 25, 2017 • Public Domain License

[ufo, paranormal, geography](#)

13 Comment

RECENT COMMENTS

Regarding the UFO sightings - MUFON is probably a good place to start - <http://www.mufon.com/> It's possible they have a direct download available.
in Understanding Bigfoot Sightings/Understanding Bigfoot Sightings

Is the timestamp field in the CSV file incomplete? I was messing with this a while back and it seemed okay to me, but I didn't dig in too deep.
in Understanding Bigfoot Sightings/Analysis

Also, there's code for mapping sightings out in this notebook: https://github.com/timothyrenner/bfro_sightings_data/tree/master/notebooks I did a clustering analysis for my blog too:
<https://timothyrenner.github.io/datascience/2017/06/30/finding-bigfoot.html>
in Understanding Bigfoot Sightings/Understanding Bigfoot Sightings

@ninja The JSON file has state and county fields, and it joins to the geolocated reports on the report number. That might be easier than a spatial join against a county/state shapefile.
in Understanding Bigfoot Sightings/Understanding Bigfoot Sightings

I'm glad you like it - I'm happy to join the conversation!
in Bigfoot Sightings/BFRO Sightings data

Bigfoot Field Researchers Organization Data

ID	8086
Date	15-Jan-1958
Title	A series of large, human-like footprints are found on a farm near Wayne National Forest
Observed	While during some yard chores, we noticed a series of tracks going into the hollow on our property. Upon examination...
Classification	Class B
County	Noble
State	Ohio
Latitude	39.63382
Longitude	-81.40079
Location Details	Closest town was Harriettville. Closest main road is State Route 145. Right on the border of Noble and Washington Counties.

Dark Sky Weather Data

High/Mid/Low Temperature	34.0 / 30.0 / 26.0 °F
Humidity & Dew Point	93% / 31.2°F
Cloud Cover	100%
Moon Phase	86%
Precipitation Intensity	6.7 mm/hr
Precipitation Probability	100%
Precipitation Type	snow
Air Pressure	1011.09 millibars
Summary	Light snow (< 1 in.) starting in the afternoon.
UV Index	1
Visibility	2.62 miles
Wind Bearing/Speed	344° @ 10.12 mph

Getting Started



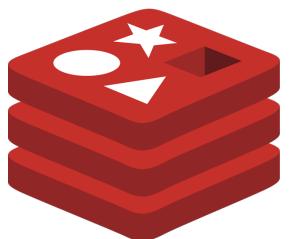
Full Instructions on GitHub

**[github.com/guyroyse/
beyond-the-cache*](https://github.com/guyroyse/beyond-the-cache)**

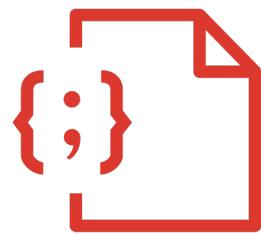
* This link will never give you up nor let you down.

How to Get Redis Stack

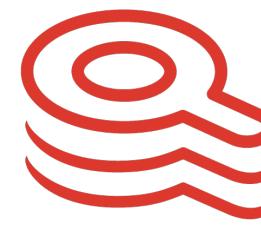
What is Redis Stack?



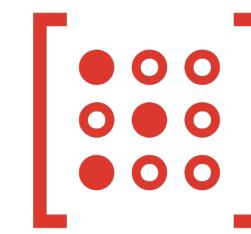
Redis



RedisJSON



RedisSearch



RedisGraph



RedisTimeSeries



RedisBloom



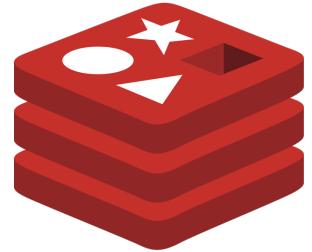
	Filter by Key Name or Pattern	🔍	
		☰	✖
		+ Key	🔧
Total: 4 586	Last refresh: 16 min	🕒	16 min
JSON	bigfoot:sighting:01G9QB55X0V9VWWQW5YTXSNT88	No limit	2 KB
JSON	bigfoot:sighting:01G9QB431AFAMBAFKV4S8G8VZE	No limit	2 KB
JSON	bigfoot:sighting:01G9QB4X07H1VRW24XGDE9E3K6	No limit	2 KB
JSON	bigfoot:sighting:01G9QB4ZD9TMOK9FAAEV6EP9B	No limit	2 KB
JSON	bigfoot:sighting:01G9QB5F3HXN36DPWRECM506ZG	No limit	3 KB
JSON	bigfoot:sighting:01G9QB41E01Q3RVZ76D3Z4BKDH	No limit	997 B
JSON	bigfoot:sighting:01G9QB4BSKVHE23R3S7AGERJF7	No limit	2 KB
JSON	bigfoot:sighting:01G9QB4ETAX15C8D8XE574BF8X	No limit	3 KB
JSON	bigfoot:sighting:01G9QB4GKFNYCQZAW9TF9VS9PR	No limit	3 KB
JSON	bigfoot:sighting:01G9QB52XJBTJGF44C5HYJS8M7	No limit	2 KB
JSON	bigfoot:sighting:01G9QB5PVE1CRADGYXQPJYYFJB	No limit	1 KB
JSON	bigfoot:sighting:01G9QB5QJCRJ34RZ12ETFN6VMF	No limit	3 KB
JSON	bigfoot:sighting:01G9QB4CZ4V1XVYW2PW4N877C9	No limit	4 KB
JSON	bigfoot:sighting:01G9QB40NATC1GJM0A5KH75R6G	No limit	4 KB
JSON	bigfoot:sighting:01G9QB4ZHKP5BCMAZV7HE4FGM1	No limit	10 KB

JSON bigfoot:sighting:01G9QB55X0V9VWWQW5YTXSNT88

Key Size: 2 KB Length: 10 TTL: No limit 16 min

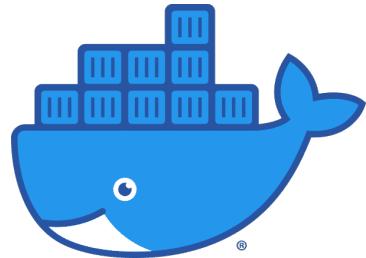
```
{  
  "id": "01G9QB55X0V9VWWQW5YTXSNT88"  
  "reportId": "3801"  
  "title": ""  
  "date": ""  
  "observed": "Two white big foot spotted in open field from car. These creatures were appx. seven and nine feet tall. The shorter a gorilla shape and female, the taller a more human shaped muscular brute head more rounded than female also. Froze in mid stride as deer will do when caught in similar spot. As a rural resident and outdoorsman I was not shocked by them at all. This area has wolverine and cougar in it that once were deemed extinct from Mn. Did not get out to investigate as I was lightly armed and in awe of their size. Would compare to Clydesdale horses in height. Would presume the creatures were in annual migration from wisconsin just like the bears. You will find reports on them to be more active moving east to west from mid August to October. Bears den on the cold [north] side of a slope. So this bigfoot likely uses the south side. Once had hell scared out of my dog by a small black creature that appeared to be sitting on ground with legs crossed eating. This was so near dark and before I saw the big ones that I assumed that the shadows in the forest "created" the little guy. Now I'm not so sure. Anyway!!!! If you want to photo bigfoot I suggest the area around the St. Croix river at dusk in early autumn. The French explorers traders called this
```

Installing Redis Stack



Redis Cloud

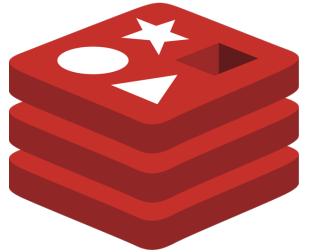
The \$200 credit you get by using code STACK200.



Docker

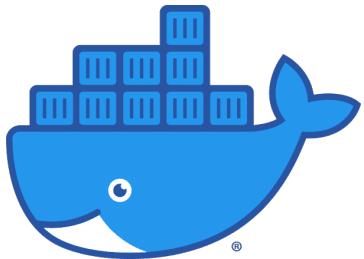
It's easier.

Installing RedisInsight



Redis Cloud

Download and install from redis.com/redis-enterprise/redis-insight/.

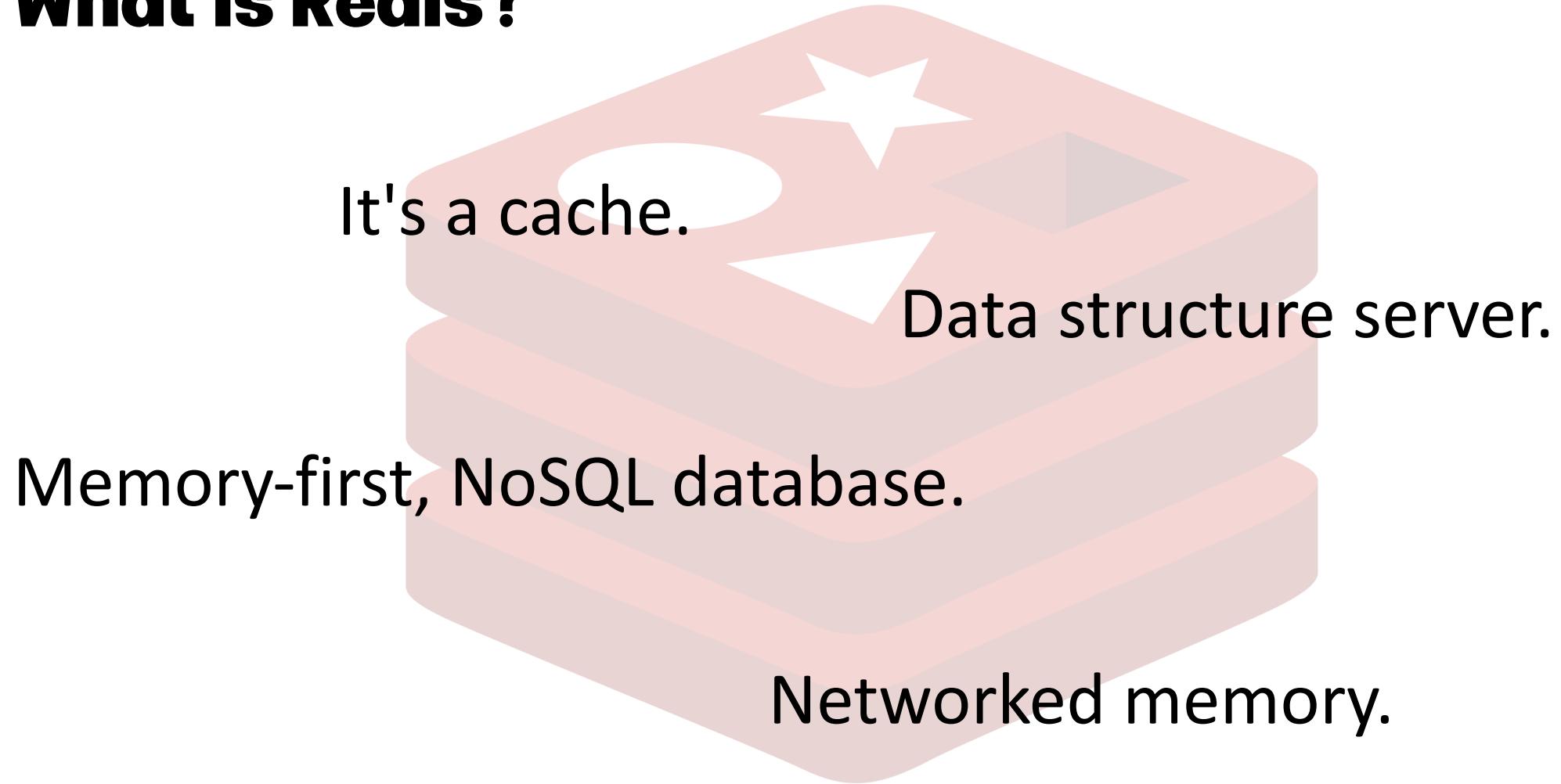


Docker

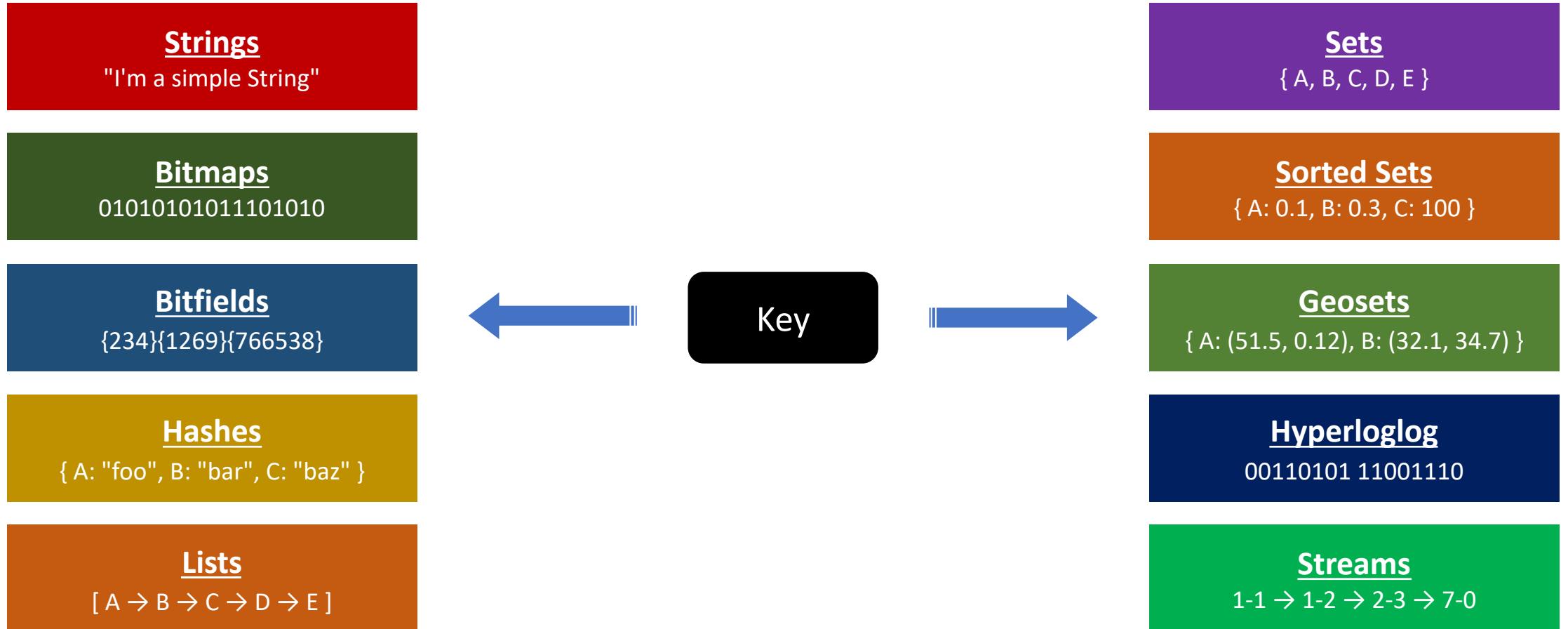
It's included.

Redis 101

What is Redis?



Data Structures



Sending Redis Commands

SET motd Greetings!

```
*3\r\n$3\r\nSET\r\n$4\r\nmotd\r\n$10\r\nGreetings!\r\n
```

OK

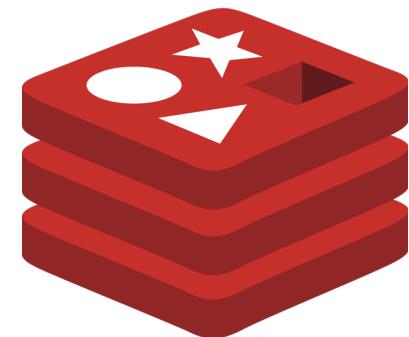
+OK\r\n

GET motd

```
*2\r\n$3\r\nGET\r\n$4\r\nmotd\r\n
```

"Greetings!"

\$10\r\nGreetings!\r\n





Persistence

RDB

redis.conf
save 3600 1
save 300 100
save 60 10000

127.0.0.1> BGSAVE

AOF

redis.conf
appendonly yes
appendfsync always | everysec | no
auto-aof-rewrite-percentage 100
auto-aof-rewrite-min-size 64mb

127.0.0.1> BGREWRITEAOF

Eviction and Expiration

DEL

```
> SET foo 1  
OK  
> DEL foo  
Memory freed  
(integer) 1  
> GET foo  
(nil)
```

UNLINK

```
> SET foo 1  
OK  
> UNLINK foo  
(integer) 1  
> GET foo  
(nil)  
...  
...  
...
```

Memory freed

EXPIRE

```
> SET foo 1  
OK  
> EXPIRE foo 10  
(integer) 1  
> GET foo  
"1"  
About 10 sec later..  
> GET foo  
(nil)
```

Eviction

```
> SET foo 1  
OK  
> SET bar 2  
OK  
> GET foo  
(nil)
```

**LRU*

Eviction Options

Eviction

```
> SET foo 1  
OK  
> SET bar 2  
OK  
> GET foo  
(nil)
```

*LRU

- volatile-lru**: Evict using approximated LRU, only keys with an expire set.
- allkeys-lru**: Evict any key using approximated LRU.
- volatile-lfu**: Evict using approximated LFU, only keys with an expire set.
- allkeys-lfu**: Evict any key using approximated LFU.
- volatile-random**: Remove a random key having an expire set.
- allkeys-random**: Remove a random key, any key.
- volatile-ttl**: Remove the key with the nearest expire time (minor TTL)
- noeviction**: Don't evict anything, just return an error on write operations.

Building the Bigfoot Tracker API

How Are We Building This?

We will build...

...a REST API.

...using Express.js.

...using Node Redis.

...using Redis Stack.

...with RedisJSON & RediSearch.



A Proper REST API

Method	URL	Description
GET	/sightings	Gets all the things. A collection.
POST	/sightings	Add a thing to the collection. <u>Not</u> idempotent.
GET	/sightings/12345	Get one of the things in the collection.
PUT	/sightings/12345	Adds or replaces a thing in the collection. Idempotent.
PATCH	/sightings/12345	Modifies an existing thing.
DELETE	/sightings/12345	Removes a thing.

Using Node Redis

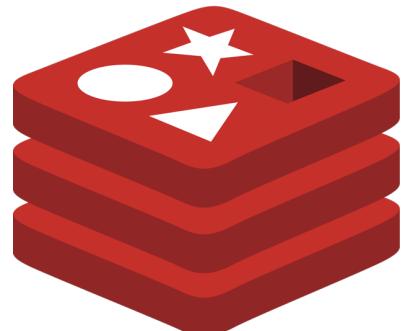
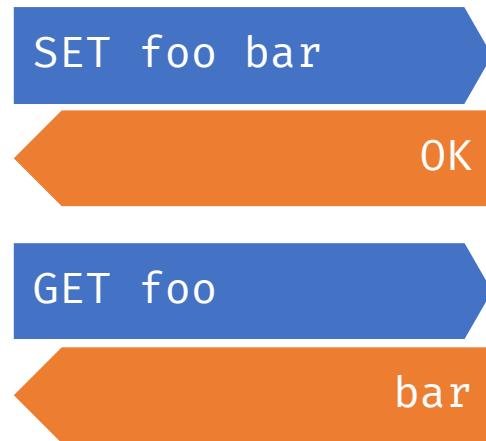
```
import { createClient } from 'redis'

const url = 'redis://alice:foobared@awesome.redis.server:6380'
const redis = createClient({ url })

await redis.connect()

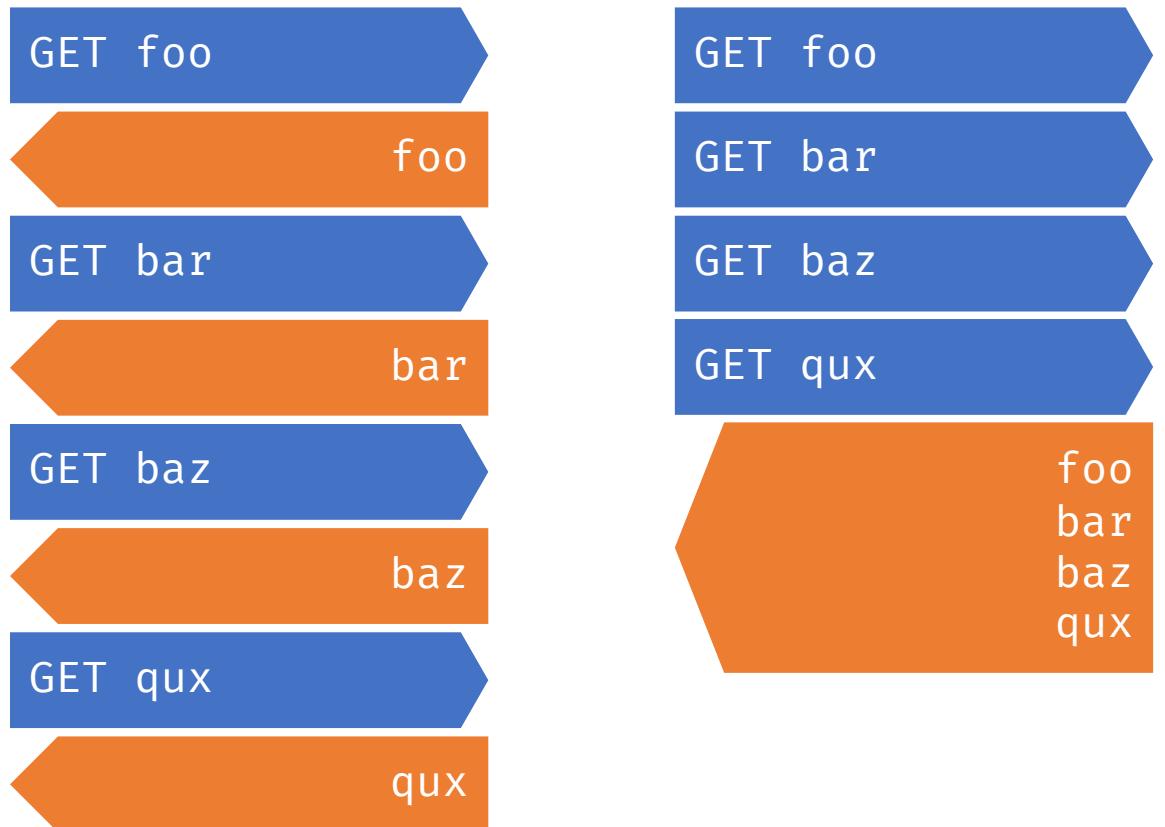
await redis.set('foo', 'bar')

const foo = await redis.get('foo')
```



Pipelining & Transactions

Pipelining



Pipelining with Node Redis

```
redis.set('foo', 'foo')
redis.set('bar', 'bar')
redis.set('baz', 'baz')
redis.set('qux', 'qux')
```

```
redis.set('foo', 'foo')
redis.set('bar', 'bar')
redis.set('baz', 'baz')
await redis.set('qux', 'qux')
```

```
const results = await Promise.all([
  redis.get('foo'),
  redis.get('bar'),
  redis.get('baz'),
  redis.get('qux')
])
```



Transactions

Transactions...

- ...must happen on the same connection.
- ...queue a series of commands.
- ...execute atomically.



Transaction Commands

MULTI: Starts a transaction.

EXEC: Executes the queued commands.

DISCARD: Discards the queued commands.

WATCH: Starts an optimistic lock on a key.

UNWATCH: Removed a lock on a key.



A Simple Example

```
> MULTI  
OK
```

```
(TX)> SADD bigfoot:sightings:ohio 8086  
QUEUED
```

```
(TX)> INCR bigfoot:sightings:count  
QUEUED
```

```
(TX)> HSET bigfoot:sighting:8086 id 8086 state Ohio  
QUEUED
```

```
(TX)> EXEC  
1) (integer) 1  
2) (integer) 1  
3) (integer) 2
```



Discarding

```
> MULTI  
OK
```

```
(TX)> SADD bigfoot:sightings:ohio 8086  
QUEUED
```

```
(TX)> INCR bigfoot:sightings:count  
QUEUED
```

```
(TX)> HSET bigfoot:sighting:8086 id 8086 state Ohio  
QUEUED
```

```
(TX)> DISCARD  
OK
```



Syntactic Errors Fail

```
> MULTI  
OK
```

```
(TX)> SADD bigfoot:sightings:ohio 8086  
QUEUED
```

```
(TX)> INCR bigfoot:sightings:count  
QUEUED
```

```
(TX)> HSTE bigfoot:sighting:8086 id 8086 state Ohio  
(error) ERR unknown command `HSTE`, with args beginning  
with: `bigfoot:sighting:8086`, `id`, `8086`,  
`state`, `Ohio`,
```

```
(TX)> EXEC  
(error) EXECABORT Transaction discarded because of  
previous errors.
```



Semantic Errors Succeed

```
> MULTI  
OK
```

```
(TX)> SADD bigfoot:sightings:ohio 8086  
QUEUED
```

```
(TX)> INCR bigfoot:sightings:ohio  
QUEUED
```

```
(TX)> EXEC  
1) (integer) 1  
2) (error) WRONGTYPE Operation against a key holding  
the wrong kind of value
```



Watching for Changes

```
> WATCH bigfoot:sightings:ohio  
OK
```

```
> MULTI  
OK
```

```
(TX)> SADD bigfoot:sightings:ohio 8086  
QUEUED
```

```
(TX)> HSET bigfoot:sighting:8086 id 8086 state Ohio  
QUEUED
```

```
(TX)> EXEC  
1) (integer) 1  
2) (integer) 1  
3) (integer) 2
```





Changes Happen!

> WATCH bigfoot:sightings:ohio
OK



> MULTI
OK

> SADD bigfoot:sightings:ohio 8087
(integer) 1



(TX)> SADD bigfoot:sightings:ohio 8086
QUEUED

(TX)> INCR bigfoot:sightings:count
QUEUED

(TX)> EXEC
(nil)

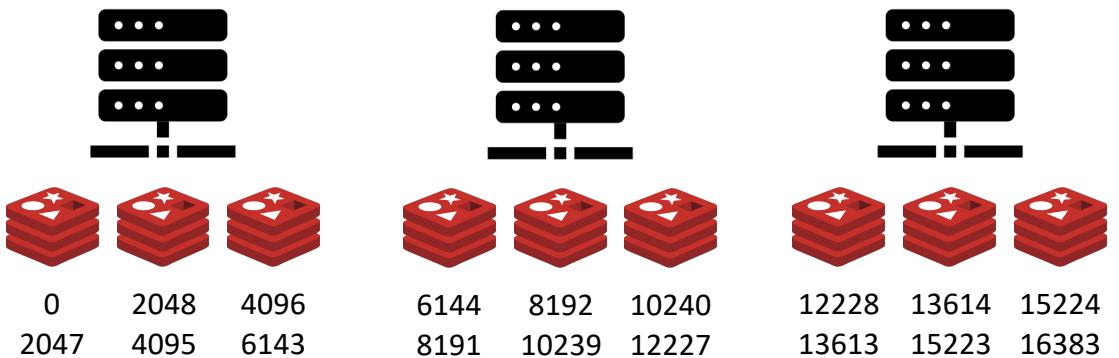
Redis Clusters

A cluster contains many **nodes**.

A nodes contain many **shards**.

A shard contains many **hash slots**.

There are always **16,384 hash slots** in a cluster.



Transactions cannot span hash slots.

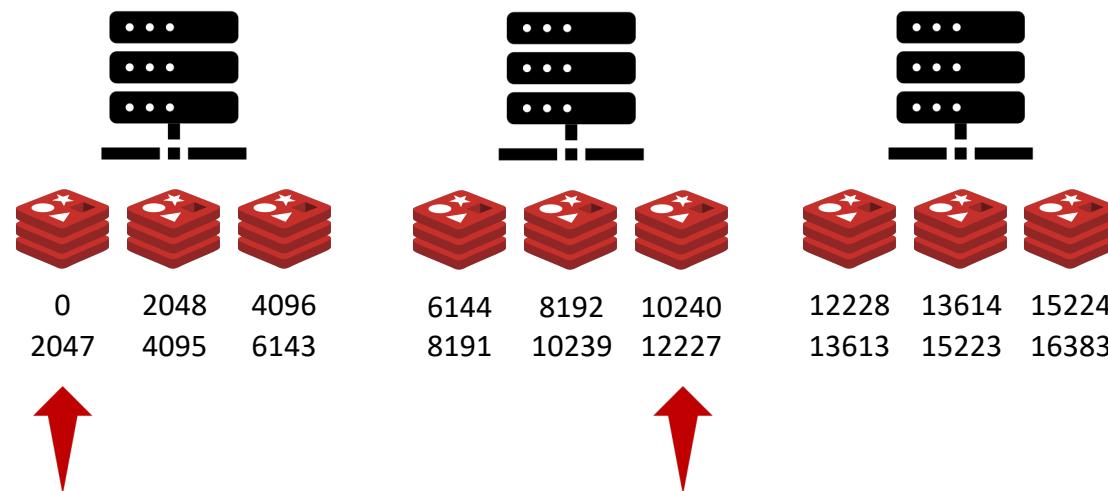
Computing Hash Slots

bigfoot:sightings:ohio

bigfoot:sightings:count

CRC16('bigfoot:sightings:ohio') mod 16384 = 12194

CRC16('bigfoot:sightings:count') mod 16384 = 1827

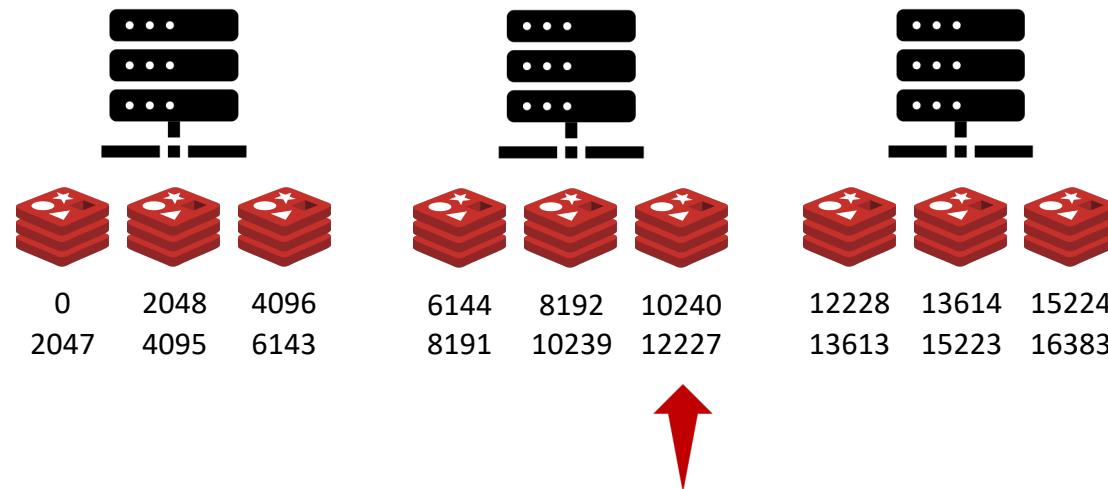


Keeping Keys Together

{bigfoot:sightings}:ohio

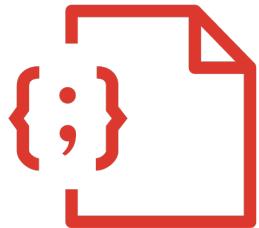
{bigfoot:sightings}:count

CRC16('bigfoot:sightings') mod 16384 = 11876



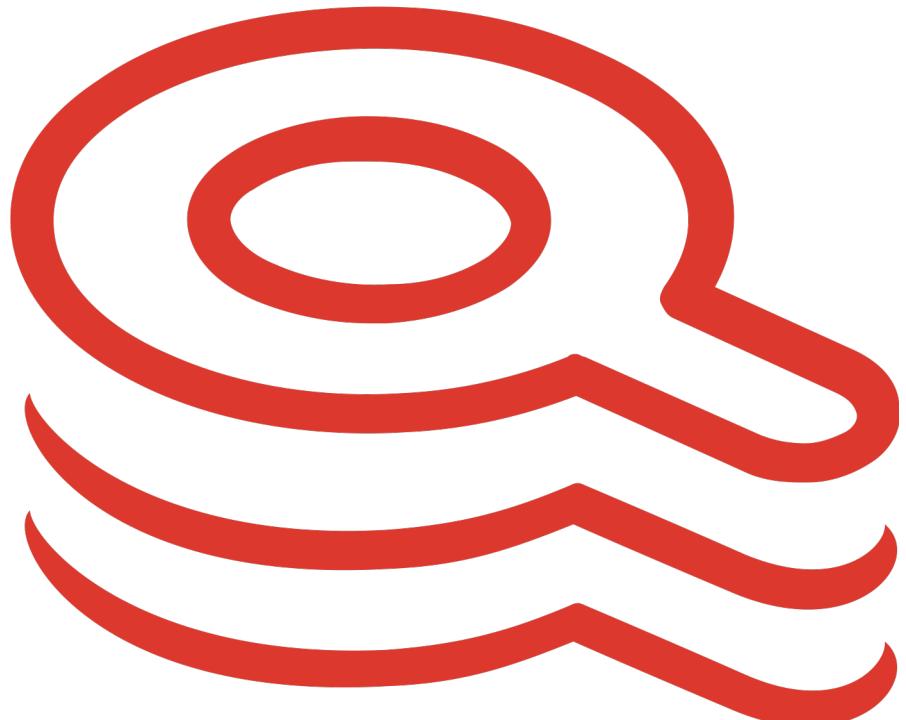
RedisJSON & RediSearch

Using RedisJSON



```
> JSON.SET foo $ '{ "bar":"bar", "baz":42, "qux":true }'  
OK  
  
> JSON.GET foo $  
"[{\\"bar\\":\\"bar\\",\\"baz\\":42,\\"qux\\":true}]"  
  
> JSON.GET foo $.bar  
"[\"bar\"]"  
  
> JSON.GET foo $.bar $.baz $.qux  
"{$.bar:[\"bar\"],$.baz:[42],$.qux:[true]}"
```

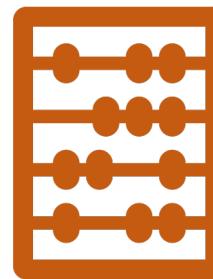
RediSearch



Indexing



Full-text search



Aggregation

Creating Indices with RediSearch

```
FT.CREATE bigfoot:sighting:index
```

```
ON hash
```

```
PREFIX 1 bigfoot:sighting:
```

```
SCHEMA
```

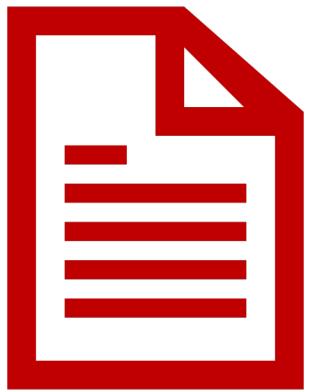
observed	TEXT	SORTABLE
----------	------	----------

Find All the Bigfoots

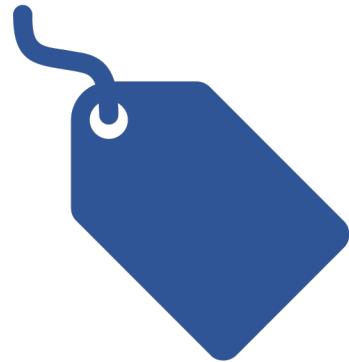
```
FT.SEARCH bigfoot:sighting:index "*" index  
LIMIT 0 3 starting index number of results query  
RETURN 2 id state fields to return
```

- 
- 1) (integer) 4586
 - 2) "bigfoot:sighting:27167"
 - 3) 1) "id"
2) "27167"
 - 3) "state"
 - 4) "Mississippi"
 - 4) "bigfoot:sighting:20002"
 - 5) 1) "id"
2) "20002"
 - 3) "state"
 - 4) "Michigan"
 - 6) "bigfoot:sighting:28711"
 - 7) 1) "id"
2) "28711"
 - 3) "state"
 - 4) "Minnesota"

Four Ways to Find Bigfoot



Text



Tag



Numeric



Geo



Texting Bigfoot

```
> FT.CREATE bigfoot:sighting:index  
ON hash PREFIX 1 bigfoot:sighting:  
SCHEMA  
    title      TEXT  
    observed   TEXT
```

```
> FT.SEARCH bigfoot:sighting:index  
    "creek"  
    "creek river"  
    "creek ~river"  
    "creek | river"  
    "(creek | river) stream"  
    "(creek | river) -stream"  
    "@title:creek @observed:river"
```



Tagging Bigfoot

```
> FT.CREATE bigfoot:sighting:index  
ON hash PREFIX 1 bigfoot:sighting:  
SCHEMA  
state          TAG  
classification TAG
```

```
> FT.SEARCH bigfoot:sighting:index  
"@state:{ Ohio }"  
"@state:{ Ohio | Kentucky }"  
"@state:{ West Virginia }"  
"@classification:{ Class\\ A }"  
"@classification:{ Class\\ A }  
@classification:{ Class\\ B }"
```



Numbering Bigfoot

```
> FT.CREATE bigfoot:sighting:index  
ON hash PREFIX 1 bigfoot:sighting:  
SCHEMA  
    temperature_high NUMERIC  
    temperature_low  NUMERIC
```

```
> FT.SEARCH bigfoot:sighting:index  
    "@temperature_high:[ 60 75 ]"  
    "@temperature_high:[ 60 +inf ]"  
    "@temperature_high:[ -inf 75 ]"  
    "@temperature_high:[ -inf +inf ]"  
    "@temperature_high:[ (60 (75 ]"
```



Locating Bigfoot

```
> FT.CREATE bigfoot:sighting:index  
ON hash PREFIX 1 bigfoot:sighting:  
SCHEMA  
location GEO
```

↙ longitude

-84.5120, 39.1031

latitude ↑



Locating Bigfoot

```
> FT.CREATE bigfoot:sighting:index  
ON hash PREFIX 1 bigfoot:sighting:  
SCHEMA  
location GEO
```

```
> FT.SEARCH bigfoot:sighting:index  
"@location:[ -84.5120 39.1031 50 mi ]"  
"@location:[ -84.5120 39.1031 100 ft ]"  
"@location:[ -84.5120 39.1031 25 m ]"  
"@location:[ -84.5120 39.1031 10 km ]"
```

Some SQL Equivalents

SQL	RediSearch
WHERE x='foo' AND y='bar'	@x:foo @y:bar
WHERE x='foo' AND y!='bar'	@x:foo -@y:bar
WHERE x='foo' OR y='bar'	(@x:foo) (@y:bar)
WHERE x IN ('foo', 'bar','hello world')	@x:(foo bar "hello world")
WHERE y='foo' AND x NOT IN ('foo','bar')	@y:foo (-@x:foo) (-@x:bar)
WHERE x NOT IN ('foo','bar')	-@x:(foo bar)
WHERE num BETWEEN 10 AND 20	@num:[10 20]
WHERE num >= 10	@num:[10 +inf]
WHERE num > 10	@num:[(10 +inf]
WHERE num < 10	@num:[-inf (10]
WHERE num <= 10	@num:[-inf 10]
WHERE num < 10 OR num > 20	@num:[-inf (10] @num:[(20 +inf]
WHERE name LIKE 'john%'	@name:john*



Guy Royse

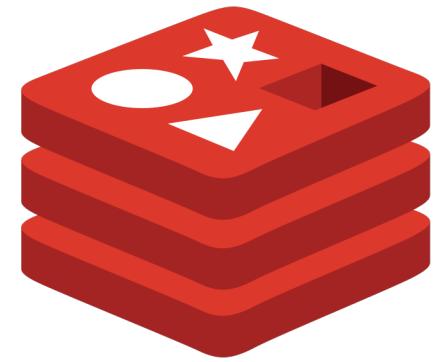
Developer Advocate



 @guyroyse

 github.com/guyroyse

 guy.dev



redis