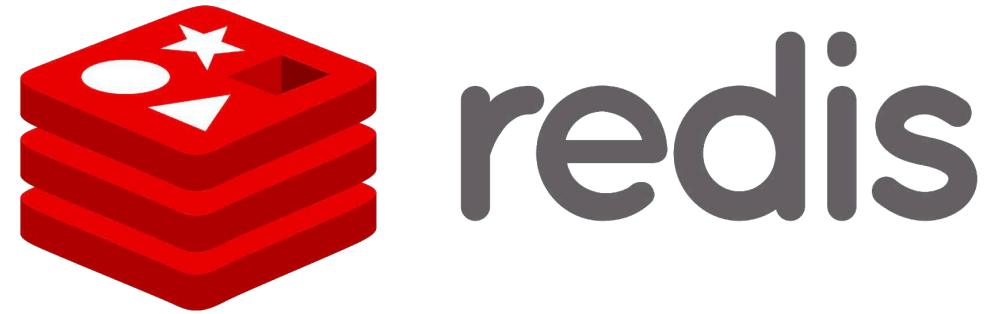


redis

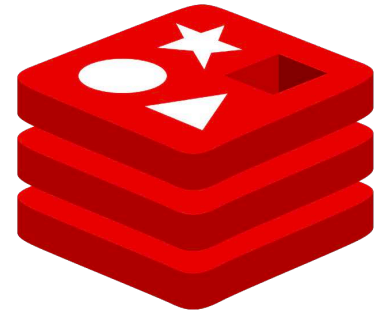
Banco de dados NoSQL - Chave/Valor com Redis

Prof. Gustavo Leitão



REmote **DI**ctionary **S**erver

- Sistema de par chave / valor
- Dados em memória
- Arquitetura Cliente / Servidor
- Escrito em C
- Open Source
- <https://github.com/antirez/redis>



redis

Chave

string

Ex: 5, _key, key

Valor

string

int

float

list

set

sorted

set

hash



string → "um texto"

int → 4

float → 2,3

list → [1, 2, "a", 2, "texto"]

set → (1, 2, "a", "texto")

sortedset → [(1, 2), (2, 4), ("a", 5), ("texto", 20)]

hash →

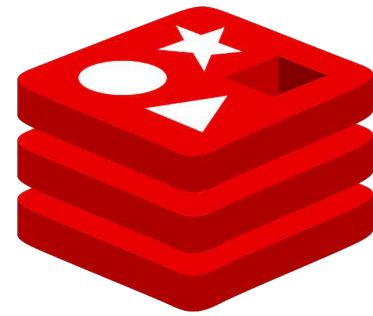
nome	Miguel
idade	34
sexo	Masculino



REDIS é utilizado como cache web



560MM de pageviews / mês
54 site no mundo
40MM de respostas
25 servidores



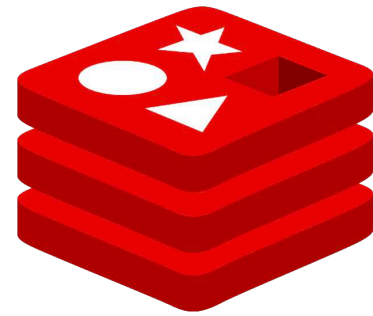
redis

—

REDIS utilizado para armazenar a homeline dos usuários, em listas com tamanho máximo de 800 elementos.



30 bilhões de updates por dia
400.000.000 de tweets por dia
5000 tweets por segundo



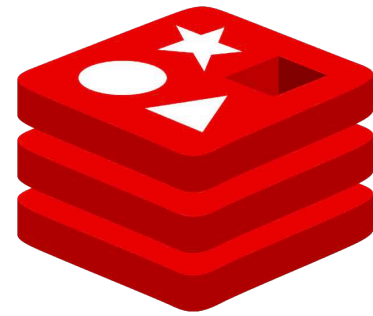
redis

Gestão de configurações de deploy.

Roteamento de requests:
mapeamento de servidores
associados a usuários.

GitHub

3,5 milhões de usuários
6 milhões de repositórios

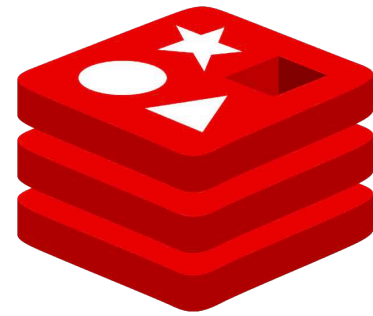


redis

REDIS utilizado como índice complementar ao MySQL, tornando buscas mais rápidas e reduzindo a carga sobre base de dados.

Além disso, usa sistema de PubSub para notificações de eventos em tempo real.

1.6 milhões de uploads por dia
92 milhões de usuários



redis

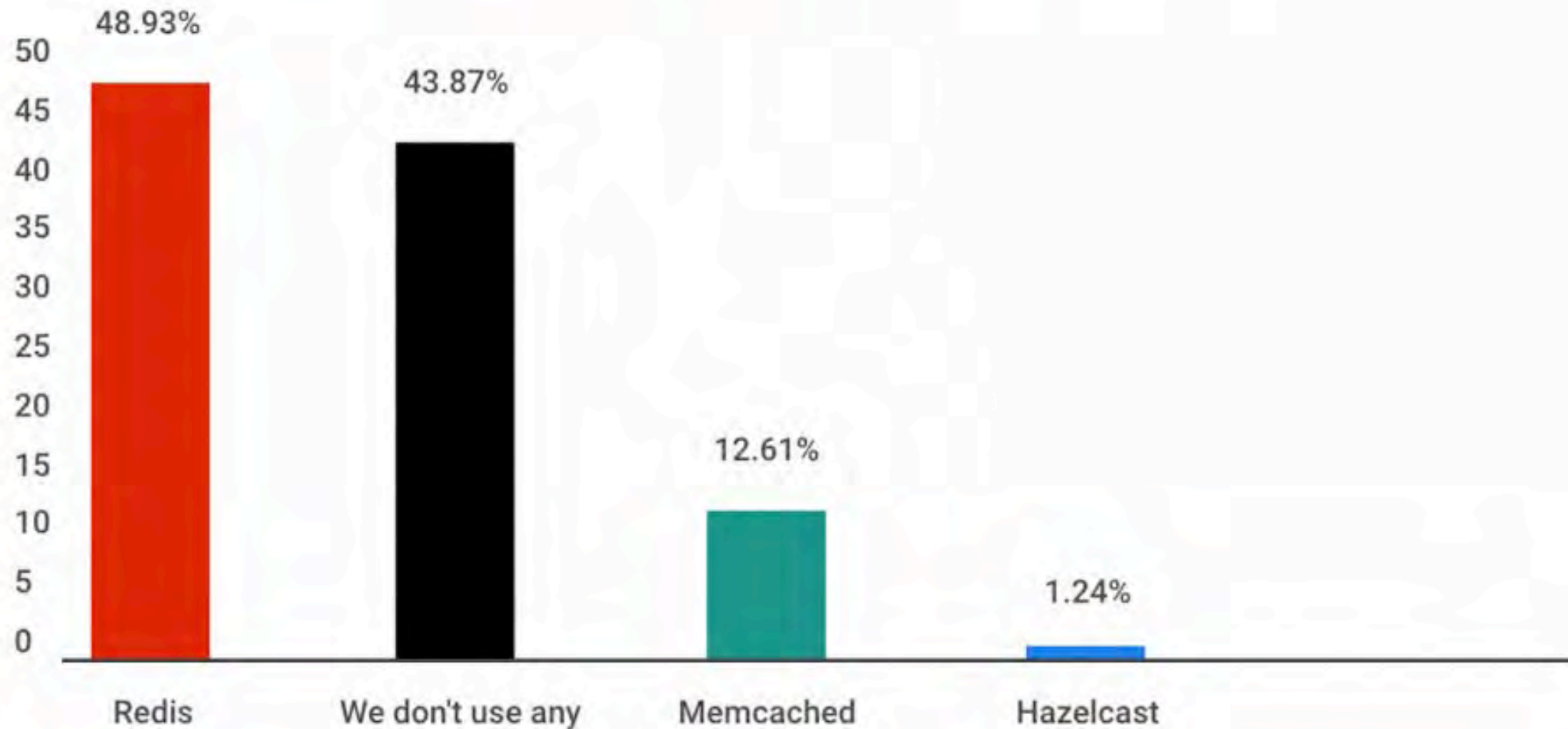
REDIS é utilizado para armazenamento do grafo social de usuários.



70MM de usuários
53MM de acessos únicos / mês
Média de 67 followers por usuário

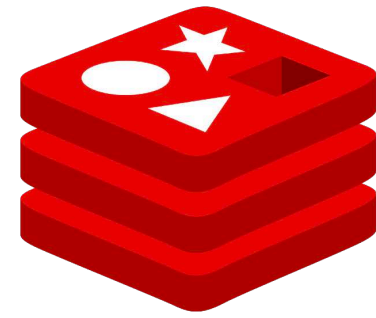
What do you use for caching?

1126 respondents - multiple choice answers

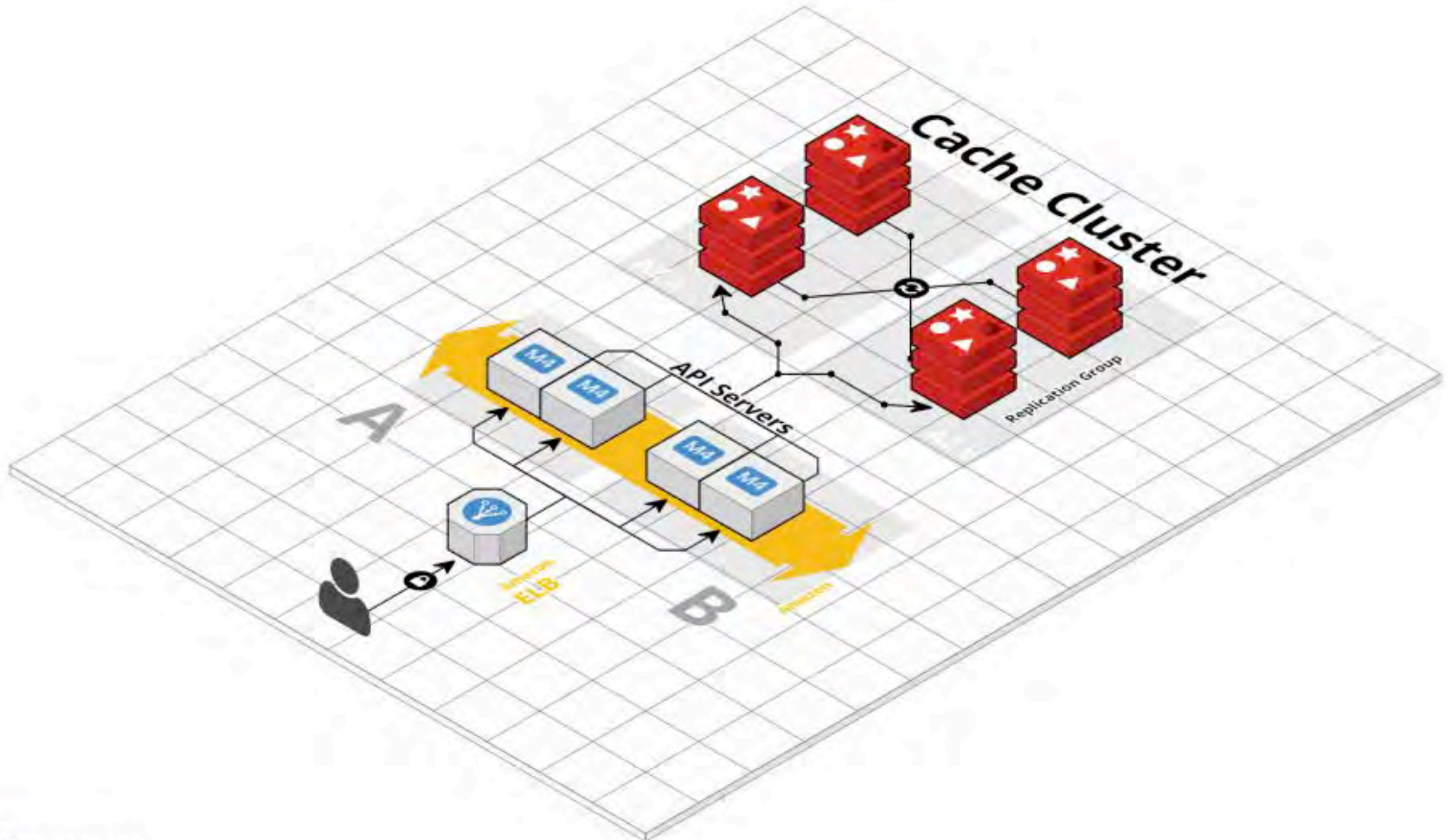


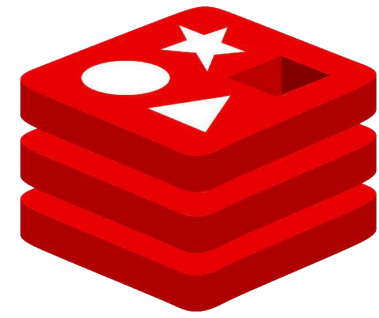


Todas estas linguagem possuem libs e drivers para REDIS

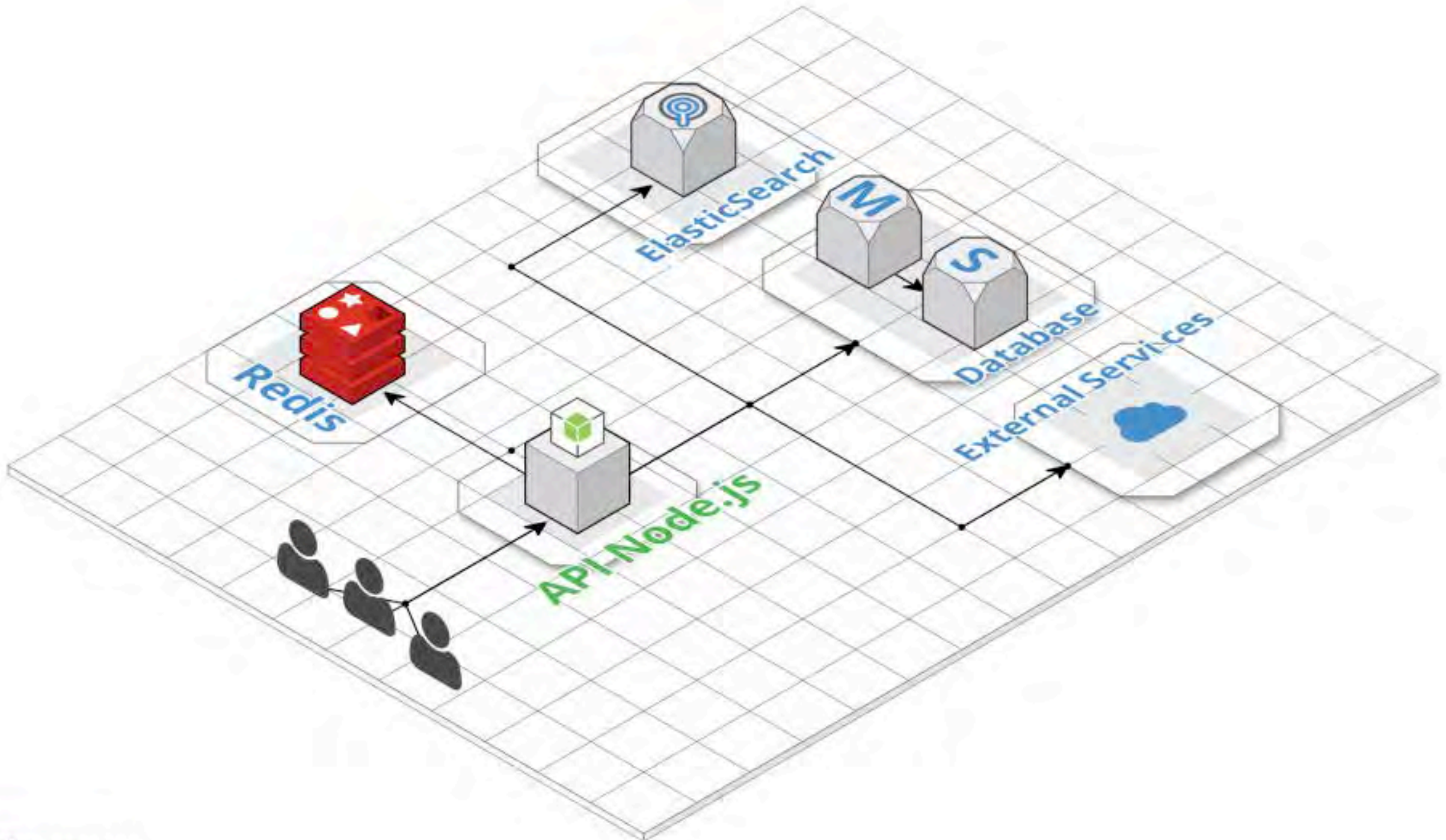


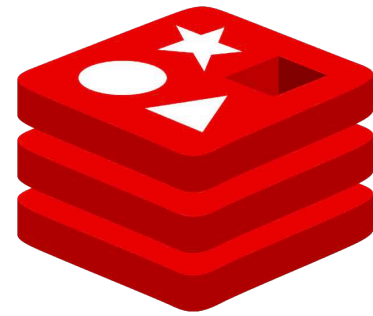
redis





redis





redis

How to use this image

start a redis instance

```
$ docker run --name some-redis -d redis
```

start with persistent storage

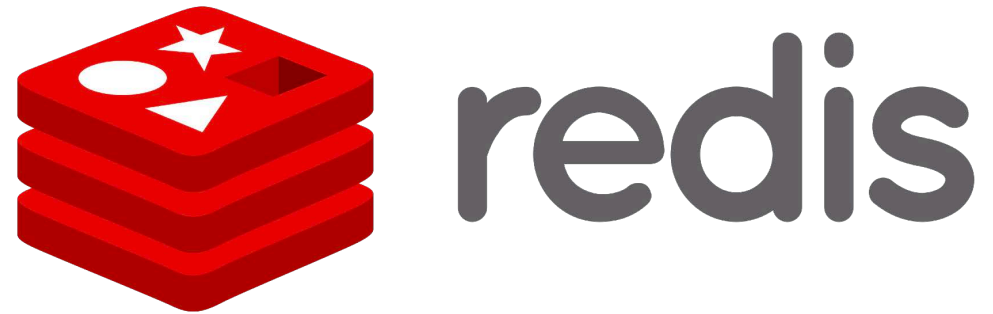
```
$ docker run --name some-redis -d redis redis-server --appendonly yes
```

If persistence is enabled, data is stored in the `VOLUME /data`, which can be used with `--volumes-from some-volume-container` or `-v /docker/host/dir:/data` (see [docs.docker.com volumes](https://docs.docker.com/volumes/)).

For more about Redis Persistence, see <http://redis.io/topics/persistence>.

connecting via redis-cli

```
$ docker run -it --network some-network --rm redis redis-cli -h some-redis
```

Instalando e executando o Redis

```
docker run --name redis-nosql p 6379:6379 -d  
redis
```

Checando os logs

```
docker logs redis-nosql
```



Executando CLI do Redis

```
docker exec -it redis-nosql redis-cli
```

Testando comunicação

```
$ ECHO TESTE
```

Se tudo estiver OK responderá com a mensagem “TESTE”



Inserindo dados em uma chave

```
$ SET <KEY> VALUE
```

Recuperando valor de uma chave

```
$ GET <KEY>
```



Exemplo:

```
$ SET ultimo_sorteio "2, 15, 18, 30, 35, 42"
```

```
$ SET "penultimo_sorteio" "14, 17, 18, 25, 32, 42"
```

```
$ SET "antepenultimo_sorteio" "2, 17, 18, 26, 35, 43"
```



Listando todas chaves armazenadas

```
$ KEYS *
```

`h?llo` matches `hello`, `hallo` and `hxlllo`

`h*llo` matches `hlllo` and `heeeello`

`h[ae]llo` matches `hello` and `hallo`, but not `hillo`

`h[^e]llo` matches `hallo`, `hblllo`, ... but not `hello`

`h[a-b]llo` matches `hallo` and `hblllo`

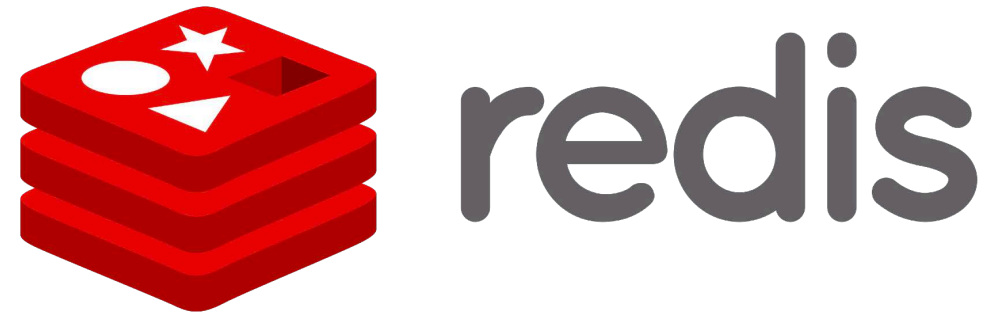


Para uma melhor organização e facilitar localizar os dados, é importante criar um padrão para as chaves

```
$ SET resultado:11-08-2019:megasena "2, 15, 18, 25,  
28, 32"
```

```
$ SET resultado:11-07-2019:megasena "4, 16, 19, 23,  
28, 43"
```

```
$ SET resultado:11-06-2019:megasena "7, 13, 16, 21,  
33, 44"
```

Buscando resultados do mes 06 e 07

```
$ KEYS "resultado:??-0[6-7]-?????:megasena"
```



É possível configurar o REDIS para automaticamente remover uma chave após determinado tempo.

```
$ EXPIRE my-key 10
```

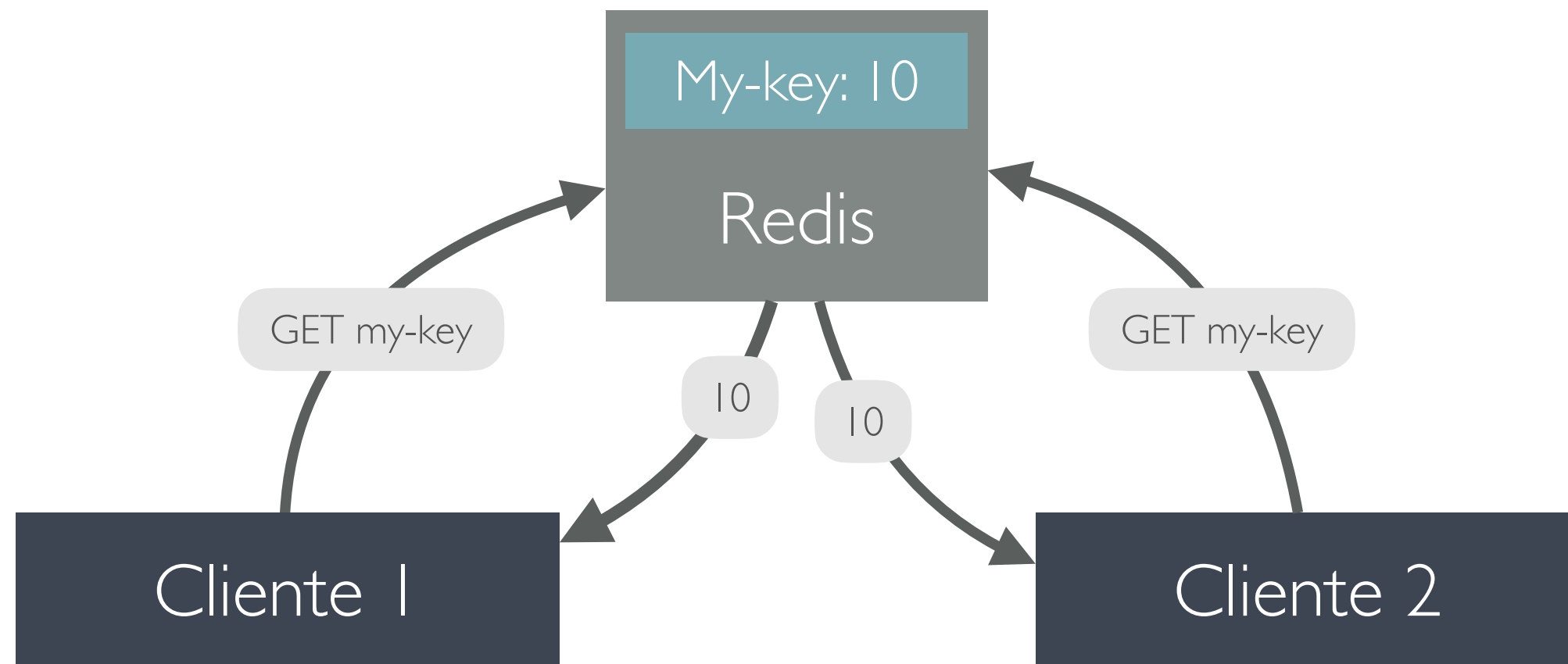
Ajusta o tempo de expiração da chave my-key para 10 segundos

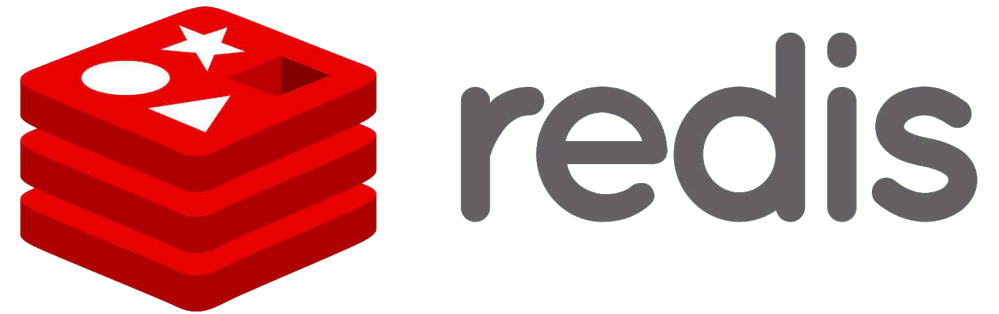
```
$ TTL my-key
```

Verifica o tempo restante para expiração para chave my-key (em segundos)

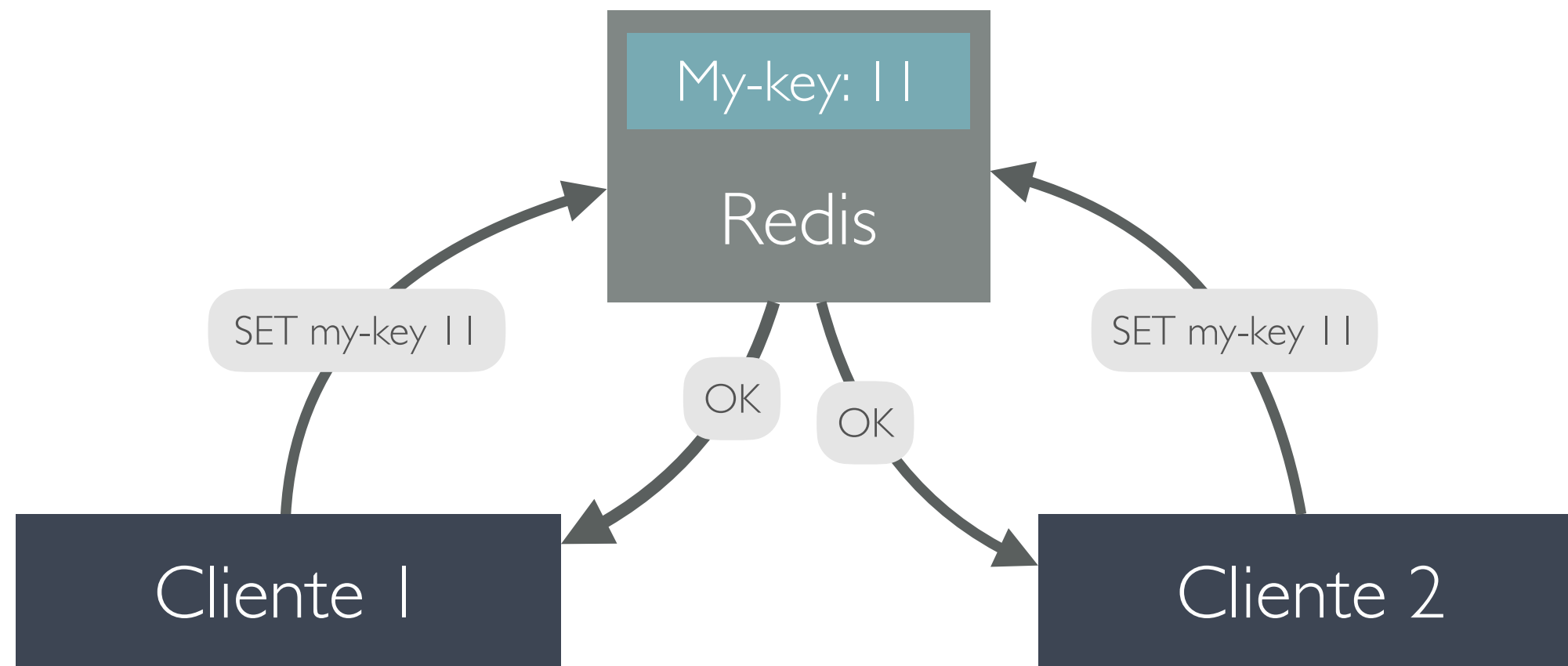


E se quisermos incrementar o valor de uma chave?





E se quisermos incrementar o valor de uma chave?





Operação de incremento/decremento

Para incrementar um valor existente utilize o operador específico para evitar inconsistências.

```
$ INCR my-key
```

Incrementa em uma unidade o valor da chave my-key

```
$ DECR my-key
```

Decrementa em uma unidade o valor da chave my-key



Operação de incremento/decremento

Para incrementar um valor existente utilize o operador específico para evitar inconsistências.

```
$ INCRBY my-key 10
```

Incrementa em 10 o valor da chave my-key

```
$ DECRBY my-key 10
```

Decrementa em 10 o valor da chave my-key



Operação de incremento/decremento

Para incrementar um valor existente utilize o operador específico para evitar inconsistências.

```
$ INCRBYFLOAT my-key 0.1
```

Incrementa em 0.1 o valor da chave my-key

```
$ INCRBYFLOAT my-key -0.1
```

Não existe o comando de decrementar float. Em vez disso, utilize valores negativos.

Integração com Javascript



Acessando via Javascript Node.js

```
$ npm i redis --save
```

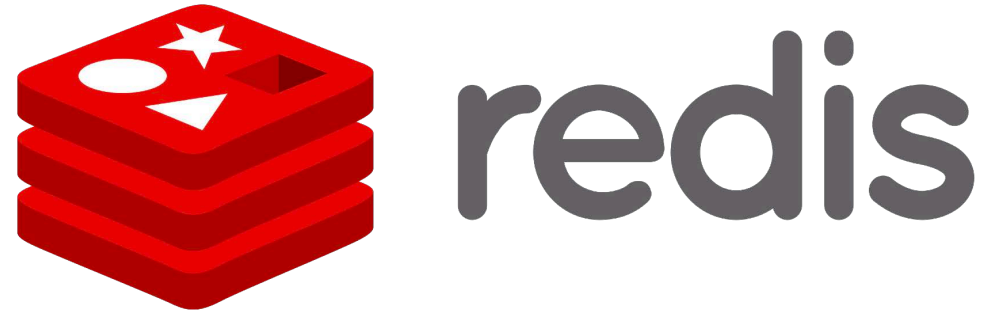
```
{
  "name": "nodejs-image-demo",
  "version": "1.0.0",
  "description": "nodejs image demo",
  "author": "Gustavo Leitão <gustavo.leitao@imd.ufrn.br>",
  "main": "app.js",
  "dependencies": {
    "express": "^4.16.4",
    "redis": "^2.8.0"
  },
  "scripts": {
    "start": "node app.js"
  }
}
```



Acessando via Javascript Node.js

```
$ npm i redis --save
```

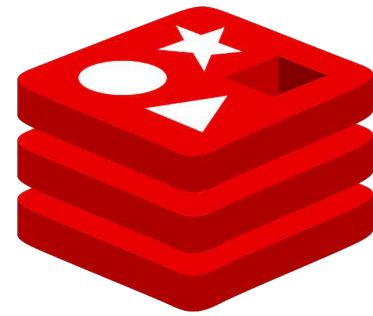
```
{
  "name": "nodejs-image-demo",
  "version": "1.0.0",
  "description": "nodejs image demo",
  "author": "Gustavo Leitão <gustavo.leitao@imd.ufrrn.br>",
  "main": "app.js",
  "dependencies": {
    "express": "^4.16.4",
    "redis": "^2.8.0"
  },
  "scripts": {
    "start": "node index.js"
  }
}
```



```
const express = require('express')
const redis = require('redis')
const cache = redis.createClient()
const app = express()
const port = 3000

cache.on('connect', () => {
  console.log('Redis is ready');
});

cache.on('error', (e) => {
  console.log('Redis error', e);
});
```



redis

```
app.get('/ler', (req, res) => {
  cache.get('sorteio:last:megasena', (err, reply) => {
    if (err){
      res.send('ERROR')
    }else{
      res.send(reply)
    }
  })
})

app.get('/escrever', (req, res) => {
  cache.set('sorteio:last:megasena', '1,2,3,4,5,6', function (err, reply){
    if (err){
      res.send('ERROR')
    }else{
      res.send('OK')
    }
  })
})
```



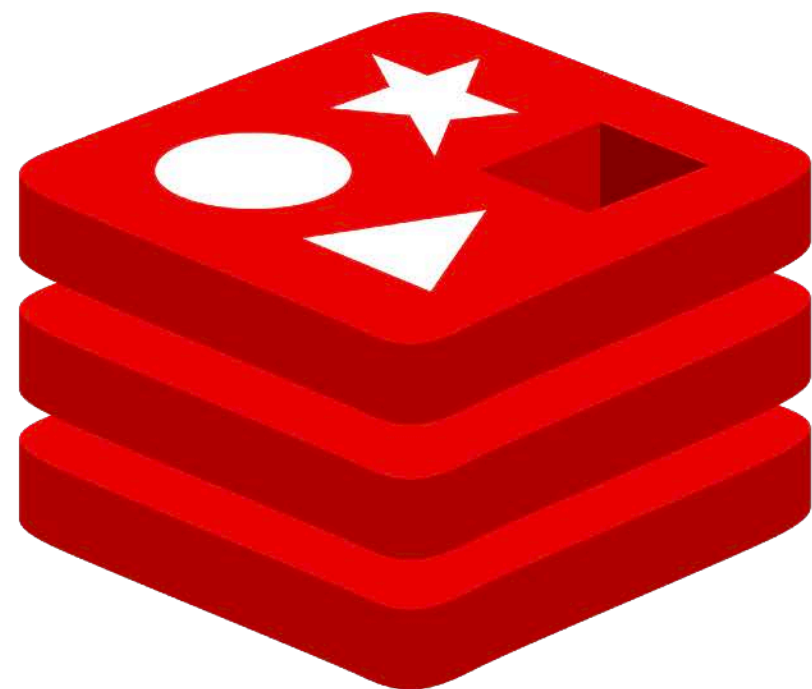

Exercício:

Crie um programa que utilize o Redis como Cache para armazenar o resultado do fatorial de um número. Na primeira vez, caso o valor deve ser calculado. Nas seguintes o cache deve ser utilizado. Seguindo os seguintes requisitos:

- O parâmetro deve ser passado (valor) na URL.
- O Cache deve expirar após 5 minutos

Link importante

https://github.com/NodeRedis/node_redis



redis

Obrigado!

Prof. Gustavo Leitão