lab

lab title

Using Amazon ElastiCache Redis V1.02

Course title

AWS Certified Solutions Architect Associate

Table of Contents

Contents

Please note that AWS services change on a weekly basis and it is extremely important you check the version number on this document to ensure you have the lastest version with any updates or corrections.

About the Lab

These lab notes are to support the instructional videos on Using AWS ElastiCache Redis in the BackSpace AWS Certified Solutions Architect course.

In this lab we will:

- Create an ElastiCache Redis cluster using the console.
- Connect to an ElastiCache Redis cluster from EC2 using the Redis CLI.
- Read and Write to an ElastiCache Redis cluster.

Please refer to the AWS JavaScript SDK documentation at:

http://docs.aws.amazon.com/AWSJavaScriptSDK/latest/AWS/ElastiCache.html

Please refer to the Redis command documentation at:

http://redis.io/commands

Please note that AWS services change on a weekly basis and it is extremely important you check the version number on this document to ensure you have the lastest version with any updates or corrections.

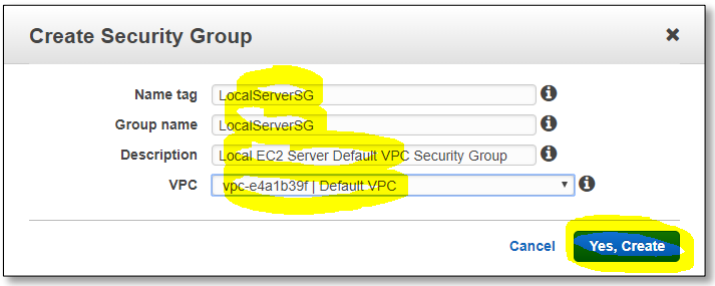
Launch an ElastiCache Redis Cluster

In this section we will create an ElastiCache Redis cluster using the console.

Create Security Groups

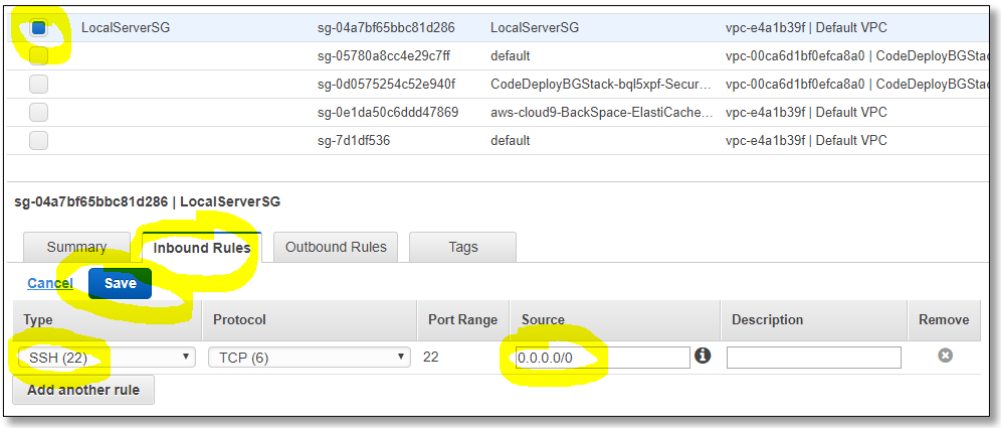
Go to the VPC console

Create a new security group in the default VPC and call it *LocalServerSG*

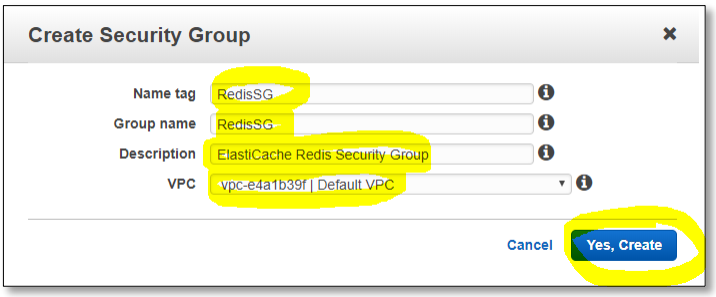


Create an inbound rule for SSH access from source *0.0.0.0/0* (if you have a static IP address you can enter that instead for more security)

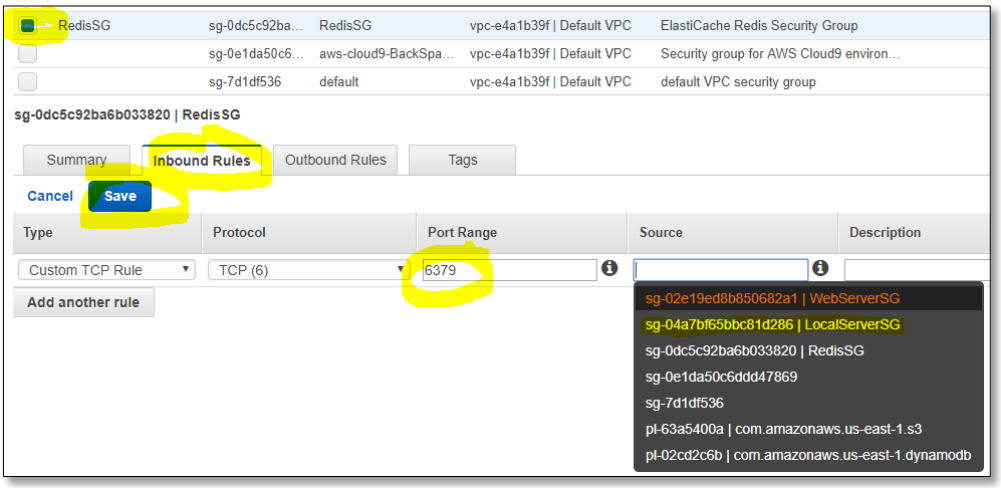
Click *Save*



Create a new security group in the default VPC and call it *RedisSG*



Create a custom TCP rule for the ElastiCache Redis port 6379 and source *LocalServerSG* security group (type *sg* to get a drop-down list of security groups)

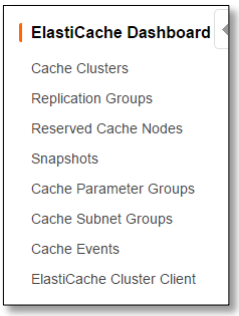


Click *Save*

Launch ElastiCache Cluster

Go to the ElastiCache console.

Select *Subnet Groups*



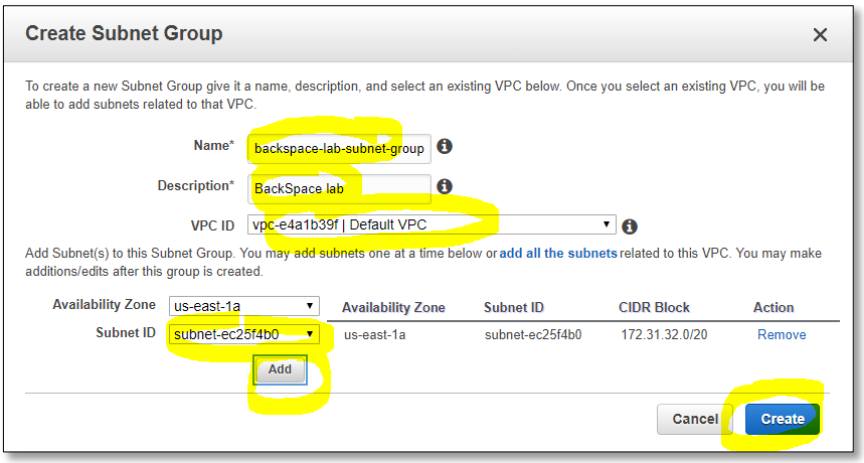
Click Create Subnet Group

Give it a name and description

Select the default VPC and an AZ and subnet.

Click Add

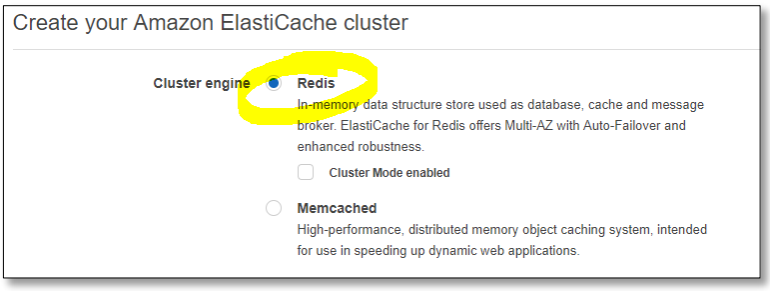
Click Create



Click ElastiCache Dashbboard

Click “Get Started Now”

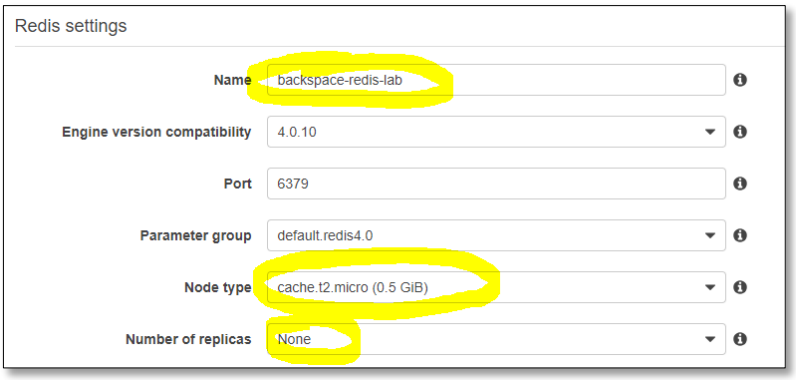
Select Redis



Call the cluster *backspace-redis-lab*

Select the t2 micro *Node type*

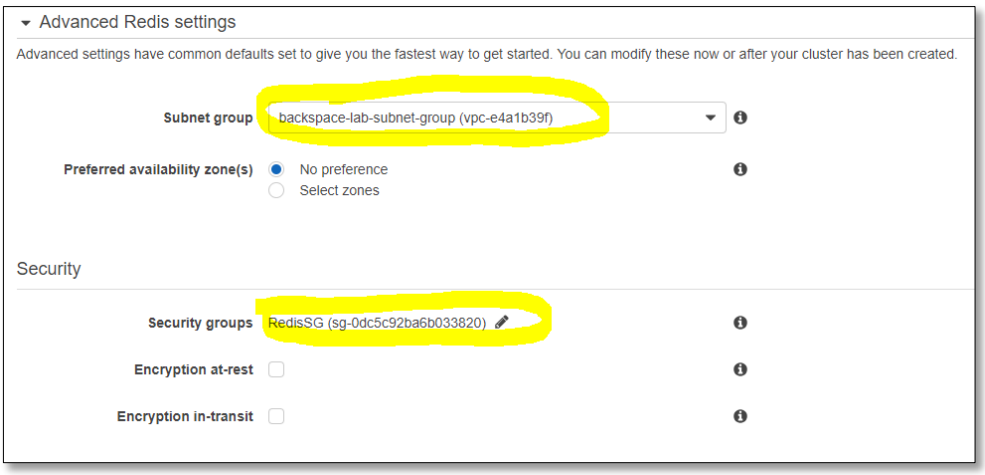
Set *Number of replicas* to zero



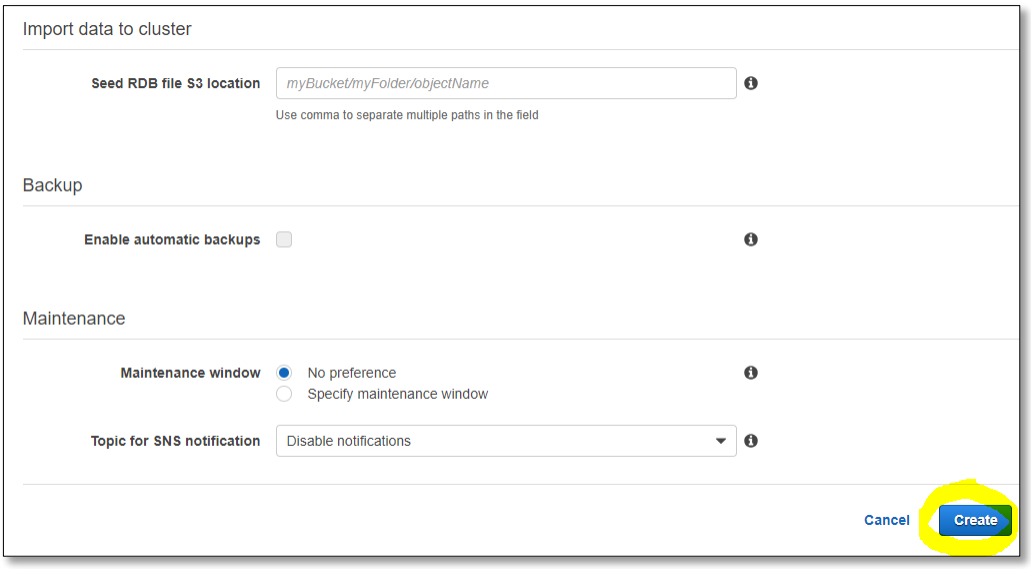
Click Next

Select your Subnet Group created previously

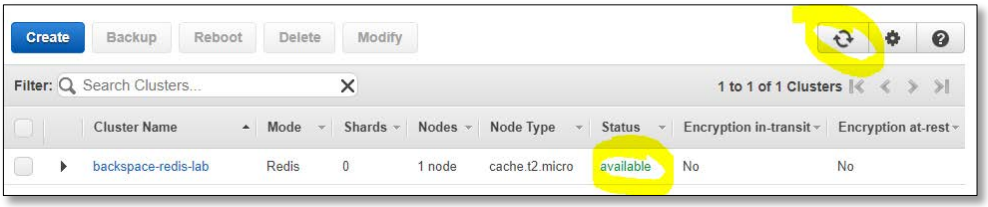
Select your RedisSG Security Group created previously



Click *create*



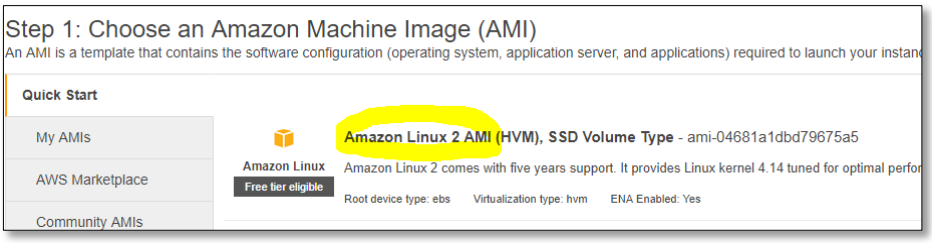
Refresh view and wait until status becomes *available*



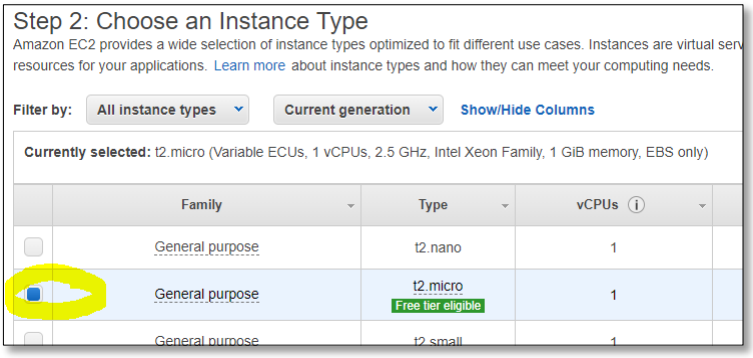
Creating an EC2 Instance

In this section we will create an EC2 instance to connect to our ElastiCache cluster.

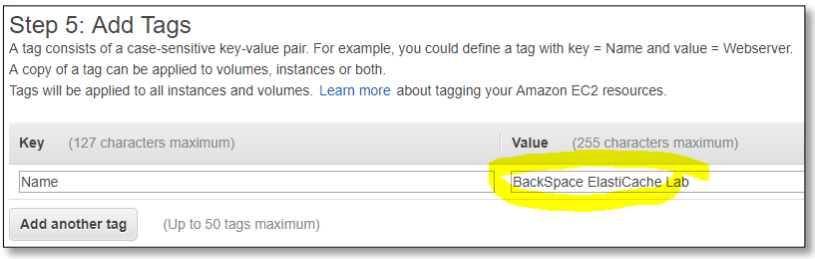
From the EC2 console create an Amazon Linux 2 instance



Select *t2.micro*

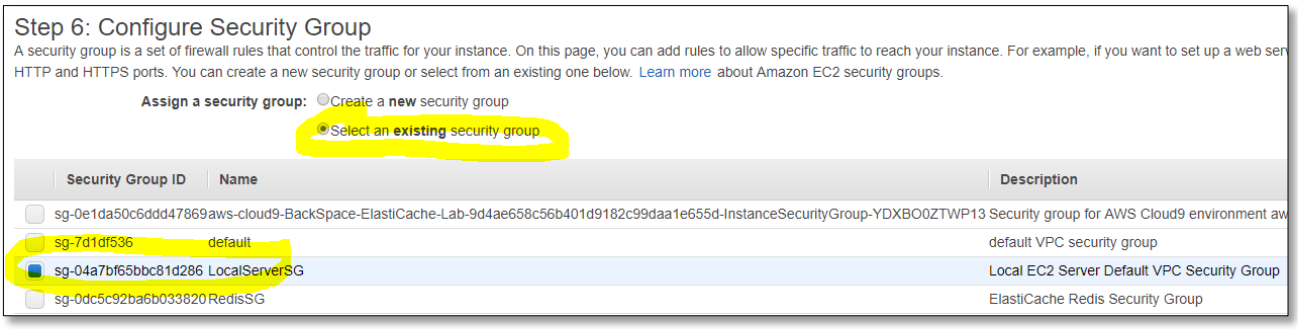


Add a Name tag



Select the *LocalServerSG* Security Group

Click *Review and Launch*



Click *Launch*

Select a key pair

Click *Launch instance*

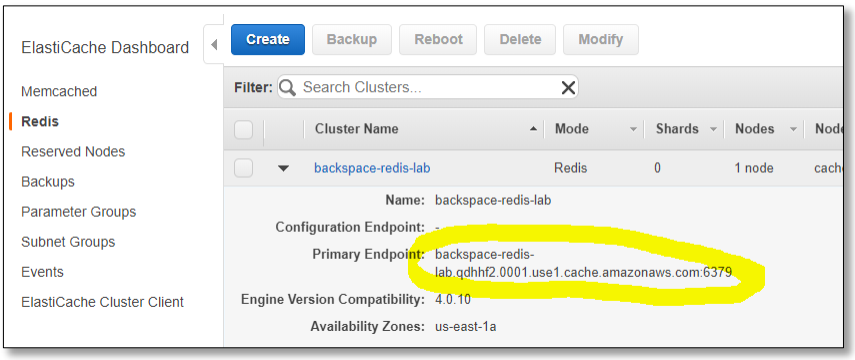
Using ElastiCache Redis with the Redis CLI

In this section we will read and write to an ElastiCache Redis cluster using the Redis Command Line Interface (CLI).

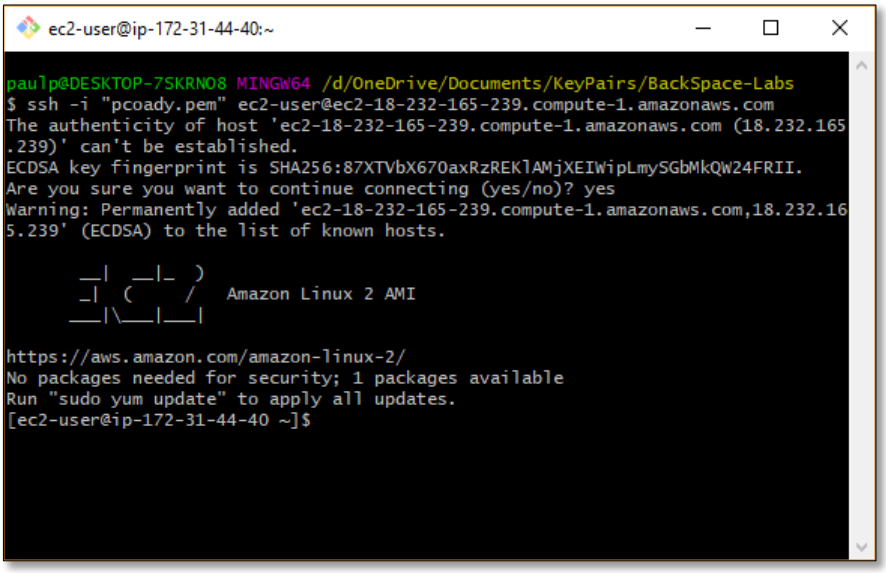
From the console go to Cache Clusters

Click on the Cluster Node in your Cache Cluster

Copy the endpoint and the port, we will need this to connect to the node.



Connect into your Amazon Linux 2 EC2 instance using Bash (Windows) or Terminal (Mac)

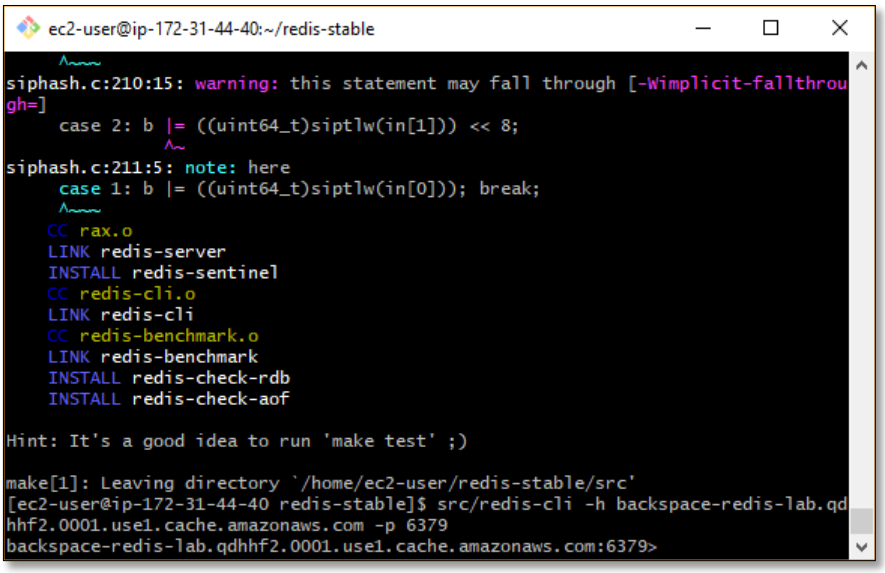


Input the following commands to install GCC and the Redis-CLI utility:

```
sudo yum install gcc
wget http://download.redis.io/redis-stable.tar.gz
tar xvzf redis-stable.tar.gz
cd redis-stable
make
```

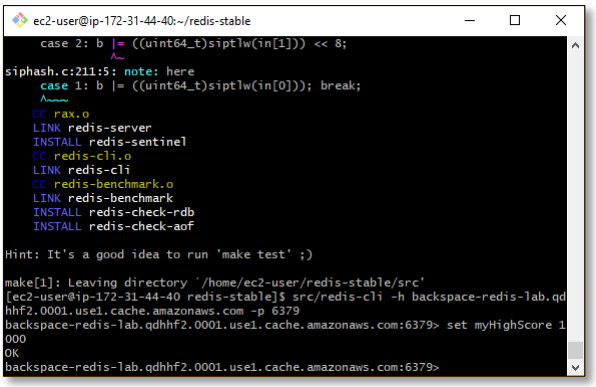
Now connect to your Redis cluster your endpoint domain (without the :6379)

```
src/redis-cli -h YOUR_ENDPOINT_GOES_HERE -p 6379
```



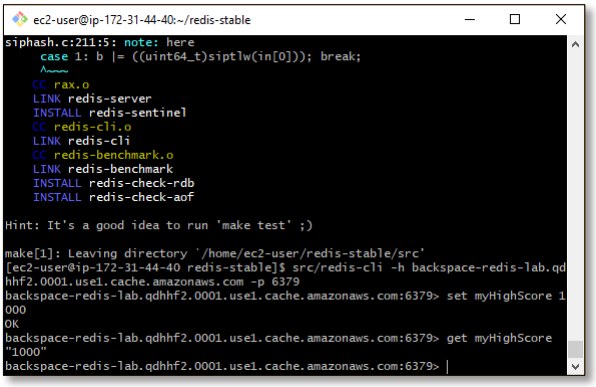
Now run a command to set a key myHighScore to 1000:

```
set myHighScore 1000
```



Now read the key:

```
get myHighScore
```



Now set an expiry time of 30s for the key:

```
expire myHighScore 30
```

It will return:

- 1 if the timeout was set.
- 0 if key does not exist or the timeout could not be set.

```
ec2-user@ip-172-31-44-40:~/redis-stable
    INSTALL redis-check-aof

Hint: It's a good idea to run 'make test' ;)

make[1]: Leaving directory `/home/ec2-user/redis-stable/src'
[ec2-user@ip-172-31-44-40 redis-stable]$ src/redis-cli -h backspace-redis-lab.qd
hhf2.0001.use1.cache.amazonaws.com -p 6379
backspace-redis-lab.qdhhf2.0001.use1.cache.amazonaws.com:6379> set myHighScore 1
000
OK
backspace-redis-lab.qdhhf2.0001.use1.cache.amazonaws.com:6379> get myHighScore
"1000"
backspace-redis-lab.qdhhf2.0001.use1.cache.amazonaws.com:6379> expire myHighScor
e 30
(integer) 1
backspace-redis-lab.qdhhf2.0001.use1.cache.amazonaws.com:6379>
```

Now wait 30 s and read the key:

```
get myHighScore
```

If 30s has past it will return (nil)

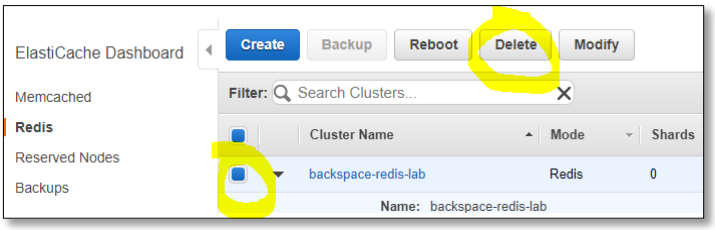
```
ec2-user@ip-172-31-44-40:~/redis-stable
Hint: It's a good idea to run 'make test' ;)

make[1]: Leaving directory `/home/ec2-user/redis-stable/src'
[ec2-user@ip-172-31-44-40 redis-stable]$ src/redis-cli -h backspace-redis-lab.qd
hhf2.0001.use1.cache.amazonaws.com -p 6379
backspace-redis-lab.qdhhf2.0001.use1.cache.amazonaws.com:6379> set myHighScore 1
000
OK
backspace-redis-lab.qdhhf2.0001.use1.cache.amazonaws.com:6379> get myHighScore
"1000"
backspace-redis-lab.qdhhf2.0001.use1.cache.amazonaws.com:6379> expire myHighScor
e 30
(integer) 1
backspace-redis-lab.qdhhf2.0001.use1.cache.amazonaws.com:6379> get myHighScore
(nil)
backspace-redis-lab.qdhhf2.0001.use1.cache.amazonaws.com:6379>
```

Type *Exit* to disconnect

```
ec2-user@ip-172-31-44-40:~/redis-stable
make[1]: Leaving directory `/home/ec2-user/redis-stable/src'
[ec2-user@ip-172-31-44-40 redis-stable]$ src/redis-cli -h backspace-redis-lab.qd
hhf2.0001.use1.cache.amazonaws.com -p 6379
backspace-redis-lab.qdhhf2.0001.use1.cache.amazonaws.com:6379> set myHighScore 1
000
OK
backspace-redis-lab.qdhhf2.0001.use1.cache.amazonaws.com:6379> get myHighScore
"1000"
backspace-redis-lab.qdhhf2.0001.use1.cache.amazonaws.com:6379> expire myHighScor
e 30
(integer) 1
backspace-redis-lab.qdhhf2.0001.use1.cache.amazonaws.com:6379> get myHighScore
(nil)
backspace-redis-lab.qdhhf2.0001.use1.cache.amazonaws.com:6379> exit
[ec2-user@ip-172-31-44-40 redis-stable]$
```

Cleaning Up

Delete your cluster in the ElastiCache console.



Terminate your EC2 instance in the EC2 console