amazon web services | Certified

Solutions Architect - Associate

lab

lab title

Using Amazon ElastiCache Redis V1.01

Course title

AWS Certified Solutions Architect Associate

BackSpace

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

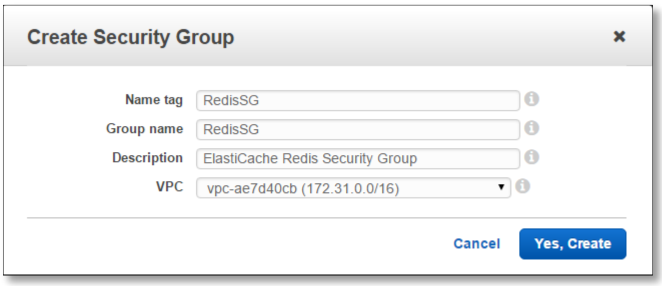
Please note that AWS services change on a weekly basis and it is extremely important you check the version number on this document to ensure you have the lastest version with any updates or corrections.

Launch an ElastiCache Redis Cluster

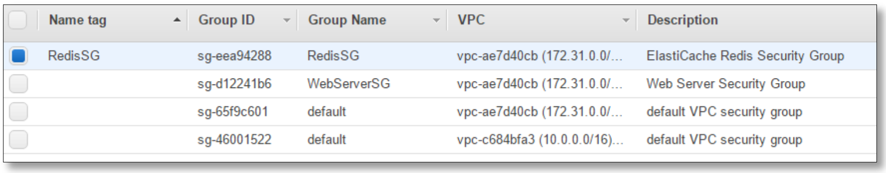
In this section we will create an ElastiCache Redis cluster using the console.

Go to the VPC console

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

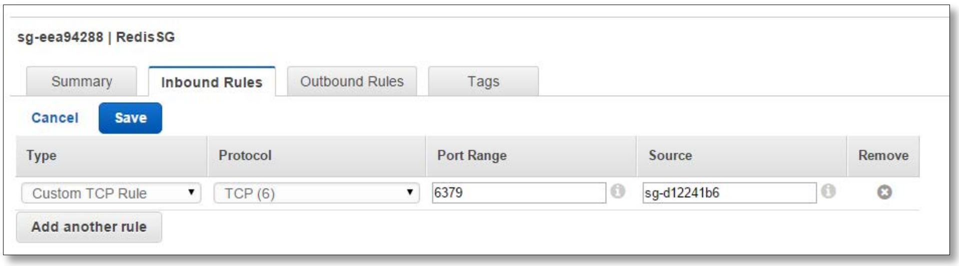


Name tag	RedisSG
Group name	RedisSG
Description	ElastiCache Redis Security Group
VPC	vpc-ae7d40cb (172.31.0.0/16)



	Name tag	Group ID	Group Name	VPC	Description
■	RedisSG	sg-eea94288	RedisSG	vpc-ae7d40cb (172.31.0.0/...	ElastiCache Redis Security Group
□		sg-d12241b6	WebServerSG	vpc-ae7d40cb (172.31.0.0/...	Web Server Security Group
□		sg-65f9c601	default	vpc-ae7d40cb (172.31.0.0/...	default VPC security group
□		sg-46001522	default	vpc-c684bfa3 (10.0.0.0/16)...	default VPC security group

Create a custom TCP rule for the ElastiCache Redis port 6379 and the WebServerSG security group.

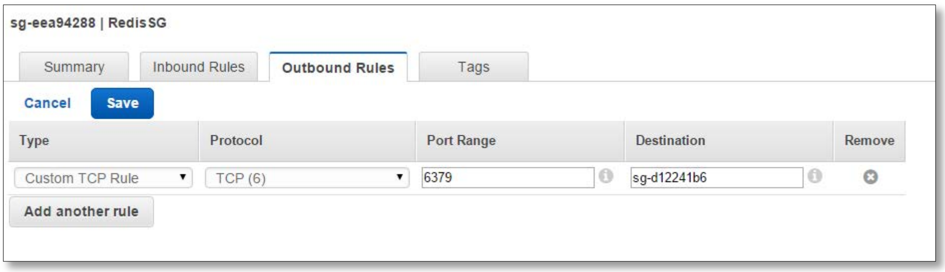


sg-eea94288 | RedisSG

Type	Protocol	Port Range	Source	Remove
Custom TCP Rule	TCP (6)	6379	sg-d12241b6	⊗

Click Save

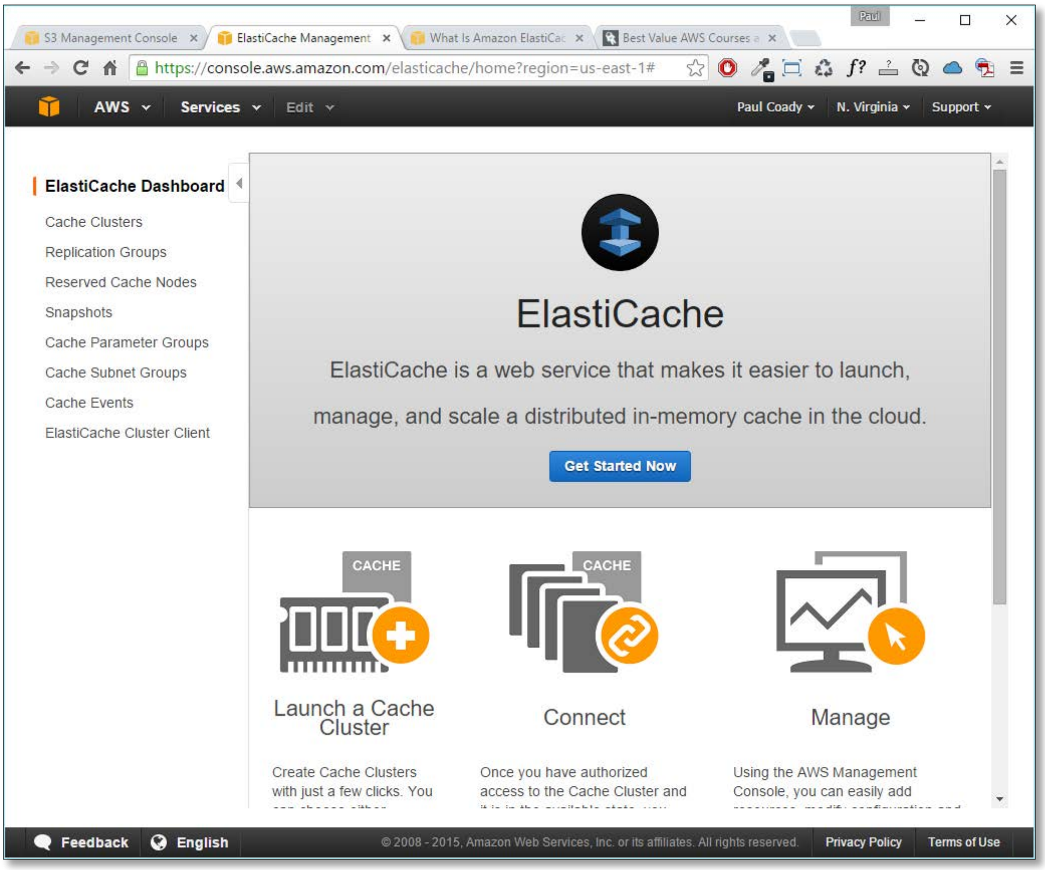
Create a custom TCP outbound rule for the ElastiCache Redis port 6379 and the WebServerSG security group.



sg-eea94288 | Redis SG

Type	Protocol	Port Range	Destination	Remove
Custom TCP Rule	TCP (6)	6379	sg-d12241b6	⊗

Go to the ElastiCache console.



https://console.aws.amazon.com/elasticache/home?region=us-east-1#

ElastiCache is a web service that makes it easier to launch, manage, and scale a distributed in-memory cache in the cloud.

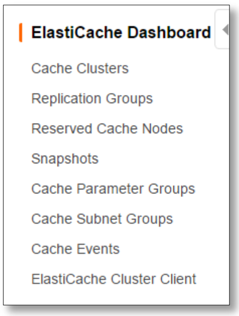
Create Cache Clusters with just a few clicks. You

Once you have authorized access to the Cache Cluster and

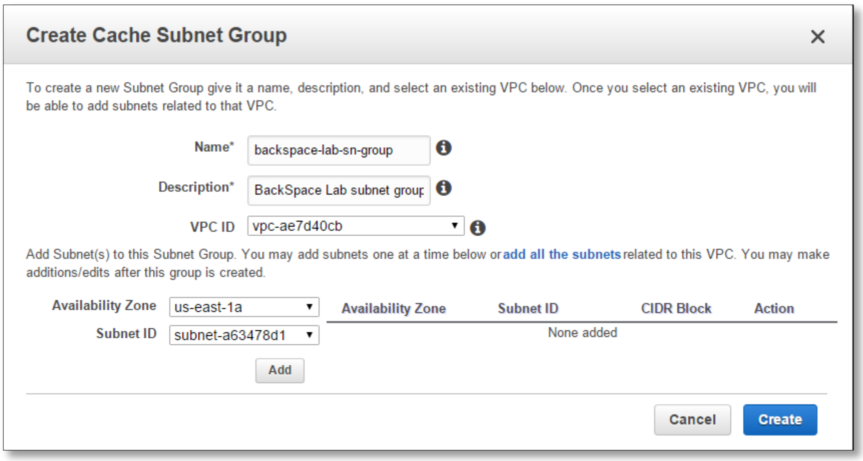
Using the AWS Management Console, you can easily add

© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Click on Cache Subnet Groups



Click Create Cache Subnet Group
Give it a name
Select the default VPC and an AZ and subnet.
Click Add
Click Create



Create Cache Subnet Group

To create a new Subnet Group give it a name, description, and select an existing VPC below. Once you select an existing VPC, you will be able to add subnets related to that VPC.

Name*: backspace-lab-sn-group
Description*: BackSpace Lab subnet group
VPC ID: vpc-ae7d40cb

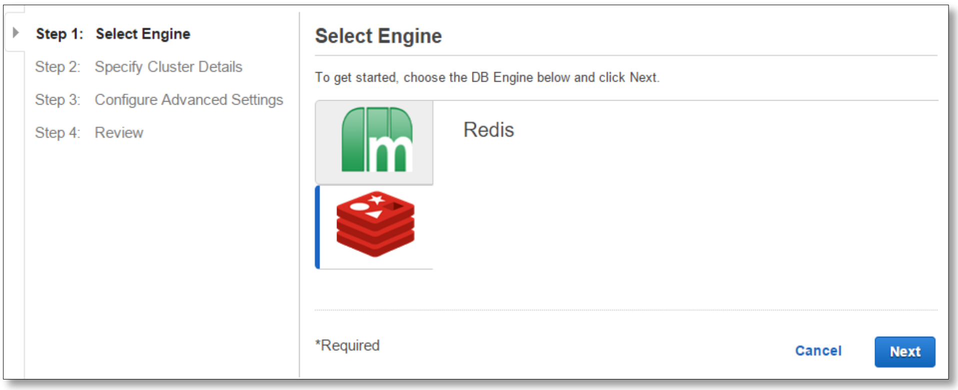
Add Subnet(s) to this Subnet Group. You may add subnets one at a time below or add all the subnets related to this VPC. You may make additions/edits after this group is created.

Availability Zone: us-east-1a
Subnet ID: subnet-a63478d1

Availability Zone	Subnet ID	CIDR Block	Action
None added			

Click ElastiCache Dashbboard
Click “Get Started Now”

Select Redis



Step 1: Select Engine
Step 2: Specify Cluster Details
Step 3: Configure Advanced Settings
Step 4: Review

Select Engine

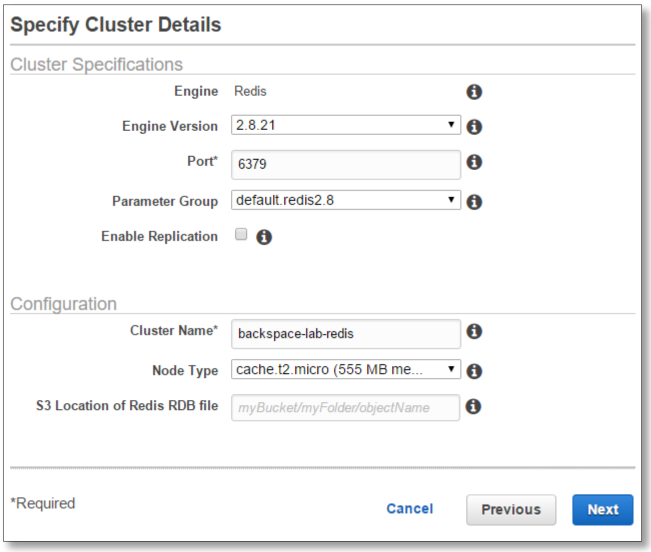
To get started, choose the DB Engine below and click Next.

Redis

*Required

Click Next

Call the cluster backspace-lab-redis
Uncheck Enable replication for the lab
Select the t2 micro node type



Specify Cluster Details

Cluster Specifications

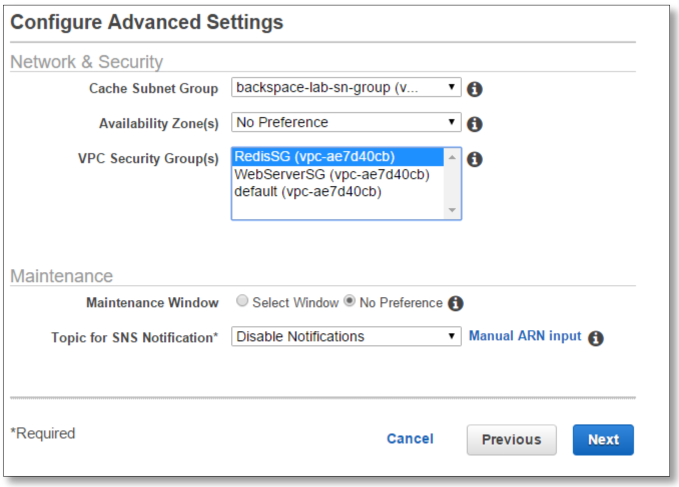
Engine	Redis
Engine Version	2.8.21
Port*	6379
Parameter Group	default.redis2.8
Enable Replication	□

Configuration

Cluster Name*	backspace-lab-redis
Node Type	cache.t2.micro (555 MB me...
S3 Location of Redis RDB file	myBucket/myFolder/objectName

*Required

Click Next
Select your Subnet Group created previously
Select default VPC
Select your Security Group created previously



Configure Advanced Settings

Network & Security

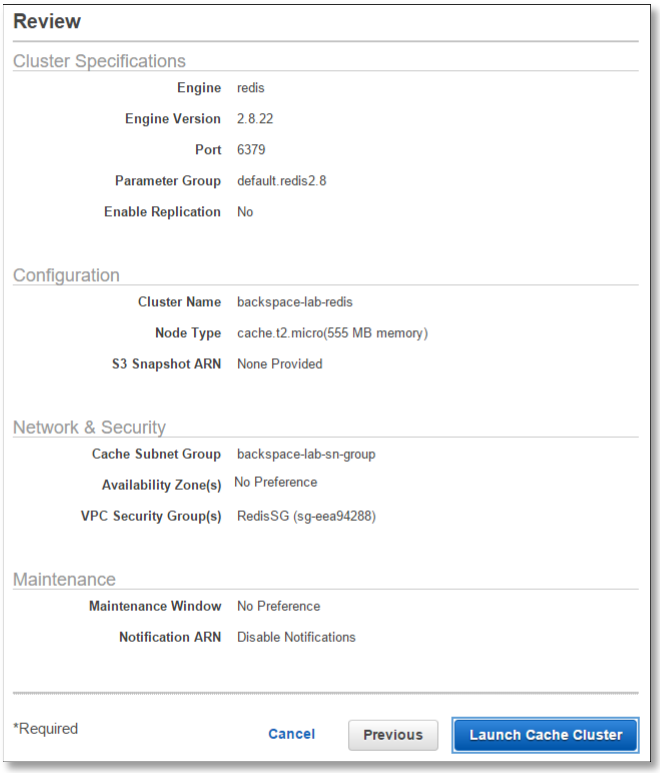
Cache Subnet Group	backspace-lab-sn-group (v...
Availability Zone(s)	No Preference
VPC Security Group(s)	RedisSG (vpc-ae7d40cb) WebServerSG (vpc-ae7d40cb) default (vpc-ae7d40cb)

Maintenance

Maintenance Window	○ Select Window ● No Preference
Topic for SNS Notification*	Disable Notifications

*Required

Click Next



Review

Cluster Specifications

Engine	redis
Engine Version	2.8.22
Port	6379
Parameter Group	default.redis2.8
Enable Replication	No

Configuration

Cluster Name	backspace-lab-redis
Node Type	cache.t2.micro(555 MB memory)
S3 Snapshot ARN	None Provided

Network & Security

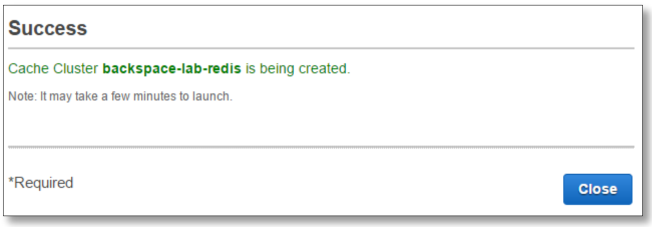
Cache Subnet Group	backspace-lab-sn-group
Availability Zone(s)	No Preference
VPC Security Group(s)	RedisSG (sg-eea94288)

Maintenance

Maintenance Window	No Preference
Notification ARN	Disable Notifications

*Required

Launch Cache Cluster



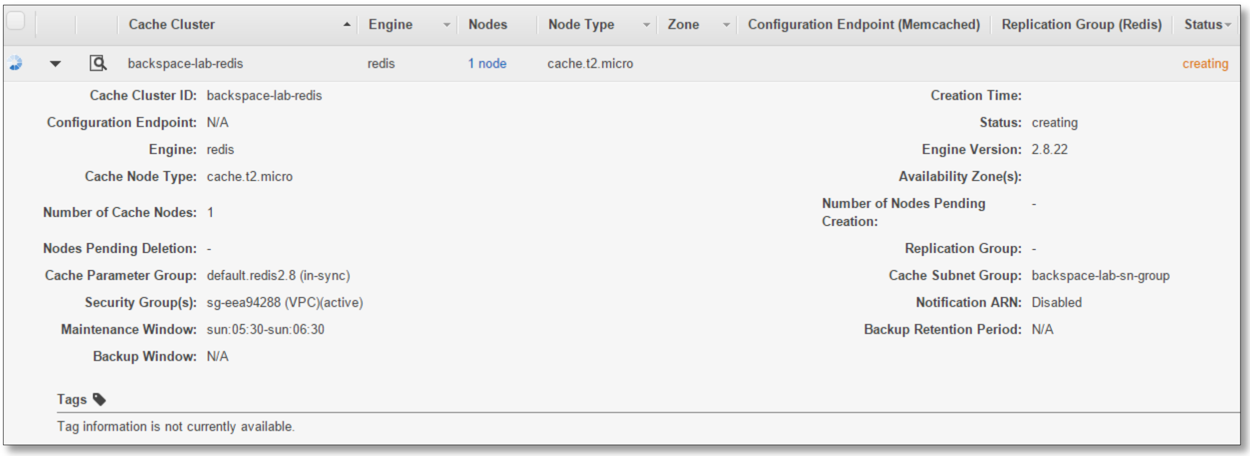
Success

Cache Cluster **backspace-lab-redis** is being created.

Note: It may take a few minutes to launch.

*Required

Click Close



Cache Cluster	Engine	Nodes	Node Type	Zone	Configuration Endpoint (Memcached)	Replication Group (Redis)	Status
backspace-lab-redis	redis	1 node	cache.t2.micro				creating

Cache Cluster ID:	backspace-lab-redis	Creation Time:	
Configuration Endpoint:	N/A	Status:	creating
Engine:	redis	Engine Version:	2.8.22
Cache Node Type:	cache.t2.micro	Availability Zone(s):	
Number of Cache Nodes:	1	Number of Nodes Pending Creation:	-
Nodes Pending Deletion:	-	Replication Group:	-
Cache Parameter Group:	default.redis2.8 (in-sync)	Cache Subnet Group:	backspace-lab-sn-group
Security Group(s):	sg-eea94288 (VPC)(active)	Notification ARN:	Disabled
Maintenance Window:	sun:05:30-sun:06:30	Backup Retention Period:	N/A
Backup Window:	N/A		

Tags

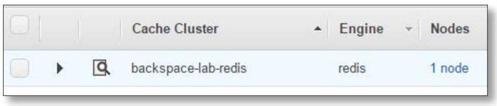
Tag information is not currently available.

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).

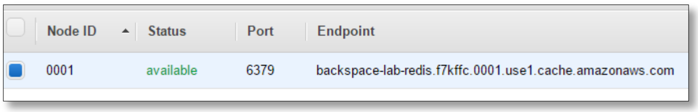
From the console go to Cache Clusters

Click on the Cluster Node in your Cache Cluster



□			Cache Cluster	Engine	Nodes
□	▸	🔍	backspace-lab-redis	redis	1 node

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

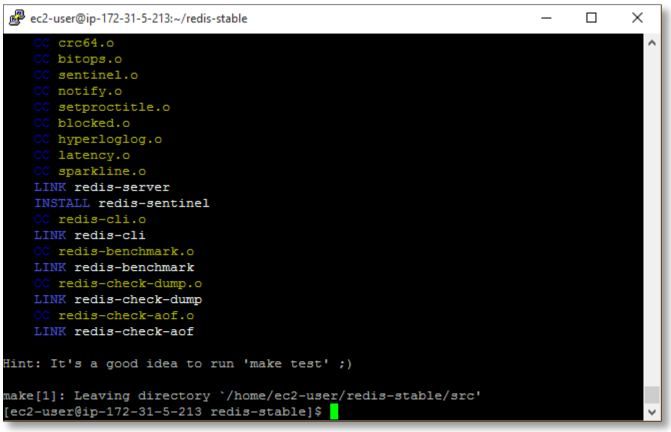


□	Node ID	Status	Port	Endpoint
☑	0001	available	6379	backspace-lab-redis.f7kffc.0001.use1.cache.amazonaws.com

Connect into your NodeJS Amazon Linux EC2 instance using Putty

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
```



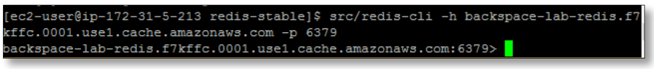
```
    CC crc64.o
    CC bitops.o
    CC sentinel.o
    CC notify.o
    CC setproctitle.o
    CC blocked.o
    CC hyperloglog.o
    CC latency.o
    CC sparkline.o
    LINK redis-server
    INSTALL redis-sentinel
    CC redis-cli.o
    LINK redis-cli
    CC redis-benchmark.o
    LINK redis-benchmark
    CC redis-check-dump.o
    LINK redis-check-dump
    CC redis-check-aof.o
    LINK 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-172-31-5-213 redis-stable]$
```

Now connect to your Redis cluster (with your endpoint):

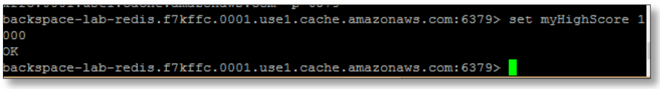
```
src/redis-cli -h YOUR_ENDPOINT_GOES_HERE -p 6379
```



```
[ec2-user@ip-172-31-5-213 redis-stable]$ src/redis-cli -h backspace-lab-redis.f7
kffc.0001.use1.cache.amazonaws.com -p 6379
backspace-lab-redis.f7kffc.0001.use1.cache.amazonaws.com:6379>
```

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

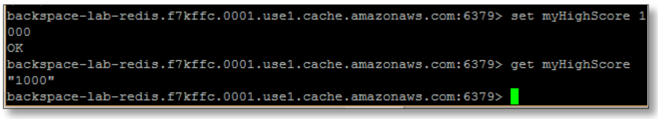
```
set myHighScore 1000
```



```
backspace-lab-redis.f7kffc.0001.use1.cache.amazonaws.com:6379> set myHighScore 1
000
OK
backspace-lab-redis.f7kffc.0001.use1.cache.amazonaws.com:6379>
```

Now read the key:

```
get myHighScore
```



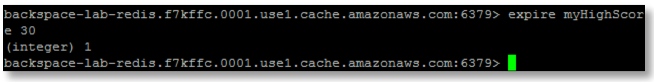
```
backspace-lab-redis.f7kffc.0001.use1.cache.amazonaws.com:6379> set myHighScore 1
000
OK
backspace-lab-redis.f7kffc.0001.use1.cache.amazonaws.com:6379> get myHighScore
"1000"
backspace-lab-redis.f7kffc.0001.use1.cache.amazonaws.com:6379>
```

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.

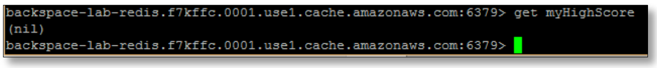


```
backspace-lab-redis.f7kffc.0001.use1.cache.amazonaws.com:6379> expire myHighScor
e 30
(integer) 1
backspace-lab-redis.f7kffc.0001.use1.cache.amazonaws.com:6379>
```

Now wait 30 s and read the key:

```
get myHighScore
```

If 30s has past it will return (nil)



```
backspace-lab-redis.f7kffc.0001.use1.cache.amazonaws.com:6379> get myHighScore
(nil)
backspace-lab-redis.f7kffc.0001.use1.cache.amazonaws.com:6379>
```

Now clean up the lab by deleting your cluster in the console.