

# Amazon ElastiCache

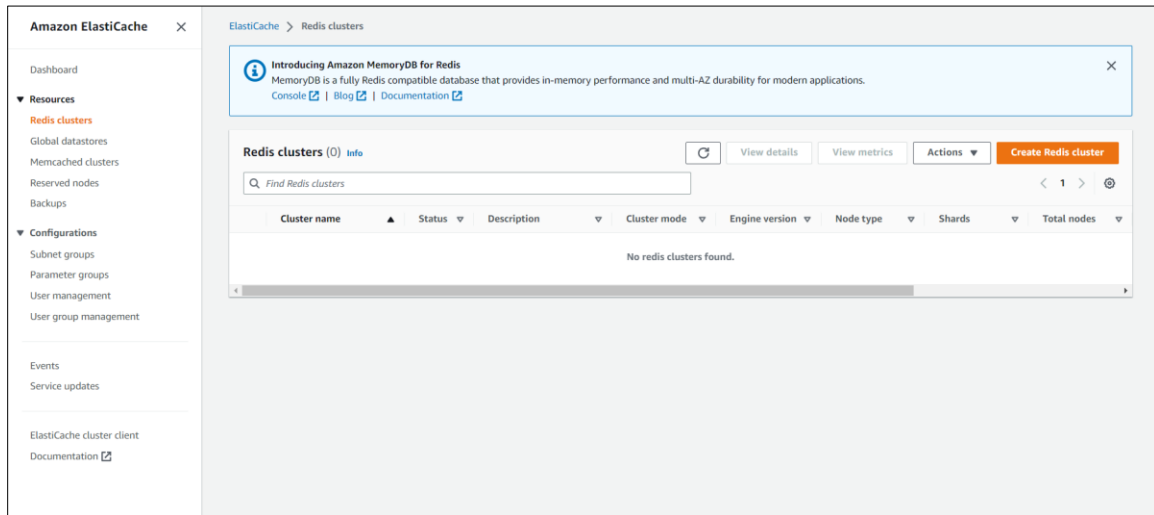
for Redis



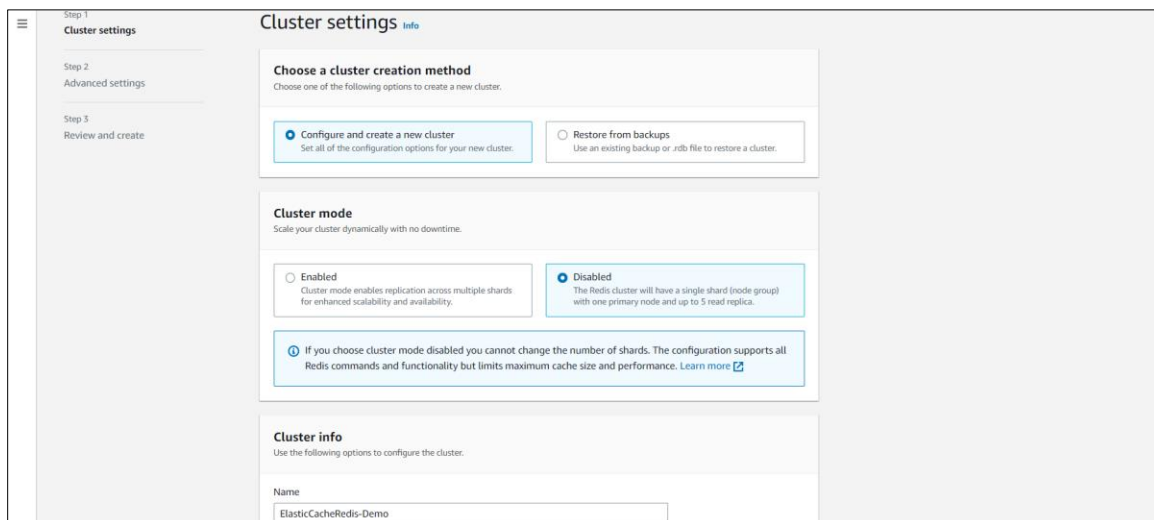
**By: Abdallah Mustafa**

# SIMPLE WALK THROUGH TO CREATE AMAZON ELASTICACHE FOR REDIS

Through the AWS Console navigate to Amazon ElastiCache. Then, on the left navigation bar, click on **Redis clusters**.



Click on **Create Redis cluster** button.



Fill the Cluster **name**. Scroll down to Cluster Settings, change the **node type** to **cache.t2.micro** for the demo purposes.

The screenshot shows the 'Cluster settings' page in the AWS Management Console. The page title is 'Cluster settings' with a subtitle 'Use the following options to configure the cluster.' The settings are as follows:

- Engine version:** 7.0 (selected from a dropdown menu)
- Port:** 6379 (text input)
- Parameter groups:** default.redis7.cluster.on (selected from a dropdown menu, with a refresh icon to the right)
- Node type:** cache.t2.micro (selected from a dropdown menu, with a tooltip showing '0.5 GiB memory' and 'Low to moderate network performance')
- Number of shards:** 3 (text input)
- Replicas per shard:** 2 (text input)

Set a **name** for the new subnet group. Leave other configuration as default and click **next**.

The screenshot shows the 'Subnet group settings' page in the AWS Management Console. The page title is 'Subnet group settings' with a subtitle 'A subnet group is a collection of subnets (typically private). Designate a subnet group for your clusters running in an Amazon Virtual Private Cloud (VPC) environment.' The settings are as follows:

- Subnet groups:** 'Create a new subnet group' (selected radio button)
- Name:** RedisCacheSubnet (text input)
- Description - optional:** (empty text input)
- VPC ID:** vpc-93ae32ee (selected from a dropdown menu, with a 'Create VPC' link to the right)
- Selected subnets (6):** A table showing one subnet selected.

Availability Zone	Subnet ID	CIDR block (IPv4)
us-east-1a	subnet-c6e4a999	172.31.32.0/20

On Advanced settings page, leave everything as is, as this is just a sample example to work with redis on AWS. Now click **next**.

Review your configurations and click on **Create**.

Meanwhile, while the Redis cache is being created, make sure to create an EC2 instance with default configuration, if you don't have any. Make sure to create/attach to that EC2 instance a security group which also allow access to port 6379 on source 0.0.0.0/0. So your instance can communicate with the redis server.

Once your redis cache and EC2 instance are available and ready. Connect to your EC2 instance and install redis:

- `sudo amazon-linux-extras install redis6 -y`

Once the installation completes, run the following command, which will start the redis server in the background

- `redis-server --daemonize yes`

Now, run this command, which will connect to your redis cache:

- `redis-cli -h <your redis endpoint> -p 6379`

The screenshot shows the AWS ElasticCache console interface. At the top, the breadcrumb navigation is 'ElastiCache > Redis clusters > elasticcacheredis-demo'. Below this, the cluster name 'elasticcacheredis-demo' is displayed with an 'Info' link and three buttons: 'Modify', 'Backup', and 'Delete'. The main section is titled 'Cluster details' and contains a table with the following information:

Cluster name	Description	Node type	Status
elasticcacheredis-demo	-	cache.t2.micro	Available
Engine	Engine version	Global datastore	Global datastore role
Redis	7.0.4	-	-
Update status	Cluster mode	Shards	Number of nodes
Up to date	Off	1	3
Data tiering	Multi-AZ	Auto-failover	Encryption in transit
Disabled	Enabled	Enabled	Disabled
Encryption at rest	Parameter group	Outpost ARN	Configuration endpoint
Disabled	default.redis7	-	-
Primary endpoint	Reader endpoint	ARN	
<a href="#">elasticcacheredis-demo.7e3l2o.ng.0001.use1.cache.amazonaws.com:6379</a>	<a href="#">elasticcacheredis-demo-ro.7e3l2o.ng.0001.use1.cache.amazonaws.com:6379</a>	<a href="#">arn:aws:elasticache:us-east-1:730651641800:replicationgroup:elasticcacheredis-demo</a>	

At the bottom of the console, there are tabs for 'Nodes', 'Metrics', 'Logs', 'Network and security', 'Maintenance and backups', 'Service updates', and 'Tags'. The 'Nodes' tab is currently selected.

Note: when you copy, make sure to omit the :6379 from the endpoint.

Hurray! Now you should be connected to redis cache. Let us try some commands:

- *set name <your name>*
- *get name (should return your name)*
- *set city <your city>*
- *set zipcode <your zipcode>*
- *set state <your state>*
- *keys \* (this will show all the keys you have set before)*

```
[root@ip-172-31-85-242 ec2-user]# redis-server --daemonize yes
[root@ip-172-31-85-242 ec2-user]# redis-cli -h elasticcacheredis-demo.7e312o.ng.0001.usel.cache.amazonaws.com -p 6379
elasticcacheredis-demo.7e312o.ng.0001.usel.cache.amazonaws.com:6379> set name Abdallah
OK
elasticcacheredis-demo.7e312o.ng.0001.usel.cache.amazonaws.com:6379> get name
"Abdallah"
elasticcacheredis-demo.7e312o.ng.0001.usel.cache.amazonaws.com:6379> set city Fairfield
OK
elasticcacheredis-demo.7e312o.ng.0001.usel.cache.amazonaws.com:6379> set ZipCode 52557
OK
elasticcacheredis-demo.7e312o.ng.0001.usel.cache.amazonaws.com:6379> set state Iowa
OK
elasticcacheredis-demo.7e312o.ng.0001.usel.cache.amazonaws.com:6379>
elasticcacheredis-demo.7e312o.ng.0001.usel.cache.amazonaws.com:6379> keys *
1) "name"
2) "ZipCode"
3) "city"
4) "state"
elasticcacheredis-demo.7e312o.ng.0001.usel.cache.amazonaws.com:6379>
```

i-0e8ed6bcb4dccc0dc (MyDemoInstance)  
PublicIPs: 3.92.49.154 PrivateIPs: 172.31.85.242

That is all for this short tutorial and guide. I hope it was somehow helpful.

Thank You & Good Luck!