

## Configure

### Create Elasticache

Database and analytics

## Amazon ElastiCache

Launch, manage, and scale a distributed in-memory cache

It provides a high performance, resizable, and cost effective in-memory cache, while removing complexity associated with deploying and managing a distributed cache environment. ElastiCache works with both the Redis and Memcached engines.

#### Get started

With a few clicks, you can get started with an Amazon ElastiCache cluster.

[Get started](#)

### Create cluster

#### ▼ Getting started with ElastiCache

Get started with ElastiCache by following these steps.



##### 1. Initial setup

Before you create an ElastiCache cluster determine the requirements for the cluster and set up VPC.

[Learn more](#)

[Create VPC](#)



##### 2. Create a cluster

Create a Redis or Memcached cluster with just a few clicks. To see which engine works best for you.

[Learn more](#)

[Create cluster](#)



##### 3. Connect with your application

When your cluster is in the available state, you can log into an Amazon EC2 instance and connect to the cluster. [Learn more](#)

### Select Redis

##### 1. Initial setup

Before you create an ElastiCache cluster determine the requirements for the cluster and set up VPC.

[Learn more](#)

[Create VPC](#)

##### 2. Create a cluster

Create a Redis or Memcached cluster with just a few clicks. To see which engine works best for you.

[Learn more](#)

[Create cluster](#)

[Create Redis cluster](#)

[Create Memcached cluster](#)

##### 3. Connect with your application

When your cluster is in the available state, you can log into an Amazon EC2 instance and connect to the cluster. [Learn more](#)

[Resource Overview](#)

[ElastiCache health](#)

### Choose cluster

## Choose a cluster creation method

Choose one of the following options to create a new cluster.



**Configure and create a new cluster**

Set all of the configuration options for your new cluster.



**Restore from backups**

Use an existing backup or .rdb file to restore a cluster.

# ElastiCache

## Cluster mode one

### Cluster mode



Scale your cluster dynamically with no downtime.

☒ **Enabled**

Cluster mode enables replication across multiple shards for enhanced scalability and availability.

☐ **Disabled**

The Redis cluster will have a single shard (node group) with one primary node and up to 5 read replica.

 Enabling cluster mode supports partitioning your data across up to 500 node groups and improves performance of Redis clusters. Some commands are unavailable in this mode. [Learn more](#) 

## Create new OAI

### Cluster info

Use the following options to configure the cluster.

Name

demo-redis

The name is required, can have up to 40 characters, and must begin with a letter. It should not end with a hyphen or contain two consecutive hyphens. Valid characters: A-Z, a-z, 0-9, and - (hyphen).

Description - *optional*

demo redis

## Select location

### Location

Choose whether to host the cluster in the AWS Cloud or on premises.

Location

☒ **AWS Cloud**

Use the AWS Cloud for your ElastiCache instances.

☐ **On premises**

Create your ElastiCache instances on an Outpost (through AWS Outposts). You need to create a subnet ID on an Outpost first.

Multi-AZ

☒ **Enable**

Multi-AZ provides enhanced high availability through automatic failover to a read replica, cross AZs, in case of a primary node failover.

Auto-failover

☒ **Enable**

ElastiCache Auto Failover provides enhanced high availability through automatic failover to a read replica in case of a primary node failover.

## Select cluster settings

### Cluster settings

Use the following options to configure the cluster.

#### Engine version

Version compatibility of the Redis engine that will run on your nodes.

6.2

#### Port

The port number that nodes accept connections on.

6379

#### Parameter groups

Parameter groups control the runtime properties of your nodes and clusters.

default.redis6.x.cluster.on

#### Node type

The type of node to be deployed and its associated memory size.

cache.t2.micro

0.5 GiB memory Low to moderate network performance

#### Number of shards

Enter the number of shards in this cluster, from 1 to 500.

3

#### Replicas per shard

Enter the number of replicas for each shard, from 0 to 5.

2

## Create subnet group

### Subnet group settings

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.

#### Subnet groups

☐ Choose existing subnet group

☒ Create a new subnet group

#### Name

redis-subnet-group

The name is required, can have up to 255 characters, and must begin with a letter. It should not end with a hyphen or contain two consecutive hyphens. Valid characters: A-Z, a-z, 0-9, and - (hyphen).

#### Description - optional

Description

## All settings as it is

### VPC ID

The identifier for the VPC environment where your cluster is to run.

vpc-0c930c8f6492bbf6c

Create VPC [↗](#)

**i** For Multi-AZ high availability mode, choose IDs for at least two subnets from two Availability Zones in the table below.

### Selected subnets (3)

Manage

Availability Zone ▲	Subnet ID ▼	CIDR block ▼
us-east-2a	subnet-09270613a3c12521f	172.31.0.0/20
us-east-2b	subnet-039d04a563f0ce14c	172.31.16.0/20
us-east-2c	subnet-07663653eeb747912	172.31.32.0/20

### Availability Zone placements

Use the following fields to configure placements for Availability Zones.

#### Slots and keyspaces

Distribution of the 16,384 Redis cluster keyspace slots across shards.

Equal distribution

#### Availability Zone placements

HA mode - Globally, distribute AZs to maximize AZ spread across shard masters. At the second level, spread nodes within a shard across AZs for within-shard HA. Low latency mode - For fast writes, put all shard masters in the same AZ.

No preference

< 1 >

Shards	Slots/keyspaces	Primary	Replica 1	Replica 2
Shard 1	Equal distribution	No preference	No preference	No preference
Shard 2	Equal distribution	No preference	No preference	No preference
Shard 3	Equal distribution	No preference	No preference	No preference

Cancel

Next

## Advance setting

ElastiCache / Redis clusters / Create

Step 1  
Cluster settings

Step 2  
**Advanced settings**

Step 3  
Review and create

### Advanced settings [Info](#)

#### Security

Use the following section to configure network security and data security for your cluster.

##### Encryption at rest

☐ Enable  
Enables encryption of data stored on disk.

##### Encryption in transit

☐ Enable  
Enables encryption of data that moves between the service and client.

#### Selected security groups (0)

A security group acts like a firewall that controls network access to your clusters.

Manage

Group ID [✕](#)



Name



No selected security groups

Add security groups by clicking on the Manage button.

Manage

### Backup

You can use backups to restore a cluster or seed a new cluster. The backup consists of the cluster's metadata, along with all of the data in the cluster.

☐ Enable automatic backups  
Allows ElastiCache to automatically create a daily backup of a set of replicas.

### Maintenance

Configure maintenance settings for the cluster.

#### Maintenance window

Specify the time range (UTC) for updates such as patching an operating system, updating drivers, and installing software or patches.

- ☒ No preference  
☐ Specify maintenance window

#### Auto upgrade minor versions

☐ Enable  
Automatically schedule cluster upgrade to the latest minor version, once it becomes available.  
Cluster upgrade will only be scheduled during the maintenance window.

#### Topic for Amazon SNS notification

Choose an SNS topic from the list, or enter the Amazon Resource Name (ARN) for an existing topic. If no topic is chosen, no notifications are sent.

Disable notifications



## ElastiCache

### Logs

Specify whether to provide the Redis slow logs or engine logs.

**Slow logs**

☐ **Enable**  
Provide the Redis slow log for queries that exceed a specified runtime.

Engine logs

☐ **Enable**  
Provide the engine log for queries that exceed a specified runtime.

## Tags

You can use tags to search and filter your clusters or track your AWS costs.

Downloaded from <http://ajphaphysoc.org/> at University of California, San Diego on November 10, 2014

Cancel Previous **Next**

**Create successfully**

Create

Actions

Refresh

Settings

Help

Filter: Search Clusters...

1 to 1 of 1 Clusters

	Cluster Name	Mode	Shards	Nodes	Node Type	Status	Update Action Status	Encryption in-transit	Encryption at-rest	Global Datastore	Global Index
	demo-redis	Redis	0	1 node	cache.t2.micro	creating	up to date	No	No	-	-

Name: demo-redis

Global Datastore: -

**Change SG rules**

EC2 > Security Groups

### Security Groups (1/1) [Info](#)

1

Create security group

<input checked="" type="checkbox"/>	Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules
<input checked="" type="checkbox"/>	sg-f353f2d1	sg-f353f2d1	default	vpc-6438241e	default VPC security group	327585364132	1 rule

sg-f353f2d1 - default

Details | **Inbound rules** | Outbound rules | Tags

#### Inbound rules

Edit inbound rules

Type	Protocol	Port range	Source	Description - optional
All traffic	All	All	sg-f353f2d1 (default)	-

# ElastiCache

## Add port

### Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

**Inbound rules**

Type	Protocol	Port range	Source	Description - optional
All traffic	All	All	Custom	
Custom TCP	TCP	6379	Custom	

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Cancel Preview changes Save rules

## Create EC2

Launch Instance

Connect

Actions

Filter by tags and attributes or search by keyword

1 to 1 of 1

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Na
	i-0acd9dd65cbfd81bf	t2.micro	us-east-1e	running	Initializing	None	ec2-34-204-81-91.com...	34.204.81.91	-	first-vm-

## connected

4. Connect to your instance using its Public DNS:

ec2-34-204-81-91.compute-1.amazonaws.com

Example:

```
ssh -i "first-vm-keys.pem" ubuntu@ec2-34-204-81-91.compute-1.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our [connection documentation](#).

Close

## connected

```
/bin/bash x /bin/bash x ubuntu@ip-172-31-55-155: ~ - 141x35
(base) prabhakar@Tue Jun 23-14:04:10:~/Downloads/awsKeys$ ssh -i "first-vm-keys.pem" ubuntu@ec2-34-204-81-91.compute-1.amazonaws.com
The authenticity of host 'ec2-34-204-81-91.compute-1.amazonaws.com (34.204.81.91)' can't be established.
ECDSA key fingerprint is SHA256:2nZkEBzXNGgShqr0Gi752usywMs52n7H5gvgcIgEuDI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-34-204-81-91.compute-1.amazonaws.com,34.204.81.91' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-1065-aws x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
```

# ElastiCache

## Update apt

```
ubuntu@ip-172-31-55-155:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
```

## Install redis

```
10 packages can be upgraded: none apt-get upgrade to see them.
ubuntu@ip-172-31-55-155:~$ sudo apt-get install redis
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libjemalloc1 redis-server redis-tools
Suggested packages:
  ruby-redis
```

## Copy redis endpoint

The screenshot shows the AWS Management Console interface for an ElastiCache Redis cluster. The cluster is named 'demo-redis' and is in the 'available' state. The 'Primary Endpoint' is highlighted with a red box and contains the text: 'demo-redis.gwlkbt.0001.use1.cache.amazonaws.com:6379'. Other details visible include the engine version (5.0.6), availability zones (us-east-1a), and various configuration options like encryption and backup retention.

## Paste here see connected

```
ubuntu@ip-172-31-55-155:~$ redis-cli -h demo-redis.gwlkbt.0001.use1.cache.amazonaws.com -p 6379
demo-redis.gwlkbt.0001.use1.cache.amazonaws.com:6379>
```

## See successfully cache store

```
demo-redis.gwlkbt.0001.use1.cache.amazonaws.com:6379> set name prabhakar
OK
demo-redis.gwlkbt.0001.use1.cache.amazonaws.com:6379> get name
"prabhakar"
demo-redis.gwlkbt.0001.use1.cache.amazonaws.com:6379> set city pune
OK
demo-redis.gwlkbt.0001.use1.cache.amazonaws.com:6379> set pin 123
OK
demo-redis.gwlkbt.0001.use1.cache.amazonaws.com:6379> set mobile 321
OK
demo-redis.gwlkbt.0001.use1.cache.amazonaws.com:6379> keys *
1) "mobile"
2) "pin"
3) "city"
4) "name"
demo-redis.gwlkbt.0001.use1.cache.amazonaws.com:6379>
```