

## **Q: What is Amazon ElastiCache?**

Amazon ElastiCache is a web service that makes it easy to deploy and run Memcached or Redis protocol-compliant server nodes in [the cloud](#). Amazon ElastiCache improves the performance of web applications by allowing you to retrieve information from a fast, managed, in-memory system, instead of relying entirely on slower disk-based databases. The service simplifies and offloads the management, monitoring and operation of in-memory environments, enabling your engineering resources to focus on developing applications. Using Amazon ElastiCache, you can not only improve load and response times to user actions and queries, but also reduce the cost associated with scaling web applications.

Amazon ElastiCache automates common administrative tasks required to operate a distributed in-memory key-value environment. Using Amazon ElastiCache, you can add a caching or in-memory layer to your application architecture in a matter of minutes via a few clicks of the AWS Management Console. Once a cluster is provisioned, Amazon ElastiCache automatically detects and replaces failed nodes, providing a resilient system that mitigates the risk of overloaded databases, which slow website and application load times. Through integration with Amazon CloudWatch monitoring, Amazon ElastiCache provides enhanced visibility into key performance metrics associated with your nodes. Amazon ElastiCache is protocol-compliant with Memcached and Redis, so code, applications, and popular tools that you use today with your existing Memcached or Redis environments will work seamlessly with the service. With the support for clustered configuration in Amazon ElastiCache, you get the benefits of fast, scalable and easy to use managed service that can meet the needs of your most demanding applications. As with all Amazon Web Services, there are no up-front investments required, and you pay only for the resources you use.

## **Q: What is in-memory caching and how does it help my applications?**

The in-memory caching provided by Amazon ElastiCache can be used to significantly improve latency and throughput for many read-heavy application workloads (such as social networking, gaming, media sharing and Q&A portals) or compute-intensive workloads (such as a recommendation engine). In-memory caching improves application performance by storing critical pieces of data in memory for low-latency access. Cached information may include the results of I/O-intensive database queries or the results of computationally-intensive calculations.

## **Q: Can I use Amazon ElastiCache for use cases other than caching?**

A: Yes. ElastiCache for Redis can be used as a primary in-memory key-value data store, providing fast, sub millisecond data performance, high availability and scalability up to 250 nodes and 250 shards, giving you up to 170.6 TB of in-memory data. See [here](#) for other use cases, such as leaderboards, rate limiting, queues, and chat.

## **Q: Can I use Amazon ElastiCache through AWS CloudFormation?**

AWS CloudFormation simplifies provisioning and management by providing AWS CloudFormation templates for quick and reliable provisioning of the services or applications. AWS CloudFormation provides comprehensive support for Amazon ElastiCache by providing templates to create cluster (both MemCached and Redis) and Replication Groups. The templates are up to date with the latest ElastiCache Redis announcement for clustered Redis configuration and provide flexibility and ease of use to Amazon ElastiCache customers.

## **Q: What does Amazon ElastiCache manage on my behalf?**

Amazon ElastiCache manages the work involved in setting up a distributed in-memory environment, from provisioning the server resources you request to installing the software. Once your environment is up and running, the service automates common administrative tasks such as failure detection and recovery, and software patching. Amazon ElastiCache provides detailed monitoring metrics associated with your nodes, enabling you to diagnose and react to issues very quickly. For example, you can set up thresholds and receive alarms if one of your nodes is overloaded with requests.

#### **Q: What are Amazon ElastiCache nodes, shards and clusters?**

A node is the smallest building block of an Amazon ElastiCache deployment. It is a fixed-size chunk of secure, network-attached RAM. Each node runs an instance of the Memcached or Redis protocol-compliant service and has its own DNS name and port. Multiple types of nodes are supported, each with varying amount of associated memory. A Redis shard is a subset of the cluster's keyspace, that can include a primary node and zero or more read-replicas. For more details on Redis deployments see the Redis section below. The shards add up to form a cluster.

#### **Q: Which engines does Amazon ElastiCache support?**

Amazon ElastiCache offers fully managed Redis, voted the most loved database by developers in the Stack Overflow 2020 Developer Survey, and Memcached for your most demanding applications that require sub-millisecond response times.

#### **Q: How do I get started with Amazon ElastiCache?**

If you are not already signed up for Amazon ElastiCache, you can click the "Get started" button on the [Amazon ElastiCache page](#) and complete the sign-up process. You must have an Amazon Web Services account; if you do not already have one, you will be prompted to create one when you begin the Amazon ElastiCache sign-up process. After you are signed up for ElastiCache, please refer to the [Amazon ElastiCache documentation](#), which includes the Getting Started Guide for [Amazon ElastiCache for Redis](#) or [Amazon ElastiCache for Memcached](#).

Once you have familiarized yourself with Amazon ElastiCache, you can launch a cluster within minutes by using the [AWS Management Console](#) or Amazon ElastiCache APIs.

#### **Q: How do I create a cluster?**

Clusters are simple to create, using the [AWS Management Console](#), Amazon ElastiCache APIs, or Command Line Tools. To launch a cluster using the AWS Management Console, click on the "Create" button in either the "Memcached" or "Redis" tab. From there, all you need to specify is your Cluster Identifier, Node Type, and Number of Nodes to create a cluster with the amount of memory you require. Alternatively, you can create your cluster using the CreateCacheCluster API or elasticache-create-cache-cluster command. If you do not specify an Availability Zone when creating a cluster, AWS will place it automatically based upon your memory requirements and available capacity.

#### **Q: What Node Types can I select?**

Amazon ElastiCache supports Nodes of the following types:

Current Generation Nodes:

- cache.m4.large: 6.42 GiB
- cache.m4.xlarge: 14.28 GiB
- cache.m4.2xlarge: 29.7 GiB

- cache.m4.4xlarge: 60.78 GiB
- cache.m4.10xlarge: 154.64 GiB
- cache.m5.large: 6.38 GiB
- cache.m5.xlarge: 12.93 GiB
- cache.m5.2xlarge: 26.04 GiB
- cache.m5.4xlarge: 52.26 GiB
- cache.m5.12xlarge: 157.12 GiB
- cache.m5.24xlarge: 314.32 GiB
- cache.m6g.large: 6.38 GiB
- cache.m6g.xlarge: 12.94 GiB
- cache.m6g.2xlarge: 26.05 GiB
- cache.m6g.4xlarge: 52.26 GiB
- cache.m6g.8xlarge: 103.68 GiB
- cache.m6g.12xlarge: 157.13 GiB
- cache.m6g.16xlarge: 209.55 GiB
- cache.r4.large: 12.3 GiB
- cache.r4.xlarge: 25.05 GiB
- cache.r4.2xlarge: 50.47 GiB
- cache.r4.4xlarge: 101.38 GiB
- cache.r4.8xlarge: 203.26 GiB
- cache.r4.16xlarge: 407 GiB
- cache.r5.large: 13.07 GiB
- cache.r5.xlarge: 26.32 GiB
- cache.r5.2xlarge: 52.82 GiB
- cache.r5.4xlarge: 105.81 GiB
- cache.r5.12xlarge: 317.77 GiB
- cache.r5.24xlarge: 635.61 GiB
- cache.r6g.large: 13.07 GiB

- cache.r6g.xlarge: 26.32 GiB
- cache.r6g.2xlarge: 52.82 GiB
- cache.r6g.4xlarge: 105.81 GiB
- cache.r6g.8xlarge: 209.55 GiB
- cache.r6g.12xlarge: 317.77 GiB
- cache.r6g.16xlarge: 419.1 GiB
- cache.t2.micro: 555 MB
- cache.t2.small: 1.55 GiB
- cache.t2.medium: 3.22 GiB

#### Previous Generation Nodes:

- cache.m1.small: 1.3 GiB
- cache.m1.medium: 3.35 GiB
- cache.m1.large: 7.1 GiB
- cache.m1.xlarge: 14.6 GiB
- cache.m2.xlarge: 16.7 GiB
- cache.m2.2xlarge: 33.8 GiB
- cache.m2.4xlarge: 68 GiB
- cache.m3.medium: 2.78 GiB
- cache.m3.large: 6.05 GiB
- cache.m3.xlarge: 13.3 GiB
- cache.m3.2xlarge: 27.9 GiB
- cache.r3.large: 13.5 GiB
- cache.r3.xlarge: 28.4 GiB
- cache.r3.2xlarge: 58.2 GiB
- cache.r3.4xlarge: 118 GiB
- cache.r3.8xlarge: 237 GiB
- cache.t1.micro: 213 MB
- cache.c1.xlarge: 6.6 GiB

Each Node Type above lists the memory available to Memcached or Redis after taking Amazon ElastiCache System Software overhead into account. The total amount of memory in a cluster is an integer multiple of the memory available in each shard. For example, a cluster consisting of ten shards of 6 GB each will provide 60 GB of total memory.

#### **Q: How do I access my nodes?**

Once your cluster is available, you can retrieve your node endpoints using the following steps on the [AWS Management Console](#):

- Navigate to the "Amazon ElastiCache" tab.
- Click on the "(Number of) Nodes" link and navigate to the "Nodes" tab.
- Click on the "Copy Node Endpoint(s)" button.

Alternatively, you can use the `DescribeCacheClusters` API to retrieve the Endpoint list.

You can then configure your Memcached or Redis client with this endpoint list and use your favorite programming language to add or delete data from your ElastiCache Nodes. In order to allow network requests to your nodes, you will need to authorize access. For a detailed explanation to get started, please refer to our Getting Started Guide for [Amazon ElastiCache for Redis](#) or [Amazon ElastiCache for Memcached](#).

#### **Q: What is a maintenance window? Will my nodes be available during software maintenance?**

You can think of the Amazon ElastiCache maintenance window as an opportunity to control when software patching occurs, in the event either are requested or required. If a "maintenance" event is scheduled for a given week, it will be initiated and completed at some point during the 60 minute maintenance window you identify.

Your nodes could incur some downtime during your maintenance window if software patching is scheduled. Please refer to Engine Version Management for more details. Patching can be user requested - for example cache software upgrade, or determined as required (if we identify any security vulnerabilities in the system or caching software). Software patching occurs infrequently (typically once every few months) and should seldom require more than a fraction of your maintenance window. If you do not specify a preferred weekly maintenance window when creating your Cluster, a 60 minute default value is assigned. If you wish to modify when maintenance is performed on your behalf, you can do so by modifying your DB Instance in the AWS Management Console or by using the `ModifyCacheCluster` API. Each of your Clusters can have different preferred maintenance windows, if you so choose.

---

## Billing

#### **Q: How will I be charged and billed for my use of Amazon ElastiCache?**

You pay only for what you use and there is no minimum fee. Pricing is per Node-hour consumed for each Node Type. Partial Node-hours consumed are billed as full hours. There is no charge for data transfer between Amazon EC2 and Amazon ElastiCache within the same Availability Zone. While standard Amazon EC2 Regional Data Transfer charges apply when transferring data between an Amazon EC2 instance and an Amazon ElastiCache Node in different Availability Zones of the same Region, you are only charged for the Data Transfer in or out of the Amazon EC2 instance. There is no Amazon ElastiCache Data Transfer charge for traffic in or out of the Amazon ElastiCache Node itself. Standard data transfer rates apply for data transferred out from a region. For more information, please visit the [pricing page](#).

**Q: When does billing of my Amazon ElastiCache Nodes begin and end?**

Billing commences for a node as soon as the node is available. Billing continues until the node is terminated, which would occur upon deletion.

**Q: What defines billable ElastiCache Node hours?**

Node hours are billed for any time your nodes are running in an "Available" state. If you no longer wish to be charged for your node, you must terminate it to avoid being billed for additional node hours.

**Q: Do your prices include taxes?**

Except as otherwise noted, our prices are exclusive of applicable taxes and duties, including VAT and applicable sales tax. For customers with a Japanese billing address, use of AWS services is subject to Japanese Consumption Tax. [Learn more](#).

---

## Reserved Nodes

**Q: What are Amazon ElastiCache Reserved Nodes?**

Reserved Nodes or Reserved Instance (RI) is an offering that provides you with a significant discount over on-demand usage when you commit to a one-year or three-year term. With Reserved Nodes, you can make a one-time, up-front payment to create a one or three year reservation to run your node in a specific Region and receive a significant discount off of the ongoing hourly usage charge. There are three Reserved Node types All Upfront, No Upfront and Partial Upfront that enable you to balance the amount you pay upfront with your effective hourly price.

**Q: How are Reserved Nodes different from On-Demand Nodes?**

Functionally, Reserved Nodes and On-Demand Nodes are exactly the same. The only difference is how your Node(s) are billed; with Reserved Nodes, you make a one-time up-front payment and receive a lower ongoing hourly usage rate (compared with On-Demand Nodes) for the duration of the term.

**Q: How do I purchase and create Reserved Nodes?**

You can use the "Purchase Reserved Nodes" option in the AWS Management Console. Alternatively, you can use the API tools to list the reservations available for purchase with the `DescribeReservedCacheNodesOfferings` API method and then purchase a cache node reservation by calling the `PurchaseReservedCacheNodesOffering` method.

Creating a Reserved Node is no different than launching an On-Demand Node. You simply specify the node class and Region for which you made the reservation. So long as your reservation purchase was successful, Amazon ElastiCache will apply the reduced hourly rate for which you are eligible to the new node.

**Q: Will there always be reservations available for purchase?**

Yes. Reserved Nodes are purchased for the Region rather than for the Availability Zone. This means that even if capacity is limited in one Availability Zone, reservations can still be purchased in that Region and used in a different Availability Zone within that Region.

**Q: How many Reserved Cache can I purchase?**

You can purchase up to 300 Reserved Nodes. If you wish to run more than 300 Nodes please complete the [Amazon ElastiCache Node request form](#).

**Q: What if I have an existing node that I'd like to convert to a Reserved Node?**

Simply purchase a node reservation with the same node class, within the same region as the node you are currently running and would like to reserve. If the reservation purchase is successful, Amazon ElastiCache will automatically apply your new hourly usage charge to your existing node.

**Q: If I sign up for a Reserved Node, when does the term begin? What happens to my node when the term ends?**

Pricing changes associated with a Reserved Node are activated once your request is received while the payment authorization is processed. You can follow the status of your reservation on the AWS Account Activity page or by using the `DescribeReservedCacheNodes` API. If the one-time payment cannot be successfully authorized by the next billing period, the discounted price will not take effect.

When your reservation term expires, your Reserved Node will revert to the appropriate On-Demand hourly usage rate for your node class and region.

**Q: How do I control which nodes are billed at the Reserved Node rate?**

The Amazon ElastiCache APIs for creating, modifying, and deleting nodes do not distinguish between On-Demand and Reserved Nodes so that you can seamlessly use both. When computing your bill, our system will automatically apply your Reservation(s), such that all eligible nodes are charged at the lower hourly Reserved Cache Node rate.

**Q: Can I move a Reserved Node from one Region or Availability Zone to another?**

Each Reserved Node is associated with a specific Region, which is fixed for the lifetime of the reservation and cannot be changed. Each reservation can, however, be used in any of the available AZs within the associated Region.

**Q: Can I cancel a reservation?**

No, you cannot cancel your reserved DB instance and the one-time payment (if applicable) is not refundable. You will continue to pay for every hour during your Reserved DB instance term regardless of your usage.

**Q: How do the payment options impact my bill?**

When you purchase an RI under the All Upfront payment option, you pay for the entire term of the RI in one upfront payment. You can choose to pay nothing upfront by choosing the No Upfront option. The entire value of the No Upfront RI is spread across every hour in the term and you will be billed for every hour in the term, regardless of usage. The Partial Upfront payment option is a hybrid of the All Upfront and No Upfront options. You make a small upfront payment, and you are billed a low hourly rate for every hour in the term regardless of usage.

---

## Security

**Q: How do I control access to Amazon ElastiCache?**

When not using [VPC](#), Amazon ElastiCache allows you to control access to your clusters through Cache Security Groups. A Security Group acts like a firewall, controlling network access to your cluster. By default, network

access is turned off to your clusters. If you want your applications to access your cluster, you must explicitly enable access from hosts in specific [EC2 security groups](#). This process is called ingress.

To allow network access to your cluster, create a Security Group and link the desired EC2 security groups (which in turn specify the EC2 instances allowed) to it. The Security Group can be associated with your cluster at the time of creation, or using the "Modify" option on the [AWS Management Console](#).

Please note that IP-range based access control is currently not enabled for clusters. All clients to a cluster must be within the EC2 network, and authorized via security groups as described above.

When using VPC, please see [here](#) for more information.

**Q: Can programs running on servers in my own data center access Amazon ElastiCache?**

Yes. You can access an Amazon ElastiCache cluster from an application running in your data center providing there is connectivity between your VPC and the data center either through VPN or Direct Connect. The details are described [here](#).

**Q: Can programs running on EC2 instances in a VPC access Amazon ElastiCache?**

Yes, EC2 instances in a VPC can access Amazon ElastiCache if the ElastiCache cluster was created within the VPC. Details on how to create an Amazon ElastiCache cluster within a VPC are given [here](#).

**Q: What is Amazon Virtual Private Cloud (VPC) and why may I want to use with Amazon ElastiCache?**

Amazon VPC lets you create a virtual networking environment in a private, isolated section of the Amazon Web Services (AWS) cloud, where you can exercise complete control over aspects such as private IP address ranges, subnets, routing tables and network gateways. With Amazon VPC, you can define a virtual network topology and customize the network configuration to closely resemble a traditional IP network that you might operate in your own datacenter.

One of the scenarios where you may want to use Amazon ElastiCache in a VPC is if you want to run a public-facing web application, while still maintaining non-publicly accessible backend servers in a private subnet. You can create a public-facing subnet for your webservers that has access to the Internet, and place your backend infrastructure in a private-facing subnet with no Internet access. Your backend infrastructure could include RDS DB Instances and an Amazon ElastiCache Cluster providing the in-memory layer. For more information about Amazon VPC, refer to the [Amazon Virtual Private Cloud User Guide](#).

**Q: How do I create an Amazon ElastiCache Cluster in VPC?**

For a walk through example of creating an Amazon ElastiCache Cluster in VPC, refer to the [Amazon ElastiCache User Guide](#).

Following are the pre-requisites necessary to create a cluster within a VPC:

- You need to have a VPC set up with at least one subnet. For information on creating Amazon VPC and subnets refer to the [Getting Started Guide for Amazon VPC](#).
- You need to have a Subnet Group (for [Redis](#) or [Memcached](#)) defined for your VPC.
- You need to have a [VPC Security Group](#) defined for your VPC (or you can use the default provided).
- In addition, you should allocate adequately large CIDR blocks to each of your subnets so that there are enough spare IP addresses for Amazon ElastiCache to use during maintenance activities such as cache node replacement.

## **Q: How do I create an Amazon ElastiCache Cluster in an existing VPC?**

Creating an Amazon ElastiCache Cluster in an existing VPC is the same as that for a newly created VPC. Here are more details for [Redis](#) or [Memcached](#).

## **Q: How do I connect to an ElastiCache Node in VPC?**

Amazon ElastiCache Nodes, deployed within a VPC, can be accessed by EC2 Instances deployed in the same VPC. If these EC2 Instances are deployed in a public subnet with associated Elastic IPs, you can access the EC2 Instances via the internet.

If you want to access Amazon ElastiCache Nodes, deployed within a VPC, from the Internet or from EC2 Instances outside the VPC, please see guidelines for [Redis](#) or [Memcached](#).

ElastiCache ensures that both the DNS name and the IP address of the cache node remain the same when cache nodes are recovered in case of failure.

## **Q: What is a Subnet Group and why do I need one?**

A Subnet Group is a collection of [subnets](#) that you must designate for your Amazon ElastiCache Cluster in a VPC. A Subnet Group is created using the [Amazon ElastiCache Console](#). Each Subnet Group should have at least one subnet. Amazon ElastiCache uses the Subnet Group to select a subnet. The IP Addresses from the selected subnet are then associated with the Node Endpoints. Furthermore, Amazon ElastiCache creates and associates Elastic Network Interfaces to nodes with the previously mentioned IP addresses.

Please note that, we strongly recommend you use the DNS Names to connect to your nodes as the underlying IP addresses can change (e.g., after cache node replacement).

## **Q: Can I change the Subnet Group of my ElastiCache Cluster?**

An existing Subnet Group can be updated to add more subnets either for existing Availability Zones or for new Availability Zones added since the creation of the ElastiCache Cluster. However, changing the Subnet Group of a deployed cluster is not currently allowed.

## **Q: How is using Amazon ElastiCache inside a VPC different from using it outside?**

The basic functionality of Amazon ElastiCache remains the same whether VPC is used or not. Amazon ElastiCache manages automatic failure detection, recovery, scaling, auto discovery, and software patching whether your ElastiCache Cluster is inside or outside a VPC.

Within a VPC, nodes of an ElastiCache cluster only have a private IP address (within a subnet that you define). Outside of a VPC, the access to the ElastiCache cluster can be controlled using Security Groups as described here.

## **Q: Can I move my existing ElastiCache Cluster from outside VPC into my VPC?**

No, you cannot move an existing Amazon ElastiCache Cluster from outside VPC into a VPC. You will need to create a new Amazon ElastiCache Cluster inside the VPC.

## **Q: Can I move my existing ElastiCache Cluster from inside VPC to outside VPC?**

Currently, direct migration of ElastiCache Cluster from inside to outside VPC is not supported. You will need to create a new Amazon ElastiCache Cluster outside VPC.

## **Q: How do I control network access to my cluster?**

Amazon ElastiCache allows you to control access to your cluster and therefore the nodes using Security Groups in non-VPC deployments. A Security Group acts like a firewall controlling network access to your node. By default, network access is turned off to your nodes. If you want your applications to access your node, you can set your Security Group to allow access from EC2 Instances with specific EC2 Security Group membership or IP ranges. This process is called ingress. Once ingress is configured for a Security Group, the same rules apply to all nodes associated with that Security Group. Security Groups can be configured with the "Security Groups" section of the [Amazon ElastiCache Console](#) or using the Amazon ElastiCache APIs.

In VPC deployments, access to your nodes is controlled using the VPC Security Group and the Subnet Group. The [VPC Security Group](#) is the VPC equivalent of the Security Group.

**Q: What precautions should I take to ensure that my ElastiCache Nodes in VPC are accessible by my application?**

You are responsible for modifying routing tables and networking ACLs in your VPC to ensure that your ElastiCache Nodes are reachable from your client instances in the VPC. To learn more see the [Amazon ElastiCache for Redis](#) or [Amazon ElastiCache for Memcached](#) Documentation.

**Q: Can I use Security Groups to configure the clusters that are part of my VPC?**

No, Security Groups are not used when operating in a VPC. Instead they are used in the non VPC settings. When creating a cluster in a VPC you will need to use VPC Security Groups.

**Q: Can I associate a regular EC2 security group with a cluster that is launched within a VPC?**

No, you can only associate VPC security groups that are part of the same VPC as your cluster.

**Q: Can nodes of an ElastiCache cluster span multiple subnets?**

Yes, nodes of an Amazon ElastiCache cluster can span multiple subnets as long as the subnets are part of the same Subnet Group that was associated with the ElastiCache Cluster at creation time.

---

## Parameter Groups

**Q: What are Parameter Groups? How are they helpful?**

A Parameter Group acts as a "container" for engine configuration values that can be applied to one or more clusters. If you create a cluster without specifying a Parameter Group, a default Parameter Group is used. This default group contains engine defaults and Amazon ElastiCache system defaults optimized for the cluster you are running. However, if you want your cluster to run with your custom-specified engine configuration values, you can simply create a new Parameter Group, modify the desired parameters, and modify the cluster to use the new Parameter Group. Once associated, all clusters that use a particular Parameter Group get all the parameter updates to that Parameter Group. For more information on configuring Parameter Groups, please refer to the [Amazon ElastiCache for Redis](#) or [Amazon ElastiCache for Memcached](#) User Guide.

**Q: How do I choose the right configuration parameters for my Cluster(s)?**

Amazon ElastiCache by default chooses the optimal configuration parameters for your cluster taking into account the Node Type's memory/compute resource capacity. However, if you want to change them, you can do so using our configuration management APIs. Please note that changing configuration parameters from recommended values can have unintended effects, ranging from degraded performance to system crashes, and

should only be attempted by advanced users who wish to assume these risks. For more information on changing parameters, please refer to the [Amazon ElastiCache User Guide](#).

**Q: How do I see the current setting for my parameters for a given Parameter Group?**

You can use the [AWS Management Console](#), Amazon ElastiCache APIs, or Command Line Tools to see information about your Parameter Groups and their corresponding parameter settings.

---

## Memcached

### Features

**Q: What can I cache using Amazon ElastiCache for Memcached?**

You can cache a variety of objects using the service, from the content in persistent data stores (such as Amazon RDS, DynamoDB, or self-managed databases hosted on EC2) to dynamically generated web pages (with Nginx for example), or transient session data that may not require a persistent backing store. You can also use it to implement high-frequency counters to deploy admission control in high volume web applications.

**Q: Can I use Amazon ElastiCache for Memcached with an AWS persistent data store such as Amazon RDS or Amazon DynamoDB?**

Yes, Amazon ElastiCache is an ideal front-end for data stores like Amazon RDS or Amazon DynamoDB, providing a high-performance middle tier for applications with extremely high request rates and/or low latency requirements.

**Q: I use Memcached today. How do I migrate to Amazon ElastiCache?**

Amazon ElastiCache is protocol-compliant with Memcached. Therefore, you can use standard Memcached operations like get, set, incr and decr in exactly the same way as you would in your existing Memcached deployments. Amazon ElastiCache supports both the text and binary protocols. It also supports most of the standard stats results, which can also be viewed as graphs via CloudWatch. As a result, you can switch to using Amazon ElastiCache without recompiling or re-linking your applications - the libraries you use will continue to work. To configure the cache servers your application accesses, all you will need to do is to update your application's Memcached config file to include the endpoints of the servers (nodes) we provision for you. You can simply use the "Copy Node Endpoints" option on the AWS Management Console or the "DescribeCacheClusters" API to get a list of the endpoints. As with any migration process, we recommend thorough testing of your new Amazon ElastiCache deployment before completing the cut over from your current solution.

You can access Amazon ElastiCache cluster in an Amazon VPC from either Amazon EC2 network, or from your own Data Center, please refer to [Amazon VPC access patterns](#) for more details.

Amazon ElastiCache uses DNS entries to allow client applications to locate servers (nodes). The DNS name for a node remains constant, but the IP address of a node can change over time, for example, when nodes are auto replaced after a failure on a non-VPC installation. See this FAQ for recommendations to deal with node failures.

---

## Configuration and Scaling

## **Q: How do I select an appropriate Node Type for my application?**

Though there is no precise answer for this question, with Amazon ElastiCache, you don't need to worry about getting the number of nodes exactly right, as you can very easily add or remove nodes later. The following two inter-related aspects could be considered for the choice of your initial configuration:

- The total memory required for your data to achieve your target cache-hit rate, and
- The number of nodes required to maintaining acceptable application performance without overloading the database backend in the event of node failure(s).

The amount of memory required is dependent upon the size of your data set and the access patterns of your application. To improve fault tolerance, once you have a rough idea of the total memory required, divide that memory into enough nodes such that your application can survive the loss of one or two nodes. For example, if your memory requirement is 13GB, you may want to use two cache.m4.large nodes instead of using one cache.m4.xlarge node. It is important that other systems such as databases will not be overloaded if the cache-hit rate is temporarily reduced during failure recovery of one or more of nodes. Please refer to the [Amazon ElastiCache User Guide](#) for more details.

## **Q: Can a cluster span multiple Availability Zones?**

Yes. When creating a cluster or adding nodes to an existing cluster, you can chose the availability zones for the new nodes. You can either specify the requested amount of nodes in each availability zones or select "spread nodes across zones". If the cluster is in VPC, nodes can only be placed in availability zones that are part of the selected cache subnet group. For additional details please see [ElastiCache VPC documentation](#).

## **Q: How many nodes can I run per region in Amazon ElastiCache Memcached?**

You can run a maximum of 300 nodes per region. If you need more nodes, please fill in the [ElastiCache Limit Increase Request form](#).

## **Q: How does Amazon ElastiCache respond to node failure?**

The service will detect the node failure and react with the following automatic steps:

- Amazon ElastiCache will repair the node by acquiring new service resources, and will then redirect the node's existing DNS name to point to the new service resources. For VPC installations, ElastiCache will ensure that both the DNS name and the IP address of the node remain the same when nodes are recovered in case of failure. For non-VPC installations, ElastiCache will ensure that the DNS name of a node is unchanged; however, the underlying IP address of the node can change.
- If you associated an SNS topic with your cluster, when the new node is configured and ready to be used, Amazon ElastiCache will send an SNS notification to let you know that node recovery occurred. This allows you to optionally arrange for your applications to force the Memcached client library to attempt to reconnect to the repaired nodes. This may be important, as some Memcached libraries will stop using a server (node) indefinitely if they encounter communication errors or timeouts with that server.

## **Q: If I determine that I need more memory to support my application, how do I increase the total memory with Amazon ElastiCache?**

You could add more nodes to your existing Memcached Cluster by using the "Add Node" option on "Nodes" tab for your Cache Cluster on the AWS Management Console or calling the `ModifyCacheCluster` API.

## Compatibility

### Q: How does Amazon ElastiCache interact with other Amazon Web Services?

Amazon ElastiCache is ideally suited as a front-end for Amazon Web Services like Amazon RDS and Amazon DynamoDB, providing extremely low latency for high performance applications and offloading some of the request volume while these services provide long lasting data durability. The service can also be used to improve application performance in conjunction with Amazon EC2 and EMR.

### Q: Is Amazon ElastiCache better suited to any specific programming language?

Memcached client libraries are available for many, if not all of the popular programming languages. If you encounter any issues with specific Memcached clients when using Amazon ElastiCache, please engage us via the [Amazon ElastiCache community forum](#).

### Q: What popular Memcached libraries are compatible with Amazon ElastiCache?

Amazon ElastiCache does not require specific client libraries and works with existing Memcached client libraries without recompilation or application re-linking (Memcached 1.4.5 and later); examples include libMemcached (C) and libraries based on it (e.g. PHP, Perl, Python), spyMemcached (Java) and fauna (Ruby).

---

## Auto Discovery

### Q: What is Auto Discovery and what can I do with it?

Auto Discovery is a feature that saves developers time and effort, while reducing complexity of their applications. Auto Discovery enables automatic discovery of cache nodes by clients when they are added to or removed from an Amazon ElastiCache cluster. Until now to handle cluster membership changes, developers must update the list of cache node endpoints manually. Depending on how the client application is architected, typically a client initialization, by shutting down the application and restarting it, is needed resulting in downtime. Through Auto Discovery we are eliminating this complexity. With Auto Discovery, in addition to being backwards protocol-compliant with the Memcached protocol, Amazon ElastiCache provides clients with information on cache cluster membership. A client capable of processing the additional information reconfigures itself, without any initialization, to use the most current nodes of an Amazon ElastiCache cluster.

### Q: How does Auto Discovery work?

An Amazon ElastiCache cluster can be created with nodes that are addressable via named endpoints. With Auto Discovery the Amazon ElastiCache cluster is also given a unique Configuration Endpoint which is a DNS Record that is valid for the lifetime of the cluster. This DNS Record contains the DNS Names of the nodes that belong to the cluster. Amazon ElastiCache will ensure that the Configuration Endpoint always points to at least one such "target" node. A query to the target node then returns endpoints for all the nodes of the cluster in question. After this, you can connect to the cluster nodes just as before and use the Memcached protocol commands such as get, set, incr and decr. For more details, see [here](#). To use Auto Discovery, you will need an Auto Discovery capable client. Auto Discovery clients for .Net , Java and PHP are available for download from the [Amazon ElastiCache console](#). Upon initialization, the client will automatically determine the current members of the Amazon ElastiCache cluster using the Configuration Endpoint. When you make changes to your cache cluster by adding or removing nodes or if a node is replaced upon failure, the Auto Discovery client automatically determines the changes and you do not need to initialize your clients manually.

## **Q: How can I get started using Auto Discovery?**

To get started, download the Amazon ElastiCache Cluster Client by clicking the “Download ElastiCache Cluster Client” link on the [Amazon ElastiCache console](#). Before you can download, you must have an Amazon ElastiCache account; if you do not already have one, you can sign up from the Amazon ElastiCache detail page. After you download the client, you can begin setting up and activating your Amazon ElastiCache cluster by visiting the [Amazon ElastiCache console](#). More details can be found [here](#).

## **Q: If I continue to use my own Memcached clients with my ElastiCache cluster – will I be able to get this feature?**

No, you will not get the Auto Discovery feature with the existing Memcached clients. To use the Auto Discovery feature a client must be able to use a Configuration Endpoint and determine the cluster node endpoints. You may either use the Amazon ElastiCache Cluster Client or extend your existing Memcached client to include the [Auto Discovery command set](#).

## **Q: What are the minimum hardware / software requirements for Auto Discovery?**

To take advantage of Auto Discovery, an Auto Discovery capable client must be used to connect to an Amazon ElastiCache Cluster. Amazon ElastiCache currently supports Auto Discovery capable clients for .Net , Java and PHP. These can be downloaded from the [Amazon ElastiCache console](#). Our customers can create clients for any other language by building upon the popular Memcached clients available.

## **Q: How do I modify or write my own Memcached client to support auto-discovery?**

You can take any Memcached Client Library and add support for Auto Discovery. If you would like to add or modify your own client to enable Auto Discovery, please refer to the [Auto Discovery command set documentation](#).

## **Q: Can I continue to work with my existing Memcached client if I don't need Auto-discovery?**

Yes, Amazon ElastiCache is still Memcached protocol compliant and does not require you to change your clients. However, for taking advantage of auto-discovery feature, we had to enhance the Memcached client capabilities. If you choose to not use the Amazon ElastiCache Cluster Client, you can continue to use your own clients or modify your own client library to understand the auto-discovery command set.

## **Q: Can I have heterogeneous clients when using Auto Discovery?**

Yes, the same Amazon ElastiCache cluster can be connected through an Auto Discovery capable Client and the traditional Memcached client at the same time. Amazon ElastiCache remains 100% Memcached compliant.

## **Q: Can I stop using Auto Discovery?**

Yes, you can stop using Auto Discovery anytime. You can disable Auto Discovery by specifying the mode of operation during the Amazon ElastiCache Cluster client initialization. Also, since Amazon ElastiCache continues to support Memcached 100% you may use any Memcached protocol-compliant client as before.

---

## **Engine Version Management**

### **Q: Can I control if and when the engine version powering Amazon ElastiCache Cluster is upgraded to new supported versions?**

Amazon ElastiCache allows you to control if and when the Memcached protocol-compliant software powering your cluster is upgraded to new versions supported by Amazon ElastiCache. This provides you with the flexibility to maintain compatibility with specific Memcached versions, test new versions with your application before deploying in production, and perform version upgrades on your own terms and timelines. Version upgrades involve some compatibility risk, thus they will not occur automatically and must be initiated by you. This approach to software patching puts you in the driver's seat of version upgrades, but still offloads the work of patch application to Amazon ElastiCache. You can learn more about version management by reading the FAQs that follow. Alternatively, you can refer to the [Amazon ElastiCache User Guide](#). While Engine Version Management functionality is intended to give you as much control as possible over how patching occurs, we may patch your cluster on your behalf if we determine there is any security vulnerability in the system or cache software.

**Q: How do I specify which supported Memcached Version my Cluster should run?**

You can specify any currently supported version (minor and/or major) when creating a new cluster. If you wish to initiate an upgrade to a supported engine version release, you can do so using the "Modify" option for your cluster. Simply specify the version you wish to upgrade to via the "Cache Engine Version" field. The upgrade will then be applied on your behalf either immediately (if the "Applied Immediately" option is checked) or during the next scheduled maintenance window for your cluster.

**Q: Can I test my cluster against a new version before upgrading?**

Yes. You can do so by creating a new cluster with the new engine version. You can point your development/staging application to this cluster, test it and decide whether or not to upgrade your original cluster.

**Q: Does Amazon ElastiCache provide guidelines for supporting new Memcached version releases and/or deprecating versions that are currently supported?**

Over time, we plan to support additional Memcached versions for Amazon ElastiCache, both major and minor. The number of new version releases supported in a given year will vary based on the frequency and content of the Memcached version releases and the outcome of a thorough vetting of the release by our engineering team.

**Q: Which version of the Memcached wire protocol does Amazon ElastiCache support?**

Amazon ElastiCache supports the Memcached text and binary protocol of versions 1.6.6, 1.5.16, 1.5.10, 1.4.34, 1.4.33, 1.4.24, 1.4.14, and 1.4.5 of Memcached.

**Q: What should I do to upgrade to the latest Memcached version?**

You can upgrade your existing Memcached cluster by using the Modify process. When upgrading from an older version of Memcached to Memcached version 1.4.33 or newer, please ensure that your existing parameter max\_chunk\_size values satisfies conditions needed for slab\_chunk\_max parameter. Please review upgrade prerequisites [here](#).

---

## Redis

### Features

**Q: What is Amazon ElastiCache for Redis?**

Amazon ElastiCache for Redis is a web service that makes it easy to deploy and run Redis protocol-compliant server nodes in the cloud. The service enables the management, monitoring, and operation of Redis nodes; creation, deletion, and modification of the nodes can be carried out through the Amazon ElastiCache console, the command line interface (CLI), or the web service APIs. Amazon ElastiCache for Redis supports high-availability configurations, including Redis cluster-mode enabled and cluster-mode disabled with auto-failover from primary to replica.

**Q: Is Amazon ElastiCache for Redis protocol-compliant with open source Redis?**

Yes, Amazon ElastiCache for Redis is protocol-compliant with open source Redis. Code, applications, drivers and tools a customer uses today with their existing standalone Redis data store will continue to work with Amazon ElastiCache for Redis and no code changes will be required for existing Redis deployments migrating to Amazon ElastiCache for Redis unless noted. We currently support Redis 6.0.5, 5.0.6, 5.0.5, 5.0.4, 5.0.3, 5.0.0, 4.0.10, 3.2.10, 3.2.6, 3.2.4, 2.8.24, 2.8.23, 2.8.22, 2.8.21, 2.8.19, 2.8.6, and 2.6.13.

**Q: How much does Amazon ElastiCache for Redis cost?**

Please see our [pricing page](#) for current pricing information.

**Q: What are Amazon ElastiCache for Redis nodes, clusters, and replication groups?**

An Amazon ElastiCache for Redis node is the smallest building block of an Amazon ElastiCache for Redis deployment. Each Amazon ElastiCache for Redis node supports the Redis protocol and has its own DNS name and port. Multiple types of Amazon ElastiCache for Redis nodes are supported, each with varying amount of CPU capability, and associated memory. An Amazon ElastiCache for Redis node may take on a primary or a read replica role. A primary node can be replicated to multiple read replica nodes. An Amazon ElastiCache for Redis cluster is a collection of one or more Amazon ElastiCache for Redis nodes; the primary node will be in the primary cluster and the read replica node will be in a read replica cluster. A cluster manages a logical key space, where each node is responsible for a part of the key space. Most of your management operations will be performed at the cluster level. An Amazon ElastiCache for Redis replication group encapsulates the primary and read replica clusters for a Redis installation. A replication group will have only one primary cluster and zero or many read replica clusters. All nodes within a replication group (and consequently cluster) will be of the same node type, and have the same parameter and security group settings.

**Q: Does Amazon ElastiCache for Redis support Redis persistence?**

Amazon ElastiCache for Redis doesn't support the AOF (Append Only File) feature but you can achieve persistence by snapshotting your Redis data using the Backup and Restore feature. Please see [here](#) for details.

**Q: How can I migrate from Amazon ElastiCache for Memcached to Amazon ElastiCache for Redis and vice versa?**

We currently do not support automatically migrating from Memcached to Redis or vice versa. You may, however, use a Memcached client to read from a Memcached cluster and use a Redis client to write to a Redis cluster. Similarly, you may read from a Redis cluster using a Redis client and use a Memcached client to write to a Memcached cluster. Make sure to consider the differences in data format, and cluster configuration between the two engines.

**Q: Does Amazon ElastiCache for Redis support Multi-AZ operation?**

Yes, with Amazon ElastiCache for Redis you can create a read replica in another AWS Availability Zone. Upon a failure of a node, we will provision a new node. In scenarios where the primary node fails ElastCache for Redis

will automatically promote an existing read replica to the primary role. For more details on how to handle node failures see [here](#).

#### **Q: What options does Amazon ElastiCache for Redis provide in case of node failures?**

Amazon ElastiCache for Redis will repair the node by acquiring new service resources, and will then redirect the node's existing DNS name to point to the new service resources. Thus, the DNS name for a Redis node remains constant, but the IP address of a Redis node can change over time. If you have a replication group with one or more read replicas and Multi-AZ is enabled, then in case of primary node failure, Amazon ElastiCache will automatically detect the failure, select a replica and promote it to become the new primary. It will also propagate the DNS so that you can continue to use the primary endpoint and after the promotion it will point to the newly promoted primary. For more details see the [Multi-AZ](#) section of this FAQ.

When Redis replication option is selected with Multi-AZ disabled, in case of primary node failure, you will be given the option to initiate a failover to a read replica node. The failover target can be in the same zone or another zone. To fallback to the original zone, promote the read replica in the original zone to be the primary. You may choose to architect your application to force the Redis client library to reconnect to the repaired Redis server node. This can help as some Redis libraries will stop using a server indefinitely when they encounter communication errors or timeouts.

#### **Q: How does failover work?**

When deploying ElastiCache for Redis with Cluster Mode Disabled, for Multi-AZ enabled replication groups, the failover behavior is described in the [Multi-AZ](#) section of this FAQ. If you choose not to enable Multi-AZ, then if Amazon ElastiCache monitors the primary node, and in case the node becomes unavailable or unresponsive, Amazon ElastiCache for Redis will repair the node by acquiring new service resources, and will then redirect the node's existing DNS name to point to the new service resources. Thus, the DNS name for a Redis node remains constant, but the IP address of a Redis node can change over time. However, if the primary node cannot be healed (and your Multi-AZ is disabled) you will have the choice to promote one of the read replicas to be the new primary. See [here](#) for how to select a new primary. The DNS record of the primary's endpoint will be updated to point to the promoted read replica node. A read replica node in the original primary's AZ will then be created to be a read replica in the replication group and will follow the new primary.

When deploying ElastiCache for Redis with Cluster Mode Enabled, you are spreading the cache key space across multiple shards. This means that your data and read/write access to that data is spread across multiple Redis nodes across Multiple-AZs ([required](#) with Cluster Mode Enabled). The role of primary node will automatically fail over to one of the read replicas. There is no need to create and provision a new primary node, because ElastiCache will handle this transparently. This failover and replica promotion ensure that you can resume writing to the new primary as soon as promotion is complete.

#### **Q: Are my read replicas available during a primary node failure?**

Yes, during a primary node failure, the read replicas continue to service requests. After the primary node is restored, either as a healed node or as a promoted read replica, there is a brief period during which the read replicas will not serve any requests as they sync the cache information from the primary.

#### **Q: How do I configure parameters of my Amazon ElastiCache for Redis nodes?**

You can configure your Redis installation using a cache parameter group, which must be specified for a Redis cluster. All read replica clusters use the parameter group of their primary cluster. A Redis parameter group acts as a "container" for Redis configuration values that can be applied to one or more Redis primary clusters. If you create a Redis primary cluster without specifying a cache parameter group, a default parameter group is used.

This default group contains defaults for the node type you plan to run. However, if you want your Redis primary cluster to run with specified configuration values, you can simply create a new cache parameter group, modify the desired parameters, and modify the primary Redis cluster to use the new parameter group.

**Q: Can I access Redis through the Amazon ElastiCache console?**

Yes, Redis appears as an Engine option in the Amazon ElastiCache console. You can create a new Redis cache cluster with the Launch Wizard by choosing the Redis engine. You can also modify or delete an existing Redis cluster using the Amazon ElastiCache console.

**Q: Can Amazon ElastiCache for Redis clusters be created in an Amazon VPC?**

Yes. If your account is a VPC by default account, your Redis clusters will be created within the default VPC associated with your account. Using the Amazon ElastiCache console, you can specify a different VPC when you create your cluster.

**Q: How can I secure my Redis cluster?**

Amazon ElastiCache for Redis supports two methods to secure your Redis cluster. You can choose between Redis AUTH or Managed Role-Based Access Control (RBAC), which are both opt-in feature and require that encryption-in transit is enabled. Redis AUTH allows you to add a password to secure access to your Redis cluster and is supported in version 3.2.6 onwards. Starting with [Redis 6](#), the RBAC feature enables you to create and manage users and user groups to secure your Redis cluster. You can assign the users to user groups aligned with a specific role (e.g. administrators, human resources, analytics, etc.) that are then deployed to one or more Amazon ElastiCache for Redis replication groups. By doing this, you can establish security boundaries between users using the same Redis replication group or groups and prevent clients from accessing each other's data. Follow these links to learn more about [Redis Auth](#) and [RBAC](#).

**Q: How do I upgrade to a newer engine version?**

You can easily upgrade to a newer engine version by using the `ModifyCacheCluster` or `ModifyReplicationGroup` APIs and specifying your preferred engine version for the `EngineVersion` parameter. On the Amazon ElastiCache console, you can select a cache cluster or replication group and click "Modify". In the "Modify Cache Cluster" or "Modify Replication Group" window select your preferred engine version from the available options. The engine upgrade process is designed to make a best effort to retain your existing data and requires Redis replication to succeed. For more details on that see [here](#).

**Q: Can I downgrade to an earlier engine version?**

No. Downgrading to an earlier engine version is not supported.

**Q: How do I scale up to a larger node type or out to more nodes?**

In Amazon ElastiCache for Redis you can easily scale up to larger instance types with cluster mode disabled and out to more instances, when using cluster mode enabled.

You can easily scale up to a larger node type by using the `ModifyCacheCluster` or `ModifyReplicationGroup` APIs and specifying your preferred node type for the `CacheNodeType` parameter. On the Amazon ElastiCache console, you can select a cache cluster or replication group and click "Modify". In the "Modify Cache Cluster" or "Modify Replication Group" window select your preferred node type from the available options. The scale up process is designed to make a best effort to retain your existing data and requires Redis replication to succeed. For more details on that see [here](#).

Cluster mode allows you to scale horizontally by adding or removing shards as opposed to vertically scaling a single node. Conceptually, horizontal scaling of the cluster is easy to understand on the server-side — a shard is simply added or removed. Each shard has a primary node and up to five read-only replica nodes. Once the new node is ready, the cluster will need to reallocate or balance the key space across the nodes as configured. With ElastiCache for Redis, the re-balance is automatic.

#### **Q: What is the correct metric to use to measure Redis utilization?**

Amazon ElastiCache provides two metrics to measure CPU utilization for Amazon ElastiCache for Redis workloads – EngineCPUUtilization and CPUUtilization. The CPUUtilization metric measures the CPU utilization for the instance (node), and EngineCPUUtilization metric measures the utilization at the Redis process level. You need the EngineCPUUtilization metric in addition to the CPUUtilization metric as the main Redis process is single threaded and uses just one CPU of the multiple CPU cores available on an instance. Therefore, the CPUUtilization metric does not provide precise visibility into the CPU utilization rates at the Redis process level.

We recommend that you use both the CPUUtilization and EngineCPUUtilization metrics together to get a detailed understanding of CPU Utilization for your Redis clusters. Both the metrics are available in all AWS regions, and you can access these metric using CloudWatch or via the [AWS Management Console](#).

In addition to CPU utilization, Amazon ElastiCache for Redis adds dynamic network processing to enhanced I/O handling in Redis versions 5.0.3 and above. By utilizing the extra CPU power available in nodes with four or more vCPUs, ElastiCache transparently delivers up to 83% increase in throughput and up to 47% reduction in latency per node. See this blog: [Boosting application performance & reducing costs with Amazon ElastiCache for Redis](#)

#### **Q: Can I have cross-region replicas with Amazon ElastiCache for Redis?**

Yes, you can create cross region replicas using the [Global Datastore](#) feature in Amazon ElastiCache for Redis. Global Datastore provides fully managed, fast, reliable and secure cross-region replication. It allows you to write to your Amazon ElastiCache for Redis cluster in one region and have the data available to be read from up to two other cross-region replica clusters, thereby enabling low-latency reads and disaster recovery across regions.

---

## **Read Replica**

#### **Q: What does it mean to run a Redis node as a Read Replica?**

Read Replicas serve two purposes in Redis:

- Failure Handing
- Read Scaling

When you run a cache node with a read replica, the “primary” serves both writes and reads. The read replica acts as a “standby” which is “promoted” in failover scenarios. After failover, the standby becomes the primary and accepts your cache operations. Read replicas also make it [easy](#) to elastically scale out beyond the capacity constraints of a single cache node for read-heavy cache workloads.

#### **Q: When would I want to consider using a Redis read replica?**

There are a variety of scenarios where deploying one or more read replicas for a given primary node may make sense. Common reasons for deploying a read replica include:

- Scaling beyond the compute or I/O capacity of a single primary node for read-heavy workloads. This excess read traffic can be directed to one or more read replicas.
- Serving read traffic while the primary is unavailable. If your primary node cannot take I/O requests (e.g. due to I/O suspension for backups or scheduled maintenance), you can direct read traffic to your read replicas. For this use case, keep in mind that the data on the read replica may be “stale” since the primary Instance is unavailable. The read replica can also be used to restart a failed primary warmed up.
- Data protection scenarios; in the unlikely event of primary node failure or that the Availability Zone in which your primary node resides becomes unavailable, you can promote a read replica in a different Availability Zone to become the new primary.

#### **Q: How do I deploy a read replica node for a given primary node?**

You can create a read replica in minutes using a CreateReplicationGroup API or a few clicks of the Amazon ElastiCache Management Console. When creating a replication group, you specify the MasterCacheClusterIdentifier. The MasterCacheClusterIdentifier is the cache cluster Identifier of the “primary” cache cluster from which you wish to replicate. You then create the read replica cluster within the replication group by calling the CreateCacheCluster API specifying the ReplicationGroupId and the CacheClusterIdentifier of the primary cluster. As with a standard cache cluster, you can also specify the Availability Zone. When you initiate the creation of a read replica, Amazon ElastiCache takes a snapshot of your primary cache cluster and begins replication. As a result, you will experience a brief I/O suspension on your primary cache cluster as the snapshot occurs. The I/O suspension typically lasts on the order of one minute.

The read replicas are as easy to delete as they are to create; simply use the Amazon ElastiCache Management Console or call the DeleteCacheCluster API (specifying the CacheClusterIdentifier for the read replica you wish to delete).

#### **Q: Can I create both a primary and read replicas at the same time?**

Yes. You can create a new cache cluster along with read replicas in minutes using the CreateReplicationGroup API or using the “Create” wizard at the Amazon ElastiCache Management Console and selecting “Multi-AZ Replication”. When creating the cluster, specify an identifier, the total number of desired shard in a cluster and read replicas per shard, along with the creation parameters such as node type, engine version, etc. You can also specify the Availability Zone for each shard in the cluster.

#### **Q: How do I connect to my read replica(s)?**

You can [connect](#) to a read replica just as you would connect to a primary cache node, using the DescribeCacheClusters API or AWS Management Console to retrieve the endpoint(s) for your read replica(s). If you have multiple read replicas, it is up to your application to determine how read traffic will be distributed amongst them. Here are more details:

- Redis (cluster mode disabled) clusters, use the individual Node Endpoints for read operations (In the API/CLI these are referred to as Read Endpoints).
- Redis (cluster mode enabled) clusters, use the cluster's Configuration Endpoint for all operations. You must use a client that supports Redis Cluster (Redis 3.2). You can still read from individual node endpoints (In the

API/CLI these are referred to as Read Endpoints).

**Q: How many read replicas can I create for a given primary node?**

At this time, Amazon ElastiCache allows you to create up to five (5) read replicas for a given primary cache node.

**Q: What happens to read replicas if failover occurs?**

In the event of a failover, any associated and available read replicas should automatically resume replication once failover has completed (acquiring updates from the newly promoted read replica).

**Q: Can I create a read replica of another read replica?**

Creating a read replica of another read replica is not supported.

**Q: Can I promote my read replica into a “standalone” primary node?**

No, this is not supported. Instead, you may snapshot your Amazon ElastiCache for Redis node (you may select the primary or any of the read-replicas). You can then use the snapshot to seed a new Amazon ElastiCache for Redis primary.

**Q: Will my read replica be kept up-to-date with its primary node?**

Updates to a primary cache node will automatically be replicated to any associated read replicas. However, with Redis's asynchronous replication technology, a read replica can fall behind its primary cache node for a variety of reasons. Typical reasons include:

- Write I/O volume to the primary cache node exceeds the rate at which changes can be applied to the read replica
- Network partitions or latency between the primary cache node and a read replica

Read replicas are subject to the strengths and weaknesses of Redis replication. If you are using read replicas, you should be aware of the potential for lag between a read replica and its primary cache node, or “inconsistency.” Amazon ElastiCache [emits a metric](#) to help you understand the inconsistency.

**Q: How do I gain visibility into active read replica(s)?**

You can use the standard `DescribeCacheClusters` API to return a list of all the cache clusters you have deployed (including read replicas), or simply click on the "Cache Clusters" tab of the Amazon ElastiCache Management Console.

Amazon ElastiCache monitors the replication status of your read replicas and updates the Replication State field to Error if replication stops for any reason. You can review the details of the associated error thrown by the Redis engine by viewing the Replication Error field and take an appropriate action to recover from it. You can learn more about troubleshooting replication issues in the "Troubleshooting a Read Replica problem" section of the Amazon ElastiCache User Guide. If a replication error is fixed, the Replication State changes to Replicating.

Amazon ElastiCache allows you to gain visibility into how far a read replica has fallen behind its primary through the Amazon CloudWatch metric ("Replica Lag") available via the AWS Management Console or Amazon CloudWatch APIs.

**Q: My read replica has fallen significantly behind its primary node. What should I do?**

As discussed in the previous questions, "inconsistency" or lag between a read replica and its primary cache node is common with Redis asynchronous replication. If an existing read replica has fallen too far behind to meet your requirements, you can reboot it. Keep in mind that replica lag may naturally grow and shrink over time, depending on your primary cache node's steady-state usage pattern.

**Q: How do I delete a read replica? Will it be deleted automatically if its primary node is deleted?**

You can easily delete a read replica with a few clicks of the AWS Management Console or by using DeleteCacheCluster, or DecreaseReplicaCount APIs. If you want to delete the read replica in addition to the primary cache node, you must use the DeleteReplicationGroup API or AWS Management Consol e.

**Q: How much do read replicas cost? When does billing begin and end?**

A read replica is billed as a standard cache node and at the same rates. Just like a standard cache node, the rate per "Cache Node hour" for a read replica is determined by the cache node class of the read replica – please see Amazon ElastiCache detail page for up-to-date pricing. You are not charged for the data transfer incurred in replicating data between your primary cache node and read replica. Billing for a read replica begins as soon as the read replica has been successfully created (i.e. when status is listed as "active"). The read replica will continue being billed at standard Amazon ElastiCache cache node hour rates until you issue a command to delete it.

**Q: What happens during failover and how long does it take?**

Initiated failover is supported by Amazon ElastiCache so that you can resume cache operations as quickly as possible. When failing over, Amazon ElastiCache simply flips the DNS record for your cache node to point at the read replica, which is in turn promoted to become the new primary. We encourage you to follow best practices and implement cache node connection retry at the application layer. Typically, start-to-finish, steps 1 to 5 below, complete within 6 minutes.

These are the automatic failover events, listed in the order of occurrence :

1. Replication group message: Test Failover API called for node group <node-group-id>
2. Cache cluster message: Failover from primary node <primary-node-id> to replica node <node-id> completed
3. Replication group message: Failover from primary node <primary-node-id> to replica node <node-id> completed
4. Cache cluster message: Recovering cache nodes <node-id>
5. Cache cluster message: Finished recovery for cache nodes <node-id>

**Q: Can I create a read replica in another region as my primary?**

No. Your read replica may only be provisioned in the same or different Availability Zone of the same Region as your cache node primary.

You can, however, [use](#) the Global Datastore for Redis to work with fully managed, fast, reliable, and secure replication across AWS Regions. Using this feature, you can create cross-Region read replica clusters for ElastiCache for Redis to enable low-latency reads and disaster recovery across AWS Regions.

**Q: Can I see which Availability Zone my primary is currently located in?**

Yes, you can gain visibility into the location of the current primary by using the AWS Management Console or `DescribeCacheClusters` API.

After failover, my primary is now located in a different Availability Zone than my other AWS resources (e.g. EC2 instances).

**Q: Should I be concerned about latency?**

Availability Zones are engineered to provide low latency network connectivity to other Availability Zones in the same Region. In addition, you may want to consider architecting your application and other AWS resources with redundancy across multiple Availability Zones so your application will be resilient in the event of an Availability Zone failure.

**Q: Can I add and remove read replica nodes for my Redis Cluster environment?**

Yes. You can add or remove replica across one or more shards in a Redis Cluster environment. The cluster continues to stay online and serve incoming I/O during this operation.

---

## Multi-AZ

**Q: What is Multi-AZ for an Amazon ElastiCache for Redis replication group?**

An Amazon ElastiCache for Redis replication group consists of a primary and up to five read replicas. If Multi-AZ is enabled then at least one replica is required per primary. See [AutoFailover](#). Redis asynchronously replicates the data from the primary to the read replicas. During certain types of planned maintenance, or in the unlikely event of Amazon ElastiCache node failure or Availability Zone failure, Amazon ElastiCache will automatically detect the failure of a primary, select a read replica, and promote it to become the new primary. Amazon ElastiCache also propagates the DNS changes of the promoted read replica, so if your application is writing to the primary node endpoint, no endpoint change will be needed.

**Q: What are the benefits of using Multi-AZ and when should I use it?**

The main benefits of running your Amazon ElastiCache for Redis in Multi-AZ mode are enhanced availability and smaller need for administration. If an Amazon ElastiCache for Redis primary node failure occurs, the impact on your ability to read/write to the primary is limited to the time it takes for automatic failover to complete. When Multi-AZ is enabled, Amazon ElastiCache node failover is automatic and requires no administration. You no

longer need to monitor your Redis nodes and manually initiate a recovery in the event of a primary node disruption.

**Q: How does Multi-AZ work?**

You can use Multi-AZ if you are using Amazon ElastiCache for Redis and have a replication group consisting of a primary node and one or more read replicas. If the primary node fails, Amazon ElastiCache will automatically detect the failure, select one from the available read replicas, and promote it to become the new primary. Amazon ElastiCache will propagate the DNS changes of the promoted replica so that your application can keep writing to the primary endpoint. Amazon ElastiCache will also spin up a new node to replace the promoted read replica in the same Availability Zone of the failed primary. In case the primary failed due to temporary Availability Zone disruption, the new replica will be launched once that Availability Zone has recovered.

**Q: Can I have replicas in the same Availability Zone as the primary?**

Yes. Note that placing both the primary and the replica(s) in the same Availability Zone will not make your Amazon ElastiCache for Redis replication group resilient to an Availability Zone disruption. Additionally this will not be allowed if Multi-AZ is turned on.

**Q: What events would cause Amazon ElastiCache to fail over to a read replica?**

Amazon ElastiCache will failover to a read replica in the event of any of the following:

- Loss of availability in primary's Availability Zone
- Loss of network connectivity to primary
- Compute unit failure on primary

**Q: When should I use Multi-AZ?**

Using Redis replication in conjunction with Multi-AZ provides increased availability and fault tolerance. Such deployments are a natural fit for use in production environments.

**Q: How do I create an Amazon ElastiCache for Redis replication group with Multi-AZ enabled?**

You can create an Amazon ElastiCache for Redis primary and read replicas by clicking Launch Cache Cluster on the Amazon ElastiCache Management Console. You can also do so by calling the [CreateReplicationGroup API](#). For existing replication groups (Redis 2.8.24, 2.8.23, 2.8.22, 2.8.21, 2.8.19, and 2.8.6), you can enable Multi-AZ by choosing a replication group and clicking Modify on the Amazon ElastiCache Management Console or by using the [ModifyReplicationGroup API](#). Switching a replication group to Multi-AZ is not disruptive to your Redis data and does not interfere your nodes' ability to serve requests.

**Q: Which read replica will be promoted in case of a primary node failure?**

If there are more than one read replicas, the read replica with the smallest asynchronous replication lag to the primary will be promoted.

**Q: How much does it cost to use Multi-AZ?**

Multi-AZ is free of charge. You only pay for the Amazon ElastiCache nodes that you use.

**Q: What are the performance implications of Multi-AZ?**

Amazon ElastiCache currently uses the Redis engine's native, asynchronous replication and is subject to its strengths and limitations. In particular, when a read replica connects to a primary for the first time, or if the primary changes, the read replica does a full synchronization of the data from the primary, imposing load on itself and the primary. For additional details regarding Redis replication please see [here](#).

**Q: What node types support Multi-AZ?**

All available cache node types in Amazon ElastiCache support Multi-AZ except T1 node family.

**Q: Will I be alerted when automatic fail over occurs?**

Yes, Amazon ElastiCache will create an event to inform you that automatic failover occurred. You can use the [DescribeEvents API](#) to return information about events related to your Amazon ElastiCache node, or click the Events section of the Amazon ElastiCache Management Console.

**Q: After failover, my primary is now located in a different Availability Zone than my other AWS resources (for example, EC2 instances). Should I be concerned about latency?**

Availability Zones are engineered to provide low latency network connectivity to other Availability Zones in the same region. You may consider architecting your application and other AWS resources with redundancy across multiple Availability Zones so your application will be resilient in the event of an Availability Zone disruption.

**Q: Where can I get more information about Multi-AZ?**

For more information about Multi-AZ, see Amazon ElastiCache [documentation](#).

---

## Backup and Restore

**Q: What is Backup and Restore?**

Backup and Restore is a feature that allows customers to create snapshots of their Amazon ElastiCache for Redis clusters. Amazon ElastiCache stores the snapshots, allowing users to subsequently use them to restore Redis clusters.

**Q: What is a snapshot?**

A snapshot is a copy of your entire Redis cluster at a specific moment.

**Q: Why would I need snapshots?**

Creating snapshots can be useful in case of data loss caused by node failure, as well as the unlikely event of a hardware failure. Another common reason to use backups is for archiving purposes. Snapshots are stored in Amazon S3, which is a durable storage, meaning that even a power failure won't erase your data.

**Q: What can I do with a snapshot?**

You can use snapshots to warm start an Amazon ElastiCache for Redis cluster with preloaded data.

**Q: How does Backup and Restore work?**

When a backup is initiated, Amazon ElastiCache will take a snapshot of a specified Redis cluster that can later be used for recovery or archiving. You can initiate a backup anytime you choose or set a recurring daily backup with retention period of up to 35 days.

When you choose a snapshot to restore, a new Amazon Amazon for Redis cluster will be created and populated with the snapshot's data. This way you can create multiple ElastiCache for Redis clusters from a specified snapshot.

**Q: Where are my snapshots stored?**

The snapshots are stored in S3.

**Q: How can I get started using Backup and Restore?**

You can select to use the Backup and Restore feature through the [AWS Management Console](#), through the Amazon ElastiCache APIs (CreateCacheCluster, ModifyCacheCluster, CreateReplicationGroup, and ModifyReplicationGroup API's) and CLI. You can deactivate and reactivate the feature anytime you choose.

**Q: How do I specify which Redis cluster and node to backup?**

Backup and Restore creates snapshots on a cluster basis. Users can specify which ElastiCache for Redis cluster to backup through the [AWS Management Console](#), CLI or through the CreateSnapshot API. In a Replication Group, you can choose to backup the primary or any of the read-replica clusters. We recommend users enable backup on one of the read-replicas, mitigating any latency effect on the Redis primary.

**Q: How can I specify when a backup will take place?**

Through the [AWS Management Console](#), CLI or APIs you can specify when to start a single backup or a recurring backup. Users are able to:

- Take a snapshot right now (through "Backup" console button in the "Redis" tab, or CreateSnapshot API)
- Set up an automatic daily backup. The backup will take place during your preferred backup window. You can set that up through Creating/Modifying cluster via console or the CreateCacheCluster, ModifyCacheCluster or ModifyReplicationGroup API's.

**Q: What is a backup window and why do I need it?**

The preferred backup window is the user-defined period of time during which your Amazon ElastiCache for Redis cluster backup will start. This is helpful if you want to backup at a certain time of day or to refrain from backups during a particularly high-utilization period.

**Q: What is the performance impact of taking a snapshot?**

While taking a snapshot, you may encounter increased latencies for a brief period at the node. Snapshots use Redis's built-in BGSAVE and are subject to its strengths and limitations. In particular, the Redis process forks and the parent continues to serve requests while the child saves the data on disk and then exits. The forking increases the memory usage for the duration of the snapshot generation. When this memory usage exceeds that of the available memory of the cache node, swapping can get triggered, further slowing down the node. For this reason, we recommend generating snapshots on one of the read replicas (instead of the primary). Also, we suggest setting the reserved-memory parameter to minimize swap usage. See [here](#) for more details.

**Q: Can I create a snapshot from an Amazon ElastiCache for Redis read replica?**

Yes. In a Replication Group with cluster mode disabled, creating a snapshot from a read replica is the best way to back up your data while minimizing performance impact. In a Replication Group with cluster mode enabled, you

cannot choose to back up the primary or any of the read-replica clusters.

**Q: In what regions is the Backup and Restore feature available?**

Backup and Restore feature is available in all regions where ElastiCache service is available.

**Q: Can I export Amazon ElastiCache for Redis snapshots to an S3 bucket owned by me?**

Yes. You can export your Amazon ElastiCache for Redis snapshots to an authorized S3 bucket in the same region as your cluster. For more details on exporting snapshots and setting the required permissions, please refer to [this](#).

**Q: Can I copy snapshots from one region to another?**

Yes. You must first copy your snapshot into an authorized S3 bucket of your choice in the same region and then use the S3 PUT object- Copy API to copy it to a bucket in another region. For more details on copying S3 objects, please see [this](#).

**Q: I have multiple AWS accounts using Amazon ElastiCache for Redis. Can I use Amazon ElastiCache snapshots from one account to warm start an Amazon ElastiCache for Redis cluster in a different one?**

Yes. You must first copy your snapshot into an authorized S3 bucket of your choice in the same region and then grant cross-account bucket permissions to the other account. For more details on S3 cross-account permissions, please see [this](#). Finally, specify the S3 location of your RDB file during cluster creation through the Launch Cache Cluster Wizard in the console or through the CreateCacheCluster API.

**Q: How much does it cost to use Backup and Restore?**

Amazon ElastiCache provides storage space for one snapshot free of charge for each active Amazon ElastiCache for Redis cluster. Additional storage will be charged based on the space used by the snapshots with \$0.085/GB every month (same price in all regions). Data transfer for using the snapshots is free of charge.

**Q: What is the retention period?**

Retention period is the time span during which the automatic snapshots are retained. For example, if a retention period is set for 5, a snapshot that was taken today will be retained for 5 days before being deleted. You can choose to copy one or more automatic snapshots to store them as manual so that they won't be deleted after the retention period is over.

**Q: How do I manage the retention of my automated snapshots?**

You can use the [AWS Management Console](#) or ModifyCreateCluster or ModifyReplicationGroup API to manage the period of time your automated backups are retained by modifying the RetentionPeriod parameter. If you desire to turn off automated backups altogether, you can do so by setting the retention period to 0 (not recommended).

**Q: What happens to my snapshots if I delete my Amazon ElastiCache for Redis cluster?**

When you delete an Amazon ElastiCache for Redis cluster, your manual snapshots are retained. You will also have an option to create a final snapshot before the cluster is deleted. Automatic cache snapshots are not retained.

**Q: What cache nodes types support backup and restore capability?**

All Amazon ElastiCache for Redis instance node types besides t1.micro support backup and restore:

Current Generation Cache Nodes:

- cache.m4.large
- cache.m4.xlarge
- cache.m4.2xlarge
- cache.m4.4xlarge
- cache.m4.10xlarge
- cache.m5.large
- cache.m5.xlarge
- cache.m5.2xlarge
- cache.m5.4xlarge
- cache.m5.12xlarge
- cache.m5.24xlarge
- cache.m6g.large
- cache.m6g.xlarge
- cache.m6g.2xlarge
- cache.m6g.4xlarge
- cache.m6g.8xlarge
- cache.m6g.12xlarge
- cache.m6g.16xlarge
- cache.r4.large
- cache.r4.xlarge
- cache.r4.2xlarge
- cache.r4.4xlarge

- cache.r4.8xlarge
- cache.r4.16xlarge
- cache.r5.large
- cache.r5.xlarge
- cache.r5.2xlarge
- cache.r5.4xlarge
- cache.r5.12xlarge
- cache.r5.24xlarge
- cache.r6g.large
- cache.r6g.xlarge
- cache.r6g.2xlarge
- cache.r6g.4xlarge
- cache.r6g.8xlarge
- cache.r6g.12xlarge
- cache.r6g.16xlarge
- cache.t2.medium
- cache.t2.small
- cache.t2.micro
- cache.t3.medium
- cache.t3.small

#### **Previous Generation Nodes:**

- cache.m1.small
- cache.m1.medium
- cache.m1.large
- cache.m1.xlarge
- cache.m2.xlarge
- cache.m2.2xlarge
- cache.m2.4xlarge

- cache.m3.medium
- cache.m3.large
- cache.m3.xlarge
- cache.m3.2xlarge
- cache.r3.large
- cache.r3.xlarge
- cache.r3.2xlarge
- cache.r3.4xlarge
- cache.r3.8xlarge
- cache.c1.xlarge

**Q: Can I use my own RDB snapshots stored in S3 to warm start an Amazon ElastiCache for Redis cluster?**

Yes. You can specify the S3 location of your RDB file during cluster creation through the “Create Cluster” Wizard in the console or through the CreateCacheCluster or CreateReplicationGroup API.

**Q: Can I use the Backup and Restore feature if I am running Amazon ElastiCache in a VPC?**

Yes.

---

## Redis Cluster

**Q: What is ElastiCache for Redis Cluster?**

ElastiCache for Redis Cluster allows customers to create and run managed Redis Clusters with multiple shards. It is compatible with open source Redis 3.2.4 onwards and comes with a number of enhancements for a more stable and robust experience (see the “enhanced engine” section below for additional details on these enhancements).

**Q: Why would I need a scale out Redis environment?**

There are three main scenarios for running a scale out Redis environment. First, if the total memory size of your Redis data exceeds or is projected to exceed the memory capacity of a single VM. Second, if the write throughput of your application to Redis exceeds the capacity of a single VM. Third, if you would like to spread the data across multiple shards so that any potential issue that comes up with a single node will have a smaller impact on the overall Redis environment.

**Q: Why would I run my Redis Cluster workload on Amazon ElastiCache?**

Amazon ElastiCache provides a fully managed distributed in-memory Redis environment, from provisioning server resources to installing the engine software and applying any configuration parameters you choose. It uses enhancements to the Redis engine developed by Amazon, which results in a more robust and stable experience

(see “enhanced engine” section for more details). Once your Redis environment is up and running, the service automates common administrative tasks such as failure detection and recovery, backups and software patching. It also provides a robust Multi-AZ solution with automatic failover. In case of a failure of one or more primary nodes in your cluster, Amazon ElastiCache will automatically detect the failure and respond by promoting the most up to date replica to primary. This process is automated and does not mandate any manual work on your behalf. Amazon ElastiCache also provides detailed monitoring metrics associated with your ElastiCache nodes, enabling you to diagnose and respond to issues very quickly.

**Q: Is ElastiCache for Redis Cluster compatible with open source Redis?**

Yes, Amazon ElastiCache for Redis Cluster is compatible with open source Redis 3.2.4 onwards. You can use the open source Redis Cluster clients to access scale-out clusters on ElastiCache for Redis.

**Q: What is the upgrade path from current ElastiCache for Redis 2.8.x to ElastiCache for Redis Cluster (version 3.2.4)?**

If you are using Redis 3.2 with cluster\_mode parameter disabled, you can simply choose the node or cluster you wish to upgrade and modify the engine version. ElastiCache will provision a Redis 3.2.4 cluster and migrate your data to it, while maintaining the endpoint.

If you are using Redis 3.2 with cluster\_mode enabled, you can migrate to Redis Cluster by first creating a snapshot of your data using the backup and restore feature. Then, select the created snapshot and click on “Restore Snapshot” to create a Redis 3.2 cluster using the snapshotted data. Finally, update the new endpoint in your client. Note that to use Redis 3.2 in cluster mode you would need to switch to a Redis Cluster client.

**Q: Is the pricing for clustered configuration different from non-clustered configuration?**

No. Amazon ElastiCache for Redis provides the flexibility of clustered and non-clustered configuration at the same price. Customers can now enjoy enhanced engine functionality within Amazon ElastiCache for Redis and use full feature support for clustered configuration and scalability at the same price.

**Q: What is Multi-AZ for ElastiCache for Redis Cluster?**

Each shard of an ElastiCache for Redis cluster consists of a primary and up to five read replicas. Redis asynchronously replicates the data from the primary to the read replicas. During certain types of planned maintenance, or in the unlikely event of ElastiCache node failure or Availability Zone failure, Amazon ElastiCache will automatically detect the failure of a primary, select a read-replica, and promote it to become the new primary.

ElastiCache for Redis Cluster provides enhancements and management for Redis 3.x environments. When running an unmanaged Redis environment, in a case of primary node failure, the cluster relies on a majority of primaries to determine and execute a failover. If such majority doesn’t exist, the cluster will go into failed state, rejecting any further reads and writes. This could lead to major availability impact on the application, as well as requiring human intervention to manually salvage the cluster. ElastiCache for Redis Multi-AZ capability is built to handle any failover case for Redis Cluster with robustness and efficiency.

**Q: How is Multi-AZ in ElastiCache for Redis Cluster different than in ElastiCache for Redis versions 2.8.x?**

Redis 3.x works with intelligent clients that store a node map with all the cluster nodes’ endpoints. During a failover, the client updates the node map with the IP endpoint for the new primary. This provides up to 4x faster failover time than with ElastiCache for Redis 2.8.x.

**Q: How does Multi-AZ work for Redis Cluster?**

You can use Multi-AZ if you are using an ElastiCache for Redis Cluster with each shard having 1 or more read-replicas. If a primary node of a shard fails, ElastiCache will automatically detect the failure, select one of the available read-replicas, and promote it to become the new primary. The Redis 3.x client will update the promoted replica as primary, no application change is required. ElastiCache will also spin up a new node to replace the promoted read-replica in the same Availability Zone of the failed primary. In case the primary failed due to a temporary Availability Zone failure, the new replica will be launched once that Availability Zone has recovered.

**Q: What is a backup in ElastiCache for Redis Cluster?**

An ElastiCache for Redis Cluster backup is a series of snapshots of the cluster's shards, stored together to keep a copy of your entire Redis data around a certain time frame.

**Q: How is a backup in ElastiCache for Redis Cluster different from a snapshot in ElastiCache for Redis?**

Since a non-clustered ElastiCache for Redis environment has a single primary node, a backup is a single file which contains a copy of the Redis data. ElastiCache for Redis Cluster can have one or more shards, thus a backup might contain multiple files.

**Q: How do I specify which ElastiCache for Redis nodes to backup in each shard?**

You cannot manually specify a node to backup within each shard. When initiating a backup, ElastiCache will automatically select the most up-to-date read replica in each shard and take a snapshot of its data.

**Q: How does ElastiCache for Redis Cluster Backup and Restore work?**

When a backup is initiated, ElastiCache will take a backup of a specified cluster; that backup can later be used for recovery or archiving. The backup will include a copy of each of the cluster's shards, thus a full backup contains a series of files. You can initiate a backup anytime you choose or set a recurring daily backup with retention period of up to 35 days.

When you choose a backup to restore, a new ElastiCache for Redis cluster will be created and populated with the backup's data. Also, you can use this feature for an easy migration path to a managed Redis Cluster experience on ElastiCache. If you are running self-managed Redis on EC2, you can take RDB snapshots of your existing workloads (both Redis Cluster and single-shard Redis) and store them in S3. Then simply provide them as input for creating a sharded Redis Cluster on ElastiCache, and the desired number of shards. ElastiCache will do the rest.

Currently, ElastiCache uses Redis' native mechanism to create and store an RDB file for each shard as the backup.

**Q: Is the backup in ElastiCache for Redis Cluster a point-in-time snapshot?**

When you initiate a backup, ElastiCache will trigger backups of all of the shards of your cluster at the same time. In rare cases there might be a need to retake a snapshot of one or more nodes that did not complete successfully the first time. ElastiCache does that automatically and no user intervention is required. But in such a case, while each individual snapshot is a point-in-time representation of the node it was taken from, not all the cluster's snapshots would be taken at the same time.

**Q: How can I specify when a backup will take place?**

Through the AWS Management Console, CLI or APIs you can specify when to start a single backup or a recurring backup. Users are able to:

- Take a backup right now (through "Create Snapshot" console button or CreateSnapshot API)

- Set up an automatic daily backup. The backup will take place during your preferred backup window. You can set that up through Creating/Modifying cluster via console or the CreateReplicationGroup and ModifyReplicationGroup API's.

**Q: Can I use my own RDB snapshots stored in S3 to pre-seed a scale out ElastiCache for Redis Cluster environment?**

Yes. You can specify the S3 location of your RDB files during cluster creation through the Create Cluster Wizard in the console or through the CreateReplicationGroup API. ElastiCache will automatically parse the Redis key-space of the RDB snapshot and redistribute it among the shards of the new cluster.

---

## Enhanced Engine

**Q: How is the engine within ElastiCache for Redis different from open-source Redis?**

The engine within ElastiCache for Redis is fully compatible with open source Redis but also comes with enhancements that improve robustness and stability. Some of the enhancements are:

- More usable memory: You can now safely allocate more memory for your application without risking increased swap usage during syncs and snapshots.
- Improved synchronization: More robust synchronization under heavy load and when recovering from network disconnections. Additionally, syncs are faster as both the primary and replicas no longer use the disk for this operation.
- Smoother failovers: In the event of a failover, your shard now recovers faster as replicas no longer flush their data to do a full re-sync with the primary.

**Q: How do I use the enhanced engine?**

To use the enhanced engine from the Amazon ElastiCache management console, just select an engine compatible with Redis engine version 2.8.22 or higher when creating a cluster. From that point on you will be using the enhanced engine. You can also use the enhanced engine through the ElastiCache API or AWS CLI by specifying the engine version when running the CreateCacheCluster API.

**Q: Do I need to change my application code to use the enhanced engine on ElastiCache?**

No. The enhanced engine is fully compatible with open-source Redis, thus you can enjoy its improved robustness and stability without the need to make any changes to your application code.

**Q: How much does it cost to use the enhanced engine?**

There is no additional charge for using the enhanced engine. As always, you will only be charged for the nodes you use.

---

## Online Cluster Resizing

**Q: What is Online Cluster Resizing?**

Amazon ElastiCache for Redis provides the ability to add and remove shards from running cluster-mode enabled Redis Cluster. You can dynamically scale-out or scale-in your Redis cluster workloads to adapt to changes in demand. Amazon ElastiCache will resize the cluster by adding or removing shards and redistributing hash slots uniformly across the new shard configuration, all while the cluster continues to stay online and serve requests.

**Q: What are the benefits of using Online Cluster Resizing?**

The ability to dynamically scale-out and scale-in a cluster can help you manage application variability and meet oscillating demands. You can right-size your clusters by adding or removing shards to scale performance and in-memory capacity. This feature eliminates the need to overprovision clusters based on peak demand, helps improve efficiency, and reduces cost.

**Q: How can I use Online Cluster Resizing?**

Online Cluster Resizing is available for cluster-mode enabled Redis Clusters on version 3.2.10 or higher. To reshards your cluster, select the cluster and specify whether you want to add or remove shards. When you resize the cluster to scale-out, Amazon ElastiCache adds shards and migrates slots from existing shards to new shards, in a way such that the slots are uniformly distributed (by count) across shards. Similarly, when resizing the cluster to scale-in, Amazon ElastiCache migrates slots to the remaining shards to uniformly distribute the slots and deletes specified shards.

**Q: How long does the Online Cluster Resizing take?**

The time taken to resize a cluster depends on multiple factors, such as number of slots that need to be migrated across shards, size of data and incoming request rate on the cluster. The workflow is optimized to parallelize slot migration for faster scale out.

**Q: Can the cluster be used while cluster resizing is in progress?**

Yes, the cluster continues to stay online and serve incoming requests, while resharding is in progress. However, snapshotting a cluster is not supported when resharding is in progress.

**Q: Is there any performance impact of this operation on the cluster?**

While Online Cluster Resizing provides the benefits to scale out/in with zero downtime, it is a compute-intensive operation and can increase the latency of your client connection. To reduce the load on the cluster during the operation, we recommend that you follow the [best practices](#) (described in the documentation).

**Q: How can I track the progress of an online resharding operation?**

You can track the progress of resharding by viewing the status of the cluster, shards and nodes. During the operation, the cluster, shards and nodes will stay in "modifying" status. Similarly, when shards are being created, deleted or participating in slot migration, the individual shard status will reflect these statuses to show progress. Additionally, the status of end-to-end operation can also be tracked using the progress indicator for the resharding operation, which indicates percentage completed and provides insight into the remaining time for the operation. Lastly, event messages indicate the progress by describing actions being taken (shard creation, slot migration, etc.) during this operation.

**Q: What is the rebalance operation for Amazon ElastiCache for Redis cluster?**

The rebalance operation can be used to redistribute slots amongst existing shards to achieve a uniform distribution. This is useful when a cluster is created with manually specified uneven slot distribution or a scale-out/in operation leaves the cluster with uneven distribution. Assuming the slots are identical in their memory and I/O requirements, uniform slot distribution by count is an easy way to load balance across shards.

## **Q: How does tagging work when a cluster scales-out?**

When new nodes are added to scale-out a cluster, the nodes carry the same set of tags that are common across all existing nodes. Additionally, users can modify tags on all nodes and continue to use tagging as before.

## **Q: Are there any client or application side changes needed to use online cluster resizing?**

No. The enhanced slot distribution used in cluster resizing is compliant with Redis cluster client behavior and does not require any application changes. Amazon ElastiCache retains cluster endpoints, enabling you to continue using existing clients without any changes.

---

## **Encryption**

### **Q: What does encryption at-rest for Amazon ElastiCache ElastiCache for Redis provide?**

Encryption-at-rest provides mechanisms to guard against unauthorized access of your data on the server. When enabled on a replication group, it encrypts the following aspects:

- Disk during sync, backup and swap operations
- Backups stored in Amazon S3

Amazon ElastiCache for Redis offers default (service managed) encryption at rest, as well as ability to use your own symmetric customer managed customer primary keys in [AWS Key Management Service \(KMS\)](#). At-rest encryption can be enabled on a replication group only when it is created. You can read more [here](#).

### **Q: What does encryption in-transit for Amazon ElastiCache for Redis provide?**

The encryption in-transit feature enables you to encrypt all communications between clients and Redis server as well as between the Redis servers (primary and read replica nodes). It is an optional feature and can only be enabled on Redis replication groups when they are created. You can read more [here](#).

### **Q: How can I use encryption in-transit, at-rest, and Redis AUTH?**

Encryption in-transit, encryption at-rest, and Redis AUTH and Managed Role-Based Access Control (RBAC) are all opt-in features. At the time of Redis cluster creation via the console or command line interface, you can specify if you want to enable at-rest and/or in-transit encryption. If you enabled in-transit encryption you can choose to use Redis AUTH or RBAC for added security and access control. Once the cluster is setup with encryption enabled, Amazon ElastiCache seamlessly manages certificate expiration and renewal without requiring any additional action from the application. Additionally, the Redis clients need to support TLS to avail of the encrypted in-transit traffic. If you choose to use Redis AUTH you will need to have Redis 3.2.6 onward, while RBAC requires that you use Redis 6.

### **Q: Is there an Amazon ElastiCache for Redis client that I need to use when using encryption in-transit, or at-rest?**

No. Encryption in-transit requires clients to support TLS. Most of the popular Redis clients (such as Lettuce, Predis, go-Redis) provide support for TLS with some configuration settings. You have to make sure that your Redis client of choice is configured to support TLS and continue to use Amazon ElastiCache for Redis as before.

### **Q: Can I enable encryption in-transit and encryption at-rest on my existing Amazon ElastiCache for Redis clusters?**

No. Encryption in-transit and encryption at-rest support is only available for new clusters and is not supported on existing Amazon ElastiCache for Redis clusters. Amazon ElastiCache for Redis versions 6.0.5, 5.0.0, 4.0.10, and 3.2.6 support these features.

**Q: Is there any action needed to renew certificates?**

No. Amazon ElastiCache manages certification expiration and renewal behind the scene. No user action is necessary for ongoing certificate maintenance.

**Q: Can I use my certificates for encryption?**

No. Currently, Amazon ElastiCache does not provide the ability for you to use your certificates. Amazon ElastiCache manages certificates transparently for you.

**Q: Which instance types are supported for encryption in transit and encryption at rest?**

All current generation instances are supported for encryption in transit and encryption at rest. For a full list of in-transit encryption conditions see [here](#) and at-rest encryption conditions see [here](#).

**Q: Are there additional costs for using encryption?**

There are no additional costs for using encryption.

---

## Compliance

**Q: Which compliance programs does Amazon ElastiCache for Redis support?**

Amazon ElastiCache for Redis supports compliance programs such as SOC 1, SOC 2, SOC 3, ISO, MTCS, C5, PCI, HIPAA, and FedRAMP. See [AWS Services in Scope by Compliance Program](#) for current list of supported compliance programs.

**Q: Is Amazon ElastiCache for Redis PCI compliant?**

Yes, the AWS PCI compliance program includes Amazon ElastiCache for Redis as a PCI compliant Service. To learn more, see the following resources:

- [Amazon ElastiCache for Redis Compliance page](#)
- [AWS PCI Compliance page](#)

To see the current list of compliance programs that Amazon ElastiCache for Redis is in scope for, see [AWS Services in Scope by Compliance Program](#).

**Q: Is Amazon ElastiCache for Redis HIPAA eligible?**

Yes, Amazon ElastiCache for Redis is a [HIPAA Eligible Service](#) and has been added to the [AWS Business Associate Addendum \(BAA\)](#). This means you can use Amazon ElastiCache for Redis to help you process, maintain, and store protected health information (PHI) and power healthcare applications.

**Q: What do I have to do to use HIPAA eligible Amazon ElastiCache for Redis?**

If you have an executed [Business Associate Agreement \(BAA\)](#) with AWS, you can use ElastiCache for Redis to build HIPAA-compliant applications. If you do not have a BAA or have other questions about using AWS for your HIPAA-compliant applications, [contact us](#) for more information. See [Architecting for HIPAA Security and](#)

[Compliance on Amazon Web Services](#) for information about how to configure Amazon HIPAA Eligible Services to store, process, and transmit PHI.

**Q: Is Amazon ElastiCache for Redis FedRAMP authorized?**

The AWS FedRAMP compliance program includes Amazon ElastiCache for Redis as a FedRAMP authorized service. United States government customers and their partners can now use the latest version of Amazon ElastiCache for Redis to process and store their FedRAMP systems, data, and mission-critical, high-impact workloads in the AWS GovCloud (US) Region, and at moderate impact level in AWS US East/West Regions.

To learn more, see the following resources:

- [Amazon ElastiCache for Redis Compliance](#) page
- [AWS FedRAMP Compliance](#) page

To see the current list of compliance programs that Amazon ElastiCache for Redis is in scope for, see [AWS Services in Scope by Compliance Program](#).

**Q: Does it cost extra to use compliance features?**

No, there is no additional cost for using compliance features.

---

## Global Datastore

**Q: What is Global Datastore for Redis?**

[Global Datastore](#) is a feature of [Amazon ElastiCache for Redis](#) that provides fully managed, fast, reliable and secure cross-region replication. With Global Datastore, you can write to your Amazon ElastiCache for Redis cluster in one region, and have the data available for read in up to two other cross-region replica clusters, thereby enabling low-latency reads and disaster recovery across regions.

Designed for real-time applications with a global footprint, Global Datastore for Redis supports cross-region replication latency of typically under one second, increasing the responsiveness of your applications by providing geo-local reads closer to the end users. In the unlikely event of regional degradation, one of the healthy cross-region replica clusters can be promoted to become the primary cluster with full read/write capabilities. Once initiated, the promotion typically completes in less than a minute, allowing your applications to remain available.

**Q: How many AWS regions can I replicate to?**

You can replicate to up to two secondary regions within a Global Datastore for Redis. The clusters in secondary regions can be used to serve low-latency local reads and for disaster recovery, in the unlikely event of a regional degradation.

**Q: Which engine versions support Global Datastore for Redis?**

Global Datastore is supported on Amazon ElastiCache for Redis 5.0.6 onward. Customers can [upgrade engine version](#) to 5.0.6 and use Global Datastore.

**Q: How can I create Global Datastore for Redis?**

You can setup a Global Datastore by using an existing cluster or creating a new cluster to be used as a primary. You can create a Global Datastore for Redis with just a few clicks on the Amazon ElastiCache Management

Console or by downloading the latest AWS SDK or CLI. There is support for Global Datastore in AWS CloudFormation.

**Q: Does Amazon ElastiCache automatically failover a Global Datastore for Redis to promote a secondary cluster in the event when primary cluster (region) is degraded?**

No, Amazon ElastiCache doesn't automatically promote a secondary cluster in the event when primary cluster (region) is degraded. You can manually initiate the failover by promoting a secondary cluster to become a primary. The failover and promotion of secondary cluster typically completes in less than one minute.

**Q: How do I perform application failover for disaster recovery if my primary cluster experiences degradation of service?**

In case your primary cluster in a Global Datastore for Redis experiences degradation of service, you can assign a secondary cluster as your new primary cluster, and then remove the old primary cluster from your Global Datastore. Once the secondary cluster is promoted to primary, Amazon ElastiCache will reconfigure the old primary (if reachable) as secondary, and setup replication to synchronize all secondary regions with the new primary. If your entire application stack is replicated to another AWS region, you may failover the entire application stack (including your compute resources) to that AWS region. If the rest of your application stack does not require failover, make sure your application has access to the secondary cluster endpoint.

**Q: How is my data secured when using Global Datastore for Redis?**

Global Datastore for Redis uses encryption in-transit for cross-region traffic to keep your data secure. Additionally, you can also encrypt your primary and secondary clusters using encryption at-rest to keep your end-to-end data secure. Each primary and secondary cluster can have a separate customer managed Customer Master Key (CMK) in AWS Key Management Service (KMS) for encryption at rest.

**Q: What Recovery Point Objective (RPO) and Recovery Time Objective (RTO) can I expect with Global Datastore for Redis?**

Amazon ElastiCache doesn't provide an SLA for RPO and RTO. The RPO varies based on replication lag between regions, and depends on network latency between regions and cross-region network traffic congestion. The RPO of Global Datastore is typically under one second, so the data written in primary region is available in secondary regions within one second. The RTO of Global Datastore for Redis is typically under a minute. Once a failover to a secondary cluster is initiated, Amazon ElastiCache typically promotes the secondary to full read/write capabilities in under a minute.

**Q: What is the pricing for Global Datastore for Redis?**

Amazon ElastiCache does not charge any premium to use Global Datastore for Redis. You pay for the primary and secondary clusters in your Global Datastore, and the [cross-region data transfer traffic](#).

[Sign In to the Console](#)

## Learn About AWS

- [What Is AWS?](#)
- [What Is Cloud Computing?](#)
- [AWS Inclusion, Diversity & Equity](#)
- [What Is DevOps?](#)
- [What Is a Container?](#)
- [What Is a Data Lake?](#)
- [AWS Cloud Security](#)
- [What's New](#)
- [Blogs](#)
- [Press Releases](#)

## Help

- [Contact Us](#)
- [AWS Careers](#)
- [File a Support Ticket](#)
- [Knowledge Center](#)
- [AWS Support Overview](#)
- [Legal](#)

## Resources for AWS

- [Getting Started](#)
- [Training and Certification](#)
- [AWS Solutions Portfolio](#)
- [Architecture Center](#)
- [Product and Technical FAQs](#)
- [Analyst Reports](#)
- [AWS Partner Network](#)

## Developers on AWS

- [Developer Center](#)
- [SDKs & Tools](#)
- [.NET on AWS](#)
- [Python on AWS](#)
- [Java on AWS](#)
- [PHP on AWS](#)
- [Javascript on AWS](#)

[Create an AWS Account](#)



Amazon is an Equal Opportunity Employer: *Minority / Women / Disability / Veteran / Gender Identity / Sexual Orientation / Age.*

#### Language

| عربي

Bahasa Indonesia |

Deutsch |

English |

Español |

Français |

Italiano |

Português |

Tiếng Việt |

Türkçe |

Русский |

ไทย |

日本語 |

한국어 |

中文 (简体) |

中文 (繁體)

#### Privacy

|

Site Terms

|

Cookie Preferences

|

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.