**ELASTIC CACHE**

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

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 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](https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/ParameterGroups.Modifying.html) or [Amazon ElastiCache for Memcached](https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/ParameterGroups.Modifying.html) User Guide.

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: 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: 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 a Redis node; creation, deletion and modification of the node can be carried out through the ElastiCache console, the command line interface 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: 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 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 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 or 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 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 node.

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: What is Multi-AZ for ElastiCache for Redis?

Multi-AZ (Multi-Availability Zone) deployments provide enhanced availability for ElastiCache replication groups, making them a natural fit for production workloads. When you provision a Multi-AZ replication group, Amazon ElastiCache automatically creates the primary and replica nodes in separate AZs and asynchronously replicates data between them. In the event of an ElastiCache node failure, Amazon ElastiCache will automatically promote a healthy read replica with the least replication lag to become primary. In the unlikely event of an AZ failure, a replica in another AZ will be promoted to become the primary node. ElastiCache also propagates the DNS changes of the promoted read replica, so your application need not be changed if it is writing to the primary node endpoint.

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

The benefits of using Multi-AZ are: (1) High Availability in the unlikely event of an AZ failure (2) ElastiCache SLA (Service Level Agreement) guarantees are available only in the Multi-AZ configuration. For more information about SLA, see ElastiCache [documentation](https://aws.amazon.com/elasticache/sla/).

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

Yes. You can export your 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](https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups.html).

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.

Q: I have multiple AWS accounts using ElastiCache for Redis. Can I use ElastiCache snapshots from one account to warm start an 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](http://docs.aws.amazon.com/AmazonS3/latest/dev/example-walkthroughs-managing-access-example2.html). 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: What does encryption in-transit for 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).

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

Encryption at-rest allows for encryption of all data on disk during backups, restore and swap, as well as backups created and stored in Amazon S3.

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

Encryption in-transit, encryption at-rest, and Redis AUTH 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 encryption and Redis AUTH and can proceed to provide an authentication token for communication with the Redis cluster. Once the cluster is setup with encryption enabled, 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.

Q: Does Amazon ElastiCache for Redis support AWS Key Management Service (KMS)?

Yes, Amazon ElastiCache for Redis supports AWS KMS to provide encryption at rest using customer managed customer master keys (CMKs) in AWS KMS. You can use your own CMKs to encrypt data at rest in ElastiCache for Redis. Customer managed CMKs are CMKs in your AWS account that you create, own and manage. For more information, see Amazon ElastiCache User Guide.

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 ElastiCache for Redis as before.

Q: Can I enable encryption in-transit and encryption at-rest on my existing 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 ElastiCache for Redis clusters. ElastiCache for Redis versions 5.0.3, 5.0.0, 4.0.10, and 3.2.6 support these features

All caches in ElastiCache:

• Support SSL in flight encryption

• **Do not support IAM authentication**

• IAM policies on ElastiCache are only used for

AWS API-level security

Memcached

• Supports SASL-based authentication (advanced)

Redis AUTH

• You can set a “password/token” when you

create a Redis cluster

• This is an extra level of security for your cache

(on top of security groups)

In this scenario, a distributed cache is suitable for storing session state data. ElastiCache can perform this role and with the Redis engine replication is also supported. Therefore, the solution is fault-tolerant and natively highly scalable.

It is common to use key/value stores for storing session state data. The two options presented in the answers are Amazon DynamoDB and Amazon ElastiCache Redis. Of these two, ElastiCache will provide the lowest latency as it is an in-memory database.

The in-memory caching provided by ElastiCache can be used to significantly improve latency and throughput for many read-heavy application workloads or compute-intensive workloads

It can be put in front of databases such as RDS and DynamoDB – sits between the application and the database.

Good if your database is particularly read-heavy and the data does not change frequently.

Also good for compute-heavy workloads such as recommendation engines and it can be used to store the results of I/O intensive database queries of compute-intensive calculation

Elasticache EC2 nodes cannot be accessed from the Internet, nor can they be accessed by EC2 instances in other VPCs.

**Exam tip:**the key use cases for ElastiCache are offloading reads from a database, and storing the results of computations and session state. Also, remember that ElastiCache is an in-memory database and it’s a managed service (so you can’t run it on EC2).

**Exam tip:** Compared to DynamoDB Accelerator (DAX) remember that DAX is optimized for DymamoDB specifically and only supports the write-through caching strategy (does not use lazy loading).

* The drawbacks of lazy loading and write through techniques can be mitigated by a TTL.
* The TTL specifies the number of seconds until the key (data) expires to avoid keeping stale data in the cache.

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.

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.