**Amazon ElastiCache**

* Managed service providing highly scalable and low latency in- memory data store
* Used for distributed caching Two Options:
  + Redis
  + Memcached

**Amazon ElastiCache for Redis**

* Highly scalable and low latency in-memory data store Can be used as a cache, database or message broker
* Automatic failover with Multi-AZ deployments (if enabled) Supports backup and restore
* Supports encryption at-rest (KMS) and in-transit
* Use cases:
  + Caching
  + Session Store
  + Chat and Messaging Gaming Leader boards
  + Geospatial Apps (Ride hailing, restaurant recommendations)
  + Queues

**Amazon ElastiCache for Redis - Cluster**

* Shard - collection of one or more nodes
* One node acts as read/write primary
* Other nodes act as read replicas (up to five read replicas)
* In case of failure:
  + Primary node is replaced
  + If Multi-AZ replication group is enabled, read replica is promoted to primary
  + DNS entry is updated

**ElastiCache Redis - Backup and Snapshot**

* Uses native backup feature of Redis (stored to S3)
* Recommended to perform snapshot against read replicas
* You can schedule snapshots
  + Configure backup window and
  + Days of backup you want to store
* Manual snapshots are available until they are manually deleted

**Amazon ElastiCache for Memcached**

* Simple caching layer intended for use in speeding up dynamic web applications
  + Pure cache
  + Non-persistent
  + Simple key-value storage
* Ideal front-end for data stores like RDS or DynamoDB Can be used as a transient session store
* Create upto 20 cache nodes
* Use Auto Discovery to discover cache nodes

**Amazon ElastiCache for Memcached - Limitations**

* Backup and restore NOT supported
* Does not support encryption or replication
* Does not support snapshots
  + When a node fails, all data in the node is lost
  + Reduce impact of failure by using large number of small nodes

**ElastiCache Memcached vs Redis**

* Use ElastiCache Memcached for
  + Low maintenance simple caching solution
  + Easy horizontal scaling with auto discovery
* Use ElastiCache Redis for
  + Persistence
  + Publish subscribe messaging
  + Read replicas and failover
  + Encryption