AWS Database Blog

Get started with Amazon ElastiCache for Valkey

by Madelyn Olson, Goumi Viswanathan, and Siva Karuturi | on 08 OCT 2024 | in Advanced (300), Amazon ElastiCache, Technical How-to | Permalink | Comments | Share

Today, Amazon ElastiCache announces support for Valkey version 7.2 with Serverless priced 33% lower and self-designed (node-based) clusters priced 20% lower than other supported engines. With ElastiCache Serverless for Valkey, customers can create a cache in under a minute and get started as low as $6/month. Valkey is an open source, high performance, key-value datastore stewarded by Linux Foundation backed by 40+ companies. Valkey is a drop in replacement of Redis OSS, developed by long standing Redis OSS contributors and maintainers, and has seen rapid adoption since project inception in March 2024. AWS is actively contributing to the Valkey project. With this launch, customers can benefit from a fully managed experience built on open-source technology while taking advantage of the 13+ years of operational excellence, security, and reliability that ElastiCache provides.

In today's post, we provide you an overview of ElastiCache for Valkey, its benefits, and how you can upgrade your ElastiCache for Redis OSS cache to ElastiCache for Valkey cache.

Overview of ElastiCache for Valkey

Amazon ElastiCache is a fully managed, Valkey-, Memcached-, and Redis OSS- compatible caching service that delivers real-time, cost-optimized performance for modern applications. ElastiCache scales to millions of operations per second with microsecond response time, and offers enterprise-grade security and reliability. With ElastiCache for Valkey, you can choose between self-designed (node-based) clusters and serverless caching deployment options. You can leverage features like automatic scaling, high availability with multi-AZ deployments, and cross-Region replication, ensuring business continuity and disaster recovery. By offering Valkey as a managed service through ElastiCache, AWS enables customers to benefit from its extensive capabilities without the operational overhead of managing Valkey themselves.

ElastiCache for Valkey offers the following benefits:

- **Lower pricing:** You can further optimize costs on ElastiCache Serverless for Valkey with 33% reduced price, and 90% lower minimum data storage of 100 MB starting as low as $6 per month. On ElastiCache for Valkey self-designed (node-based) clusters, you can benefit from up to 20% lower cost compared to other engines. Additionally, ElastiCache supports size flexibility for reserved nodes within an instance family and AWS Region. If you are using ElastiCache reserved nodes, when you switch from ElastiCache for Redis OSS to ElastiCache for Valkey, you retain your existing discounted reserved node rates across all node sizes within the same family and also get more value out of your reserved nodes discount.
- **Operational excellence:** ElastiCache for Valkey provides a fully managed experience built on open-source

technology while leveraging the security, operational excellence, 99.99% availability, and reliability from AWS.

- **Performance**: Customers choose ElastiCache to power some of their most performance-sensitive, real-time applications. It can provide microseconds read and write latency and can scale to 500 million requests per second (RPS) on single self-designed (node-based) cluster.
- **API compatibility:** ElastiCache for Valkey is compatible with Redis OSS APIs and data formats and customers can migrate their applications without needing to rewrite code or make changes to their architecture.
- **Zero-downtime migrations**: Existing users of ElastiCache for Redis OSS can quickly upgrade to ElastiCache for Valkey with zero downtime.
- **Continuous innovation:** The commitment from AWS to support Valkey ensures that customers are not only adopting a stable solution but also one that is poised for future growth and innovation. As the Valkey community continues to develop and enhance the project, AWS customers will benefit from ongoing improvements and new features that keep their applications competitive. AWS is also actively contributing to Valkey and you can read more about that on Amazon ElastiCache and Amazon MemoryDB announce support for Valkey.

Solution Overview

You can get started with Amazon ElastiCache for Valkey with just a few steps:

1. Create a ElastiCache Serverless for Valkey cache.
2. Create an Amazon Elastic Compute Cloud (Amazon EC2) instance.
3. Download and set up the valkey-cli utility.
4. Connect to the cache from an application.

We walk through these steps in the following sections. Then we demonstrate performing basic operations on the database. We also discuss how to upgrade from ElastiCache for Redis OSS to ElastiCache for Valkey.

Creating an ElastiCache Serverless for Valkey cache

You can create an ElastiCache Serverless for Valkey cache using the AWS Management Console, AWS Command Line Interface (AWS CLI), or ElastiCache API. The following code is an example of creating a ElastiCache Serverless for Valkey cache using the AWS CLI. Ensure that your CLI version is up-to-date to use Valkey resources in ElastiCache.

```
aws elasticache create-serverless-cache \
```

```
    --serverless-cache-name ec-valkey-serverless \
    --engine valkey \
    --region us-east-1
```

This creates an ElastiCache Serverless for Valkey cache in your default VPC and uses the default security group.

You can verify the status of ElastiCache Serverless cache creation using the `describe-serverless-caches` command.

```
aws elasticache describe-serverless-caches \
--serverless-cache-name ec-valkey-serverless \
--region us-east-1
{
    "ServerlessCaches": [
        {
            "ServerlessCacheName": "ec-valkey-serverless",
            "Description": " ",
            "CreateTime": "2024-08-14T21:28:11.116000+00:00",
            "Status": "available",
            "Engine": "valkey",
            "MajorEngineVersion": "7",
            "FullEngineVersion": "7.2",
            "SecurityGroupIds": [
                "sg-bba254aa"
            ],
            "Endpoint": {
                "Address": "ec-valkey-serverless-xxx.cache.amazonaws.com",
                "Port": 6379
```

Setting up EC2 for connecting to an ElastiCache Serverless for Valkey cache

ElastiCache can be accessed through an Amazon Elastic Compute Cloud (Amazon EC2) instance either in the same Amazon Virtual Private Cloud (Amazon VPC) or from an EC2 instance in a different Amazon VPC using VPC peering.

Create an EC2 instance using Getting started with EC2. ElastiCache Serverless for Valkey cache uses both ports 6379 and port 6380. To successfully connect and execute Valkey commands from your EC2 instance, your security group must allow access to these ports as needed.

Downloading and setting up valkey-cli

Connect to your EC2 instance and download valkey-cli utility by running the following commands.

```
sudo yum install gcc jemalloc-devel openssl-devel tcl tcl-devel -y
wget https://github.com/valkey-io/valkey/archive/refs/tags/7.2.7.tar.gz
tar xvzf 7.2.7.tar.gz
cd valkey-7.2.7/
make BUILD_TLS=yes install
```

For detailed instructions on using valkey-cli to connect and execute commands against Valkey engine, see the valkey-cli documentation. It's important to build support for TLS when installing valkey-cli. ElastiCache Serverless for Valkey caches are only accessible when TLS is enabled.

Connecting to Valkey cache for reading and writing

To connect to ElastiCache Serverless for Valkey cache, obtain the endpoint for your new serverless cache using the describe-serverless-caches AWS CLI command. You will find two endpoints.

```
aws elasticache describe-serverless-caches \
--serverless-cache-name ec-valkey-serverless \
--region us-east-1
            "Endpoint": {
                "Address": "ec-valkey-serverless-xxx.cache.amazonaws.com",
                "Port": 6379
            },
            "ReaderEndpoint": {
                "Address": "ec-valkey-serverless-xxx.cache.amazonaws.com",
                "Port": 6380
            }
```

Connect to ElastiCache Serverless for Valkey cache using the valkey-cli utility.

```
valkey-cli -h ec-valkey-serverless-xxx.cache.amazonaws.com -p 6379 -c --tls
```

- -h = hostname
- -p = port
- -c = cluster mode
- –tls = TLS Enabled clusters

Now, you're ready to execute basic GET and SET operations against ElastiCache Serverless for Valkey cache. The following is an example of a HSET operation to create a HASH object in Valkey. Valkey hashes are data structures used to store collections of field-value pairs. Hashes are useful when you need to represent objects or store related data in a single entity, such as a user profile with multiple attributes (for example, name, age and email).

```
ec-valkey-serverless-xxx.cache.amazonaws.com:6379> hset car:1 make ferrari model sf9
(integer) 7
ec-valkey-serverless-xxx.cache.amazonaws.com:6379>
```

This creates a Hash object car:1 with attributes such as make, model, year, engine, horsepower, transmission, and price.

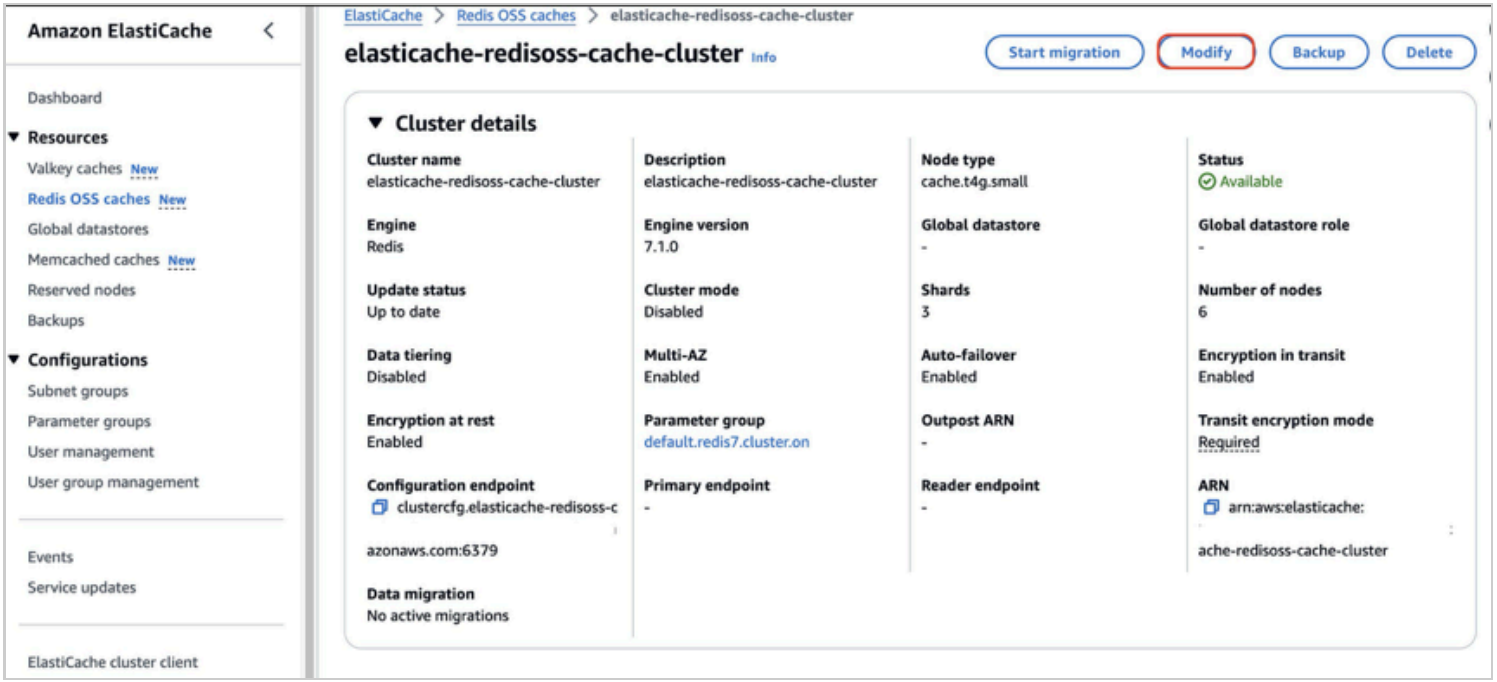
Now, you can retrieve individual or all field value pairs using the HMGET or HGETALL operation.

```
ec-valkey-serverless-xxx.cache.amazonaws.com:6379> HMGET car:1 make model price
1) "ferrari"
2) "sf90spider"
3) "580000"
ec-valkey-serverless-xxx.cache.amazonaws.com:6379>
```

Upgrade from ElastiCache for Redis OSS to ElastiCache for Valkey

If you're an existing user of ElastiCache for Redis OSS, you can quickly upgrade to ElastiCache for Valkey with zero downtime. The following example shows an existing ElastiCache for Redis OSS cache ready to be upgraded to ElastiCache for Valkey cache.

1. Initiate the upgrade process in the ElastiCache Console by choosing **Redis OSS caches** from the navigation pane, and then choosing **Modify**.



2. In the Modify ElastiCache window, under Cluster settings, you will see multiple Engine options, including Redis OSS and Valkey. Select **Valkey** as the engine option and choose **Preview Changes**

3. The next screen shows a summary of modifications. Select **Yes** under **Apply Immediately** to confirm your engine change from Redis OSS to Valkey, and then choose **Modify**. A notification **The action to modify the cluster was initiated successfully** will appear after selecting **Modify**.

4. Your existing ElastiCache for Redis OSS cache will have a status of “Modifying”.

5. Upon successful upgrade, **elasticache-redisoss-cache-cluster** no longer appears under Redis OSS caches. Instead, appears under **Valkey caches**.

6. With minimal application disruption, you have now upgraded your engine from Redis OSS to Valkey.

Cleanup

To maintain the principle of least privilege and avoid incurring future charges, delete the resources you created as part of this post. Delete the ElastiCache cluster (see deleting a cluster for more information) and EC2 instance (delete-instance)

Conclusion

In this post, we showed you how to create an ElastiCache for Valkey cache, how to setup Elastic Compute Cloud (EC2) instance for connecting to an ElastiCache Serverless for Valkey cache, and how to connect to ElastiCache Serverless for Valkey cache for reading and writing. We also saw an example of upgrading from ElastiCache for Redis OSS cache to ElastiCache for Valkey cache.

The addition of Valkey support to Amazon ElastiCache represents a significant step forward in the commitment at AWS to providing robust, open-source solutions for your database applications. If you're not already signed up, you can choose **Get started** on the ElastiCache page and complete the sign-up process. After you sign up, see the ElastiCache documentation, which includes the Getting Started guides for Amazon ElastiCache for Valkey. After you have familiarized yourself with ElastiCache for Valkey, you can create a serverless cluster within minutes by using the ElastiCache console, AWS CLI or ElastiCache API.

About the authors

Madelyn Olson is a maintainer of the Valkey project and a Principal Software Development Engineer at Amazon

ElastiCache and Amazon MemoryDB, focusing on building secure and highly reliable features for the Valkey engine. In her free time she enjoys taking in the natural beauty of the pacific northwest with long hikes and serene bike rides.

Goumi Viswanathan is a Senior Product Manager in the Amazon In-Memory Databases team. She has over 12 years of product experience and is focused on leading cross functional teams delivering database solutions. Outside of work, she enjoys traveling and spending time outdoors.

Siva Karuturi is a Worldwide Specialist Solutions Architect for In-Memory Databases based out of Dallas, TX. Siva specializes in various database technologies (both Relational & NoSQL) and has been helping customers

implement complex architectures and providing leadership for In-Memory Database & analytics solutions including cloud computing, governance, security, architecture, high availability, disaster recovery and performance enhancements. Off work, he likes traveling and tasting various cuisines Anthony Bourdain style!

Like Share

Comments

Log in to comment