Reference Architecture: Run My SQL Databases in Cloud Volumes Service

Prabu Arjunan, NetApp

February 2020

Abstract

This reference architecture provides a brief overview of MySQL database deployment using Cloud Volumes Service for AWS.

Table of Contents

MySQL is the most widely adopted open source relational database. It is the primary relational data store for many popular websites, applications, and commercial products. With more than 20 years of community-backed development and support, MySQL is a reliable, stable, and secure SQL-based database management system. The MySQL database is suitable for a wide variety of use cases, including mission critical apps, dynamic websites, and as an embedded database for software, hardware, and appliances.

AWS supports MySQL in a variety of ways, including hosting MySQL on Amazon EC2 and self-manage the database. MySQL workloads are mission-critical for most enterprises and feature prominently in discussions over an enterprise cloud migration. This blog provides a brief overview of MySQL databases, and gives a reference architecture for deploying MySQL on Cloud Volumes Service (CVS) and AWS. It also explains the benefits of running MySQL databases on CVS and AWS.

MySQL Storage Solution for Cloud

With Cloud Volumes Service for AWS, you can run a high-performance database with maximum data protection. Underlying that security is NetApp Snapshot™ technology, which offers a crucial option for rapid, efficient database backup and restoration. By design, CVS for AWS provides nine 9s of data durability.

With consistently high performance of **over 460k IOPS,** Cloud Volumes Service provides shared persistent storage with high throughput and low latency. It easily meets the demands of large MySQL databases, with SLAs that guarantee performance.

Increase the Resilience of MySQL Databases with Snapshot Copies

You can easily create a snapshot copy of a MySQL database using NetApp Snapshot ™ technology. Snapshot copies act as logical backups. They're point-in-time representations of your data, with a rapid revert function that allows you to restore your database. You create snapshot copies manually or schedule the creation of snapshot copies using the Cloud Volumes Service API or graphical user interface (GUI); rapid revert is only available from the API.

Snapshot copies are fast, plentiful, and nondisruptive. A NetApp Snapshot copy simply manipulates block pointers, creating a "frozen" read-only view of a volume that enables your applications to access older versions of files and directory hierarchies without special programming. Snapshot copy creation takes only a few seconds (typically less than 1 second) regardless of the size of the volume or the level of activity within the environment. Since they are read-only, incremental copies, you only pay for the space consumed by new data written.

Supported Protocols

Cloud Volumes Service for AWS delivers fully-managed file services for both NFS, SMB, or dual protocol support. Cloud Volumes Service for AWS support NFS v4.1 and the MySQL official documentation support the use of NFS v4 or later version. Please refer the official MySQL documentation.

Cloud Backup Service Expands Data Protection Capabilities

Cloud Backup Service (CBS) is now fully integrated into Cloud Volumes Service for AWS. It's a simple and efficient way to backup MySQL databases.

Cloud Backup Service expands the data protection capabilities of Cloud Volumes Service by delivering dedicated backups for long-term recovery, archive, and compliance. Backups created by the service are stored in AWS S3 object storage. Backups created with CBS are independent of snapshot copies, which are available for near-term recovery and rapid cloning.

Speed Up Time to Market With Instant Copy

Most organizations need multiple copies of data for testing and development. MySQL landscapes are littered with system copies for variety of uses; creating and refreshing those copies are cumbersome. Typically, creating copies of MySQL landscapes is a time-consuming and tedious process. Cloud Volumes Service for AWS allows you to fast copy and backup database files, drastically improving the process of copying, backing up, and reverting. The process takes almost no time, which ultimately leads to lower costs by way of a quicker time to market.

Data Durability

With Cloud Volumes Service, data is protected not just against multiple drive failures, but also against numerous storage media errors that can harm your data durability and your data integrity. And with 99.9999999% durability—based on the experience of over 300,000 customers—you don't have to worry that your data is going to disappear, which is underpinned by the product's SLA.

High Availability

Cloud volumes service offers 99.99% availability. Built on industry leading hardware and software, NetApp Cloud Volumes Service is characterized by high availability and uptime, both of which are enabled by architectural features, such as redundant network paths, failover, and advanced data protection.

Because NetApp Cloud Volumes Service for AWS sits centrally in relation to each of the Availability Zones within an Amazon Web Services (AWS) region, your service is unaffected by Availability Zone outages. You can access your data from any Availability Zone within the region without having to replicate content. This availability is covered by CVS's SLA.

Security and Encryption

NetApp Cloud Volumes Service uses at-rest encryption, relying on the XTS-AES 256-bit encryption algorithm. CVS encrypts your data without compromising your storage application performance. NetApp manages and rotates encryption keys for you, thus, this single-source solution can increase your organization's overall compliance with industry and government regulations without compromising your user experience.

Average Cost Savings of About 70%

When you use CVS for AWS, you control your cloud performance by dynamically adjusting service levels. If you need to increase performance, you can increase the allocation (for example, 10TB provides 160MB/s) and/or choose a higher service level.

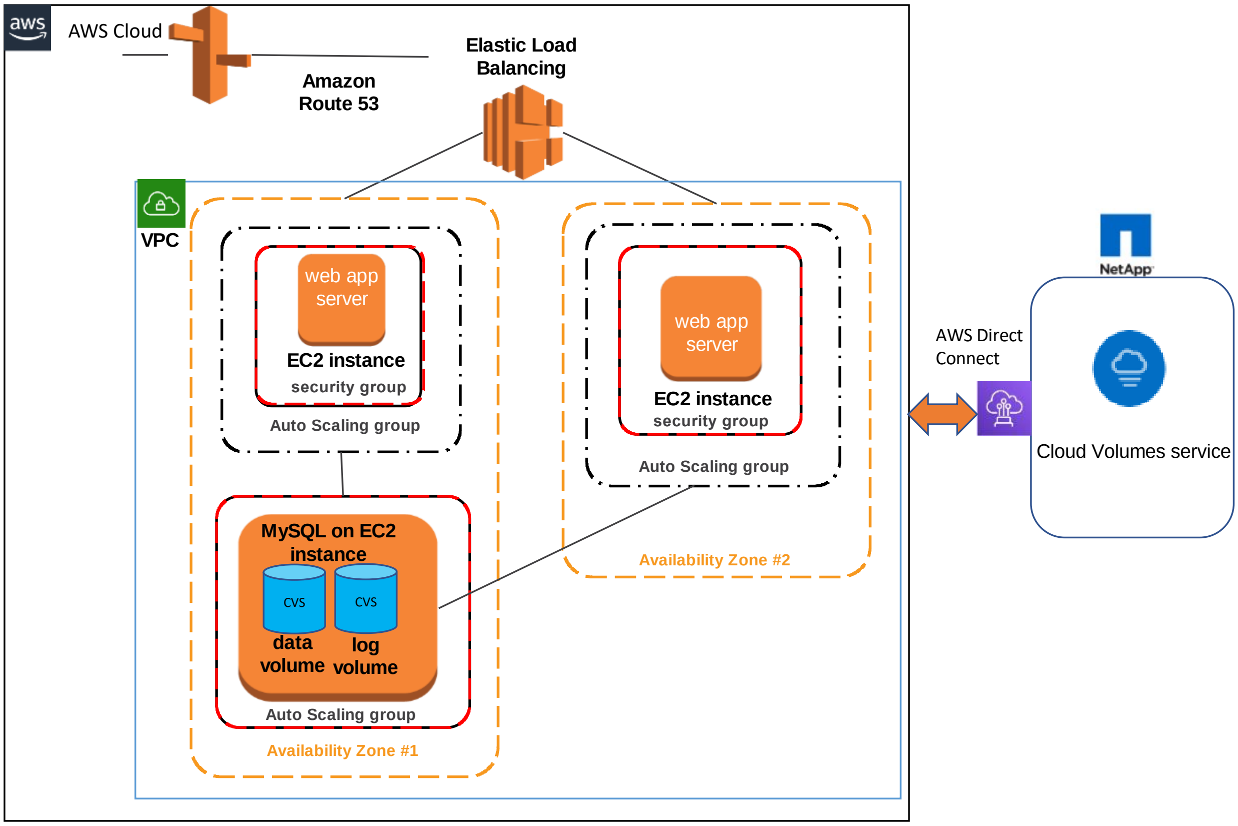
- The Standard service level offers very economical cloud storage, at just $0.10 per gigabyte per month. It enables throughput up to 16MB/s for each terabyte allocated. This level is ideal as a low-cost solution for infrequently accessed data.

- The Premium service level delivers a good mix of cost and performance. At a cost of $0.20 per gigabyte per month, it offers 4x the performance of the Standard level, with 64MB/s for each terabyte allocated. This is a good fit for many applications where data capacity and performance needs are balanced.
- The Extreme service level provides the best performance. At a cost of $0.30 per gigbytes per month, it enables up to 128MB/s for each terabyte allocated, and cloud volumes can scale to deliver several GB/s for reads and writes. Extreme is the best fit for high-performance workloads.

One of the unique features of NetApp Cloud Volumes Service for AWS is the capability to change performance on-the-fly. If the requirement is to have the Extreme performance tier for 2 hours a day and Standard performance for the rest, Cloud Volumes Service for AWS can use API calls or a scheduler in Linux to facilitate that process.

Detailed Architecture Design

Self-managed MySQL instance using CVS storage:



In the architecture diagram, you can see that the MySQL database database volume and log volumes are configured on Cloud Volumes Service for AWS. With the combination of backups, snapshot copies, and right-sized throughput, you can easily host your high-performance database in the cloud with maximum data protection and eight 9s of data durability (soon to be nine 9s).

In the diagram above, you can see:

- The MySQL database is configured on an Amazon EC2 instance.

- The database volume is provisioned using the Extreme service class because that class provides the highest throughput at a manageable cost.
- An additional volume is dedicated to logs. A second volume is provisioned using the standard service class.
- Cloud Backup Service backs data up to the S3 cloud.
- For more details on the configuration details, please refer to Cloud Backup Service

The key components of the solution include:

- MySQL database engine
- Amazon EC2 instances
- NetApp Cloud Volumes Service for AWS (storage)
- NetApp Snapshot Technology
- NetApp Cloud Backup Service

Performance

Proven performance:

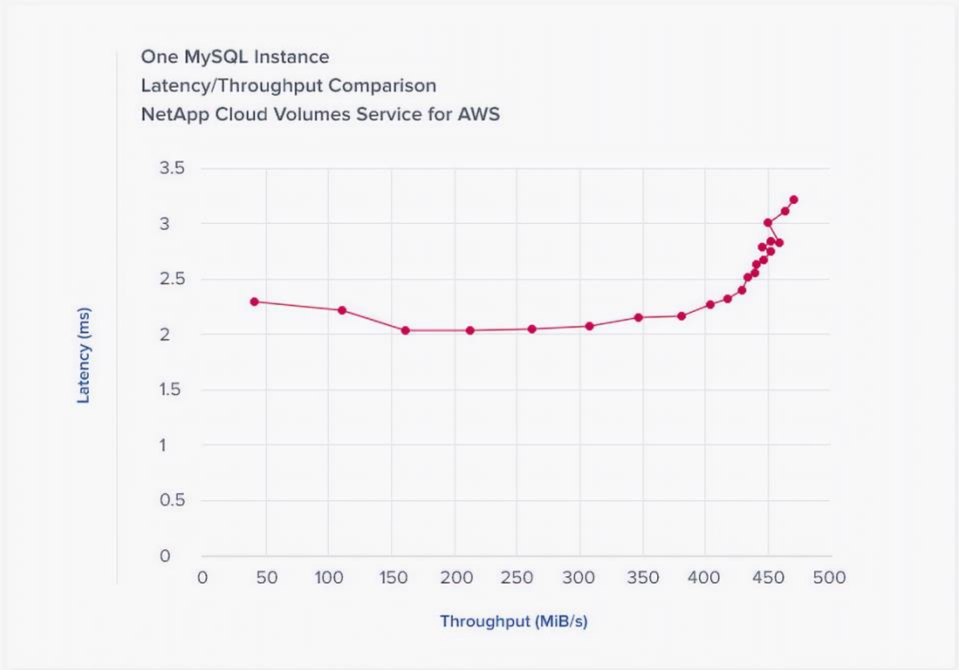
The graph below illustrates the performance of an MySQL database on Cloud Volumes Service. We ran the benchmark with various workload mixtures and volume counts. The results were stunning.

Environment:

- Instance type: C5.9xlarge
- MySQL Version: 10.3.2
- Linux Version: Redhat Enterprise Linux 7.6
- Workload Distribution to storage: 70/30 read/write with 4KiB operation database page size*
- Volume Count: database volume (8TiB Extreme), 1 log volume (1TiB Standard)
- Allocated Storage Bandwidth: database volume 1024MiB/s, log volume 16MiB/s
- Database Size: 1.25TiB
- A c5.9xlarge instance can drive up to 40,000 disk IOPS, and the c5.18xlarge supports up to 80,000 disk IOPS.

MySQL Workload – Latency Relative to Throughput

The 450MiB/s throughput observed in this benchmark test of MySQL on Cloud Volumes Service for AWS is sidling up the 5Gbps per network flow limit imposed by AWS.
The metrics in the following graph–450MiB/s maximum throughput–are taken from nfsiostat on the database server and, as such, represent the perspective of the NFS client.



The Economic Benefits of Cloud Volumes Service

Cloud Volumes Service is the lowest cost, highest quality solution to database hosting in the cloud. You can save a significant amount of time and money by changing performance levels on demand in CVS for AWS; other cloud storage solutions recommend that you configure performance and capacity to meet peak requirements, which means peak prices. Database performance requirements are rarely consistent—they require a system that's adaptable, agile. But other cloud solutions offer up monolithic structures without swift-footed switching between performance levels, recommending static throughput of the highest level for databases. You can use the NetApp API to change performance on the fly, or interface the service with a scheduler such as cron in Linux. That saves a lot of money.

For example, let's say that you're using Cloud Volumes Service and configured a volume at the Standard performance level ($0.10/GB). If you realize that you need more performance, you can update the volume with an API call or scheduler and the change happens in seconds—it's nondisruptive to clients. It's just as easy to revert to the lower performance tier. So instead of continually paying for peak performance, you only incur added costs for the time you used the higher performance tier.

Think about it like this: If you have a performance intensive workload at certain times (such as peak times), you may need a volume to perform at the Extreme level for 30TB at $.030/GB, but only during those peak periods. If you were to run at this level all the time it would cost $9000/month. But with Cloud Volumes Service for AWS, when the intensive task finishes, you can quickly drop down to the Standard performance level for 160MB/s (16MB/s x 10TB) and meet the I/O needs for off-peak loads at a significantly lower cost. This performance level costs $1,000 per month (10TB at $0.10/GB). The cost savings vary, but if you run the processing intensive workload for 20% of the time and adjust the Service level, you can usually save about $6,400 each month.

Note that the formula we used to calculate savings is: $9000 – (($9000*0.2) +($1000*0.8)) = $6,400, which **equals savings of more than 70%.**

Additionally, Cloud Volumes Service for AWS provides savings from:

- Space efficient snapshot copies, which only incur costs for unique data used in snapshot copies. This single 4KB copy is enough to protect all the data in the snapshot.
- High performance storage that enables you to use fewer compute instances, which saves time and results in lower EC2 costs.
- Support for both NFS and SMB, which enables a dataset to be shared between Linux and Windows instances.
    - Alternative solutions require an expensive and slow data copy between multiple volumes.

Code Snippet: Rest APIs

Cloud Volumes Service has rest APIs that can be called by various orchestration engines and scripting languages. Here are some example scripts that you can leverage to get started.

```
import requests
import json
import time
#Base URL
CVAPI_BASEURL="https://cv.us-west-1.netapp.com:8080/v1/"
CVAPI_APIKEY = "Supply your CVS API key here"
CVAPI_SECRETKEY = "Supply your CVS Secret key here "
#Headers
HEADERS = {
    'content-type': 'application/json',
    'api-key': CVAPI_APIKEY,
    'secret-key': CVAPI_SECRETKEY
    }

getfilesystemDetailsHeaders = {
    'content-type': 'application/json',
    'api-key': CVAPI_APIKEY,
    'secret-key': CVAPI_SECRETKEY
}
filesystemURL = CVAPI_BASEURL + "/FileSystems"
filesystemCreateURL = CVAPI_BASEURL

class cvsAPI(object):
    # get FileSystems
    def get_fileSystems(self):
        getResult = requests.get(url=filesystemURL, headers=HEADERS)
        print("File system creation success, the response code : ", getResult.status_code)
        fileSystemsData = getResult.json()
        for i in fileSystemsData[:]:
            fileSystemId = (i['fileSystemId'])
            name = (i['name'])
            print("FileSystemId : ", fileSystemId, " =   VolumeName : ", name)

    # create Volume/filesystem
    def create_fileSystems(self):
        payload = {
            "name": "IAAS",
            "creationToken": "IAAS",
            "region": "us-west-1",
            "serviceLevel": "basic",
            "quotaInBytes": 100000000000
        }
        postfileSystems = requests.post(filesystemURL, data=json.dumps(payload), headers=HEADERS)
        print("FileSystem Created", filesystemURL, postfileSystems.content)
        datafilesystems = postfileSystems.json()
        datafilesystems1 = datafilesystems['fileSystemId']
        exportname = datafilesystems['creationToken']
        time.sleep(30)
        datafilesystems1 = datafilesystems['fileSystemId']
        self.test = datafilesystems1
        self.export = exportname
        return datafilesystems1, exportname

volume = cvsAPI()
volume.get_fileSystems()
volume.create_fileSystems()
```

Ready to Get Started?

Check out Cloud Volumes Service for AWS to learn more and signup for a personalized demo.