CumuLogic DbaaS Overview Guide

February 2013

Table of Contents

CumuLogic DbaaS Overview Guide

Introduction

CumuLogic Database-as-a-Service provides fully managed instances of SQL and NoSQL database servers on any IaaS clouds, giving you access to the full functionality of **MySQL** and **MongoDB**. CumuLogic's database service provides you with access to SQL as well as NoSQL databases with a single User Interface and command line tools or RESTful API.

CumuLogic Database-as-a-Service automates the provisioning, configuration, performance optimization, management, failover, backups, updates, patching, security and access control, eliminating over 75% of administration tasks and boost developer productivity, hence time-to-market.

In addition, CumuLogic Database-as-a-Service is integrated with CumuLogic PaaS making it simple to deploy applications in any language using the database service.

Detailed Description:

CumuLogic Database-as-a-Service provides a quick way for developers and system administrators to provision fully managed instances of MySQL and MongoDB database servers in the cloud. With a single click or a single API command, you can provision a fault-tolerant, multi-node MySQL cluster, a single node instance of a standard version of MySQL server or multi-node MongoDB replica set.

When provisioning the database engines, you can configure the performance parameters (referred as DB parameters) to optimize the database service for the applications and workload types.

Once provisioned, the database instances are fully monitored and managed by the **Health Monitor**, which manages the following aspects:

- Monitoring and reporting the instance failures
- Self-healing from failures such as process crash
- Backing up the database periodically to an alternate storage, most commonly object storage if available on the target IaaS cloud
- Configuring security and access controls by using firewall settings, security groups on IaaS clouds and encrypted passwords
- Scheduling patches and applying minor updates to the running instances

Users can override the default time intervals of the scheduled automated backups, define the backup data retention policies, change the maintenance time windows, configure security and access credentials, and override any updates and patching policies of the running instances.

CumuLogic Database-as-a-Service also allows you to create multiple read-only replicas of a master database instance to scale out the applications. Ready-only replicas are synchronously replicated and can be used in case of master database failures.

CumuLogic Database-as-a-Service can be used in conjunction with CumuLogic Elastic Cache to enhance application performance.

You can deploy applications using CumuLogic PaaS and connect applications using a standard JDBC connection string or JNDI lookup name for datasource.

Service Features

Features supported by CumuLogic database service vary depending on the database engine selected. Below are some common features:

Provisioning – Single click or single API call to provision the desired database instance: size with configuration and performance parameters, single or multi-node cluster, backup settings, size of storage volumes and rules for auto update and patches.

Monitoring – Visibility into key functional and operational metrics of the database engine and instance, including CPU, memory, I/O, database connections and storage utilization.

Backup and Recovery – Users can define the backup frequency, as well as the time window for the service to initiate backups. The flexible retention policy can be modified per database instance based on specific requirements.

Database Snapshots – Users can initiate database snapshots on a running database instance and also restore a database snapshot or launch a new database instance with an existing snapshot. Users can initiate a Point In-Time recovery from the database backups or launch a new database instance using one of the existing database backups. Point-In-Time recovery is possible usually up to last 4 minutes from the time of crash.

Scaling and Replication – Database instances can be scaled out to handle excessive workloads by adding read-only replicas of running database instances. Replication of master database is managed by using native MySQL built-in replication capability. Users may also use multi-node MySQL clusters to scale out the database instances.

Automated Software Patching – CumuLogic database service provides an optional feature to apply minor updates and patches to the database engine. Users can choose when the updates are to be implemented and schedule the implementation.

Optimization – CumuLogic database service comes with a set of default performance parameters for the chosen database engine and the instance size. Users can modify the

performance parameters to suit the application workloads in real-time and in most cases without shutting down the running database instances or rebooting.

Service Benefits

CumuLogic Database-as-a-Service eliminates over 75% of common database management tasks while providing the flexibility to control the performance and scalability of the database instances.

Easy to use with single-click deployment: new database instances can be quickly spun up whether for development, QA/Testing or production purposes.

Low TCO: database instances are fully managed and monitored, eliminating most manual tasks and lowering the cost of database and application operations by over 75%.

Cost effective: CumuLogic database service scales on-demand so that users are able to optimally size instances and only scale out when needed, thusly avoiding costly over provisioning of resources.

Security at multiple levels: database instances are secured using the firewall settings and security groups of the chosen IaaS cloud, allowing users to control remote access to all database instances. Additionally, database instances can be configured to use secured connections only.

Compatible: CumuLogic database service is fully compatible with standard open source versions of MySQL, MySQL cluster and MongoDB database servers.

Multi-cloud: CumuLogic's platform abstracts the underlying APIs of any IaaS cloud, VMware vCloud or vSphere environment, allowing users to deploy the platform and the database service on any supported private or public cloud.

Use Cases

The database tier is a key architecture component for any application. Applications may use a relational database such as MySQL or a NoSQL datastore such as MongoDB, depending on the application type, the data type and the size of the data to be stored. With CumuLogic DbaaS, developers, DevOps and Ops teams can quickly launch the desired type of database, the size of the database nodes with a single API call or a single click on the UI. This eliminates the need for installing, provisioning, configuring and managing the databases completely.

MySQL for Web Applications

MySQL database service is a popular database for web applications since it can scale on-demand to handle peak workloads for a given application. You can simply launch a single DB instance and add read-only replicas when the number of users is expected to grow, and shut down the replicas when the peak load has stabilized to lower levels again. This allows you to pay for the extra database nodes only when needed.

MySQL or MongoDB for Disaster Recovery

CumuLogic MySQL and MongoDB database services make it easy and cost-effective to architect a disaster recovery (DR) solution to support mission critical applications. Typically, enterprises are required to maintain one standby deployment environment to failover to in case of downtime. When using CumuLogic DbaaS is possible to launch a fully functioning database server on the cloud or on different cloud within minutes. It's also possible to launch a replica of the database server on the alternate cloud or alternate availability zone on the same cloud. The replicas are fully synchronized and in case of downtime, applications can quickly use the replica database server nodes. If the applications are deployed on CumuLogic PaaS, the platform will provision the applications on the alternate availability zone in case of failover. CumuLogic DbaaS eliminates the need for expensive standby environments, lowering the overall cost of maintaining and implementing a DR plan.

Big Data Analysis Using MongoDB

In modern applications, the data tends to be unstructured and of very large volumes. Such data can be generated by social applications or machines, such as log files from web servers or debug data from large number of applications. Such data is usually required to be analyzed for further action by the applications. For example, analyzing the web server log files to identify the demographics of the site visitors. Because of the unstructured nature of the data and its volume, developers prefer to use NoSQL databases such as MongoDB. Unlike relational databases such as MySQL, MongoDB can store unstructured data by collecting it with JSON files. Commonly available tools can in turn analyze JSON files.

With CumuLogic MongoDB service, developers can launch a fully managed instance of MongoDB server in a single API call or command line option. MongoDB can be scaled on-demand by adding additional nodes to the datasets. CumuLogic DbaaS identifies any failed replica sets and re-provisions them as needed. MongoDB databases are automatically backed-up so it's easy to restore a new database instance when needed either to recover from a failure or to launch a staging or development database instance.

How to Use CumuLogic Database-as-a-Service

CumuLogic Database-as-a-Service can be used to launch database instances using the User Console or a simple set of API. Users can launch database instances, create read replicas, take snapshots and create and apply parameter groups. Below are the steps to use CumuLogic DbaaS:

Launching a Database Instance - Select the database engine and size of the instance (depending on the IaaS cloud) and launch your database. Optionally, you can configure the DB parameters and customize the preferences for backup and the time windows for maintenance updates. Once you have launched a DB instance you can get credentials and access point, and you can use them to configure the datasource of your application to connect to the database instance. You can also use your preferred database management tools, such as MySQL Workbench to create a schema and import data into the database instance.

Creating and Applying a New DB Parameter Group - DB parameter groups allow users to optimize the performance of database for specific workloads. You can easily create a new parameter group from a parameter family group of the engine and modify the selected parameters, and then apply that parameter group to your DB instance. You can apply DB parameter group to running instances without requiring a restart of the instance or start a new database instance with new a parameter group.

Scaling out the DB Instance - You can add additional read-only replica sets to your master database instance to handle excessive workloads. Creating additional replicas will synchronize the database with the master database instance. You can configure your applications to read from read-only replica, but you must write to the master database instance only. There may be a delay in synchronization between the master database instance and the replicas, which cause the application to get stale data at some point in time. **Use caution** when allowing application to read data from read-only replicas. Optionally, you may want to consider using in-memory cache servers using CumuLogic Elastic Cache to help you scale your applications.

Taking a Snapshot - We recommend taking full database snapshots at regular intervals or at critical points when you are making major database or application code changes. In case of required recovery, you may restore the database to in-point snapshots from these full database snapshots.

Monitoring and Optimizing - You can use monitoring metrics and charts to identify any potential issues with your database instance. As needed, you can edit the desired performance parameters by using DB parameters and/or using an in-memory caching service to scale out your database performance.

CumuLogic Database-as-a-Service FAQs

What is CumuLogic Database-as-a-Service?

CumuLogic Database-as-a-Service provides access to MySQL or MongoDB database in 3 clicks. CumuLogic Database-as-a-Service is fully managed, scalable and fault-tolerant and eliminates over 75% of tasks associated with database administration. This service is ideal for highly scalable application architectures that require flexibility and control of the application infrastructure. CumuLogic Database-as-a-Service provides fully managed and monitored database, automated backup/restore and on-demand scalability for MySQL and autoscaling for MongoDB services.

What are benefits of using CumuLogic Database-as-a-Service?

CumuLogic Database-as-a-Service provides a simple way to setting up a database in the cloud. There are technical and business benefits of using CumuLogic DbaaS:

- Managed database: Eliminates over 75% of administration tasks of MySQL or MongoDB database in the cloud.
- Instant access: Developers can launch database with a single click via the User Interface or by using a single command line or API call.
- Automated backups: The database service automatically backups the data at a time frame window defined by the user at a given frequency.
- Point-in-time recovery: Easy-to-restore database from an existing database backup or snapshot to the last backup database.
- Scale out: Easily scales out the database by adding read-only replicas of the master database.

- Performance optimization: Developers can monitor key performance metrics using the monitoring charts, and can optimize parameters accordingly without requiring to restart the database

How is CumuLogic DbaaS deployed in private datacenters?

CumuLogic Database-as-a-Service - just like any other CumuLogic cloud service - requires a core platform to be deployed first. CumuLogic provides two options to deploy CumuLogic platform and cloud services:

1. On-premise deployment: CumuLogic provides software packages, which can be deployed on-premise on VMs or servers. The CumuLogic Installer will download all necessary packages and install the software and help configure and customize for on-premise private cloud. Please refer to the User Guide and Administrator Guide for details.
2. Hosted CumuLogic Platform: The hosted version of CumuLogic Platform eliminates the need for installing and configuring the platform so customers can access the platform quickly on demand. The hosted version of the software can be customized to integrate with private clouds, as well as for the look-and-feel to comply with corporate requirements.

Does Database Service require IaaS clouds?

CumuLogic PaaS and CumuLogic Cloud Services benefit from virtualized environments to scale resources up and down as needed. CumuLogic PaaS does require either VMware vSphere virtualized environment or Infrastructure-as-a-Service (IaaS) clouds such as VMware vCloud, Citrix CloudPlatform, Apache CloudStack, OpenStack or Eucalyptus. Some of CumuLogic offerings may also be available on bare metal systems for large and high performance databases.

What clouds does CumuLogic Database-as-a-Service support?

CumuLogic Database-as-a-Service supports VMware vSphere virtualized environment and Infrastructure-as-a-Service (IaaS) clouds such as VMware vCloud, Citrix CloudPlatform, Apache CloudStack, OpenStack and Eucalyptus.

What database engines does CumuLogic database service support?

MySQL and MongoDB are the supported database engines for SQL and NoSQL databases respectively. Please refer to the latest release notes for the complete list of database engines supported.

What storage engines does MySQL database service support?

MySQL Database service supports MyISAM and InnoDB storage engines. Please refer to specific documentation on limitations on backup and restore functionalities of each storage engines.

Does CumuLogic Database-as-a-Service support NoSQL database engines?

CumuLogic Database-as-a-Service supports MongoDB for NoSQL database services. Please refer to the latest release notes for support of additional database engines.

What are the availability, resiliency and failover features of CumuLogic DbaaS?

High availability can be achieved by creating one or more database replicas of the master database instance. When the master database instance becomes unavailable, the

database service will failover to one of the replicas available. The failover will be transparent to the application. In order to increase resiliency, we recommend using one or more replicas of MySQL databases and using replicasets of MongoDB.

How does MySQL database scale in the cloud?

MySQL scalability can be achieved by adding one or more replicas of the master database instance. You can configure your application to do read-only operations against one or more replicas and use the master database for writes only.

How does NoSQL database scale in the cloud?

NoSQL database such as MongoDB can be scaled in real time by adding nodes to the replicaset. MongoDB replicasets synchronize the nodes automatically providing additional nodes to redirect the queries to.

How do the backup and restore functions work?

CumuLogic Database-as-a-Service schedules automated backups normally once in 24 hours. Each database instance is fully backed up on object storage if available on the target clouds. Users can specify the backup window, a two-hour duration time window appropriate for backups. This avoids any unwanted performance degradation on the running database instances. The database service also backs up all transaction log files for point-in-time recovery. You can also take database snapshots and full database backups. This can be useful when you are making changes to your environment, application code or database schema, and you want to keep the latest back up available.

Users can launch a new database instance to restore any previously available database backup or snapshot. For point-in-time recovery, users can specify the time up until

when they would like to restore the data, which is typically available for last five minutes of restoration.

Can the end user optimize the performance parameters of the databases?

Users can use DB parameter groups to optimize the performance of running DB instances. Default DB parameter groups are optimized for SQL query performance for a given size of the database instance.

How can I monitor database utilization and performance metrics?

Performance monitoring charts for each database instances are available on the User Console.

What are the benefits of using CumuLogic DbaaS for enterprises?

Developers need easy and timely access to application platforms so they can develop applications and test and deploy them without having to wait for resources or installing, configuring and managing the infrastructure. By providing a PaaS, developers can be much more productive, improve time-to-market of applications and deliver quality code. At the same time, IT can manage security and compliance by providing the cloud services in-house that developers seek from external clouds.

What are the benefits of using CumuLogic DbaaS for Cloud Providers?

Cloud Providers can offer higher value and higher margin application infrastructure services to their customers using CumuLogic PaaS and Cloud Services. Developers and

enterprises that are migrating applications or developing applications on clouds are looking for readily available cloud services to speed time-to-market. Cloud Providers that only offer commodity IaaS risk losing their customers to Cloud Providers who make it easy for developers to develop applications quickly.

How do CumuLogic PaaS and Cloud Services integrate with public clouds?

CumuLogic's platform is tightly integrated with IaaS clouds and relies on APIs to manage the lifecycle of VM instances, storage volumes and networking.

Can CumuLogic DbaaS be skinned to provide a different look-and-feel?

CumuLogic provides a white label solution for Cloud Providers with all the resources to customize the user interface with logos and themes to match their cloud offerings.

How can Cloud Providers bill for database services?

Cloud providers can chose their pricing models for services based on their metering and billing capabilities for cloud services. Most of the cloud providers can meter VM hours and can price their database offerings per VM hour as well.

Who provides technical support to end customers?

Cloud Providers may choose to manage the first level customer support and use CumuLogic for backend support for escalations. CumuLogic also provides full operational and customer support options for Cloud Providers.

How is DbaaS deployed in private datacenters?

Enterprises can deploy CumuLogic DbaaS on bare metal systems or a VM and use any VMware vSphere, vCloud Director or any IaaS cloud as target cloud for compute and storage resources.