



Best Practices Guide for MySQL on Amazon EC2

A MySQL[®] White Paper

June 2009

Table of Contents

Executive Summary	3
What is Cloud Computing?	3
Amazon Web Services and MySQL	3
Amazon Elastic Computing Cloud (EC2)	4
Amazon Simple Storage Service (S3)	5
Amazon Elastic Block Store (EBS)	5
Deploying MySQL on Amazon EC2	6
Best Practices for Running MySQL on Amazon EC2	7
Lower TCO with MySQL Enterprise for Amazon EC2	8
Managing and Monitoring MySQL in the Cloud	9
Conclusion	10
Resources	10
About MySQL	11

Executive Summary

Over the last ten years, open source, specifically the LAMP (Linux, Apache, MySQL and PHP, Perl, Python) stack has become the software of choice to power websites and other online applications. The rise of the LAMP stack can be attributed to its quality, ease of use and lower TCO over proprietary software. Similarly, “the cloud” is becoming more and more popular with developers and organizations looking to lower the costs of provisioning and managing their own hardware without sacrificing manageability or availability. In turn, open source software has become a natural fit for “the cloud” as developers and organizations look to the cloud as way to speed development and application delivery with a proven software stack. In this paper we present an introduction to the most popular cloud computing platform, Amazon Web Services (AWS), offer some guidance on deploying MySQL in the cloud, as well as, additional resources for monitoring and managing MySQL in the cloud.

What is Cloud Computing?

Cloud computing has emerged as a disruptive business model and set of technologies which via exposed web-services, virtualizes and to some degree automates, many of the aspects associated with the development, deployment and management of web-based applications. Cloud computing promises to deliver on-demand scale at lower costs when compared to traditional data center provisioning models. This is accomplished via “pay-as-go” schemes where companies only pay for the computing and storage resources they are actually using, typically by the hour or gigabyte stored or transferred. Thus a company can operate its computing resources more efficiently by avoiding the costs associated with underutilized computing resources or conversely over utilized where resources are exhausted business needs go unmet. Cloud providers leverage virtualization as their core technology - offering dynamically scalable storage, networking and compute resources on demand. Many software vendors are now looking at how to offer their products in the cloud, which, just like the hardware, can be easily deployed, dynamically scaled, managed and monitored. With cloud computing businesses can focus their efforts on core-competencies and push efforts related to “running a datacenter” onto the cloud providers.

Amazon Web Services and MySQL

Currently, the most popular cloud computing platform is Amazon Web Services (AWS) and the most popular database in the cloud is MySQL. Although Amazon originally launched AWS in 2002 and since has made available many new computing services including infrastructure, e-commerce and Web information services. For the purposes of this paper we will focus on those most relevant to deploying MySQL - specifically Elastic Computing Cloud (EC2), Simple Storage Service (S3) and Elastic Block Store (EBS). These are services which developers can access using web services, specifically REST and SOAP protocols.

With a subscription to MySQL Enterprise for Amazon EC2, developers can more cost-effectively deliver web-scale database applications in the "cloud", fully backed by the database experts at MySQL. Amazon EC2 and MySQL are a great fit for organizations

wanting to reduce the capital expenditures and operating costs and at the same time dynamically scale their applications with minimal cost and effort.

Amazon Elastic Computing Cloud (EC2)

The Amazon EC2 service debuted in 2006 and became generally available in late 2008. EC2 enables dynamically scalable computing capacity in Amazon's cloud. It enables developers to more easily deliver web-scale applications. Amazon Cloud computing takes provisioning of hardware out of the picture. Instead, additional virtual hardware can be commissioned (or decommissioned) as the need arises.

Powering the EC2 service is Xen virtualization technology. Xen is open source software that allows guest operating systems like Linux, Windows or Solaris to run as "virtual machines" on the same hardware concurrently. Using Xen EC2 can rapidly provision virtual servers to customer specifications customized specification of hardware characteristics such as number of CPUs and memory, and software fault tolerance.

EC2 instances can be deployed in either the United States or Europe. The Amazon EC2 Service Level Agreement commits to 99.95% availability for each region.

In EC2 virtual machines are known as "instances". EC2 instances are available in a variety of configurations divided into two main categories, "Standard Instances" and "High-CPU Instances". These instance types are based on "EC2 Compute Units", which roughly equate to the underlying CPU capacity of the physical hardware. An EC2 compute unit is currently equal to a 1.0-1.2 Ghz 2007 Opteron or 2007 Xeon processor. Amazon offers both 32-bit and 64-bit instance types which are billed by the hour.

The most popular instance types suitable for the majority of applications will be from the Standard family. These include the following configurations:

Small Instance - the default configuration, is 32-bit, has 1.7 GB of memory, 160 GB of storage and is equivalent to one EC2 Compute Unit.

Large Instance - is 64-bit, has 7.5 GB of memory, 850 GB of storage and is equivalent to four EC2 Compute Units.

Extra Large Instance - is 64-bit, has 15 GB of memory, 1690 GB of storage and is equivalent to eight EC2 Compute Units.

Amazon also offers High-CPU Instances which are geared towards applications which are characteristically CPU intensive. These instance types include:

High-CPU Medium Instance - is 32-bit, has 1.7 GB of memory, 350 GB of storage and is equivalent to five EC2 Compute Units.

High-CPU Extra Large Instance - is 64-bit, has 7 GB of memory, 1690 GB of storage and is equivalent to twenty EC2 Compute Units.

For more information concerning instance pricing and configurations please visit:

<http://aws.amazon.com/ec2/#pricing>

EC2 instances are deployed using an Amazon Machine Image (AMI). The AMI is comprised of an operating system and any additional software you chose to pre-load. Amazon has cataloged many pre-built AMIs which make getting started with EC2 that much easier. The catalog of currently available AMIs can be accessed here:

<http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=171>

Sun Microsystems offers many AMIs preloaded with MySQL and other Sun software like OpenSolaris and GlassFish. For a complete list of AMIs offered by Sun, please visit:

<http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=204>

If you have specific needs, Amazon allows you to create your own customized AMIs using their SDK.

Amazon Simple Storage Service (S3)

In 2006, Amazon also launched S3, its online storage web service. S3 provides developers with an easy, secure, persistent online storage with essentially, unlimited capacity. At a high-level, S3 can be thought of as essentially a very large disk drive or a SAN in the cloud. Amazon charges end-users per GB stored as well as bandwidth and request charges when storing and retrieving S3 data.

With S3 you can store and fetch unstructured data, which Amazon refers to as objects. These objects can be anywhere from 1 byte to 5 GB in size with 2 KB of associated metadata. Although there are no directories or file names in S3, their functional equivalents are buckets and objects. Objects are stored in buckets and are retrieved using unique developer-assigned keys. Buckets and the objects contained within them physically reside in one of the two regions, but they can, be accessed from anywhere. Using a REST or SOAP interfaces developers can create, list and retrieve buckets and objects, while downloads can be initiated via the GET interface or BitTorrent protocol.

Amazon Elastic Block Store (EBS)

When EBS became available in 2008, it was one of the most anticipated features for Amazon EC2. Prior to EBS, EC2 storage was tied to the local instance. This meant that if the EC2 instance failed, the data stored locally on the instance became unavailable. To address this, Amazon created EBS which provides block level storage volumes that persist regardless of the status of the EC2 instance. EBS volumes appear to developers as standard block devices that range from 1 GB to 1 TB in size. After assigning device names and block device interfaces, the user can then mount a file system of their choice the EBS volume.

EBS volumes provide a greater degree of data portability for EC2. For example, if you want to increase your storage capacity, you simply need to take a snapshot of the existing volume, create a new volume with more capacity and attach it back to the EC2 instance. This also works in scenarios where you need to upgrade or downgrade to a different EC2 instance type. Its important to mention that multiple EBS volumes can be mounted to the same EC2 instance. This supports striping or partitioning data across

multiple volumes for better I/O and throughput performance. This is especially valuable for databases which are I/O intensive.

Another benefit of EBS volumes over local EC2 storage is that EBS volumes are replicated and placed in what Amazon calls, "Availability Zones." With EBS you have multiple Availability Zones within a single region (United States or Europe.) The idea behind Availability Zones is to mitigate data loss in the event of hardware or site failure, and provide lower latency within a region. In order to provide this each Availability Zone has its own physically distinct datacenter infrastructure. As you might expect, EBS volumes you attach to your EC2 instance must reside in the same Availability Zone.

EBS also supports backup using point-in-time snapshots to S3. Each backup is a differential snapshot -only the blocks which have changed since the last snapshot are saved. Point-in-time snapshots provide an effective and efficient way to create durable backups, especially for your MySQL databases.

If the availability of your data in your MySQL database is critical, it is really imperative to leverage EBS volumes. Since EC2 instances that fail take the local data with them, you shouldn't allow a failed EC2 instance to render your data unavailable. With EBS you can simply start up another EC2 instance and attach it to the EBS volume. Next, according to Amazon's EBS page, "The latency and throughput of Amazon EBS volumes is designed to be significantly better than the Amazon EC2 instance stores in nearly all cases." This means the I/O performance of your MySQL database should be better running on EBS rather than local EC2 storage. Another reason to consider EBS for your MySQL database is that it enables you to upgrade from one instance type to another by simply attaching the EBS volume to the larger EC2 instance. Finally, as mentioned, EBS provides the ability to create snapshots of your MySQL databases, so if you need to run additional copies of your database or simply require simple way to create point-in-time recovery for your database, then EBS volumes are the way to go.

Deploying MySQL on Amazon EC2

Getting started with MySQL on EC2 is straight-forward. Initially, assuming you have already set up your accounts with Amazon, you can use an existing AMI pre-configured with MySQL from Amazon's AMI catalog or create your own using Amazon's SDK. Or you can get already available "templated" AMI images from other sources. Next you should upload your AMI to S3 for reuse and safe keeping once you have made any customizations. Finally, select which EC2 instance type you want to deploy, and configure the security and network controls

Other items to consider include which region (United States or Europe), whether or not you will be attaching EBS volumes and selecting the appropriate availability zone.

For additional information and detailed instructions on deploying MySQL on EC2 please consult the following resources:

Amazon EC2 Getting Started Guide

<http://docs.amazonwebservices.com/AWSEC2/latest/GettingStartedGuide/>

MySQL Reference Manual

<http://dev.mysql.com/doc/refman/6.0/en/ha-vm-aws.html>

Running MySQL on Amazon EC2 with Elastic Block Store

<http://developer.amazonwebservices.com/connect/entry.jspa?externalID=1663>

Best Practices for Running MySQL on Amazon EC2

1. Run only one MySQL Server per EC2 instance

MySQL will arguably perform better with dedicated resources. By dedicating resources you ensure that MySQL does not compete with other components like web servers, caching servers or application servers for CPU, memory and especially disk access. By having dedicated resources, you can configure larger buffers and caches for the database. Finally, if your database tuning strategy typically involves throwing hardware at the problem, in this case virtual, then having plenty of dedicated resources is the way to go.

2. Leverage Large or X-Large EC2 instance types

The Standard Large and X-Large EC2 instance types offer plenty of instance storage by default (850 GB or 1690 GB), a 64-bit operating platform, decent amounts of RAM (7.5 GB and 15 GB) for buffers and caches, 4 or 8 EC2 Compute Units and better I/O performance than the 32-bit instance types.

3. Dedicate additional partitions for InnoDB tablespaces

Databases are notorious for being I/O bound. Consider adding dedicated EBS partitions in order to get better I/O performance.

4. Warm up InnoDB tablespace partitions

By default, there is a “first write” penalty with EC2. Subsequent writes will be much faster after the disk has been warmed up. The partitions can be “warmed up” ahead of time using various methods including the “dd” Unix command. Depending on the size of the partition, this can take a very long time, but ultimately will result in faster initial writes to your database.

5. Tune MySQL appropriately for your application, architecture and workload

EC2 in and of itself doesn't make MySQL run faster, scale, or make it easier to manage by default. Tuning MySQL appropriately only means you will be able to better utilize EC2's resources and cost savings. Here it is important to review all the available tools at your disposal detailed in the “Managing and Monitoring MySQL in the Cloud” section.

6. Leverage MySQL Replication when appropriate

MySQL's asynchronous replication can be used to increase the scalability of reads or as a first step towards creating a more highly-available MySQL architecture.

7. Leverage pre-built AMIs

Building AMI's from scratch can be time consuming, error-prone and tedious. If you do not require extensive customizations to your software, take advantage of the catalog of pre-built and officially supported AMI's available from a variety of vendors.

8. Secure your EC2 instances

Aside from using the standard well documented security measures for protecting MySQL, you should also restrict access to the EC2 instance such that only authorized clients and

replication servers can connect. By default, EC2 has a built in firewall which denies all inbound traffic, so you must explicitly open additional ports. Unfortunately, this firewall does have the flexibility to restrict traffic by protocol, port and IP. Also be sure to leverage security groups to create access rules applicable to a group of EC2 instances.

For more information on securing MySQL, please visit:

<http://dev.mysql.com/doc/refman/5.1/en/security.html>

For more information on Amazon Web Services Security Processes, please visit:

http://s3.amazonaws.com/aws_blog/AWS_Security_Whitepaper_2008_09.pdf

9. Mount partitions with “noatime” and “nodiratime”

Mounting your partitions with the “noatime” and “nodiratime” options should yield better I/O performance, up to 10%. This is because Linux will not have to perform a write operation after every read access. It should be noted that noatime is a superset of nodiratime.

10. If your data matters, use EBS with MySQL

With EBS you get persistent storage for your data and log files. It also gives you the ability to fail over to another MySQL Server in the event the existing EC2 instance it is hosted on fails by simply remounting the EBS volume to the new EC2 instance.

11. Leverage Snapshots on S3

As part of your backup strategy you should leverage EC2’s ability to make a snapshot of your volumes and store them on S3. These efficient backups can be leveraged for long term storage or for point-in-time recovery operations.

12. Load balance your application

Any application with reasonably high load or that uses MySQL Replication or where you have deployed a “sharding” architecture, will benefit from some degree of load balancing. There are several options, among them Amazon Elastic Load Balancing and HAProxy, a popular open source load balancer. More information on these load balancers at:

<http://developer.amazonwebservices.com/connect/entry.jspa?externalID=1639>

and

<http://aws.amazon.com/elasticloadbalancing/>

Lower TCO with MySQL Enterprise for Amazon EC2

MySQL Enterprise for Amazon EC2 is a subscription based support offering which enables developers to cost-effectively deliver web-scale database applications in the cloud with the world’s most popular open source database in the cloud, MySQL.

Benefits of MySQL Enterprise for Amazon EC2 include:

- Deliver massively scalable web applications using MySQL proven reliability, performance and ease of use on Amazon EC2
- Implement highly-available applications using MySQL Replication for master/slave database replication, failover and backup
- Optimize MySQL for Amazon EC2 with MySQL performance and scalability best practices
- Quickly and easily add computing capacity as your requirements change by adding more computing resources
- Reduce database TCO by up to 90% by eliminating proprietary database license fees
- Cut IT operating costs by only using Amazon EC2 computing resources that you need, when you need them.
- Reduce downtime with 24x7 Production Support from the database experts at MySQL

For more information concerning MySQL Enterprise on EC2 visit:

<http://www.mysql.com/products/enterprise/ec2.html>

Managing and Monitoring MySQL in the Cloud

There are many open source and proprietary tools for deploying, monitoring and managing MySQL on EC2. Below is brief list of available resources:

Amazon EC2 API Tools

The API tools serve as the client interface to the Amazon EC2 web service. Use these tools to register and launch instances, manipulate security groups, and more.

<http://developer.amazonwebservices.com/connect/entry.jspa?externalID=351>

Rightscale

Provides control, administration, and lifecycle support of your cloud deployments.

<http://www.rightscale.com/>

Elasticfox Firefox Extension for EC2

ElasticFox is a Mozilla Firefox extension for managing your Amazon EC2 account. Launch new instances, mount Elastic Block Storage volumes, map Elastic IP addresses, and more.

<http://developer.amazonwebservices.com/connect/entry.jspa?externalID=609>

Amazon CloudWatch

Amazon CloudWatch is a web service that provides monitoring for AWS cloud resources, starting with Amazon EC2

<http://aws.amazon.com/cloudwatch/>

Amazon Auto Scaling

Auto Scaling allows you to automatically scale your Amazon EC2 capacity up or down according to conditions you define.

<http://aws.amazon.com/autoscaling/>

Javascript Scratchpad for Amazon EC2

Simple HTML/Javascript based application that allows you to explore the Amazon EC2 API without writing any code.

<http://developer.amazonwebservices.com/connect/entry.jspa?externalID=1424>

AWS Simple Monthly Calculator

<http://calculator.s3.amazonaws.com/calc5.html>

AWSzone.com

AWS code samples and generator

<http://www.awszone.com/>

Amazon EC2 AMI Tools

The Amazon EC2 AMI Tools are command-line utilities to help bundle an Amazon Machine Image (AMI), create an AMI from an existing machine or installed volume, and upload a bundled AMI to Amazon S3.

<http://developer.amazonwebservices.com/connect/entry.jspa?externalID=368>

For more tools that support Amazon EC2, please visit:

<http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=89>

Conclusion

In this paper we covered a few of the benefits cloud computing offers organizations looking to leverage consumption based pricing models for their computing infrastructure. We provided a brief overview of the Amazon Web Services, specifically EC2, S3 and EBS, and how MySQL interacts with them. We showed some basic best practices for deploying MySQL in the Amazon cloud and presented some additional resources for monitoring and managing MySQL in the cloud.

Resources

White Papers

<http://www.mysql.com/why-mysql/white-papers/>

Case Studies

<http://www.mysql.com/why-mysql/case-studies/>

Press Releases, News and Events

<http://www.mysql.com/news-and-events/>

Live Webinars

<http://www.mysql.com/news-and-events/web-seminars/>

Webinars on Demand

<http://www.mysql.com/news-and-events/on-demand-webinars/>

About MySQL

MySQL is the most popular open source database software in the world. Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, critical business systems and packaged software - including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube and Zappos.com. At <http://www.mysql.com>, Sun provides corporate users with commercial subscriptions and services, and actively supports the large MySQL open source developer community.

To discover how Sun's offerings can help you harness the power of next-generation Web capabilities, please visit <http://www.sun.com/web>.