



# **Database-as-a-Service vs. Do-It-Yourself MySQL in the Cloud**

Cashton Coleman, ClearDB



- Database-as-a-Service vs. Do-It-Yourself MySQL in the Cloud
  - ◆ When you're looking to move your website, app, or your whole business from your traditional hosting provider, one of the first challenges is whether or not to run your own MySQL database and MySQL-powered infrastructure (the DIY approach) vs. using a cloud database service to run it for you.
  - ◆ The ability to run 80-100% of your app using readily available services to reduce time, resources and cost vs. DIY and "rolling your own" components, including your database, is very appealing. This session will look at the pros and cons of working with a cloud database-as-a-service.

# MySQL in the Cloud: What Does It Really Cost?

# DIY MySQL Cost of Ownership

- ◆ What should be included when calculating DIY MySQL cost in the cloud?
  - ◆ Server instance
  - ◆ Durable storage
  - ◆ Durable storage I/O transactions
  - ◆ Cold storage for backups
  - ◆ Bandwidth
  - ◆ Availability monitoring software

# DIY MySQL Cost of Ownership (cont' d)

- What about H/A and Disaster recovery cost?
  - ◆ H/A software suite for MySQL (if you have it)
  - ◆ Multiple server instances
  - ◆ Shared-nothing architecture (no support for shared storage in the cloud)

# DIY MySQL Cost of Ownership (cont' d)

## ➤ What about human cost?

- ◆ IT operations staff for 24/7 management and uptime
- ◆ DBA with 24/7 on-call for MySQL specific workload issues
  - For when runaway queries need to be killed
  - For when lock contention brings MySQL to a standstill
- ◆ All staff must be fully qualified to handle both routine as well as catastrophic events, **even at 3a.m. on weekends!**

# DIY MySQL Cost of Ownership (cont' d)

- Unless you are in the database operations business, you're not focused on making money
  - ◆ By leveraging \*aaS services that are available in the cloud today, you can avoid having to spend significant portions of your IT budget on people, licenses, servers, and other critical infrastructure required to operate your database.
  - ◆ **Why should you re-invent the wheel?**

# MySQL costs less when used via DBaaS providers



# Things Fail in the Cloud (Why? Lightening.. Rain.. Hail!! 😊 )

# Things Fail in the Cloud

- ◆ The cloud is always moving, changing, breaking, and recovering
  - ◆ Never assume that one server is enough for **anything**
  - ◆ Never assume that **two** servers are enough for much more!
- ◆ Database systems hold state!
  - ◆ Do you have a data durability policy that you can implement today?
  - ◆ Do you have a failover strategy in the event that your primary master dies?
  - ◆ Do you have a recovery strategy to revive a failed master?
    - › **Can you execute that strategy in 10 minutes or less?**

- **Are you sure you can recover from a cloud failure?**
  - ◆ **Have you ever tried it?**
  - ◆ **Were you successful?**
  - ◆ **Did you find it to be easy? (trick question)**

# The Best Way to Fight Failure: Strength in Numbers

# H/A MySQL-as-a-Service to the Rescue!

- A true H/A MySQL solution can save you from having to worry about these issues:
  - ◆ Sub-second failover with a multi-master design ensures that your application never goes offline due to a failure
  - ◆ Auto-rebuild / self-healing technology ensure that failed nodes self-recover
- Most important advantage:
  - ◆ You don't have to jump up at 3 a.m. to fix a VERY complex problem that could be costing your customers every second that your app is down

# What Level of H/A is Acceptable?

- ◆ Each use case has its own set of priorities:
  - ◆ Departmental workloads may not need 24/7 support
    - › Small catalog apps, blogs, etc.
  - ◆ Internal business apps only need to be online while people (or things) are using them
    - › Corporate intranets, internal catalog systems, etc.
  - ◆ Enterprise customer-facing workloads need to be online 24/7
    - › Corporate websites
    - › Airline travel reservation systems and services
    - › Banking and trading systems
    - › Enterprise shipping systems and services
    - › Telecom systems and services

# How Many Nines Do You Need?

- ◆ Some providers only support 99.9% uptime
  - ◆ That's 43 minutes of unplanned downtime per month
- ◆ Higher grade H/A providers support 99.999 uptime
  - ◆ That's 40 seconds of unplanned downtime per month
- ◆ Even 100% uptime **is possible**, but it takes a lot of work
  - ◆ No unplanned downtime is the best for enterprise workloads
  - ◆ Requires a geo-distributed solution with real-time failover support

# Use a DBaaS to Get Higher Uptime

- ◆ Database-as-a-service providers focus on your database, uptime, and availability to enable you to reduce your cost and risk.
  - ◆ Some providers focus on scalability and elasticity
  - ◆ Others focus on availability, durability, and security



# The Goal: Save money and reduce risk!

- ❖ Leverage the collective mountain of expertise, capability, and investment that cloud service providers have made to make your life easier
- ❖ Reduce complexity by working with providers who know how to simplify your database experience
- ❖ Reduce risk by working with providers who know how to ensure your database is always online, available, and secure

# Thank You!

**cash@cleardb.com**

**Twitter: @cash\_coleman**