

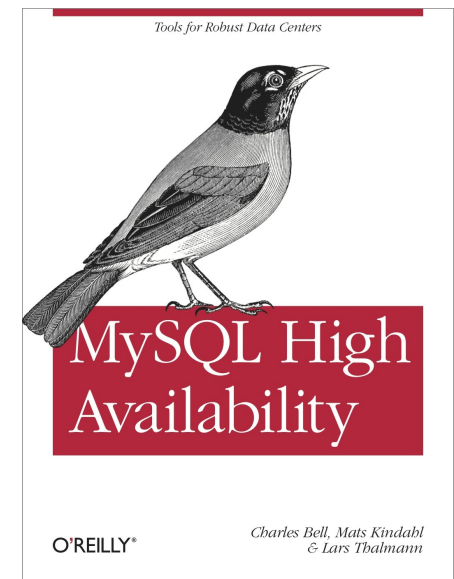
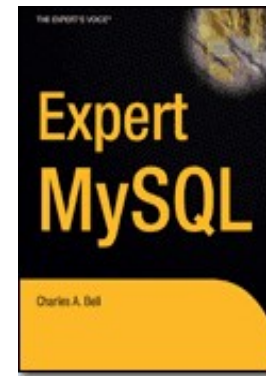
ORACLE[®]

Introduction to Using MySQL in Cloud Computing

Chuck Bell, Mats Kindahl, Lars Thalmann

About the Speakers

- Chuck Bell, PhD
 - Enterprise Backup and Replication
 - (recovering) Windows Developer
 - chuck.bell@oracle.com
- Mats Kindahl, PhD
 - Replication Expert and Lead Developer
 - mats.kindahl@sun.com
- Lars Thalmann, PhD
 - Development Manager, Replication and Backup
 - lars.thalmann@sun.com



Topics

- What is Cloud Computing?
- Vendors
- Amazon Web Services
 - Getting Started
 - Walk-through: Launching an Instance
- Economics
- Use Cases
- Benefits and Risks
- Using MySQL in the Cloud
- MySQL Cloud Computing Best Practices

What is Cloud Computing?

- Cloud Computing **is not**:
 - A new technology
 - One vendor's dream of conquest
 - Simply the internet
 - Simply virtualization
 - Rebadged bespoke environments

What is Cloud Computing?

- Cloud Computing **is**:
 - Grid computing + virtualization + API = new uses for old toys!
 - On-demand, self-service
 - Internet accessible
 - Resource pooling
 - On-demand resources (elasticity)
 - Measured services
 - Many things to many people (sadly)

What is Cloud Computing?

- Service Models
 - Infrastructure as a Service (IaaS)
 - Resources are provided as virtual instances of complete hardware or operating system platforms.
 - Platform as a Service (PaaS)
 - The vendor provides the hosting environment and programming tools to permit customers to build solutions for the specific environment.
 - Software as a Service (SaaS)
 - Software is provided as a resource in the form of applications that run on the provider's hardware.
- Deployment Models
 - Private
 - Community
 - Public
 - Hybrid

Vendors

- 3Tera (<http://www.3tera.com/>)
 - An IaaS provider specializing in rapid scale out capabilities.
- Akamai (<http://www.akamai.com/>)
 - An IaaS provider specializing in managing data on the web.
- Amazon (<http://aws.amazon.com/>)
 - A cloud computing vendor offering virtualized SaaS, PaaS, and IaaS solutions along with storage solutions.
- Enki Consulting (<http://www.enkiconsulting.net/>)
 - An IaaS provider specializing in virtual private data center solutions.
- IBM Blue Cloud (<http://www.ibm.com/ibm/cloud/>)
 - A cloud computing vendor offering virtualized SaaS, PaaS, and IaaS solutions.

Vendors

- Joyent (<http://www.joyent.com/>)
 - An IaaS provider specializing in the needs of large enterprises.
- Layered Technologies (<http://www.layeredtech.com/>)
 - A PaaS and IaaS provider.
- Rackspace (<http://www.rackspace.com/>)
 - A PaaS specializing in providing host services for web applications.
- Salesforce.com (<http://www.salesforce.com/>)
 - A SaaS vendor specializing in shared CRM solutions.
- Terremark (<http://www.terremark.com/>)
 - An IaaS provider.

Amazon Web Services

- Portfolio of Web Enabled Services
 - Computational (“cloud”)
 - Database support
 - e-Commerce tools
 - Messaging
 - Monitoring
 - Storage Solutions
 - ...and much more!

Amazon Web Services

- Fee-based service but pay only for what you use
- No hardware maintenance required
- Many tools and resources to support developers

Compute

Amazon Elastic Compute Cloud (EC2)
Amazon Elastic MapReduce
Auto Scaling

Content Delivery

Amazon CloudFront

Database

Amazon SimpleDB
Amazon Relational Database Service (RDS)

E-Commerce

Amazon Fulfillment Web Service (FWS)

Messaging

Amazon Simple Queue Service (SQS)

Monitoring

Amazon CloudWatch

Networking

Amazon Virtual Private Cloud (VPC)
Elastic Load Balancing

Payments & Billing

Amazon Flexible Payments Service (FPS)
Amazon DevPay

Storage

Amazon Simple Storage Service (S3)
Amazon Elastic Block Storage (EBS)
AWS Import/Export

Support

AWS Premium Support

Web Traffic

Alexa Web Information Service
Alexa Top Sites

Workforce

Amazon Mechanical Turk

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

Amazon Web Services (AWS): Getting Started

- Sign up for AWS
 - Easy process
 - Requires method of payment (it's not 'free'!)
- Get Your Credentials
- Read about Core Technologies
 - Amazon Elastic Compute Cloud (EC2)
 - Amazon Simple Storage Service (S3)
 - Amazon Elastic Block Storage (EBS)
- Read about Common Tools
 - Amazon Console
 - EC2 Command Line Tools
 - EC2 API Tools
 - Browser Plugins (ElasticFox)

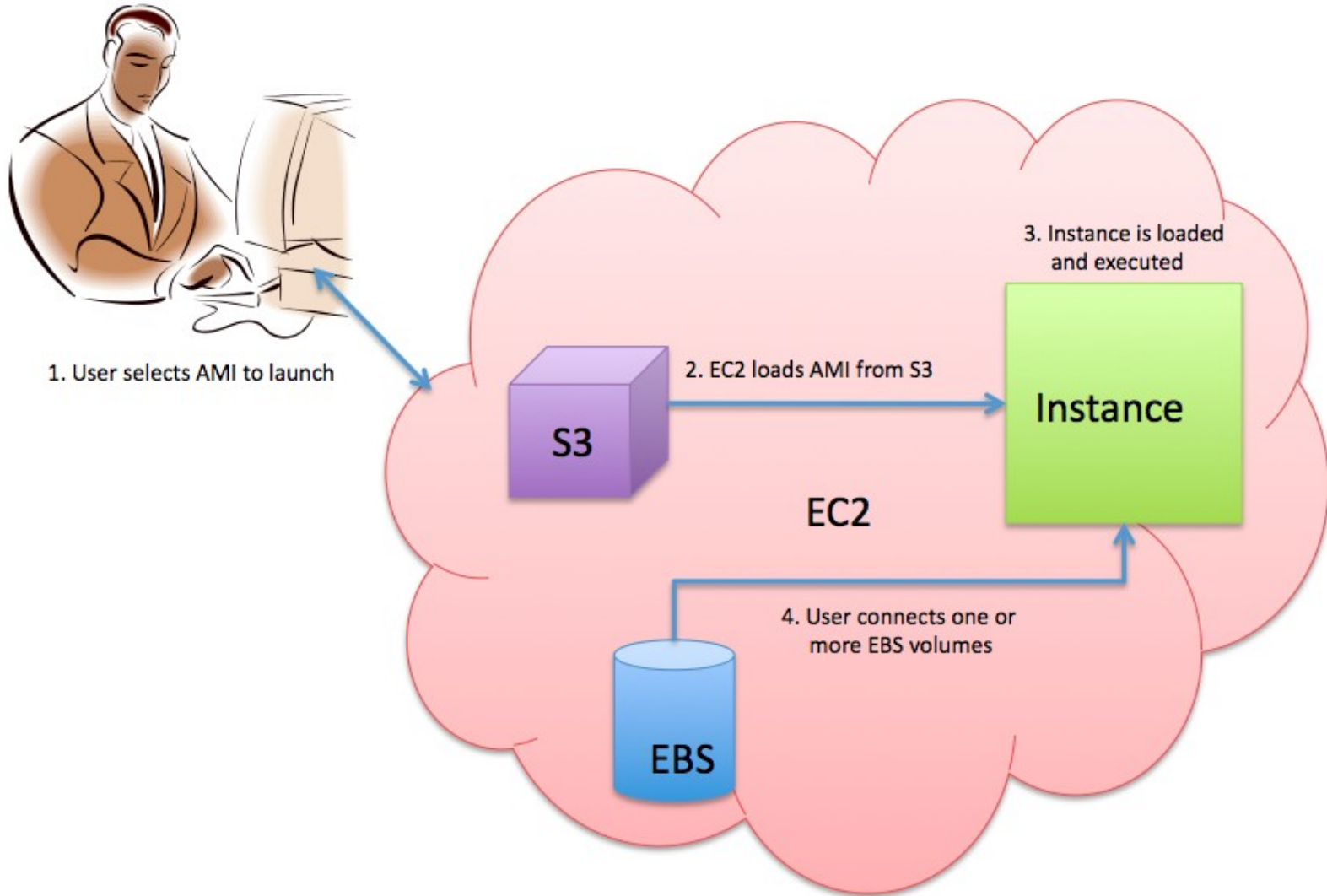
AWS: Credentials

- Amazon Login and Password
 - AWS Console and all account actions
- Access KeyID and Secret Access Key
 - Query APIs, Elastic Fox, etc.
- SOAP and EC2 Command-line Tools
 - X.509 certificate
- Working with Instances
 - SSH key pair
- Other services may require special credentials
- To get these, go to the Account | Security Credentials site from the AWS Console:
<http://aws.amazon.com>

Concepts

- Amazon Simple Storage Service (S3)
 - Bucket storage for images and data
 - Independent 'internet' storage
- Amazon Machine Image (AMI)
 - Binary file containing a server extract (OS, tools, etc)
- Instance
 - An AMI that has been loaded and is in an executable state
 - SSH key pair
- Amazon Elastic Block Store (EBS)
 - Block storage device for data
 - Used with instances

Amazon Cloud : Launching an Instance



Amazon Console Walk-through

Example: Launch an Instance

The screenshot displays the Amazon EC2 Console Dashboard. At the top, the navigation bar includes the AWS logo, the URL 'aws.amazon.com', and links for Products, Developers, Community, Support, and Account. A user is logged in as 'Welcome, [redacted]' with links for Settings and Sign Out.

The main content area is titled 'Amazon EC2 Console Dashboard' and is divided into several sections:

- Navigation:** A sidebar on the left with a 'Region' dropdown set to 'US East'. It lists categories like INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORKING & SECURITY, each with sub-links such as 'Instances', 'AMI's, 'Volumes', 'Snapshots', 'Elastic IPs', 'Security Groups', 'Key Pairs', and 'Load Balancers'.
- Getting Started:** A central yellow box with the text: 'To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.' Below this is a prominent 'Launch Instance' button. A note states: 'Note: Your instances will launch in the US East [redacted] region.'
- My Resources:** A summary of resources in the 'US East [redacted]' region, with a 'Refresh' button. The counts are: 0 Running Instances, 0 Elastic IPs, 0 EBS Volumes, 0 EBS Snapshots, 2 Key Pairs, and 3 Security Groups. There are also 0 Load Balancers.
- Service Health:** A section showing the 'Current Status' of Amazon EC2 (US East - N. [redacted]) as 'Service is operating normally'. A link is provided to 'View complete service health details'.
- Related Links:** A list of links including Documentation, All EC2 Resources, Forums, Feedback, and Report an Issue.

At the bottom of the dashboard, there is a footer with copyright information: '© 2008 - 2009, Amazon Web Services LLC or its affiliates. All right reserved.' and links for Feedback, Support, Privacy Policy, Terms of Use, and 'An amazon.com company'.

AWS Console: Launch Instance

Amazon EC2 Console Dashboard


Getting Started

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

[Launch Instance](#)

Note: Your instances will launch in the [REDACTED] region.


Service Health

Current Status	Details
 Amazon EC2 ([REDACTED])	Service is operating normally

[View complete service health details](#)

AWS Console: Launch Instance













Request Instances Wizard

Cancel 



Choose an Amazon Machine Image (AMI) from one of the tabbed lists below by clicking its **Select** button.

Quick Start | My AMIs | Community AMIs

	Getting Started on Fedora Core 8 (AMI Id: ami-b232d0db) Minimal Fedora Core 8, 32-bit architecture, Apache 2.0, and Amazon EC2 AMI Tools.	Select 
	Ruby on Rails Web Starter (AMI Id: ami-22b0534b) Fedora Core 8, 32-bit architecture, Ruby 1.8.6, Rails 2.2.2, RubyGems 1.3.1, Mongrel 1.1.5, and MySQL 5.0.45	Select 
	LAMP Web Starter (AMI Id: ami-2cb05345) Fedora Core 8, 32-bit architecture, PHP 5.2, Apache 2.2, and MySQL 5	Select 
	Basic Fedora Core 8 (AMI Id: ami-84db39ed) Minimal Fedora Core 8, 32-bit architecture, and Amazon EC2 AMI Tools.	Select 
	Basic 64-bit Fedora Core 8 (AMI Id: ami-86db39ef) Fedora Core 8, 64-bit architecture, and Amazon EC2 AMI tools.	Select 
	Getting Started on Microsoft Windows Server 2008 (AMI Id: ami-69c32f00) Microsoft Windows Server 2008 R1 SP2 Datacenter edition, 32-bit architecture, Microsoft SQLServer 2008 Express, Internet Information Services 7, ASP.NET 3.5.	Select 

AWS Console: Launch Instance

Request Instances Wizard Cancel

CHOOSE AN AMI **INSTANCE DETAILS** CREATE KEY PAIR CONFIGURE FIREWALL REVIEW

Provide the details for your instance(s). You may also decide whether you want to launch your instances as "on-demand" or "spot" instances.

Number of Instances: **Availability Zone:**

Instance Type:

Launch Instances

EC2 Instances let you pay for compute capacity by the hour with no long term commitments. This transforms what are commonly large fixed costs into much smaller variable costs.

Request Spot Instances

[< Back](#)

AWS Console: Launch Instance

Request Instances Wizard Cancel X

CHOOSE AN AMI **INSTANCE DETAILS** CREATE KEY PAIR CONFIGURE FIREWALL REVIEW

Number of Instances: 1
Availability Zone: No Preference

Advanced Instance Options

Here you can choose a specific **kernel** or **RAM disk** to use with your instances. You can also choose to enable CloudWatch Monitoring or enter data that will be available from your instances once they launch.

Kernel ID:

RAM Disk ID:

Monitoring: Enable CloudWatch Monitoring for this instance
(additional charges will apply)

User Data:

base64 encoded

[< Back](#) [Continue >](#)

AWS Console: Launch Instance

Request Instances Wizard Cancel

CHOOSE AN AMI INSTANCE DETAILS **CREATE KEY PAIR** CONFIGURE FIREWALL REVIEW

Public/private key pairs allow you to securely connect to your instance after it launches. To create a key pair, enter a name and click **Create & Download your Key Pair**. You will then be prompted to save the private key to your computer. Note, you only need to generate a key pair once - not each time you want to deploy an Amazon EC2 instance.

Choose from your existing Key Pairs

Your existing Key Pairs*:

Create a new Key Pair

Proceed without a Key Pair

[< Back](#) [Continue !\[\]\(58187f59de15d6f7dd2494352d583a00_img.jpg\)](#)

AWS Console: Launch Instance

Request Instances Wizard Cancel X

CHOOSE AN AMI INSTANCE DETAILS CREATE KEY PAIR **CONFIGURE FIREWALL** REVIEW

Security groups determine whether a network port is open or blocked on your instances. You may use an existing security group, or we can help you create a new security group to allow access to your instances using the suggested ports below. Add additional ports now or update your security group anytime using the Security Groups page. All changes take effect immediately.

Choose one or more of your existing Security Groups


Create a new Security Group

1. Name your Security Group

2. Describe your Security Group

3. Define allowed Connections

Application	Transport	Port	Source Network (IPv4 CIDR)	Actions
SSH	tcp	22	All Internet	<input type="button" value="Remove"/>
HTTP	tcp	80	All Internet	<input type="button" value="Remove"/>
MySQL	tcp	3306	All Internet	<input type="button" value="Remove"/>
<input type="text" value="Select..."/>	-	-	All Internet Change	<input type="button" value="Add Rule"/>


[< Back](#)  The security group 'default' is reserved

AWS Console: Launch Instance

Request Instances Wizard Cancel

CHOOSE AN AMI INSTANCE DETAILS CREATE KEY PAIR CONFIGURE FIREWALL **REVIEW**

Please review the information below, then click **Launch**.


AMI:  Other Linux AMI ID ami-2cb05345 (i386)
Name: LAMP Web Starter
Description: Fedora Core 8, 32-bit architecture, PHP 5.2, Apache 2.2, and MySQL 5 [Edit AMI](#)

Number of Instances: 1
Availability Zone: No Preference
Monitoring: Disabled
Instance Type: Small (m1.small)
Instance Class: On Demand [Edit Instance Details](#)

Kernel ID: Use Default
Ramdisk ID: Use Default
User Data: [Edit Advanced Details](#)

Key Pair Name: orig [Edit Key Pair](#)

Security Group(s): LAMP [Edit Firewall](#)

[< Back](#) **Launch** 

AWS Console: Launch Instance

My Instances

Launch Instance Instance Actions Reserved Instances Show/Hide Refresh Help

Viewing: All Instances All Instance Types 1 to 2 of 2 Instances

Instance	AMI ID	Root Device Type	Type	Status	Lifecycle	Public DNS	Security Groups
<input checked="" type="checkbox"/> i-3d5e9556	ami-2cb	instance-store	m1.small	running	normal	ec2-184-73-10-112.com	LAMP
<input type="checkbox"/> i-c95a91a2	ami-2cb	instance-store	m1.small	terminated	normal		LAMP

1 EC2 Instance selected

EC2 Instance: i-3d5e9556

Description Monitoring

AMI ID: ami-2cb05345 **Zone:** us-east-1a

Security Groups: LAMP **Type:** m1.small

Status: running **Owner:** 936657535344

Reservation: r-72c0891a **Ramdisk ID:** ari-dbc121b2

Platform: - **Key Pair Name:** orig

Kernel ID: aki-f5c1219c **Monitoring:** disabled

AMI Launch Index: 0 **Elastic IP:** -

Root Device: - **Root Device Type:** instance-store

Block Devices: N/A - Instance Store

Lifecycle: normal

Public DNS: ec2-184-73-10-112.compute-1.amazonaws.com

Private DNS: ip-10-212-106-195.ec2.internal

Launch Time: 2010-03-02 19:58 EST

State Transition Reason:

AWS Console: Launch Instance

Connect Help - Secure Shell (SSH) Cancel

Instance: i-e7bb658c

To access your instance using any SSH client

1. Open the SSH client of your choice (e.g., PuTTY, terminal).
2. Locate your private key file, `new_mac.pem`
3. Use `chmod` to make sure your key file isn't publicly viewable, ssh won't work otherwise:
`chmod 400 new_mac.pem`
4. Connect to your instance using instance's public DNS [`ec2-184-73-64-130.compute-1.amazonaws.com`].

Example

Enter the following command line:

```
ssh -i new_mac.pem root@ec2-184-73-64-130.compute-1.amazonaws.com
```



Close

AWS Console: Create EBS Volume


EBS Volumes

Create Volume Delete Attach Volume Detach Volume Create Snapshot Show/Hide Refresh Help

Viewing: All Volumes 1 to 1 of 1 Items

	Volume ID	Capacity	Snapshot	Created	Zone	Status	Attachment Information
<input checked="" type="checkbox"/>	 vol-a7fe22ce	1 GiB	--	2010-03-09 17:14 EST	us-east-1a	 available	

1 Elastic Block Store Volume selected

 **Volume ID:** vol-a7fe22ce

Capacity: 1 GiB **Snapshot:** **Zone:** us-east-1a

Status: available

Attachment:

Created: 2010-03-09 17:14 EST

AWS Console: Attach Volume

Attach Volume Cancel

Volume: vol-a7fe22ce in us-east-1a

Instances: in us-east-1a

Device:

Windows Devices: xvdf through xvdp
Linux Devices: /dev/sdf through /dev/sdp

Command line Walk-through

Commonly Used Commands

- `ec2-add-key-pair`
 - Creates a new SSH key pair.
- `ec2-run-instances`
 - Launches EC2 instances. You must specify at least the name of the image and your key pair. You can launch multiple instances at the same time.
- `ec2-describe-images`
 - Lists available images. Output includes the image ID, the location of the image in S3, and whether the image is available for launching. There are a number of parameters you can use to limit the search.
- `ec2-stop-instances`
 - Stops or pauses instances. You can stop multiple instances at the same time.
- `ec2-start-instances`
 - Starts or resumes instances. You can start multiple instances at the same time.
- `ec2-terminate-instances`
 - Terminate instances. You can terminate multiple instances at the same time.

AWS: Command-line Access

- Search for images to launch

```
Chucks-MacBook-Pro:~ Chuck$ ec2-describe-images -o self -o amazon |  
grep mysql
```

```
IMAGE    ami-225fba4b ec2-public-images/fedora-core4-apache-mysql-  
v1.07.manifest.xmlamazon    available    public        i386machine  
instance-store
```

```
IMAGE    ami-25b6534c ec2-public-images/fedora-core4-apache-  
mysql.manifest.xmlamazon    available    public        i386machine  
instance-store
```

```
IMAGE    ami-255fba4c ec2-public-images/fedora-core4-mysql-  
v1.07.manifest.xmlamazon    available    public        i386machine  
instance-store
```

```
IMAGE    ami-22b6534b ec2-public-images/fedora-core4-mysql.manifest.xml  
amazon    available    public        i386machine  
instance-store
```

AWS: Command-line Access

- Launch the instance

```
Chucks-MacBook-Pro:~ Chuck$ ec2-run-instances ami-225fba4b -k new_mac
RESERVATION  r-2249194a    936657535344 default
INSTANCE     i-75af711e    ami-225fba4b    pending  new_macm1.small
              2010-03-09T02:13:27+0000  us-east-1d      monitoring-
disabled     instance-store
```

- View instance status

```
Chucks-MacBook-Pro:~ Chuck$ ec2-describe-instances
RESERVATION  r-2249194a    936657535344 default
INSTANCE     i-75af711e    ami-225fba4b  ec2-184-73-9-65.compute-
1.amazonaws.com  domU-12-31-39-02-EC-E7.compute-1.internal
running  new_mac  0    m1.small 2010-03-09T02:13:27+0000  us-east-1d
              monitoring-disabled  184.73.9.65  10.248.243.21
              instance-store
```

AWS: Command-line Access

- Authorize access

```
Chucks-MacBook-Pro:~ Chuck$ ec2-authorize default -p 22
GROUP          default
PERMISSION     default  ALLOWS   tcp 22  22  FROMCIDR0.0.0.0/0
```

- Connect

```
Chucks-MacBook-Pro:~ Chuck$ ssh -i ./ec2_credentials/new_mac.pem
root@ec2-184-73-9-65.compute-1.amazonaws.com
  __|  __|_ )  Rev: 2
  _|  (      /
  ___|\___|___|
Welcome to an EC2 Public Image
                :-)

    Apache2+MySQL4
  __ c __ /etc/ec2/release-notes.txt
[root@domU-12-31-39-02-EC-E7 ~]# exit
logout
Connection to ec2-184-73-9-65.compute-1.amazonaws.com closed.
```

AWS: Command-line Access

- Terminate the instance

```
Chucks-MacBook-Pro:~ Chuck$ ec2-terminate-instances i-75af711e
```

```
INSTANCE    i-75af711e  running shutting-down
```

```
Chucks-MacBook-Pro:~ Chuck$ ec2-describe-instances
```

```
RESERVATION r-2249194a  936657535344  default
```

```
INSTANCE    i-75af711e  ami-225fba4b  ec2-184-73-9-65.compute-1.amazonaws.com domU-12-31-39-02-EC-E7.compute-1.internal shutting-down  new_macm1.small  2010-03-09T02:13:27+0000 us-east-1d  monitoring-disabled  184.73.9.65 10.248.243.21  instance-store
```

- Ensure it is terminated

```
Chucks-MacBook-Pro:~ Chuck$ ec2-describe-instances
```

```
RESERVATION r-2249194a  936657535344  default
```

```
INSTANCE    i-75af711e  ami-225fba4b  terminated new_mac 0 m1.small  2010-03-09T02:13:27+0000  us-east-1d  monitoring-disabled  instance-store
```

Economics

- Initial cost of traditional hardware is very expensive
- Long-term cost of highly active cloud can exceed the daily operating cost of traditional infrastructures
- Typically, cloud computing is cheaper in the long run because:
 - You never, ever buy hardware
 - You pay for what you need and nothing more
- Depending on your usage, it is possible cloud computing may not be cost effective
 - Training costs
 - Retooling your application
- The bottom line: Do your homework!

Use Cases

- Traditional web services
 - Content providers
- Shared services
 - One or more applications shared by multiple users/customers
- Scale out from enterprise
 - Expand traditional information technologies into the web
 - Amazon Virtual Private Cloud (VPC)
- Cloud bursting
 - Allow rapid expansion of an application to meet needs that may wax and wane over time
- Research and Development
 - Conduct experimentation without needing dedicated (and temporary) hardware resources

Benefits

- Reduced run time and response time
 - Use grid or scale out techniques to meet demands and improve throughput times
- Minimize infrastructure risk and eliminate maintenance
 - Hardware failure is no longer your responsibility
 - Do more with less staff and less hardware
- Lower cost of entry
 - Pay only for what you need
- Increased pace of development
 - Develop and deploy faster with less investment

Risks

- Potentially run-away costs
 - Heavy load and vast scale out will increase costs
 - Duplication of work will incur unnecessary costs
- Lack of features
 - Some vendors may not have the features you need making it difficult to continue to use their products.
- Security risks
 - Since you are sharing resources with others, you need to focus on security more than ever.
- Service failure
 - If the service you rely on fails, you're toast

Using MySQL in the Cloud

- Where's the magic?
 - Nothing special about MySQL in the cloud (in general)
 - Full features available
 - Same tools to manage and monitor
- So what is different?
 - Rapid deployment
 - Rapid scale out

Example: MySQL Replication and EC2

- Master is in the 'cloud', slave is local (real hardware)
- Process
 - Launch an instance of the LAMP Web Starter image
 - Connect to instance and configure it as a master
 - Start the MySQL server on the instance
 - Start a local MySQL server
 - Configure the local MySQL server to be a slave
 - Connect to the master in the cloud
- As you can see, no surprises here

Example: MySQL Replication and EC2

```
Chucks-MacBook-Pro:~ Chuck$ ssh -i ./keys/orig.pem root@ec2-184-73-10-112.compute-1.amazonaws.com
```

```
  __|  __|_ )  Fedora 8
  _| (      /   32-bit
  __|\__|__|
```

Welcome to an EC2 Public Image

:~)

Base

```
--[ see /etc/ec2/release-notes ]--
```

```
[root@ip-10-212-106-195 ~]# mysql -uroot
```

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 2

Server version: 5.0.45 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

```
mysql>
```

```
mysql> SHOW MASTER STATUS;
```

```
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000001 |      98 |              |                   |
+-----+-----+-----+-----+
```

1 row in set (0.00 sec)

```
mysql>
```

```
mysql> GRANT REPLICATION SLAVE ON *.* TO 'rpl'@'%' IDENTIFIED BY 'rpl';
```

Example: MySQL Replication and EC2

```
mysql> CHANGE MASTER TO MASTER_HOST='ec2-184-73-10-112.compute-1.amazonaws.com',
    MASTER_USER='rpl', MASTER_PASSWORD='rpl', MASTER_PORT=3306, MASTER_LOG_FILE='mysql-
    bin.000001', MASTER_LOG_POS=98;
Query OK, 1 row affected (0.00 sec)
mysql> START SLAVE;
Query OK, 1 row affected (0.00 sec)
mysql> SHOW SLAVE STATUS \G
***** 1. row *****
Slave_IO_State: Waiting for master to send event
Master_Host: ec2-204-236-207-171.compute-1.amazonaws.com
Master_User: rpl
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mysql-bin.000001
Read_Master_Log_Pos: 325
Relay_Log_File: mysqld-relay-bin.000002
Relay_Log_Pos: 470
Relay_Master_Log_File: mysql-bin.000001
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
...
  Exec_Master_Log_Pos: 325
    Relay_Log_Space: 626
    Until_Condition: None
...
```

MySQL Cloud Computing Best Practices

- Run only one MySQL Server per EC2 instance
 - MySQL will run faster with dedicated computational units and memory resources.
- Leverage larger instance types for heavy usage
 - You should consider using larger instance types for high transactional or heavy read or write databases.
- Use additional EBS volumes for InnoDB tablespaces
 - Use multiple EBS volumes to separate your databases and use multiple InnoDB tablespaces (e.g., one per EBS volume) to get better I/O performance.
- Warm up data partitions
 - You can use the Linux dd command to write to the disk. While the penalty still occurs and cannot be avoided, at least the first write to your databases will not suffer the effects.

MySQL Cloud Computing Best Practices

- Be sure to configure MySQL properly
 - Simply running MySQL in EC2 isn't going to make it faster.
- Don't forget monitoring
 - You can and should monitor your MySQL servers running in the cloud.
- Use MySQL replication
 - MySQL replication is for scale out, load balancing, and high availability. The cloud makes these features easier to use.
- Use standard AMIs
 - Unless you have vast experience in running the host operating system in a virtual environment, the task of building custom AMIs can be time consuming and error prone. Whenever possible, use the existing Amazon AMIs or known stable community AMIs.

MySQL Cloud Computing Best Practices

- Mount partitions with noatime or nodiratime options
 - Mounting your partitions with either of these options should yield up to 10% better I/O performance.
- Use EBS with MySQL
 - EBS is a block store device with good performance and sustainability from instance failures, as well as elasticity.
- Perform snapshotting Using S3
 - Snapshots are an effective and efficient way to take a backup and provide a mechanism for rapid recovery in the event that data becomes corrupt.
- Use load balancing
 - You can use the Amazon Elastic Load Balancing resource or even run your own software load balancer such as HAProxy.

Where To Go From Here

The Amazon cloud products are very complex and can lead to a steep learning curve. Fortunately, a lot of resources are available on the AWS website. We list a few of the more commonly accessed and must-read links here.

<http://aws.amazon.com/documentation/ec2/>

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

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

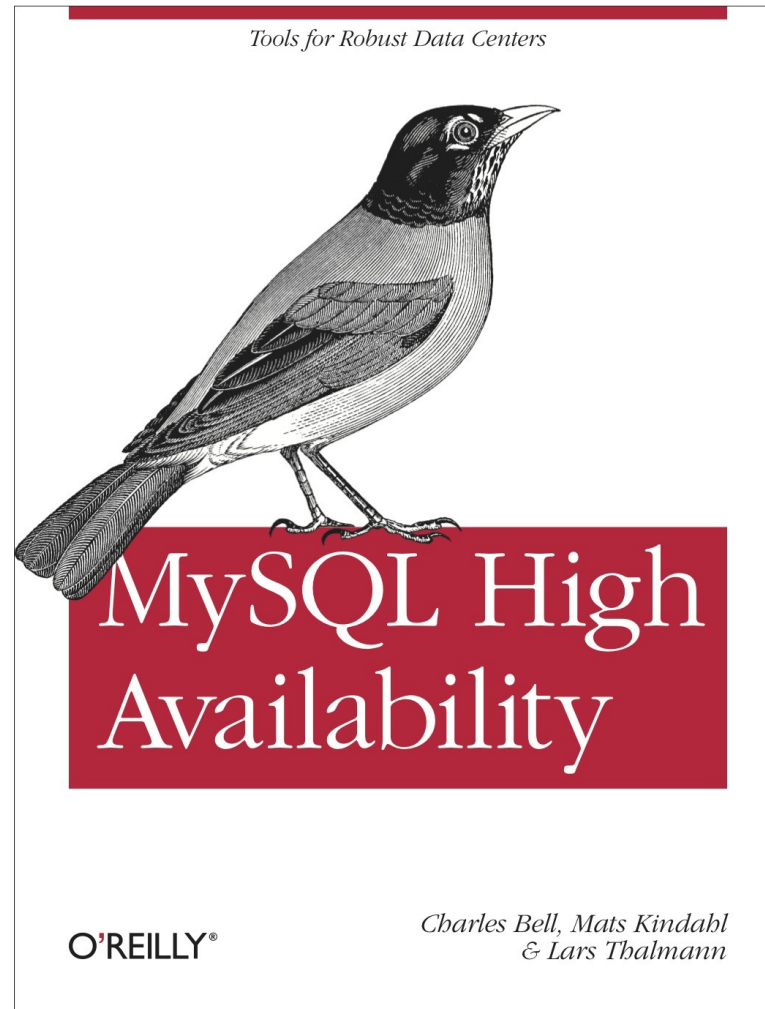
<http://aws.amazon.com/s3/>

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

Questions?

Buy this book!

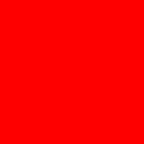
**Coming soon:
June 2010**



Buy the book!



ORACLE®



The presentation is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

ORACLE®