



Java is a trademark of Sun Microsystems, Inc.

JavaOneSM

Architecting Robust Applications for Amazon EC2

Chris Richardson

Chris Richardson Consulting, Inc

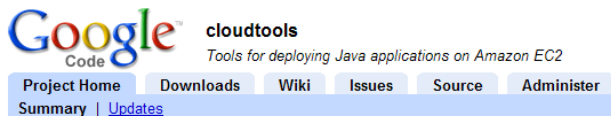
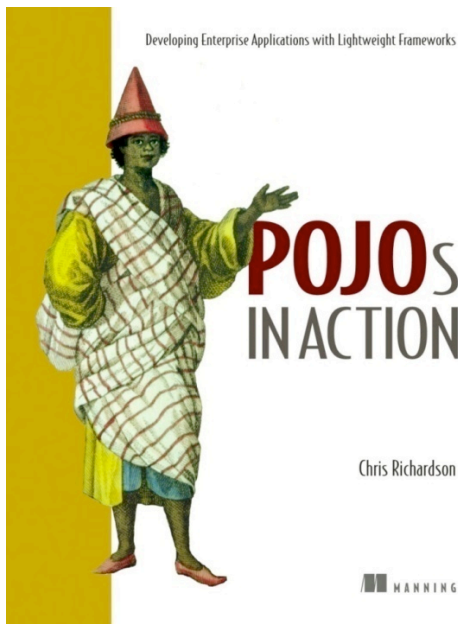
www.chrisrichardson.net

chris@chrisrichardson.net / @crichardson

Overall presentation goal

Show how to deploy Java and Grails
applications on
Amazon Elastic Compute Cloud

About Chris



- Grew up in England and live in Oakland, CA
- Over 20+ years of software development experience including 12 years of Java
- Author of POJOs in Action
- Speaker at JavaOne, SpringOne, NFJS, JavaPolis, Spring Experience, etc.
- Chair of the eBIG Java SIG in Oakland (www.ebig.org)
- Run the Groovy/Grails meetup (<http://java.meetup.com/161>)
- Run a consulting and training company that helps organizations reduce development costs and increase effectiveness
- Founder of Cloud Tools, an open-source project for deploying Java applications on Amazon EC2: <http://code.google.com/p/cloudtools>
- Founder of a startup that provides outsourced, automated, and Java-centric datacenter management on the cloud: www.cloudfoundry.com

Agenda

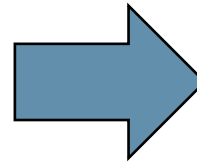
- > **Amazon-style cloud computing**
- > Using Amazon EC2
- > Deploying on Amazon EC2
- > Running the web tier
- > Deploying app servers
- > Deploying a database
- > Handling security
- > Building highly available systems

Power generation

Past



Present

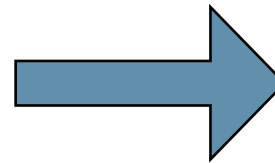


Computing has come a long way...

Past



www.computermuseum.org.uk

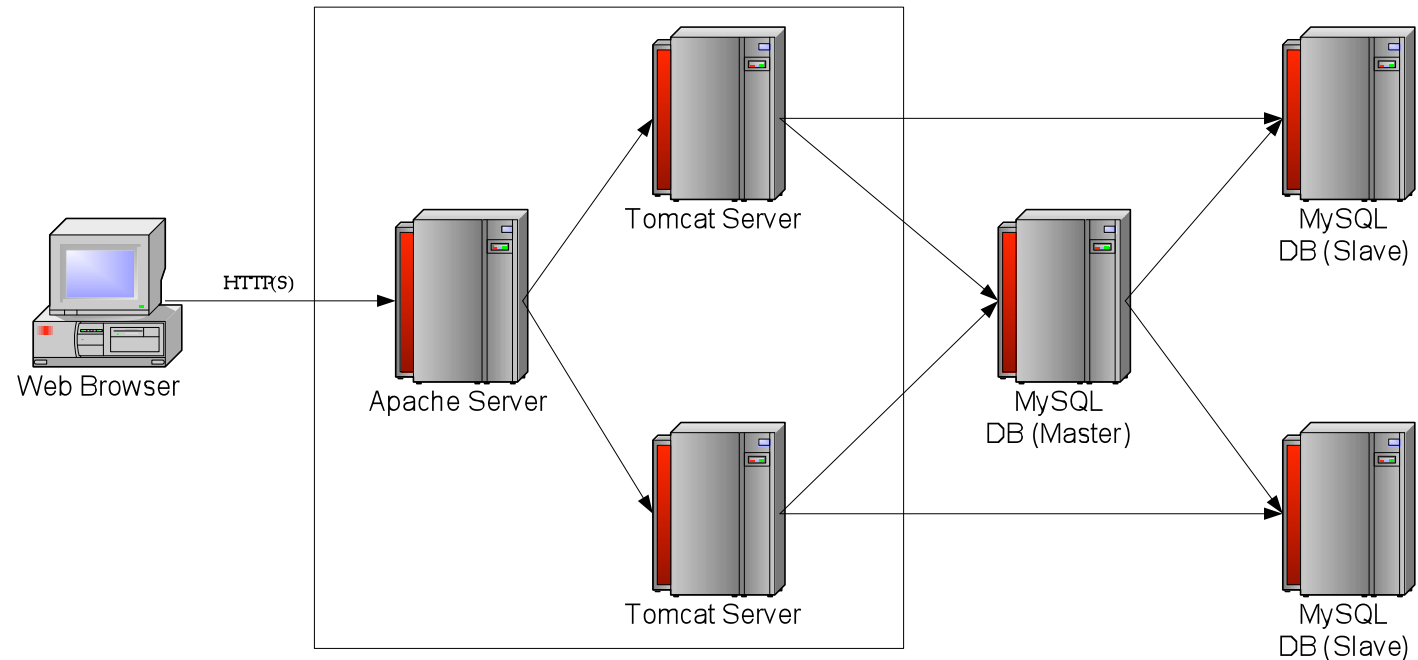


Present



www.dell.com

Yet ...



Do we know how much hardware to buy?
Can we afford it?
How long does it take to buy and install?
Who is going to set it up?
Can we afford a test lab?

Cloud computing

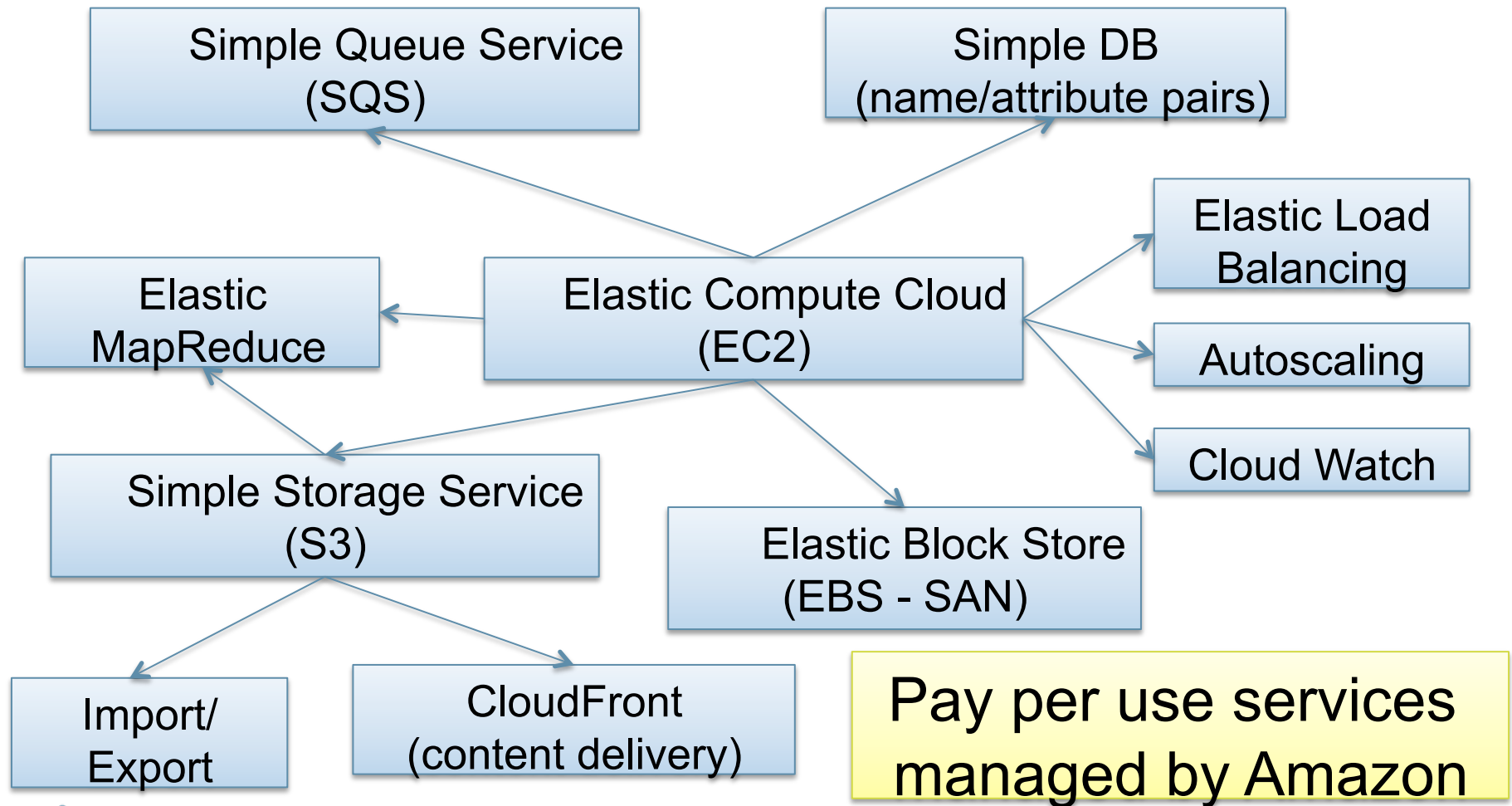
A pool of highly scalable, abstracted infrastructure that hosts your application, and is billed by consumption

By James Staten
of Forrester
Research

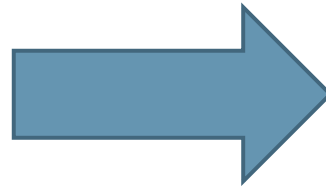
**AND
is managed via a web services API**

me

Amazon-Style Cloud Computing



Sign up



- Login using your existing Amazon account
- Select the web services you want to use
- Only takes a few minutes
- But can sometimes be confusing: various ids, keys, certificates etc

Make web service calls...

```
https://ec2.amazonaws.com?  
Action=RunInstances  
&ImageId=ami-3795705e  
&MaxCount=1  
&MinCount=1  
...
```



```
<RunInstancesResponse>  
<reservationId>r-60907709</reservationId>  
<ownerId>556666664445</ownerId>  
...  
<instancesSet>  
  <item>  
    <instanceId>i-4ef21327</instanceId>  
    <imageId>ami-3795705e</imageId>  
    <instanceState>  
      <code>0</code>  
      <name>pending</name>  
    </instanceState>  
    <placement>  
      <availabilityZone>us-east-1b</availabilityZone>  
    </placement>  
    <dnsName/>  
    <reason/>  
    <keyName>gsg-keypair</keyName>  
    <amiLaunchIndex>0</amiLaunchIndex>  
  </item>  
</instancesSet>  
</RunInstancesResponse>
```

... a few minutes later

```
cer@arrakis ~  
$ ssh ... root@ec2-67-202-41-150.compute-1.amazonaws.com  
Last login: Sun Dec 30 18:54:43 2007 from 71.131.29.181  
[root@domU-12-31-36-00-38-23:~]
```

Pay per use computing

	Virtual Cores	Compute Units / core*	32/64 Bit	Memory	Storage	\$/hr **
Small	1	1	32 bit	1.7G	160G	0.10
High-CPU Medium	2	2.5	32 bit	1.7G	350G	0.20
Large	2	2	64 bit	7.5G	850G	0.40
Extra Large	4	2	64 bit	15G	1690G	0.80
High-CPU XL	8	2.5	64 bit	7G	1690G	0.80

* EC2 Compute Unit = 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor

** Windows more expensive, reserved instances are cheaper, external bandwidth: \$0.10-0.18/Gbyte

Operating systems

- > Use Amazon provided Machine Image (AMI)
 - 32/64-bit Fedora Core 4/6/8
 - Windows Server 2003 (\$0.125-\$2/hour)
 - Optional SQL Server Standard (\$1.10-3.20/hour)
- > Many 3rd parties have public AMIs
 - Various Linux distributions
 - E.g. Redhat, RightScale
- > Sun provides OpenSolaris
- > Build your own AMI:
 - Install applications starting with existing AMI and save new AMI
 - Create an AMI from scratch

Using AWS in your application

- > S3 - Store media etc in S3
- > SQS - messaging between loosely coupled components
- > SimpleDB – alternative to RDBMS
- > CloudFront – to distribute content
- > Using these APIs
 - A highly scalable infrastructure that is maintained by Amazon
 - Couples your application to AWS
 - But using them is optional

Developing on EC2

- > Immediate access to many servers
- > Simplified setup
- > Great for testing

Deploying on Amazon EC2 – startups/ small businesses

- > Get up and running quickly
- > Validate your business idea without:
 - Upfront costs
 - Long-term financial commitment
- > Leverage operational expertise of others
- > Easily identify the right hardware for your application
- > Scale up/down with load
- > Reduces the risk of a success catastrophe

Required by some
VCs

Deploying on Amazon EC2 – enterprises

- > Increased agility - no need to wait for corporate IT
 - In some companies it can take 2 months to acquire hardware
 - Requires a long-term financial commitment, upfront costs
- > Use for short-term projects, e.g.
 - Websites for marketing campaigns
 - New York Times style projects
- > Reduce costs - use for applications that have fluctuating loads, e.g.
 - heavily used once a week, once a month

Example – beer on the cloud

- > Grails application
- > Short-term marketing campaign site
- > Fluctuating load
 - Sat/Sun 4 servers
 - Mon-Fri 1 server



Agenda

- > Amazon-style cloud computing
- > **Using Amazon EC2**
- > Deploying on Amazon EC2
- > Running the web tier
- > Deploying app servers
- > Deploying a database
- > Handling security
- > Building highly available systems

EC2 API and Tools

- > Amazon provided CLI tools
 - CLI equivalents of APIs
 - AMI creation tools
- > AWS CLI tools from Tim Kay
 - CLI for S3 and EC2
 - Alternatives to Amazon CLI tools
- > AWS Console - very slick
- > ElasticFox
 - Awesome Firefox plugin
 - Launch and manage instances
- > S3 Organizer
 - Firefox plugin
 - Manipulate S3 buckets and objects
- > AWS Eclipse plugin
- > ...

AWS Management Console

Home > Your Account > AWS Management Console BETA Show Navigation

Overview **Amazon EC2** Welcome, Chris Richardson | Sign Out

Navigation

- > EC2 Dashboard
- IMAGES & INSTANCES
 - > **Instances**
 - > AMIs
 - > Bundle Tasks
- ELASTIC BLOCK STORE
 - > Volumes
 - > Snapshots
- CONFIGURATION
 - > Elastic IPs
 - > Key Pairs
 - > Security Groups

My Instances

Launch Instances Reboot Terminate Connect Output Password Bundle Show/Hide Refresh Help

Viewing: All Instances 1 to 1 of 1 Instances

	Instance	AMI ID	Security Groups	Type	Status	Public DNS	Key Pair Name
<input checked="" type="checkbox"/>	i-52ff7f3b	ami-6f2cc906	default	m1.small	running	ec2-67-202-33-45.compute-1.	gsg-keypair

1 EC2 Instance selected

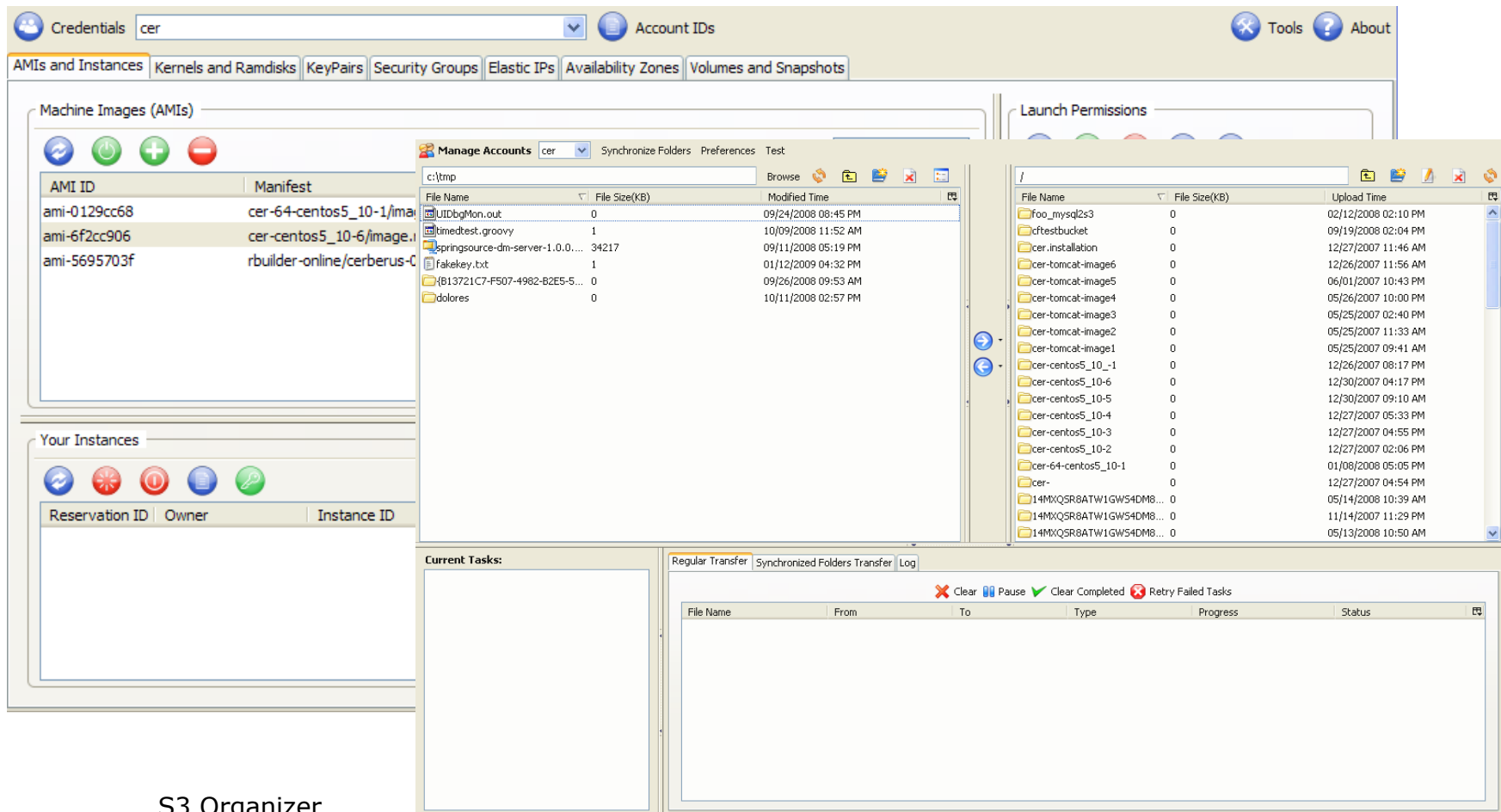
Instance: i-52ff7f3b

AMI ID:	ami-6f2cc906	Zone:	us-east-1a
Security Groups:	default	Type:	m1.small
Status:	running	Owner:	811742389611
Reservation:	r-9f8e3bf6	Ramdisk ID:	-
Platform:	-	Key Pair Name:	gsg-keypair
Kernel ID:	-	AMI Launch Index:	0
Elastic IP:	-		
Public DNS:	ec2-67-202-33-45.compute-1.amazonaws.com		
Private DNS:	domU-12-31-39-03-CC-15.compute-1.internal		
Launch Time:	2009-01-18 15:37 PST		
State Transition Reason:	-		

© 2008, Amazon Web Services LLC or its affiliates. All right reserved. | [Feedback](#) | [Support](#) | [Privacy Policy](#) | [Terms of Use](#) | An  **amazon.com** company

Firefox plugins

ElasticFox



The screenshot displays the S3 Organizer application interface, which is used for managing AWS S3 buckets and files. The interface includes several panels:

- Credentials:** A dropdown menu showing 'cer'.
- Account IDs:** A dropdown menu showing 'cer'.
- Tools:** A button labeled 'Tools'.
- About:** A button labeled 'About'.
- Machine Images (AMIs):** A table listing AMIs with columns for AMI ID and Manifest.
- Manage Accounts:** A panel with a 'Synchronize Folders' button and a 'Test' button.
- Launch Permissions:** A panel with a 'Launch Permissions' button.
- Your Instances:** A panel with a 'Reservation ID' column and an 'Instance ID' column.
- Current Tasks:** A panel with a 'Regular Transfer' button and a 'Synchronized Folders Transfer' button.
- File Transfer:** A panel with a table showing file transfer progress, including columns for File Name, From, To, Type, Progress, and Status.

The 'Machine Images (AMIs)' table contains the following data:

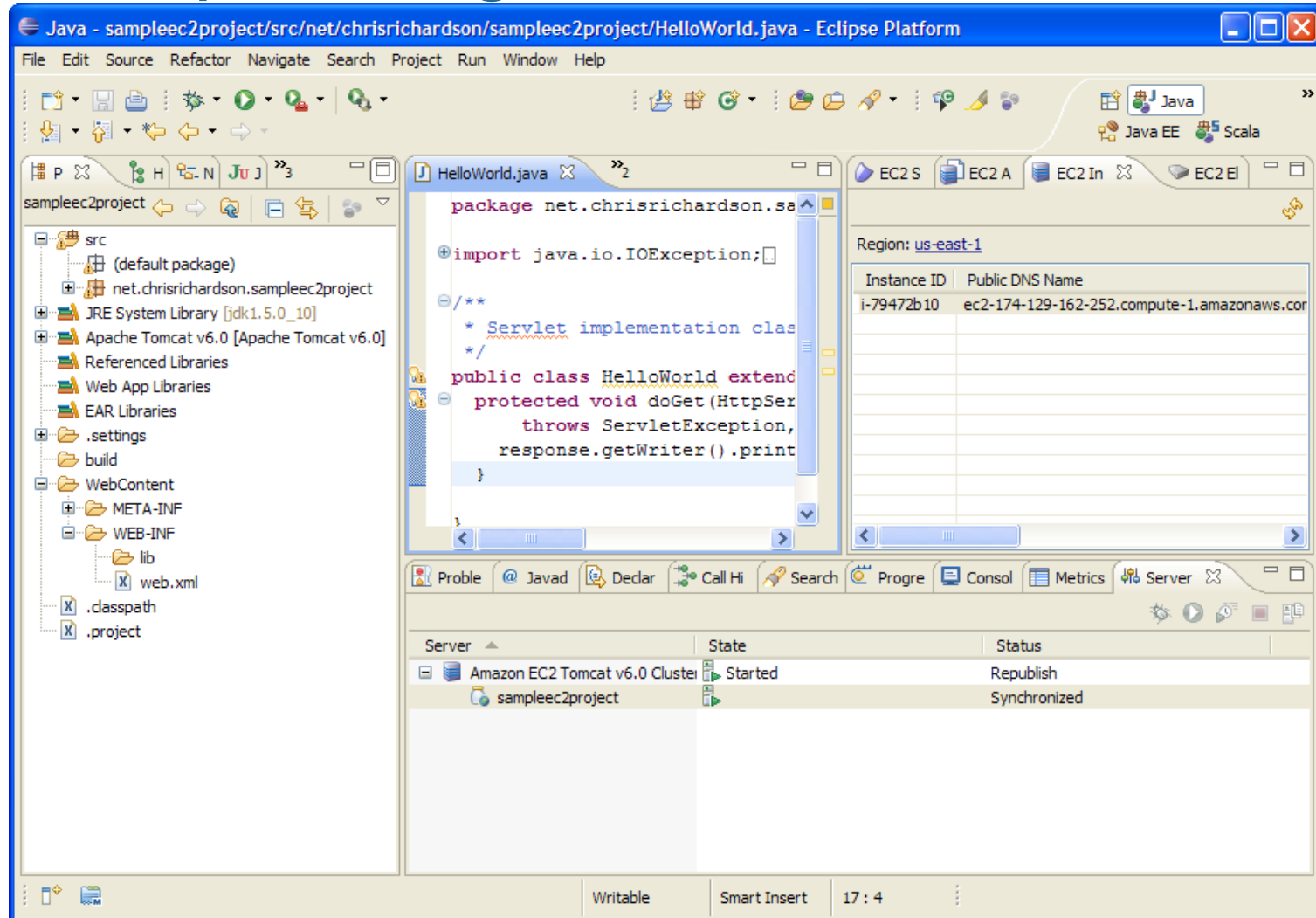
AMI ID	Manifest
ami-0129cc68	cer-64-centos5_10-1/ima
ami-6f2cc906	cer-centos5_10-6/image.i
ami-5695703f	rbuilder-online/cerberus-0

The 'File Transfer' table contains the following data:

File Name	From	To	Type	Progress	Status
foo_mysql2s3					
cftestbucket					
cer-installation					
cer-tomcat-image6					
cer-tomcat-image5					
cer-tomcat-image4					
cer-tomcat-image3					
cer-tomcat-image2					
cer-tomcat-image1					
cer-centos5_10_1					
cer-centos5_10-6					
cer-centos5_10-5					
cer-centos5_10-4					
cer-centos5_10-3					
cer-centos5_10-2					
cer-64-centos5_10-1					
cer-					
14MXQSR8ATW1GWS4DM8...					
14MXQSR8ATW1GWS4DM8...					
14MXQSR8ATW1GWS4DM8...					

S3 Organizer

AWS Eclipse Plugin



Cloud Tools

- > Open-source project
- > 32 and 64 bit AMIs
 - Cent OS 5.10
 - Apache/Tomcat/MySQL/JMeter/JetS3t installed
- > EC2Deploy framework
 - Extensible, object-oriented
 - Launches instances
 - Configures Tomcat, MySQL, Apache
 - Deploys web applications
 - Runs Jmeter tests
 - Written in Groovy
- > Maven and Grails plugins
 - Quick and easy deployment to EC2

- Quicker deployment
- More accurate configuration

Cloud Foundry

**CLOUD
FOUNDRY**

Home Applications **Clusters** Settings

Upload Application [About the Application Environment](#)

Application name:

Web Application

JVM options:
e.g. -Xmx512m -DdbHostName=\${databasePrivateDnsName}

War file:

Context Root:
Path relative to server's root URL

Database

Database name:
Name of the database to create

User id:
Database user's id

Password:
Database user's password

Copyright (c) 2008, Chris Richardson Consulting, Inc
Version: 1.0-SNAPSHOT, 2009/01/12 14:46:15 PST

Home Applications **Clusters** Settings

Cluster Details

Name:

Application:

External IP:

State: LAUNCHED

Hourly cost: \$0.10/hr

Started on: 16-Feb-09 15:10:49

Uptime: 1 hour, 15 minutes

Health: HEALTHY

Applications

Context	Health
myapp	HEALTHY

Home Applications **Clusters** Settings

Launch Cluster

Cost to run this cluster (hourly/monthly): \$0.10 / \$72.00

Name:

Application:

JVM Options:
e.g. -Xmx512m -DdbHostName=\${databasePrivateDnsName}

Topology:

☒ Single instance ☐ Multiple instances
Deploy on a single instance vs. multiple instances

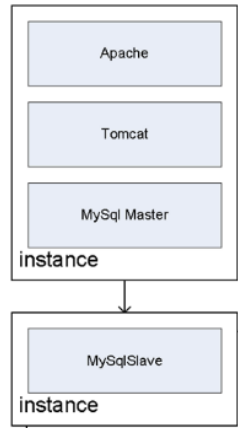
Number of Tomcat Servers:

Database Storage:

Number of MySQL Slaves:

HA MySQL Slave: ☐
Run one MySQL slave in a different availability zone.

Public IP:
The Elastic IP address of the Apache Server



```
graph TD
    Apache[Apache] --> Tomcat[Tomcat]
    Tomcat --> MySQLMaster[MySQL Master]
    MySQLMaster -- instance --> MySQLSlave[MySQL Slave]
    MySQLSlave -- instance --> MySQLSlave
```

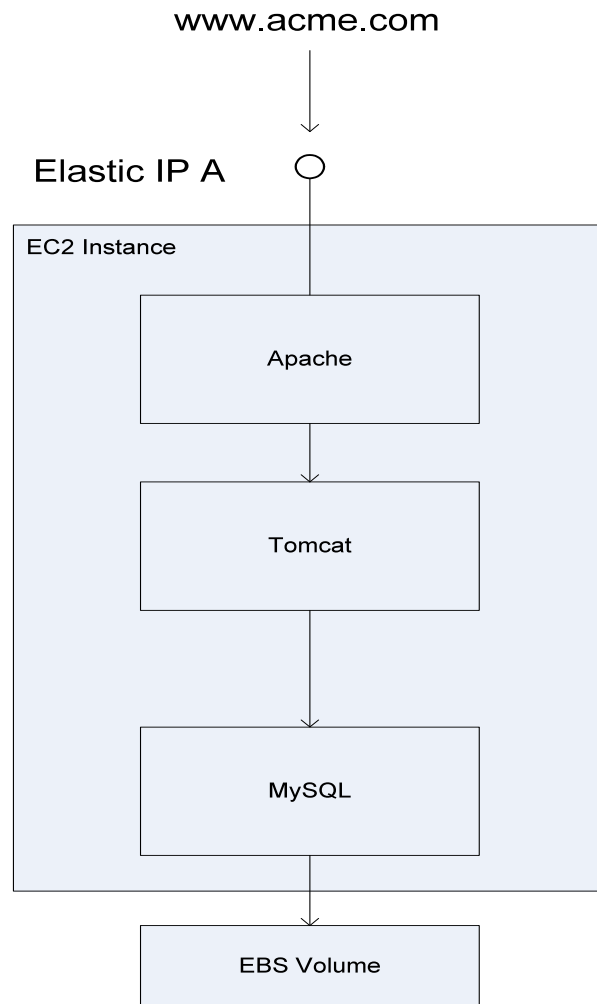
Agenda

- > Amazon-style cloud computing
- > Using Amazon EC2
- > **Deploying on Amazon EC2**
- > Running the web tier
- > Deploying app servers
- > Deploying a database
- > Handling security
- > Building highly available systems

Amazon EC2 supports standard software

```
yum -y install mysql-server  
yum -y install httpd  
yum -y install tomcat5
```

Starter website - \$



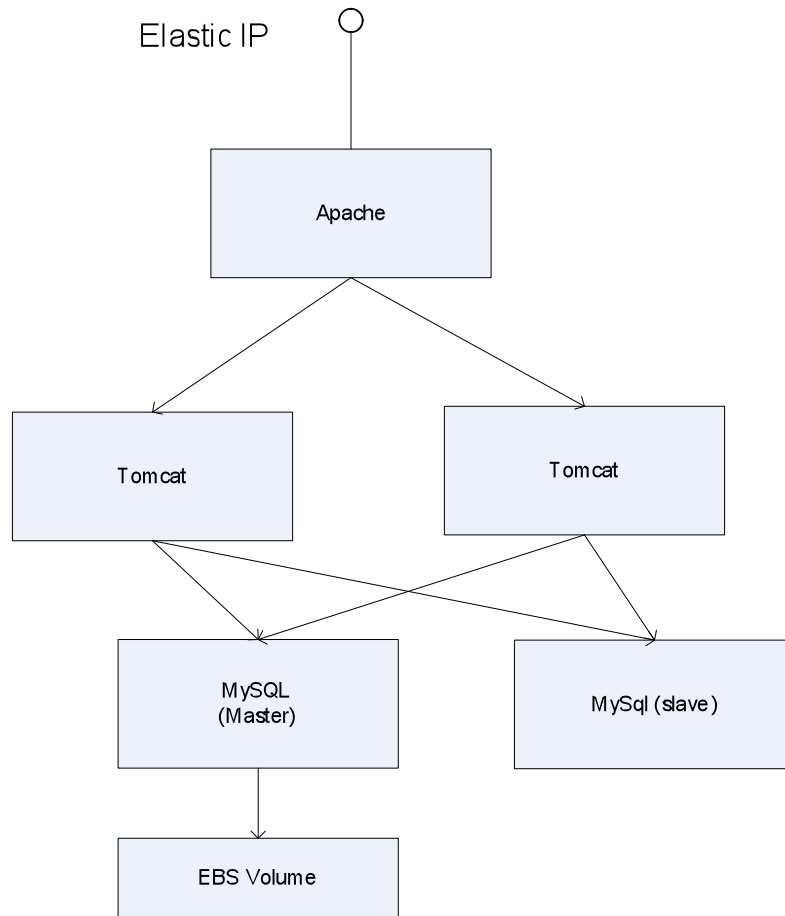
Low cost - \$72/month

Elastic - load increases \Rightarrow expand in a few minutes

Available –instance crashes \Rightarrow replace in a few minutes

Higher capacity website - \$\$

www.acme.com



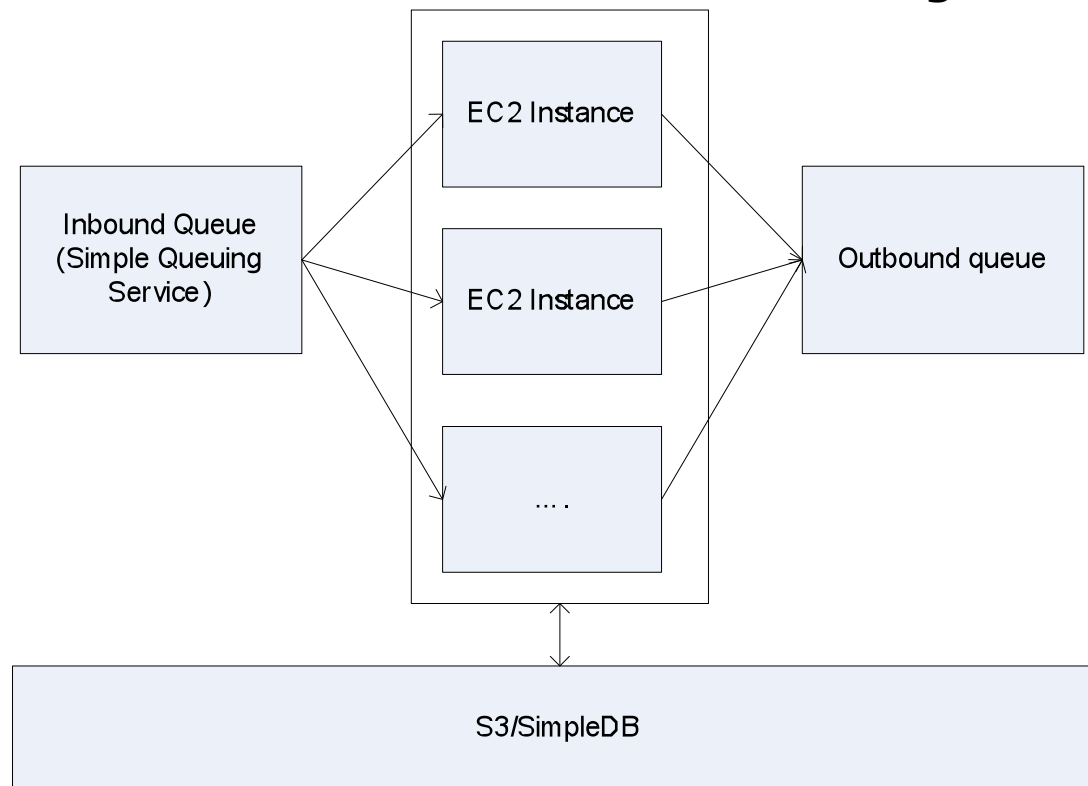
Low cost - > ~\$216/month (1 or more Tomcats, 0 or more Slaves)

Elastic - load changes \Rightarrow quickly expand/subtract Tomcats with no downtime

Available –instance crashes \Rightarrow replace in a few minutes

Batch processing architecture

e.g. media transcoding



Easy upgrades

- > Clone production environment
 - Make read-only or turn off
 - Snapshot EBS volumes and create new volumes
- > Apply upgrades to clone
- > Test clone
- > Move elastic IP addresses to clone
- > Terminate old instances once you are sure that everything works

Issues with AWS

- > Security:
 - Runs HIPAA compliant apps BUT
 - Lack of PCI compliance
 - Discomfort with sending customer data to a 3rd party
- > Technology:
 - Not yet suitable for extremely large relational databases
 - Lack of very large machines, e.g. 64G memory
 - Lack of multicast and multiple IP addresses
- > Financials:
 - Cost of bandwidth
 - Steady state costs > your own hardware

Cloud Computing Survey: IT Leaders See Big Promise, Have Big Security Questions

Greatest Concerns Surrounding Cloud Adoption at Your Company

Security	45%
Integration with existing systems	26%
Loss of control over data	26%
Availability concerns	25%
Performance issues	24%
IT governance issues	19%
Regulatory/compliance concerns	19%
Dissatisfaction with vendor offerings/pricing	12%
Ability to bring systems back in-house	11%
Lack of customization opportunities	11%
Measuring ROI	11%
Not sure	7%
Other	6%

*Respondents selected up to three criteria.

SOURCE: CIO Research

www.cio.com/article/455832/Cloud_Computing_Survey_IT_Leaders_See_Big_Promise_Have_Big_Security_Questions

Cost issues

- > Running larger servers 24 x 7 looks expensive (e.g. \$560/month)
- > Bandwidth can be expensive (\$0.10-0.17/GB)

BUT when owning your own hardware

- > Lack of elasticity
 - Long procurement time
 - Must buy for the estimated peak load
 - Must buy redundant hardware
 - Risk of a success catastrophe
- > Cost
 - Electricity, cooling, space
 - System administration costs
 - Management overhead

Agenda

- > Amazon-style cloud computing
- > Using Amazon EC2
- > Deploying on Amazon EC2
- > **Running the web tier**
- > Deploying app servers
- > Deploying a database
- > Handling security
- > Building highly available systems

Load balancing

- > Amazon-managed service
 - Released in May 2009
 - Load balances across multiple regions
 - Costs \$0.025/hr + \$0.008/Gbyte
 - BUT No sticky sessions

- > Use software load balancer
 - Apache
 - HAProxy
 - ...

Elastic IP addresses

- > Instance IP addresses are dynamically allocated on start-up
 - Does not work well for publicly accessible services, e.g. a website
- > Elastic IP addresses:
 - Statically allocated public IP addresses
 - Associated with your account
 - Attached to an instance (e.g. public facing web server) = it's public IP address
 - You configure DNS to resolve to the elastic IP address
- > Pricing:
 - Non-attached Elastic IP address - \$0.01/hour
 - \$0.10 per remap (if > 100 in a month)

Agenda

- > Amazon-style cloud computing
- > Using Amazon EC2
- > Deploying on Amazon EC2
- > Running the web tier
- > **Deploying app servers**
- > Deploying a database
- > Handling security
- > Building highly available systems

No multicast for resource discovery

- > Prevents the use of standard clustered resource discovery
 - E.g. JGroups multicast etc
- > Use a registry:
 - JGroups with TCP
 - Database
 - SimpleDB
 - Security groups
 - ...
- > Use Terracotta to cluster Tomcat

Elastic Autoscaling

- > Released in May 2009
- > Autoscaling group: Min/max # of instances
- > Autoscaling trigger
 - Specifies when to scale up/down autoscale group
 - e.g. CPU load
- > Use for:
 - Scaling up/down based on load
 - Automatically restarting failed instances
- > **BUT** there is still the problem of discovery

Agenda

- > Amazon-style cloud computing
- > Using Amazon EC2
- > Deploying on Amazon EC2
- > Running the web tier
- > Deploying app servers
- > **Deploying a database**
- > Handling security
- > High availability

The trouble with local storage

- > It's plentiful

BUT

- > Local storage is ephemeral
- > First write performance penalty

Elastic Block Storage

- > Mountable storage volumes
 - "On-demand SAN"
 - Size: 1 GB to 1 TB
 - Mount on a single instance
- > Create snapshots
 - Stored in S3
 - Create new volumes from the snapshot
- > Cost:
 - \$0.10/GByte/month
 - \$0.10 per 1 million I/O requests

Using EBS Volumes

AWS:

CreateVolume Size=50G

AttachVolume InstanceId=... Device=/dev/sdh

```
mkfs.xfs /dev/sdh
```

```
echo "/dev/sdh /vol xfs noatime 0 0" >> /etc/fstab
```

```
mkdir /vol
```

```
mount /vol
```

```
mkdir /vol/lib /vol/log
```

```
mv /var/lib/mysql /vol/lib
```

```
[mysql.server]
```

```
user=mysql
```

```
basedir=/vol/lib
```

Backing up your database

```
mysqldump --add-drop-database --databases foo | gzip > backup.sql.gz  
now=`date +%d%m%y_%H%M`  
aws put $bucket/${object}_${now}.sql.gz backup.sql.gz  
aws copy $bucket/${object}_latest $bucket/${object}_${now}.sql.gz
```

FLUSH TABLES WITH READ LOCK
SHOW MASTER STATUS

```
xfs_freeze -f /vol
```

```
# AWS WS: CreateSnapshot
```

```
xfs_freeze -u /vol
```

UNLOCK TABLES

Agenda

- > Amazon-style cloud computing
- > Using Amazon EC2
- > Deploying on Amazon EC2
- > Running the web tier
- > Deploying app servers
- > Deploying a database
- > **Handling security**
- > **Building highly available systems**

Security benefits of cloud computing

- > Leverages the security expertise of amazon.com
- > Cloud infrastructure enables:
 - Unlimited logging
 - Ability to test changes on a clone
 - Clone servers and volumes for forensic analysis

The usual security best practices

- > Turn off unused services
- > File ownership and permissions
- > Disabling password based ssh login
- > Standard Linux, Apache, Tomcat and MySQL best practices

Network security

- > Cannot sniff traffic for other instances
- > Consider encrypting network traffic
- > Limit SSH access to only your location
- > Use EC2 firewall – aka. security groups

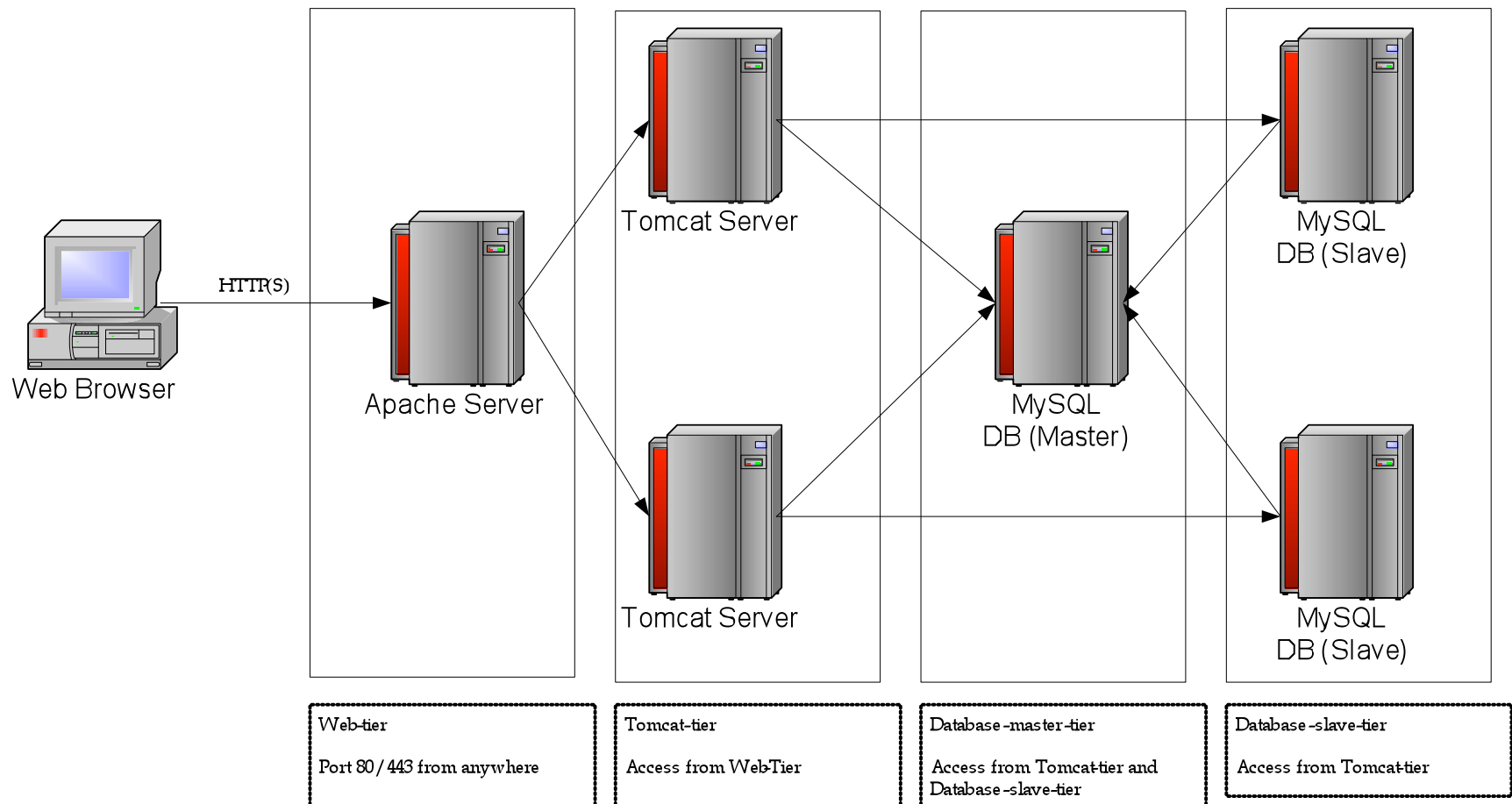
Security Groups

- > Named set of firewall rules associated with your account
- > An instance
 - Belongs to one or more security groups
 - Defaults to “default” security group
- > Permits inbound traffic
 - Protocol: tcp, udp
 - Range of ports
- > From:
 - Anywhere – specific port range
 - An IP address (range) – specific port range
 - Another group - **all ports**
- > Common usage
 - Port 80 (http)/443 (https) – anywhere
 - Port 22 (ssh) – just from your location

```
?Action=RunInstances  
&SecurityGroup.1=g1  
&SecurityGroup.2=g2
```

When you first
signup don't forget
to enable SSH traffic

Using security groups



Use a software firewall

- > E.g. iptables
- > In addition to security groups
 - Security Group: Tomcat Servers are only accessible from Apache Server
 - iptables: Tomcat servers only allow port 22 and port 8009 (AJP)

Storage security

- > Amazon wipes disks so one customer cannot see another's data
- > **But**
 - You don't know where it is
 - Amazon could be subpoena'd
- > Consider encrypting data
 - Encrypted file systems
 - Encrypting sensitive data in DB
 - Encrypting backups in S3

Agenda

- > Amazon-style cloud computing
- > Using Amazon EC2
- > Deploying on Amazon EC2
- > Running the web tier
- > Deploying app servers
- > Deploying a database
- > Handling security
- > **Building highly available systems**

Deploying highly available applications

- > AWS has had very well publicized outages

BUT...

- > Is internal IT really any better?
- > In reality: AWS is (more) reliable
- > Don't forget:
 - You are not responsible for the hardware
 - Instance fails \Rightarrow Launch a new one in a few minutes

But once in a blue moon

Hello,

We have noticed that one or more of your instances are running on a host degraded due to hardware failure.

i-14d00b7d

The host needs to undergo maintenance and will be taken down at 12:00 GMT on 2009-04-03. Your instances will be terminated at this point.

The risk of your instances failing is increased at this point. We cannot determine the health of any applications running on the instances. We recommend that you launch replacement instances and start migrating to them.

Feel free to terminate the instances with the `ec2-terminate-instance` API when you are done with them.

Let us know if you have any questions.

Sincerely,

The Amazon EC2 Team

Lack of virtual IP addresses

- > One IP address for use in the cloud
- > Using Elastic IP = \$0.01 per GB
- > Handling active/standby failover is difficult:
 - E.g. Cannot migrate IP address of failed database to standby database
- > Have your own host names
 - Update /etc/hosts
 - Run DNS server

Regions and availability zones

- > By default, your database master and slave could run on the same physical host!
- > Regions - geographically dispersed locations
- > Availability zone - engineered to be insulated from failure in other zones
- > Specify availability zone when launching instances
- > SLA with 99.95% availability with multiple availability zones
- > Inter-zone network traffic = \$0.01 per GB

us-east-1

us-east-1a,

us-east-1b

us-east-1c

eu-west-1

eu-west-1a

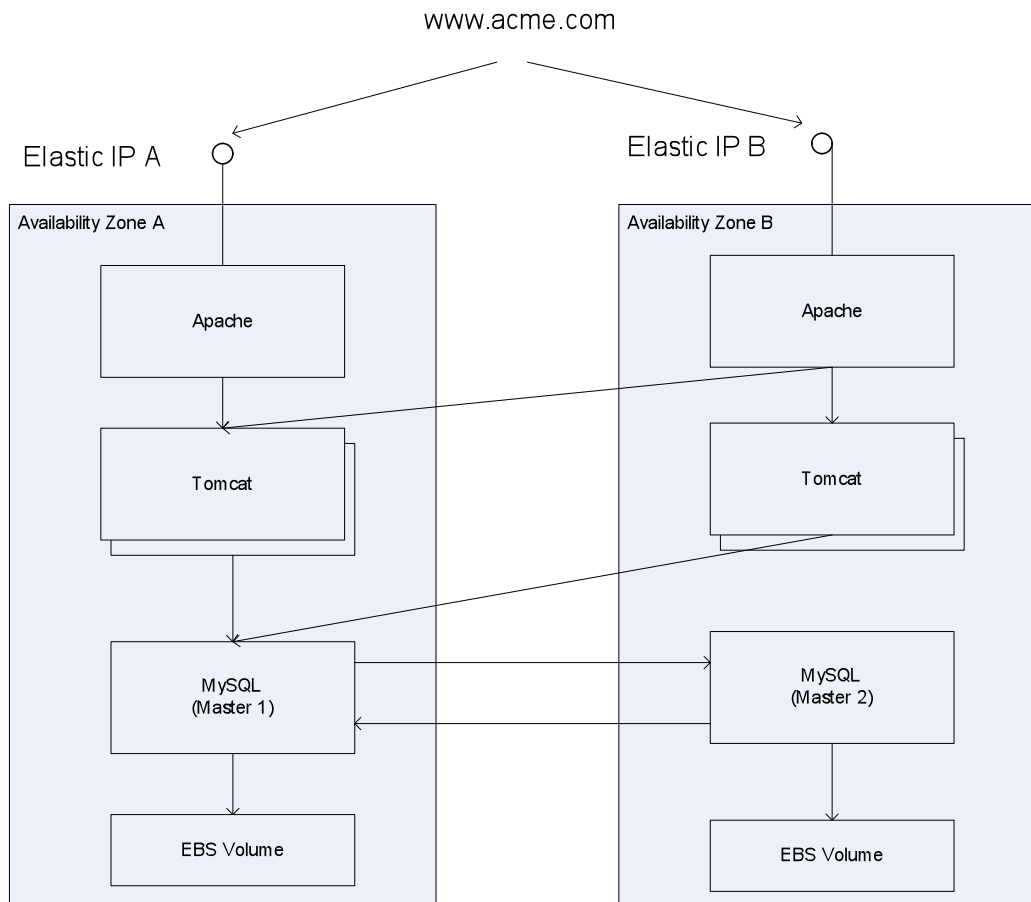
eu-west-1b

Amazon EC2 SLA*

- > 99.95% availability if you are using >1 availability zone
- > Availability
 - Instances have external connectivity
 - You can launch new instances
- > Service credit for not meeting SLA

* Read the small print

Highly available - \$\$\$



Higher cost - > ~\$ 360/month (2 Apaches, 2 MySQLs, 1 or more Tomcats, 0 or more Slaves)

Elastic - load changes \Rightarrow quickly expand/subtract Tomcats with no downtime

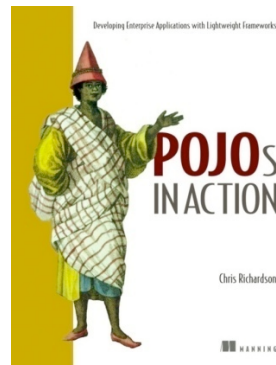
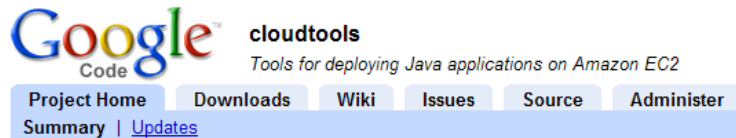
Available – No SPOF, instance crashes \Rightarrow replace in a few minutes

Summary

Amazon-style cloud computing provides

- Immediate access to a scalable infrastructure
- Pay as you go – no upfront investment/commitment required
- Easily scale up/down
- Optional AWS services

Final thoughts



Download or contribute to Cloud Tools today :

www.cloudtools.org

Checkout Cloud Foundry:

www.cloudfoundry.com

Buy my book ☺

Send email:

chris@chrisrichardson.net

Visit my website:

www.chrisrichardson.net

Talk to me about consulting and training

Phone: 510 904 9832



JavaOneSM

Thank You

Chris Richardson

Chris Richardson Consulting, Inc

www.chrisrichardson.net

chris@chrisrichardson.net / @crichardson

