



Amazon S3 + EC2 + Rails = Dream Team?

Thoughts by
Adam Groves and Martin Rehfeld
@ BRUG 05-Apr-2007

Amazon S3

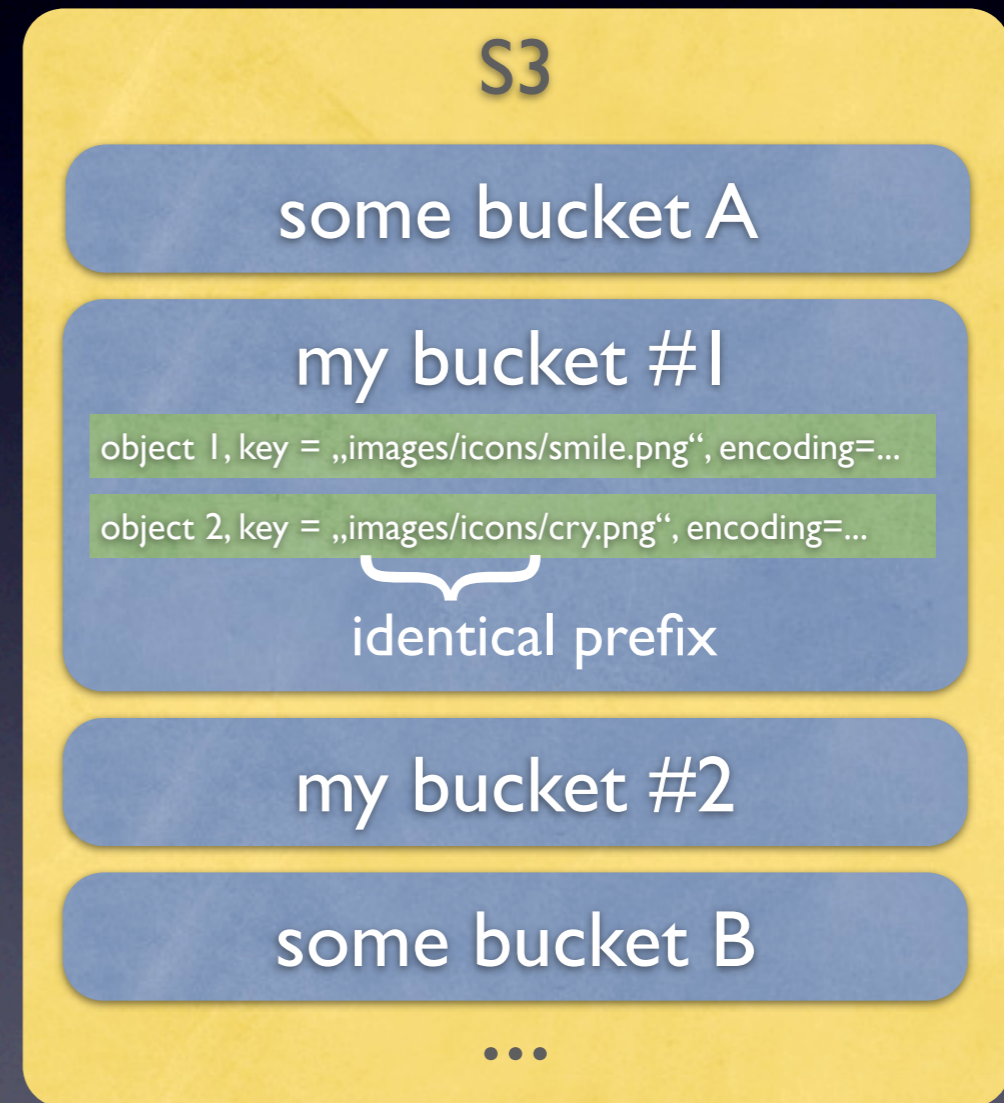
Simple Storage Service

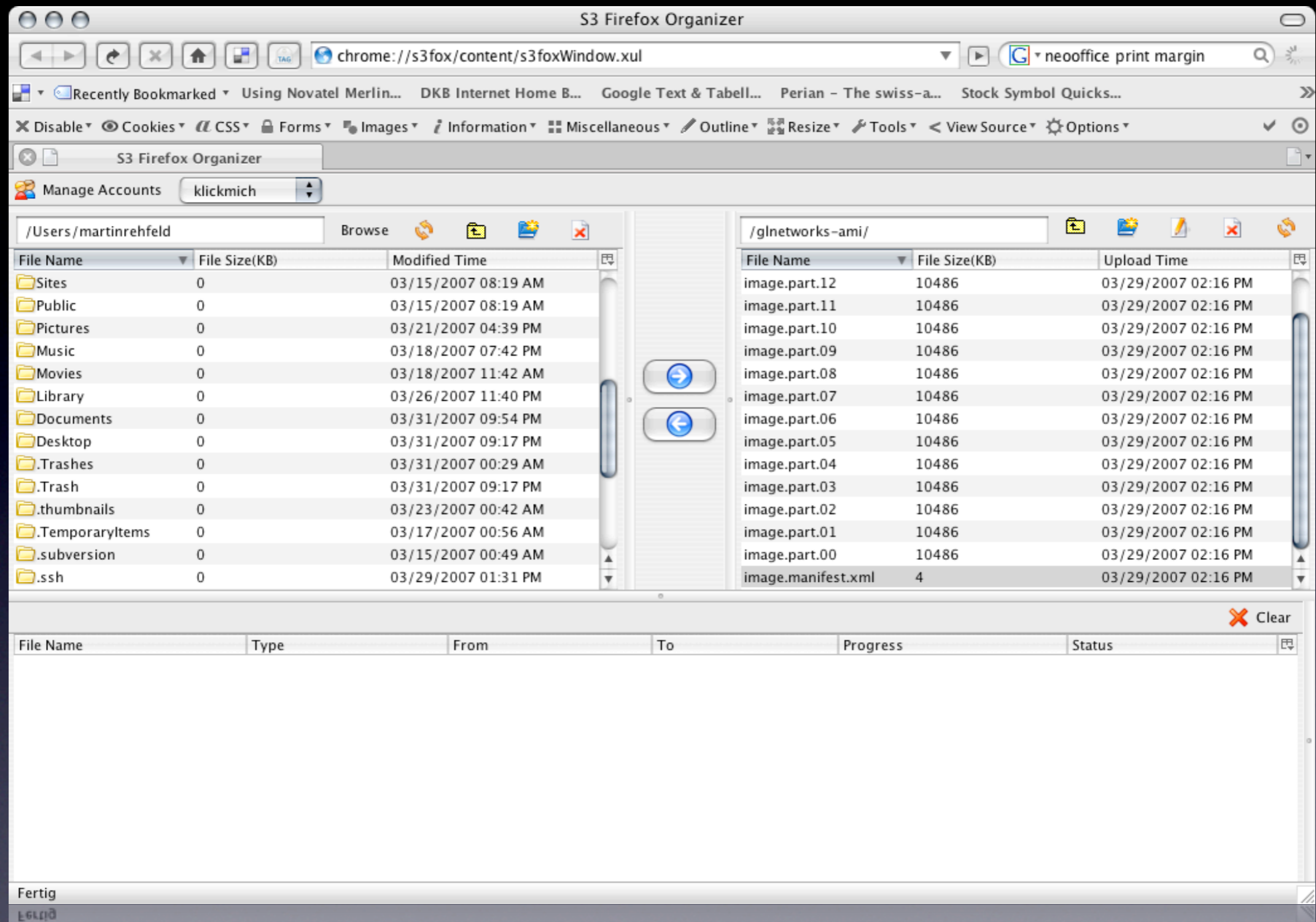
- Pricing
 - \$0.15 per GB-Month
(10 GB = \$1.50 per month) 
 - \$0.20 per GB of data transferred
(100 GB = additional \$20 per month) 

Amazon S3

Common Terms

- buckets:
global name space
- objects:
accessed by key [~ path+name]
can have metadata
- prefixes:
searching by prefix emulates
directory structure
- ACL:
everyone, authenticated users,
owner / named users





S3Fox Extension

Amazon EC2

Elastic Compute Cloud

- Pricing

- \$0.10 per instance-hour consumed
(1 month 24x7 ~ \$72)



- \$0.20 per GB of data transferred
(just internet traffic;
no charge for EC2-S3 traffic)



Amazon EC2 Instance Specs

- ~ 1.7 GHz x86 CPU
- 1.75 GB RAM
- 160 GB local disk space
- 250 Mb/s network bandwidth

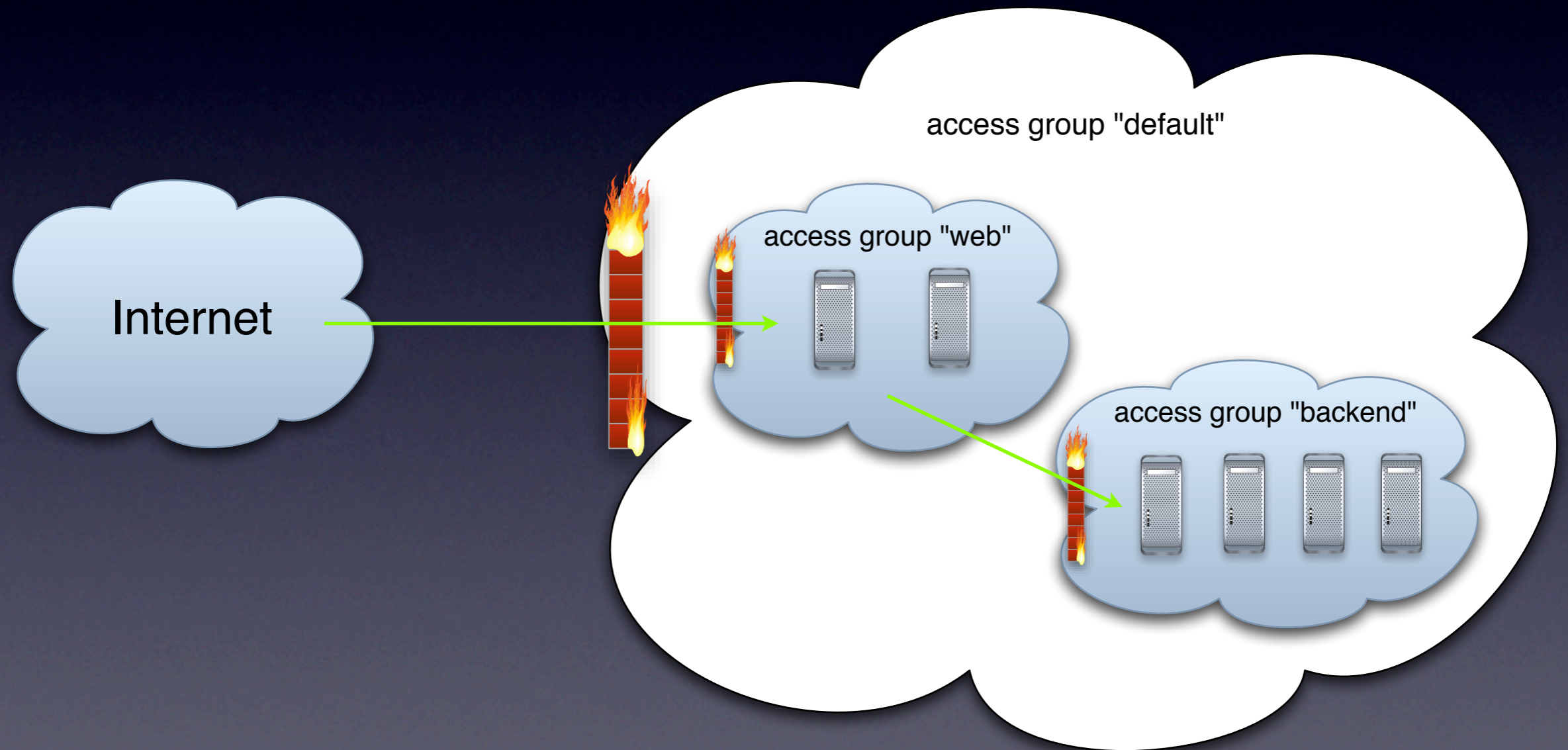


Amazon EC2 Common Terms

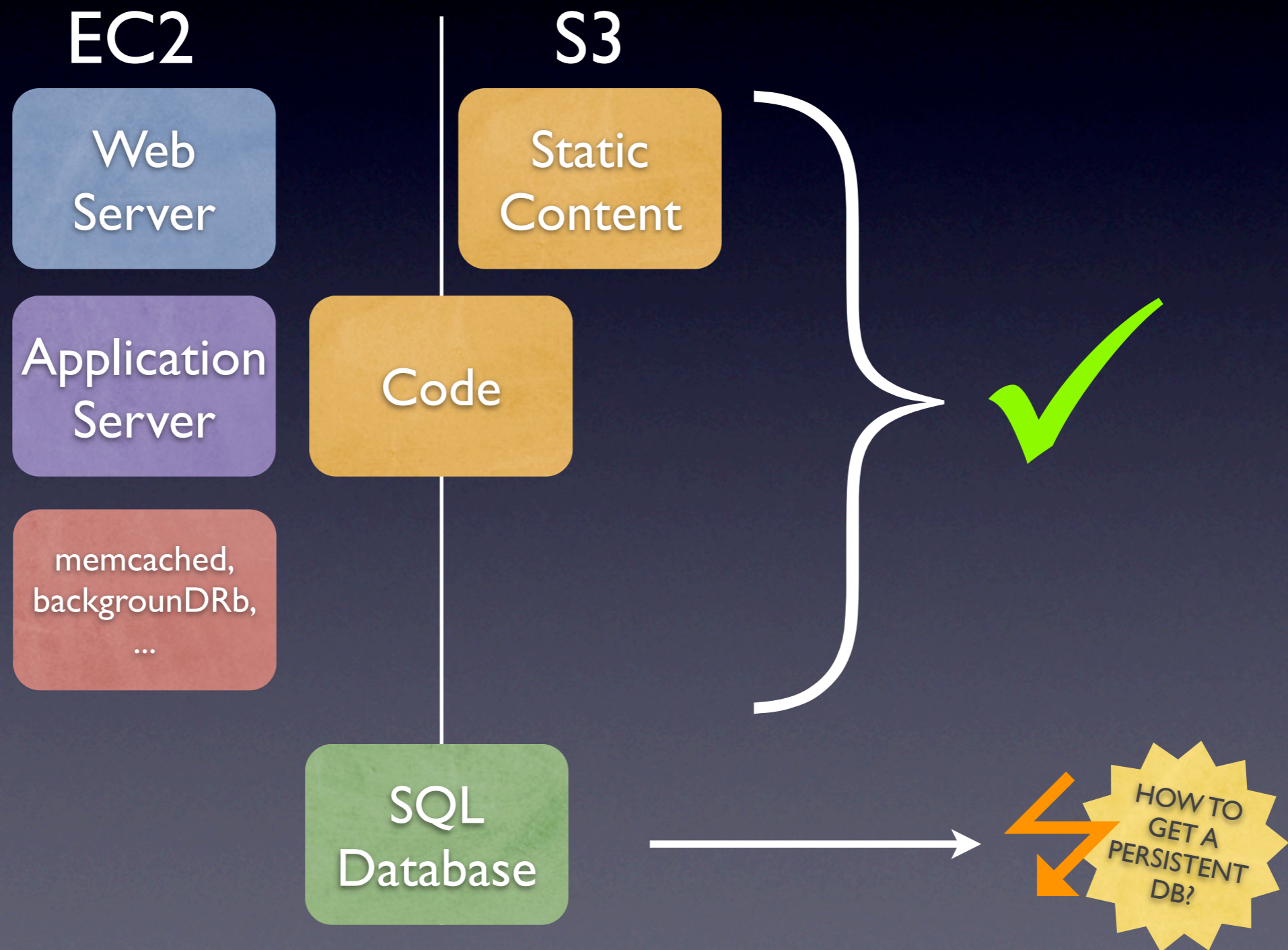
- images:
named OS images stored in S3: „AMI“
- instances:
virtual machines running an AMI
- bundling:
saving customized images back to S3
- non-persistence:
local disk storage will not survive instance
shutdown or failure



Amazon EC2 Network Security



Amazon EC2 Rails Scenario

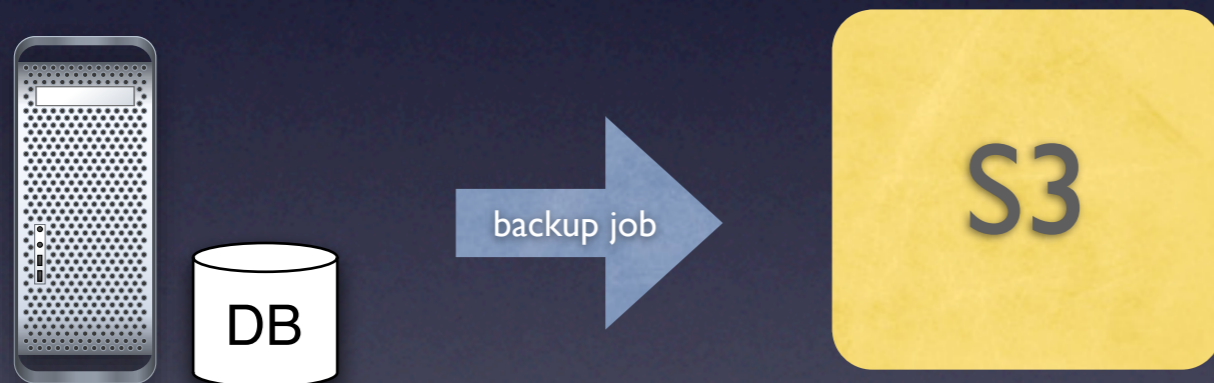


Amazon EC2

Database Persistence

Flavor A:

Frequent backup to S3



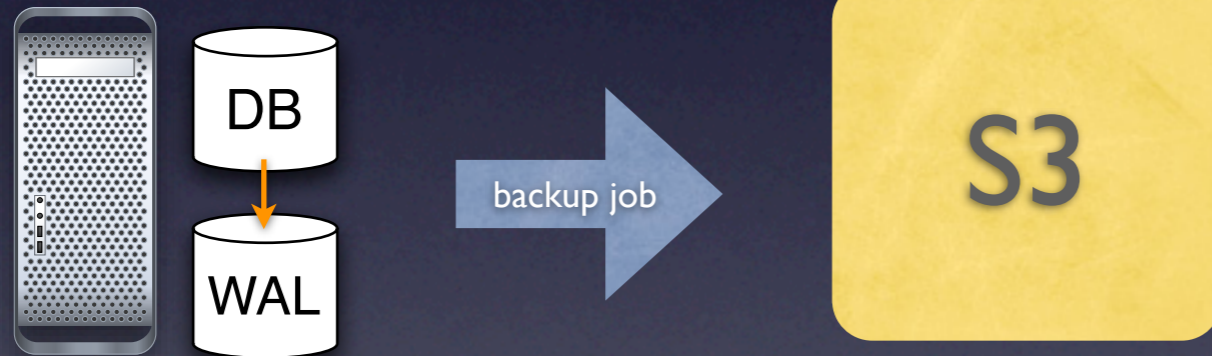
- on instance failure you will lose all changes since last backup
- backup using a lot of resources
- no automatic failover

Amazon EC2

Database Persistence

Flavor B:

Backup write ahead logs to S3

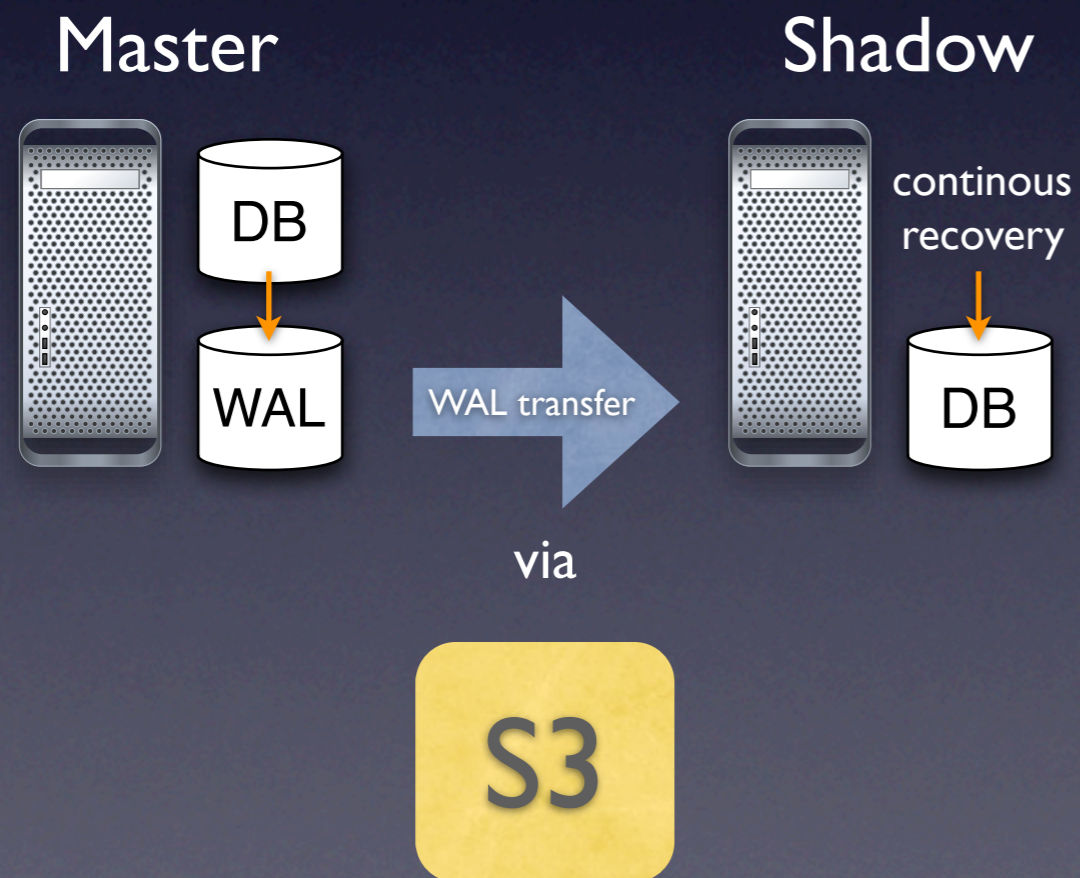


- on instance failure you will lose all changes since last log switch
- backup is fast, but recovery will take longer
- still no automatic failover

Amazon EC2

Database Persistence

Flavor C: Shadow database



- you might still lose all changes since last log switch
- backup is fast, recovery is usually not necessary (only if master and slave should fail at the same time)
- no load balancing between master and slave
- automatic failover can be achieved

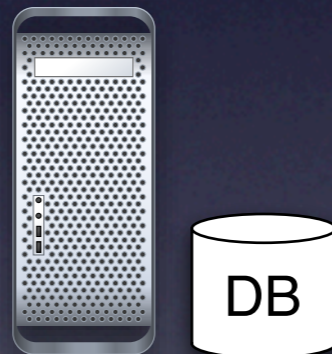
Amazon EC2 Database Persistence

Flavor D: (Multi-Master-)Replication

Master #1



Master #2



- data loss only on failure of all instances
- reduced write performance
- sophisticated configuration
→ can be tricky
- should be combined with flavor A or B for backup
- load balancing can be achieved
- automatic failover can be achieved

Amazon EC2

Database Persistence

Flavor E (highly Experimental):
DB files on S3DFS



- unless caching is used there should be no data loss
- (very?) reduced performance
- reliability of S3DFS yet to be proven
- can be combined with flavors A or B for backup
- can be combined with flavors C or D for failover and load balancing (flavor D only)

Resources

- Amazon EC2 / S3

aws.amazon.com/ec2 / aws.amazon.com/s3

- Rails & EC2 & Capistrano tasks

<http://niblets.wordpress.com/2007/02/12/capistrano-ec2-sitting-in-a-tree-k-i-s-s-i-n-g/>

- Amazon Web Services Ruby Gem (aws/s3)

<http://amazon.rubyforge.org/>

- S3DFS filesystem driver

<http://www.openfount.com/blog/s3dfs-for-ec2>

- Discussion on persistence with EC2

<http://blog.awswebshop.com/2007/01/07/thoughts-on-current-functionality-requests-for-future-functionality/>

- Postgres WAL / continuous archiving

<http://www.postgresql.org/docs/8.2/static/wal-intro.html> / <http://www.postgresql.org/docs/8.2/static/continuous-archiving.html>

- MySQL multi-master replication

<http://captofu.livejournal.com/1752.html>

Q & A



Adam Groves
adam.groves@gmail.com

Martin Rehfeld
martin.rehfeld@glnetworks.de