High Performance MySQL Choices in Amazon Web Services: Beyond RDS

Andrew Shieh, SmugMug Operations
shandrew @ smugmug.com

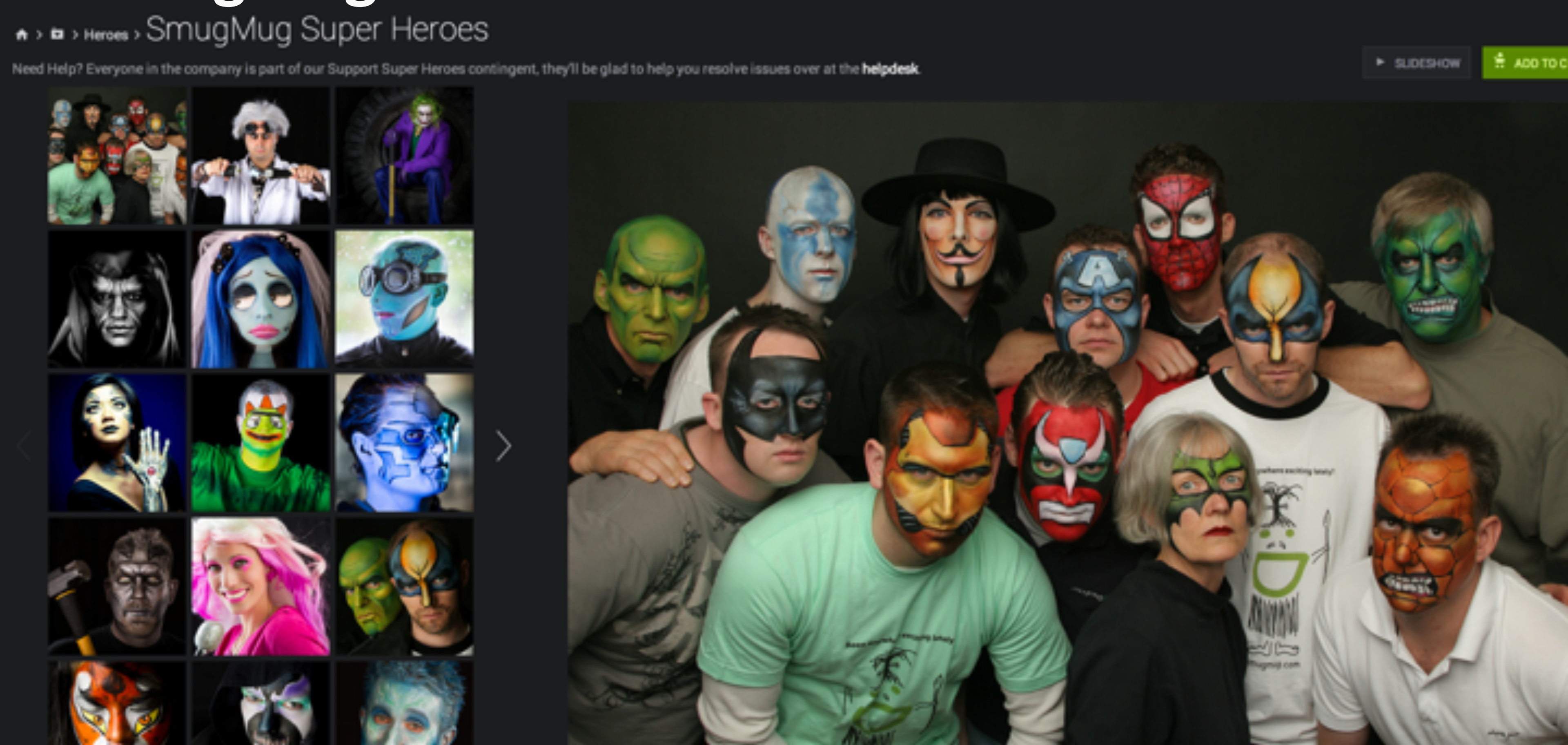
April 15, 2015

MySQL Worldwide Conference & Expo
PERCONA LIVE

Agenda

- All about AWS
- Current RDS alternatives
- Cloud failures -> Cloud improvements
- Migrating MySQL into AWS
- MySQL backups in AWS
- The future

SmugMug—Who are we?



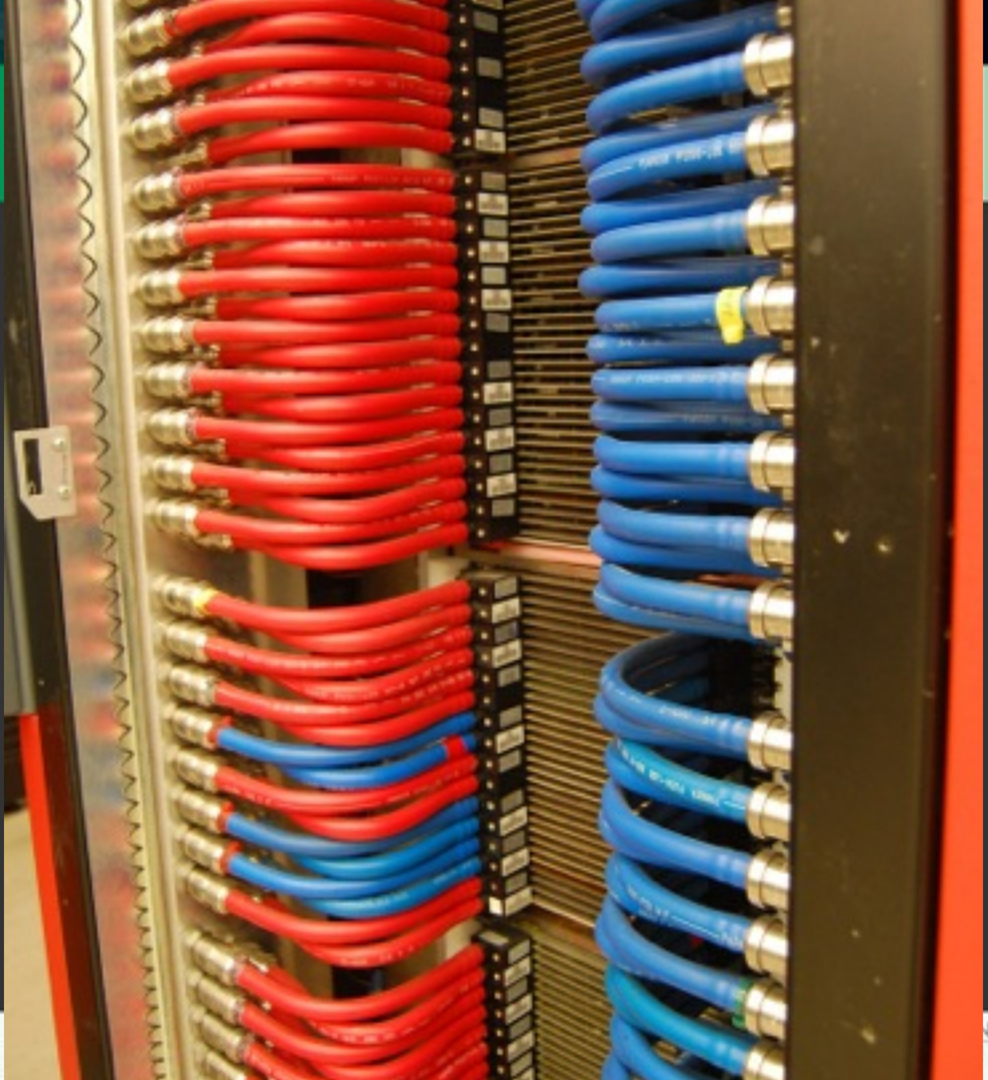
Heroes › SmugMug Super Heroes

Need Help? Everyone in the company is part of our Support Super Heroes contingent, they'll be glad to help you resolve issues over at the **helpdesk**.

The early days of SmugMug

- Gradual bootstrapped growth
- Multiple self-managed datacenter cages
- Too many servers of varying types
- Too many disks
- Tons of valuable skilled employee hours spent in cages

Data Center Fantasy



Data
Center
Reality

SmugMug <3 AWS

- Early adopter of Amazon S3
- Over the years, moved rendering, upload, archiving, payments, permissions, email, and more compute to AWS

- But DBs? Before mid-2012, no ultra-high performance I/O in AWS
- In 2013, we went 100% AWS

All About AWS

- AWS Customer Focus - amazing support
- Features
- Tech
- Cloud is not easy, and is far from a commodity!

RDS? If it works great for you, great!

- In our 2012 evaluation, it lacked scalability and features that we desired
- Loss of SUPER, shell access
- no local SSD
- We already knew how to run MySQL effectively

RDS Alternatives

- Run your database hardware yourself,
- Connect via AWS Direct Connect (peering to AWS).

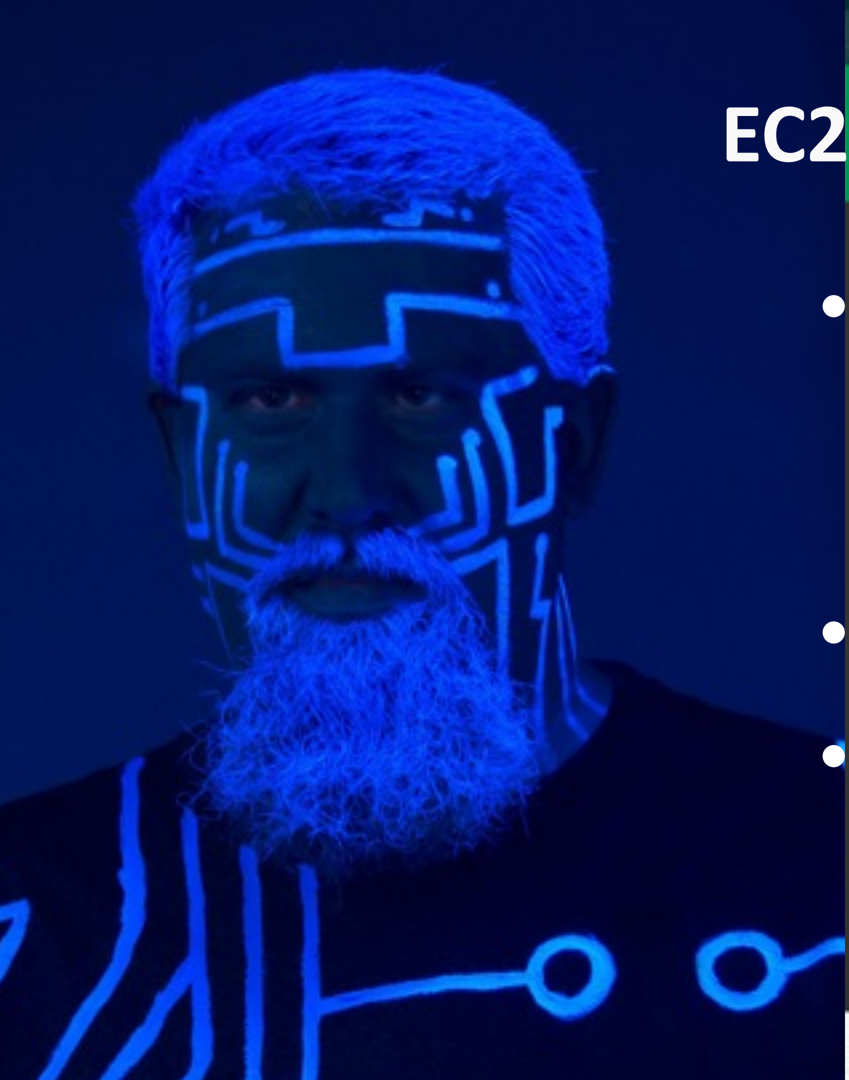
RDS Alternatives

- No! The hybrid cloud is a trap that will come back to bite you
- High I/O hardware, complexity goes up, costs can skyrocket

Data Center Reality

Use Amazon EC2

- Pre-EC2, our custom, obscure SSD hardware => difficult to resolve problems, difficult to upgrade
    - TRIM through FS, LVM, MD, driver layers: nightmare
    - Micron/Crucial 5000 hour bug
- hi1 overall DB IO performance comparable to 8 x SSD RAID10
- hi1 < 3%/yr instance failure rate! Many small important SSD details solved for you



EC2 i2.* instances: even better

- 1/2/4/8x sizes (up to 16 core, 244 GB RAM, 6.4 TB SSD)
- Better everything/$ vs hi1
- Slightly higher failure rate, ~5%/yr

Two faces of EBS

- Since 2008, AWS' primary persistent storage
- Prior to 2013, large source of the major AWS outages: April 2011, October 2012, December 2012. Regional failures caused by cascading EBS failures
- Caused EBS-dependent services to fail as well
- So, we avoided it for production

Two faces of EBS

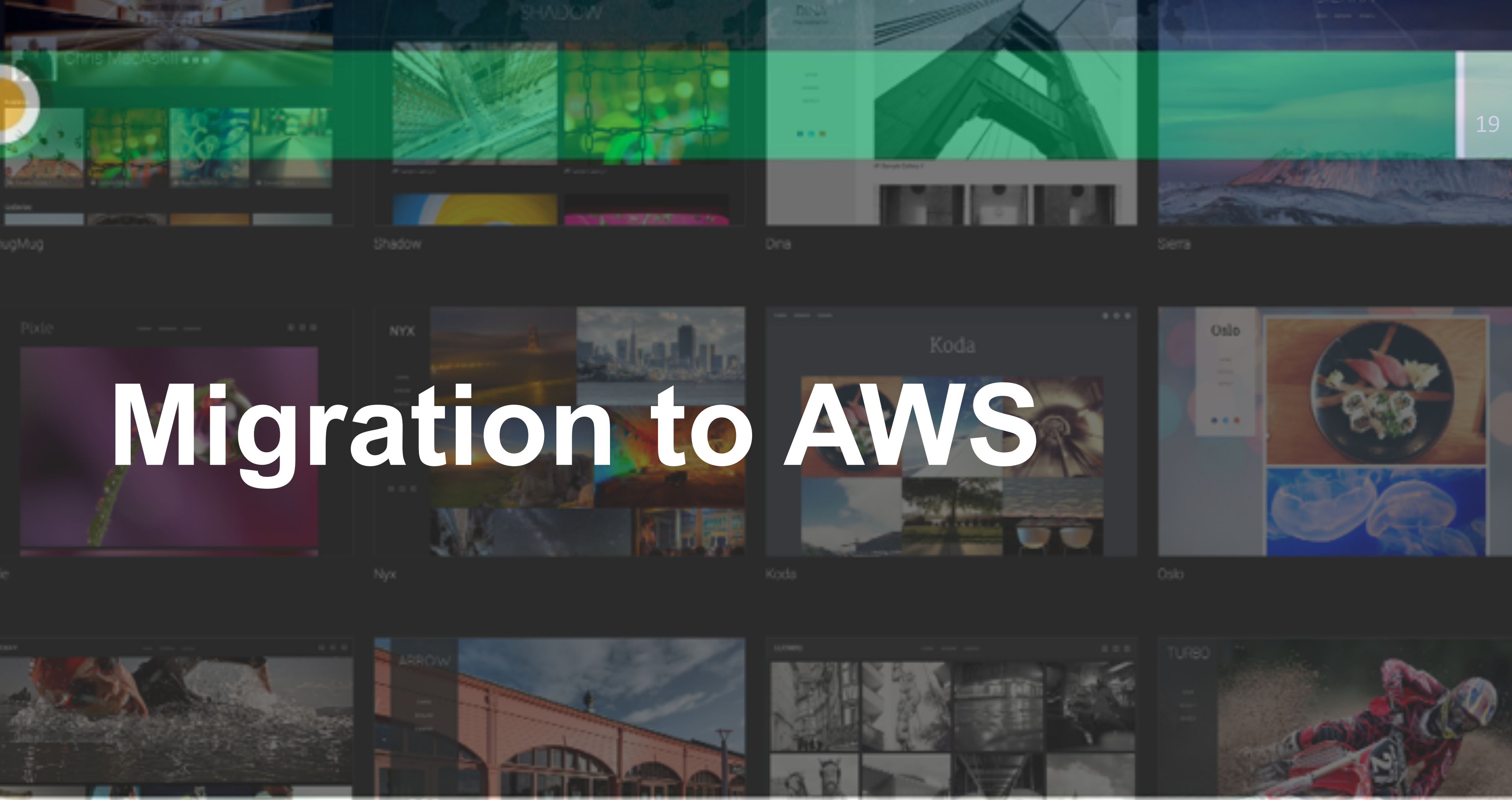
- But AWS has learned and continuously improved
- Limits on amount of reprovisioning when there's a failure
- Avoiding multi-AZ outages is key. Separate AZs as much as possible: independent power, networking, cooling, pushes

Two faces of EBS

- Mid-2012 EBS provisioned-IOPS: EBS on SSDs
- Late-2014 EBS gp2: EBS on SSDs now standard
- "EBS-optimized" lower latency high bw connections from EC2 to EBS at small cost
- 2015 EBS: 16 TB volumes

Use Amazon EC2+EBS?

- Improved EBS reliability/ technology=>greater use of it in production
- Cost advantages, select compute separately from SSD size
- Particularly useful for replicas and backups



Migration to AWS

SmugMug Architecture ~2006

AWS: S3

SV: Web, DB, Image*

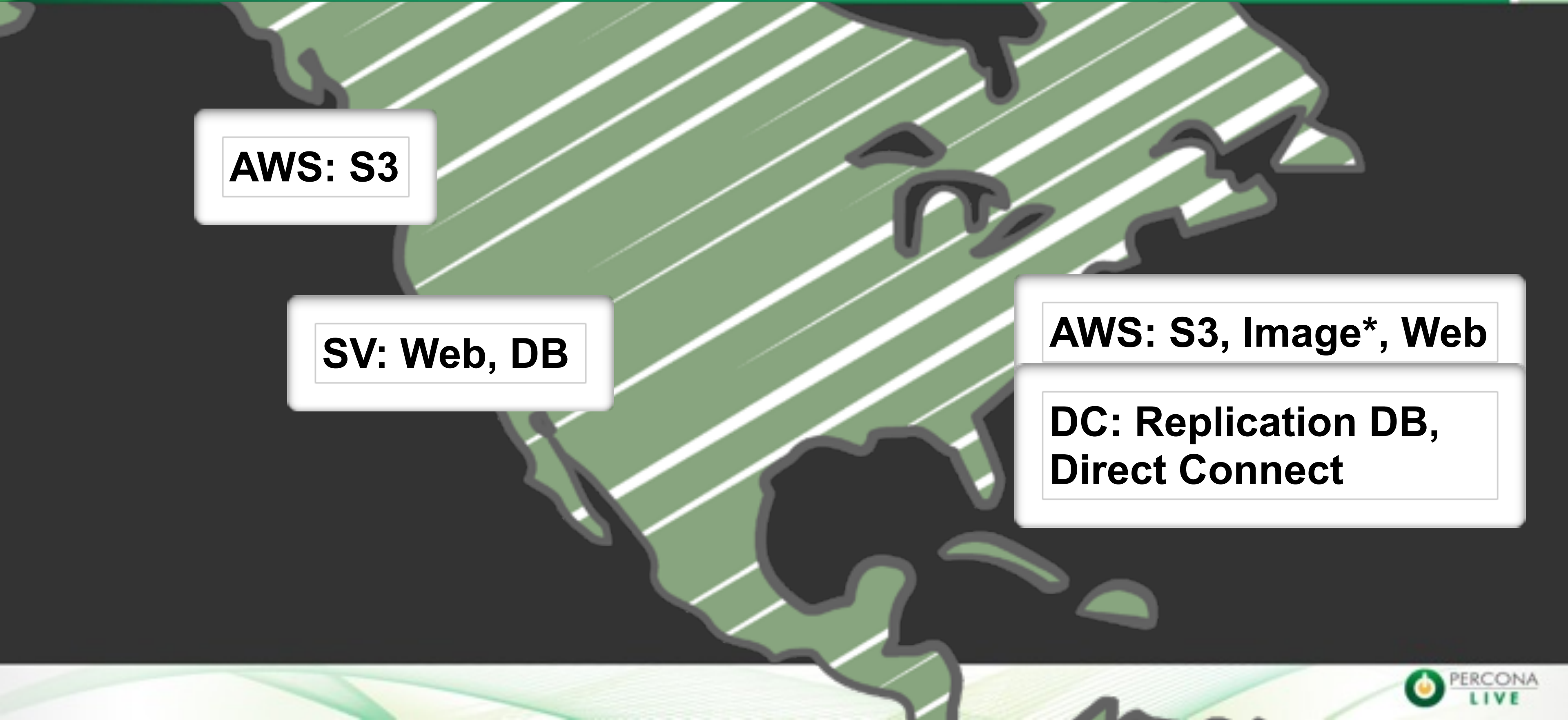
SmugMug Architecture ~2011

AWS: S3

SV: Web, DB

AWS: S3, Image (upload, processing, render, video, …)

SmugMug Architecture - Transition



AWS: S3

SV: Web, DB

AWS: S3, Image*, Web

DC: Replication DB,
Direct Connect

SmugMug Architecture Today

Ø

AWS: S3, Image*, Web, DB

Zero Downtime Move Requirements

- Read-only site mode
- Traffic control — shadow load
- Cross country MySQL replication + sufficient bandwidth
- Make sure replicas are caught up, go read only, then switch masters to your AWS MySQL instances

“High Availability” Solution for conventional MySQL

- Percona MySQL 5.5/5.6: Stable and well tested
- Other HA solutions address SPOF of conventional MySQL master replication
- But newer software may be less tested, more buggy. Bugs = SPOF

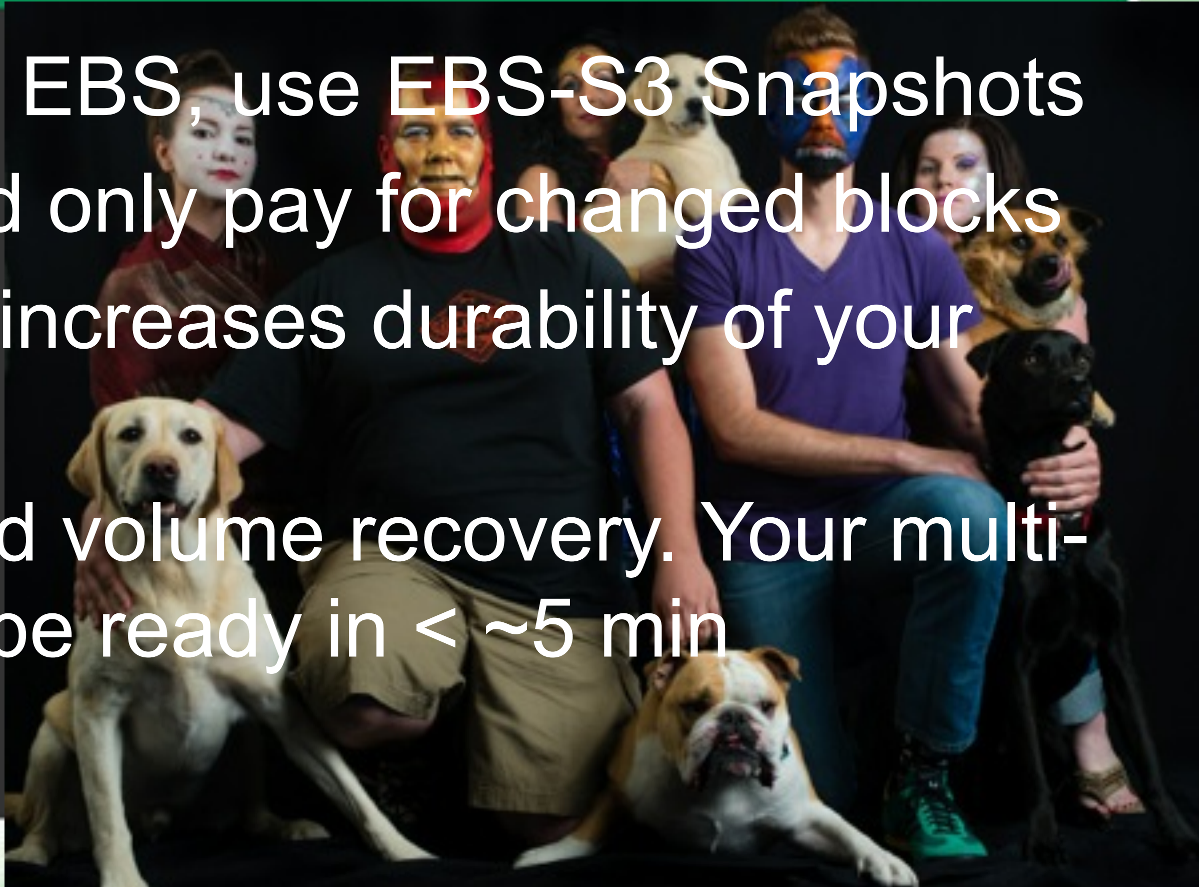
“High Availability” Solution for conventional MySQL

- Reduce failure exposure by distributing replicas in separate AZs, masters in one AZ
- Minimize master failure impact by switching to a read-only mode
- Recover quickly by reducing time needed to swap in new master

Controlling AWS costs

- Reserved instances, 1/3 year commit discount rates
- Monitor usage always
- Talk to your account manager

MySQL Backups in AWS



- With replicas on EBS, use EBS-S3 Snapshots
- Simple, fast, and only pay for changed blocks
- Snapshots also increases durability of your volumes
- Fast, lazy-loaded volume recovery. Your multi-TB backup can be ready in < ~5 min

MySQL Backups in AWS

- Percona xtrabackup
- Consistent, reliable, easy to manage
- Runs on production DB with minimal impact
- Choose your favorite storage method for short-term/long-term

The Future: PXC / Galera

- High durability
- Masterless, easy architecture
- Easy to migrate in/out with replication
- But little track record
- And performance penalties, esp cross-datacenter

Amazon Aurora

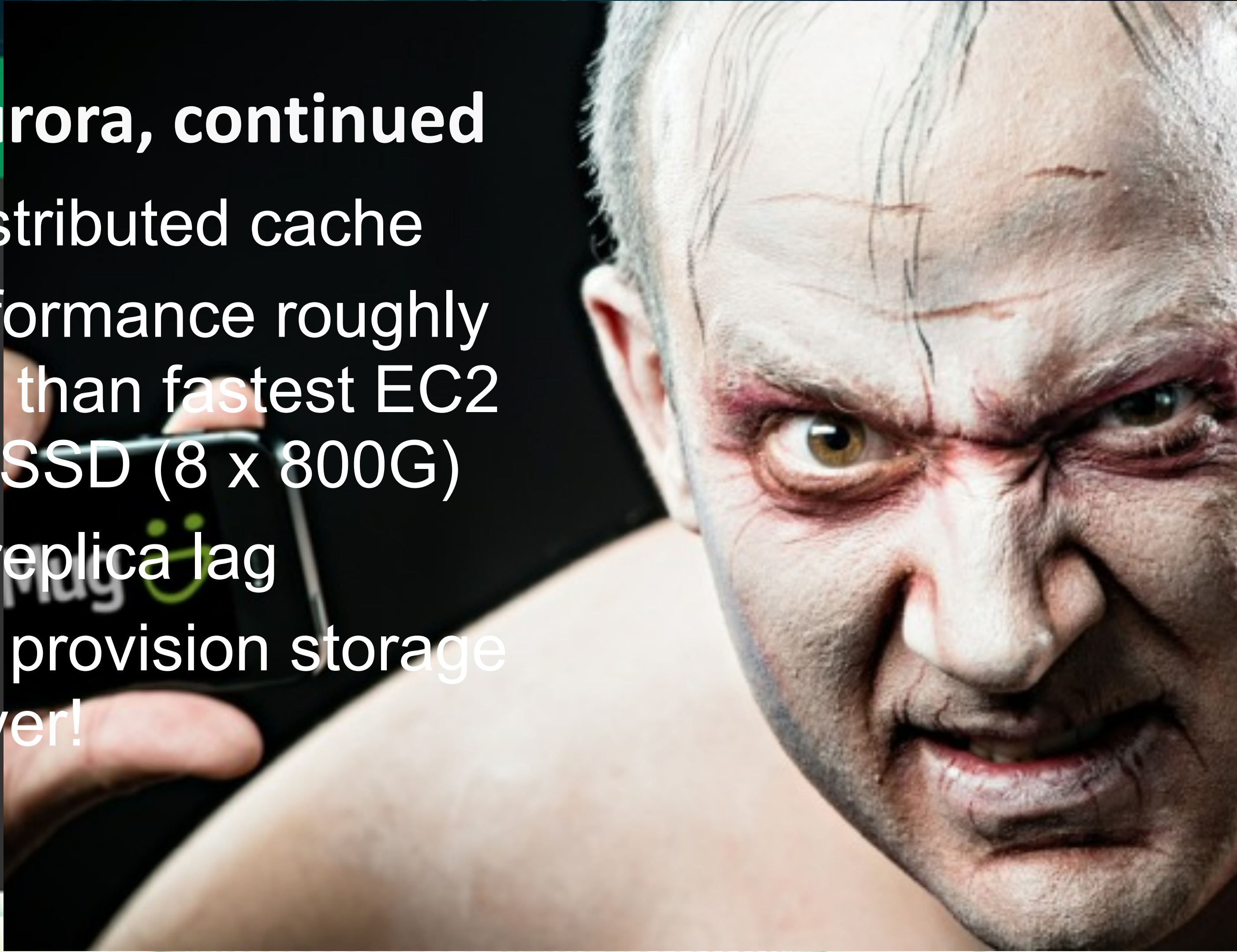
- Admin similar to RDS, backend all different
- MySQL 5.6 hacked up to separate cache, compute, and storage
- Compute: r3 (high RAM) EC2 instances only

Amazon Aurora: Storage

- A new storage method for AWS: Log-structured file system, on a single shared three-datacenter distributed system
- Master and all replicas share the same storage--each proceeding with replication by points in time on the log

Amazon Aurora, continued

- External distributed cache
- Master performance roughly 2-3x higher than fastest EC2 ephemeral SSD (8 x 800G)
- Near-zero replica lag
- No need to provision storage or IOPS, ever!



Amazon Aurora, continued

- Server cost is 65-70% higher than equivalent EC2 instances - a bargain!
- Storage cost is same as GP2 EBS SSD, 10 cents/GB-mo, despite having 6 copies on SSD!
- IO charges similar to RDS
- Increased commit latency
- No replication in currently
- More details: watch Anurag Gupta's presentation (30m) at re:Invent 2014

Make your picks

- PXC and Aurora fix failure modes of traditional MySQL replication
- But add lots of unknowns
- The leading MySQL users continue to choose MySQL 5.6 with (mostly) traditional replication

Make your picks

- Amazon designed Aurora for customers like SmugMug!
- Trust = Reputation + Time x Track record
- AWS wants to "lock you in" by being the best choice. This is their time tested track record (YMMV)

Stuff we really like

- AWS S3 and VPC
- Stackdriver and NewRelic for monitoring
- Percona’s support and tools
- Puppet, configuration management
- SmugMug’s Super Heroes

Questions?

Andrew Shieh, Sunnyvale, CA
shandrew @ smugmug.com
@shandrew

http://www.smugmug.com/
http://pics.shieh.info/

Thank you!

Please email me to request an audio recording of my session.