infosecurity NORTH AMERICA TECHTALKS

Robert Gibbons, Chief Technology Officer

# Making Big Moves:

## The Security and Infrastructure Implications from a Historic Data Migration Case Study

datto

infosecurity
NORTH AMERICA
TECHTALKS

# “Why we decided to divorce AWS”

datto

infosecurity
NORTH AMERICA
TECHTALKS

# "Why we decided to divorce AWS…and the lessons learned during the 18 months it took us to migrate 300+ EC2 instances and 12 PB of data from AWS to our private cloud."

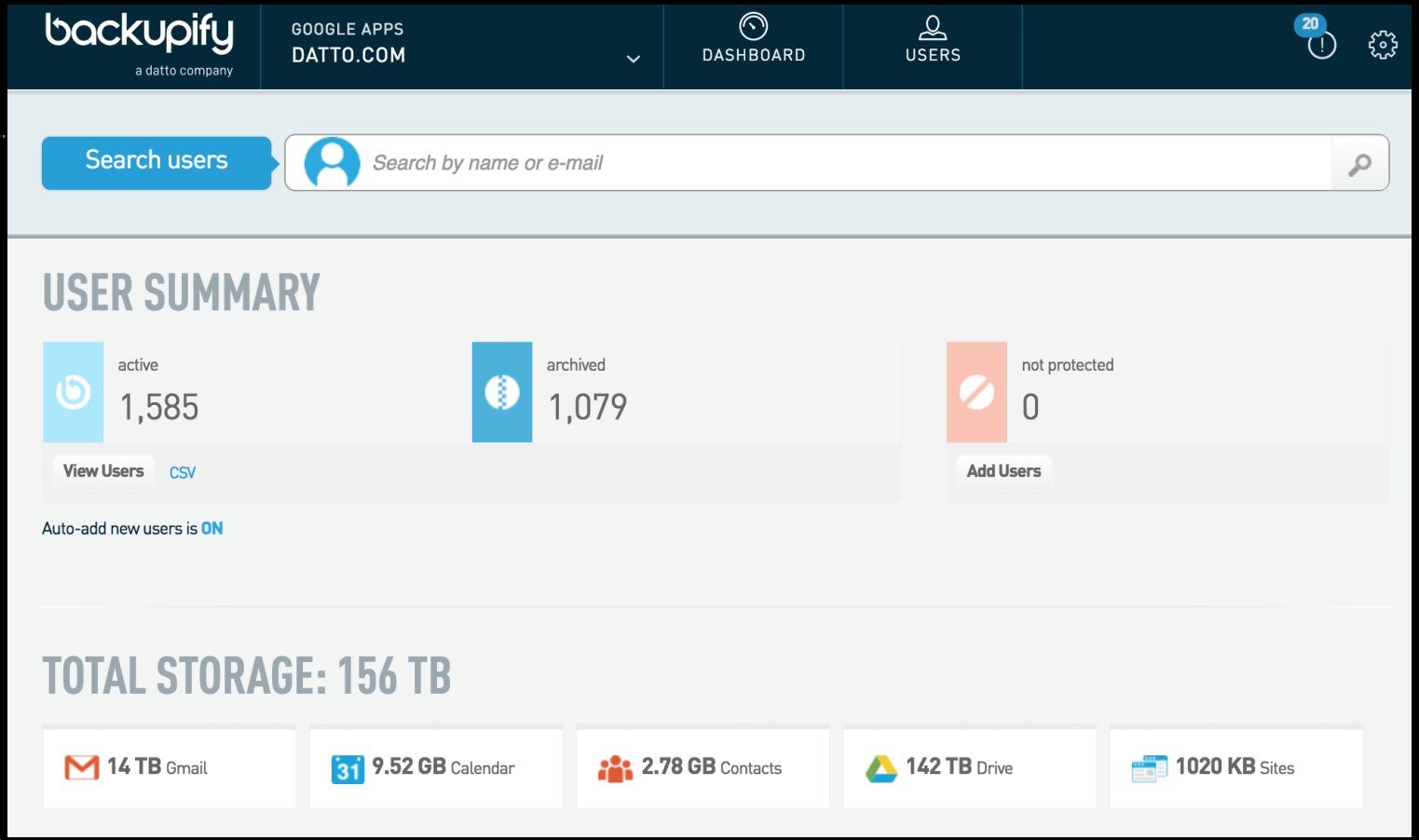
datto

# What did we move?

- Backupify: Cloud-to-cloud backup service
- Market leader in C2C backup
- Protects Gmail, O365, SalesForce

datto

infosecurity NORTH AMERICA TECHTALKS




datto

# What are we moving?

- Backupify: Cloud-to-cloud backup service
- Market leader in C2C backup
- Protects gmail, O365, SalesForce
- …wait, why would you need to backup Google?

datto

# What are we moving?

- Backupify: Cloud-to-cloud backup service
- Market leader in C2C backup
- Protects gmail, O365, SalesForce
- …wait, why would you need to backup Google?
- …wait, why did someone build a backup service on AWS???

datto

# The Public Cloud Dilemma

- Great when you're in the basement
- But the easy button makes it super-trivial to get hooked
- Infrastructure death spiral:
  - Just as you begin to succeed…
  - costs grow to be unmanageable…
  - and you haven't developed the technical knowledge to extricate yourself.

# Why private clouds make sense

- Tight control of cost structure
- Much fewer participants using/virtualizing
- Less notoriety
- You never know what the public cloud policies are
- You can respond to government entities…or fight for yourself and your customers

# Pros of Private Cloud

- Maintain ownership of data, control over access, and insight into infrastructure changes that ay affect security posture and risk exposures.
  - Sharing of compute is limited, or at least under your control.
- Large public clouds attract attackers
  - Due to the complexity of configuration that often leads to configuration mistakes and open access points (i.e. open buckets, incorrect IAM policies, etc.)
  - There are bots and automation that constantly scan for such misconfigurations and can leave you nearly exposed.

# Pros of Private Cloud (cont.)

- Private Clouds permit you to build in security measures and processes that are purpose built to protect data and applications.
  - Public clouds offer generic controls that need to be applied correctly by the administrator in order to offer basic protections.
- Easier to understand where data is located.
  - Can help assure and meet compliance with data privacy and data breach regulations.
- Physical Security
  - Ability to independently audit the physical locations where data is resident and assure proper physical security controls via visual inspection.
  - You don't get that ability with large public cloud providers. You have to blindly trust or trust based on someone else's assessment.

# Cons of Private Cloud

- Full control can lead to a false sense of security
- Public cloud providers often have more resources and attract strong talent that may not be available to small organizations building private clouds.
- With public cloud you do not have to be concerned with any response to hardware supply chain issues like those being talked about in the press recently.

# Datto Cloud

- Purpose-built cloud for disaster recovery
- 450 PB of storage
- 4 million backups a day
- Thousands of compute instances

datto

infosecurity
NORTH AMERICA
TECHTALKS

datto

infosecurity
NORTH AMERICA
TECHTALKS

# datto

# Datto Cloud

- Purpose-built cloud for disaster recovery
- 450 PB of storage
- 4 million backups a day
- Thousands of compute instances

datto

# Why did it make sense to move?

- Purpose-built cloud for disaster recovery…sat unused for most of day
- Buy infrastructure at sufficient scale to make it worthwhile
- Our view: You can't be an infra company and not have your own infra
- Our view: Once you >1PB of storage, time to move.

# Project Exodus – Initial State

- Hundred of EC2 instances
  - Typically more expensive instance types
- Tens of billions of S3 objects
  - On the way to hundreds of billions*
- $800k/month bill and growing

infosecurity NORTH AMERICA TECHTALKS

# Explosion in cargo bay 4!

**S3 storage usage goes from 2PB to 10PB in 3 months**

datto

# Project Exodus – Initial State

- Hundred of EC2 instances
  - Typically more expensive instance types
- Tens of billions of S3 objects
  - On the way to hundreds of billions*
- $800k/month bill and growing

# Strategies Considered

- Private line into AWS to copy data
- Copy everything over the WAN
- Migrate at the speed of FedEx
  - USB2 side-load speeds (at the time)
  - Hundreds of round-trips for hundreds of servers

# What made the project easier?

- Other than EC2 and S3 minimal dependency on other AWS services
- Application designed to be ephemeral
  - E.g., chaos monkey architecture
  - perfect for running compute as a background tax
- …not much else.

# What made the project harder?

- No process of data deduplication
- Data initially removed by deleting encryption keys (lazy storage cleanup)
- A few customers with a typical storage use patterns drove infrastructure usage.

# Moving: Compute

- Before:
  - EC2
  - 100s of compute instances
  - Variety of workloads and system requirements
- After:
  - Combination of (i) bare-metal servers (ii) VMs (iii) Docker
  - Created AWS-like provisioning service internally
  - Designed allocation strategy for resources in cloud

# Moving: Cassandra

- Before:
  - 2 clusters, more than 100 EC2 nodes total
  - 1 cluster read-oriented; 1 cluster write-oriented
  - Mix of Cass versions, some approaching end-of-life
- After:
  - Docker
  - Migrations & upgrades in parallel
- Migration Strategies
  - Lift-and-ship it (rsync FTW)
  - Application-driven migration
  - Later, after upgrades: Cassandra streaming replication

# Moving: Redis

- Before:
  - 5 clusters, more than 60 nodes total
  - Mixed-use ephemeralish data
- After:
  - VMs
- Migration Strategies:
  - Rebuild cache data
  - Replace, drain, retire
  - Redis built-in replication
  - Life it & ship it

# Moving: Postgres

- Before:
  - 3 master-slave AWS instance pairs
  - 1 monolith; 2 service-oriented
  - 24/7 uptime requirements
  - >250 GB
- After:
  - Bare-metal
- Migration Strategy – replicate; failover; retire
  - PG master-slave replication to new slave host
  - Once slave is caught up, promote slave to master
  - Build new slave from master
  - Once slave is caught up, retire former master

# Moving: Block storage

- Before:
  - S3
  - 10+ PB of data
- After:
  - OpenStack Swift

infosecurity NORTH AMERICA TECHTALKS

# Explosion in cargo bay 5,6, & 7

## The week we DDoSed AWS, blew a hole in Cassandra, and broke Swift

datto

# Moving: Block storage

- Before:
  - S3
  - 10+ PB of data
- After:
  - OpenStack Swift
- Haunted by all the times an engineer said “Well that’s Amazon’s problem not ours”
- Most data was stored in one S3 bucket – tens of billions of objects
- Swift couldn’t come close to that…Swift depends on SQLite
- Deduplication required every object to be re-encrypted

infosecurity
NORTH AMERICA
TECHTALKS

# Lessons from the front-line

## Avoid these patterns

datto

# Lock-in pattern…

**Reliance on proprietary orchestration**

infosecurity
NORTH AMERICA
TECHTALKS

# Lock-in pattern…

**Solving infrastructure problems with the easy button**

datto

# Lock-in pattern…

**Lose internal roll-your-own capabilities**

# Lock-in pattern…

**No muscle memory for controlling hardware costs**

datto

# When does a private cloud make sense?

- Probably when you have…
  - More than 1 PB of storage
  - 300+ compute nodes in constant use
  - On-going stable (or predictable) compute & storage needs

# Lessons Learned

- Plan – get an initial sense of your growth velocity
- Define "done"
  - When has everything been migrated?
  - 75% of the work will get you 90% of the benefit
- Know your bottlenecks and problem services

# Lessons Learned

- Consider alternatives to migration
  - S3 infrequent access storage
  - Glacier
  - Snowball import/export service
  - Retrieve the data again

# Lessons Learned

- Replacing AWS services will be harder than you think
- Determine what needs to be upgraded now and what can wait
  - More recent versions of software usually have better migration capabilities
  - But try to separate migrations from upgrades when possible

# Lessons Learned

- Timing – identify biggest contributors to cost and focus on those first
- Consider building out multiple options for exfiltration simultaneously – you don't know what won't work
- Avoid temptation to solve "tech debt" problems during migration – you will regret it.
- Bias towards incremental progress over total completion.

# Lessons Learned

- Keep a record of decisions and why you made them. Be kind of future engineers.
- Beware egress costs – they add up, consider running some operation before egress.
- Think through your cross-cloud communication strategy – you'll be using it a lot.

# Lessons Learned

- Metrics – you will need them
  - Find metrics that show the current plan is the right plan.
  - Find metrics that show the success of the migration.
  - Find metrics that show the overall health of the application as you are migrating.

infosecurity
NORTH AMERICA
TECHTALKS

# Questions?

datto