Booking.com

Database migration to The Cloud

Nicolai Plum
MySQL Database Engineering - Booking.com

Agenda

- **Included:**
- Moving data
- Replication
- Recommendations

- **DIY:**
- Cloud config
- Cloud deploy
- Cloud IAM
- Cloud networking setup
- Marketing spiel

Types of Cloud Database

Managed service

- AWS RDS
- AWS Aurora
- OCI MySQL Database Service
- OCI MySQL Heatwave
- … and many others

Self-managed MySQL

- In compute container
  - AWS EC2
  - OCI Compute VM
  - … very many others
- In Kubernetes
  - With operator, maybe

Networking to Clouds

- Try to get good bandwidth
  - Direct/Private/whatever connect
  - Or at least very fast VPN
- If you’re bandwidth limited, try to transfer binary dumps instead of text, and compress them.

Data transfer interfaces

Managed Service

- Most things you can do via the SQL interface
- Limited and particular bulk data import methods
- Native support for object storage

Self-managed MySQL

- Anything you can do via the SQL interface
- All the usual bulk import/export features
- Many unofficial but useful hacks (copying datadir)
- Interfacing to object stores is your (or your tools') problem

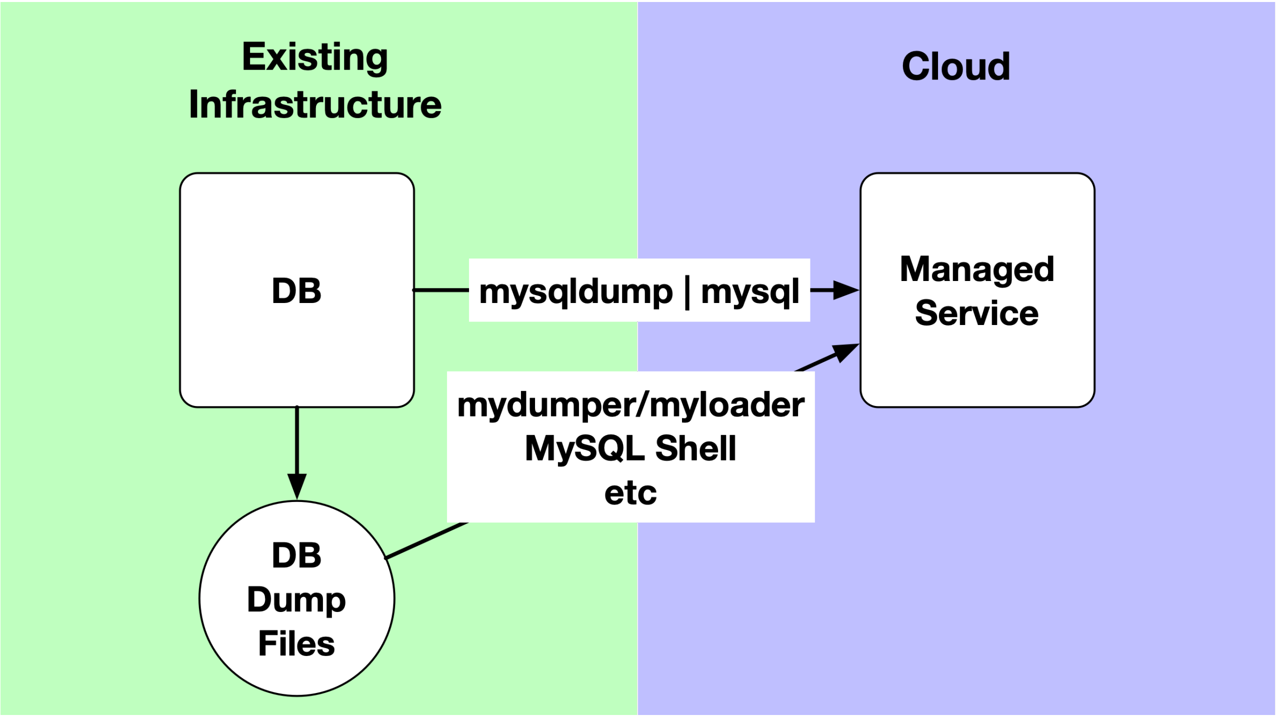
Transfer strategy - technical

- Direct to managed service (DBaaS)
- To managed service via object storage
- To self-managed MySQL in cloud

Transfer strategy – non-technical

- Who will be running the cloud instance?
- Same people as before?
- Or another team?
- Cross-account or cross-team access and responsibility?

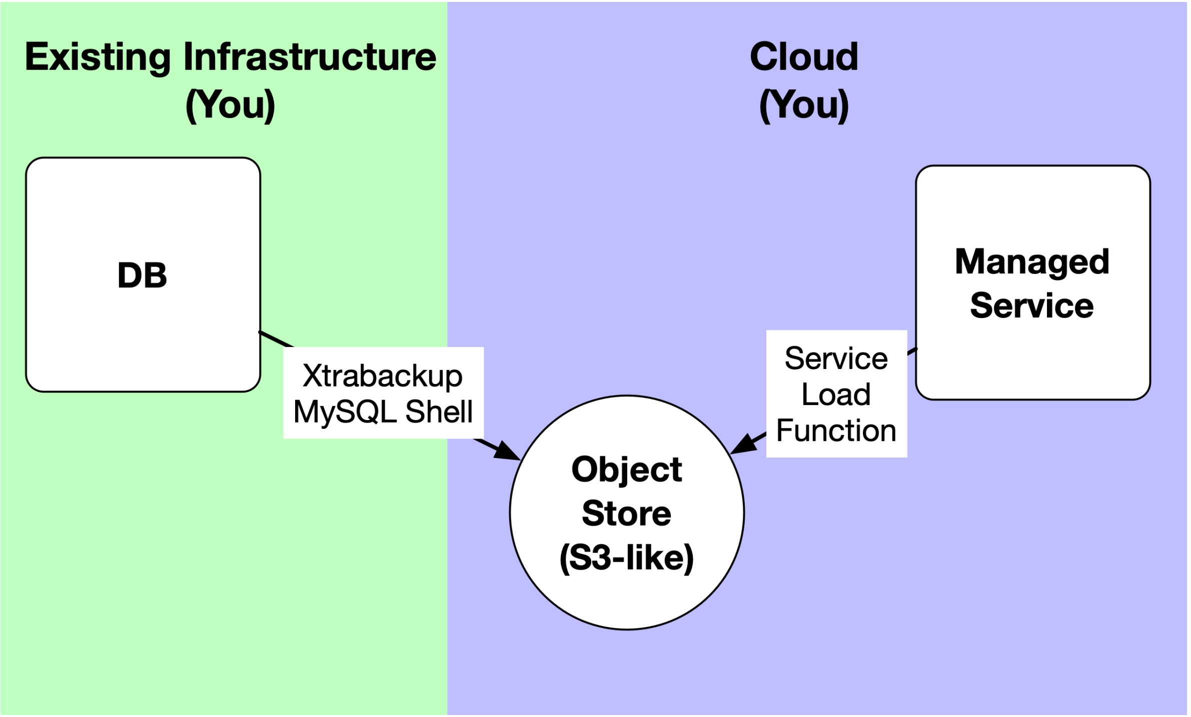
Direct to Managed Service



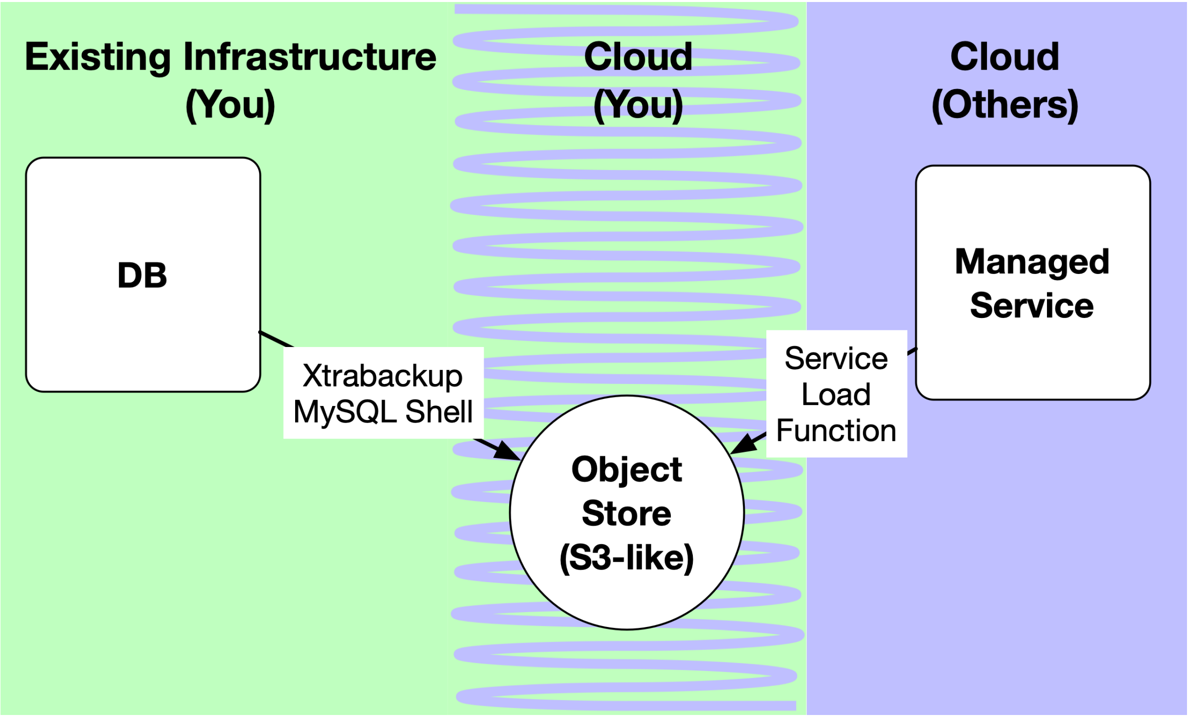
Direct to Managed Service

- Mostly a DIY experience
- Faster methods need to store local copy before uploading to managed service
  - mydumper / myloader
  - MySQL Shell

Via Object Storage



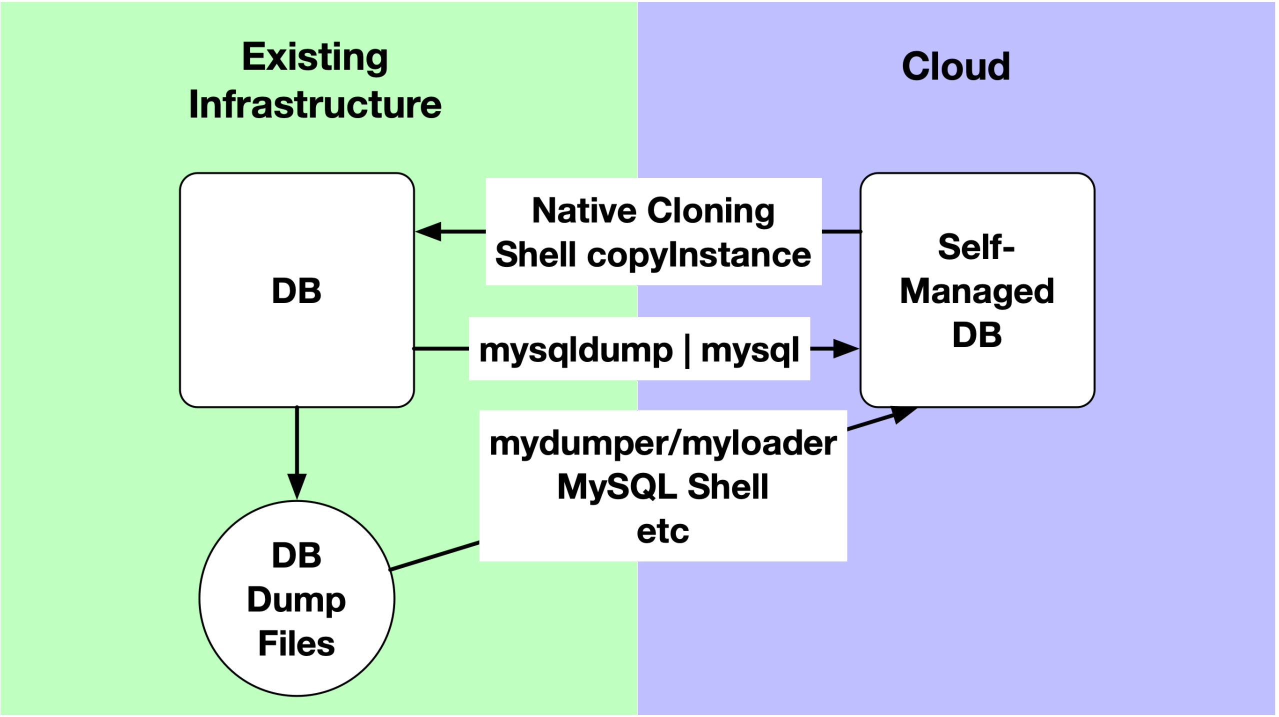
Via Object Storage (cross-org)



Via Object Storage

- Tested, simple process
- Limited to the transfer methods supported by the service
- Can be slower (due to object store)
- Significant downtime for migration
    - Unless you set up replication

To Self-Managed MySQL in Cloud



Booking.com

Tools | SQL-like

- mysqldump
  - Slow
  - Very clear data format (SQL statements)
- mydumper / myloader
  - Same same but parallel
- MySQL Shell dumpInstance() / loadDump()
  - Fast (but not as fast as Native Cloning)
  - Handles S3 (including on AWS & OCI)
  - Replaces mysqldump everywhere (probably)
  - Can cross vendors (MariaDB/MySQL, probably others)
  - Used by OCI migration tools

Tools | Database dumpers

- Xtrabackup
  - Supported by AWS
  - Main way to get an S3 dump that RDS can ingest natively
  - Sometimes has to catch up with new DB releases
  - Fast

Tools | Database dumpers

- Native Cloning
  - Only works for copying to self managed MySQL in cloud
  - It Would Be Nice if AWS and OCI supported it
  - It Would Be Nice if it could dump to S3
  - … like xtrabackup
  - Very fast, even over a WAN

Tools | Data pumps

- AWS DMS
  - Replicate like it’s 2015 (no GTID support)
  - OK for copying
  - Only OK for copying in low-stress cases, not more
  - Awful for replication
  - Can do data transformations while copying

Tools | Data pumps

- Goldengate?
  - Support is documented
  - Even dedicated Oracle RDBMS people say “it’s expensive”
  - Pretty solid reputation, unlike DMS
  - Anyone tried it? Tell me later

Cloud import performance

- Cloud data ingress is free
  - … but the fast connects are not
- Time also has cost
  - Even if your migration deadline is *mañana*
- “Importing data” is a transient condition
  - Optimise hardware for transfer speed and reliability first, re-optimise for cost later

Performance | Native Cloning

- Native cloning will use all net/storage BW
  - So maximise them, at least while migrating
- Scale RDS or EC2 instance appropriately
- Native cloning limited by network and IOPS
  - You can turn AWS RDS and EBS IO capacity up and down online

Real-life Performance | Native Cloning

- 4Gbit/s to/from SATA SSD on 10G network
- 10Gbit/s to/from Pure Storage on 10G net
- 16k IOPS pagesize to AWS EC2 instance with 16k IOPS storage configured, across WAN (~2Gbit/s)
- **Strong benefit from large hardware**

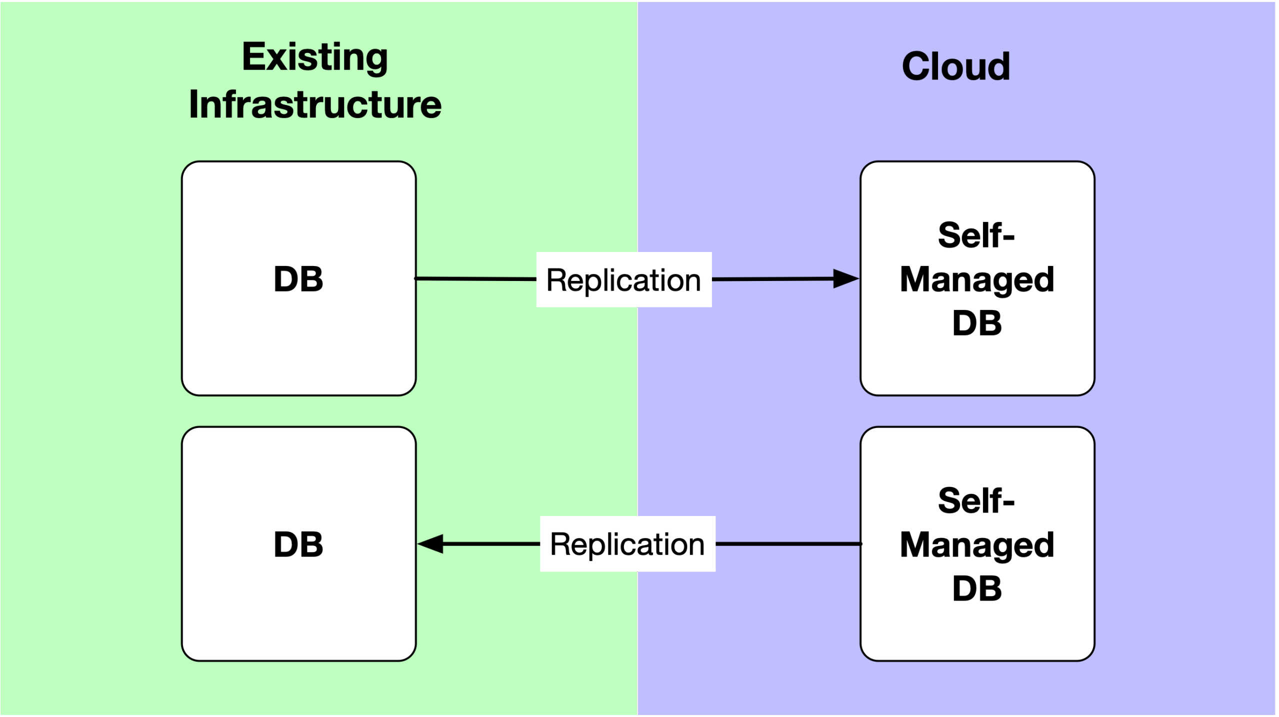
Performance | MySQL Shell

- Can use many IOPS and much CPU, so configure a large/fast instance
- See Kenny Gryp’s benchmarks for tuning hints
- Used by OCI migration tools
- Buy plenty of networking bandwidth
- Treat as Native Cloning for performance planning

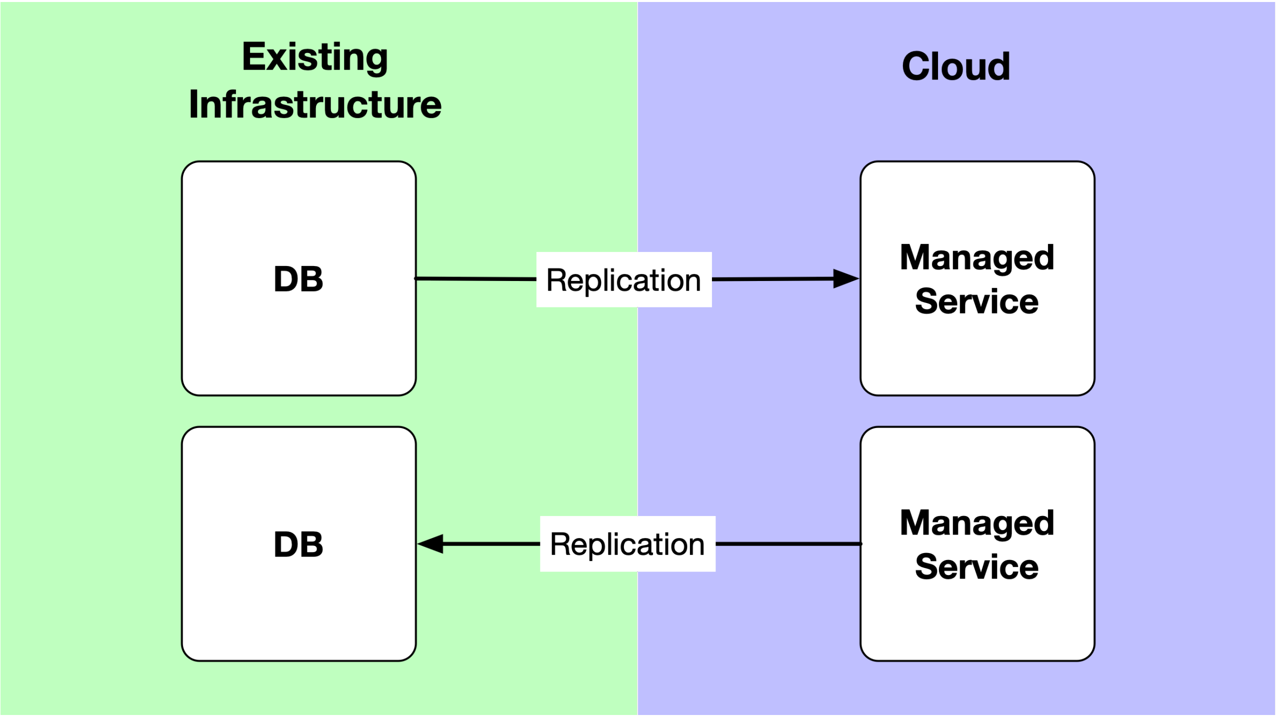
Performance | Object stores

- Ensure good bandwidth to S3 / Object Store
- AWS S3 write speed for one prefix is somewhat limited and you can’t increase that
- Object store restore speed – new instance can take time to reach full performance
- If dump and migration speed is a critical part of your migration, test at full scale before committing.

You know how to do this



Not as good as you hope



Existing Infrastructure

Cloud

DB

Replication

Managed Service

DB

Replication

Managed Service

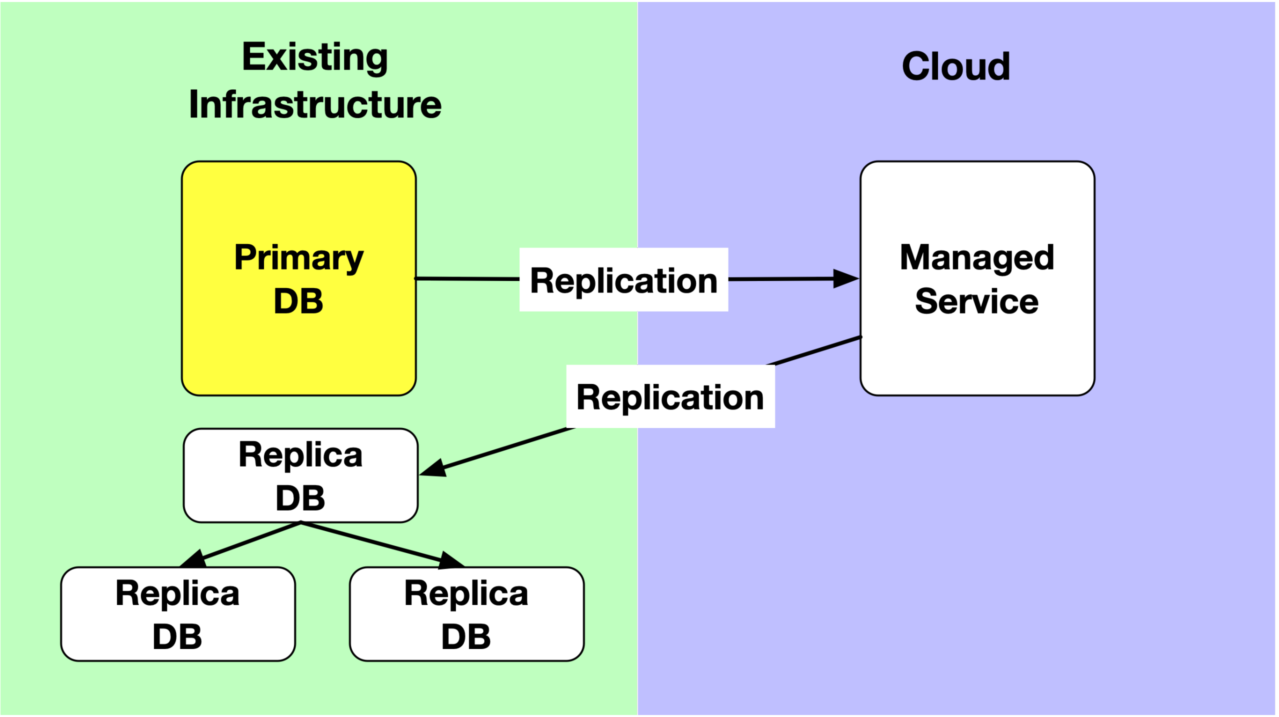
Replication to/from Managed Services

- Replication out (off-cloud replica) has the same problems as managing any other replica
  - But if you use the cloud read/write endpoint as your source, the cloud mostly keeps things working.
- Replication in (off-cloud source) lacks good features on managed services
  - No source list and auto failover
  - External management and monitoring needed
  - Strong case for anycast routing, load balancer, proxy (e.g. ProxySQL) or other “single source endpoint” abstractions.
- Strong management is needed for any non-temporary replication setup

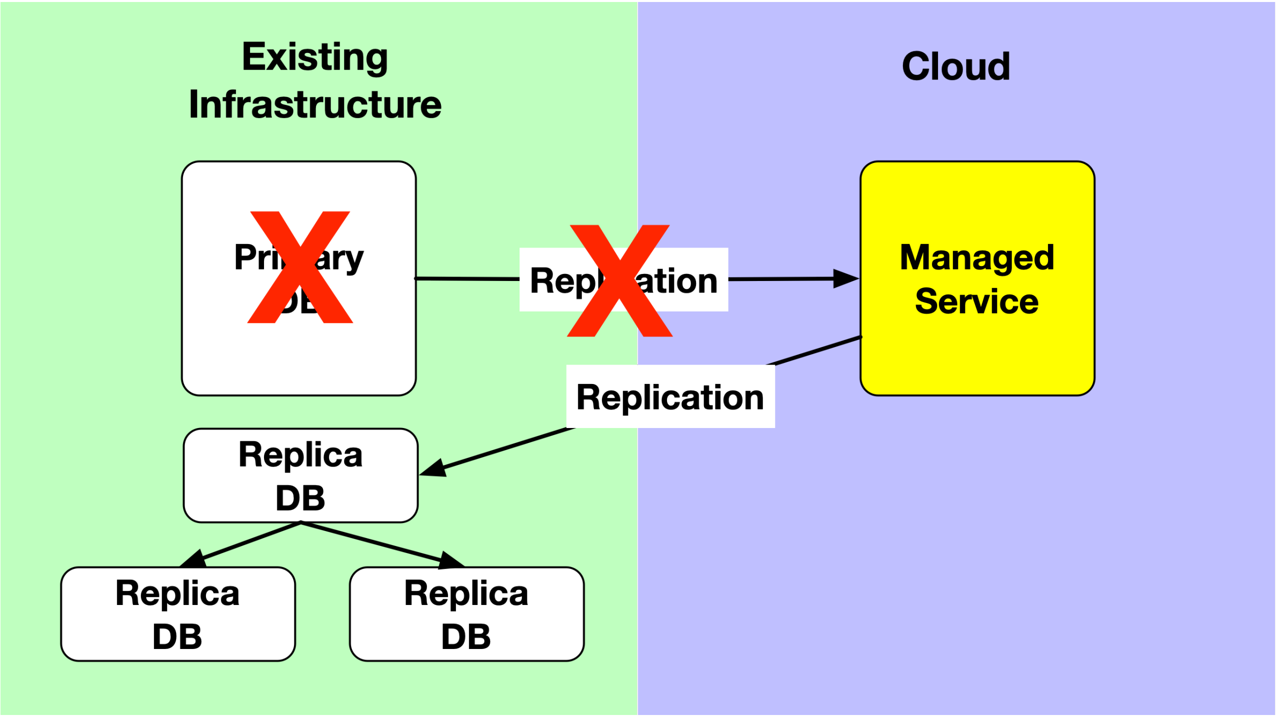
Migration with replication

- Allows some testing
- But once you cut over writes, going back is hard
- Replication there and back can help you just in case
- Cloud network link reliability is critical

Before migration

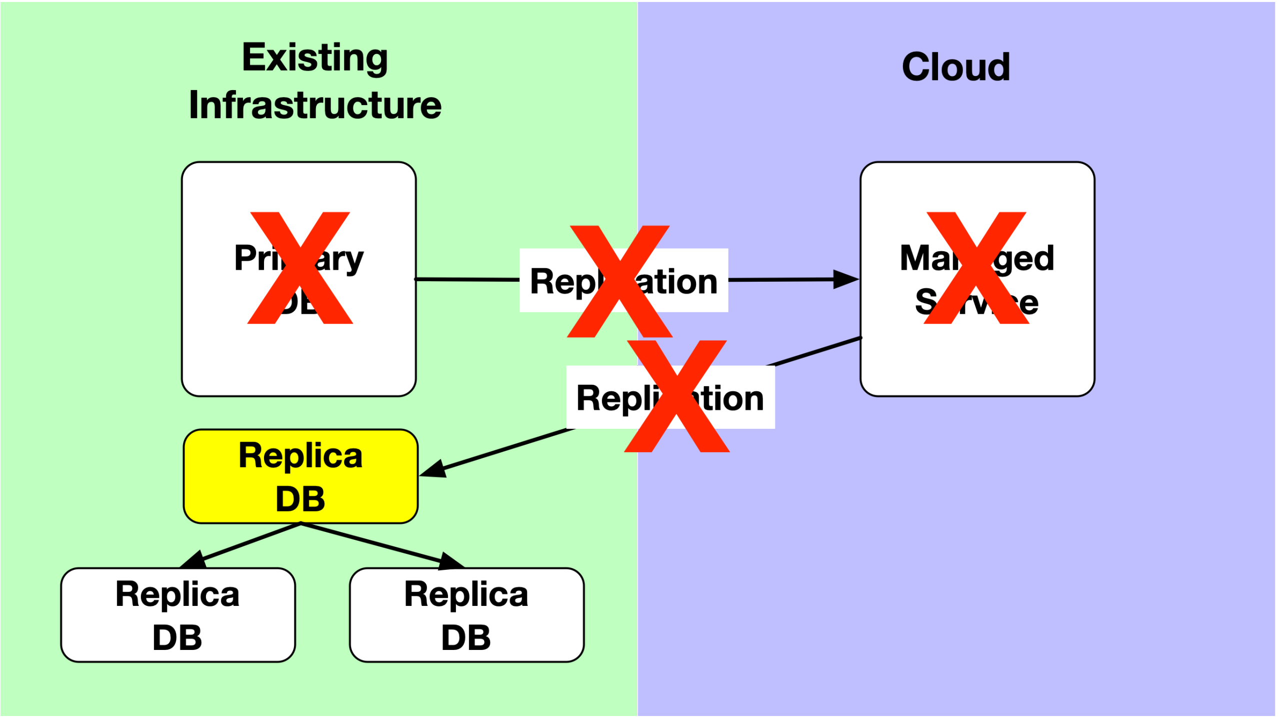


Successful migration



Booking.com

Back-out



Booking.com

Migration without replication

- Either
  - Run services and datastores in parallel
- Or
  - Have a lot of downtime so you can test before committing
  - …and a way to get recent data out of the cloud service if you do have to go back.
- Or
  - This is a courageous act, Minister

Self-managed instances

- Stepping stone or end solution
- Testing, playground
- Easy data copy inside a cloud
- More advanced MySQL features (at cost of more management overhead)
- Fewer clever storage features

KISS

- Clouds have many features
- Using many of them adds complexity that can be very hard to troubleshoot
- “Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.” (Oracle MySQL Service documentation)

With thanks to

Simon Mudd
Hrvoje “Henry” Tonkovac

?

nicolai.plum@booking.com

https://jobs.booking.com/careers