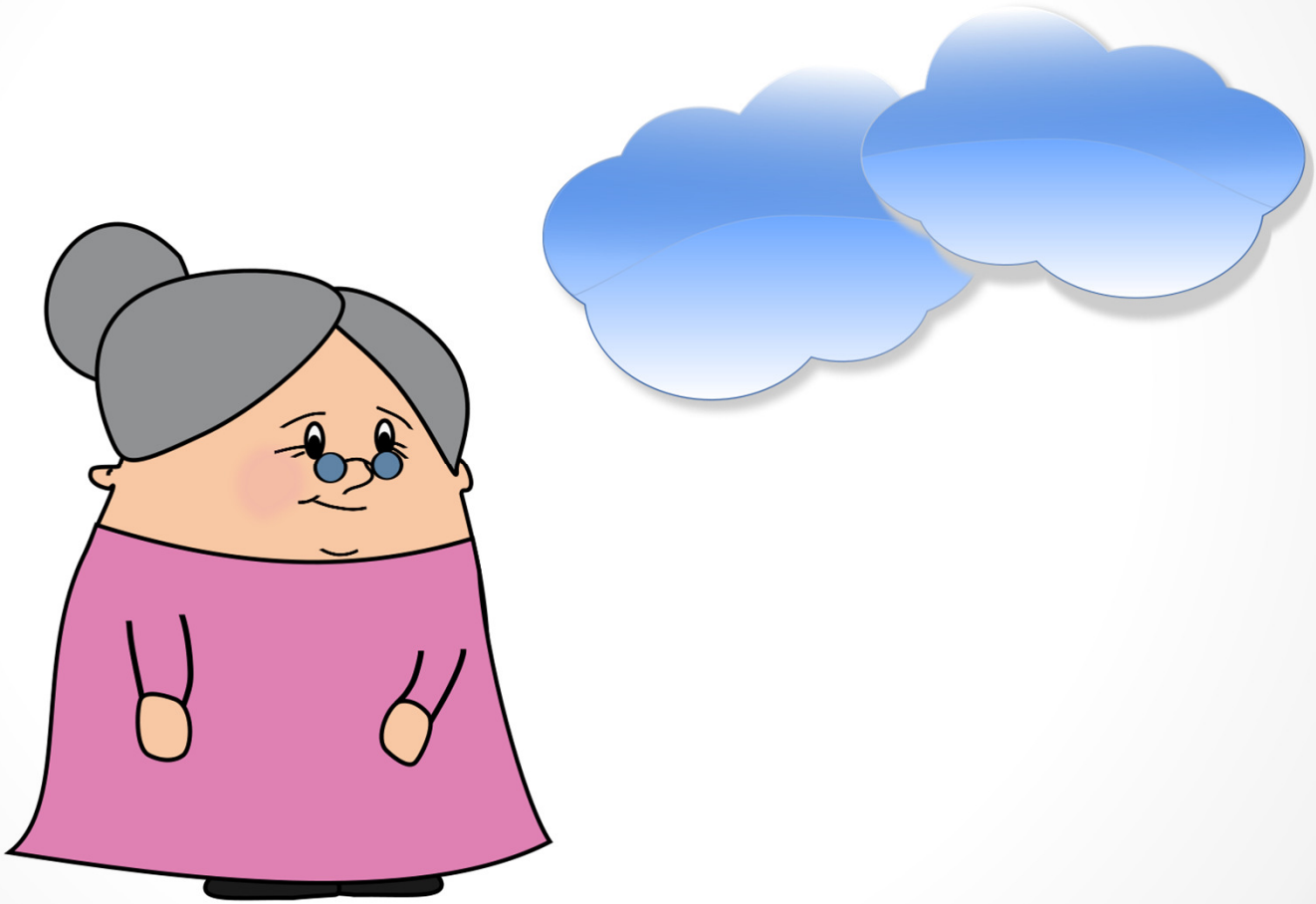
Between Ground and Clouds

Similarities and differences in database backend implementation

About me

- 10+ years of experience working with SQL Server
- Microsoft SQL Server MVP
- MCITP (DBA, DB Developer), MCPD
- Blog: http://aboutsqlserver.com
- Email: dmitri@aboutsqlserver.com

What does “Clouds” mean?

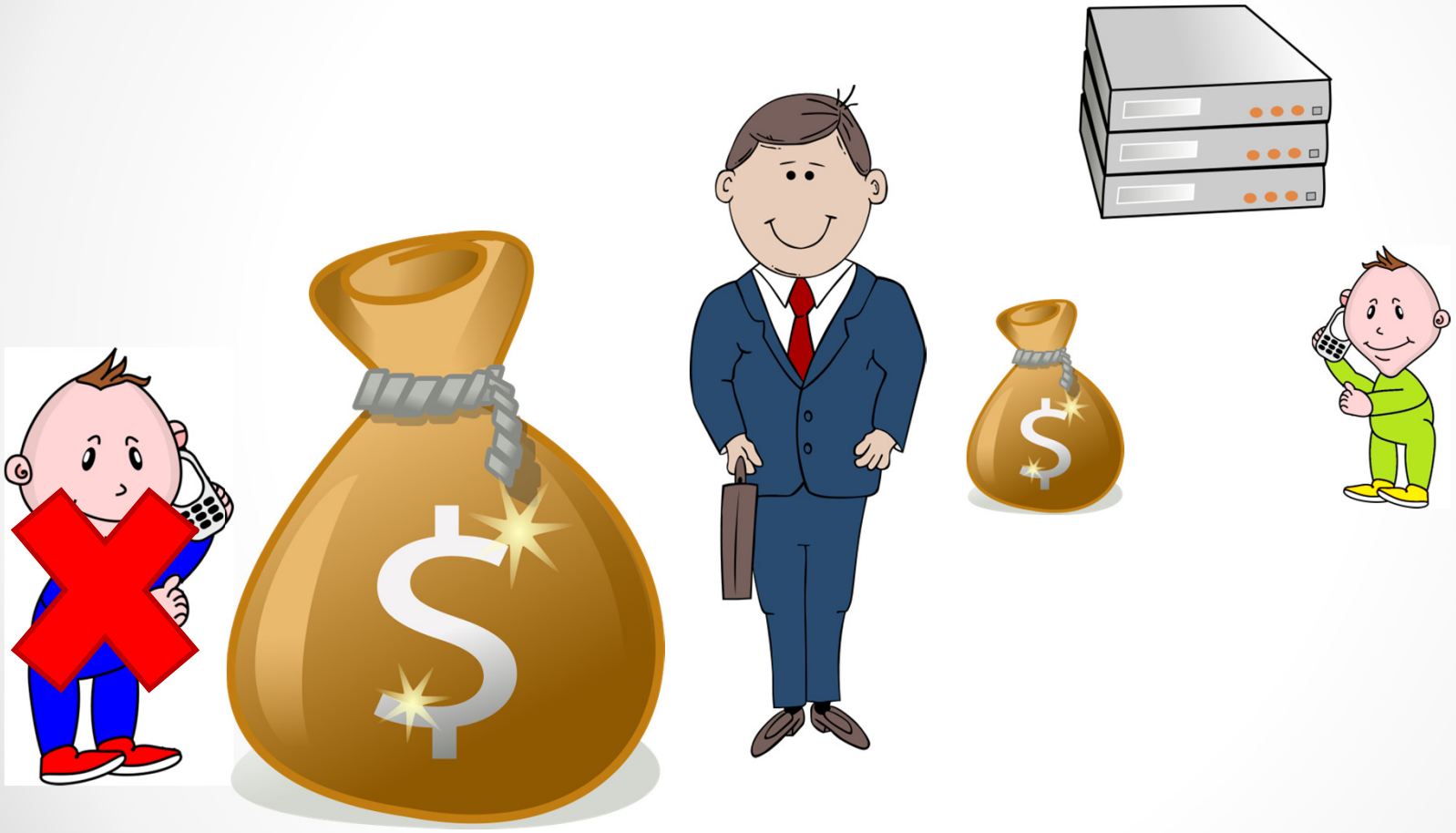


What does “Clouds” mean?

What does “Clouds” mean?

What does “Clouds” mean?

What does “Clouds” mean?



What does “Clouds” mean?

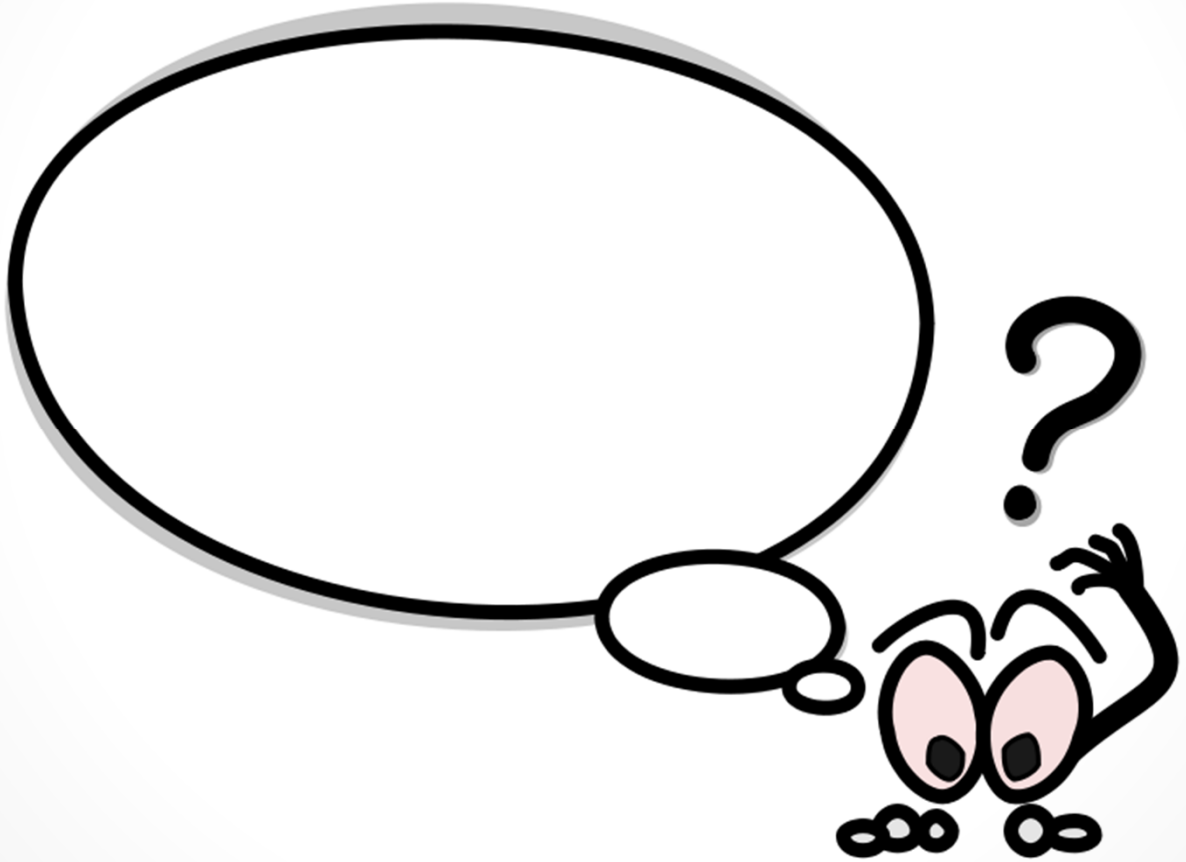
What we are going to do today

- Compare

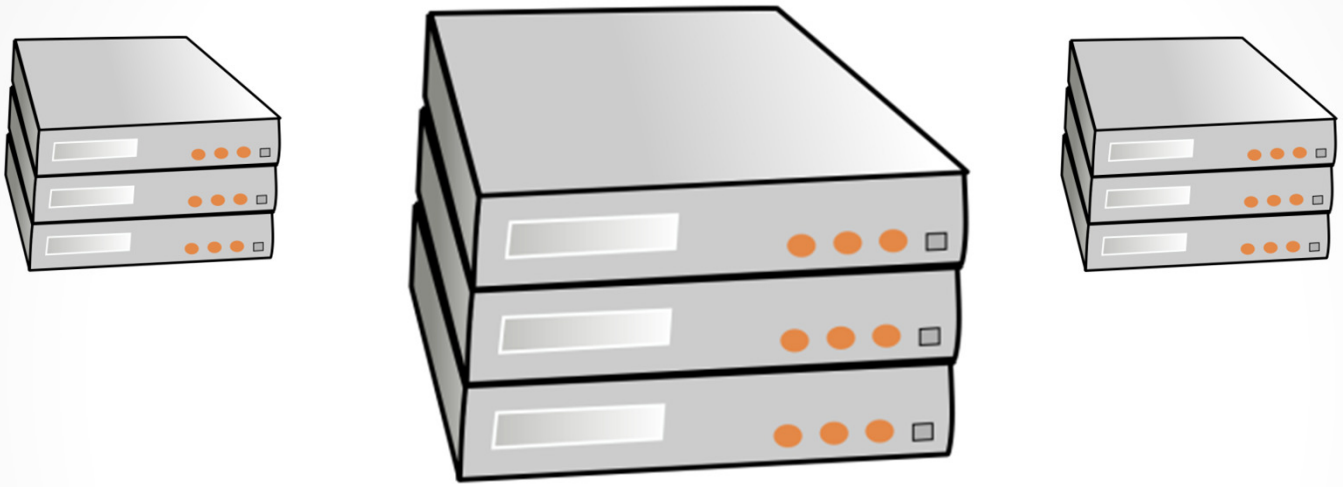
- Focusing on database backend
  - My goal is provide an overview of the platforms from database backend standpoint and define the list of "think about" items

- Things to keep in mind
  - Each platform has pros and cons that varies for different systems and workload patterns
  - Content is valid as of September 2012

Why do we use Clouds?



Reduce Cost
Capacity on Demand

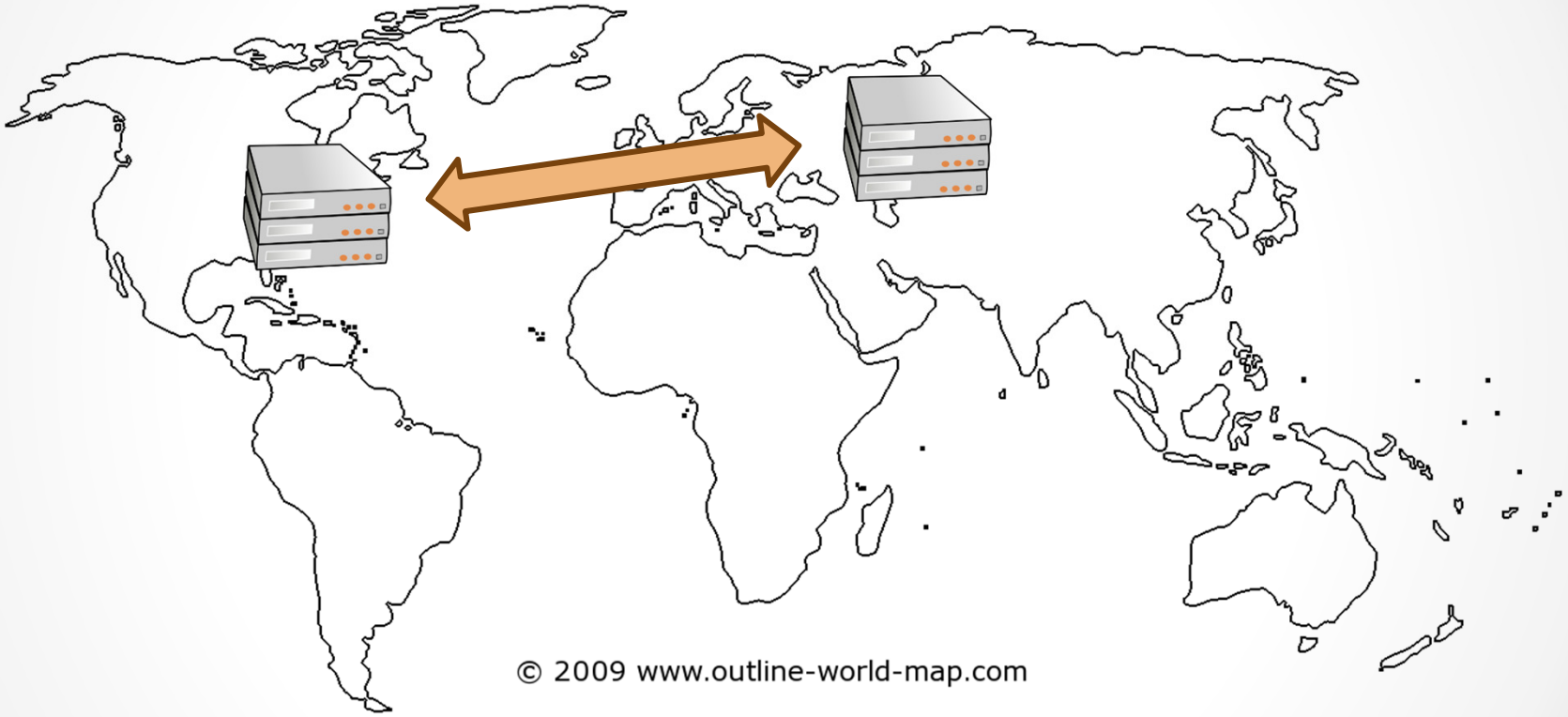


- Very low upfront cost (hardware, software)
- Elasticity
  - Growing customer base
  - Seasonal Activity
- Pay-as-you-go

Reduce Cost
Lower Operational Cost

- Lower operational expenses (power, rack space,..)
- *Potentially* less IT stuff

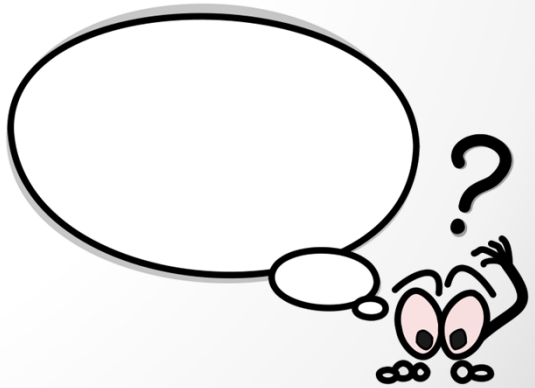
Reduce Cost
High*er* Availability



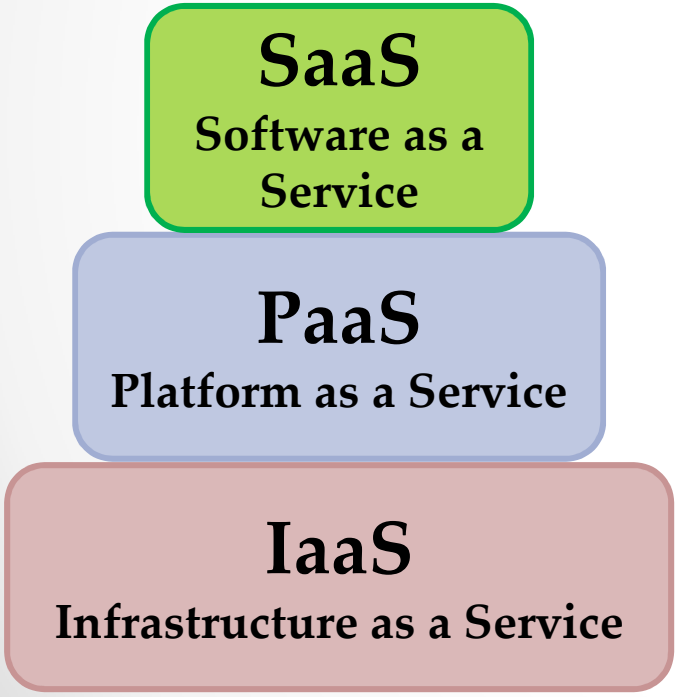
- Data is replicated across different Data Centers *automatically (sometimes)*

But what about Development Cost?

- Do we always plan to run system in the Clouds?
  - And in that particular platform?
- Can we port existing systems to the Clouds?
  - What limitations do we have in each platform?

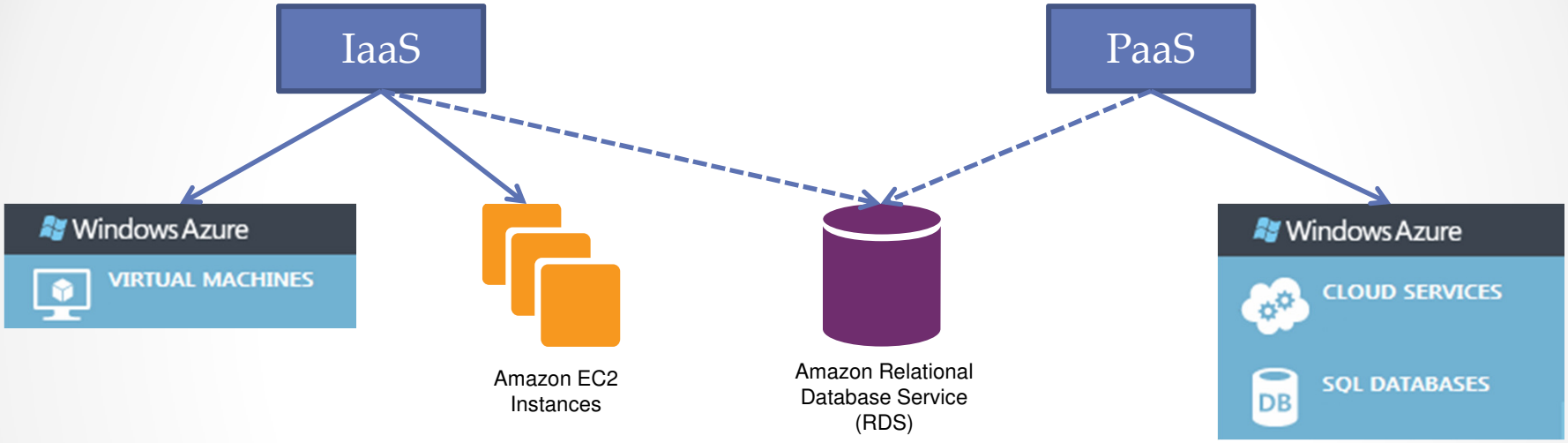


Cloud Models



- PaaS (Platform as a Service)
  - Renting the platform rather than infrastructure
  - Scalability (almost) on Demand
  - Abstracted physical storage
  - Behind the scenes maintenance and patching
    - Both: OS and SQL Server
- Iaas (Infrastructure as a Service)
  - "Renting" the Infrastructure
    - VM, Network, Storage, etc.
  - Scalability via configuration
  - On-premises infrastructure can be ported with minimal changes
  - Full version of SQL Server can be used
  - You are your own worst enemy
    - Management, maintenance, HA/DR is your own responsibility

Cloud Models



IaaS - from SQL Server prospective

Windows Azure VM

- Up to 8 cores with14GB of RAM
- SQL Server images
  - SQL Server 2012 Evaluation Edition

Amazon EC2 / RDS

- Up to 8 cores with 68.4GB of RAM
- SQL Server images
  - SQL Server 2008 Express
  - SQL Server 2008 Web
  - SQL Server 2008 Standard
  - SQL Server 2012 Standard

Storage

- Storage is usually the main bottleneck nowadays
  - It's even bigger issue with in the Clouds
- Both platforms are NAS driven <=1Gb Ethernet (for now)
  - Amazon AWS has an option with 10Gb Ethernet backed up by SSD. ($$$)
  - Network connection shared between multiple volumes
  - RAID is possible
    - Increases IOPS, does not increase throughput
  - Data Warehouse (Sequential Read) – network throughput is the limit
  - OLTP (Random Read)
    - High IOPs are achievable but latency is a bit on the high side
    - Beware of non-optimized queries
- There are limits on # of volumes and max volume size
- Research, research, research
  - Warm up the volumes
  - Strip the volumes
  - Test drivers

Storage in AWS

- Three types:

Amazon Elastic Block Storage (EBS)

Amazon Simple Storage Service (S3)

Amazon Glacier

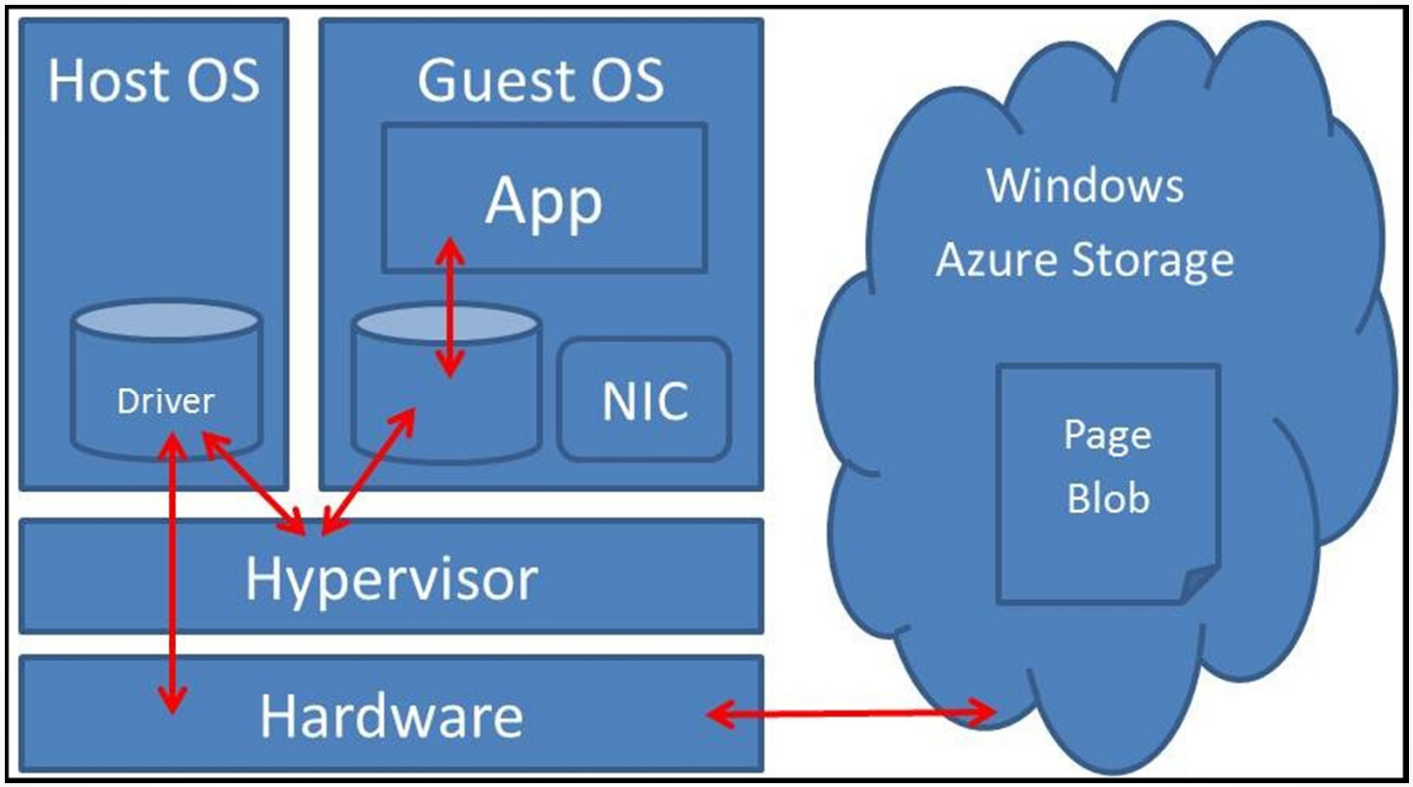
- EBS - Multiple options
  - Low/Medium/High IO Performance – still through <=1Gb Ethernet
  - Provisioned IOPS – address inconsistency issues
  - Can be SSD based in some instance types
    - Cannot be easily moved between VMs

Storage in AWS Latency

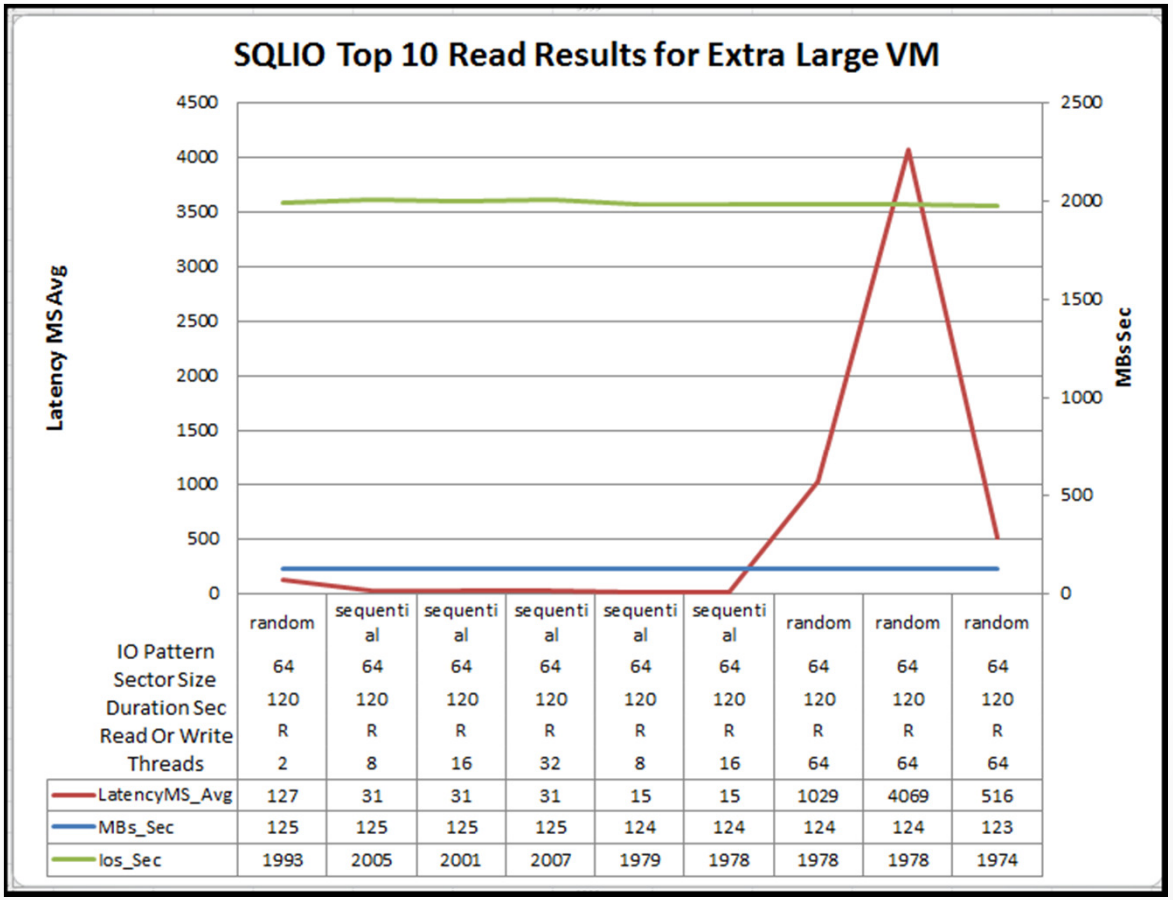
	DB Name	File Name	Latency ms
1	tempdb	tempdev	52
2	tempdb	templog	17
3	tempdb	tempdev2	75
4	ProdDb	ProdDb	54
5	ProdDb	ProdDb_log	12
6	ProdDb	ProdDb_FG4	21
7	ProdDb	ProdDb_FG5	41
8	ProdDb	ProdDb_FG6	47
9	ProdDb	ProdDb_FG3	33
10	ProdDb	ProdDb_FG2	20
11	ProdDb	ProdDb_FG1	19

- IO Latency from production VM (OLTP workload):
  - Extra Large VM
    - 68Gb RAM
    - No IOPS provisioning
- Log files latency (sequential IO): ~10-20ms
- Data files latency (random IO): ~30-50ms
- Different drives behave differently
  - Test the drives with SQLIO/SQLIOSIM

Storage in Azure VM

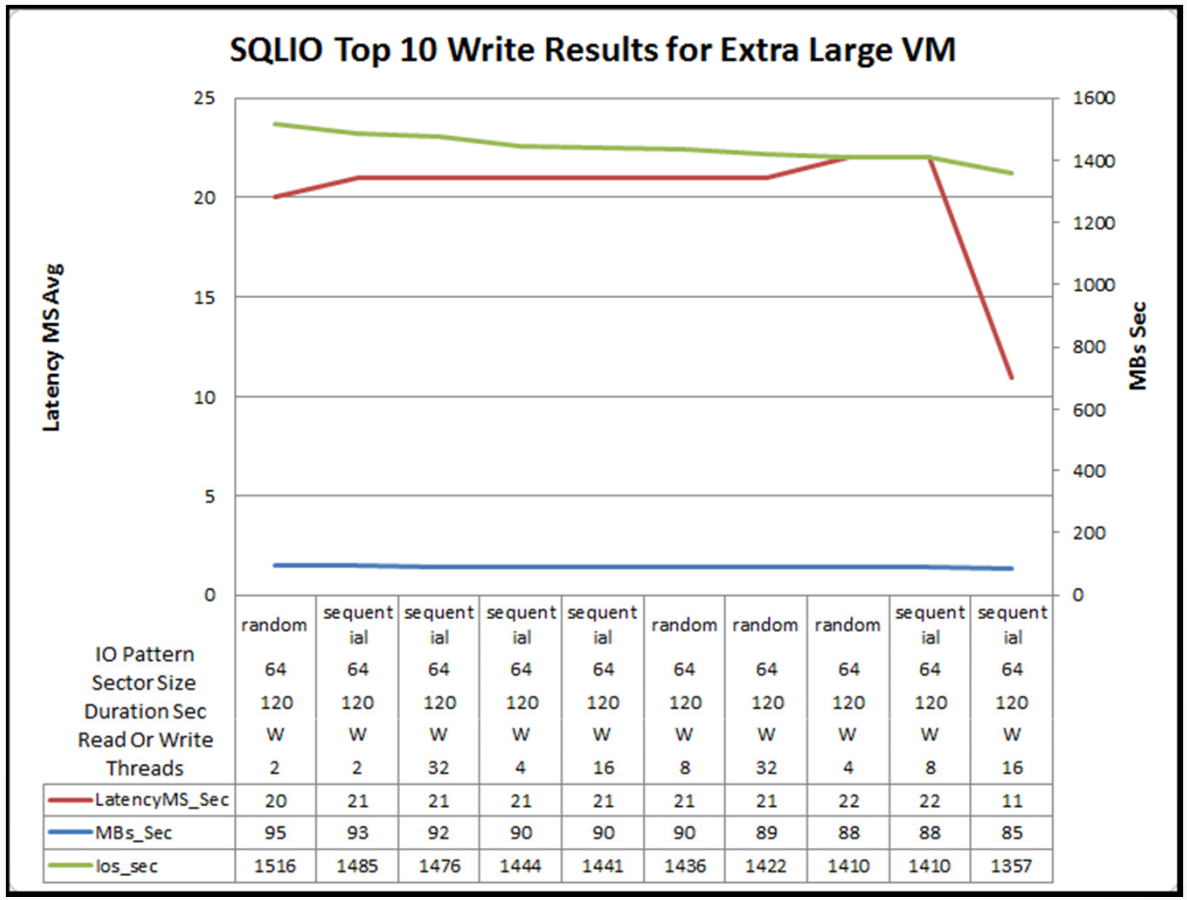


Storage in Azure VM Reads



IO Pattern	random	sequential	sequential	sequential	sequential	sequential	random	random	random
Sector Size	64	64	64	64	64	64	64	64	64
Duration Sec	120	120	120	120	120	120	120	120	120
Read Or Write	R	R	R	R	R	R	R	R	R
Threads	2	8	16	32	8	16	64	64	64
LatencyMS_Avg	127	31	31	31	15	15	1029	4069	516
MBs_Sec	125	125	125	125	124	124	124	124	123
Ios_Sec	1993	2005	2001	2007	1979	1978	1978	1978	1974

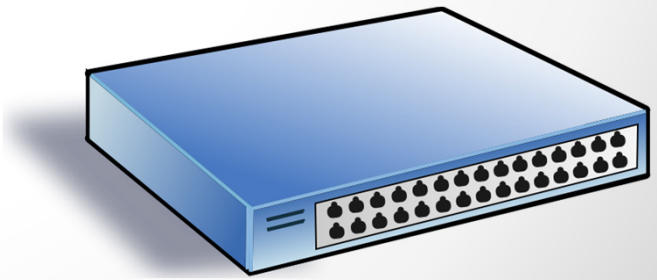
Storage in Azure VM Writes



IO Pattern	random	sequential	sequential	sequential	sequential	random	random	random	sequential	sequential
Sector Size	64	64	64	64	64	64	64	64	64	64
Duration Sec	120	120	120	120	120	120	120	120	120	120
Read Or Write	W	W	W	W	W	W	W	W	W	W
Threads	2	2	32	4	16	8	32	4	8	16
LatencyMS_Sec	20	21	21	21	21	21	21	22	22	11
MBs_Sec	95	93	92	90	90	90	89	88	88	85
Ios_sec	1516	1485	1476	1444	1441	1436	1422	1410	1410	1357

Network Infrastructure In General

- Both platforms allow you to build complex network infrastructure
  - Multiple Private Networks
  - Domain infrastructure
  - Load balancing
- Both platforms can *extend* on-premises infrastructure
- Both platforms can be integrated with on-premises Active Directory infrastructure



Network High Availability

- Network Latency May Vary
  - Different Availability Groups/Data Centers/Regions
- Affects HA/DR and Backup strategies
  - SQL Server Standard Edition does not support async HA technologies
    - Beware of Network Latency with 2 phase commit especially with OLTP systems

Network High Availability

- Log Shipping and Replication would work
  - As long as it can keep up with the load
- Database Mirroring
  - Test network latency especially when servers are in the different regions
- SQL Failover Cluster is not supported
- Always On
  - Network latency with synchronous commit
  - No Automatic Failover in AWS. Unknown in Windows Azure VM
  - AWS: More at: http://www.brentozar.com/archive/2012/07/alwayson-availability-groups-aws-revisited/

Designing Backup Strategy

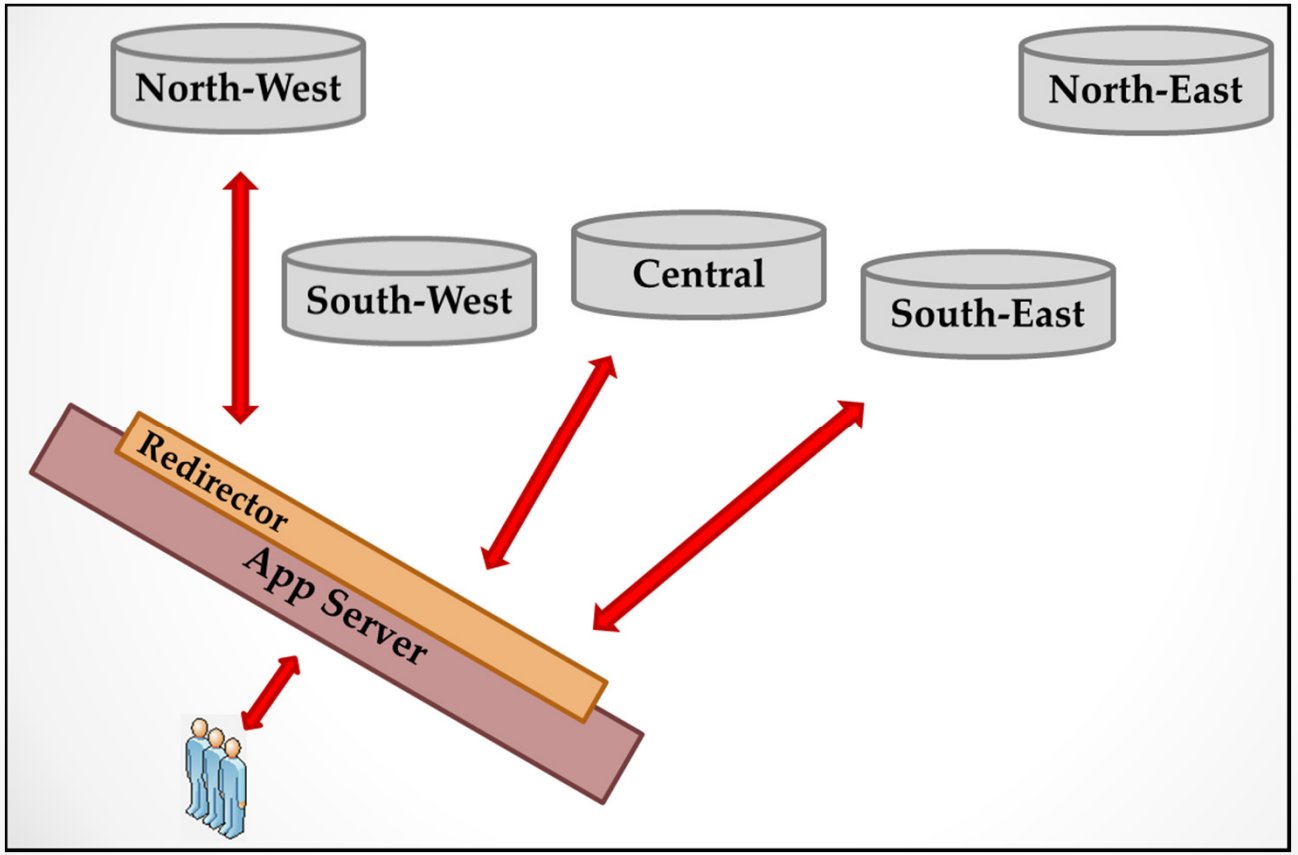
- Snapshot backup
  - Similar to SAN snapshot
- Regular backups
  - Amazon AWS – copy backup files to Amazon S3/Glacier storage
  - Consider IO throughput

Moving systems: On-Premises → IaaS

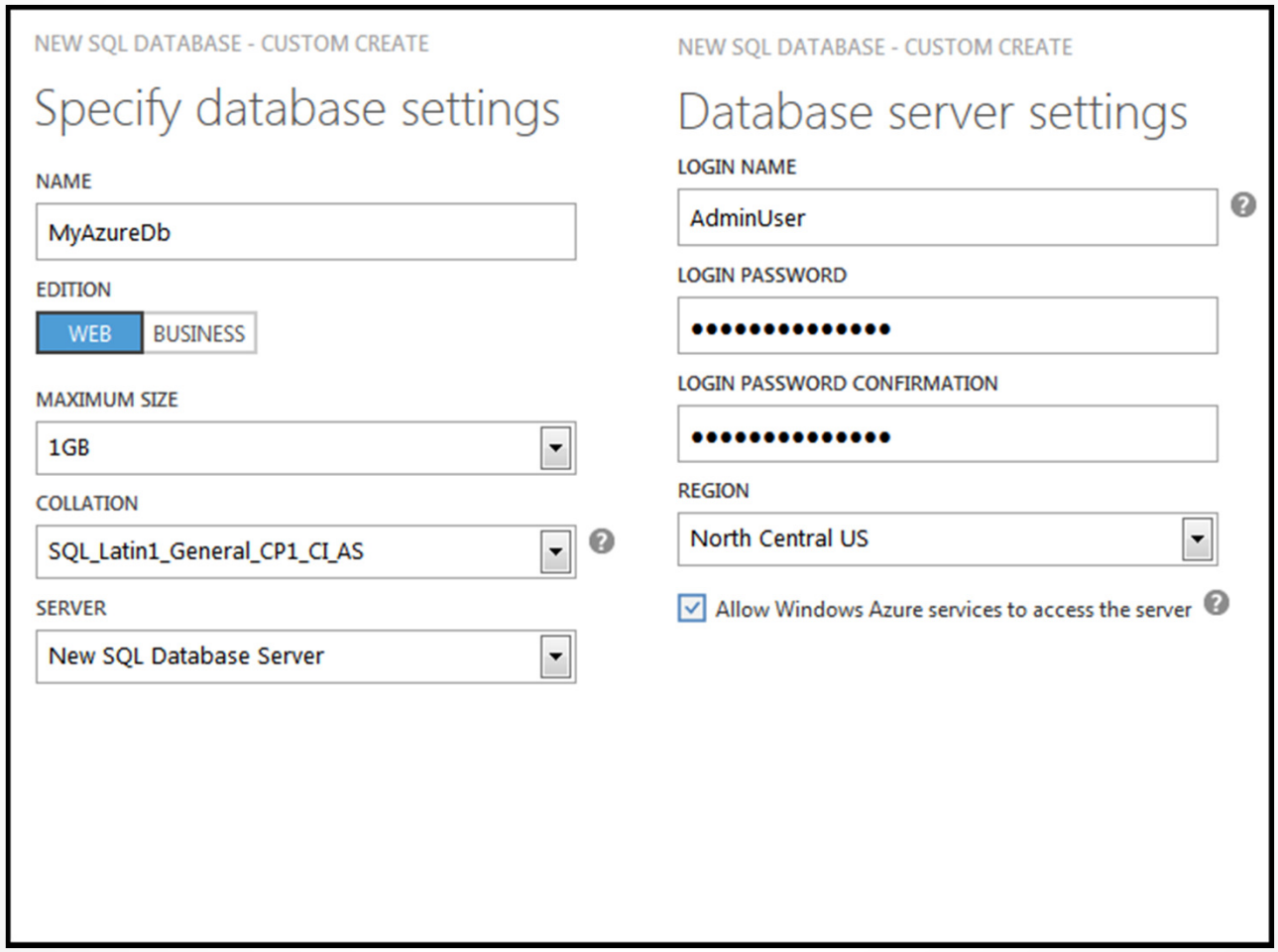
- Features *should* work "as is"
  - Security – Active directory integration
- Take infrastructure limitations into consideration
  - VM scalability limits and cost
  - **IO and Network latency**
- Design High Availability strategy keeping network latency in mind
  - Consider SQL Server Licensing model for HA when choosing VM type
- Design Backup and Disaster Recovery strategies
  - Network latency could be an issue
  - Snapshot backup gives additional flexibility

Additional architecture considerations

- Data Sharding - consider development cost



Windows Azure Sql Database

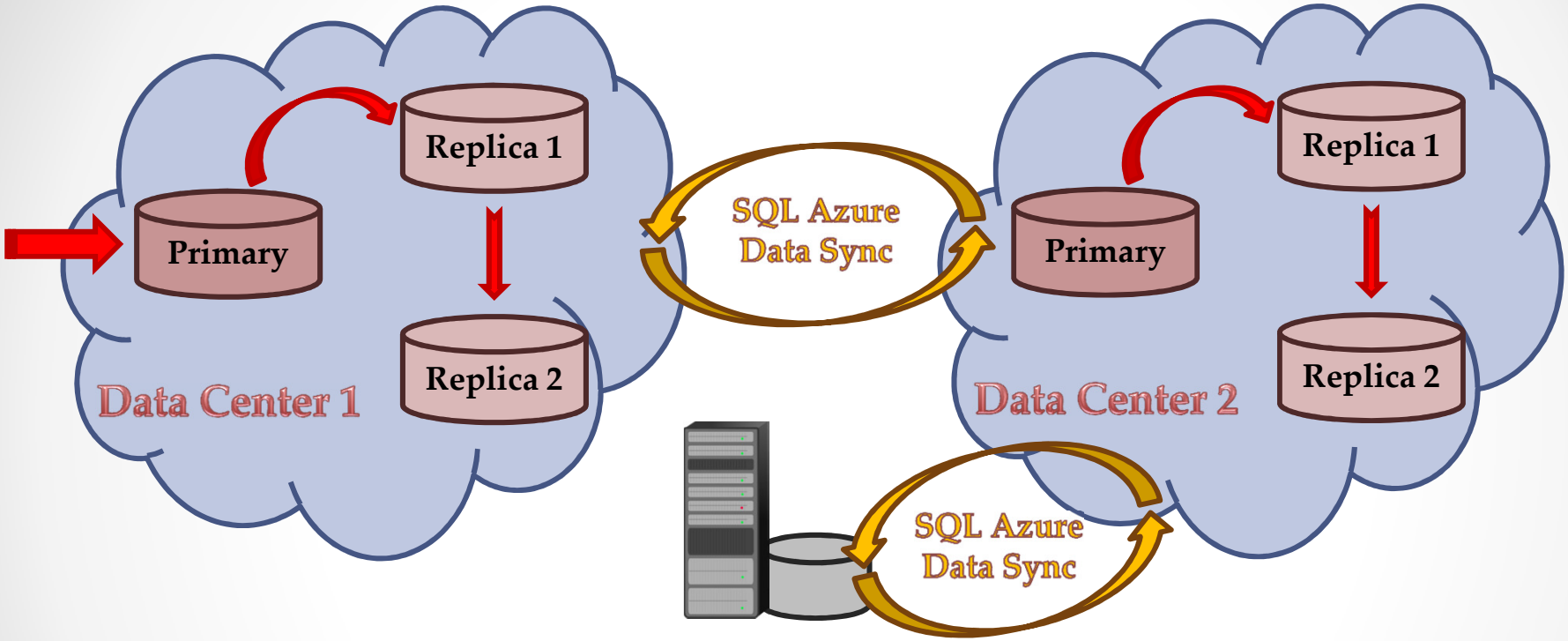


Sounds too good to be true

(Windows Azure SQL Databases)

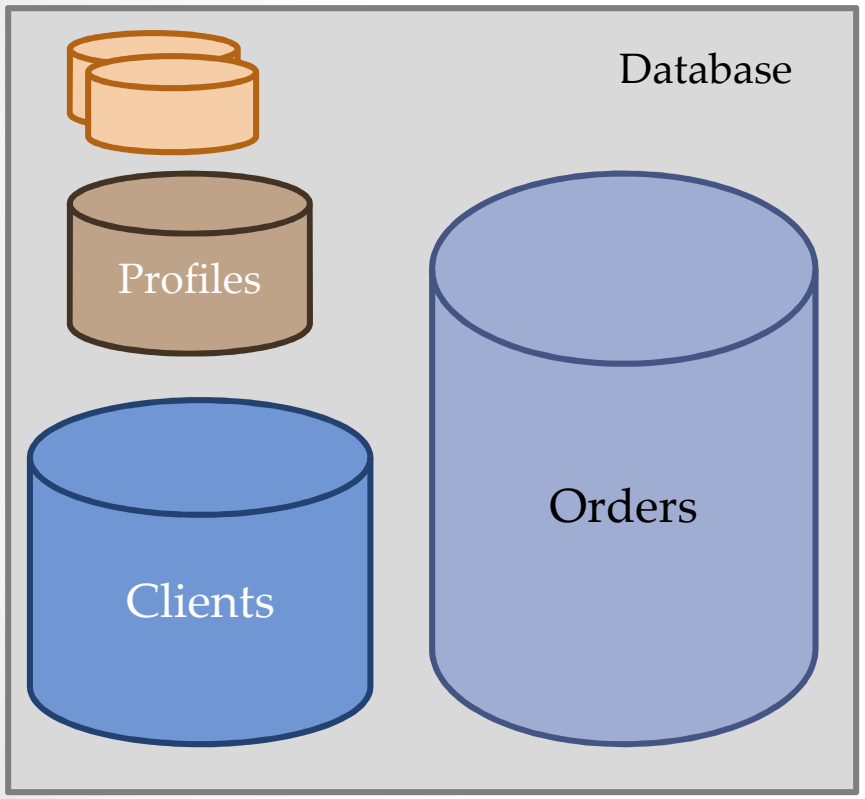
- There is still index maintenance required
  - FILLFACTOR cannot be set
- Database size needs to be monitored
  - Database becomes read-only once capacity limit is reached
  - Think about index fragmentation
- Read Committed Snapshot Isolation level is ON
  - Remember when you move from Azure SQL DB to another platform
- There is no parallelism
  - Think about D/W workload
- **There is the throttling**
  - Think about D/W workload

HA and Backup/Restore



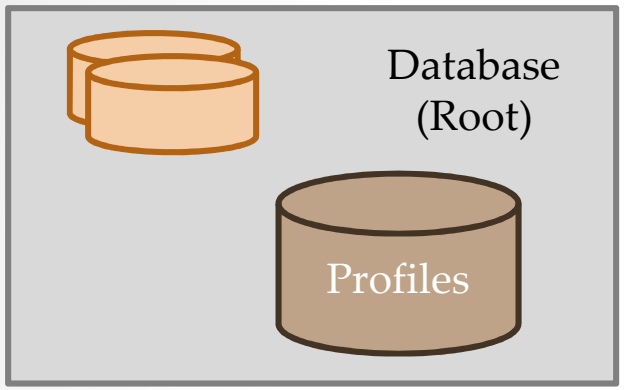
- Each Database has 2 replicas within the same data center
- SQL Azure Data Sync (CTP) replicates data between data centers (Azure and On-Premises)
- Backup / Restore
  - CREATE DATABASE AS COPY OF
  - SQL Database Import/Export Service

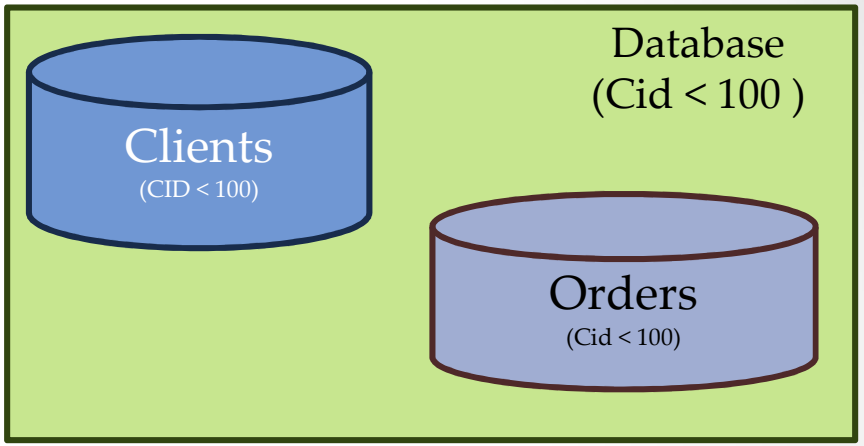
Azure SQL Database Federations

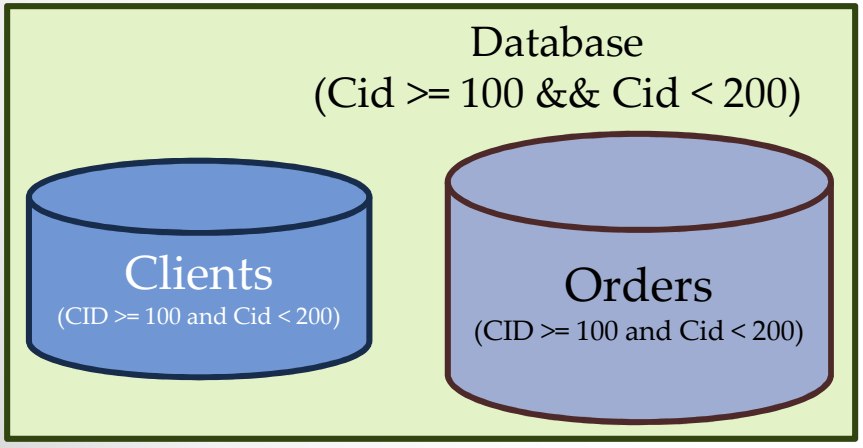


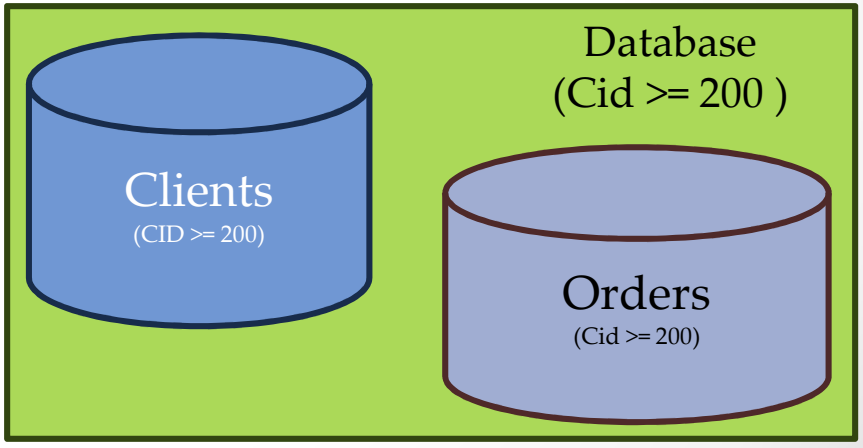

```
CREATE FEDERATION ClientFederation (CID int RANGE)
go

ALTER FEDERATION ClientFederation
SPLIT AT (CID=100)
go

ALTER FEDERATION ClientFederation
SPLIT AT (CID=200)
go
```

Azure SQL Database Federations









Azure SQL Database Federations

- Share-Nothing approach
  - No cross-database / cross-federations joins
  - No fan-out queries
    - Herve Roggero's Enzo Shard Library on CodePlex
- Federation members can *technically* have different schemas
- Carefully choose Federation Key
  - Federate multiple related tables
  - Split in the way that spreads data equally across the members

Amazon RDS IaaS or PaaS

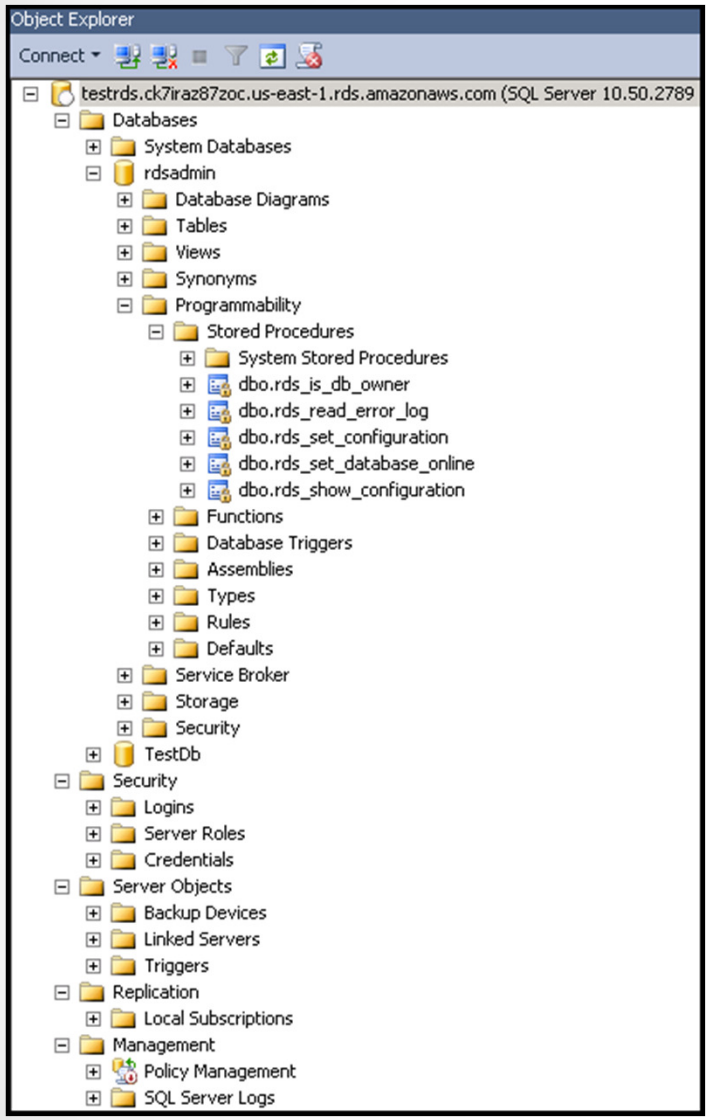
IaaS

- Scalability via configuration
- Database and tables/indexes layout needs to be configured
- Database maintenance is not automated

PaaS

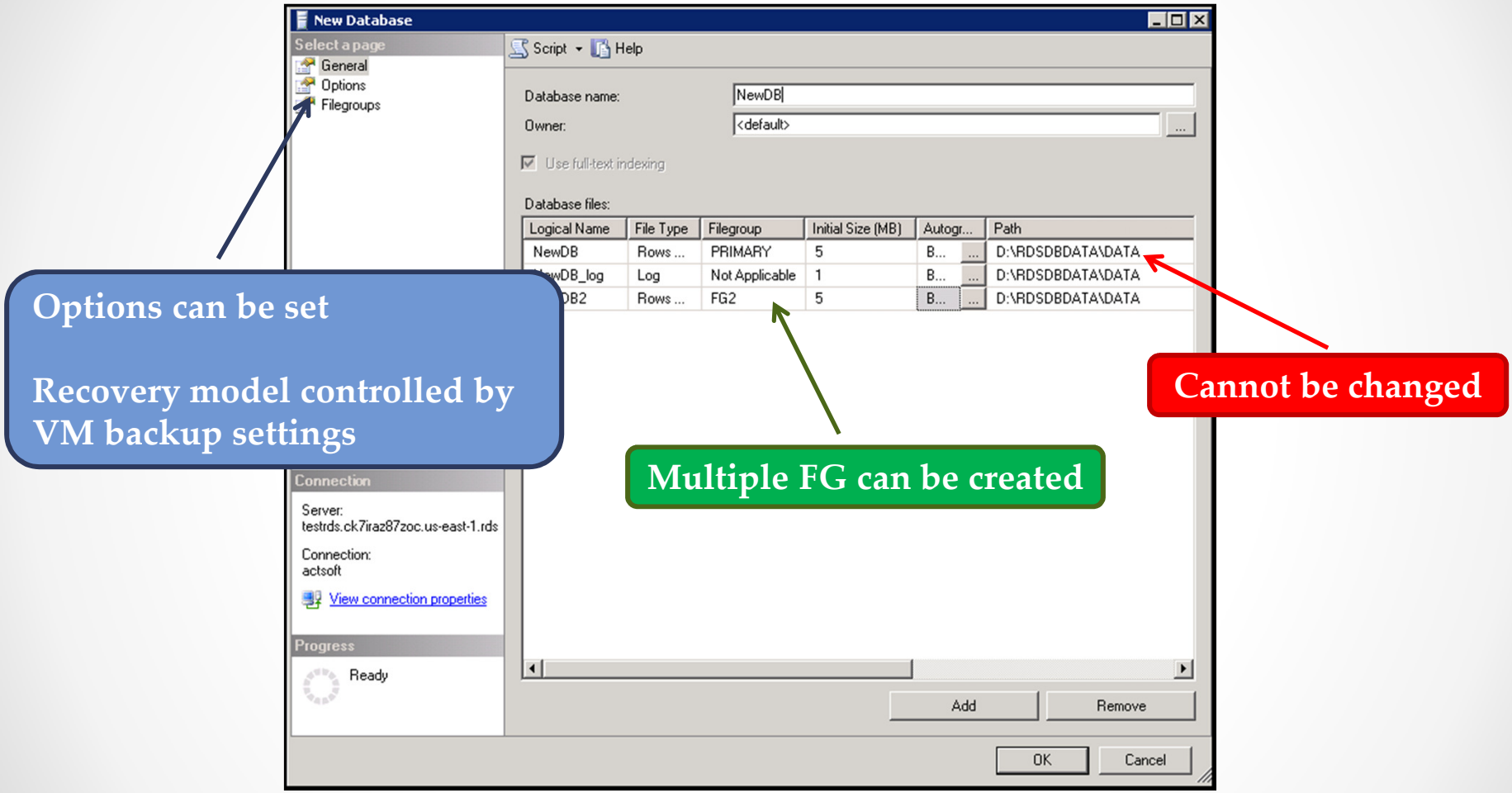
- OS and Storage are abstracted
  - Kind of (more later)
- Automatic OS / SQL Server maintenance and patching
- Automatic Replication
- Automatic Backup and Restore

Amazon RDS



- SQL Server 2008 R2 SP1
- Special database for DBA tasks:
  - Rdsadmin
- 30 Databases Max per RDS instance
- 1TB max for storage
  - Shared across databases
- Disk size needs to be specified during VM creation
  - No disk resize option

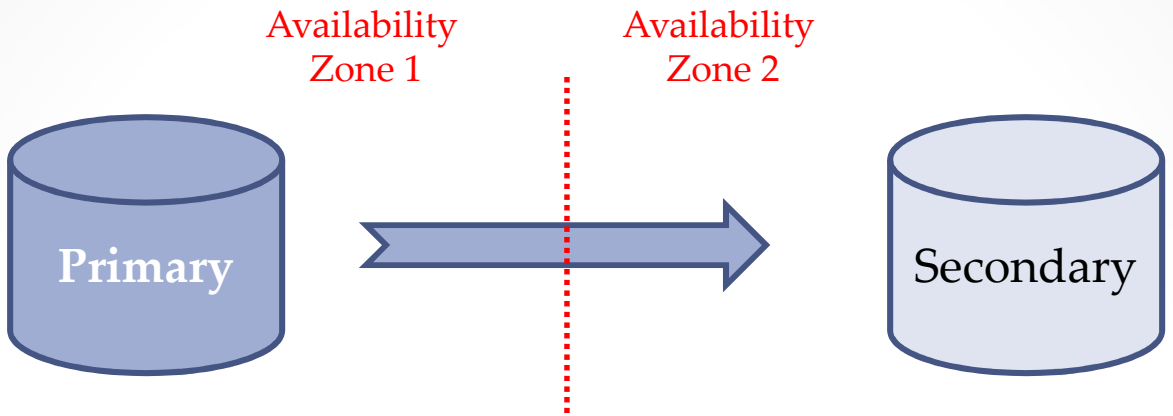
Amazon RDS


New Database
Select a page
General
Options
Filegroups
Script
Help
Database name:
NewDB
Owner:
<default>
Use full-text indexing
Database files:
Logical Name
File Type
Filegroup
Initial Size (MB)
Autogr...
Path
NewDB
Rows ...
PRIMARY
5
B...
D:\RDSDBDATA\DATA
NewDB_log
Log
Not Applicable
1
B...
D:\RDSDBDATA\DATA
DB2
Rows ...
FG2
5
B...
D:\RDSDBDATA\DATA
Connection
Server:
testrds.ck7iraz87zoc.us-east-1.rds
Connection:
actsoft
View connection properties
Progress
Ready
Add
Remove
OK
Cancel
Options can be set
Recovery model controlled by VM backup settings
Multiple FG can be created
Cannot be changed

Amazon RDS

- All SQL Server Editions are supported
  - Two license models
    - License included (all but Enterprise edition)
      - Higher price per hour
    - Bring-your-own-license
- You are responsible for:
  - Database Maintenance
    - Size management
    - *Transaction log truncation*
  - Index Maintenance

RDS - HA and Backup/Restore

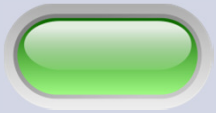
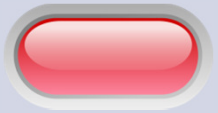
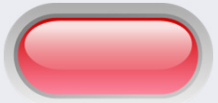
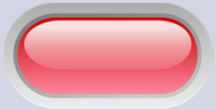
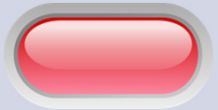
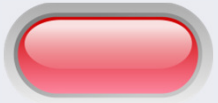
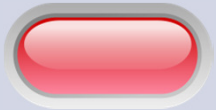
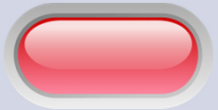
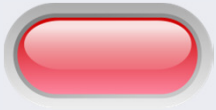
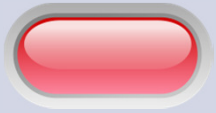
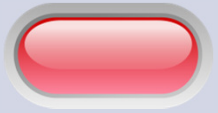


- Optional second instance – replica
  - Cannot be accessed by clients
  - Automatic failover
- Backup via snapshot
  - Suspend IO activity
  - Can be done from the replica (slows down IO activity on primary)
- Ability to set "preferred maintenance and backup windows"

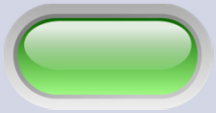
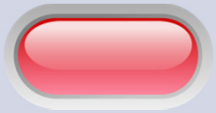
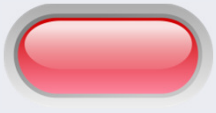
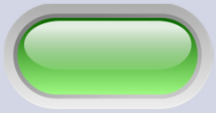
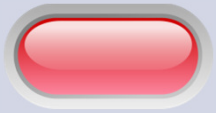
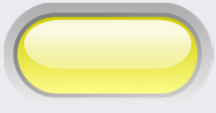
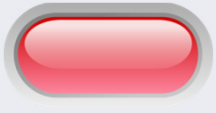
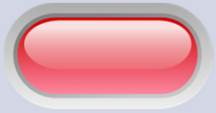
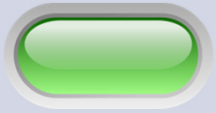
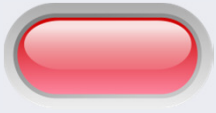
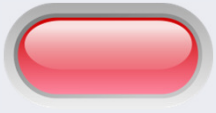
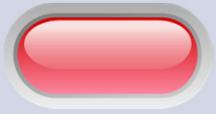
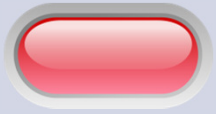
Moving Systems:
On-Premises ↔ Azure / RDS

- Azure / RDS → On-Premises
  - Azure/RDS features are supported by full SQL Server
    - Azure Federations is an exception
  - Maintenance, HA/DR strategies need to be revisited
- On-Premises → Azure / RDS
  - It depends:
    - Performance standpoint - System workload and blueprint
    - Functionality standpoint – SQL features used
      - You can workaround a lot of limitations but consider development code

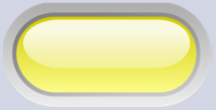
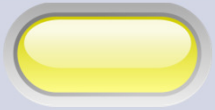
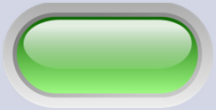
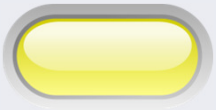
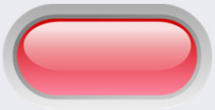
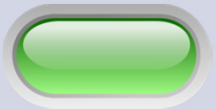
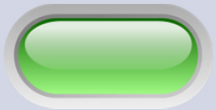
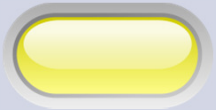
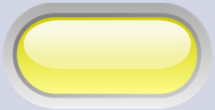
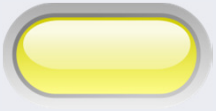
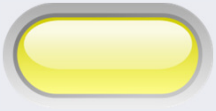
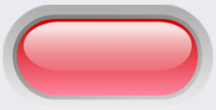
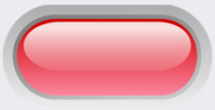
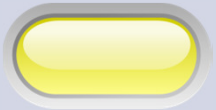
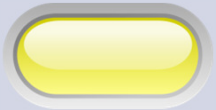
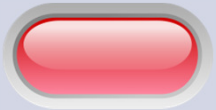
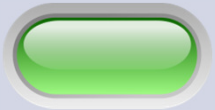
SQL Features

Feature	Amazon RDS	Azure SQL Database
Service Broker		
SQL Server Agent		
Database Mail		
Full-Text Search		
Windows Authentication		
SSL Connections		
TDE		

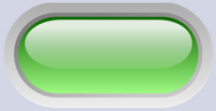
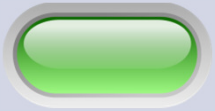
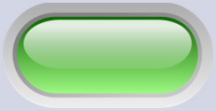
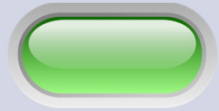
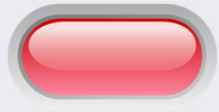
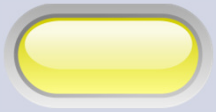
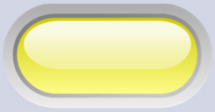
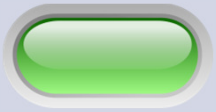
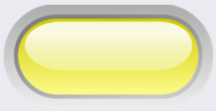
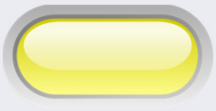
SQL Features

Feature	Amazon RDS	Azure SQL Database
Typed XML XML Indexing		
Safe CLR		
Global temporary tables		
Table Partitioning		
Federations		
Distributed Queries		
FILESTREAM		

Side-by-Side

	Amazon EC2	Windows Azure VM	Amazon RDS	Azure SQL Database
New development (reduce cost as the goal)				
Porting existing systems (depends on the features)				
Data Size				
Small				
Medium				
Large / VLDB				
VLDB Sharding-friendly				

Side-by-Side

	Amazon EC2	Windows Azure VM	Amazon RDS	Azure SQL Database
OLTP Workload				
D/W Workload				
High*er* Availability				
Integration with on-premises infrastructure				

We covered

- IaaS and PaaS cloud models and current offerings
- Platform limitations from performance and functional standpoints

References

- Windows Azure
  - Platform portal: http://www.windowsazure.com
  - Windows Azure Blog: http://blogs.msdn.com/b/windowsazure/
- Amazon Web Services
  - Platform portal: http://aws.amazon.com/
  - Amazon RDS portal: http://aws.amazon.com/rds/
  - Jeremiah Peschka's blog posts: http://www.brentozar.com

- Images from: http://openclipart.org

Q & A

- Thank you for attending!
- Session will be available for download
  - http://aboutsqlserver.com/presentations

- Email: dmitri@aboutsqlserver.com