Errors or corrections? Email us at aws-course-feedback@amazon.com.
Other questions? Email us at aws-training-info@amazon.com.

Architecting on AWS

Choosing a Structured Datastore

Choosing a Structured Datastore | What we'll cover

1. Running a database in the cloud
2. Benefits of managed systems
3. Advantages of DynamoDB
4. Highly-available datastores and AWS

Choosing a Structured Datastore | Running a database in the cloud

1

Running a database in the cloud

Tenets

- There are many RDBMS options
- NoSQL software and systems offer an alternate approach

Tenets

- Storage is the #1 bottleneck for structured datastores
- Trade unit durability for system performance by moving resilience to the software tier
- Follow best practices for HA and DR

Many RDBMS options

- Managed
  - RDS MySQL
  - RDS Oracle
  - RDS SQL Server

Many RDBMS options

- Managed
  - RDS MySQL
  - RDS Oracle
  - RDS SQL Server
- BYOL / DevPay / Licensed Instance
  - Oracle, SQL Server, DB2

Many RDBMS options

- Managed
  - RDS MySQL
  - RDS Oracle
  - RDS SQL Server
- BYOL / DevPay / Licensed Instance
  - Oracle, SQL Server, DB2
- OSS
  - MySQL, PostgreSQL, FirebirdSQL, Interbase, etc

Many RDBMS options

- Managed
  - RDS MySQL
  - RDS Oracle
  - RDS SQL Server
- BYOL / DevPay / Licensed Instance
  - Oracle, SQL Server, DB2
- OSS
  - MySQL, PostgreSQL, FirebirdSQL, Interbase, etc

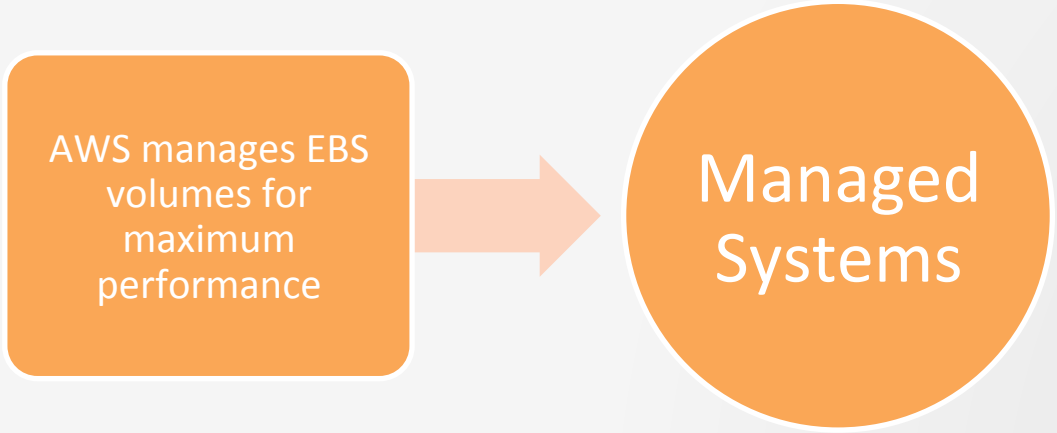
Choosing a Structured Datastore | Benefits of Managed Systems

2

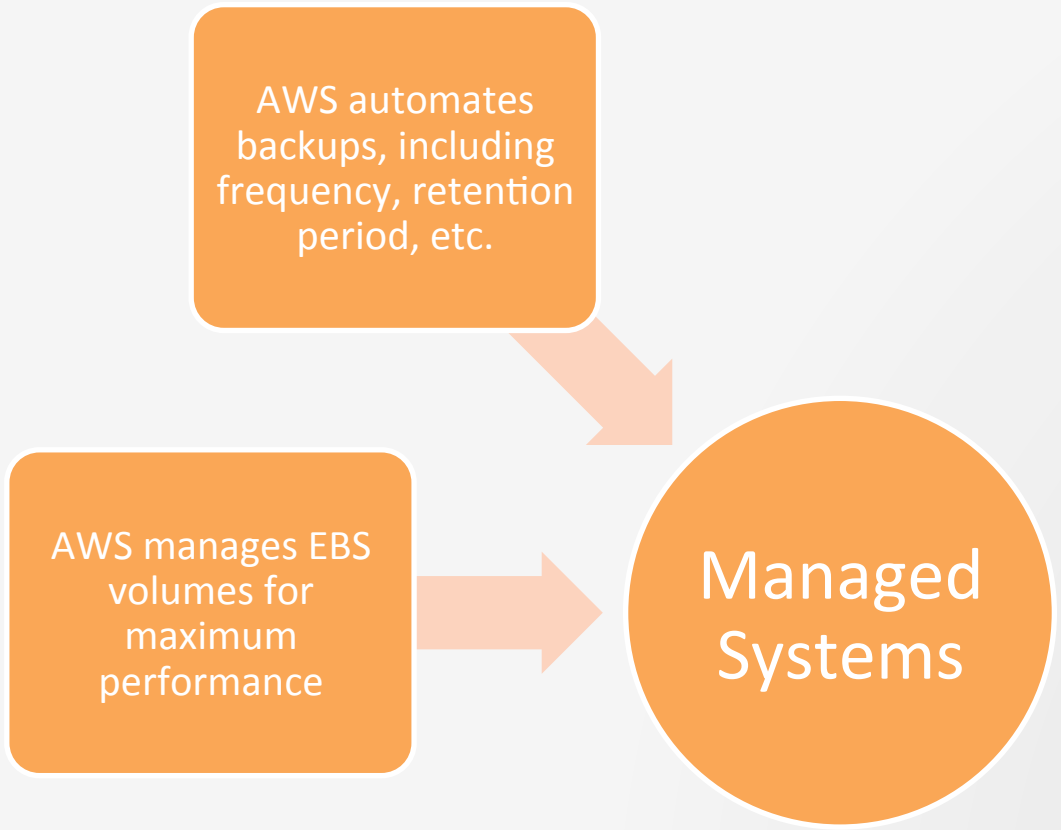
Benefits of managed systems

Choosing a Structured Datastore | Benefits of Managed Systems

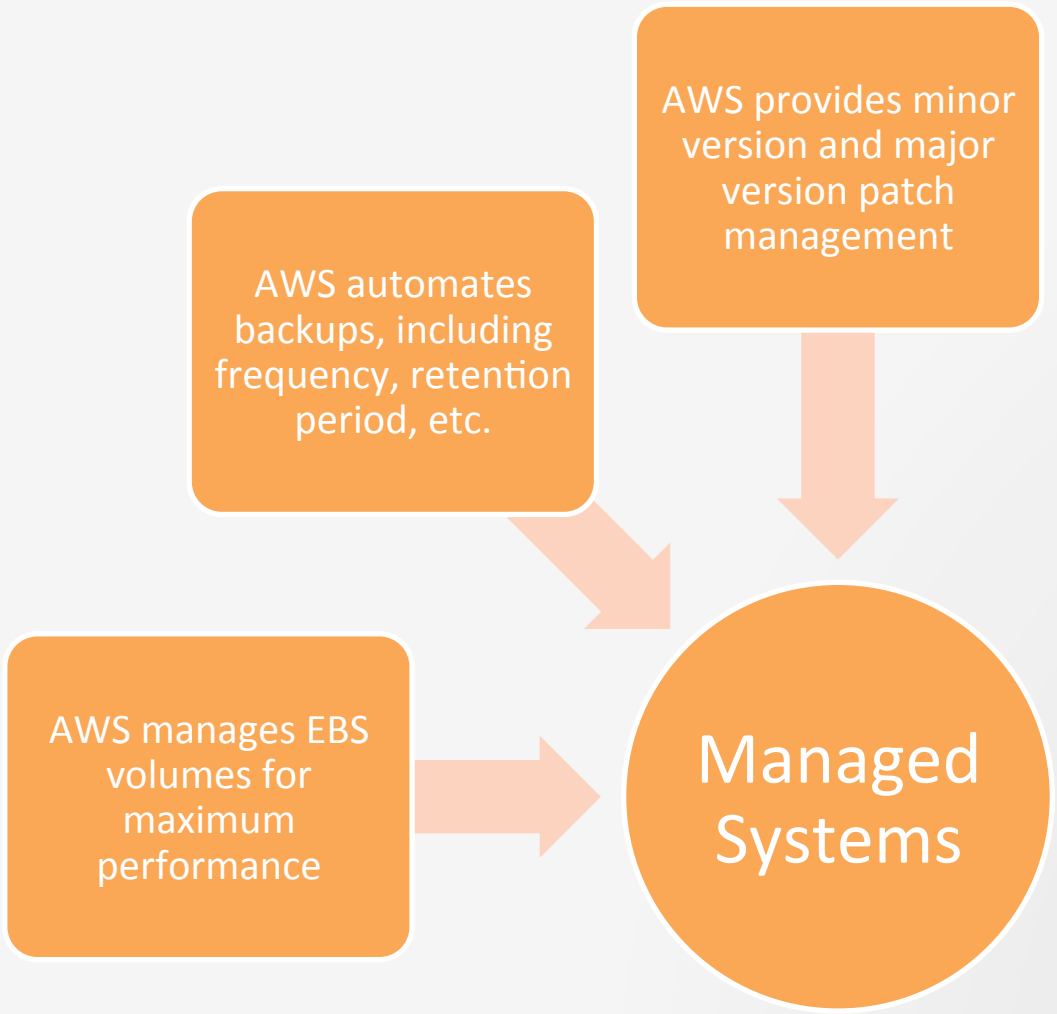
Choosing a Structured Datastore | Benefits of Managed Systems



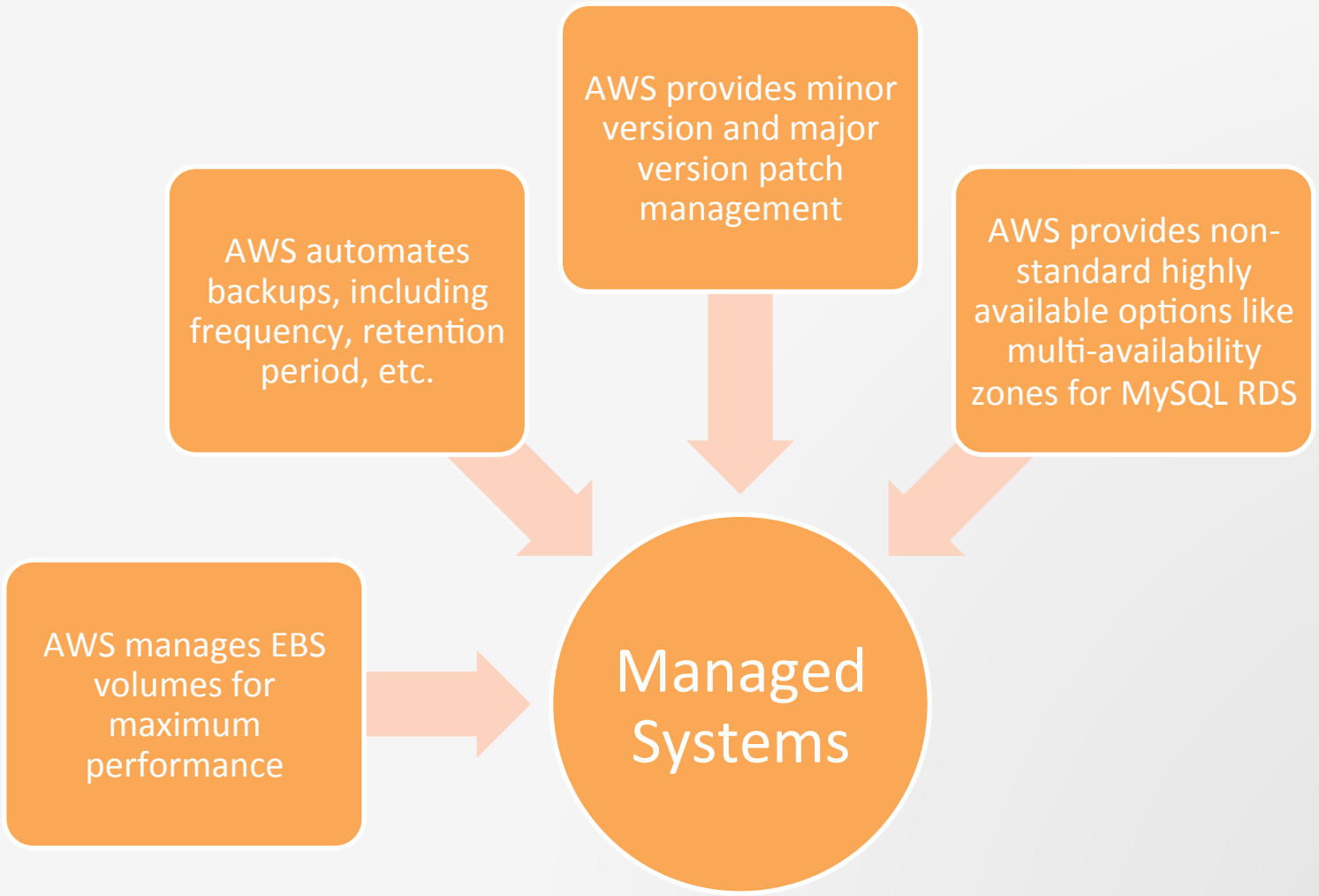
Choosing a Structured Datastore | Benefits of Managed Systems



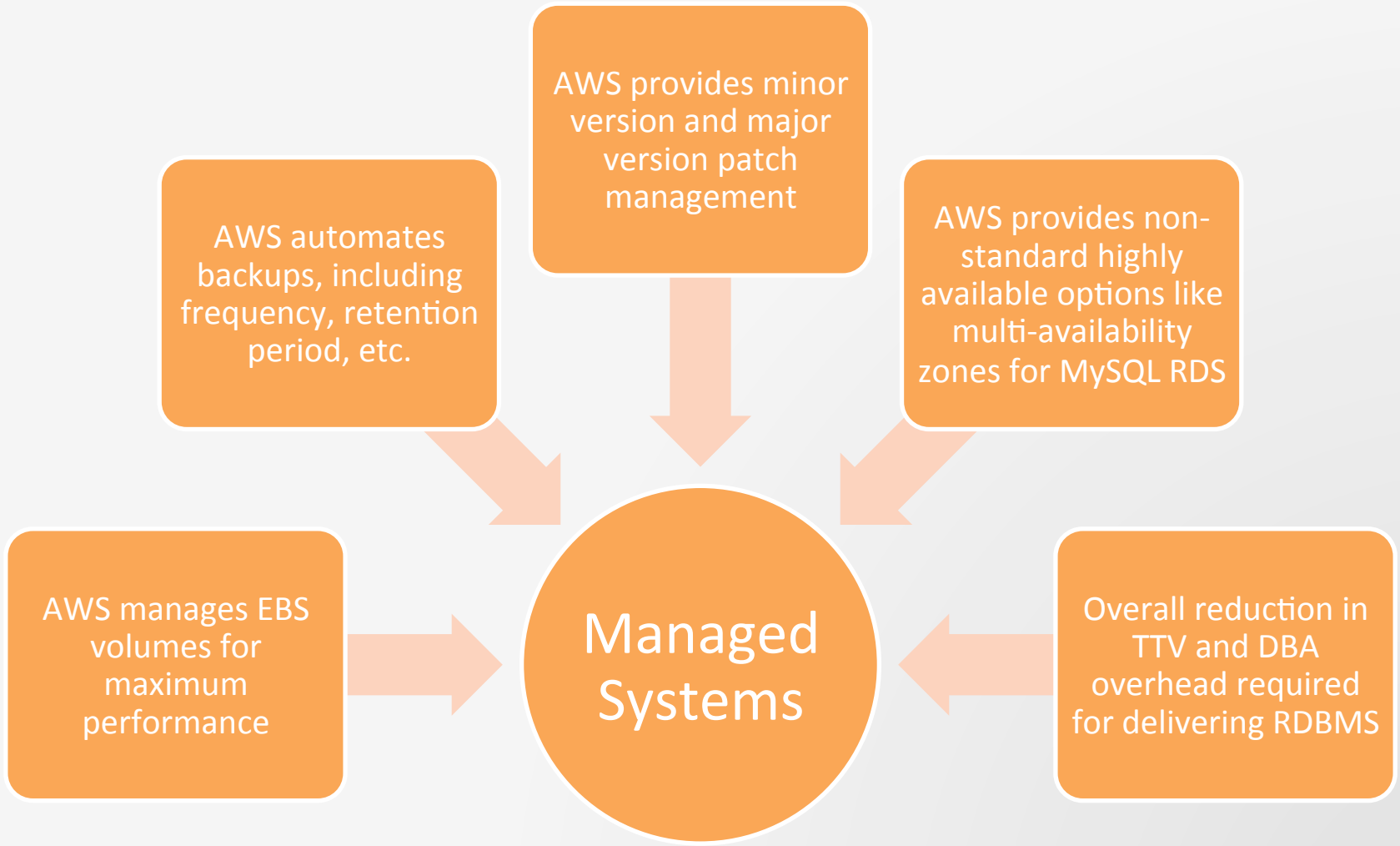
Choosing a Structured Datastore | Benefits of Managed Systems



Choosing a Structured Datastore | Benefits of Managed Systems



Choosing a Structured Datastore | Benefits of Managed Systems



Choosing a Structured Datastore | Advantages of DynamoDB

3

Advantages of DynamoDB

NoSQL Approaches

- Managed NoSQL
  - DynamoDB
  - S3

NoSQL Approaches

- Managed NoSQL
  - DynamoDB
  - S3
- OSS NoSQL
  - Cassandra, CouchDB, Riak, MongoDB, Hbase, Redis, etc…

NoSQL Approaches

- Managed NoSQL
  - **DynamoDB**
  - S3
- OSS NoSQL
  - Cassandra, CouchDB, Riak, MongoDB, Hbase, Redis, etc…

What is Amazon DynamoDB

- Fully managed AWS non-relational (NoSQL) database service

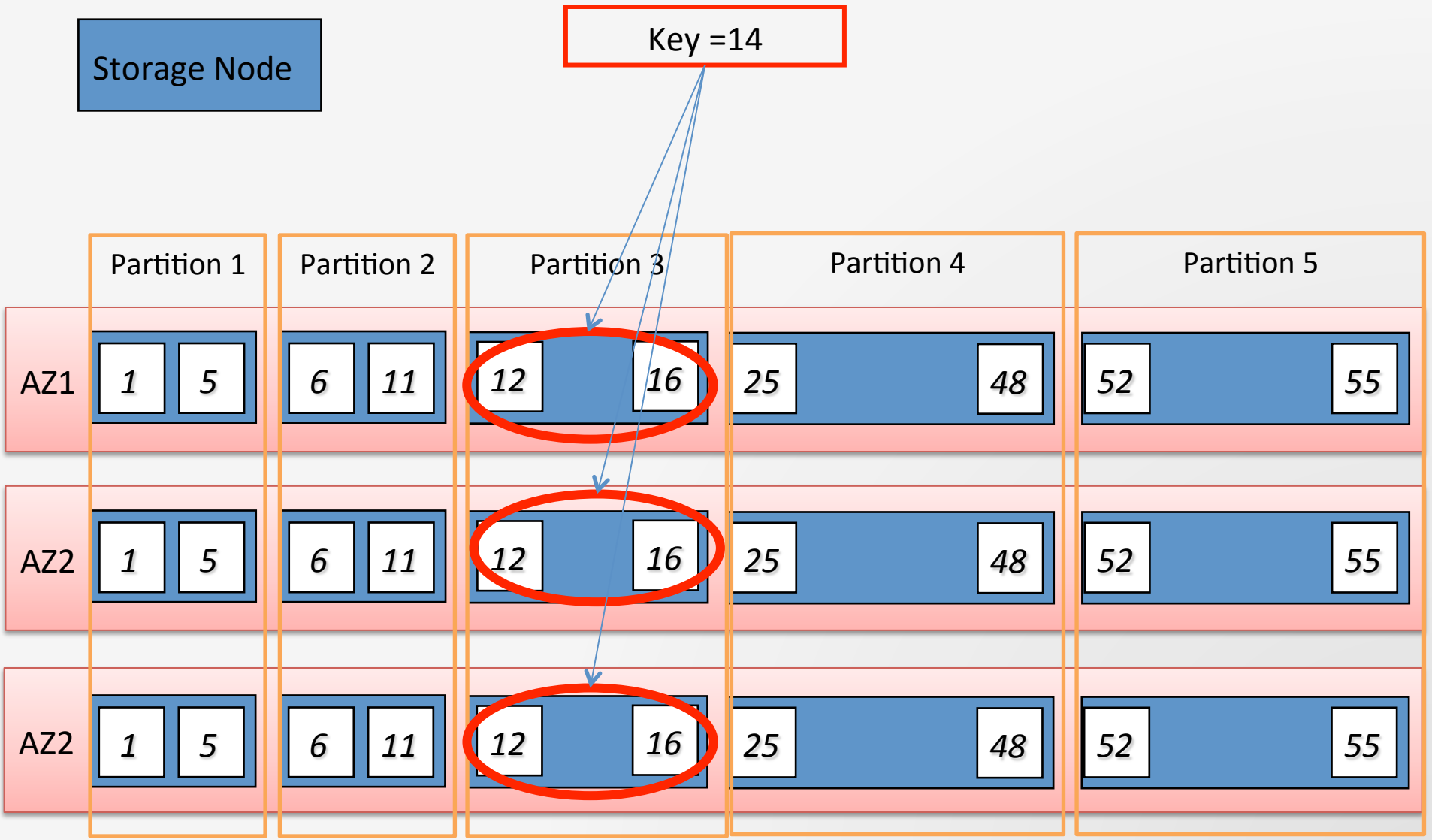
Scale, durability, availability

- Massive and Seamless Scale
  - No table size & throughput limits
    - Apart from 64K row size limit
  - Live/auto repartitioning for size and throughput increase i.e. client-side sharding not required
- High Durability and Availability
  - 3 way on disk replication across the AZs
  - Puts are synchronously & durably persisted to AZs before a successful response

Performance, latency

- Predictable performance
  - Provisioned throughput model
- Low latency
  - SSD-based Storage Nodes
  - Single digit latency

Choosing a Structured Datastore | Advantages of DynamoDB



Partitioning architecture

- Architecture lends to seamless scale
  - Partition 100% independent parallel computation unit
  - No support for Multi-Statement transactions
  - No distributed coordination across partitions
- More partitions -> More size and throughput
  - As long as workload distribution is well spread
- Live repartitioning makes service elastic

Partitioning and indexing

- Data is indexed and partitioned by Primary Key
  - PK defined by user at table creation time
  - Partitioning handled by DynamoDB and abstracted from user
- Queries allowed **only** against Primary Key
  - Scan API targeted at troubleshooting & scanning a table from beginning to end.

Data Model

- Sparse schema support
  - Table = Collection of Items
  - Item = Arbitrary collection of Name-Value attribute pairs
  - Primary Key only mandatory attribute(s)
  - 64KB Item Size limit

Data types

- Native data type support
  - UTF-8 String (S)
  - Number (N)
  - String Set (SS) i.e. Multi-Valued Attributes
  - Number Set (NS)

Index choices

- Single Hash Key
  - Distributed Hash Table abstraction
  - Hashing maximizes workload distribution
  - No Query Support: (Batch)Put, (Batch)Get, Scan only
- Composite Hash, Range Key
  - Items grouped by Hash Key & sorted by Range Key
  - Hash Groups have no size limits

Table with single hash key

- Supported data plane API
  - GetItem, PutItem, UpdateItem, BatchWriteItem, BatchGetItem, Scan

Required | Optional

SSN (hash key)	Name	Phone	Gender
123-123-4567	Joe Sherman	206-777-1234 (mobile) 206-123-8745 (home)	
345-123-8967	Jack Lace		Male
123-098-1234			

Table with composite hash, range key

- Supported data plane API
  - GetItem, PutItem, UpdateItem, BatchWriteItem, BatchGetItem, **Query** & Scan

Required attributes

Optional attributes

OrderID (hash key)	Date (range key)	Recipient	Items	Price	Order Total
104-4786594-1580225	2011/11/26	John Doe	ASIN:B004YIB6DO ASINA004YIB6DO	30.07 49.00	79.07
103-7937382-2727449	2011/11/22	Jack Mar	ASIN:B005MW1118	125.77	125.77

Query patterns

- GetItem
  - Only works on a table with a single Hash key
- Query
  - Only works on a table with a composite hash, range key
  - Hask key = ‘xxxxx’ and range key EQ, GT, LT, GE, LE, BEGINS_WITH, BETWEEN
- Scan
  - Works on both types of tables
  - No index used
  - Starts at the beginning of table and applies filters

Limits (No Limits?)

- Table Size: **no limit**
- Throughput (i.e. reads/sec, writes/sec): provisioned by user, **no limit**
- Item (Row size): 64K (includes attribute name and value sizes)
- Hash Key value: 2048 bytes
- Range key value: 1024 bytes

Choosing a Structured Datastore | Advantages of DynamoDB

Any questions about DynamoDB?

Choosing a Structured Datastore | Highly-available datastores and AWS

4

Highly-available datastores and AWS

Storage: #1 bottleneck for structured datastores

- EBS provides up to 2,000 IOPS per volume
  - Many volumes can be used
  - Separate data from logs and swap; separate EBS not just separate partition.
  - RAID0 increases aggregate performance but reduces durability
  - RAID5 is not recommended in most cases due to network overhead

Storage: #1 bottleneck for structured datastores

- EBS provides up to 2,000 IOPS per volume
  - Many volumes can be used
  - Separate data from logs and swap; separate EBS not just separate partition.
  - RAID0 increases aggregate performance but reduces durability
  - RAID5 is not recommended in most cases due to network overhead
- Ephemeral disk for Swap
  - Reduces EBS IO consumption, saved for log/data
  - Only if TempDB isn't improperly storing critical values!

How would you lay out a disk for a 2TB database?

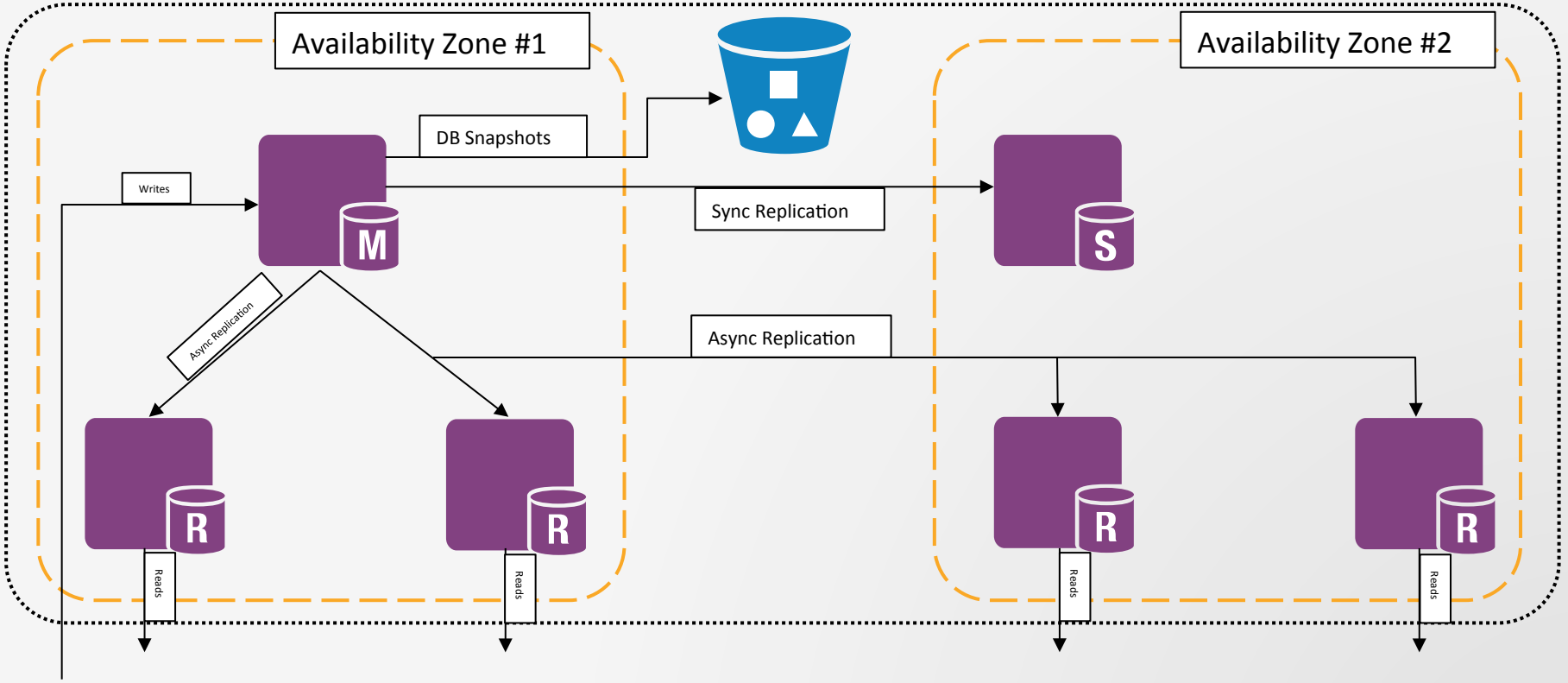
Storage: #1 bottleneck for structured datastores

- Some customers choose higher performance/lower durability of RAID0
  - Trading durablity for system performance
  - **Must move resilience to the software tier**

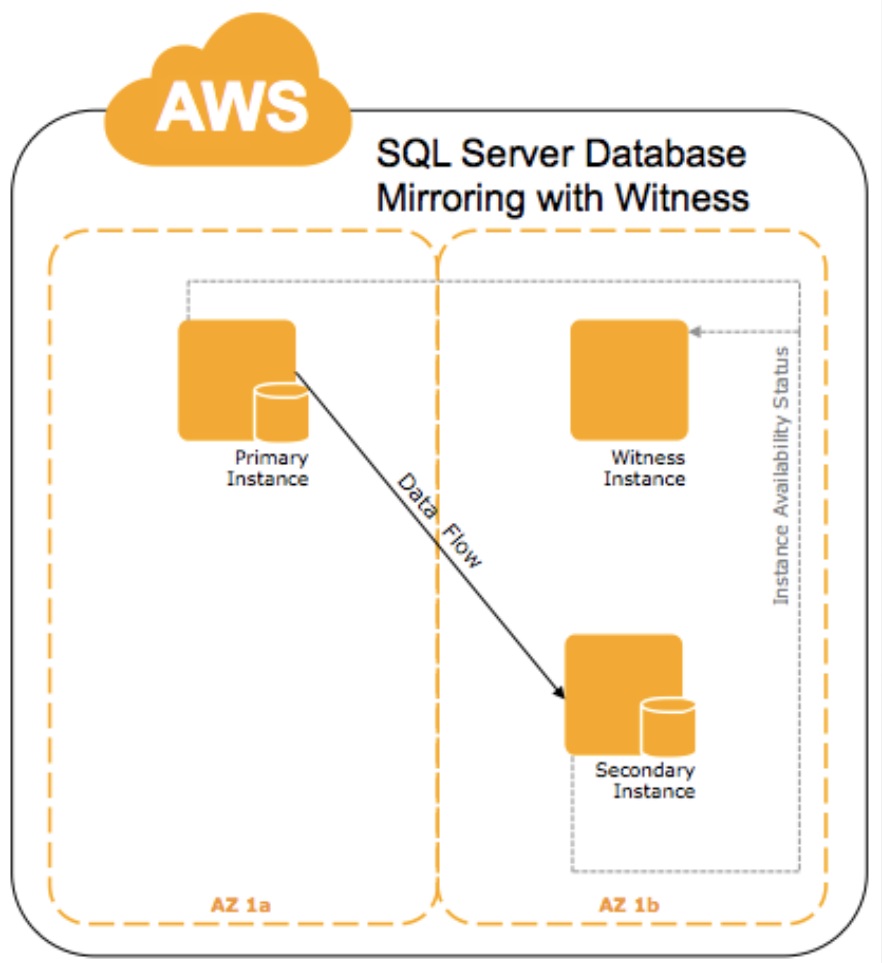
Storage: #1 bottleneck for structured datastores

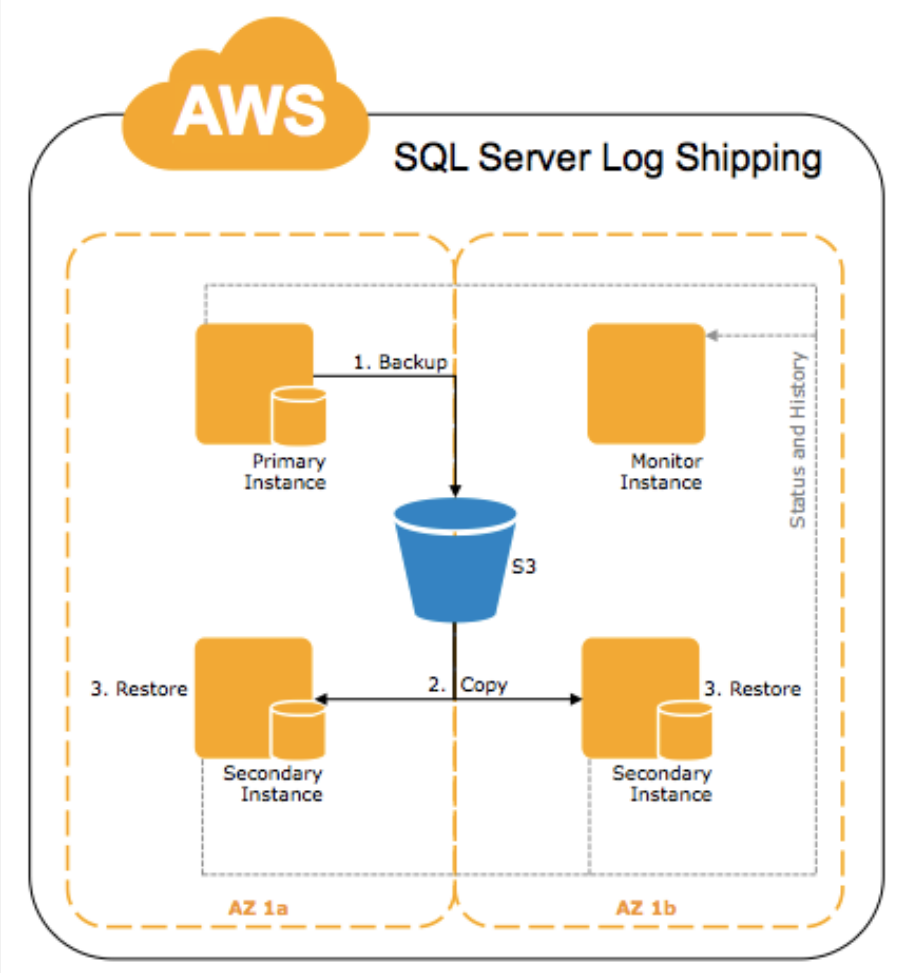
- Some customers choose higher performance/lower durability of RAID0
  - Trading durablity for system performance
  - **Must move resilience to the software tier**
- Each DB has features for HA, like mirroring, clustering, etc, which mitigate the impact of individual system failure.
  - E.g., SQL Server mirroring, MySQL async replication

Tenets in Action: Managed RDB

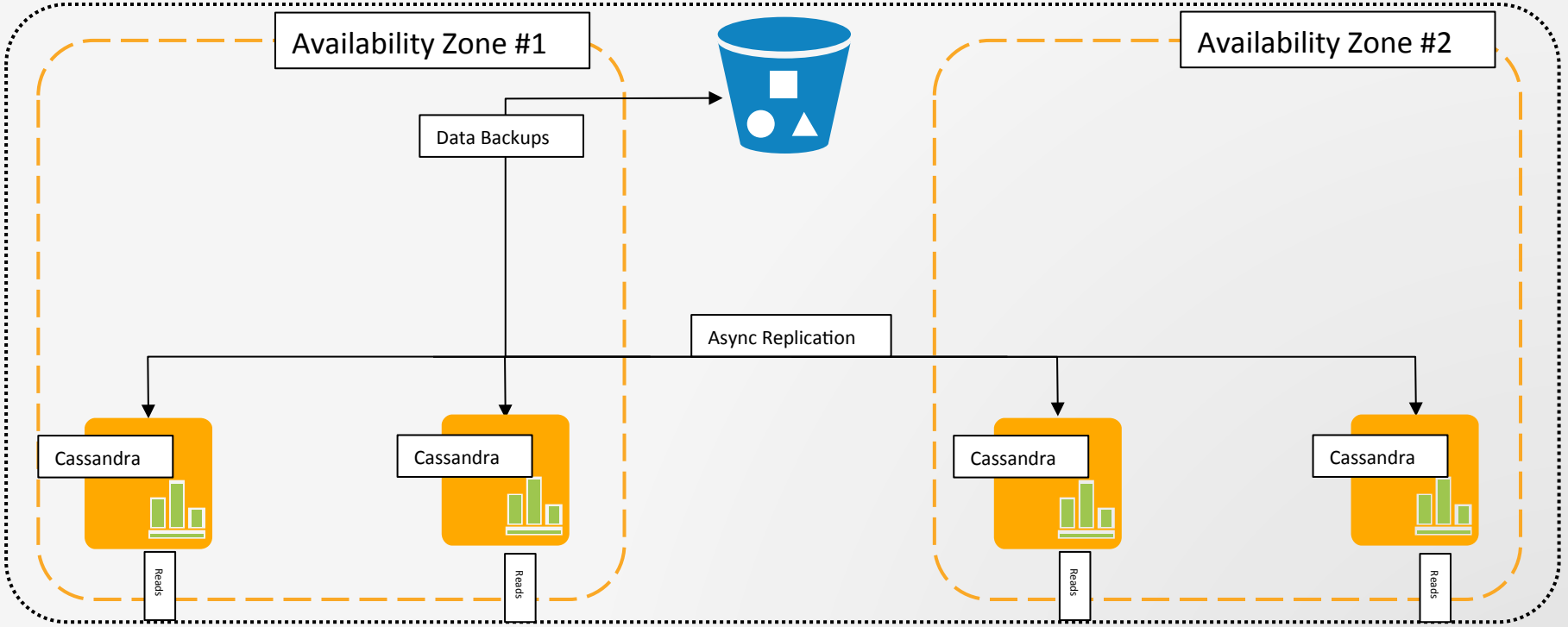


Tenets in Action: SQL Server on EC2





Tenets in Action: SQL Server on EC2



Choosing a Structured Datastore | Conclusion

For review:

- 3 methods of running a database in the cloud
- 5 benefits of managed systems
- 4 benefits of DynamoDB
- How to configure a highly-available datastore in AWS