

# Amazon Web Services

DynamoDB, Redshift, Kinesis and Lambda

Markus Dale, November 2015

# Managed Services With Rich Environment

- ▶ EC2 - Elastic Load Balancer service, Auto-Scaling, Security groups...
- ▶ S3 - Server-Side Encryption, Versioning, Notification (Object created/removed)...
- ▶ RDS - Relational Database Service - managed RDBMS
  - ▶ Supports Aurora, MySQL, PostgreSQL, Maria, Oracle, SQL Server
  - ▶ Multi-AZ, read replicas, backups

Developer productivity - mission/business over infrastructure

# Right Tool for the Right Job

- ▶ When you need that corkscrew...
- ▶ DynamoDB - NoSQL database
- ▶ Redshift - data warehouse
- ▶ Kinesis - Streaming lots of data
- ▶ Lambda - run code on event or timer

# AWS DynamoDB

- ▶ Managed NoSQL database since 2012
  - ▶ Dynamo key-value paper 2007 (open source Cassandra, Riak...)
- ▶ Right tool: High write throughput, query on few attributes, evolving schema, small rows
- ▶ Cost: Provisioned read/write throughput and actual storage
- ▶ Scalable (up and down via API)
- ▶ fault-tolerant: synchronously replicate across multi-AZ
- ▶ SSD storage
- ▶ No updates/patching

# DynamoDB Tables

- ▶ Tables have primary key with optional range key (sort key)
- ▶ Use primary key to determine partition (should have high variability!)
- ▶ Can query on primary key (and range key)
- ▶ Each item (~ row) has unique primary key + range key, attribute (key/value)
- ▶ No fixed schema (other than primary key)
- ▶ 400KB per item (store reference to BLOBs in S3)
- ▶ Primary key: String, Number, or Binary
- ▶ Attributes
  - ▶ Number, String, Binary, Boolean, and Null.
  - ▶ Document types – List and Map.
  - ▶ Set types – String Set, Number Set, and Binary Set

# DynamoDB Provisioned Capacity

- ▶ Read capacity: Number of item reads per second  $\times$  4 KB item size (x2 for eventually consistent)
- ▶ Write capacity: Number of item writes per second  $\times$  1 KB item size

# DynamoDB Sample Table - Reply

- ▶ Hash Key: Amazon DynamoDB#DynamoDB Thread 1  
(denormalized - reference Forum table)
- ▶ Range Key: ReplyDateTime
- ▶ Scan (with filter)
- ▶ Query - by hash (with optional range)

# DynamoDB Query

- ▶ On Hash Key
- ▶ On Hash Key, Range Key
- ▶ Can add filter
- ▶ Secondary Index - local/global



# Amazon Redshift

- ▶ Fully managed, petabyte-scale data warehouse
- ▶ Online analytic processing (OLAP) and business intelligence (BI) - complex queries at scale
  - ▶ Joins!
- ▶ Use SQL (PostgreSQL 8.0.2 syntax)
- ▶ Single or multi-node cluster in a single AZ
- ▶ Continuously backed up to S3
- ▶ Can also enable manual/automated snapshots
- ▶ \$1,000 per terabyte per year (often 3x compression - \$333/TB/year)

# Redshift Architecture

- ▶ Client (JDBC/ODBC) to Leader node
- ▶ One to many compute nodes (same AZ)
- ▶ Massively parallel processing (MPP) - query optimizer
- ▶ Columnar data storage

# Redshift Integration

- ▶ Load/store in S3
- ▶ DynamoDB
- ▶ Elastic MapReduce
- ▶ Kinesis