Databases on AWS
Agenda

• AWS Database Services
• Traditional vs AWS Data services model
• Amazon RDS
• Redshift
• DynamoDB
• ElastiCache
AWS Database Services

- **Amazon RDS**: Managed Relational Database Service
- **Amazon Redshift**: Petabyte-scale Data Warehouse Service
- **DynamoDB**: Fast & Scalable NoSQL Service
- **ElastiCache**: In-memory Caching Service
Traditional Database Architecture

- Client Tier
- App/Web Tier
- RDBMS

*one database for all workloads*
Traditional Database Architecture

- Key-value access
- Complex queries
- OLAP transactions
- Analytics

All forced into the relational database
On AWS choose best database service for each workload.

AWS Data Tier Architecture

- Client Tier
- App/Web Tier
- Data Tier
  - Cache
  - NoSQL
  - Data Warehouse
  - RDBMS
  - Blob Store
  - Search
Workload Driven Data Store Selection

- Cache
- Data Warehouse
- Blob Store
- NoSQL
- RDBMS
- Search

- Hot reads
- Complex queries & transactions
- Logging
- Key/value & transactions
- Simple query
- Analytics
- Rich search
AWS Database Services for the Data Tier

- Amazon DynamoDB: key/value & transactions
- Amazon RDS: complex queries & transactions
- Amazon ElastiCache: hot reads
- Amazon Redshift: analytics
- Amazon S3: logging, rich search
- Amazon CloudSearch: analytics

Data Tier
Relational Databases

MySQL, Aurora, PostgreSQL, Oracle, SQL Server, MariaDB

Fully managed; zero admin
If you host your databases on-premises

- App optimization
- Scaling
- High availability
- Database backups
- DB s/w patches
- DB s/w installs
- OS patches
- OS installation
- Server maintenance
- Rack & stack
- Power, HVAC, net
If you host your databases in Amazon EC2
If you choose Amazon RDS

- Scaling
- High availability
- Database backups
- DB s/w patches
- DB s/w installs
- OS patches
- OS installation
- Server maintenance
- Rack & stack
- Power, HVAC, net

App optimization
## The Self-managed vs. AWS-managed decision

<table>
<thead>
<tr>
<th>Self-managed database</th>
<th>AWS-managed database</th>
</tr>
</thead>
<tbody>
<tr>
<td>You have full responsibility for upgrades and backup</td>
<td>AWS provides upgrades, backup, and failover as a service</td>
</tr>
<tr>
<td>You have full responsibility for security</td>
<td>AWS provides high infrastructure security, certifications; gives you tools to ensure DB security</td>
</tr>
<tr>
<td>Full control over parameters of server, OS, and database</td>
<td>Database is a managed appliance, so you can easily automate</td>
</tr>
<tr>
<td>Replication is expensive, complex, and requires a lot of engineering</td>
<td>AWS provides failover as a packaged service</td>
</tr>
</tbody>
</table>
## Key Amazon RDS Features

<table>
<thead>
<tr>
<th>Amazon RDS Configuration</th>
<th>Improve Availability</th>
<th>Increase Throughput</th>
<th>Reduce Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push-Button Scaling</td>
<td></td>
<td>![Checkmark]</td>
<td></td>
</tr>
<tr>
<td>Multi AZ</td>
<td>![Checkmark]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read Replicas</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td></td>
</tr>
<tr>
<td>Provisioned IOPS</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
</tr>
</tbody>
</table>

**Push-Button Scaling**

- Multi-AZ:
  - **M**: Availability zone
  - **S**: Availability zone

**Provisioned IOPS**
Amazon RDS -- Aurora

- Amazon Aurora is a MySQL-compatible relational database engine
  - Built from the ground up to leverage AWS
  - Speed and availability of high-end commercial databases
  - Simplicity and cost-effectiveness of open source databases

- Retains compatibility with MySQL 5.6

- Up to 5x better performance than MySQL

- At a price point 1/10 of a commercial database

- Data is transparently replicated 6 ways among 3 Availability Zones

- Encryption at rest and in transit

- Add up to 15 Replicas

- Compatibility with PostgreSQL New!
Petabyte scale
Massively parallel
Columnar Store
Relational data warehouse
Fully managed = no admin

*for as low as $999/TB per year*
Amazon Redshift

- Fast and powerful, petabyte-scale data warehouse
  - Fully managed Relational Database
  - High-parallel
  - Columnar Data Store

- Data warehouse-type queries
  - Aggregations, historical analysis
  - BI Tool integration

- Grow with your data
  - 160 GB → 1.6 PB

- SSD and SAS Options
  - SSD provides 10-15x perf @ 5.5x the cost per TB/year
NoSQL Database
Seamless scalability
Zero admin
Single digit millisecond latency
Amazon DynamoDB

- Fully managed NoSQL database service
- Massively scalable, distributed key/value store
- Reserved capacity model
- Fast and predictable
- Built-in fault tolerance
- Strong consistency model
- Unlimited potential storage and throughput
NoSQL vs. SQL for a new app: how to choose?

- Want simplest possible DB management?
- Want app to manage DB integrity?

- Need joins, transactions, frequent table scans?
- Want DB engine to manage DB integrity?
- Team has SQL skills?

Amazon DynamoDB

Amazon RDS
Amazon ElastiCache

- In-memory cache in the cloud
- Improve latency and throughput for read-heavy workloads
- Supports open-source caching engines
  - Memcached
  - Redis
- Fully managed
- Multi-AZ

Examples
- Caching of MySQL database query results
- Caching of complex query post-processing results
It’s all about choice

Performance-oriented
Cost-oriented
Any Questions?