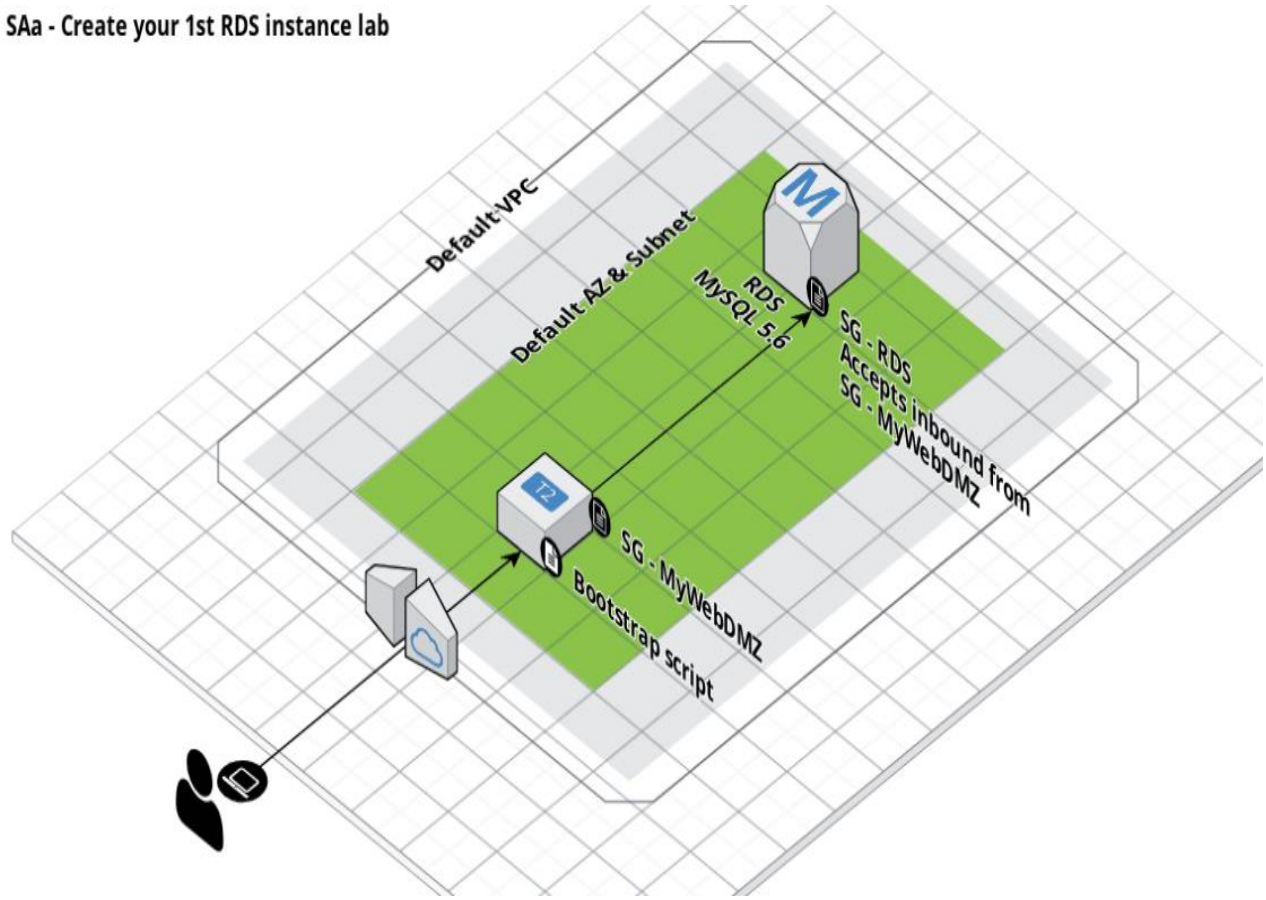
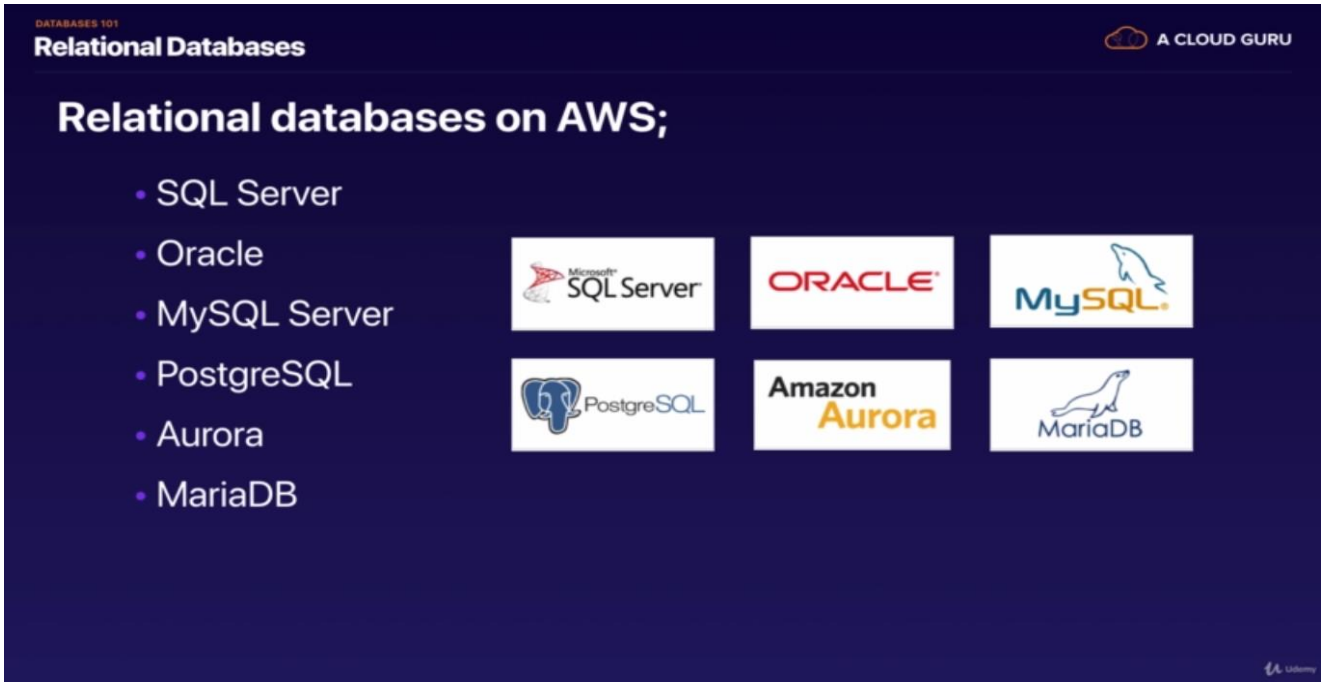
DATABASES ON AWS





Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. Amazon RDS provides you six familiar database engines to choose from, including Amazon Aurora, Oracle, Microsoft SQL Server, PostgreSQL, MySQL and MariaDB.

- Traditional relational databases that include tables, rows, fields
- On-Line Transaction Processing (OLTP) type DB.
- You can copy a snapshot to another region if you want to have your database available in another region
- You scale your DB by taking a snapshot and doing a restore to a larger sized tier
- RDS maximum size for a MS SQL Server DB with SQL Server Express Edition is 10GB per DB
- Supported RDS Platforms:
    - MS SQL Server
    - Oracle
    - MySQL Server
    - PostgreSQL
    - Aurora
    - MariaDB
- When a backup is restored, the restore will always be a new RDS instance, with a new DNS name
- Backup types:
    - Automated backups
        - Allows you to recover your database to any point in time within a retention period
        - Retention periods can be between 1 and 35 days
        - Takes a full daily snapshot and will also store transaction logs through the day
        - When you do a recovery, AWS will choose the most recent daily backup and then apply transaction logs
        - Allows you to do a point in time recover down to a second within the retention period
        - Enabled by default
        - Backup data is stored in S3
        - You get free storage space equal to the size of your database.
        - Taken within a defined window
        - During the backup, storage I/0 may be suspended and you may experience extended latency
    - Database snapshots
        - User initiated from the console
        - Stored even after you delete the original RDS instance unlike automatic backups
- Encryption:
- Encryption at rest is supported for MySQL, Oracle, SQL Server, PostgreSQL, and MariaDB
- Encryption is done using the AWS Key Management Service (KMS)

- Once your RDS instance is encrypted the data stored at rest in the underlaying storage is encrypted, as are its automated backups, read replicas and snapshots
- To use RDS encryption, create a new DB instance with encryption enabled and migrate your data to it
- Encrypting an existing DB instance is not supported
- Multi-AZ:
    - Allows you to have an exact copy of your production database in another AZ
    - AWS handles the replication for you, so when your prod database is written to, the write will automatically be synchronized to the stand-by DB
    - In the event of DB maintenance, instance failure or AZ failure, RDS will automatically fail-over to the standby so that database operations can resume quickly without Admin intervention.
    - In a fail-over scenario, the same DNS name is used to connect to the secondary instance, There is no need to reconfigure your application
    - Multi AZ configurations are used for HA/DR only, and is not used for improving performance
    - To scale for performance you need to set up read replicas
    - Available for SQL Server, Oracle, MySQL, PostGreSQL, and Aurora
- Read Replica's:
    - Uses asynchronous replication, from the primary instance to other instances that can be read from
    - You can have up to 5 read replicas of your main database
    - Allow you to have a read only copy of your prod database
    - Used primarily for very read-heavy database workloads
    - SQL Server and Oracle are not supported
    - Used for scaling not DR
    - Must have automatic backups setup
    - You can have read replicas of read replicas (but could incur latency as its daisy chained)
    - Each read replica will have its own DNS endpoint
    - You cannot have read replicas that have Multi-AZ
    - You can create read replicas of Multi-AZ source databases however
    - Read Replicas can be promoted to be their own databases, however this breaks replication
    - Read Replicas in a second region for MySQL and MariaDB, not for PostgreSQL
    - Read Replicas can be bigger than the primary source DB from a resource perspective
- Aurora:
    - MySQL compatible relational database engine that combines speed and availability of high end commercial databases with the simplicity and cost-effectiveness of open source databases
    - Provides up to 5 times better performance than MySQL at a price point 1/10th of a commercial database while delivering similar performance and availability
    - Starts with 10GB, scales in 10GB increments up to 64TB (Storage Auto scaling)
    - Compute resources can scale up to 32 vCPUs and 244 GB of memory
    - Maintains 2 copies of your data contained in each availability zone, with minimum of 3 AZs. 6 copies of your data

- Designed to transparently handle the loss of up to two copies of data without affecting the DB write availability and up to 3 copies without affecting read availability
- Designed to handle loss of up to 2 copies without affecting DB write availability
- Designed to handle loss of up to 3 copies without affecting DB read availability
- Self healing storage, data blocks and disks are continuously scanned for errors and repaired automatically
- 2 Types of replicas available:
  - Aurora Replicas - Separate aurora DB, can have up to 15 replicas
  - MySQL read replicas, can have up to 5
- If a failure occurs of the primary database, a fail-over will happen automatically to an aurora replica, but will NOT auto fail over to a MySQL read replica.
- Only available in certain regions, not all

Resource or Operation	Default Limit
Clusters:	40
Cluster parameter groups:	50
DB Instances:	40
Event subscriptions:	20
Manual snapshots:	50
Manual cluster snapshots:	50
Option groups:	20
Parameter groups:	50
Read replicas per master:	5
Aurora only read replicas per master:	15

Resource or Operation	Default Limit
Reserved instances (purchased per month):	40
Rules per security group:	20
Security groups:	25
Security groups (VPC):	5
Subnet groups:	20
Subnets per subnet group:	20
Tags per resource:	50
Total storage for all DB instances:	100 TB

DYNAMO DB (No SQL)

Fast and flexible NoSQL DB service for all apps that need consistent, single-digit millisecond latency at any scale. It is a fully managed database and supports both document and key-value data models. Its flexible data model and reliable performance make it a great fit for mobile, web, gaming, ad-tech, IoT, and many other applications.

- Non Relational DB (No-SQL), comprised of collections (tables), of documents (rows), with each document consisting of key/value pairs (fields)
- Document oriented DB
- Offers push button scaling, meaning that you can scale your db on the fly without any downtime
- RDS is not so easy, you usually have to use a bigger instance size or add read replicas
- Stored on SSD Storage
- Spread across 3 geographically distinct data centers
- Eventual Consistent Reads (Default)
    - Consistency across all copies of data is usually reached within 1 second
    - Repeating a read after a short time should return updated data
    - Best Read Performance
- Strongly Consistent Reads
    - Returns a result that reflects all writes that received a successful response prior to the read
- Structure:
    - Tables
    - Items (Think rows in a traditional table)
    - Attributes (Think columns of data in a table)
- Provisioned throughput capacity
- Write throughput 0.0065 per hour for every 10 units
- Read throughput 0.0065 per hour for every 50 units
- First 25 GB of storage is free
- Storage costs of 25 cents per additional GB per Month
- Can be expensive for writes, but really really cheap for reads
- The combined key/value size must not exceed 400 KB for any given document

US East (N. Virginia) Region	**Default Limit**
Maximum capacity units per table or global secondary index:	40,000 read capacity units and 40,000 write capacity units
Maximum capacity units per account:	80,000 read capacity units and 80,000 write capacity units

All Region Resource or Operation	Default Limit
Maximum capacity units per table or global secondary index:	10,000 read capacity units and 10,000 write capacity units
Maximum capacity units per account:	20,000 read capacity units and 20,000 write capacity units
Maximum number of tables:	256

Elasticache:

Amazon ElastiCache is a web service that makes it easy to deploy, operate, and scale an in-memory data store or cache in the cloud.

- Can be used for DB caching in conjunction with services like RDS
- Web service that makes it easy to deploy, operate, and scale in memory cache in the cloud
- Improves the performance of web applications by allowing you to retrieve information from fast, managed in-memory caches, instead of relying entirely on slower disk based databases
- Improves application performance by storing critical pieces of data in memory for low-latency access
- Cached information may include the results of I/O intensive database queries or the results of computationally intensive calculations
- Supports 2 open-source in-memory caching engines:
    - Memcached:
        - Widely adopted memory object caching system
        - Elasticache is protocol complaint with memcached, so popular tools that you use today with existing memcached environments will work seamlessly with the service
        - No Multi AZ support
    - Redis:
        - Popular open-source in-memory key-value store that supports data structures such as sorted sets and lists
        - Elasticache supports Master/Slave replication and Multi-AZ which can be used to achieve cross AZ redundancy
        - Good choice if your db is read heavy and not prone to frequent changing

All Region Resource or Operation	Default Limit	Description
Nodes per region:	50	The maximum number of nodes across all clusters in a region.
Nodes per cluster (Memcached):	20	The maximum number of nodes in an individual Memcached cluster.
Nodes per cluster (Redis):	1	The maximum number of nodes in an individual Redis cluster.
Clusters per replication group (Redis):	6	The maximum number of clusters in a Redis replication group. One is the read/write primary. All others are read only replicas.
Parameter groups per region:	20	The maximum number of parameters groups you can create in a region.
Security groups per region:	50	The maximum number of security groups you can create in a region.
Subnet groups per region:	50	The maximum number of subnet groups you can create in a region.
Subnets per subnet group:	20	The maximum number of subnets you can define for a subnet group.

Redshift:

Fast and powerful, fully managed, petabyte-scale data warehouse service in the cloud. Customers can start small for just 25 cents per hour with no commitments or upfront costs and scale to a petabyte or more for 1000 per TB per year. Less than a tenth of most other data warehousing solutions.

- Used for data warehousing / business intelligence
- Uses 1024KB/1MB block size for its columnar storage
- Tools like Cognos, Jaspersoft, SQL Server Reporting Services, Oracle Hyperion, SAP NetWeaver
- Used to pull in very large and complex data sets
- Used by management to do queries on data such as current current performance vs target

- 10 times faster than traditional RDS
- Massively Parallel Processing (MPP)
- Automatically distributes data and query load across all nodes
- Currently only available in 1 AZ at a time
- Can restore snapshots to new AZ's in the event of an outage
- 2 types of transactions:
    - On-line Transaction Processing (OLTP) - Standard transaction driven database insert/retrieval -Pulls up a row of data such as Name, Date etc..
    - On-line Analytics Processing (OLAP) - Pulls up a row of data such as Name, Date etc..
        - Uses different type of architecture both from a DB and infrastructure layer
        - Pull in data from multiple queries, gathering tons of information depending on what type of report is required
- Start with Single Node (160GB)
- Multi-node configurations available:
    - Leader Node - Manages client connections and receives queries
    - Compute Node - Store data and perform queries and computations
    - Can have up to 128 compute nodes
- Columnar data storage:
    - Instead of storing data as a series of rows, redshift organizes data by column.
    - Unlike row-based systems, which are ideal for transaction processing, Column-based systems are ideal for data warehousing and analytics where queries often involve aggregates performed over large data sets.
    - Only columns involved in the queries are processed and columnar data is stored sequentially on the storage media
    - Column-based systems require far fewer I/Os, greatly improving query performance
- Advanced compression:
    - Columnar data stores can be compressed much more than row-based data stores because similar data is stored sequentially on disk
    - Redshift employs multiple compression techniques and can often achieve significant compression relative to traditional relational data stores
    - Does not require indexes or materialized views so uses less space than traditional relational db systems
    - Automatically samples your data and selects the most appropriate compression scheme
- Priced on 3 things
    - Total number of hours you run across your compute nodes for the billing period
    - You are billed for 1 unit per node per hour, so 3-node cluster running an entire month would incur 2,160 instance hours
    - You will not be charged for leader node hours, only compute nodes will incur charges
    - Charged on backups
    - Charged for data transfers (only within VPC not outside)
- Security:
    - Encrypted in transit using SSL
    - Encrypted at rest using AES-256 encryption
    - Takes care of key management by default
    - Manage your own keys through Hardware Security Module (HSM)
    - AWS Key Management Service

Resource or Operation	Default Limit
Nodes per cluster:	101
Nodes per cluster:	200
Reserved Nodes:	200
Snapshots:	20
Parameter Groups:	20
Security Groups:	20
Subnet Groups:	20
Subnets per Subnet Group:	20
Event Subscriptions:	20