AWS – Certified Security Architect

Databases

|  |  |
| --- | --- |
| Relational | NonRelational |
| Database | Database |
| Tables | Collections |
| Row | Document |
| Field | Key Value Pairs |

RDS - Relational Databases (OLTP)

* MSSQL
* Oracle
* MySQL
* PostgreSQL
* Amazon Aurora
* MariaDB

NonRelational – No SQL

* DynamoDB

OLAP

* RedShift

Data Warehousing

This is used for Business Intelligence. Used to pull in large and complex volumes of data and execute queries on them.

OLTP vs OLAP: Online Transaction Processing vs Online Analytics processing

OLTP: Interact with one record at a time

OLAP: Pulls in a large number of records to query

OLAP systems are architected differently

Elasticache

Allows you to deploy, operate and scale an in memory cache in the cloud. Allows for fast retrieval of frequently accessed information and reduces load on database servers

AWS Hosts:

* Memcached
* Redis

LAB 1: Set up a RDS Database

Console 🡪 Databases 🡪 Relational Database Service 🡪 Instances 🡪 MySQL

Create a EC2 Instance with the following in the advanced section to bootstrap

#!/bin/bash

yum install httpd php php-mysql -y

yum update -y

chkconfig httpd on

service httpd start

echo "<?php phpinfo();?>" > /var/www/html/index.php

cd /var/www/html

wget <https://s3.eu-west-2.amazonaws.com/acloudguru-example/connect.php>

VPC 🡪 Security Groups 🡪 MYSQL Security group

Add 3306(mysql) from security group of the EC2 instance

connect.php:

<?php

$username = "acloudguru";

$password = "acloudguru";

$hostname = "yourhostnameaddress";

$dbname = "acloudguru";

//connection to the database

$dbhandle = mysql\_connect($hostname, $username, $password) or die("Unable to connect to MySQL");

echo "Connected to MySQL using username - $username, password - $password, host - $hostname<br>";

$selected = mysql\_select\_db("$dbname",$dbhandle) or die("Unable to connect to MySQL DB - check the database name and try again.");

?>

Get the **Endpoint** from the DB instance and insert into $hostname

http to ec2instanceIP/connect.php

BACKUPS

* Automated Backups:
  + Allow you to restore to any point in the “**Retention Period**”.
  + The retention period is from 1-35 days
  + Daily Backups + Transaction Logs
  + Enabled automatically
  + Stored in S3
  + Free storage equivalent to size of database
  + Fixed Backup time
  + Storage IO may be suspended during backup causing latency
* Database snapshots
  + Fixed Point in time image
  + Done manually
  + Persist after database deletion
* Restore
  + Restored instance becomes the new instance with DNS changed from original to restored. You have a new endpoint
* Encryption
  + Available for RDS
  + Done using KMS
  + Backups, snapshots and read replicas are encrypted as well
  + Cannot encrypt an existing Database
  + Snapshot, Copy, encrypt the copy
* Multi-AZ
  + Synchronously replicated to another AZ
  + For DR only – if AZ1 dies or planned maintenance or DB instance failure, the RDS instance will automatically failover to the other AZ
  + DNS address will point to the failover AZ
* Read Replica
  + All transactions are pushed to replicas of the main DB
  + Can have up to 5 Read Replicas
  + Read operations are distributed
  + Can have a Read Replica of a read replica (may have some latency)
  + Read replicas could be in same or different AZ or Region
  + Asynchronous replication
  + Not available for Oracle or SQL Server
  + Used for scaling out
  + Must have automatic backups turned on
  + Each read replica has its own DNS endpoint
  + Read Replicas can be multi AZ
  + Can create read replicas of multi AZ DBs
  + Can be promoted to become their own DBs (This breaks the replication

DynamoDB

Low latency No SQL database supporting documents and key value pairs