RDS & DynamoDB basics

Databases

In the world of databases there are two main categories:

(1) ***Relational Databases*** know as "***SQL***"
(2) ***Non-Relational Databases*** known as "***NoSQL***"

Amazon offers services for both types of databeses

RDS for SQL databses
and
DynamoDB for NoSQL databases

RDS = **R**elational **D**atabase **S**ervice

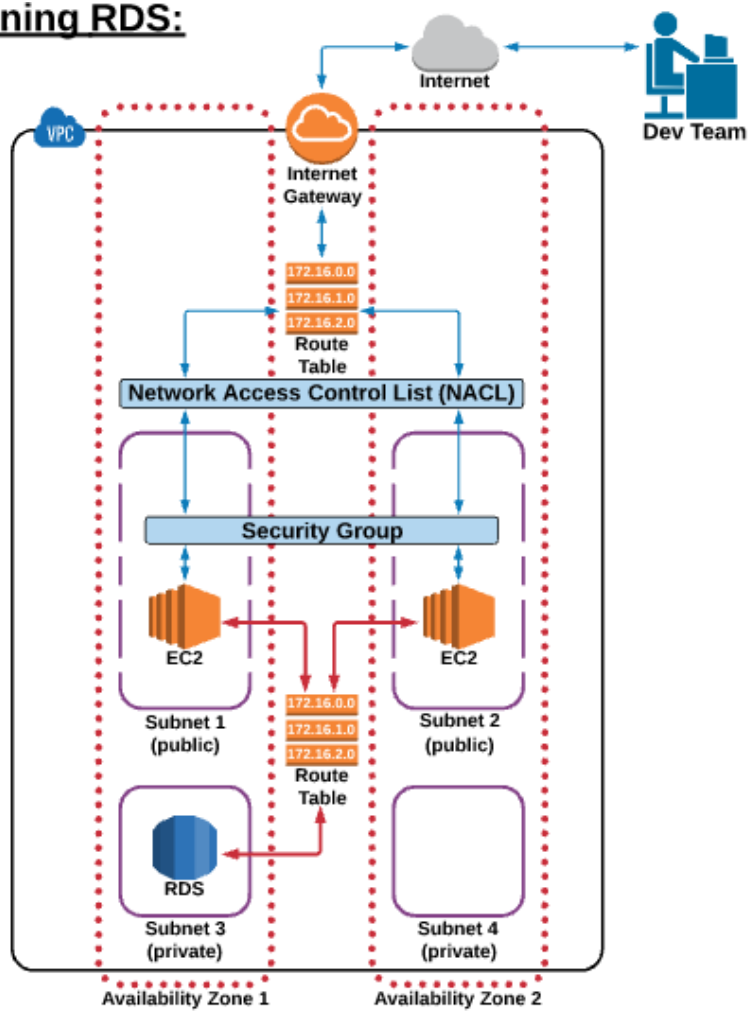
What is RDS?

"Essentials" Definition:
RDS is a ***SQL database service*** that provides a wide range of SQL database options to select from.

SQL Options Include:
(1) Amazon Aurora
(2) MySQL
(3) MariaDB
(4) PostgreSQL
(5) Oracle (sever Oracle options are available)
(6) Microsoft SQLServer (several Microsoft options are available)

AWS Definition:
"Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides ***cost-efficient***, ***resizeable capacity*** for an industry-standard relational database and manages common database administration tasks."

Provisioning RDS:



DynamoDB

What is DynamoDB?

"Essentials" Definition:
DynamoDB is a ***NoSQL database service***. Unlike RDS, DynamoDB does NOT provide other NoSQL software options.

DymanoDB can replace (or is very similar to):
(1) MongoDB
(2) Cassandra DB
(3) Oracle NoSQL

AWS Definition:
"Amazon DynamoDB is a fast and flexible ***NoSQL database service*** for all applications that need ***consistent, single-digit millisecond latency at any scale***. Its flexible data model and reliable performance make it a great fit for ***mobile, web, gaming, ad-tech, IoT***, and many other applications."

What is the Difference Between SQL and NoSQL?

RDS
(SQL)

DynamoDB
(NOSQL)

SQL Database Structure

NoSQL Database Structure

(1) Stores related data in tables (using columns and rows).

(2) Typically used for very stuctured data, such as contact lists.

(1) Stores related data in JSON-like, name-value documents.

(2) Typically used for non-structured data such as cataloging documents.

Database Basics:

***RDS* Pricing/Cost Overview**:

Free Tier use is available for all RDS options ***EXCEPT Aurora***.

How are you charged for using RDS?

(1) ***The RDS "engine" you choose***
- -Amazon Aurora
- -MySQL
- -MariaDB
- -PostgreSQL
- -Oracle (sever Oracle options are available)
- -Microsoft SQLServer (several Microsoft options are available)

(2) ***RDS Instance Classes***
- -This is very similar to EC2 Instance types

(3) **Purchasing Terms**:
- -On Demand
- -Reserved

(4) ***Database Storage***

(5) ***Data Transfer*** in/out of RDS

NOTE: Before doing any major usage of RDS, you should make sure to review AWS's current pricing model to make sure you understand how much you will be required to pay.

Detailed RDS pricing info:
https://aws.amazon.com/rds/pricing/

***DynamoDB* Pricing/Cost Overview**:

Free Tier use is available for DynamoDB.

How are you charged for using DynamoDB?

(1) ***Provisioned Throughput Capacity***

(2) ***Indexed Data Storage***

(3) ***DynamoDB Streams***

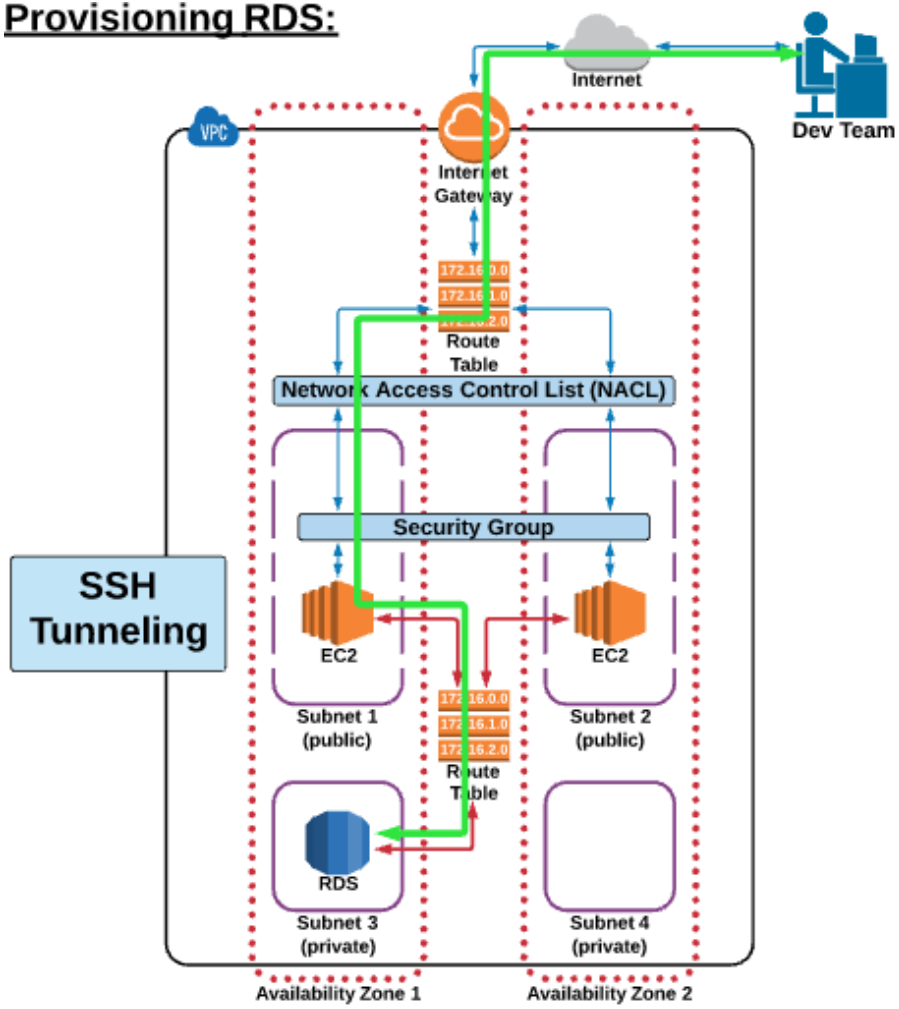
(4) ***Reserved Capacity***

(5) ***Data Transfer*** in/out of DynamoDB

NOTE: Before doing any major usage of DynamoDB, you should make sure to review AWS's current pricing model to make sure you understand how much you will be required to pay.

Detailed DynamoDB pricing info:
https://aws.amazon.com/dynamodb/pricing/

Provisioning RDS:



Configuring a Private Subnet Group:

Within RDS, we need to create a subnet group that contains our two private subnets. This is how we provision the RDS database into a private subnet.

Basic Steps:

(1) Naviage to "Subnet Groups"
(2) Create DB subnet group
(3) Complete the form, making sure to select our two private subnets.

Note: Two subnets are required to form a subnet group.

Launching an RDS Database:

Basic Steps:

(1) Select an engine:
-For Free Tier usage, select MySQL -> Dev/Test Option.
(2) Specify DB details:
a) Instance specifications
b) Settings
(3) Configure advanced settings:
a) Network & Security
-Select the private subnet group; do not use "default"
-"Publicly Accessible" should be set to ***NO***.
b) Database options
c) Backup
d) Monitoring
e) Maintainance
(4) Launch DB instance!

Connecting to a MySQL RDS Database:

To connect to an RDS database, you need to use an appropriate third party application.
You cannot access/use an RDS database from the AWS Console.

Basic Steps: For MySQL RDS Database

(1) Download and install MySQL Workbench (download here)
(2) Open MySQL Workbench
(3) Set up a new connection in MySQL Workbench:
a) Give the connection a name
b) For connection method, select "***Standard TCP/IP Over SSH***"
c) ***SSH Hostname*** = public IP address of the EC2 instance we are going to tunnel through.
d) ***SSH username*** = ec2-user (this is the default username used when SSHing into EC2).
e) ***SSH Key File*** = the .pem key we used when SSHing into EC2
b) Copy and paste the "***Writer Endpoint***" from the RDS Console and paste it into ***MySQL Hostnem*** field.
c) Make sure ***Port*** is set to 3306.
d) Enter the ***Username and Password*** (store in keychain) that you used when creating the database.
e) ***Test the connection***, and if successful, click "ok" to connect.

You should now be connected to the database!

https://dev.mysql.com/downloads/workbench/