

**RAHUL GHOSH**

**Homework 9: Database Services (Total Points: 100)**

Due: Sunday November 8 11:59PM ET

**Problem 1. Relational Database (50 points)**

1. Create new MariaDB database instance (Make sure to check the Free Tier option):
  - a. Use “<your\_initials>mariadb”, e.g. “gmmariadb”, as DB Instance Identifier (that’s the name of your database space to hold all your MariaDB databases), “admin” as Master username, and “admin123” as Master password. You can use your own DB name, master username and password, just don’t forget it.
  - b. **Configure Network & Security to use default VPC, default subnet, make it not publicly accessible, create new VPC security group.**
  - c. Set database name to “e90\_grade\_tracker” , leave other options with default values, uncheck “Deletion protection”.

Paste the screenshot of the Summary of your DB instance:

The screenshot displays the AWS Management Console interface for a MariaDB database instance named **rgmariadb**. The interface includes a top navigation bar with 'Modify' and 'Actions' buttons. Below the instance name, there is a 'Summary' section with a table of key metrics:

| DB identifier | CPU              | Info      | Class       |
|---------------|------------------|-----------|-------------|
| rgmariadb     | 1.86%            | Available | db.t2.micro |
| Role          | Current activity | Engine    | Region & AZ |
| Instance      | 1 Connections    | MariaDB   | us-east-2a  |

Below the summary, there are tabs for 'Connectivity & security', 'Monitoring', 'Logs & events', 'Configuration', 'Maintenance & backups', and 'Tags'. The 'Connectivity & security' tab is selected, showing three main sections:

- Endpoint & port:** Endpoint is [rgmariadb.cwmfaoahejcm.us-east-2.rds.amazonaws.com](https://rgmariadb.cwmfaoahejcm.us-east-2.rds.amazonaws.com) and Port is 3306.
- Networking:** Availability zone is [us-east-2a](#), VPC is [vpc-13e71178](#), Subnet group is [default-vpc-13e71178](#), and Subnets are [subnet-41273929](#), [subnet-98fd42d4](#), and [subnet-46baee3c](#).
- Security:** VPC security groups include [maridb-securitygroup \(sg-0437eb0e05197eb7e\)](#) (active). Public accessibility is set to 'No'. Certificate authority is 'rds-ca-2019' and the certificate authority date is 'Aug 22nd, 2024'.

2. Launch an EC2 Instance to be used as your MySQL DB client.
  - a. Create a t2.micro EC2 instance with the mysql client installed on it, using the following script (use a security group with open ssh and http ports):

```
#!/bin/bash
yum install httpd php php-mysql mysql -y
yum update -y
chkconfig httpd on
service httpd start
echo "<?php phpinfo();?>" > /var/www/html/index.php
```

- b. Paste the screenshot of the Description tab of your instance:

The screenshot displays the AWS Management Console interface for an EC2 instance. At the top, the breadcrumb navigation shows 'EC2 > Instances > i-06108ea2e0df99884'. Below this, the 'Instance summary for i-06108ea2e0df99884 (MARIADB-2-DEMO)' is shown, with an 'Info' link and buttons for 'Connect' and 'Instance state'. The summary is organized into three columns:

| Instance ID                          | Public IPv4 address  | Private IPv4 addresses                      |
|--------------------------------------|--|---|
| i-06108ea2e0df99884 (MARIADB-2-DEMO) | 18.219.67.152   <a href="#">open address</a>                                     | 172.31.38.211                               |
| Instance state                       | Public IPv4 DNS  | Private IPv4 DNS                            |
| Running                              | ec2-18-219-67-152.us-east-2.compute.amazonaws.com   <a href="#">open address</a> | ip-172-31-38-211.us-east-2.compute.internal |
| Instance type                        | Elastic IP addresses   | VPC ID                                      |
| t2.micro                             | -  | vpc-13e71178                                |
| IAM Role                             | Subnet ID  |   |
| -                                    | subnet-98fd42d4  |   |

Paste the screenshot of the mysql section from the PHP web page displayed by your EC2 instance if you access it through browser (scroll down to find mysql section or search for it in the page)

chromeFileEditViewHistoryBookmarksPeopleTabWindowHelp

Inbox - r XInbox - r XPure Soc XPure Sto XFound: T XAlert 12 XEvacuati XSearch - XHW09 - XInstance XRDS - All XTutorial Xphpinfo[] X

Not Secure18.219.67.152

OktaMailCalendarDashboardWorkdaySalesforcePaylocityNHTJiraPure1SkylineSource of upgrad...Supportmysql8/60Fidelity

Close find bar

mysql

|                         |  |  |
|-------------------------|--|--|
| MySQL Support           | enabled  |  |
| Active Persistent Links | 0  |  |
| Active Links            | 0  |  |
| Client API version      | mysqlnd 5.0.10 - 20111026 - \$Id: e707c415db32080b3752b232487a435ee0372157\$ |  |

| Directive                | Local Value               | Master Value              |
|--------------------------|---------------------------|---------------------------|
| mysql.allow_local_infile | On                        | On                        |
| mysql.allow_persistent   | On                        | On                        |
| mysql.connect_timeout    | 60                        | 60                        |
| mysql.default_host       | no value                  | no value                  |
| mysql.default_password   | no value                  | no value                  |
| mysql.default_port       | no value                  | no value                  |
| mysql.default_socket     | /var/lib/mysql/mysql.sock | /var/lib/mysql/mysql.sock |
| mysql.default_user       | no value                  | no value                  |
| mysql.max_links          | Unlimited                 | Unlimited                 |
| mysql.max_persistent     | Unlimited                 | Unlimited                 |
| mysql.trace_mode         | Off                       | Off                       |

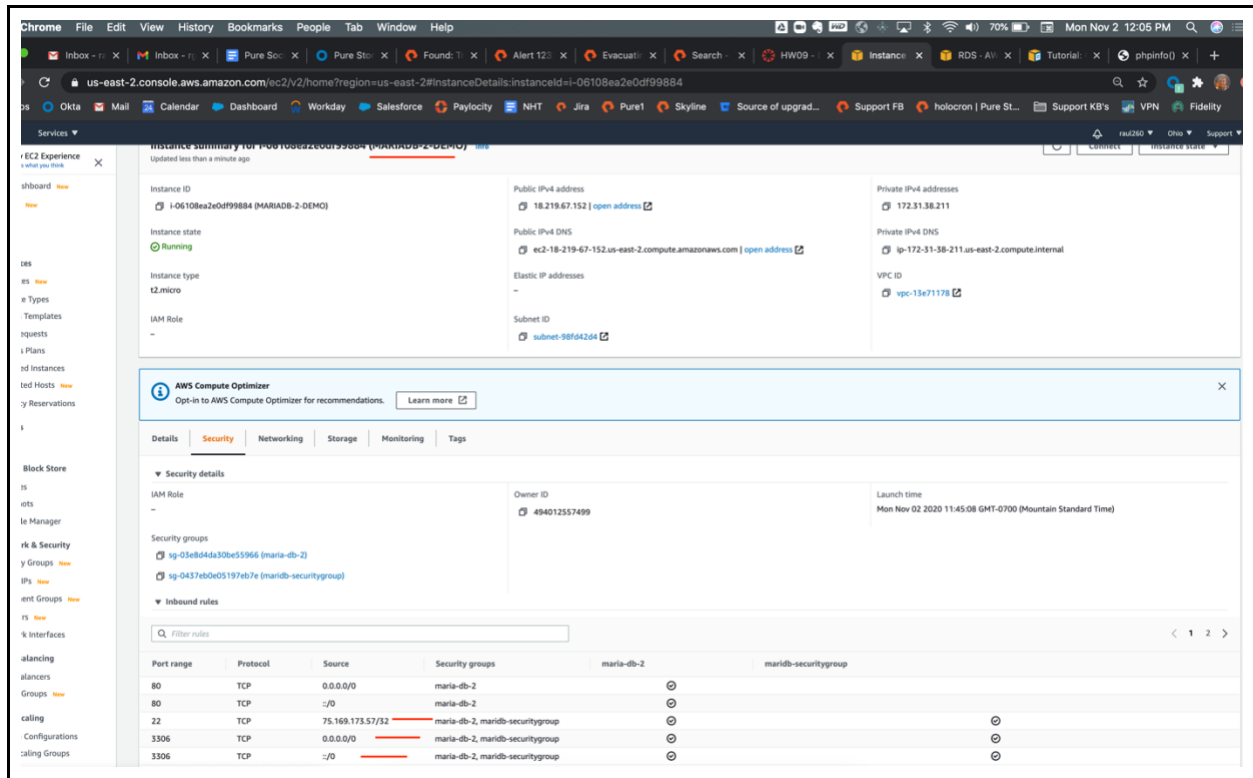
mysqli

|                            |  |  |
|----------------------------|--|--|
| Mysqli Support             | enabled  |  |
| Client API library version | mysqlnd 5.0.10 - 20111026 - \$Id: e707c415db32080b3752b232487a435ee0372157\$ |  |
| Active Persistent Links    | 0  |  |
| Inactive Persistent Links  | 0  |  |
| Active Links               | 0  |  |

| Directive                 | Local Value               | Master Value              |
|---------------------------|---------------------------|---------------------------|
| mysqli.allow_local_infile | On                        | On                        |
| mysqli.allow_persistent   | On                        | On                        |
| mysqli.default_host       | no value                  | no value                  |
| mysqli.default_port       | 3306                      | 3306                      |
| mysqli.default_pw         | no value                  | no value                  |
| mysqli.default_socket     | /var/lib/mysql/mysql.sock | /var/lib/mysql/mysql.sock |
| mysqli.default_user       | no value                  | no value                  |

3. Ensure that your RDS instance and/or EC2 instance is using proper security group settings for access.

Screenshot of the security group configuration used to allow access from your EC2 instance to RDS:



4. Connect to your DB from your EC2 instance:  
`mysql -h <database endpoint> -P <database port number> -u <your user name> -p`

Screenshot of the results from running a `show databases` query:

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 134
Server version: 10.4.8-MariaDB-log Source distribution

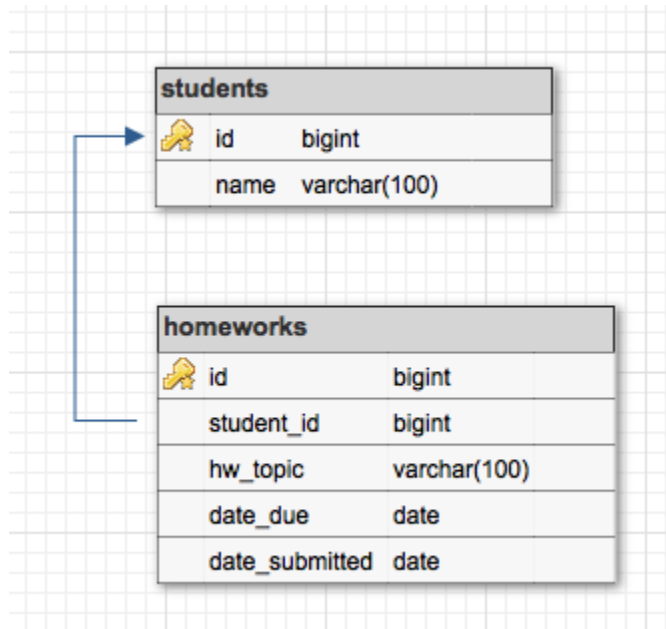
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use e90_grade_tracker
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [e90_grade_tracker]> show database
->
```

Now run a sequence of SQL statements to create the tables according to the following schema diagram:



- a. Run the following SQL to create your DB and tables:

```
CREATE DATABASE e90_grade_tracker;

USE e90_grade_tracker;

CREATE TABLE `students` (
  `id` bigint NOT NULL AUTO_INCREMENT,
  `name` varchar(100) NOT NULL,
  PRIMARY KEY (`id`));

CREATE TABLE `homeworks` (
  `id` bigint NOT NULL AUTO_INCREMENT,
  `student_id` bigint NOT NULL,
  `hw_topic` varchar(100) NOT NULL,
  `date_due` DATE NOT NULL,
  `date_submitted` DATE,
  PRIMARY KEY (`id`));
```

- b. Add students and homeworks to the tables:

|   |
|---|
| INSERT INTO students (name) VALUES ('Johnny Depp'); |
|---|

|  |
|--|
| INSERT INTO students (name) VALUES ('Jennifer Aniston'); |
|--|

|   |
|---|
| INSERT INTO students (name) VALUES ('Natalie Portman'); |
|---|

|  |
|--|
| INSERT INTO homeworks (student_id, hw_topic, date_due, date_submitted)<br>VALUES (1, 'HW1', '2018-11-10', '2018-11-09'); |
|--|

|  |
|--|
| INSERT INTO homeworks (student_id, hw_topic, date_due, date_submitted)<br>VALUES (1, 'HW2', '2018-11-10', '2018-11-09'); |
|--|

|  |
|--|
| INSERT INTO homeworks (student_id, hw_topic, date_due, date_submitted)<br>VALUES (2, 'HW1', '2018-11-10', '2018-11-10'); |
|--|

|  |
|--|
| INSERT INTO homeworks (student_id, hw_topic, date_due, date_submitted)<br>VALUES (2, 'HW2', '2018-11-10', '2018-11-14'); |
|--|

|  |
|--|
| INSERT INTO homeworks (student_id, hw_topic, date_due, date_submitted)<br>VALUES (3, 'HW1', '2018-11-10', '2018-11-12'); |
|--|

|  |
|--|
| INSERT INTO homeworks (student_id, hw_topic, date_due, date_submitted)<br>VALUES (3, 'HW2', '2018-11-10', '2018-11-11'); |
|--|

- c. Write a SQL query to list all students with late homeworks. Show the student's name, homework topic, date\_submitted, and number of days the HW is late. Sort your results by the number of days the HW is late.

A good beginners reference can be found here:

<https://www.w3schools.com/sql/default.asp>

**Paste your SQL query below:**

**SELECT s.name AS "Students Name",h.hw\_topic AS "Topic", h.date\_submitted AS "Submission Date",**

**-> DATEDIFF('2018-11-14','2018-11-10') as "No Of Days Late"**

**-> FROM students s**

**-> JOIN homeworks h**

**-> on s.id = h.student\_id**

**-> WHERE h.date\_submitted > h.date\_due**

**-> ORDER by s.name ASC;**

**Paste the screenshot of the result of the above query:**

```
MariaDB [e90_grade_tracker]>
MariaDB [e90_grade_tracker]> SELECT s.name AS "Students Name",h.hw_topic AS "Topic", h.date_submitted AS "Submission Date",
-> DATEDIFF('2018-11-14','2018-11-10') as "No Of Days Late"
-> FROM students s
-> JOIN homeworks h
-> on s.id = h.student_id
-> WHERE h.date_submitted > h.date_due
-> ORDER by s.name ASC;
```

| Students Name    | Topic | Submission Date | No Of Days Late |
|------------------|-------|-----------------|-----------------|
| Jennifer Aniston | HW2   | 2018-11-14      | 4               |
| Natalie Portman  | HW1   | 2018-11-12      | 4               |
| Natalie Portman  | HW2   | 2018-11-11      | 4               |

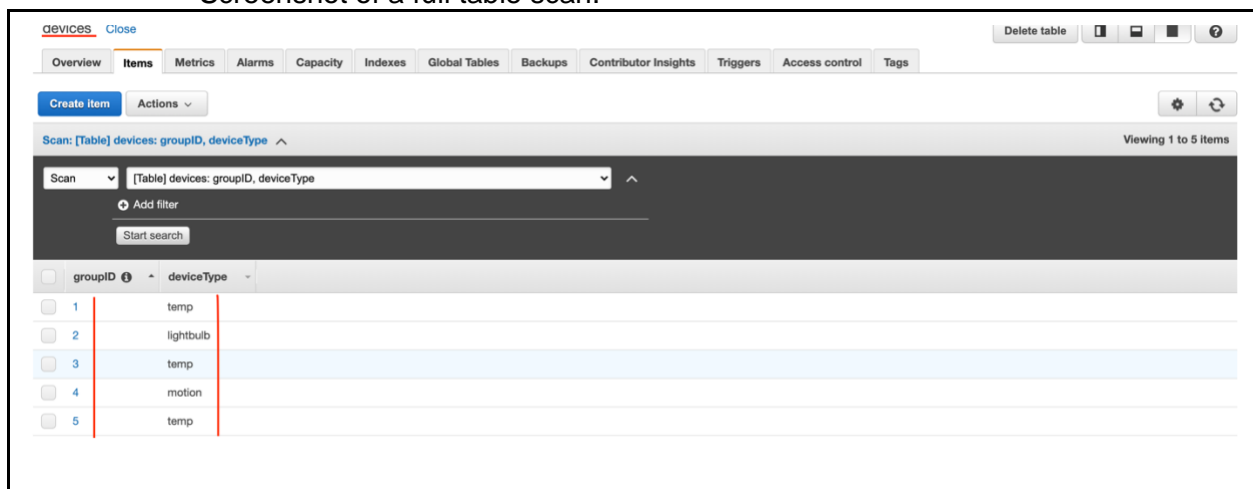
```
3 rows in set (0.00 sec)

MariaDB [e90_grade_tracker]>
```

**Problem 2: Set up a NoSQL DynamoDB Database to store device info for an IOT project (50 points).**

1. Create a table with the name “devices” with a partition key of type number named “groupID” and with a sort key of type String named “deviceType”.
2. Use default settings for all other options, except set both your read and write capacity to ‘1’.
3. Initially we want to list 5 devices with just their groupID and deviceType, so we need to add 5 items:
  - groupID=1, deviceType=temp
  - groupID=2, deviceType=lightbulb
  - groupID=3, deviceType=temp
  - groupID=4, deviceType=motion
  - groupID=5, deviceType=temp

Screenshot of a full table scan:



4. Try adding an item with the same id but different type:
  - groupID=5, deviceType=motion

Describe what happened and explain why:

**It lets me add this record as this record is not unique on both group and device type**

5. Try adding an item with the same id and type:
  - groupID=1, deviceType=temp

Describe what happened and explain why:

**It failed to add this record as we have already added this record earlier which makes it unique in Dynamodb**



6. Add an additional attribute “tempValue” of type number to all devices of type “temp” with some various values.

Screenshot of a full table scan:

devices [Close](#) Delete table 📄 📱 🖨️ ?

[Overview](#) [Items](#) [Metrics](#) [Alarms](#) [Capacity](#) [Indexes](#) [Global Tables](#) [Backups](#) [Contributor Insights](#) [Triggers](#) [Access control](#) [Tags](#)

[Create item](#) [Actions](#) ⚙️ 🔄

Scan: [Table] devices: groupId, deviceType [^](#) Viewing 1 to 6 items

Scan [\[Table\] devices: groupId, deviceType](#) [^](#)

[+ Add filter](#)

[Start search](#)

|                          | groupId <a href="#">?</a> | deviceType | tempValue |
|--------------------------|---------------------------|------------|-----------|
| <input type="checkbox"/> | 1                         | temp       | 999       |
| <input type="checkbox"/> | 2                         | lightbulb  |           |
| <input type="checkbox"/> | 3                         | temp       |           |
| <input type="checkbox"/> | 4                         | motion     |           |
| <input type="checkbox"/> | 5                         | motion     |           |
| <input type="checkbox"/> | 5                         | temp       | 777       |

- For each of your temp devices, add a new attribute of type string called “manufacturer”. For the first and second device, set the value to “EMM” and for the last device set the value to “TAM”.

Screenshot of your full table scan:

devices [Close](#)

OverviewItemsMetricsAlarmsCapacityIndexesGlobal TablesBackupsContributor InsightsTriggersAccess controlTags

Create itemActions

Scan: [Table] devices: groupID, deviceType Viewing 1 to 6 items

Scan [Table] devices: groupID, deviceType

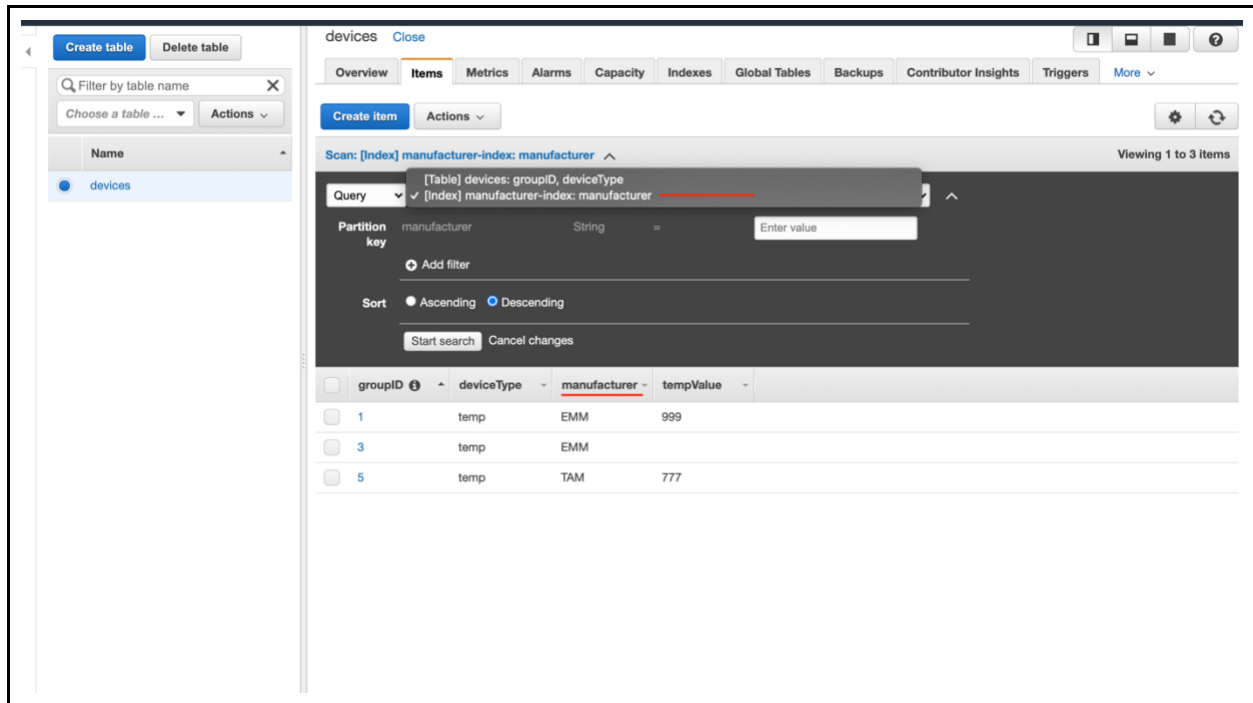
Add filter

Start search

|                          | groupID | deviceType | tempValue | manufacturer |
|--------------------------|---------|------------|-----------|--------------|
| <input type="checkbox"/> | 1       | temp       | 999       | EMM          |
| <input type="checkbox"/> | 2       | lightbulb  |           |              |
| <input type="checkbox"/> | 3       | temp       |           | EMM          |
| <input type="checkbox"/> | 4       | motion     |           |              |
| <input type="checkbox"/> | 5       | motion     |           |              |
| <input type="checkbox"/> | 5       | temp       | 777       | TAM          |

8. Create a secondary index on the manufacturer attribute to optimize your table's searches by manufacturer.

Screenshot showing the results of a full scan of all records stored in your secondary index:



### Bonus (5 pts.)

In lecture we discussed at a high level some of the benefits (e.g. less hands-on administration) vs. some of the drawbacks (e.g. increased costs) of using AWS RDS. We would like you to do a bit more research on your own. Pick one of the RDS supported databases and read through the AWS documentation. List 3 specific limitations of running said database in RDS vs. running on an EC2 instance or your own on-prem environment. Here is one such example for Oracle running on RDS:

- **RAC (real application clusters) are not currently supported on RDS.**
1. **Storage Performance:** Database performance is also a consideration in the cloud. In the public cloud, performance of your storage layer is often tied to how much capacity you provision. Your options are to either over-provision your capacity in order to get the performance you require out of your managed database service, or you can switch to one of the more-expensive disk options. If your workloads have large performance spikes, this means that the storage you require for your DBaaS might increase your costs significantly.
  2. **Data Protection:** Your database may be in the cloud, but it still needs to be backed up and protected. While a DBaaS usually comes with daily snapshots and backups out of the box, these capabilities are often insufficient in enterprise-grade scenarios. There are managed storage options that can help here, such as how Cloud Volumes ONTAP

offers [with point-in-time snapshot copies of your database](#), plus built-in high availability that makes sure you never miss strict RPO=0, RTO<60-second objectives.

3. **Lift and Shift:** If you are moving your workloads to the cloud, one of the first challenges you'll run into is finding out how to run your workloads in the cloud. The DBaaS idea sounds great, but do you know how to use them? Are you using a specialized database? Does your database "fit" in a DBaaS model? What don't you know, and how prepared is your staff to handle situations that may arise?  
By sticking with a traditional database deployment, many of those concerns go away, and you can just continue processing in the cloud.  
A DBaaS offers a pre-built model for your database. It isn't customizable and may require you to change the way your database or the application that depends on that database works. With managed storage for cloud-based components, there's more flexibility and adaptability which can minimize the learning curve and "gotchas" when moving your workloads to the cloud.

**Source:** <https://cloud.netapp.com/blog/cloud-based-database-challenges-and-advantages>