

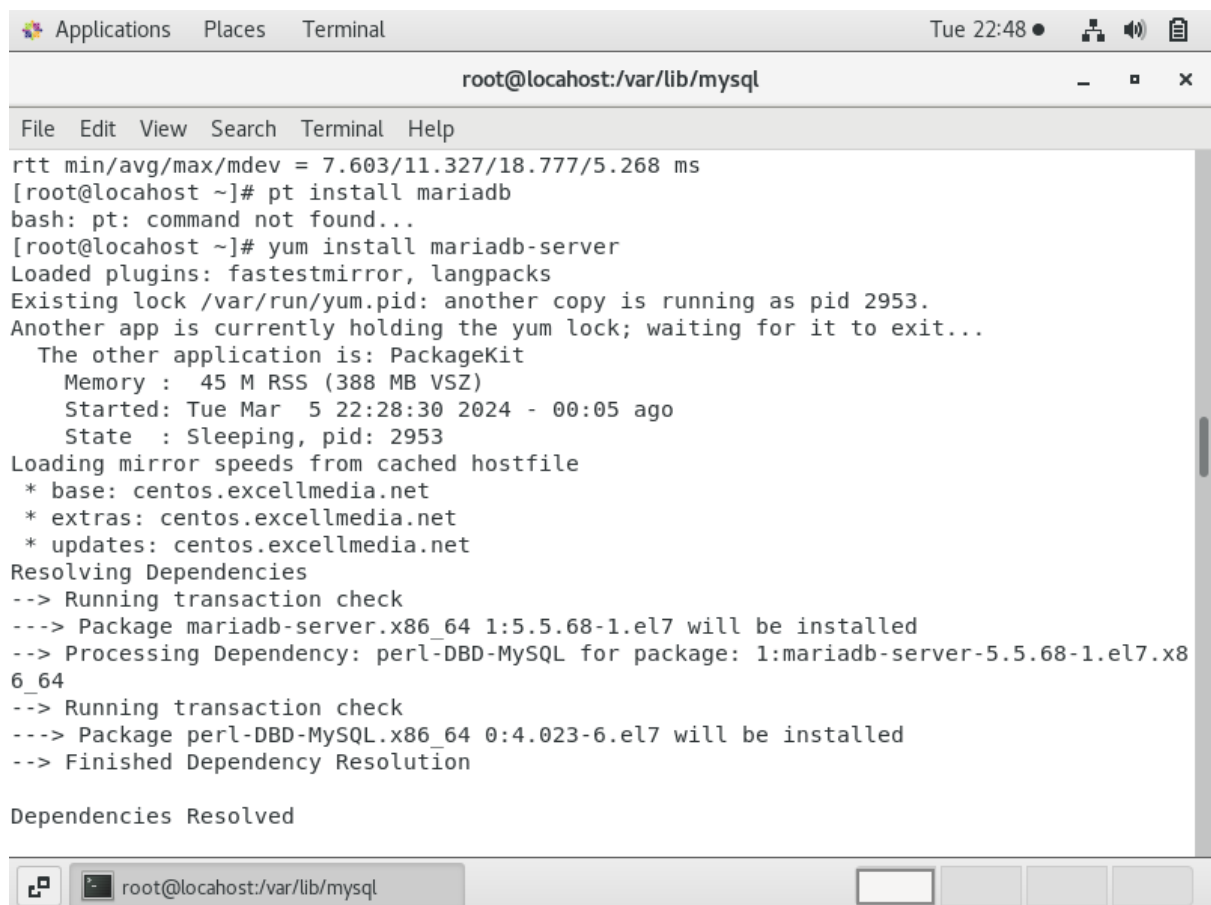
Wednesday, 6 March 2024

On-premises to cloud migration

First step is to create database in local machine. For this we are using Virtual Box as our on-premises server.

Go to your VirtualBox server and log in to it, Download mariadb on it.

#yum install mariadb



The screenshot shows a terminal window titled 'root@localhost:/var/lib/mysql'. The terminal output displays the execution of 'yum install mariadb-server'. It shows the resolution of dependencies, including 'perl-DBD-MySQL', and the final state of the installation process. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The status bar at the bottom shows the current directory and some window controls.

```
rtt min/avg/max/mdev = 7.603/11.327/18.777/5.268 ms
[root@localhost ~]# pt install mariadb
bash: pt: command not found...
[root@localhost ~]# yum install mariadb-server
Loaded plugins: fastestmirror, langpacks
Existing lock /var/run/yum.pid: another copy is running as pid 2953.
Another app is currently holding the yum lock; waiting for it to exit...
  The other application is: PackageKit
    Memory : 45 M RSS (388 MB VSZ)
    Started: Tue Mar 5 22:28:30 2024 - 00:05 ago
    State   : Sleeping, pid: 2953
Loading mirror speeds from cached hostfile
 * base: centos.excellmedia.net
 * extras: centos.excellmedia.net
 * updates: centos.excellmedia.net
Resolving Dependencies
--> Running transaction check
--> Package mariadb-server.x86_64 1:5.5.68-1.el7 will be installed
--> Processing Dependency: perl-DBD-MySQL for package: 1:mariadb-server-5.5.68-1.el7.x86_64
--> Running transaction check
--> Package perl-DBD-MySQL.x86_64 0:4.023-6.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

Let's create database on it. The same database we will migrate to our on premises server. Login to MariaDB server as root user using following command.

#mysql -h localhost -u root

After that run following command to create database "test" in Virtual Box using mariadb.

#show databases;

#CREATE DATABASES test;

```
[root@localhost ~]# cd /var/lib/mysql/
[root@localhost mysql]# ls
[root@localhost mysql]# mysql -h localhost -u root
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/lib/mysql/mysql.sock' (2)
[root@localhost mysql]# syetemctl start mariadb
bash: syetemctl: command not found...
[root@localhost mysql]# systemctl start mariadb
[root@localhost mysql]# mysql -h localhost -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 5.5.68-MariaDB MariaDB Server

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)]> show databases;
```

use TEST;

#CREATE TABLE cars (car name VARCHAR (20) , company VARCHAR (20)) ;

```
MariaDB [(none)]> use test
Database changed
MariaDB [test]> CREATE TABLE cars (car_name VARCHAR(20),company VARCHAR(20));
Query OK, 0 rows affected (0.01 sec)
```

#SHOW TABLES;

#INSERT INTO cars (car name, company) VALUES ("nexon", "tata"), ("Eco Sport",
"Ford"), ("creata", "hyundai");

#SELECT * FROM cars

```

MariaDB [test]> SHOW TABLES;
+-----+
| Tables_in_test |
+-----+
| cars            |
+-----+
1 row in set (0.00 sec)

MariaDB [test]> INSERT INTO cars(car_name,company) VALUES("nexon","tata"),("Eco Sport",
"Ford"),("creata","hyundai");
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [test]> SELECT * FROM cars;
+-----+-----+
| car_name | company |
+-----+-----+
| nexon    | tata    |
| Eco Sport | Ford    |
| creata   | hyundai |
+-----+-----+
3 rows in set (0.00 sec)

```

Create backup file of our on-premises database using below command.

```
#mysqldump -h localhost -u root test > mydump.sql
```

```

[root@localhost mysql]# mysqldump -h localhost -u root test > mydump.sql
[root@localhost mysql]# ls
aria_log.000000001  ibdata1      ib_logfile1  mysql        performance_schema
aria_log_control    ib_logfile0  mydump.sql   mysql.sock   test

```

Create RDS database in AWS.

The screenshot displays the AWS Management Console for an Amazon RDS database instance named 'database-1' in the 'ap-northeast-2' region. The console shows the instance is 'Available' and running on a 'db.t3.micro' class. The engine is 'MariaDB'. The console also shows the instance's connectivity and security details, including its endpoint, port, and VPC configuration.

Summary

DB identifier	Status	Role	Engine	Recommendations
database-1	Available	Instance	MariaDB	
CPU	Class	Current activity	Region & AZ	
1.73%	db.t3.micro	1 Connections	ap-northeast-2c	

Connectivity & security

Endpoint & port	Networking	Security
Endpoint: database-1.c28o2uawk3qs.ap-northeast-2.rds.amazonaws.com	Availability Zone: ap-northeast-2c	VPC security groups: rds-ec2-1 (sg-0f79d5cdece84ffb9)
Port: 3306	VPC: vpc-0b5a3e49afceb543a	Active: default (sg-073dfe977db9b24c2)
	Subnet group: rds-ec2-db-subnet-group-1	Publicly accessible: No

Now that our backup is created we need to migrate this backup file to on premises server and for that we need to use “scp” command that is ‘Secure Copy’.

For secure copy the mydump.sql backup file to migrate we need key pair associated with our ec2-instance. We can not copy anything outside of Virtual Box in our virtual box instance so we are creating key pair from our VirtualBox browser.

First log in to your AWS account, go to EC2 instance, Key pair, create key. This key will be Downloaded on your Virtual box, use this same key to launch your EC2 instance.

Check key.

```
#cd Downloads
```

```
#ls
```

Copy backup file using following command.

```
#sp -i abcd.pem /var/lib/mysql/mydump.sql ec2-user@13.125.224.6:/home/ec2-user/
```

```
[root@localhost ~]# cd Downloads
[root@localhost Downloads]# ls
abcd.pem
[root@localhost Downloads]# scp -i abcd.pem /var/lib/mysql/mydump.sql ec2-user@13.125.224.6:/home/ec2-user/
```

Connect to our EC2 instance and download mariadb-server.

```
#yum install mariadb-server
```

```
#systemctl start mariadb-server
```

```
#systemctl enable mariadb-server
```

Connect to RDS instance using below command followed by RDS password.

```
#mysql -h database-1.cz802uawk3qs.ap-northeast-2.rds.amazonaws.com -u
admin -p
```

```
[root@ip-172-31-37-139 ~]# mysql -h database-1.cz8o2uawk3qs.ap-northeast-2.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 105
Server version: 10.11.6-MariaDB-log managed by https://aws.amazon.com/rds/

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)]> CREATE DATABASE migrate;
Query OK, 1 row affected (0.003 sec)
```

#CREATE DATABASE migrate;

```
MariaDB [(none)]> CREATE DATABASE migrate;
Query OK, 1 row affected (0.003 sec)

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| innodb |
| migrate |
| mysql |
| performance_schema |
| sys |
+-----+
6 rows in set (0.002 sec)

MariaDB [(none)]> EXIT
Bye
```

Exit from it and again create table and populate it using backup file mydump.sql.

#mysql -h <RDS-end-point> -u admin -p migrate < mydump.sql

```
[root@ip-172-31-37-139 ~]# cd /home/ec2-user
[root@ip-172-31-37-139 ec2-user]# mysql -h database-1.cz8o2uawk3qs.ap-northeast-2.rds.amazonaws.com -u admin -p migrate < mydump.sql
Enter password:
[root@ip-172-31-37-139 ec2-user]# mysql -h database-1.cz8o2uawk3qs.ap-northeast-2.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 112
Server version: 10.11.6-MariaDB-log managed by https://aws.amazon.com/rds/

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

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

#SHOW DATABASES;

```
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| innodb |
| migrate |
| mysql |
| performance_schema |
| sys |
+-----+
6 rows in set (0.001 sec)
```

#USE migrate

#show tables;

```
MariaDB [(none)]> USE migrate;
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 [migrate]> show tables;
+-----+
| Tables_in_migrate |
+-----+
| cars |
+-----+
1 row in set (0.001 sec)
```

```
#select * from cars;
```

```
MariaDB [migrate]> select * from cars;  
+-----+-----+  
| car_name | company |  
+-----+-----+  
| nexon    | tata    |  
| Eco Sport | Ford    |  
| creata   | hyundai |  
+-----+-----+  
3 rows in set (0.001 sec)
```

If you can see this, that means data migration is successful.