

MAM

How do I test MAM with a Postgres 9.6 database without disturbing my production database?

This document will show you how to test a Postgres database upgrade for Moab Accounting Manager (MAM). It will explain how to install Postgres on a separate host from MAM so that you can leave your existing MAM database intact just in case something goes wrong. The instructions were performed using Postgres 9.6 running on Centos 6 (which should be compatible with Red Hat 6).

First you will need to install the RPMs and configure Postgres to allow connections from external hosts. This is explained in the first two sections of the following article

<http://kb.adaptivecomputing.com/phpmyfaq/index.php?action=artikel&cat=5&id=228&artlang=en>

That article will explain how to download and install the RPMs and how to configure Postgres to allow connections from external hosts. Once those steps are completed you can enable and start the Postgres service

```
#####  
[root]# chkconfig postgresql-9.6 on  
[root]# service postgresql-9.6 start  
#####
```

For the MAM database you will need to make sure Postgres to avoid interpreting backslashes as escape characters as described in <http://docs.adaptivecomputing.com/9-1-3/installGuide/installRH6/installRH6.htm#topics/hpcSuiteInstall/rpm/installing/installingMAMrpm.htm>

To do this examine the postgresql.conf file and make sure "standard_conforming_strings" setting is set to "on" or commented out since the default is "on". Comments begin with a "#" character.

```
#####  
[root]# cat /var/lib/pgsql/9.6/data/postgresql.conf | grep  
standard_conforming_strings  
#standard_conforming_strings = on  
#####
```

SSH to the host where MAM is running and determine what user/password are used to connect to the Postgres database.

```
#####  
[root]# ssh root@mam-host  
[root@mam-host]# cat /opt/mam/etc/mam-server.conf | grep database  
...  
database.user = mam  
database.password = changeme!
```

MAM

```
[root@mam-host]# exit
#####
```

Back on the host where Postgres 9.6 is running switch to the Postgres user (since root is not a Postgres admin by default). Create the mam database. Also create a user/password pair that matches the database.user/database.password you just found in mam-server.conf.

```
#####
[root]# su - postgres
[postgres]$ psql
> CREATE USER mam WITH PASSWORD 'changeme!';
> CREATE DATABASE mam WITH OWNER=mam;
> \q
#####
```

Create a dump of your existing mam database. Your existing mam database may be on a separate host then the host where the postgres 9.6 server is. So ssh into the mam host and create the dump.

```
#####
[root]# ssh root@mam-host
[root]# service mam stop
[root@mam-host]# su - postgres
[postgres@mam-host]$ pg_dump mam > /tmp/mam.sql
#####
```

Copy the database dump (e.g. /tmp/mam.sql) to the host where you are installing postgres 9.6

```
#####
[postgres@mam-host]$ scp /tmp/mam.sql root@new-postgres-host:/tmp
[postgres@mam-host]$ exit
[root@mam-host]$ exit
#####
```

Back on the host where postgres 9.6 is being installed apply the dump to the newly created (and currently empty) mam database.

```
#####
[root]# su - postgres
[postgres]$ psql mam < /tmp/mam.sql
[postgres]$ exit
#####
```

It would be a good idea to test that the new Postgres 9.6 database can be connected to from the mam host. This will verify postgres is set up to listen for external connections properly and that port 5432 is open.

MAM

```
#####  
[root]# ssh root@mam-host  
[root@mam-host]# psql -h new-postgres-host -p 5432 -U mam -d mam  
#####
```

This should open a Postgres prompt. Run \dt to list the tables

```
#####  
> \dt  
#####
```

If the dump worked correctly you should see a list of tables similar to the following.

```
#####  
Schema |          Name          | Type | Owner  
-----+-----+-----+-----  
public | g_account              | table | mam  
public | g_account_log          | table | mam  
public | g_account_user         | table | mam  
public | g_account_user_log     | table | mam  
...  
(54 rows)  
#####
```

If you see this then you successfully applied the dump the new database and you can connect to this database from the mam host. Exit the Postgres prompt by entering \q but keep a shell open on the mam host as we will use it the next step to configure MAM.

```
#####  
> \q  
#####
```

Configuring MAM to use the new 9.6 database (without touching the old database)

On the host where mam (e.g. mam-host) is running edit the MAM server configuration file. You will edit the database.datasource property to be the host where Postgres 9.6 was just installed.

```
#####  
[root@mam-host]# vim /opt/mam/etc/mam-server.conf  
#####
```

The line that configures which host MAM will connect to for the database will look something like this

```
#####  
database.datasource = DBI:Pg:dbname=mam;host=localhost
```

MAM

#####

Change the "localhost" portion of this so that it points to the new host where you just intalled Postgres 9.6. For example if Postgres 9.6 is installed on a server called new-postgres-host this line should look like this

#####

```
database.datasource = DBI:Pg:dbname=mam;host=new-postgres-host
```

#####

Restart MAM

#####

```
[root@mam-host]# service mam restart
```

#####

Tell Moab to reread from the resource managers and make sure Moab is able to communicate with MAM.

#####

```
[root@mam-host]# mschedctl -r  
scheduling will resume immediately
```

```
[root@nativerm moab]# mdiag -R  
diagnosing resource managers
```

...

```
AM[mam] Type: MAM State: 'Active'
```

#####

Submit a job that will run for at least 3 minutes (180 seconds)

#####

```
[root@mam-host]# su - jjones -c "echo sleep 180 | msub -l walltime=180"  
Moab.1234
```

#####

Once the job has completed make sure there is a usage record for this job in MAM

#####

```
[root@mam-host]# /opt/mam/bin/mam-list-usagerecords -J Moab.1234
```

...

```
6703 Job Moab.1234      0.00 End  jjones jjones chemistry ...
```

#####

Now let's check and make sure that MAM is writing to the new Postgres 9.6 database instead of the old database. Go back to the host where Postgres 9.6 was

MAM

installed.

```
#####  
[root@mam-host]# exit  
#####
```

Query the Postgres 9.6 database, and make sure the usage record is there. We can query the database for the most recent 10 job usage records by doing the following.

```
#####  
[root]# su - postgres  
[postgres]$ psql -d mam  
> SELECT g_instance,g_end_time FROM g_usage_record ORDER BY g_end_time  
DESC LIMIT 10;  
#####
```

It it works we'll see a row for the job we just ran

```
#####  
g_instance | g_end_time  
-----+-----  
Moab.1234 | 1601401564  
70       | 1598997376  
69       | 1598997372  
Moab.11  | 1598997167  
Moab.10  | 1598993467  
#####
```

If you see the job's usage record appear in the new Postgres 9.6 database it means you have successfully connected MAM to this new database.

Unique solution ID: #1229
Author: Nate Seeley
Last update: 2020-11-18 20:06