Readme

There are 2 tests in this package. They are

1. test-postgres.sh
2. update-postgres.sh

test-postgres.sh

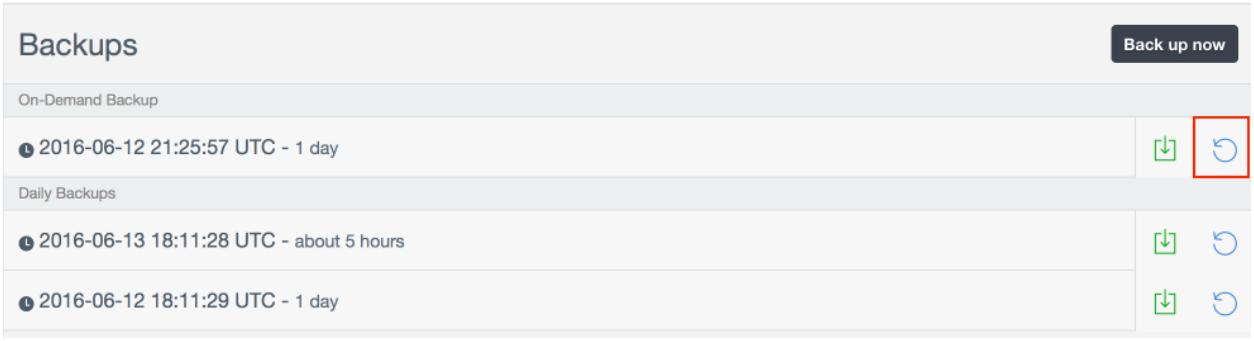
This tests the postgres database on compose by a node.js application on bluemix.

1. Create a Postgres database using compose (compose.io)
2. Create a micro service in bluemix.
    a. This can be achieved using bluemix console
    b. This also can be added by using create-service-postgres.sh. The usage of this is ./create-service-postgres.sh <hostname> <port no> <dbname> <userid> <password>.
3. Run test-postgres.sh. This will show whether the connection to the postgres database is successful or not.

Backup postgres database in compose.io

To backup the existing database, need to login to compose.io website. By default the database will be backed up everyday. Either the scheduled backup can be used or new backup can be created on demand. Once the database is backed up, it looks like

Figure 1



Restore a backed up database

To restore a backed up database, select the back version that needs to be restored and select the restore icon (circled red in the above picture). Complete it by giving a new name to the database.

Once the database is restored, get the database details. These details are needed to update the service and application in Bluemix. For Example after restoration the database connection details look like
postgres://[username]:[password]@aws-us-east-1-portal.19.dblayer.com:10371/compose

update-postgres.sh

This script updates the node.js application that deployed by test-postgres.sh. This also updates the micro service that was created to connect to the database. The usage of the script is

./update-postgres.sh <hostname> <port no> <dbname> <userid> <password>