Testing Bahmni DB Restore

Using Docker

Background

- Problem Statement
  - Weekly once, backups from all clients get copied to the central server
  - To test the sanctity of these backups every backup should be restored
    - In case of failure to restore, PostgreSQL is not able to start and is required to be reinstalled.
- Solution
  - Use Docker to test the sanctity of backups
    - Create Docker Image for PostgreSQL and MySQL
    - Create a script to use these image for restore – bahmni-db-restore
    - Automate the script through cron job
  - Advantages of using Docker for Restore
    - If crashed/failed, Docker image can easily be recycle

How to use bahmni-db-restore?

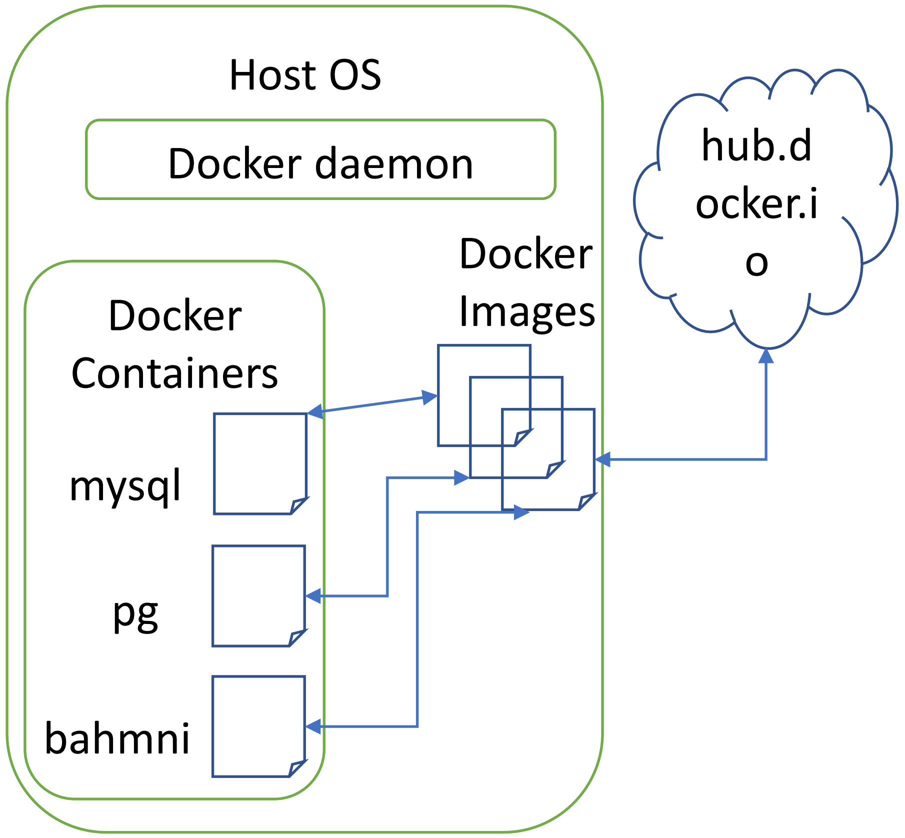
- Backup tar files are required to have
  - For ODOO – opener<week # of the month>.tar.gz, openerpbackup_info<week # of the month>.txt and openerpbfbackup.info<week # of the month>
  - For OpenMRS - openmrs<week # of the month>.tar.gz, openmrs_backup_info<week # of the month>.txt
- Clone and copy all files to /root
- Copy backups in the respective client folders in /root/backup/<client 1>… /root/backup/<client 2>… /root/backup/<client n>…
- Clean old restores from /root/openmrs
- Run using a suitable option
  - Restore latest backupl
    - Docker_run_bahmniDBRestore –d<[mysq|l<[mysq|lpg]>]> -l
  - Restore backup of specific week of the month
    - Docker run_bahmniDBRestore -d<[mysq|lpg]> -n<week # of the month>
  - Restore latest / specific week's backup of a specific client
    - Docker_run_bahmniDBRestore -d<[mysq|lpg]> -c<client folder name> [-l | -n<specific week #>
  - Restore all backups for current week
    - Docker_run_bahmniDBRestore -d<[mysq|lpg]>
  - Restore all backups but exclude specific client
    - Docker_run_bahmniDBRestore -d<[mysq|lpg]> -e<client folder name to be excluded>
- Check status in /root/<mysql|pg>-db-restore.txt and detailed log in /root/<client foldername-mysql|pg>-db-restore.log
- WIP - SMS is sent listing failed backup
- All containers are cleaned at the end of the script

Use Case – All Restored but few failed

- Check status in /root/<[mysql|pg]>-db-restore.txt and detailed log in /root/<client foldername-[mysql|pg]>-db-restore.log
- If that does not reveal much start the docker container manually and use restore
  - On host command prompt
    - export DB=mysql
    - docker run --privileged -d --name=bahmni$DB -e container=docker -v /root/:/data bjkdoc/bahmni-$DB-restore
    - docker exec -it bahmni$DB bash
  - On the container command promptrestore latest/specific week's failed backup
    - /data/restore.sh -d<[mysq|lpg]> -c<failed client folder name> [-l | -n<specific week #>
    - E.g. if Client1 backup filaed then to use any of the
      - /data/restore.sh -d<[mysq|lpg]> -cclient1 –l
      - /data/restore.sh -d<[mysq|lpg]> -cclient1 –n2
    - Since the container is live the specific database can be checked for details

Basics of Docker image

- Docker service
  - Service docker [start|stop|status]
- Download docker images from hub.docker.io
  - Docker pull <repo path/name of the image:tag>
  - Eg docker pull ramashishjoshi/bahmni-pg-restore:latest
- Activate the docker image which is known as container
  - docker run --name=<container name> <repo path/name of the image:tag>
  - Eg docker run –name=bahmni-pg ramashishjoshi/bahmni-pg-restore:latest
- Use the activated docker image ie container
  - docker exec –it –name= =<container name> <command>
  - Eg docker run –name=bahmni-pg bash
- List Docker images downloaded on local host
  - docker images
- List docker
  - docker container ps -a



Demo