OS Repositories in Docker Containers

Summary:

The task involves creating Docker images and corresponding container management script for local mirror repositories of various Linux based Operating Systems flavors and versions. By creating OS package repositories in Docker containers, the deployment architecture would need only the Docker host server on which repository docker containers for various OS flavours can be deployed. This approach reduces the hardware resource requirements and allows the system administrator to provide local repositories for multiple OS flavours and versions on a single host instead of a dedicated host for each OS flavour and version. This also helps in reducing the Opex and streamlining the IT operations.

Prerequisites:

The solution would require the following components:

1. Machine with Centos 7 which would be docker host
2. Adequate disk space for the configured repositories
3. Docker host machine should have high speed internet connection

Solution Architecture:

The docker package is installed on Host machine for managing the docker images and containers. Each OS flavour and version has an independent docker image and container for providing the local repository. The repository container has the OS repository and also exposes the repository through HTTP/HTTPS using Nginx web server. There is an additional docker image and container which provides the reverse proxy functionality using Nginx. The Proxy container listens on HTTP/HTTPS for web requests and forwards the requests to appropriate OS repository containers based on the request URI endpoint.

Components

Installation Script:

1. Script Path: <REPOS SRC>/repos_install.sh
2. Reads configuration from /etc/repos_install.conf
3. Removes the previous containers and images for each of the configured OS flavours and versions.
4. Pull the Docker image for each of the configured OS flavour and version, and Nginx Proxy from configured Jetpatch Docker Hub repository.
5. Create Docker volumes for each of the configured OS flavours and versions, if not available. (Note: Old Docker volumes are not deleted)
6. Creates a Docker Network named ‘repos_network’. All the containers are part of this network.

The Management Script

1. Script path - /bin/jetpatch_repos.sh
2. Reads config from /etc/repos.conf
3. Start/Stop/restart the repository containers

Repository Containers

Each of the repository containers have the following features:

1. The repositories are created at /var/www/html/.
2. Docker volume named as <os>_repos_vol mounted at /var/www/html
3. Nginx web server with Root Directory at /var/www/html/
4. Repository created at start of container
5. Repository updated every 24 hours
6. The repository containers retain the cache in the corresponding Docker volumes. Restart of container will use the previously downloaded repository cache.

Nginx Proxy Container

Nginx Proxy container runs Nginx web server. The web server implements a reverse proxy and https termination for the repository containers. The OS repositories can be accessed using Host IP and OS name as endpoint. The reverse proxy supports both HTTPS and HTTPS requests.

Ex: To access RHEL 7 repository, URL to use would be

http://<Host IP>/rhel7/ or *https://<Host IP>/rhel7/*

Installation & Management Steps:

1. Login to Host Centos 7 system as ‘root; user, create a work folder and cd to the same.

```
# mkdir work_repos
# cd work_repos
```

2. Copy the given tarball(jetpatch_repos.tar.gz) in this new folder.
3. Untar the tarball using following command

```
# tar -xzf jetpatch_repos.tar.gz
```

4. Edit the installation config file ‘*repos_install.conf’* and configure the OS flavours and version for which Docker image/containers of local repositories have to be installed
   Ex: Enabling local eCentos 7 and RHEL 8

```
OS_REPOS=”centos7 rhel8”
```

5. Edit the Docker Hub account and repository details from which the Docker images have to be pulled. Currently images are at the following location

```
DOCKER_HUB_ACCOUNT="itjetpatch/repos"
```

6. Execute the installation *script repos_install.sh*

```
#sh repos_install.sh
```

7. Verify the Docker images have been installed

```
# docker images
REPOSITORY          TAG             IMAGE ID      CREATED       SIZE
itjetpatch/repos    rhel8_repos_img    eeb77a198132   6 days ago
324MB
```

```
itjetpatch/repos        centos7_repos_img    d145b88e9904   6 days ago
683MB
```

8. Copy the sample local repository configuration file from *extras/repos.conf* to */etc/repos.conf*

```
# cp extras/repos.conf /etc/repos.conf
```

9. Edit the /etc/repos.conf file to enable the required repository containers and other configurations.

10. Start the Docker containers using the following command

```
# jetpatch_repos.sh start
```

11. Verify that all the configured containers are started

```
# docker ps
```

12. To stop the containers, use the following command

```
# jetpatch_repos.sh stop
```

13. To cleanup everything including docker containers, images, volumes and network, use the command from the *work_repos* folder.

```
#sh repos_install.sh cleanupall
```

14. To cleanup the docker container, image and volume for specific OS repo, run the following command.

```
#sh repos_install.sh clean <os>
Ex:
#sh repos_install.sh clean rhel7
```

The same can also be done in the following way.
Edit the config file ‘*repos_install.conf*’ and set the OS_REPOS variable to list of OS’s to be cleaned up
Ex: OS_REPOS=”centos7 rhel8”
Then run the following command from the *work_repos* folder.

```
#sh repos_install.sh clean
Ex:
#sh repos_install.sh clean
```

Endpoint URL of Repositories to be Configured on Client Machines

Centos 6

http://<HOST-IP>/centos6/base/
http://<HOST-IP>/centos6/centosplus/
http://<HOST-IP>/centos6/extras/
http://<HOST-IP>/centos6/updates/

Centos 7

http://<HOST-IP>/centos7/repos/base/
http://<HOST-IP>/centos7/repos/centosplus/
http://<HOST-IP>/centos7/repos/extras/
http://<HOST-IP>/centos7/repos/updates/

Centos 8

http://<HOST-IP>/centos8/centos/8/x86_64/baseos/
http://<HOST-IP>/centos8/centos/8/x86_64/appstream/
http://<HOST-IP>/centos8/centos/8/x86_64/extras/

RHEL 7

http://<HOST-IP>/rhel7/main/
http://<HOST-IP>/rhel7/server/
http://<HOST-IP>/rhel7/rhscl/

RHEL 8

http://<HOST-IP>/rhel8/baseOS/

http://<HOST-IP>/rhel8/AppStream/

Ubuntu 18.04

http://<HOST-IP>/ubuntu18/mirror/
http://<HOST-IP>/ubuntu18/skel/
http://<HOST-IP>/ubuntu18/var/

Ubuntu 20.04

http://<HOST-IP>/ubuntu20/mirror/
http://<HOST-IP>/ubuntu20/skel/
http://<HOST-IP>/ubuntu20/var/

Deliverables:

Following are the deliverables for the task

1. Docker images pushed to Jetpatch account on Docker Hub public repository for the following OS flavors
    a. Centos 6
    b. Centos 7
    c. Centos 8
    d. RHEL 7
    e. RHEL 8
    f. Ubuntu 18.04
    g. Ubuntu 20.04

2. Source code for creating the repository docker images for supported OS flavours and versions
3. Installation script - for installing the docker images/containers on the host machine
4. Management script to manage these images/containers on the host machine