Docker Private Repository setup

Prerequisites:

2 ubuntu 14.04 machines :- one will act as registry server , another act as a client for testing

Steps:

Step 1: Login to first server (registry server) and setup hostname

Example:

root@registry:~/docker-registry# vim /etc/hostname
registry.krishna.com

root@registry:~/docker-registry# hostname registry.krishna.com
root@registry:~/docker-registry#

Step 2: Install Docker and Docker compose packages on server

- wget -qO- https://get.docker.com/ | sh
- sudo usermod -aG docker $(whoami)
- sudo apt-get -y install python-pip
- sudo pip install docker-compose
- apt-get install apache2-utils

Step 3: Installing and Configuring the Docker Registry

- mkdir ~/docker-registry && cd $_
- mkdir data
- mkdir ~/docker-registry/nginx
- vim docker-compose.yml

#Add the following contents to the file:

```
nginx:
  image: "nginx:1.9"
  ports:
    - 443:443
  links:
    - registry:registry
  volumes:
    - ./nginx/:/etc/nginx/conf.d
```

```
registry:
  image: registry:2
  ports:
    - 127.0.0.1:5000:5000
  environment:
    REGISTRY_STORAGE_FILESYSTEM_ROOTDIRECTORY: /data
  volumes:
    - ./data:/data
```

- Vim ~/docker-registry/nginx/registry.conf

#Copy the following into the file

```
upstream docker-registry {
  server registry:5000;
}

server {
  listen 443;
  server_name registry.krishna.com;

  # SSL
  ssl on;
  ssl_certificate /etc/nginx/conf.d/domain.crt;
  ssl_certificate_key /etc/nginx/conf.d/domain.key;

  # disable any limits to avoid HTTP 413 for large image uploads
  client_max_body_size 0;

  # required to avoid HTTP 411: see Issue #1486 (https://github.com/docker/docker/issues/1486)
  chunked_transfer_encoding on;

  location /v2/ {
    # Do not allow connections from docker 1.5 and earlier
    # docker pre-1.6.0 did not properly set the user agent on ping, catch "Go *" user agents
    if ($http_user_agent ~ "^(docker\/1\.(3|4|5(?!\.[0-9]-dev))|Go ).*$" ) {
      return 404;
    }

    # To add basic authentication to v2 use auth_basic setting plus add_header
    auth_basic "registry.localhost";
    auth_basic_user_file /etc/nginx/conf.d/registry.password;
    add_header 'Docker-Distribution-Api-Version' 'registry/2.0' always;

    proxy_pass                          http://docker-registry;
    proxy_set_header  Host              $http_host;   # required for docker client's sake
    proxy_set_header  X-Real-IP         $remote_addr; # pass on real client's IP
```

```
    proxy_set_header  X-Forwarded-For   $proxy_add_x_forwarded_for;
    proxy_set_header  X-Forwarded-Proto $scheme;
    proxy_read_timeout                  900;
  }
}
```

- cd ~/docker-registry/nginx
- htpasswd -c registry.password krishna

Note: If you want to add more users in the future, just re-run the above command without the -c option (the cis for create):

- cd ~/docker-registry/nginx
- openssl genrsa -out devdockerCA.key 2048
- openssl req -x509 -new -nodes -key devdockerCA.key -days 10000 -out devdockerCA.crt
- openssl genrsa -out domain.key 2048
- openssl req -new -key domain.key -out dev-docker-registry.com.csr

Note: update the FQDN

Common Name (e.g. server FQDN or YOUR name) []:**registry.krishna.com**

- openssl x509 -req -in dev-docker-registry.com.csr -CA devdockerCA.crt -CAkey devdockerCA.key -CAcreateserial -out domain.crt -days 10000

- sudo mkdir /usr/local/share/ca-certificates/docker-dev-cert
- sudo cp devdockerCA.crt /usr/local/share/ca-certificates/docker-dev-cert
- sudo update-ca-certificates
- sudo service docker restart
- cd ~/docker-registry
- sudo mv ~/docker-registry /docker-registry
- sudo chown -R root: /docker-registry
- Vim /etc/init/docker-registry.conf

paste the content

```
description "Docker Registry"

start on runlevel [2345]
stop on runlevel [016]

respawn
respawn limit 10 5

chdir /docker-registry

exec /usr/local/bin/docker-compose up
```

- sudo service docker-registry start
- docker ps
- ##test it by running curl command with registryaddress , you should get output "{}"

Example:

root@registry:~/docker-registry# curl -k https://krishna:raju@registry.krishna.com/v2/
{}

Test from Client machine

- Login to client machine
- Add the registry server hostname and ip in hosts file

Example:

root@ip-172-31-85-58:~# vim /etc/hosts
35.174.172.231 registry.krishna.com

- Go to registry server copy the content of the file "/docker-registry/nginx/devdockerCA.crt"
- Then go back to client machine and create a folder

mkdir /usr/local/share/ca-certificates/docker-dev-cert

- Vim /usr/local/share/ca-certificates/docker-dev-cert/devdockerCA.crt

paste the content which you copied from server

- sudo update-ca-certificates
- sudo service docker restart
- docker login https://YOUR-DOMAIN

Example:

root@ip-172-31-85-58:~# docker login https://registry.krishna.com
Username: krishna
Password:
Login Succeeded
root@ip-172-31-85-58:~#

- Create small container and push it to repo

Example:

root@ip-172-31-85-58:~# docker run -t -i ubuntu /bin/bash

root@ip-172-31-85-58:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
950200c884d7 ubuntu "/bin/bash" 6 seconds ago Up 5 seconds friendly_torvalds

root@ip-172-31-85-58:~# docker commit 950200c884d7 ubuntu:14.04

```
sha256:b2d867dff5019c7ac43e135d1e1c7dfd27ce53f0836e0189a10df7e98d681a60
root@ip-172-31-85-58:~#


root@ip-172-31-85-58:~# docker tag ubuntu:14.04
registry.krishna.com/ubuntu:14.04

root@ip-172-31-85-58:~# docker push registry.krishna.com/ubuntu:14.04
The push refers to repository [registry.krishna.com/ubuntu]
db584c622b50: Pushed
52a7ea2bb533: Pushed
52f389ea437e: Pushed
88888b9b1b5b: Pushed
a94e0d5a7c40: Pushed
14.04: digest:
sha256:39cb02c3c97394cdf14693eabb91a8aa6137a48f7b24ba6550823fdf0bbd9cb8
size: 1357
root@ip-172-31-85-58:~#
```