QVD Disk Image Creation (Ubuntu 12.04)

QVD DOCUMENTATION

<documentation@theqvd.com>

July 11, 2017

Contents

Introduction

The QVD provides demo desktop images for use with the product, but sooner or later the serious user will need to consider building their own desktops, tailored to the needs of their particular users.

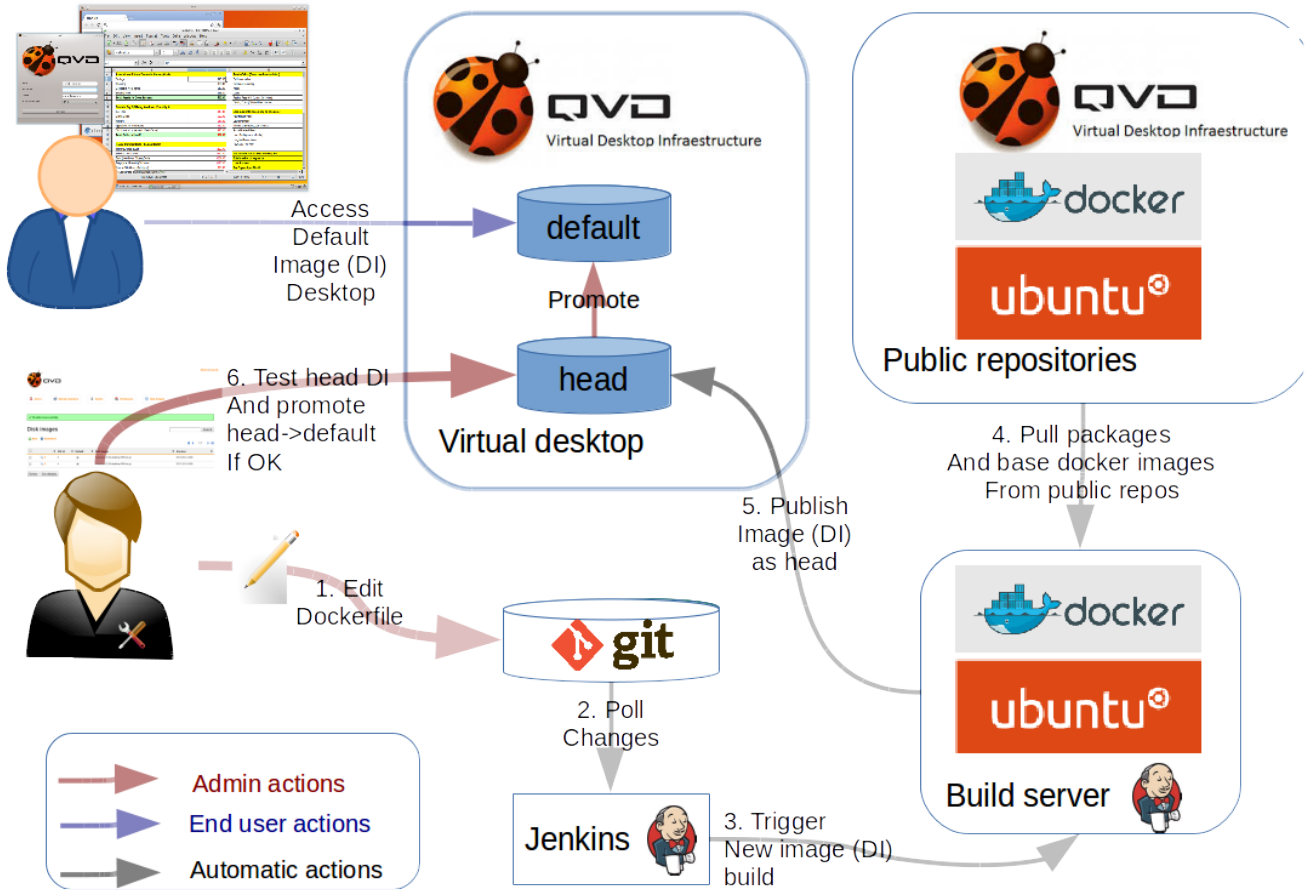
This guide aims to explain this process using a DevOps Methodology by using a continuous integration pipeline, and to give the administrator the basic building blocks for their own desktop images.

This document was last updated: 2015-05-19

Its current revision number is set at: 26935

1 General process overview

In the following picture you will find the general process outlined which will be expanded in this document.



1.1 Components

The components used are the following:

- QVD server: QVD installation with LXC images backend
- Disk Image: Xubuntu as the Linux/Desktop distribution. In this example only one OSF is used.
- GIT: Using git to track the changes to the Docker configuration.
- Docker: Using Docker to create the disk image tar.gz file.
- Public repositories: For Docker, Ubuntu and QVD
- Jenkins: Using Jenkins to create and publish those images.
- QVD client: Using the QVD client to validate that everything works

1.2 Roles

We define three different 4 different roles:

- The end user: who uses the QVD platform. In the picture this is the blue dressed user.
- The QVD administrator: who promotes the disk image from the test stage to the production stage. In the picture this is the black dressed user.
- The test team: who validates the new image. In the picture this role is also assumed by the black dressed user.
- The disk image developer: who makes the changes to the Disk Image. In the picture this role is also assumed by the black dressed user.

2 Process

For this process, we assume that the current image used is the basicdesktop for qvd published in the Docker Hub repository.

The process is as follows:

0) The end user passes some new requirements to the disk image developer to add the gimp package.

1) The disk image developer, updates Dockerfile in the git repository with the following info:

```
FROM theqvd/qvdimageubuntu:basicdesktop
MAINTAINER Me <me@domain.com>

ENV DEBIAN_FRONTEND noninteractive
RUN apt-get update && apt-get install -y gimp

# Cleanup
RUN echo "" > /etc/udev/rules.d/70-persistent-net.rules
RUN apt-get autoremove -y
RUN apt-get clean
```

2) Jenkins is polling the repository for changes (this can be polling or on change trigger based)

3) Once commited Jenkins will kickoff a build process in the build server (Ubuntu 14.04 server with docker) installed. Please see the link:DiskImageCreationUbuntu.html for details on how this server is created.

4,5) Jenkins triggers a build. The script is basically creating the image and publishing it in the qvd server.

```
sudo docker build -t theqvd/qvdimageubuntu:mydiskimage .
vmid=$(sudo docker run -d -t -i theqvd/qvdimageubuntu:mydiskimage /bin/bash -c "read a;  ↩
    echo $a")
sudo docker export $vmid  | gzip -c > qvd-image-ubuntu-14.04-mydiskimage.tgz
sudo docker kill $vmid
rsync -av qvd-image-ubuntu-14.04-mydiskimage.tgz qindel@qvdserver:/var/lib/qvd/storage/ ↩
    staging/
ssh qvdserver 'sudo qa di add path=/var/lib/qvd/storage/staging/qvd-image-ubuntu-14.04- ↩
    mydisimage.tgz osf_id=1'
ssh qvdserver 'sudo qa vm stop -f di_tag=head'
ssh qvdserver 'sudo /usr/local/bin/delete_unused_overlays_and_basefs.sh' || echo "Error in  ↩
    delete_unused_overlays_and_basefs.sh see the log"
sleep 30
ssh qvdserver 'sudo qa vm start -f di_tag=head'
```

6) The test team validates the image and tells the QVD administrator to publish it. The QVD administrator runs the following to publish the image

```
mytag=head
targettag=default
head_di=$(sudo qa di list -f osf_id=1,tag=$mytag | tail -n +3 | awk "{print \$1}")
[ -z $head_di ] && echo "di with tag $mytag not found" && exit 1
sudo qa di tag di_id=$head_di tag=$targettag expire-soft=now expire-hard=tomorrow'
```

3 Enhancements

Some enhancements can be done in this process, particularly if you need to speed up the build process and/or you are using a highly controlled/secured environment where you don't want to rely on updates on external repositories:

- Creating internal mirrors for your preferred distribution/desktop
- Alternatively using prxoy caches to speed up the process
- Using RAM based filesystems
- Customize the disk image expire hooks for seamless desktop upgrades

Feel free to contact theqvd team if you have further queries

4 Useful Links

Some useful links:

- Configure Jenkins polling or post commit trigger. https://wiki.jenkins-ci.org/display/JENKINS/Git+Plugin
- Dockerfile reference documentation. http://docs.docker.com/reference/builder/
- Docker Hub. https://docs.docker.com/userguide/dockerrepos/
- QVD Documentation. http://docs.theqvd.com
- Ubuntu repositories. https://help.ubuntu.com/stable/ubuntu-help/addremove-sources.html and http://packages.ubuntu.com/