

3.5. DOCKER ON DL160

a. Configure Harbor

(1) Login

(a) Browse to **dockervm.dmss**.

NOTE:

If issues browsing to dockervm.dmss try command: *systemctl restart cvah.dockervm*.

(b) Login with:

1. **admin**

2. **password**

VMware Harbor™

admin

.....

☐ Remember me [Forgot password](#)

LOG IN

Popular Repositories

Name

Figure 1. Harbor Configuration

(2) Change the Admin Password

(a) Change the administrator password by going to the top right, press the **admin dropdown**, select **Change Password**.

1. Enter your current password and your new passwords and select **ok**.

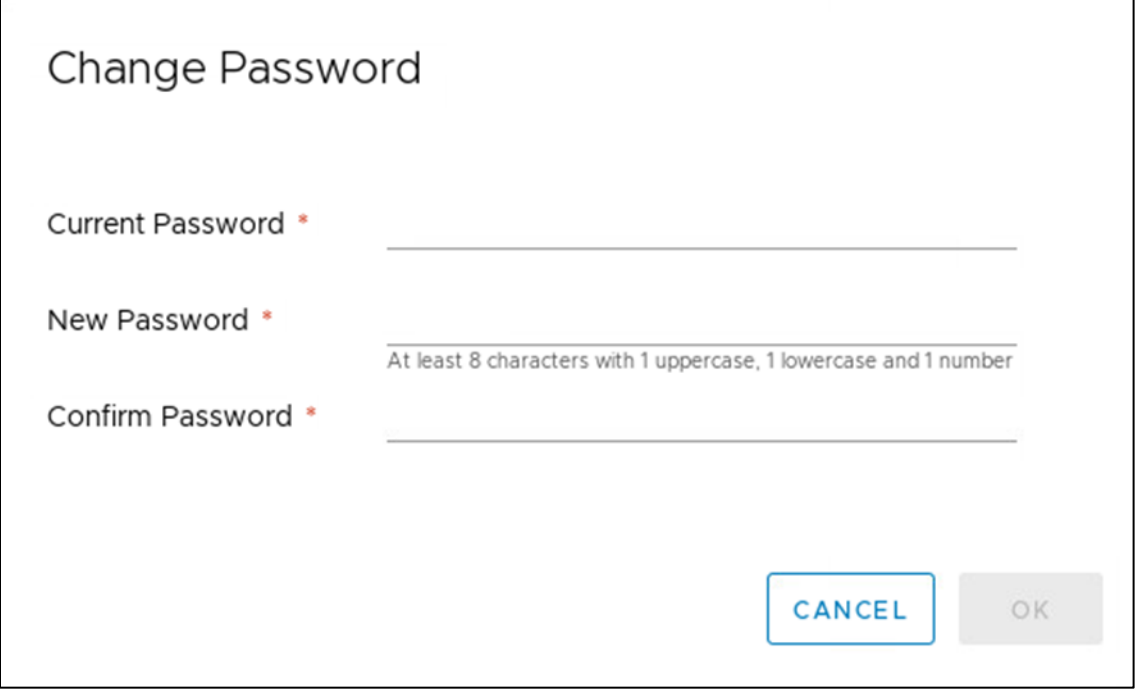
A screenshot of a 'Change Password' form. The form has a title 'Change Password' at the top. Below it are three input fields: 'Current Password *', 'New Password *', and 'Confirm Password *'. The 'New Password *' field has a hint text below it: 'At least 8 characters with 1 uppercase, 1 lowercase and 1 number'. At the bottom right of the form are two buttons: 'CANCEL' and 'OK'.

Figure 2. Harbor Credentials

(3) Create a New Project

- (a) Create a new project by going to the left-hand side, select **Projects**, select **+PROJECT**, add a **Project Name**, and press **OK**.

NOTE:

Remember this project name. You will need it to push containers to harbor.

1. Check the box **Public** unless you want to configure either TLS certificates or HTTP auth for your registry.

WARNING:

The PMO has not tested Harbor with HTTP authentication. You do this at your own risk.

WARNING:

The name of the registry must match exactly. You cannot use an IP address in place of a domain name or vice versa.

4.3. DOCKERVM

WARNING:

On some imports of docker we had trouble with the PID file preventing docker from running. If `systemctl status docker` says inactive after import try running `rm -f /var/run/docker.pid && systemctl start docker`. See the troubleshooting section for more details.

a. Harbor

The Breakdown of Terminology in Docker section below does not describe mandatory steps, but provides a background on interpreting Docker output. The PMO highly recommends reading this section before working with Harbor.

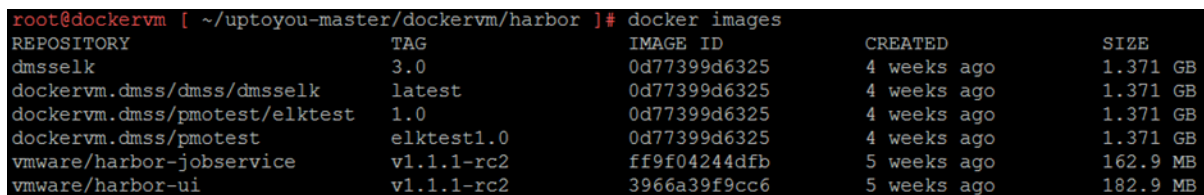
(1) Breakdown of Terminology in Docker

(a) Image and Container

1. A Docker image is the file whose contents become a docker container. In documentation, the words container and image are frequently used interchangeably which can be highly confusing. The distinction is a container is an image that is currently running. IE, you ran a command like **docker run**. You can have multiple containers all of which are using the same image.

(b) View Images in Docker

1. To view your Docker images, run: **docker images**.



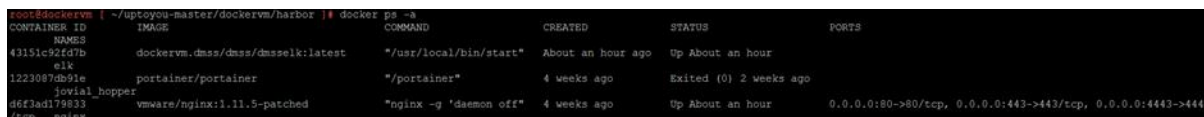
```
root@dockervm [ ~/uptoyou-master/dockervm/harbor ]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
dmsselk	3.0	0d77399d6325	4 weeks ago	1.371 GB
dockervm.dmss/dmss/dmsselk	latest	0d77399d6325	4 weeks ago	1.371 GB
dockervm.dmss/pmotest/elktest	1.0	0d77399d6325	4 weeks ago	1.371 GB
dockervm.dmss/pmotest	elktest1.0	0d77399d6325	4 weeks ago	1.371 GB
vmware/harbor-jobservice	v1.1.1-rc2	ff9f04244dfb	5 weeks ago	162.9 MB
vmware/harbor-ui	v1.1.1-rc2	3966a39f9cc6	5 weeks ago	182.9 MB

Figure 3. Docker Images List

(c) View Containers in Docker

1. To view the containers which are currently running use: **docker ps**.



```
root@dockervm [ ~/uptoyou-master/dockervm/harbor ]# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
43151c92fd7b	dockervm.dmss/dmss/dmsselk:latest	"/usr/local/bin/start"	About an hour ago	Up About an hour	
elk	portainer/portainer	"/portainer"	4 weeks ago	Exited (0) 2 weeks ago	
1223087db91e	portainer/portainer	"/portainer"	4 weeks ago	Exited (0) 2 weeks ago	
joval hopper	portainer/portainer	"/portainer"	4 weeks ago	Exited (0) 2 weeks ago	
d6f3ad179833	vmware/nginx:1.11.5-patched	"nginx -g 'daemon off'"	4 weeks ago	Up About an hour	0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp, 0.0.0.0:4443->4443/tcp
/cop noing	portainer/portainer	"/portainer"	4 weeks ago	Exited (0) 2 weeks ago	

Figure 4. List of Containers in Docker

NOTE:

docker ps only shows the running containers. Containers may also be stopped. To view both running and stopped containers, use docker ps -a.

(d) Tag and Repository

```
root@dockervm [ ~/uptoyou-master/dockervm/harbor ]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
dockervm.dmss/dmss/dmsselk	latest	0d77399d6325	4 weeks ago	1.371 GB
dockervm.dmss/pmotest/elktest	1.0	0d77399d6325	4 weeks ago	1.371 GB
vmware/harbor-jobservice	v1.1.1-rc2	ff9f04244dfb	5 weeks ago	162.9 MB
vmware/harbor-ui	v1.1.1-rc2	3966a39f9cc6	5 weeks ago	182.9 MB
vmware/harbor-adminserver	v1.1.1-rc2	fabdeab472a3	5 weeks ago	141.6 MB
vmware/harbor-db	v1.1.1-rc2	ca166fa6033c	5 weeks ago	328.5 MB
phusion/baseimage	latest	23a90f4f77f2	8 weeks ago	247.5 MB
vmware/harbor-notary-db	mariadb-10.1.10	64ed814665c6	8 weeks ago	324.1 MB
vmware/nginx	1.11.5-patched	8ddadb143133	8 weeks ago	199.2 MB
dockervm.dmss/test/portainer	latest	89883cee365b	9 weeks ago	9.958 MB
portainer/portainer	latest	89883cee365b	9 weeks ago	9.958 MB
vmware/notary-photon	signer-0.5.0	b1eda7d10640	10 weeks ago	155.7 MB
vmware/registry	photon-2.6.0	6cb4438d7197	11 weeks ago	145.7 MB
vmware/notary-photon	server-0.5.0	6e2646682e3c	11 weeks ago	156.9 MB
vmware/harbor-log	v1.1.1-rc2	9c46a7b5e517	3 months ago	192.4 MB
photon	1.0	e6e4e4a2ba1b	11 months ago	127.5 MB

Figure 5. Docker Repositories

- (e) The image tag is the portion after the colon (or in the output above, the portion under the tag column). The name of an image is the repository + the tag. For example, the first entry's full name in the output above is *dockervm.dmss/dmss/dmsselk:latest*.
- (f) The tag is "latest" and the repository name is *dockervm.dmss/dmss/dmsselk*. You'll notice that both *dockervm.dmss/dmss/dmsselk* and *dockervm.dmss/pmotest/elktest:1.0* have the same image ID. A repository in the context of docker is not the same as a repository in the rest of computer science.
- (g) A repository is a collection of docker images (possibly the same image or maybe different images) with the same repository name, but have different tags. For example, I could rename Portainer to place it in the *dockervm.dmss/dmss/dmsselk* repository and just give it a different tag:

```
root@dockervm [ ~/uptoyou-master/dockervm/harbor ]# docker tag portainer/portainer:latest dockervm.dmss/dmss/dmsselk:IsSecretlyPortainer
root@dockervm [ ~/uptoyou-master/dockervm/harbor ]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
dockervm.dmss/dmss/dmsselk	latest	0d77399d6325	4 weeks ago	1.371 GB
dockervm.dmss/pmotest/elktest	1.0	0d77399d6325	4 weeks ago	1.371 GB
elk	3.0	0d77399d6325	4 weeks ago	1.371 GB
vmware/harbor-jobservice	v1.1.1-rc2	ff9f04244dfb	5 weeks ago	162.9 MB
vmware/harbor-ui	v1.1.1-rc2	3966a39f9cc6	5 weeks ago	182.9 MB
vmware/harbor-adminserver	v1.1.1-rc2	fabdeab472a3	5 weeks ago	141.6 MB
vmware/harbor-db	v1.1.1-rc2	ca166fa6033c	5 weeks ago	328.5 MB
phusion/baseimage	latest	23a90f4f77f2	8 weeks ago	247.5 MB
vmware/harbor-notary-db	mariadb-10.1.10	64ed814665c6	8 weeks ago	324.1 MB
vmware/nginx	1.11.5-patched	8ddadb143133	8 weeks ago	199.2 MB
dockervm.dmss/dmss/dmsselk	IsSecretlyPortainer	89883cee365b	9 weeks ago	9.958 MB
dockervm.dmss/test/portainer	latest	89883cee365b	9 weeks ago	9.958 MB
portainer/portainer	latest	89883cee365b	9 weeks ago	9.958 MB
vmware/notary-photon	signer-0.5.0	b1eda7d10640	10 weeks ago	155.7 MB
vmware/registry	photon-2.6.0	6cb4438d7197	11 weeks ago	145.7 MB
vmware/notary-photon	server-0.5.0	6e2646682e3c	11 weeks ago	156.9 MB
vmware/harbor-log	v1.1.1-rc2	9c46a7b5e517	3 months ago	192.4 MB
photon	1.0	e6e4e4a2ba1b	11 months ago	127.5 MB

Figure 6. Portainer Rename

- (h) Now there is an entry *dockervm.dmss/dmss/dmsselk:IsSecretlyPortainer*. This is the portainer image, I just put it in the *dockervm.dmss/dmss/dmsselk* repository. It didn't change locations, I didn't copy it, it's the same files as *portainer/portainer*, all I changed was the repository name and tag.

WARNING:

To rename an image you use the *docker tag* command. In the context of renaming images, *docker tag* can also change the repository name.

(i) Registry

1. A docker registry is a place for storing docker images. When you install docker on a computer, it comes with a local registry automatically. When you run **docker images** you are looking at the images in your local registry. DockerHub is a registry, Harbor is a registry, your installation of Docker comes with a local registry.

(j) Naming Schemes and Uploading to a Registry

1. When you upload an image to a registry, the repository must match the name of the registry. For example, on the weapon system you have a local repository called *dockervm.dmss/dmss/dmsselk* and it has the image *dockervm.dmss/dmss/dmsselk:latest*. This repository resides in the running docker registry on docker. You have a second registry in the form of Harbor. Prepare yourself for container inception. Harbor is comprised of multiple containers. Those containers are running from your local docker registry on Docker.
2. One of the Harbor containers, contains the second registry. When you want to upload to Harbor, you must retag your images to have a repository that matches the name of the host on which Harbor resides. In this case, Harbor is residing on *dockervm.dmss*, hence when you want to upload an image to Harbor you must retag it to the structure **dockervm.dmss:<some_tag_here>**. You'll notice the weapon system version of elk has the repository name **dockervm.dmss/dmss/**. Because ELK is residing on the local docker registry, the */dmss* portion of the repository is largely arbitrary. However, when you want to upload to Harbor you will have the hostname **dockervm.dmss** and then after that must follow the folder name of your project. For example:

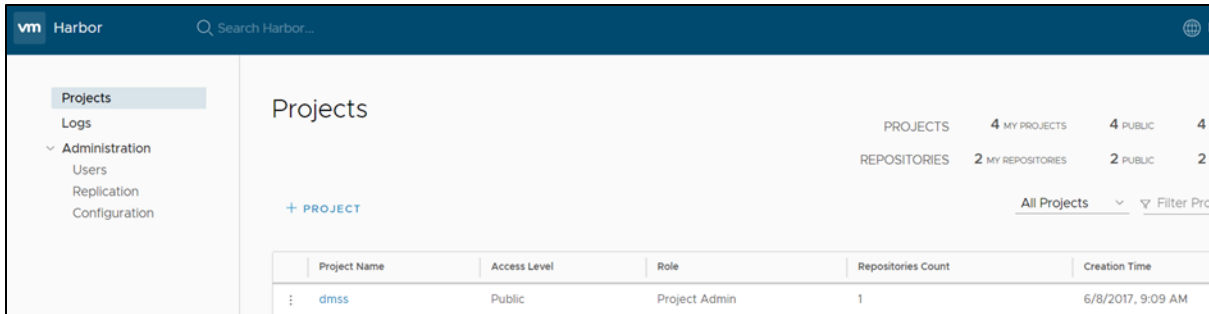


Figure 7. Project

3. In the above screenshot, you see the project name dmss. If you want to upload an image to that project the image must be tagged as *dockervm.dmss/dmss/<you_can_put_whatever_you_want_here>:<TAG>*.

(2) The PMO places all containers on the system in Harbor under the name library.

WARNING:

If the project name does not match and you attempt to push an image to the Harbor repository, Harbor will return an HTTP 500 error and give no indication whatsoever that a name mismatch is the problem.

(3) Push an Image

- (a) Begin by tagging the image with: *docker tag <current_image_name> dockervm.dmss/<project_name>/<new_image_name>* Example:

1. **docker tag 192.168.1.111/Elasticsearch \ dockervm.dmss/dmss/Elasticsearch:latest**

- a. Where 192.168.1.111 is the current registry name
- b. Elasticsearch is the current name of the image.
- c. dockervm.dmss is the name of the new registry (the registry to which we are pushing)
- d. dmss is the project name in that registry (the one you created in harbor)
- e. Elasticsearch:latest is the new image tag we are applying in the registry we are pushing to

- (b) Login to harbor with: **docker login dockervm.dmss**

1. Default Username: **admin**
2. Default Password: **password**

(c) Push the image to harbor with: **docker push **
dockervm.dmss/<project_name>/<new_image_name>

1. From the example above this would be: **docker push **
dockervm.dmss/dmss/Elasticsearch:latest

(4) Pull an Image

(a) To pull an image, use command:

1. **docker pull dockervm.dmss/<project_name>/<Image_name>**

a. Example: **docker pull dockervm.dmss/dmss/elk:1.0.0**