<https://www.codeproject.com/Articles/1263817/How-to-Setup-Our-Own-Private-Docker-Registry>

**Part 1: How to set up your private Docker registry?**

Follow the steps below to create your private Docker registry:

* Open a PowerShell console (terminal in Linux).
* First, I want you to create a folder to share with the container and it will be used in the upcoming steps.
* Navigate to *C:\* drive and create a folder with the name of ***localhub*** (md localhub). For Linux, create the same folder under ***/home***.
* Navigate to ***C:\localhub*** folder in windows or ***/home/localhub*** in Linux and create a subfolder with the name of "***registry***".
* Type the following command to pull the registry image from the docker hub:

Copy Code

**Windows/Linux:**

docker pull registry

* Before going to the docker registry set up, I want to set up a meaningful local domain name for your private registry instead of using localhost but this step is completely optional. I prefer to have the local domain as **hub.docker.local**. To configure the local domain in windows and Linux, do the following steps:
  + **Windows**
    - Open an elevated Notepad in docker host.
    - Open the *C:\Windows\System32\drivers\etc\hosts* file in the Notepad and add this **hub.docker.local 127.0.0.1**as an entry in it.
    - Save and close the file.
  + **Linux**
    - Open a terminal in docker host and type nano **/etc/hosts** or vi /etc/hosts
    - Add this hub.docker.local 127.0.0.1 as an entry in it.
    - Save and close the file.
* Spin up a container with the docker registry image:

Copy Code

**Windows:**

docker run -d -p 5000:5000 -v C:/localhub/registry:/var/lib/registry

--restart=always --name hub.local registry

**Linux:**

docker run -d -p 5000:5000 -v /home/localhub/registry:/var/lib/registry

--restart=always --name hub.local registry

*Docker registry uses the 5000 port as default. --restart=always flag is enable the auto start after the docker restarted. -v will bind the given host folder with container file system.*

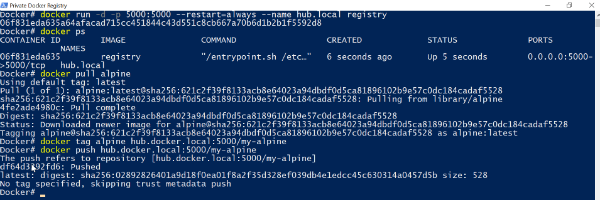
*Docker might ask you for the user name and password to share the localhub folder with container if you have not setup the share folder already in docker.*

* Ensure the registry container is up and running:

Copy Code

**Windows/Linux:**

docker ps or docker container ls



* The next step is to prepare the docker image and push to our private registry. You can use any one of the existing images to quickly understand the concepts of the Docker Registry. Alpine is one of the lite weight Linux distribution(~2MB) so you can use this image for a quick assessment:

Copy Code

**Windows/Linux:**

docker pull alpine

* Create a tag to alpine Linux with **hub.docker.local:5000/my-alpine**.

Copy Code

**Windows/Linux:**

docker tag alpine hub.docker.local:5000/my-alpine

* It creates an additional tag on an existing alpine image. Tag format will be like **registry hostname**:**port/new name**. Docker will extract the location from the given tag while pushing to your private registry
* Push the my-alpine image to your private registry:

Copy Code

**Windows/Linux:**

docker push hub.docker.local:5000/my-alpine

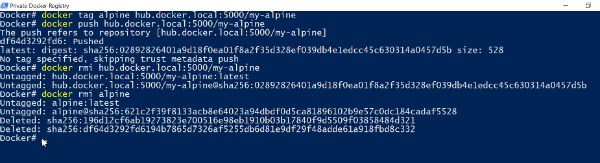
* Remove the alpine and its tagged version to ensure docker is pulling the image from your private registry instead of docker hub:

Copy Code

**Windows/Linux:**

docker rmi hub.docker.local:5000/my-alpine

docker rmi alpine



* Now, pull the my-alpine image from your private registry.

Copy Code

**Windows/Linux:**

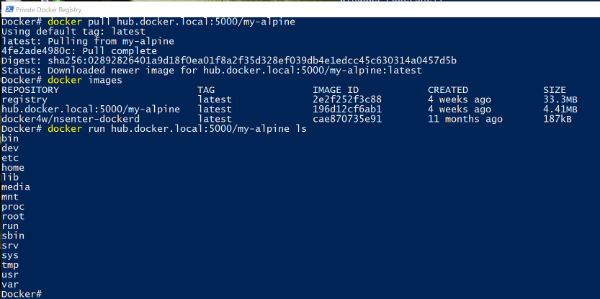
docker pull hub.docker.local:5000/my-alpine

* Spin up a container with newly downloaded image and ask the container to list out its root directory.

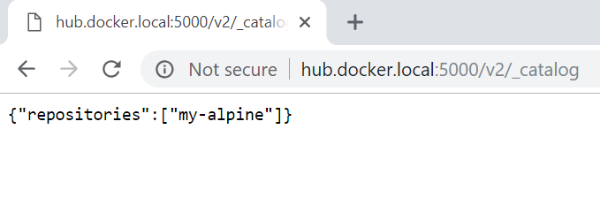
Copy Code

**Windows/Linux:**

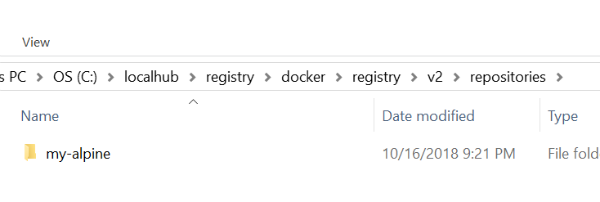
docker run hub.docker.local:5000/my-alpine ls



* You can check registry catalog on this ***http://hub.docker.local:5000/v2/\_catalog*** address.



* You can inspect the *localhub/registry* folder to understand how docker images are stored in the docker registry.



* Well done! You have successfully created your own private Docker registry.
* (Optional)Stop and remove the registry container and image to move on to the next articles.

Copy Code

**Windows/Linux:**

docker container stop hub.local

docker container rm hub.local

* (Optional)Finally, remove the my-alpine image as well.

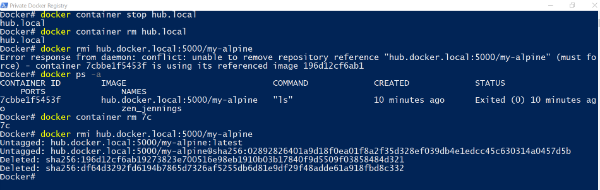
Copy Code

**Windows/Linux:**

docker ps -a

docker container rm container\_id

docker rmi hub.docker.local:5000/my-alpine



Storage Customization

Docker registry stores the images on the host file system(*/var/lib/registry*). Storing images on the file system is not a reliable solution for a production environment. You can use SSD and SAN for storing your private registry images. First, you have to mount these drives into your host, then you can easily bind to the container through volume binding. For example, you have mounted an SSD or SAN into *R:* drive in windows then you can mount these drives into the container through -v R:/registry:/var/lib/registry. For Linux, the volume binding will be like this -v /mnt/registry:/var/lib/registry.

Docker registry also supports to use storage driver compliant storage back-ends so you can use some third party storage back-ends like Amazon S3 bucket, Google Cloud Platform, etc.

Thanks for reading this article and I hope you find something useful. Please share your comments below. In the next article, I have written about [**"how to secure your private registry?"**](https://www.codeproject.com/Articles/1263831/How-to-secure-your-private-Docker-Registry).