

Push your first image to a private Docker container registry using the Docker CLI

📅 11/29/2017 ⌚ 3 minutes to read Contributors  [all](#)

In this article

[Prerequisites](#)

[Log in to a registry](#)

[Pull the official Nginx image](#)

[Run the container locally](#)

[Create an alias of the image](#)

[Push the image to your registry](#)

[Pull the image from your registry](#)

[Start the Nginx container](#)

[Remove the image \(optional\)](#)

[Next steps](#)

An Azure container registry stores and manages private [Docker](#) container images, similar to the way [Docker Hub](#) stores public Docker images. You can use the [Docker command-line interface](#) (Docker CLI) for [login](#), [push](#), [pull](#), and other operations on your container registry.

In the following steps, you download an official [Nginx image](#) from the public Docker Hub registry, tag it for your private Azure container registry, push it to your registry, and then pull it from the registry.

Prerequisites

- **Azure container registry** - Create a container registry in your Azure subscription. For example, use the [Azure portal](#) or the [Azure CLI 2.0](#).
- **Docker CLI** - To set up your local computer as a Docker host and access the Docker CLI commands, install [Docker](#).

Log in to a registry

There are [several ways to authenticate](#) to your private container registry. The recommended method when working in a command line is with the Azure CLI command [az acr login](#). For example, to log in to a registry named *myregistry*.

Azure CLI

 Copy

```
az acr login --name myregistry
```

You can also log in with [docker login](#). The following example passes the ID and password of an Azure Active Directory [service principal](#). For example, you might have [assigned a service principal](#) to your registry for an automation scenario.

Bash	Copy
<pre>docker login myregistry.azurecr.io -u xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx -p myPas:</pre>	

Both commands return `Login Succeeded` once completed. If you use `docker login`, you might also see a security warning recommending the use of the `--password-stdin` parameter. While its use is outside the scope of this article, we recommend following this best practice. For more information, see the [docker login](#) command reference.

💡 Tip

Always specify the fully qualified registry name (all lowercase) when you use `docker login` and when you tag images for pushing to your registry. In the examples in this article, the fully qualified name is *myregistry.azurecr.io*.

Pull the official Nginx image

First, pull the public Nginx image to your local computer.

Bash	Copy
<pre>docker pull nginx</pre>	

Run the container locally

Execute following [docker run](#) command to start a local instance of the Nginx container interactively (`-it`) on port 8080. The `--rm` argument specifies that the container should be removed when you stop it.

Bash	Copy
<pre>docker run -it --rm -p 8080:80 nginx</pre>	

Browse to <http://localhost:8080> to view the default web page served by Nginx in the running container. You should see a page similar to the following:



Because you started the container interactively with `-it`, you can see the Nginx server's output on the command line after navigating to it in your browser.

To stop and remove the container, press `Control + C`.

Create an alias of the image

Use [docker tag](#) to create an alias of the image with the fully qualified path to your registry. This example specifies the `samples` namespace to avoid clutter in the root of the registry.

Bash	Copy
<pre>docker tag nginx myregistry.azurecr.io/samples/nginx</pre>	

For more information about tagging with namespaces, see the [Repository namespaces](#) section of [Best practices for Azure Container Registry](#).

Push the image to your registry

Now that you've tagged the image with the fully qualified path to your private registry, you can push it to the registry with [docker push](#):

Bash	Copy
<pre>docker push myregistry.azurecr.io/samples/nginx</pre>	

Pull the image from your registry

Use the [docker pull](#) command to pull the image from your registry:

Bash

 Copy

```
docker pull myregistry.azurecr.io/samples/nginx
```

Start the Nginx container

Use the [docker run](#) command to run the image you've pulled from your registry:

Bash

 Copy

```
docker run -it --rm -p 8080:80 myregistry.azurecr.io/samples/nginx
```

Browse to <http://localhost:8080> to view the running container.

To stop and remove the container, press Control + C.

Remove the image (optional)

If you no longer need the Nginx image, you can delete it locally with the [docker rmi](#) command.

Bash

 Copy

```
docker rmi myregistry.azurecr.io/samples/nginx
```

To remove images from your Azure container registry, you can use the Azure CLI command [az acr repository delete](#). For example, the following command deletes the manifest referenced by a tag, any associated layer data, and all other tags referencing the manifest.

Azure CLI

 Copy

```
az acr repository delete --name myregistry --repository samples/nginx --tag latest
```

Next steps

Now that you know the basics, you're ready to start using your registry! Deploy container images from your registry to:

- [Azure Kubernetes Service \(AKS\)](#)
- [Azure Container Instances](#)
- [Service Fabric](#)