Instructions for Getting Started

1. **1**

Install Docker

Before you can try out the Azure Container Registry, you should install Docker.

Refer to the [Mac](https://docs.docker.com/mac/started/) or [Windows](https://docs.docker.com/windows/started/) or [Linux](https://docs.docker.com/linux/started/) getting started instructions for Docker.

1. **2**

Run the "hello-world" base image

The following command will get you a running container straight off [Docker Hub](https://hub.docker.com/_/hello-world/).

This should display "Hello from Docker!".

* 1. **docker run -it hello-world**

1. **3**

Login to your container registry

Let's login to your container registry. You can get the login URI and credentials from Access keys blade.

* 1. **docker login rpscontainer.azurecr.io**

**or**

* 1. **Az acr login –-name rpscontainer**

1. **4**

Push to your registry

Let's first prefix the image with your registry login URI so that it can be pushed to your private registry.

The first command tags the image for uploading to your registry and the second pushes the image.

*docker tag yourImageName yourLoginServer/yourRegistryName:yourTag*

*docker push yourLoginServer/yourRegistryName:yourTag*

docker tag eydemowebapp eycontainerregistry2019.azurecr.io/eydemowebapp

docker tag eydemowebapp **webappreg.**azurecr.io/demo

docker tag repo/mavenglobalins **globalinsreg.**azurecr.io/demo

docker tag globalins **globalinsreg.**azurecr.io/insdemo

docker tag globalins mstrainingreg.azurecr.io/trainingins

docker tag cosmomusic **musicreg.**azurecr.io/cosmov1

docker tag receiverimg **musicreg.**azurecr.io/receiverv1

docker tag datamaticswebapp datamaticsregistry.azurecr.io/v1

docker push eycontainerregistry2019.azurecr.io/ eydemowebapp

docker push **webappreg.**azurecr.io/demo

docker push **globalinsreg.**azurecr.io/demo

docker push **globalinsreg.**azurecr.io/insdemo

docker push **mstrainingreg.**azurecr.io/ trainingins

docker push **musicreg.**azurecr.io/cosmov1

docker push **musicreg.**azurecr.io/ receiverv1

docker push datamaticsregistry.azurecr.io/v1

* 1. **docker tag hello-world rpscontainer.azurecr.io/hello-world**
  2. **docker push rpscontainer.azurecr.io/hello-world**

1. **5**

Pull from your registry

Let's try to pull the image from your registry. You should find that your image is up to date.

* 1. **docker pull rpscontainer.azurecr.io/hello-world**

1. And you're ready!

You now have a working Azure Container Registry for your applications!

1. List images
2. az acr repository list --name rpscontainer --output table

Kubernetes

1. az aks install-cli

Downloading client to "C:\Users\Balasubramaniam\.azure-kubectl\kubectl.exe" from "https://storage.googleapis.com/kubernetes-release/release/v1.14.1/bin/windows/amd64/kubectl.exe"

Please add "C:\Users\Balasubramaniam\.azure-kubectl" to your search PATH so the `kubectl.exe` can be found. 2 options:

1. Run "set PATH=%PATH%;C:\Users\Balasubramaniam\.azure-kubectl" or "$env:path += 'C:\Users\Balasubramaniam\.azure-kubectl'" for PowerShell. This is good for the current command session.

2. Update system PATH environment variable by following "Control Panel->System->Advanced->Environment Variables", and re-open the command window. You only need to do it once

2.az login

Resource group and cluster name

3. az aks get-credentials -g healthcaregroup -n **healthcarecluster**

az aks get-credentials –g rg1 –n musiccluster

az aks get-credentials –g rpstraining –n insurancecluster

4.

kubectl create secret docker-registry SECRET\_NAME --docker-server=REGISTRY\_NAME.azurecr.io --docker-username=USERNAME --docker-password=PASSWORD --docker-email=ANY\_VALID\_EMAIL

kubectl create secret docker-registry eycontainerregistry2019.azurecr.io --docker-username= eycontainerregistry2019 --docker-password= Z2m1pqVuf1p1Y0iAEWiAVjoJ/CDckq9C [--docker-email=Parameswaribala@gmail.com](mailto:--docker-email=Parameswaribala@gmail.com)

kubectl create secret docker-registry acr-auth --docker-server <acr-login-server> --docker-username <service-principal-ID> --docker-password <service-principal-password> --docker-email <email-address>

kubectl create secret docker-registry acr-auth --docker-server eycontainerregistry2019.azurecr.io --docker-username= eycontainerregistry2019 --docker-password= Z2m1pqVuf1p1Y0iAEWiAVjoJ/CDckq9C [--docker-email=Parameswaribala@gmail.com](mailto:--docker-email=Parameswaribala@gmail.com)

kubectl create secret docker-registry rps-secret --docker-server webappreg.azurecr.io --docker-username= webappreg --docker-password= EGzKNZXIFd0mK+zckwhIU3bixkOWdGoy [--docker-email=Parameswaribala@gmail.com](mailto:--docker-email=Parameswaribala@gmail.com)

kubectl create secret docker-registry rps-key --docker-server rpscontainer.azurecr.io --docker-username=rpscontainer --docker-password= 0dWveMibuI=MVXZMKPggeKefqjsFmXJF [--docker-email=Parameswaribala@gmail.com](mailto:--docker-email=Parameswaribala@gmail.com)

kubectl create secret docker-registry music-key --docker-server musicreg.azurecr.io --docker-username=musicreg --docker-password= iSI=xDQa/fZHdH0iI4iKOrM2YZ6oiybQ [--docker-email=Parameswaribala@gmail.com](mailto:--docker-email=Parameswaribala@gmail.com)

kubectl create secret docker-registry receiver-key --docker-server musicreg.azurecr.io --docker-username=musicreg --docker-password= iSI=xDQa/fZHdH0iI4iKOrM2YZ6oiybQ [--docker-email=Parameswaribala@gmail.com](mailto:--docker-email=Parameswaribala@gmail.com)

kubectl create secret docker-registry globalins-key --docker-server globalinsreg.azurecr.io --docker-username= globalinsreg --docker-password= 0fiJhbK+KO12u1ji8rYGDOwOnXMAdkU4 [--docker-email=eswaribala1970@outlook.com](mailto:--docker-email=eswaribala1970@outlook.com)

kubectl create secret docker-registry globalins-key --docker-server mstrainingreg.azurecr.io --docker-username= mstrainingreg --docker-password= =pED7FK5ZA/rdjJFsCfraJfDQjHhmLXv [--docker-email=eswaribala1970@outlook.com](mailto:--docker-email=eswaribala1970@outlook.com)

F:\dotnetfidelity2019>kubectl create -f globalinsdeployment.yml

kubectl run testcontainer --image=globalins.azurecr.io/demo:latest

Kubectl get pods

Kubectl describe pods

Kubectl describe services

kubectl.exe get service/ey-api –w

Auto scaling

kubectl autoscale deployment ey-api --min=3 --max=3 --cpu-percent=80

## Start the Kubernetes dashboard

## az aks browse --resource-group myResourceGroup --name myAKSCluster

## az aks browse --resource-group healthcaregroup --name healthcarecluster