Tutorial: Deploy and use Azure Container Registry

Article • 12/20/2021 • 5 minutes to read • 21 contributors

In this article

Azure Container Registry (ACR) is a private registry for container images. A private container registry lets you securely build and deploy your applications and custom code. In this tutorial, part two of seven, you deploy an ACR instance and push a container image to it. You learn how to:

- ✓ Create an Azure Container Registry (ACR) instance
- ✓ Tag a container image for ACR
- ✓ Upload the image to ACR
- ✓ View images in your registry

In later tutorials, this ACR instance is integrated with a Kubernetes cluster in AKS, and an application is deployed from the image.

Before you begin

In the previous tutorial, a container image was created for a simple Azure Voting application. If you have not created the Azure Voting app image, return to Tutorial 1 – Create container images.

Azure CLI | Azure PowerShell

This tutorial requires that you're running the Azure CLI version 2.0.53 or later. Run `az --version` to find the version. If you need to install or upgrade, see Install Azure CLI.

Create an Azure Container Registry

To create an Azure Container Registry, you first need a resource group. An Azure resource group is a logical container into which Azure resources are deployed and managed.

Azure CLI | Azure PowerShell

Create a resource group with the az group create command. In the following example, a resource group named *myResourceGroup* is created in the *eastus* region:

Azure CLI Copy

```
az group create --name myResourceGroup --location eastus
```

Create an Azure Container Registry instance with the az acr create command and provide your own registry name. The registry name must be unique within Azure, and contain 5-50 alphanumeric characters. In the rest of this tutorial, `<acrName>` is used as a placeholder for the container registry name. Provide your own unique registry name. The *Basic* SKU is a cost-optimized entry point for development purposes that provides a balance of storage and throughput.

Azure CLI Copy

```
az acr create --resource-group myResourceGroup --name <acrName> --sku
Basic
```

Log in to the container registry

Azure CLI | Azure PowerShell

To use the ACR instance, you must first log in. Use the az acr login command and provide the unique name given to the container registry in the previous step.

Azure CLI Copy

```
az acr login --name <acrName>
```

The command returns a *Login Succeeded* message once completed.

Tag a container image

To see a list of your current local images, use the docker images command:

Console Copy

```
docker images
```

The above command's output shows list of your current local images:

Output Copy

```
REPOSITORY                                    TAG                 IMAGE ID
CREATED             SIZE
mcr.microsoft.com/azuredocs/azure-vote-front   v1
84b41c268ad9        7 minutes ago       944MB
mcr.microsoft.com/oss/bitnami/redis           6.0.8
3a54a920bb6c        2 days ago          103MB
tiangolo/uwsgi-nginx-flask                    python3.6
a16ce562e863        6 weeks ago         944MB
```

To use the *azure-vote-front* container image with ACR, the image needs to be tagged with the login server address of your registry. This tag is used for routing when pushing container images to an image registry.

Azure CLI | Azure PowerShell

To get the login server address, use the az acr list command and query for the *loginServer* as follows:

Azure CLI Copy

```
az acr list --resource-group myResourceGroup --query "[].
{acrLoginServer:loginServer}" --output table
```

Now, tag your local *azure-vote-front* image with the *acrLoginServer* address of the container registry. To indicate the image version, add *:v1* to the end of the image name:

Console Copy

```
docker tag mcr.microsoft.com/azuredocs/azure-vote-front:v1
<acrLoginServer>/azure-vote-front:v1
```

To verify the tags are applied, run docker images again.

Console Copy

```
docker images
```

An image is tagged with the ACR instance address and a version number.

Copy

```
REPOSITORY                                          TAG                 IMAGE ID
CREATED             SIZE
mcr.microsoft.com/azuredocs/azure-vote-front    v1
84b41c268ad9        16 minutes ago      944MB
mycontainerregistry.azurecr.io/azure-vote-front v1
84b41c268ad9        16 minutes ago      944MB
mcr.microsoft.com/oss/bitnami/redis             6.0.8
3a54a920bb6c        2 days ago          103MB
tiangolo/uwsgi-nginx-flask                      python3.6
a16ce562e863        6 weeks ago         944MB
```

Push images to registry

With your image built and tagged, push the *azure-vote-front* image to your ACR instance. Use docker push and provide your own *acrLoginServer* address for the image name as follows:

Console Copy

```
docker push <acrLoginServer>/azure-vote-front:v1
```

It may take a few minutes to complete the image push to ACR.

List images in registry

Azure CLI | Azure PowerShell

To return a list of images that have been pushed to your ACR instance, use the az acr repository list command. Provide your own `<acrName>` as follows:

Azure CLI | Copy

```
az acr repository list --name <acrName> --output table
```

The following example output lists the *azure-vote-front* image as available in the registry:

Output | Copy

```
Result
----------------
azure-vote-front
```

To see the tags for a specific image, use the az acr repository show-tags command as follows:

Azure CLI | Copy

```
az acr repository show-tags --name <acrName> --repository azure-vote-
front --output table
```

The following example output shows the *v1* image tagged in a previous step:

Output | Copy

```
Result
--------
v1
```

You now have a container image that is stored in a private Azure Container Registry instance. This image is deployed from ACR to a Kubernetes cluster in the next tutorial.

Next steps

In this tutorial, you created an Azure Container Registry and pushed an image for use in an AKS cluster. You learned how to:

- ✓ Create an Azure Container Registry (ACR) instance
- ✓ Tag a container image for ACR

- ✓ Upload the image to ACR
- ✓ View images in your registry

Advance to the next tutorial to learn how to deploy a Kubernetes cluster in Azure.

Deploy Kubernetes cluster

Recommended content

Kubernetes on Azure tutorial - Update an application - Azure Kubernetes Service

In this Azure Kubernetes Service (AKS) tutorial, you learn how to update an existing application deployment to AKS with a new version of the application code.

Kubernetes on Azure tutorial - Deploy an application - Azure Kubernetes Service

In this Azure Kubernetes Service (AKS) tutorial, you deploy a multi-container application to your cluster using a custom image stored in Azure Container Registry.

Kubernetes on Azure tutorial - Upgrade a cluster - Azure Kubernetes Service

In this Azure Kubernetes Service (AKS) tutorial, you learn how to upgrade an existing AKS cluster to the latest available Kubernetes version.

Kubernetes on Azure tutorial - Prepare an application - Azure Kubernetes Service

In this Azure Kubernetes Service (AKS) tutorial, you learn how to prepare and build a multi-container app with Docker Compose that you can then deploy to AKS.

Kubernetes on Azure tutorial - Deploy a cluster - Azure Kubernetes Service

In this Azure Kubernetes Service (AKS) tutorial, you create an AKS cluster and use kubectl to connect to the Kubernetes master node.

Build, test, and deploy containers to Azure Kubernetes Service using GitHub Actions - Azure Kubernetes Service

Learn how to use GitHub Actions to deploy your container to Kubernetes

Develop on Azure Kubernetes Service (AKS) with Helm - Azure Kubernetes Service

Use Helm with AKS and Azure Container Registry to package and run application containers in a cluster.

Kubernetes on Azure tutorial - Scale Application - Azure Kubernetes Service

In this Azure Kubernetes Service (AKS) tutorial, you learn how to scale nodes and pods in Kubernetes, and implement horizontal pod autoscaling.

Show more