Tutorial: Run applications in Azure Kubernetes Service (AKS)

Article • 01/04/2022 • 4 minutes to read • 20 contributors

In this article

Kubernetes provides a distributed platform for containerized applications. You build and deploy your own applications and services into a Kubernetes cluster, and let the cluster manage the availability and connectivity. In this tutorial, part four of seven, a sample application is deployed into a Kubernetes cluster. You learn how to:

- ✓ Update a Kubernetes manifest file
- ✓ Run an application in Kubernetes
- ✓ Test the application

In later tutorials, this application is scaled out and updated.

This quickstart assumes a basic understanding of Kubernetes concepts. For more information, see Kubernetes core concepts for Azure Kubernetes Service (AKS).

Tip

AKS clusters can use GitOps for configuration management. This enables declarations of your cluster's state, which are pushed to source control, to be applied to the cluster automatically. To learn how to use GitOps to deploy an application with an AKS cluster, see the tutorial **Use GitOps with Flux v2** and follow the **prerequisites for Azure Kubernetes Service clusters**.

Before you begin

In previous tutorials, an application was packaged into a container image, this image was uploaded to Azure Container Registry, and a Kubernetes cluster was created.

To complete this tutorial, you need the pre-created `azure-vote-all-in-one-redis.yaml` Kubernetes manifest file. This file was downloaded with the application source code in a previous tutorial. Verify that you've cloned the repo, and that you have changed directories into the cloned repo. If you haven't done these steps, and would like to follow along, start with 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.

Update the manifest file

In these tutorials, an Azure Container Registry (ACR) instance stores the container image for the sample application. To deploy the application, you must update the image name in the Kubernetes manifest file to include the ACR login server name.

Azure CLI | Azure PowerShell

Get the ACR login server name using the az acr list command as follows:

Azure CLI

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

The sample manifest file from the git repo cloned in the first tutorial uses the images from Microsoft Container Registry (*mcr.microsoft.com*). Make sure that you're in the cloned *azure-voting-app-redis* directory, then open the manifest file with a text editor, such as `vi`:

Console

```
vi azure-vote-all-in-one-redis.yaml
```

Replace *mcr.microsoft.com* with your ACR login server name. The image name is found on line 60 of the manifest file. The following example shows the default image name:

YAML Copy

```yaml
containers:
- name: azure-vote-front
  image: mcr.microsoft.com/azuredocs/azure-vote-front:v1
```

Provide your own ACR login server name so that your manifest file looks like the following example:

YAML Copy

```yaml
containers:
- name: azure-vote-front
  image: <acrName>.azurecr.io/azure-vote-front:v1
```

Save and close the file. In `vi`, use `:wq`.

Deploy the application

To deploy your application, use the kubectl apply command. This command parses the manifest file and creates the defined Kubernetes objects. Specify the sample manifest file, as shown in the following example:

Console Copy

```console
kubectl apply -f azure-vote-all-in-one-redis.yaml
```

The following example output shows the resources successfully created in the AKS cluster:

Console Copy

```console
$ kubectl apply -f azure-vote-all-in-one-redis.yaml

deployment "azure-vote-back" created
service "azure-vote-back" created
deployment "azure-vote-front" created
service "azure-vote-front" created
```

Test the application

When the application runs, a Kubernetes service exposes the application front end to the internet. This process can take a few minutes to complete.

To monitor progress, use the kubectl get service command with the `--watch` argument.

Console Copy

```
kubectl get service azure-vote-front --watch
```

Initially the *EXTERNAL-IP* for the *azure-vote-front* service is shown as *pending*:

Output Copy

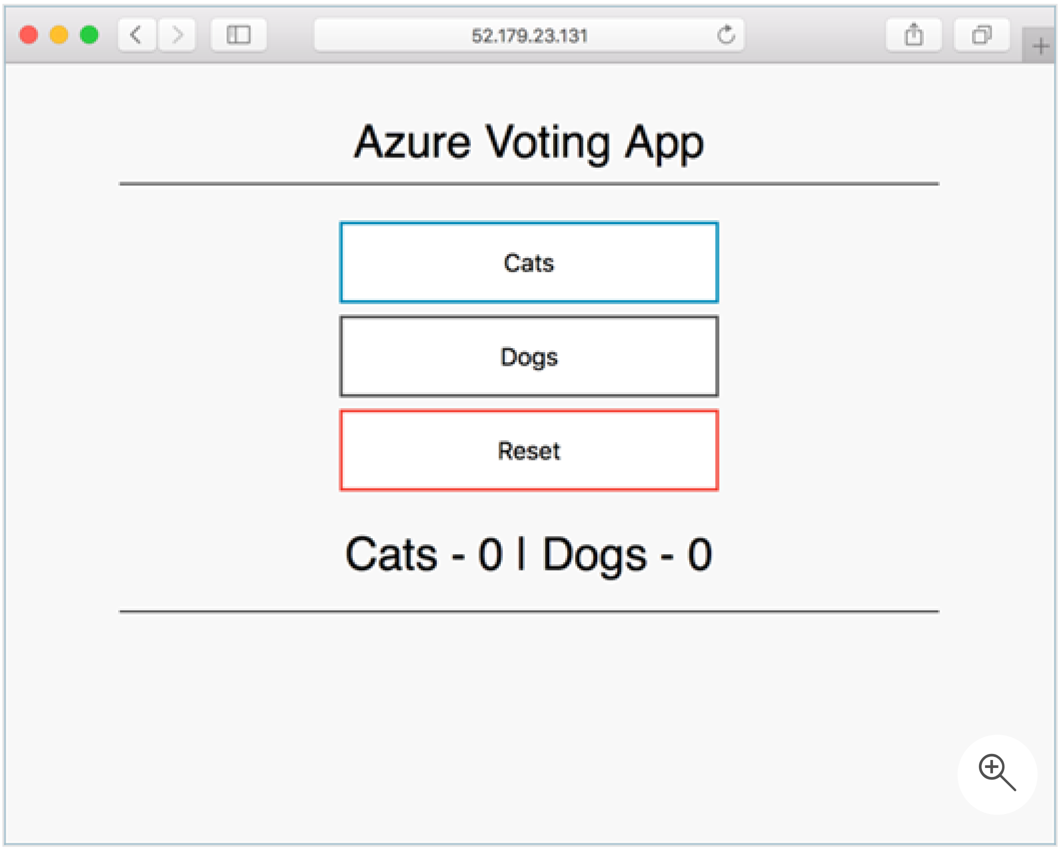
```
azure-vote-front   LoadBalancer   10.0.34.242   <pending>     80:30676/TCP
5s
```

When the *EXTERNAL-IP* address changes from *pending* to an actual public IP address, use `CTRL-C` to stop the `kubectl` watch process. The following example output shows a valid public IP address assigned to the service:

Output Copy

```
azure-vote-front   LoadBalancer   10.0.34.242   52.179.23.131   80:30676/TCP
67s
```

To see the application in action, open a web browser to the external IP address of your service:



If the application didn't load, it might be due to an authorization problem with your image registry. To view the status of your containers, use the `kubectl get pods` command. If the container images can't be pulled, see Authenticate with Azure Container Registry from Azure Kubernetes Service.

Next steps

In this tutorial, a sample Azure vote application was deployed to a Kubernetes cluster in AKS. You learned how to:

- ✓ Update a Kubernetes manifest files
- ✓ Run an application in Kubernetes
- ✓ Test the application

Advance to the next tutorial to learn how to scale a Kubernetes application and the underlying Kubernetes infrastructure.

Scale Kubernetes application and infrastructure

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 - 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 - Create a container registry - Azure Kubernetes Service

In this Azure Kubernetes Service (AKS) tutorial, you create an Azure Container Registry instance and upload a sample application container image.

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

Show more