Quickstart: Deploy your first container app with containerapp up

Article • 08/04/2024

The Azure Container Apps service enables you to run microservices and containerized applications on a serverless platform. With Container Apps, you enjoy the benefits of running containers while you leave behind the concerns of manually configuring cloud infrastructure and complex container orchestrators.

In this quickstart, you create and deploy your first container app using the `az containerapp up` command.

Prerequisites

- An Azure account with an active subscription.
  - If you don't have one, you can create one for free .
- Install the Azure CLI.

Setup

To sign in to Azure from the CLI, run the following command and follow the prompts to complete the authentication process.

Bash

Azure CLI

```
az login
```

To ensure you're running the latest version of the CLI, run the upgrade command.

Bash

Azure CLI

```
az upgrade
```

Next, install or update the Azure Container Apps extension for the CLI.

If you receive errors about missing parameters when you run `az containerapp` commands in Azure CLI or cmdlets from the `Az.App` module in Azure PowerShell, be sure you have the latest version of the Azure Container Apps extension installed.

Bash

Azure CLI

```
az extension add --name containerapp --upgrade
```

Note

Starting in May 2024, Azure CLI extensions no longer enable preview features by default. To access Container Apps preview features, install the Container Apps extension with `--allow-preview true`.

Azure CLI

```
az extension add --name containerapp --upgrade --allow-preview true
```

Now that the current extension or module is installed, register the `Microsoft.App` and `Microsoft.OperationalInsights` namespaces.

Bash

Azure CLI

```azurecli
az provider register --namespace Microsoft.App
```

Azure CLI

```azurecli
az provider register --namespace Microsoft.OperationalInsights
```

Create an Azure resource group

Create a resource group to organize the services related to your container app deployment.

Bash

Azure CLI

```azurecli
az group create \
  --name my-container-apps \
  --location centralus
```

Create and deploy the container app

Create and deploy your first container app with the `containerapp up` command. This command will:

- Create the Container Apps environment
- Create the Log Analytics workspace
- Create and deploy the container app using a public container image

Note that if any of these resources already exist, the command will use them instead of creating new ones.

Bash

Azure CLI

```
az containerapp up \
  --name my-container-app \
  --resource-group my-container-apps \
  --location centralus \
  --environment 'my-container-apps' \
  --image mcr.microsoft.com/k8se/quickstart:latest \
  --target-port 80 \
  --ingress external \
  --query properties.configuration.ingress.fqdn
```

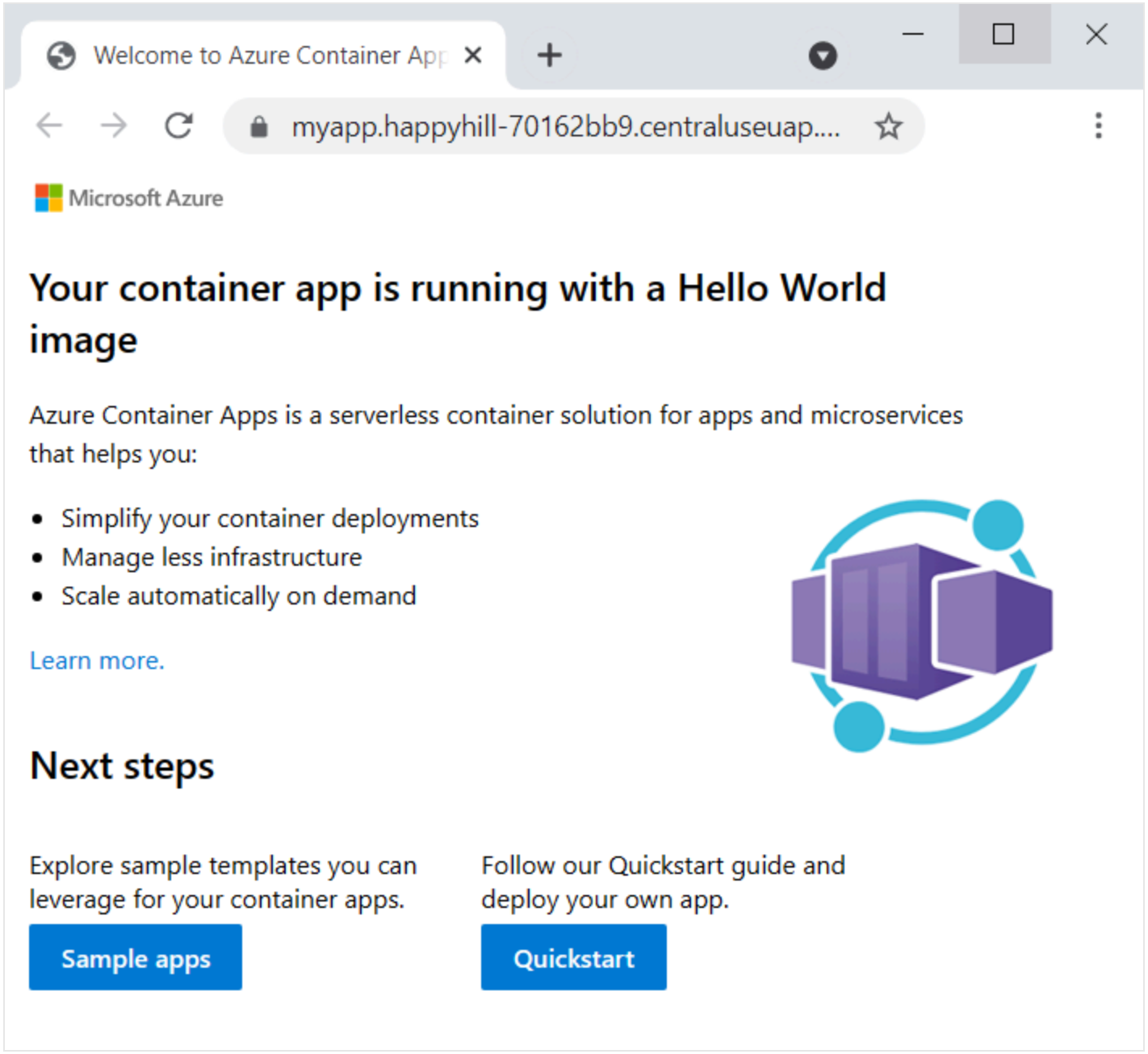
> **Note**
>
> Make sure the value for the `--image` parameter is in lower case.

By setting `--ingress` to `external`, you make the container app available to public requests.

Verify deployment

The `up` command returns the fully qualified domain name for the container app. Copy this location to a web browser.

The following message is displayed when the container app is deployed:



Clean up resources

If you're not going to continue to use this application, run the following command to delete the resource group along with all the resources created in this quickstart.

> **Caution**
>
> The following command deletes the specified resource group and all resources contained within it. If resources outside the scope of this quickstart exist in the specified resource group, they will also be deleted.

Azure CLI

```azurecli
az group delete --name my-container-apps
```

> **Tip**
>
> Having issues? Let us know on GitHub by opening an issue in the Azure Container Apps repo .

Next steps

Communication between microservices

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A