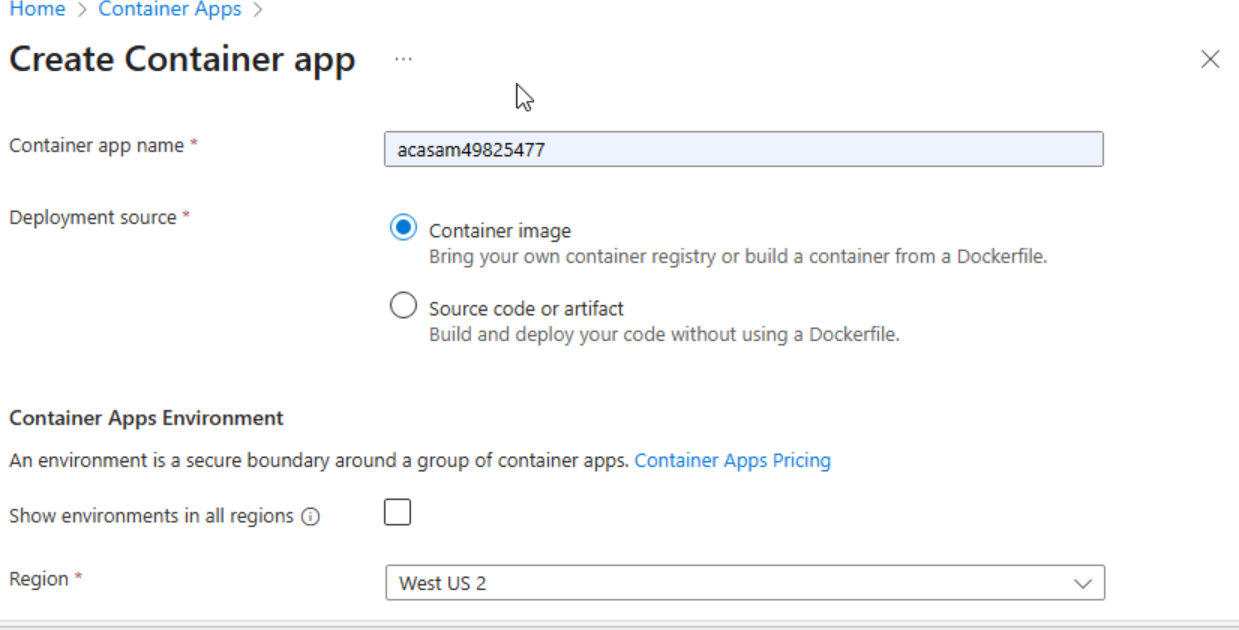


az acr build --registry acrsam49825477 --image imagename:latest .

<a href="http://20.99.147.67:8080/">Home page - My Gallery</a>

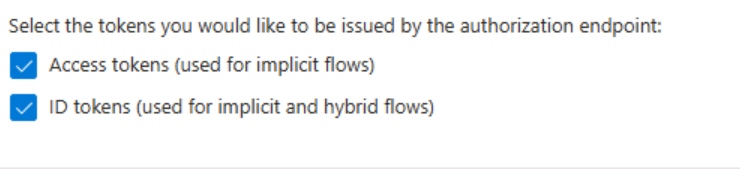
<a href="http://dns-name.fkewb4cjdtetf8gf.westus2.azurecontainer.io:8080/">Home page - My Gallery (fkewb4cjdtetf8gf.westus2.azurecontainer.io)</a>



**az container restart –name --resource-group [--no-wait]**

**az container restart -n acisam49825477 -g acr-sam-rg [--no-wait]**

**acr-sam-rg**

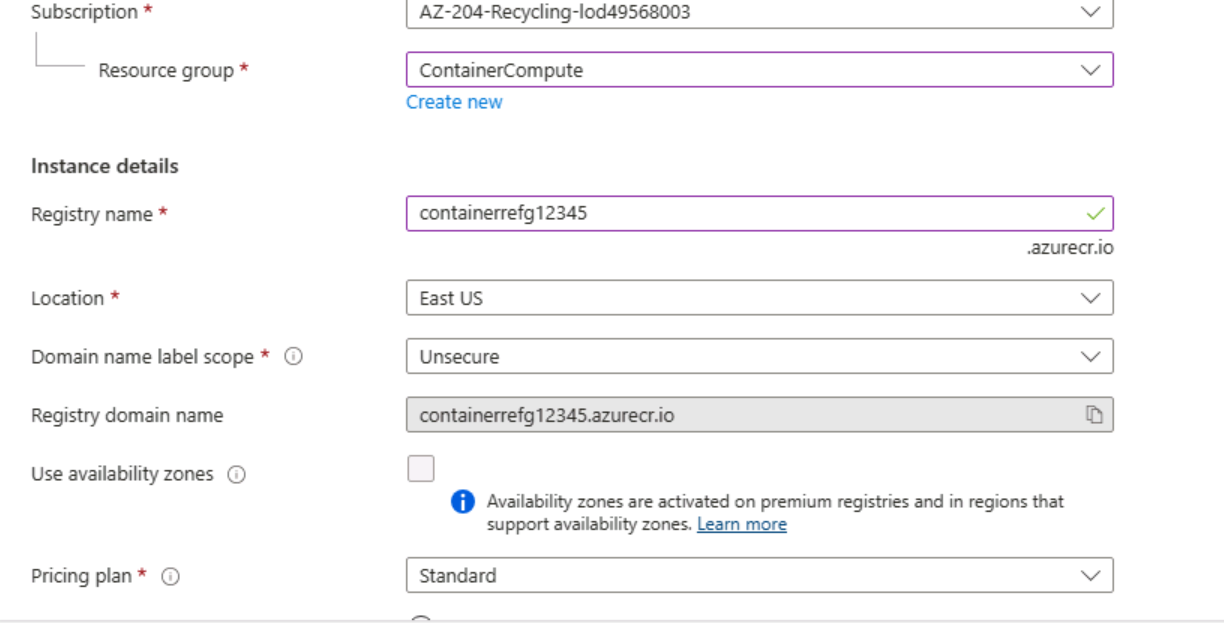
****

**dotnet new console --output . --name ipcheck --framework net8.0**

**New-Item -ItemType File Dockerfile**

**code .**

**Compress-Archive -Path .\\* -DestinationPath .\lab05.zip**

****

**mkdir ~/ipcheck**

**unzip ~/lab05.zip -d ~/ipcheck**

**chmod -R +xr ~/ipcheck**

**cd ~/ipcheck**

**registryName=conregistry$RANDOM**

**az acr check-name --name $registryName**

**az acr create --resource-group ContainerCompute --name $registryName --sku Basic**

**{**

**"adminUserEnabled": false,**

**"anonymousPullEnabled": false,**

**"creationDate": "2025-06-06T08:06:55.762784+00:00",**

**"dataEndpointEnabled": false,**

**"dataEndpointHostNames": [],**

**"encryption": {**

**"keyVaultProperties": null,**

**"status": "disabled"**

**},**

**"id": "/subscriptions/3843a206-dd7e-41fb-81a8-5d6533c0cd6e/resourceGroups/ContainerCompute/providers/Microsoft.ContainerRegistry/registries/conregistry7383",**

**"identity": null,**

**"location": "eastus",**

**"loginServer": "conregistry7383.azurecr.io",**

**"metadataSearch": "Disabled",**

**"name": "conregistry7383",**

**"networkRuleBypassOptions": "AzureServices",**

**"networkRuleSet": null,**

**"policies": {**

**"azureAdAuthenticationAsArmPolicy": {**

**"status": "enabled"**

**},**

**"exportPolicy": {**

**"status": "enabled"**

**},**

**"quarantinePolicy": {**

**"status": "disabled"**

**},**

**"retentionPolicy": {**

**"days": 7,**

**"lastUpdatedTime": "2025-06-06T08:07:02.748363+00:00",**

**"status": "disabled"**

**},**

**"softDeletePolicy": {**

**"lastUpdatedTime": "2025-06-06T08:07:02.748407+00:00",**

**"retentionDays": 7,**

**"status": "disabled"**

**},**

**"trustPolicy": {**

**"status": "disabled",**

**"type": "Notary"**

**}**

**},**

**"privateEndpointConnections": [],**

**"provisioningState": "Succeeded",**

**"publicNetworkAccess": "Enabled",**

**"resourceGroup": "ContainerCompute",**

**"sku": {**

**"name": "Basic",**

**"tier": "Basic"**

**},**

**"status": null,**

**"systemData": {**

**"createdAt": "2025-06-06T08:06:55.762784+00:00",**

**"createdBy": "LabUser-51983975@LODSPRODMSLEARNMCA.onmicrosoft.com",**

**"createdByType": "User",**

**"lastModifiedAt": "2025-06-06T08:06:55.762784+00:00",**

**"lastModifiedBy": "LabUser-51983975@LODSPRODMSLEARNMCA.onmicrosoft.com",**

**"lastModifiedByType": "User"**

**},**

**"tags": {},**

**"type": "Microsoft.ContainerRegistry/registries",**

**"zoneRedundancy": "Disabled"**

**}**

**az acr list --resource-group ContainerCompute**

**az acr list --resource-group ContainerCompute**

**az acr list --resource-group ContainerCompute --query "max\_by([], &creationDate).name" --output tsv**

**acrName=$(az acr list --resource-group ContainerCompute --query "max\_by([], &creationDate).name" --output tsv)**

**acrName=$(az acr list --resource-group ContainerCompute --query "max\_by([], &creationDate).name" --output tsv)**

**Deploy a Docker container image to Container Registry**

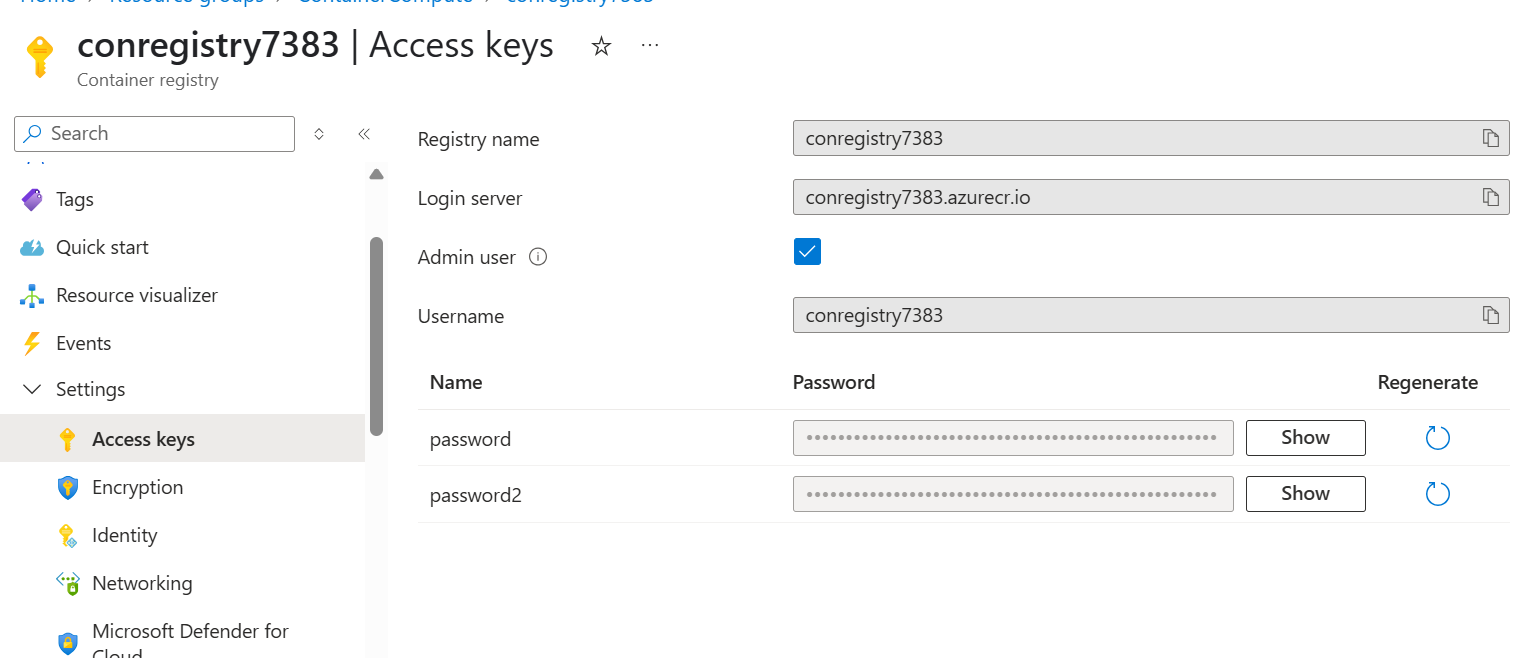
**cd ~/ipcheck**

**dir**

**az acr build --registry $acrName --image ipcheck:latest .**

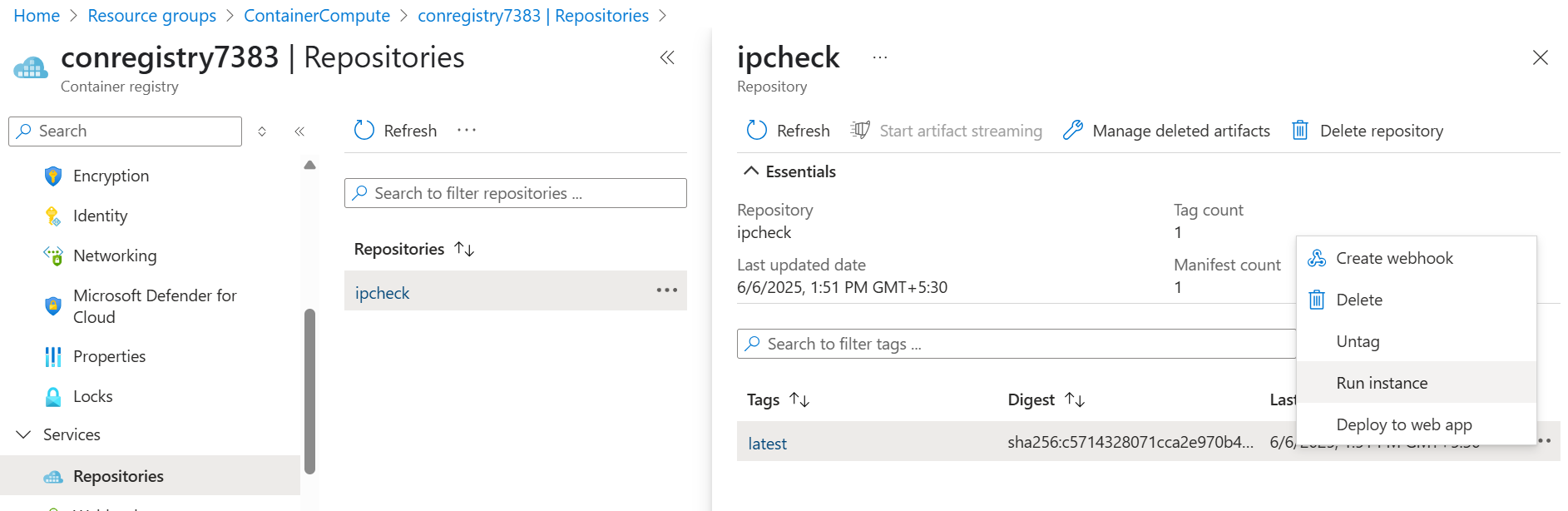
**Task 1: Enable the admin user in Container Registry**

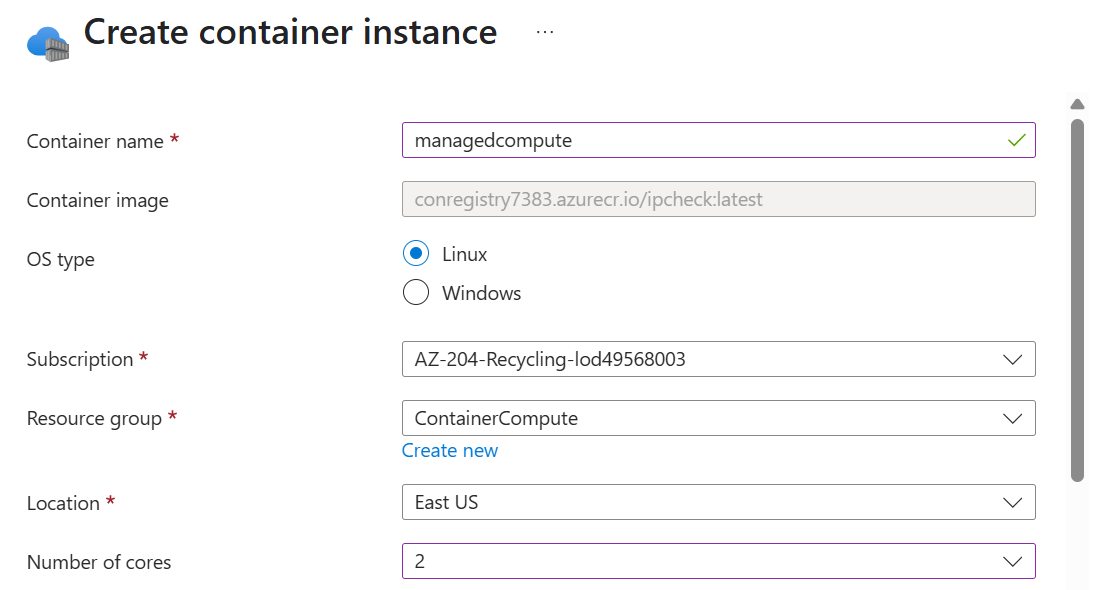
1. **In the ContainerCompute resource group, select the container registry that you created previously in this lab, and then select the Access keys blade.**
2. **Toggle the switch in the admin user section to enable the admin user for this container registry.**

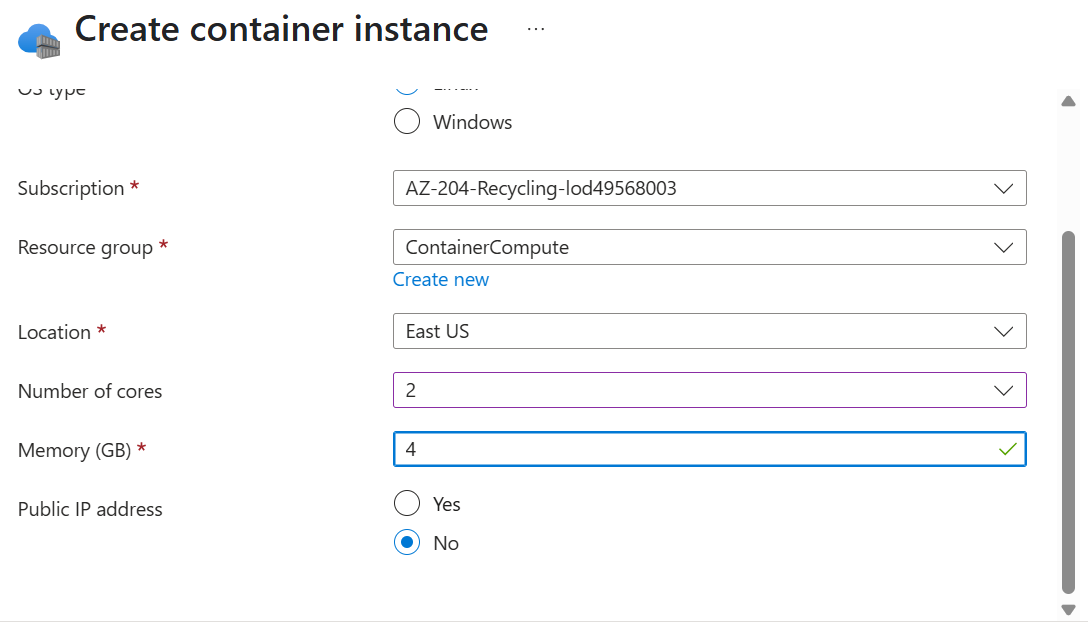
****

**Task 2: Automatically deploy a container image to an Azure container instance**

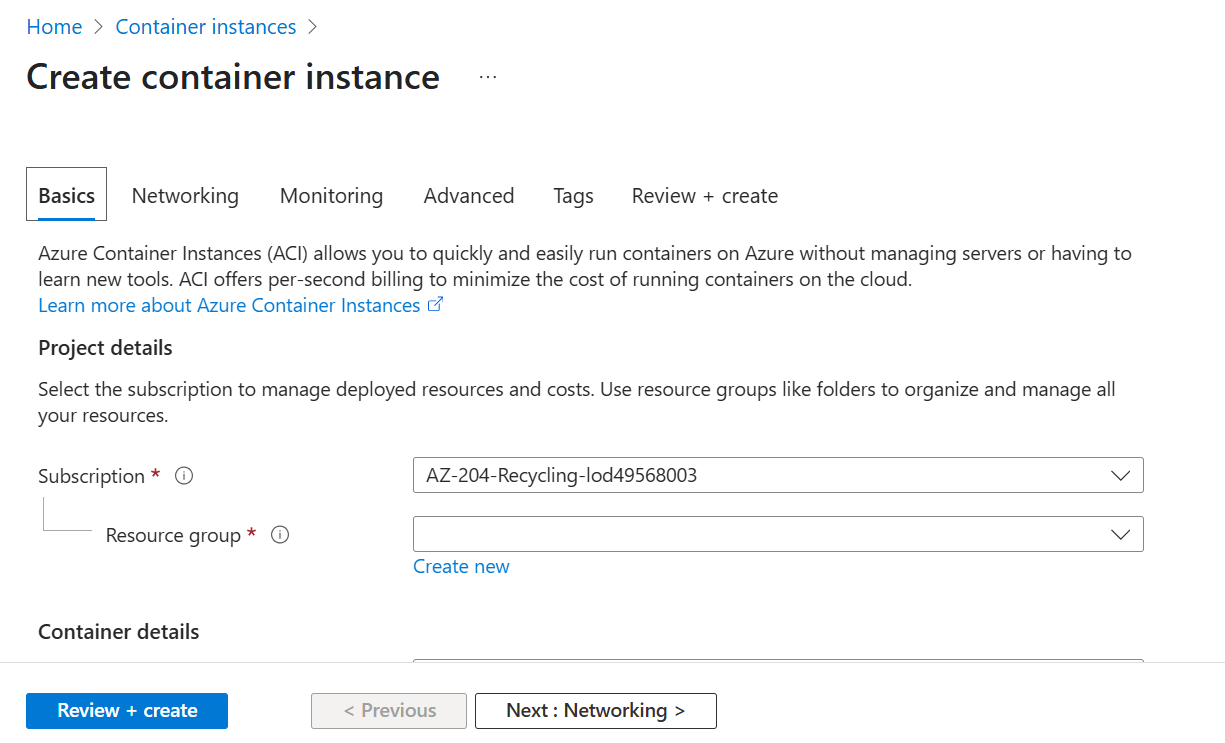
1. **On the Container Registry blade, in the Services section, select the Repositories link.**
2. **In the Repositories section, select the repository containing the ipcheck container image.**
3. **On the ipcheck pane, select the ellipsis menu associated with the latest tag entry, and then select Run instance.**

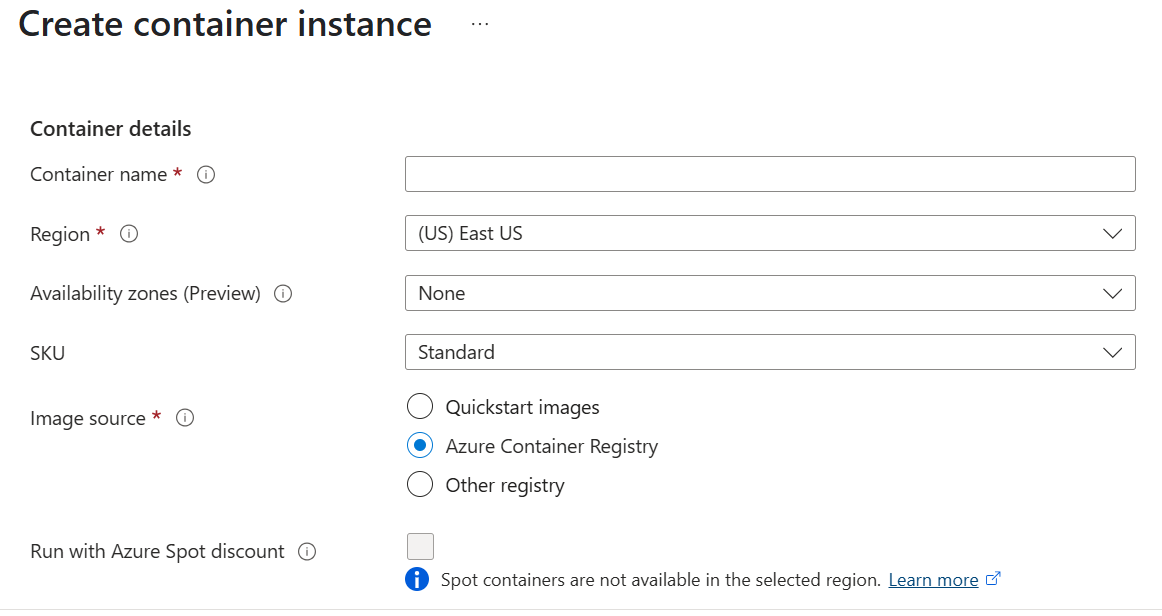
****

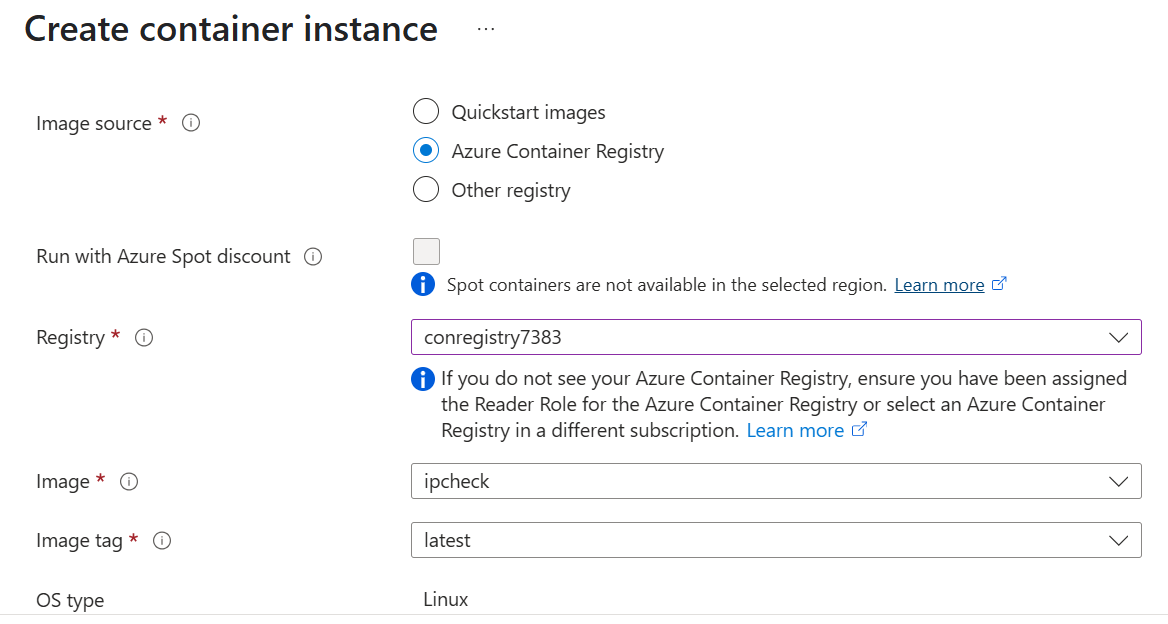
****

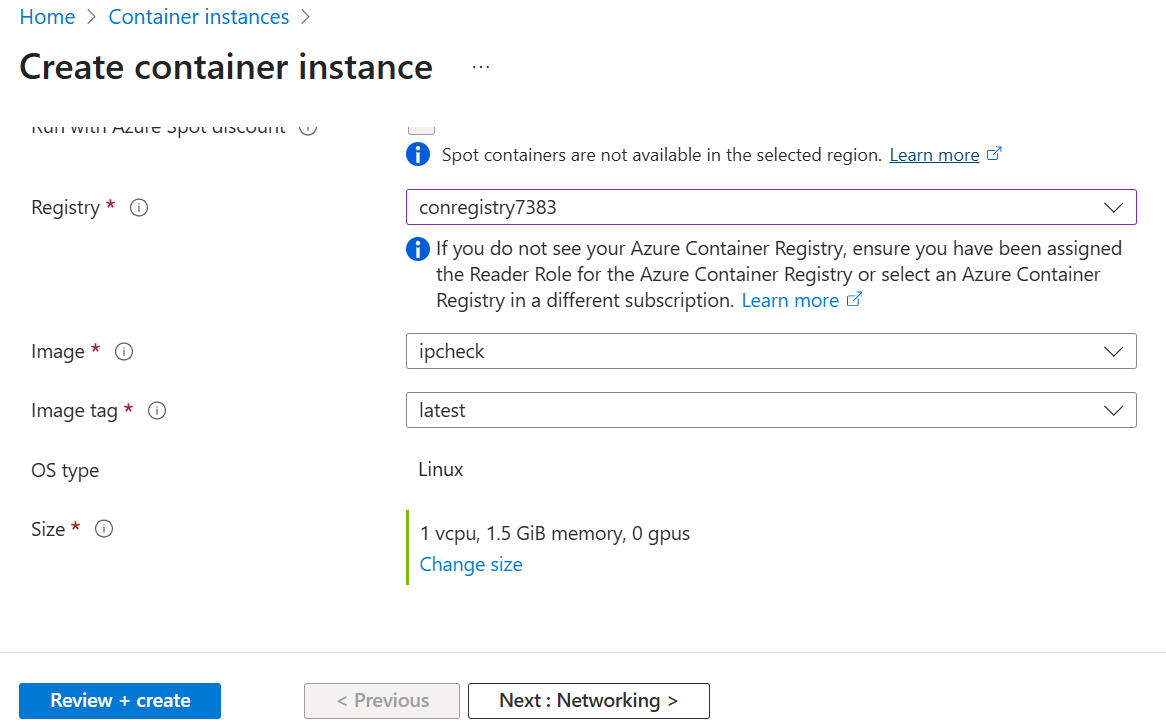
****

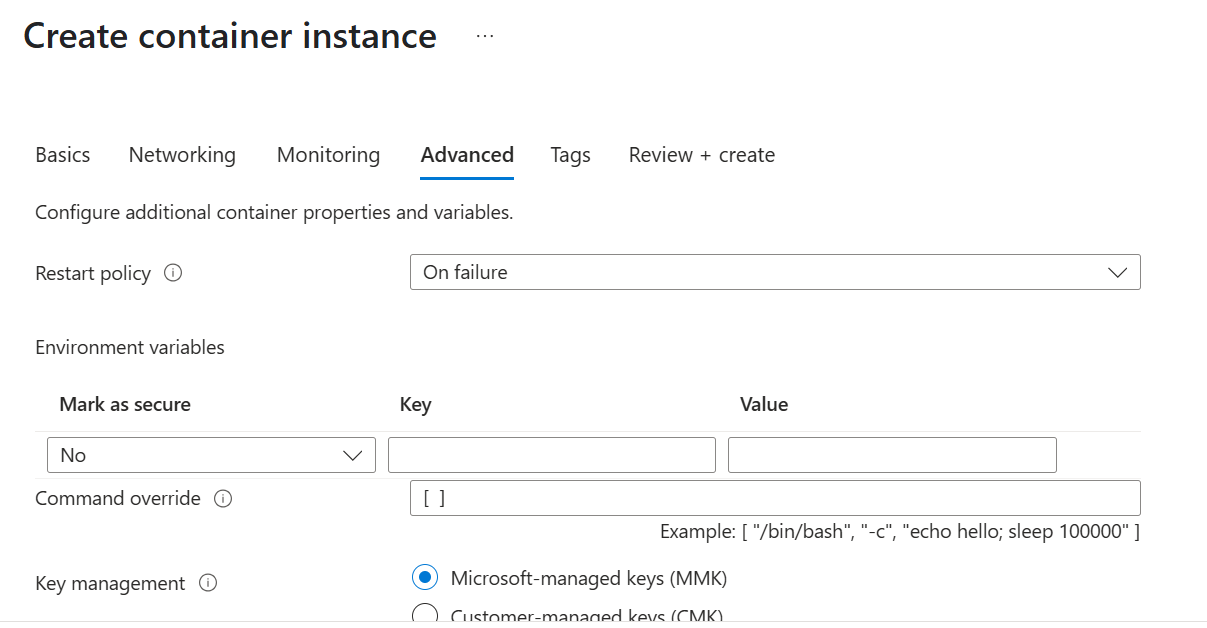
**Second way to create container instance**

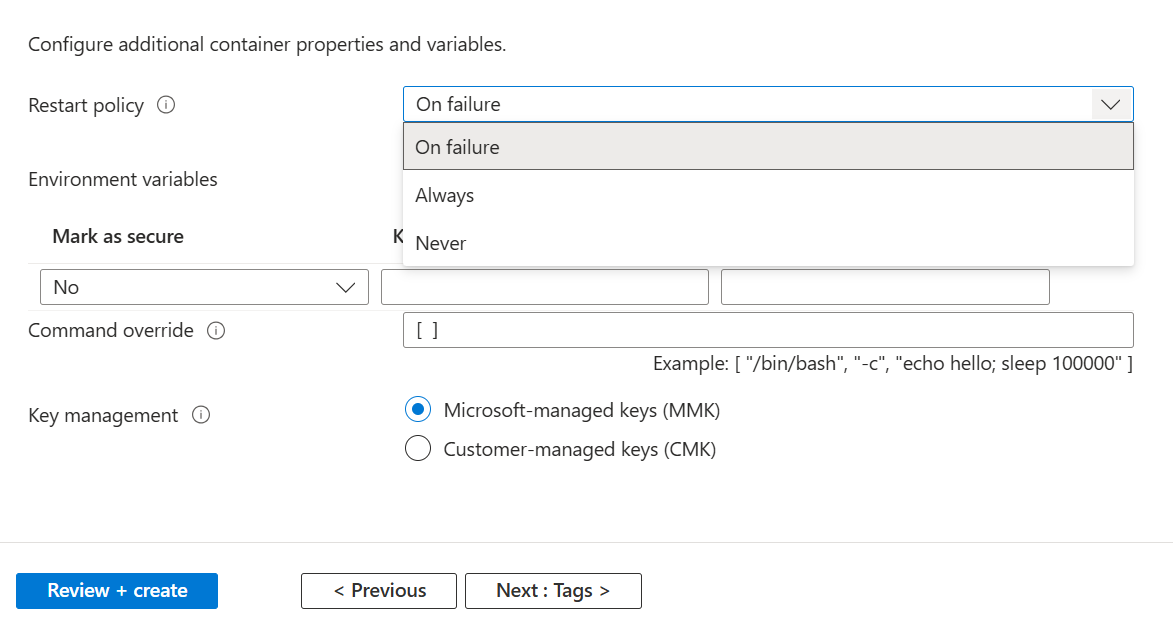
****

****

****

****

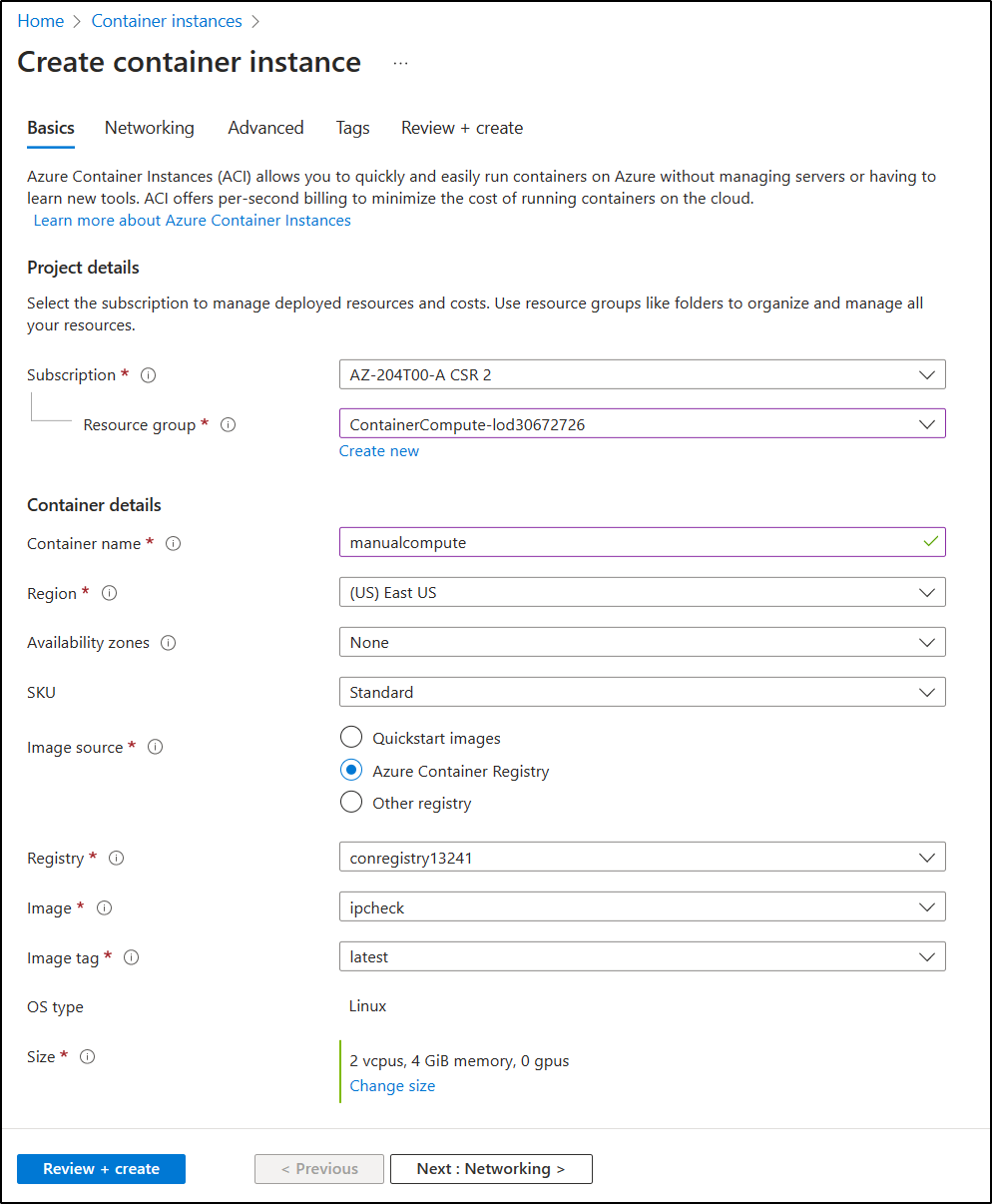
****

****

**Task 3: Manually deploy a container image to Container Instances**

1. **On the Azure portal's navigation pane, select the Create a resource link.**
2. **On the Create a resource blade, in the Search services and marketplace text box, enter container instances, and then select Enter.**
3. **On the Marketplace search results blade, select the Container Instances result.**
4. **On the Container Instances blade, select Create.**
5. **On the Create Container Instance blade, on the Basics tab, perform the following actions, and then select Review + create:**

| **Setting** | **Action** |
| --- | --- |
| **Subscription drop-down list** | **Retain the default value** |
| **Resource group drop-down list** | **Select ContainerCompute** |
| **Container name text box** | **enter manualcompute** |
| **Region drop-down list** | **Select (US) East US** |
| **Image source section** | **Select Azure Container Registry** |
| **Registry drop-down list** | **Select the Azure Container Registry resource that you created previously in this lab** |
| **Image drop-down list** | **Select ipcheck** |
| **Image tag drop-down list** | **Select latest** |
| **Size : cores** | **enter 2** |
| **Size : Memory (GiB)** | **enter 4** |

1. **The following screenshot displays the configured settings on the Create container instance blade.**
2. ****
3. **From the Review + create tab, review the selected options.**
4. **Select Create to create the container instance by using your specified configuration.**

**Note: Wait for the container instance to be created before you continue with this lab.**

**Exercise 3: Create a secure Container Apps environment and deploy container app.**

**Task 1: Prepare your environment**

1. **Sign in to the Azure portal.**
2. **Select the Cloud Shell icon, select the Bash environment.**
3. **In the Cloud Shell command prompt in the portal, run the following command to install the Azure Container Apps extension for the CLI:**

**bash**

**az extension add --name containerapp --upgrade**

1. **Run the following command to register the Microsoft.App namespace:**

**bash**

**az provider register --namespace Microsoft.App**

**Note: Azure Container Apps resources have migrated from the Microsoft.Web namespace to the Microsoft.App namespace.**

1. **Run the following command to Register the Microsoft.OperationalInsights provider for the Azure Monitor Log Analytics workspace if you haven't used it before:**

**bash**

**az provider register --namespace Microsoft.OperationalInsights**

**Note: Registering the Microsoft.App namespace and Microsoft.OperationalInsights can each take a few minutes to complete.**

1. **Set environment variables used later in this exercise.**

**bash**

**myRG=ContainerCompute**

**myAppContEnv=az204-env-51983975**

**With the CLI upgraded and environment variables created, you can create a Container Apps environment and deploy your container app.**

**Task 2: Create an environment**

**An environment in Azure Container Apps creates a secure boundary around a group of container apps. Container Apps deployed to the same environment are deployed in the same virtual network and write logs to the same Log Analytics workspace.**

1. **In the Cloud Shell command prompt in the portal, run the following command to create an environment by using the az containerapp env create command:**

**bash**

**az containerapp env create \**

**--name $myAppContEnv \**

**--resource-group $myRG \**

**--location eastus**

**Task 3: Create a container app**

**After the container app environment finishes deployment, you deploy a container image to Azure Container Apps.**

1. **In the Cloud Shell command prompt in the portal, run the following command to deploy a sample app container image by using the az containerapp create command:**

**bash**

**az containerapp create \**

**--name my-container-app \**

**--resource-group $myRG \**

**--environment $myAppContEnv \**

**--image mcr.microsoft.com/azuredocs/containerapps-helloworld:latest \**

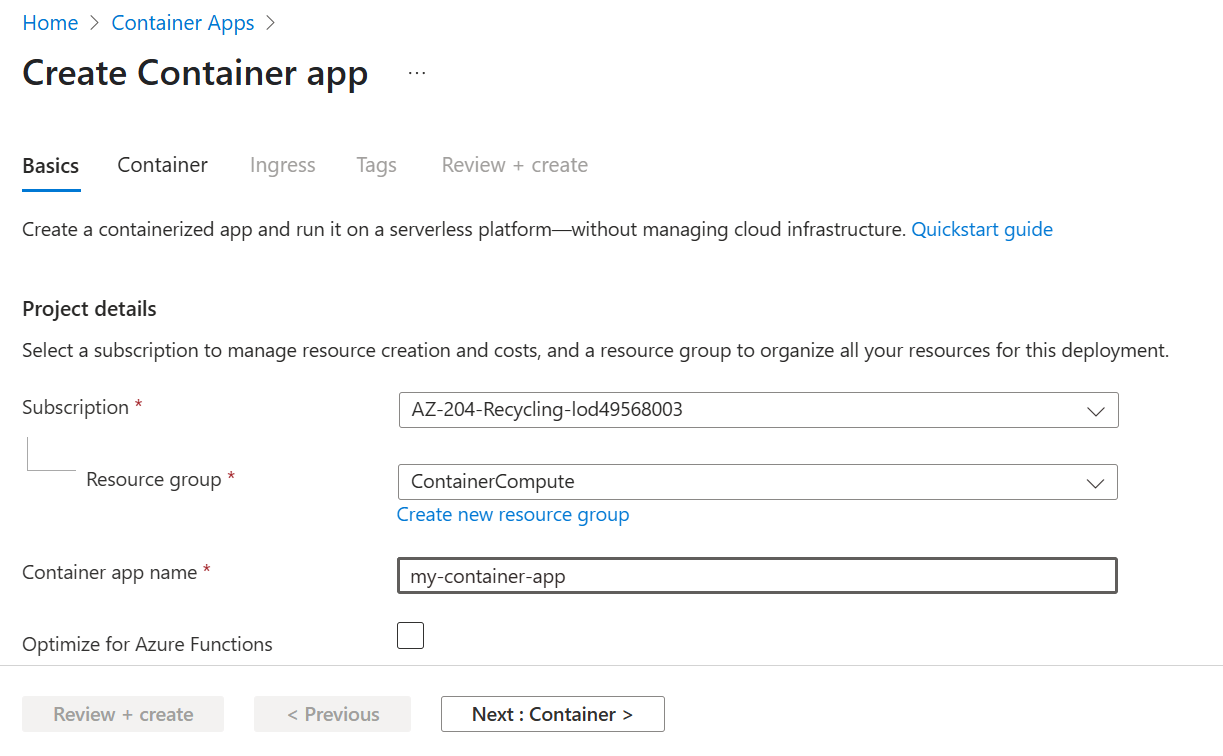
**--target-port 80 \**

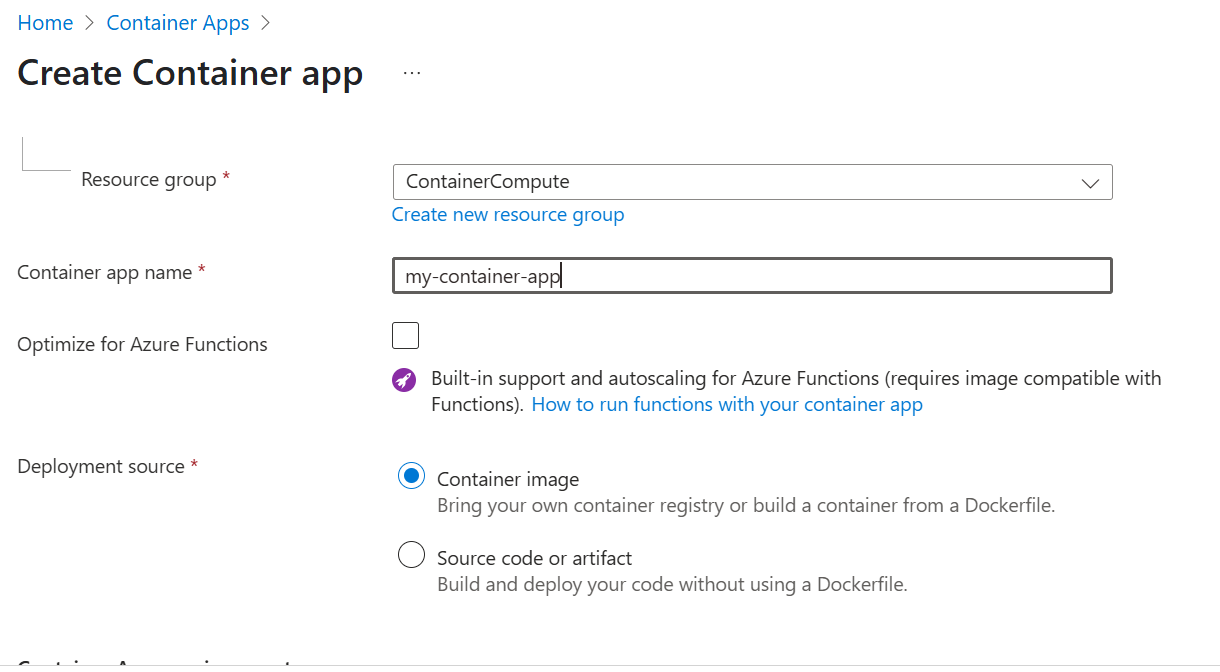
**--ingress 'external' \**

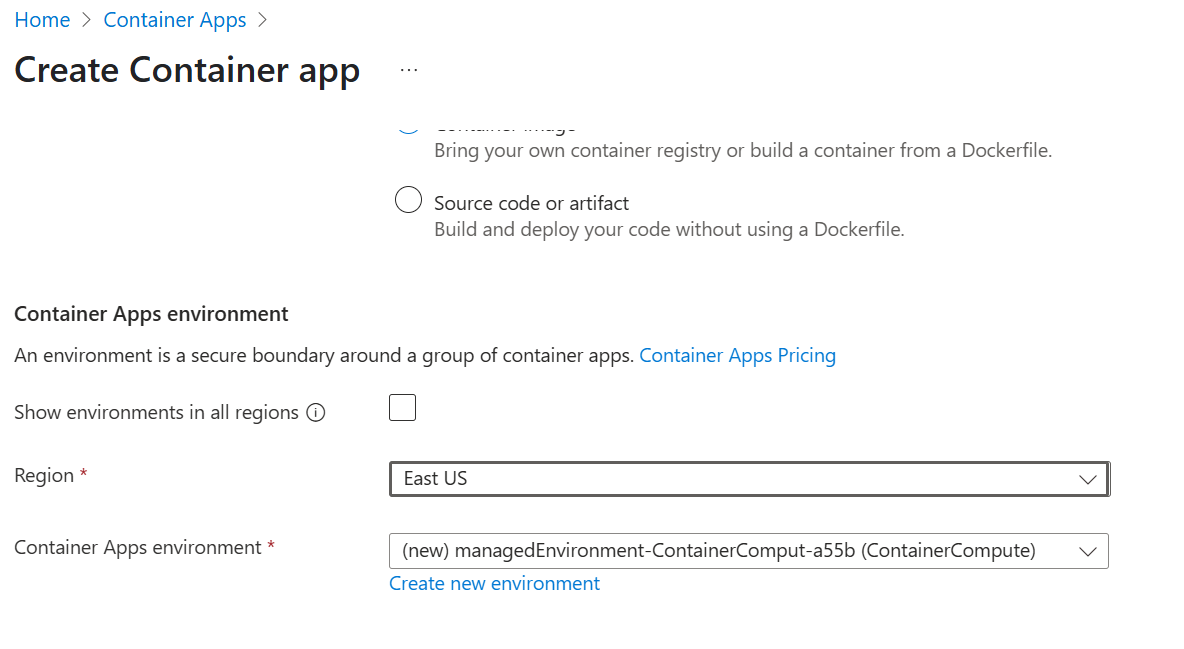
**--query properties.configuration.ingress.fqdn**

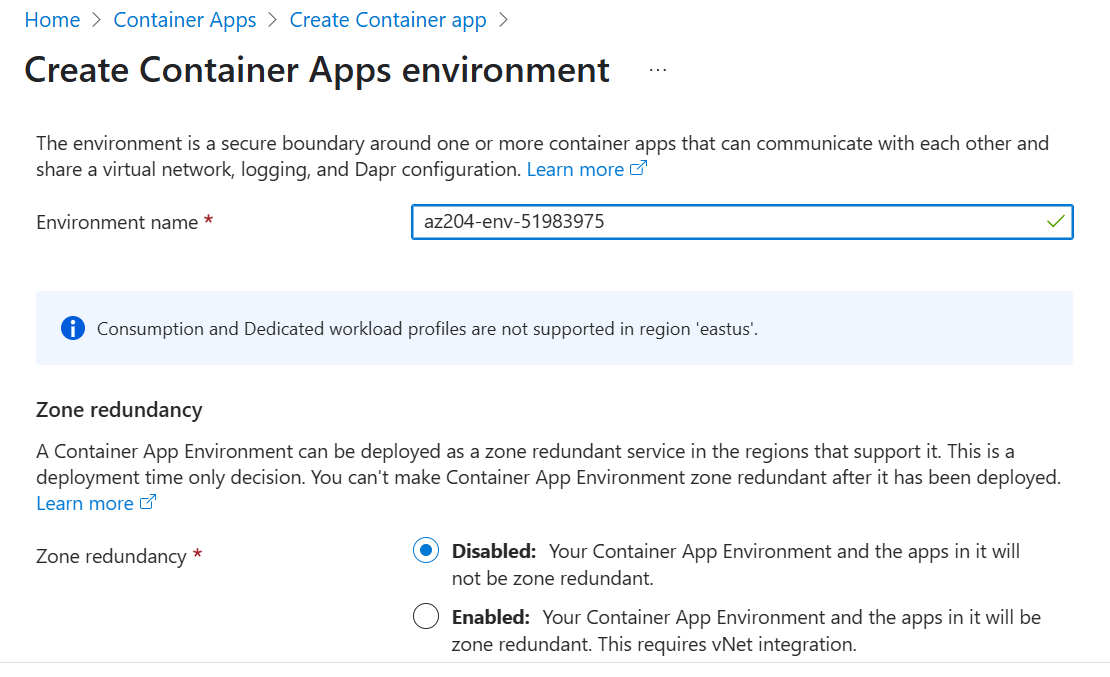
**By setting --ingress to external, you make the container app available to public requests. The command returns a link to access your app.**

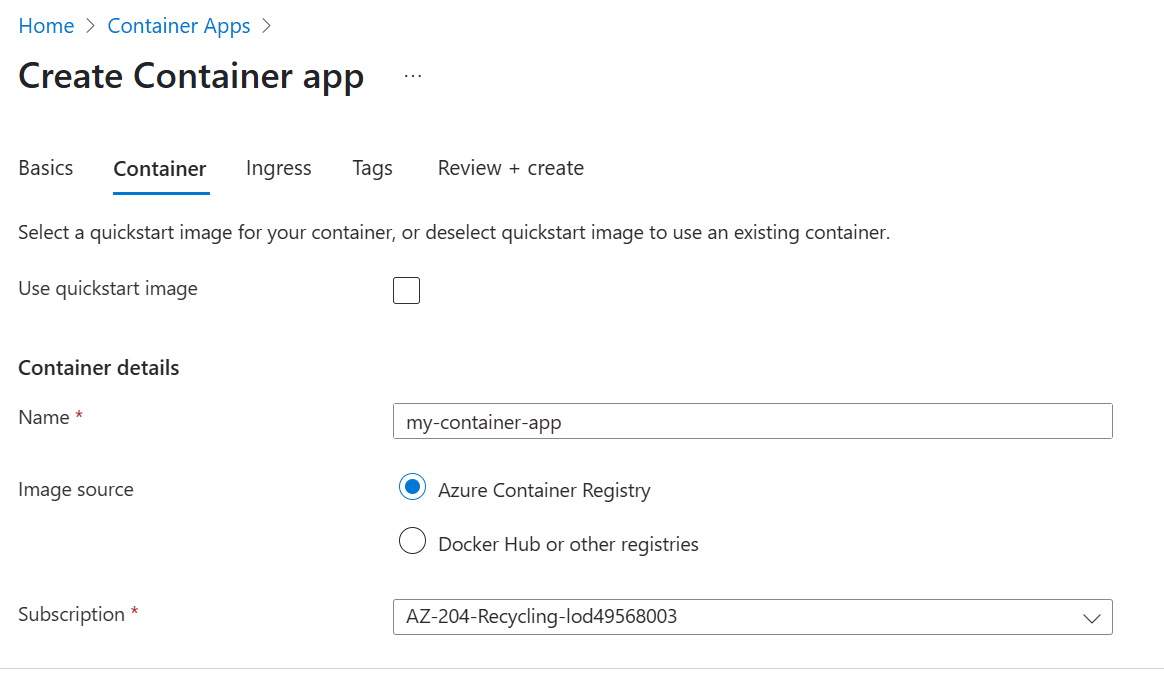
1. **Select the link returned by the az containerapp create command to verify the container app is running.**

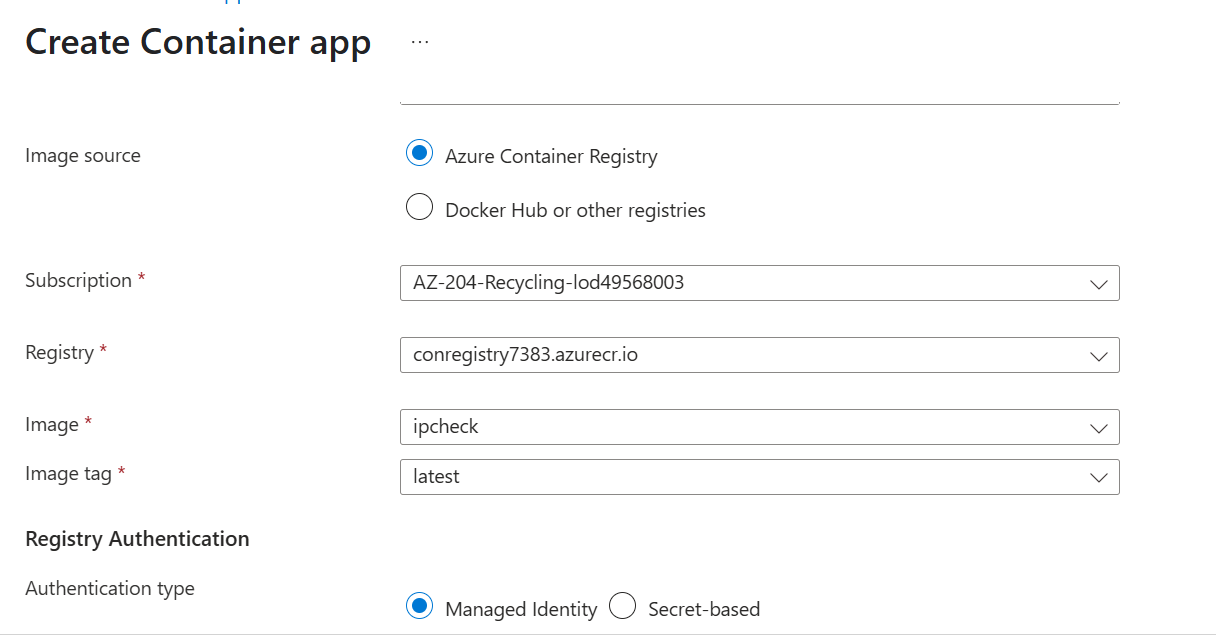
****

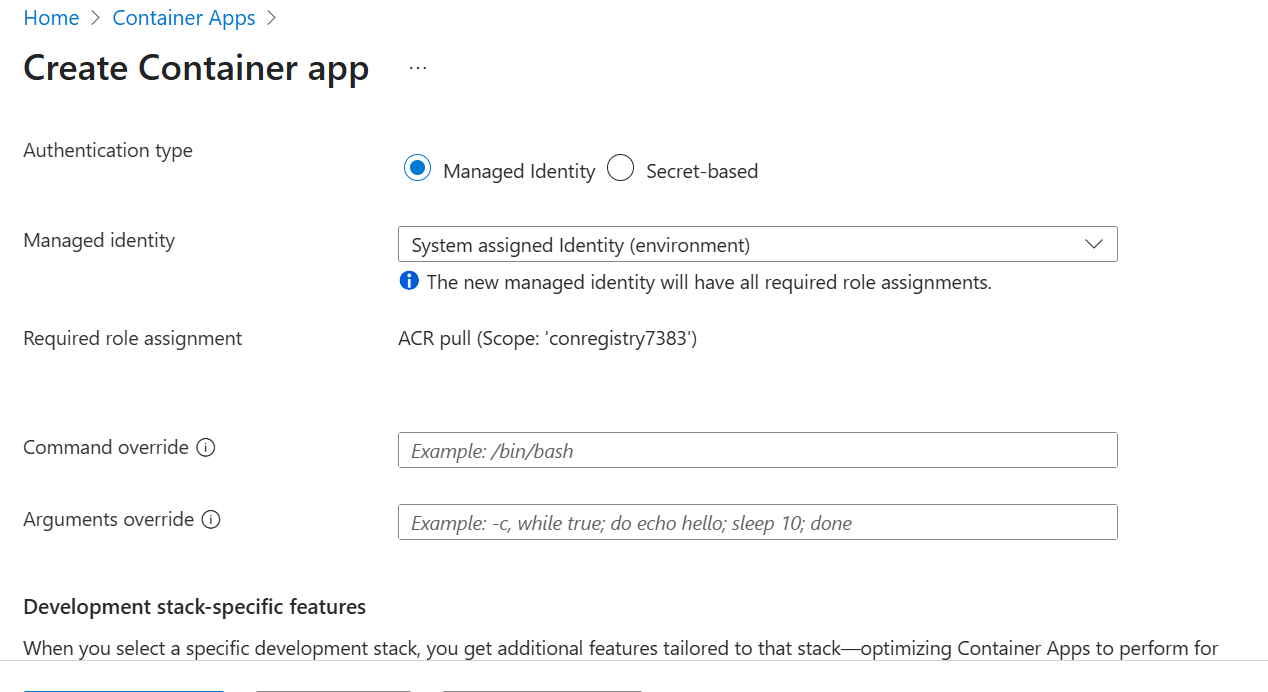
****

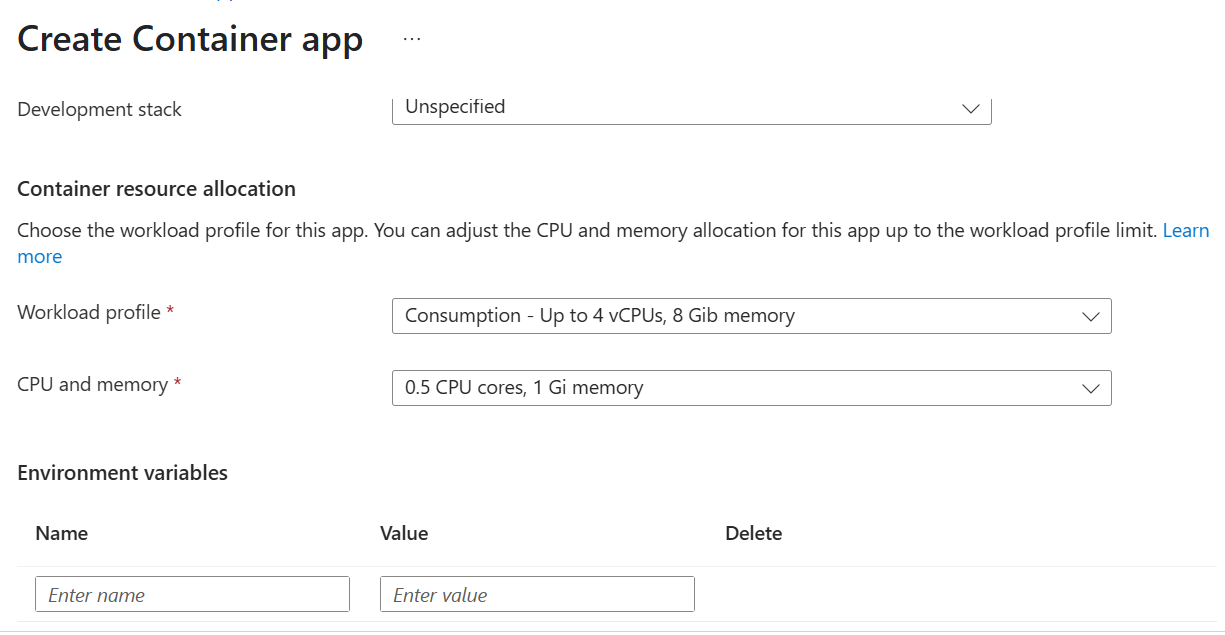
****

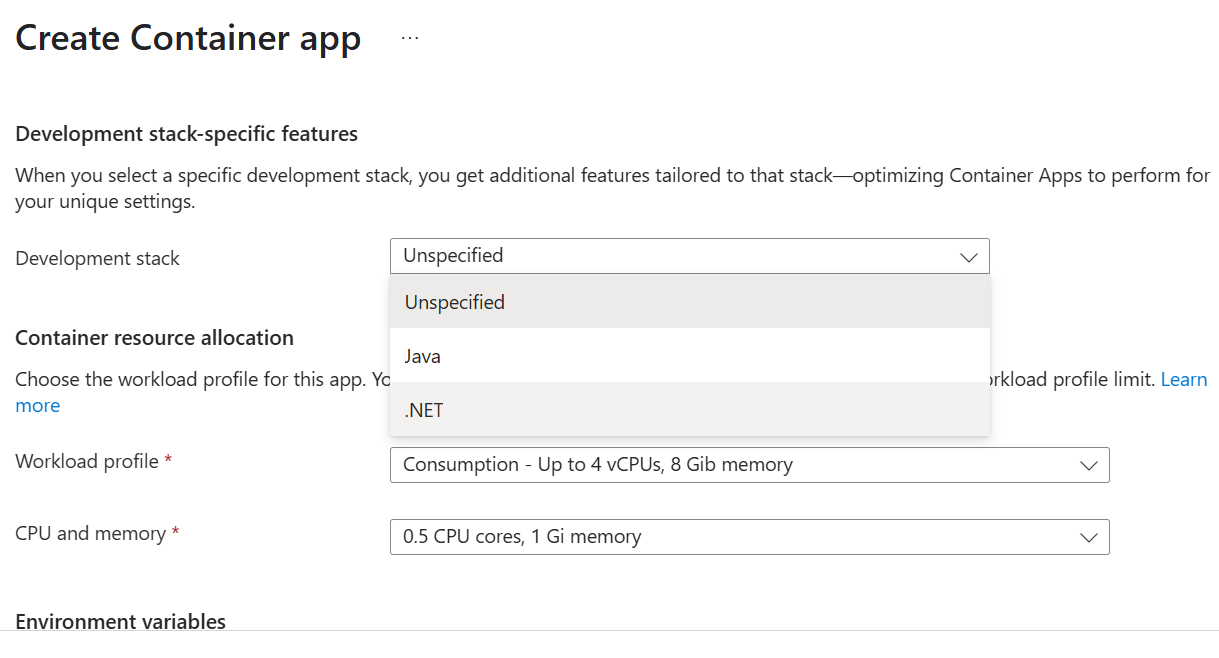
****

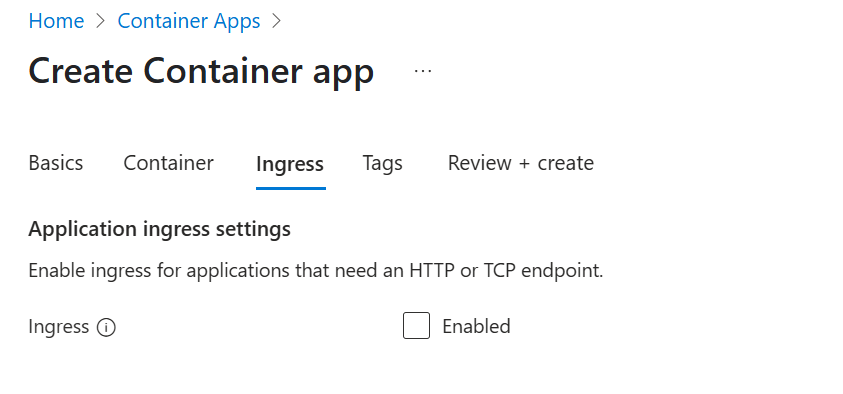
****

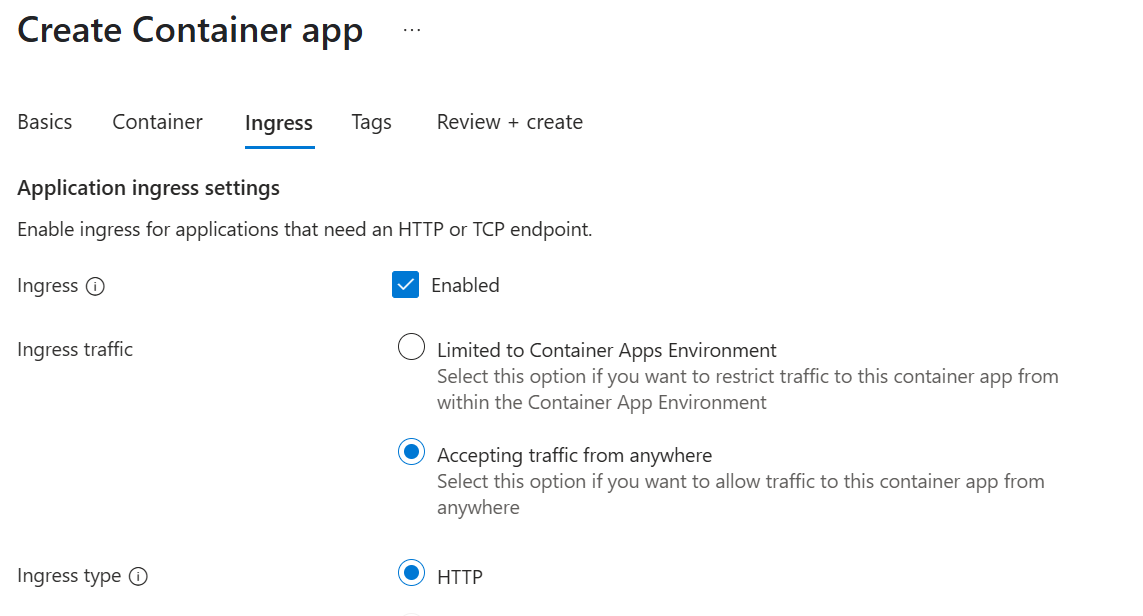
****

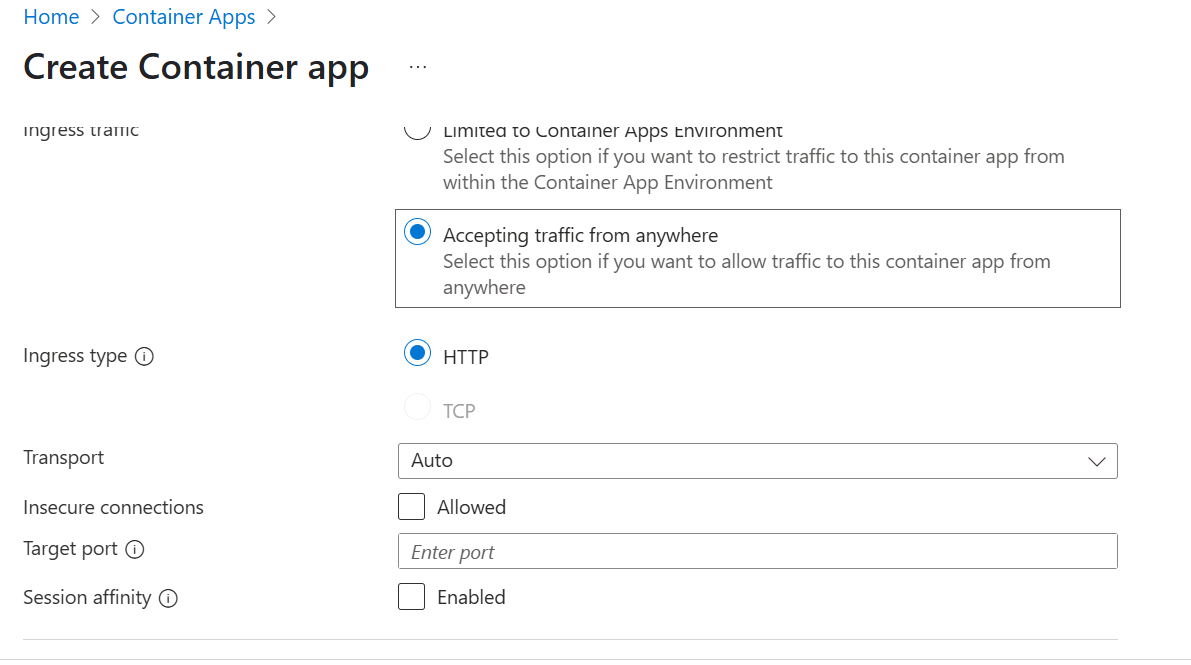
****

****

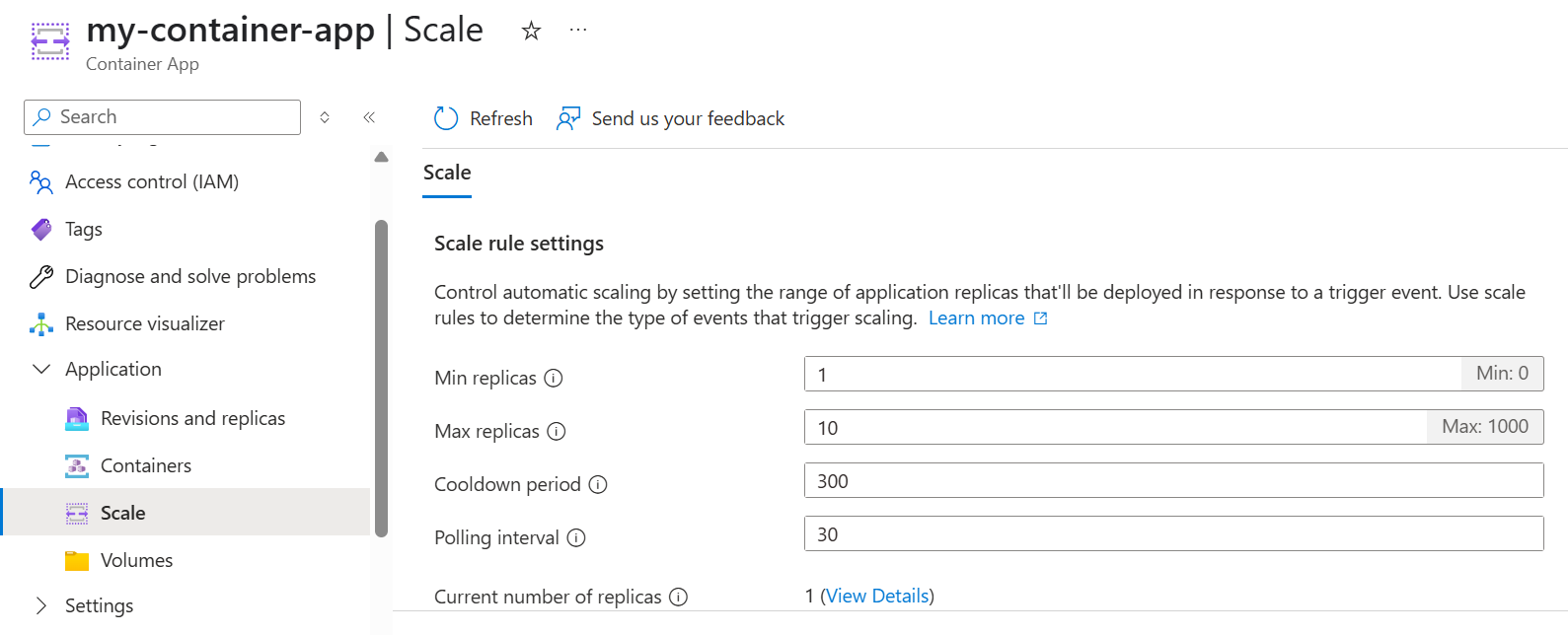
****

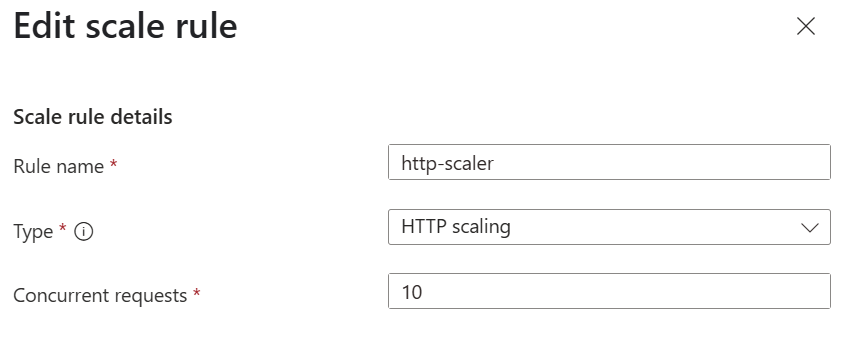
****

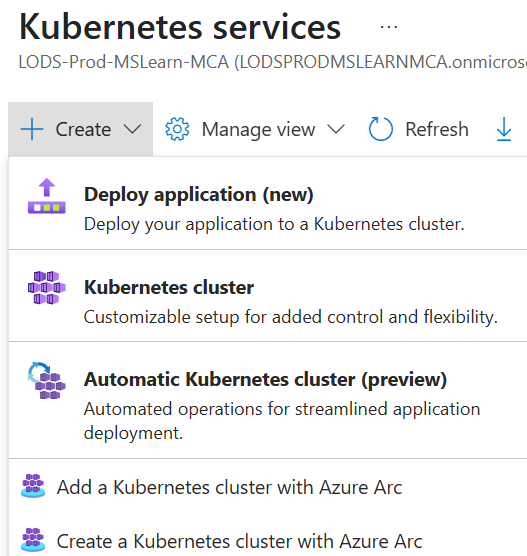
****

****

**my-container-app.salmonmoss-61cfcf75.eastus.azurecontainerapps.io**

****

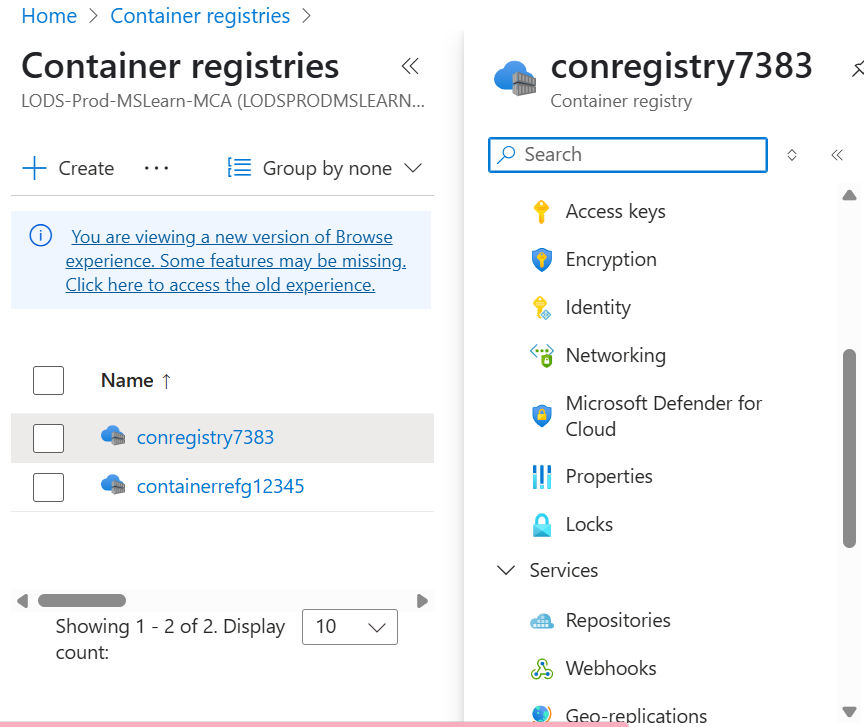
****

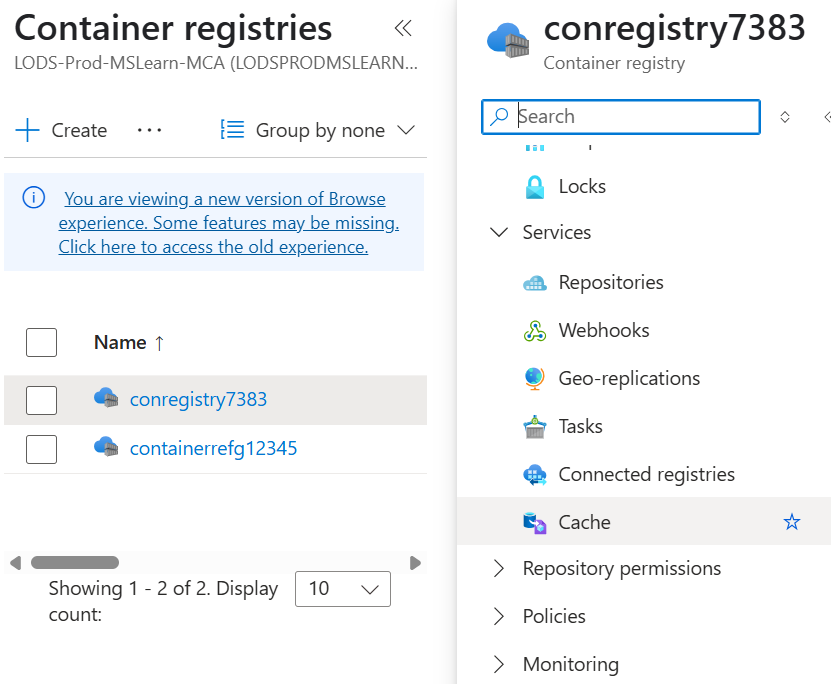
****

**AZ-1001: Deploy an Azure Kubernetes Service cluster**

**AZ-2003: Deploy cloud-native apps using Azure Container Apps**

**AZ-500: Microsoft Azure Security Technologies - Course Overview**

****

****

**Azure Container Instances documentation**

**Run Docker containers on-demand in a managed, serverless Azure environment. Azure Container Instances is a solution for any scenario that can operate in isolated containers, without orchestration. Run event-driven applications, quickly deploy from your container development pipelines, and run data processing and build jobs.**

**In this quickstart, you use an Azure Resource Manager template (ARM template) to deploy an isolated Docker container and make its web application available with a public IP address.**

**This quickstart shows how to create an Azure Container Registry instance by using an Azure Resource Manager template (ARM template). The template sets up a**[**geo-replicated**](https://learn.microsoft.com/en-us/azure/container-registry/container-registry-geo-replication)**registry, which automatically synchronizes registry content across more than one Azure region. Geo-replication enables network-close access to images from regional deployments, while providing a single management experience. It's a feature of the**[**Premium**](https://learn.microsoft.com/en-us/azure/container-registry/container-registry-skus)**registry service tier.**

**https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/quickstarts/microsoft.storage/storage-account-create/azuredeploy.json**

[**https://learn.microsoft.com/en-us/azure/azure-resource-manager/templates/template-tutorial-create-multiple-instances?tabs=CLI%2Cazure-cli**](https://learn.microsoft.com/en-us/azure/azure-resource-manager/templates/template-tutorial-create-multiple-instances?tabs=CLI%2Cazure-cli)

[**https://learn.microsoft.com/en-us/azure/container-instances/container-instances-quickstart-template**](https://learn.microsoft.com/en-us/azure/container-instances/container-instances-quickstart-template)

[**https://portal.azure.com/#create/Microsoft.Template**](https://portal.azure.com/#create/Microsoft.Template)