SAAS: Aplicación usando Docker

Instalar docker

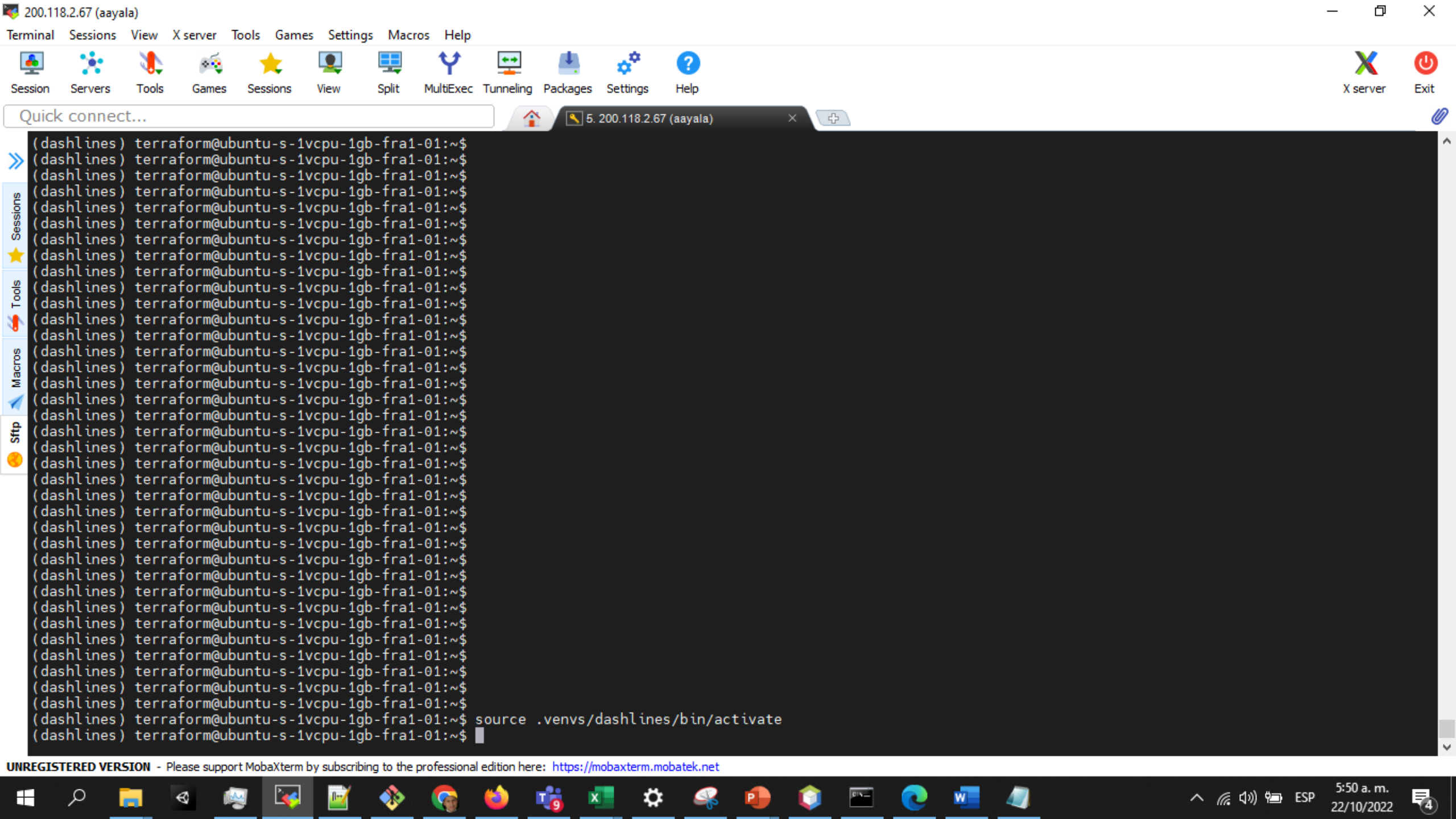
Clone a git repository

Git clone https://github.com/alfonsoayalapaloma/python-docker.git

Create a virtual environment

python3 -m venv .venvs/dashlines

source .venvs/dashlines/bin/activate


200.118.2.67 (aayala)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
5. 200.118.2.67 (aayala)
Sessions Tools Macros Sftp
(dashlines) terraform@ubuntu-s-1vcpu-1gb-fra1-01:~$
(dashlines) terraform@ubuntu-s-1vcpu-1gb-fra1-01:~$ source .venvs/dashlines/bin/activate
(dashlines) terraform@ubuntu-s-1vcpu-1gb-fra1-01:~$
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
ESP
5:50 a. m.
22/10/2022

Install requirements

Cd Python-docker

Pip install –r requirements.txt

Create a Docker image

docker build --tag python-docker .

200.118.2.67 (aayala)

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help X server Exit

Quick connect...

5. 200.118.2.67 (aayala)

```
root@ubuntu-s-1vcpu-1gb-fra1-01:/home/terraform/python-docker# docker build --tag python-docker .
Sending build context to Docker daemon  63.49kB
Step 1/6 : FROM python:3.8-slim-buster
 ---> 5f3ce1d922f0
Step 2/6 : WORKDIR /python-docker
 ---> Using cache
 ---> b454bd89624c
Step 3/6 : COPY requirements.txt requirements.txt
 ---> Using cache
 ---> ceaa8b203b41
Step 4/6 : RUN pip3 install -r requirements.txt
 ---> Using cache
 ---> 954bdd70b940
Step 5/6 : COPY . .
 ---> 181b6473dce6
Step 6/6 : CMD [ "python3", "application.py"]
 ---> Running in c6708922d01a
Removing intermediate container c6708922d01a
 ---> aa0ade5e828e
Successfully built aa0ade5e828e
Successfully tagged python-docker:latest
root@ubuntu-s-1vcpu-1gb-fra1-01:/home/terraform/python-docker#
root@ubuntu-s-1vcpu-1gb-fra1-01:/home/terraform/python-docker#
root@ubuntu-s-1vcpu-1gb-fra1-01:/home/terraform/python-docker#
root@ubuntu-s-1vcpu-1gb-fra1-01:/home/terraform/python-docker#
root@ubuntu-s-1vcpu-1gb-fra1-01:/home/terraform/python-docker#
root@ubuntu-s-1vcpu-1gb-fra1-01:/home/terraform/python-docker#
root@ubuntu-s-1vcpu-1gb-fra1-01:/home/terraform/python-docker#
root@ubuntu-s-1vcpu-1gb-fra1-01:/home/terraform/python-docker#
root@ubuntu-s-1vcpu-1gb-fra1-01:/home/terraform/python-docker#
root@ubuntu-s-1vcpu-1gb-fra1-01:/home/terraform/python-docker#
root@ubuntu-s-1vcpu-1gb-fra1-01:/home/terraform/python-docker#
root@ubuntu-s-1vcpu-1gb-fra1-01:/home/terraform/python-docker#
root@ubuntu-s-1vcpu-1gb-fra1-01:/home/terraform/python-docker#
root@ubuntu-s-1vcpu-1gb-fra1-01:/home/terraform/python-docker#
root@ubuntu-s-1vcpu-1gb-fra1-01:/home/terraform/python-docker#
root@ubuntu-s-1vcpu-1gb-fra1-01:/home/terraform/python-docker#
root@ubuntu-s-1vcpu-1gb-fra1-01:/home/terraform/python-docker#
```


ESP 5:55 a. m. 22/10/2022

Create a kubernetes cluster in azure

docker build --tag python-docker .

Create Kubernetes cluster

Basics | Node pools | Access | Networking | Integrations | Advanced | Tags | Review + create

Azure Kubernetes Service (AKS) manages your hosted Kubernetes environment, making it quick and easy to deploy and manage containerized applications without container orchestration expertise. It also eliminates the burden of ongoing operations and maintenance by provisioning, upgrading, and scaling resources on demand, without taking your applications offline. Learn more about Azure Kubernetes Service

Project details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : Azure for Students

Resource group * : (New) Resource group

Create new

Cluster details

Cluster preset configuration : Standard ($$)

To quickly customize your Kubernetes cluster, choose one of the preset configurations above. You can modify these configurations at any time.
Learn more and compare presets

Review + create | < Previous | Next : Node pools >

Create a kubernetes cluster in azure

docker build --tag python-docker .

Create Kubernetes cluster

Basics | Node pools | Access | Networking | Integrations | Advanced | Tags | Review + create

Azure Kubernetes Service (AKS) manages your hosted Kubernetes environment, making it quick and easy to deploy and manage containerized applications without container orchestration expertise. It also eliminates the burden of ongoing operations and maintenance by provisioning, upgrading, and scaling resources on demand, without taking your applications offline. Learn more about Azure Kubernetes Service

Project details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : Azure for Students

Resource group * : (New) Resource group

Create new

Cluster details

Cluster preset configuration : Standard ($$)

To quickly customize your Kubernetes cluster, choose one of the preset configurations above. You can modify these configurations at any time.
Learn more and compare presets

Review + create | < Previous | Next : Node pools >

Create Kubernetes cluster

99.5%
Optimize for cost.

99.5% API server availability is recommended for dev/test configuration.

Primary node pool

The number and size of nodes in the primary node pool in your cluster. For production workloads, at least 3 nodes are recommended for resiliency. For development or test workloads, only one node is required. If you would like to add additional node pools or to see additional configuration options for this node pool, go to the 'Node pools' tab above. You will be able to add additional node pools after creating your cluster.
Learn more about node pools in Azure Kubernetes Service

Node size * ⓘ

Standard B2s
2 vcpus, 4 GiB memory
Standard B4ms is recommended for dev/test configuration.
Change size

Scale method * ⓘ

- (•) Manual
- ( ) Autoscale

Autoscaling is recommended for dev/test configuration.

Node count * ⓘ — 2

Review + create | < Previous | Next : Node pools >

Create Kubernetes cluster

In addition to the required primary node pool configured on the Basics tab, you can also add optional node pools to handle a variety of workloads Learn more about node pools

Add node pool | Delete

Name	Mode	OS type	Node count	Node size
agentpool	System	Linux	1	Standard_B2s

Enable virtual nodes

Virtual nodes allow burstable scaling backed by serverless Azure Container Instances. Learn more about virtual nodes

Enable virtual nodes

Node pool OS disk encryption

By default, all disks in AKS are encrypted at rest with Microsoft-managed keys. For additional control over encryption, you can supply your own keys using a disk encryption set backed by an Azure Key Vault. The disk encryption set will be used to encrypt the OS disks for all node pools in the cluster. Learn more

Encryption type: (Default) Encryption at-rest with a platform-managed key

Review + create | < Previous | Next : Access >

Create Kubernetes cluster

Validation passed

Basics | Node pools | Access | Networking | Integrations | Advanced | Tags | Review + create

Basics

Subscription	Azure for Students
Resource group	(new) aks_rg
Region	UK West
Kubernetes cluster name	aks
Kubernetes version	1.23.12
Enable automatic upgrades	False

Node pools

Node pools	1
Enable virtual nodes	Disabled

Create | < Previous | Next > | Download a template for automation

Create Kubernetes cluster

Validation passed

Access

Resource identity	System-assigned managed identity
Local accounts	Enabled
Authentication and Authorization	Local accounts with Kubernetes RBAC
Encryption type	(Default) Encryption at-rest with a platform-managed key

Networking

Network configuration	Kubenet
DNS name prefix	aks-dns
Load balancer	Standard
Private cluster	Disabled
Authorized IP ranges	Disabled
Network policy	None
HTTP application routing	No

Integrations

Create | < Previous | Next > | Download a template for automation

microsoft.aks-20221021190537 | Overview

Deployment

Search
Overview
Inputs
Outputs
Template

Delete | Cancel | Redeploy | Download | Refresh

Your deployment is complete

Deployment name: microsoft.aks-20221021190537
Subscription: Azure for Students
Resource group: aks_rg

Start time: 10/21/2022, 7:19:52 PM
Correlation ID: d21914ae-6534-4777-b07c-c9099233d6de

Deployment details

Next steps

Create a quick start application Recommended

Create a Kubernetes deployment Recommended

Integrate automatic deployments within your cluster Recommended

Connect to cluster Recommended

Go to resource | Connect to cluster

Give feedback

Tell us about your experience with deployment

Cost Management

Get notified to stay within your budget and prevent unexpected charges on your bill.

Set up cost alerts >

Container Insights

Comprehensive health and performance data in addition to the default cluster metrics

Go to Azure Monitor insights

Free Microsoft tutorials

Introduction to Docker containers

Build and store container images with Azure Container Registry

Introduction to Azure Kubernetes Service

Create application on Cluster

Create a starter application

Deploy a basic web or single-image application to get started with Kubernetes. No prior Kubernetes experience required.

If you already have a YAML file or want to create a more complex deployment, add with YAML instead.

Create a basic web application

Learn how to create and deploy a basic web app with just a few clicks.

Create >

Create a single-image application

Learn how to create and deploy a single-image from a container registry with a quick tutorial.

Create >

Create a single-image application

1 Get image | 2 Application details | 3 Review YAML | 4 Deploy

This experience will walk through the process of selecting an image from a registry or adding an image to a registry, configuring key deployment parameters, and reviewing the YAML file that will be sent to the cluster.

If you already have a YAML file or want to create a more complex deployment, add with YAML instead.

Container registry details

Before containers can be deployed to your cluster, a container image must be uploaded to the container registry.

Container registry type

- (•) Azure Container Registry
- ( ) Other registry

Container registry *

No registries found

Create new

Image details

After choosing a registry, you must choose or create a container image. A container image is the foundation for a Kubernetes deployment and is the blueprint used to create all containers.

Previous | Next

Create a single-image application

...

Container registry details

Before containers can be deployed to your cluster, a container image must be uploaded to the container registry.

Container registry type ⓘ

- (•) Azure Container Registry
- ( ) Other registry

Container registry ⓘ * No registries found

The value must not be empty.

Create new

Image details

After choosing a registry, you must choose or create a container image. A container image is the foundation for a Kubernetes deployment and is the blueprint used to create all containers.

Image ⓘ * No images found

The value must not be empty.

Image tag ⓘ * No image tags found

The value must not be empty.

Previous | Next

Create a container registry

Create a basic container registry using the defaults Azure has provided. If you want to customize your registry, use the full Azure Container Registry creation experience.

Registry name * aksregistry93

Subscription * Azure for Students

Resource group * (New) aksregistry_group

Create new

Region ⓘ (Europe) UK West

Admin user ⓘ * Yes

SKU ⓘ * Standard

Create | Cancel

Create a single-image application

Container registry details

Before containers can be deployed to your cluster, a container image must be uploaded to the container registry.

Container registry type

- (•) Azure Container Registry
- ( ) Other registry

Container registry * aksregistry93

Create new

Image details

After choosing a registry, you must choose or create a container image. A container image is the foundation for a Kubernetes deployment and is the blueprint used to create all containers.

Image * No images found

The value must not be empty.

Refresh

Looks like there are no images available in this registry. Choose another registry or learn how to upload an image to a container registry.

Previous | Next

Upload a local image

Use your local Docker installation and the Azure CLI to push it to your container registry.

Go back

1 Open the Azure CLI

Learn how to install the Azure CLI

2 Set the subscription

```
# Set the subscription
az account set --subscription "Azure for Students"

# Attach the registry to the cluster
az aks update -n aks -g aks_rg --attach-acr aksregistry93

# Login to the registry
az acr login -n  aksregistry93
```

3 Attach the registry to the cluster

Image name *

Close

Upload your image to Azure

az account set --subscription "Azure for Students"

az aks update -n aks -g aks_rg --attach-acr aksregistry93

az acr login -n aksregistry93

docker tag python-docker:latest aksregistry93.azurecr.io/python-docker:latestdocker

push aksregistry93.azurecr.io/python-docker:latest

Create a single-image application

Container registry details

Before containers can be deployed to your cluster, a container image must be uploaded to the container registry.

Container registry type: Azure Container Registry / Other registry

Container registry * : aksregistry93

Create new

Image details

After choosing a registry, you must choose or create a container image. A container image is the foundation for a Kubernetes deployment and is the blueprint used to create all containers.

Image * : python-docker

Image tag * : latest

OS type : Linux

Previous | Next

Upload a local image

3 Attach the registry to the cluster

Image name * : python-docker

Image tag * : latest

```
# Tag your local image
 docker tag python-docker:latest aksregistry93.azurecr.io/python-docker:latest

# Push the image to the registry
 docker push aksregistry93.azurecr.io/python-docker:latest
```

4 Try pulling from your registry (optional)

```
az acr run --registry aksregistry93 --cmd '$Registry/python-docker:latest' /dev/nu
```

Close

Create a single-image application

Deployment name * (i) python-docker

Replicas

Replicas are the copies of your container that'll be created. Each copy of your container will run as a pod within the deployment.

Replicas * (i) 1

Resources

Resources are the memory and CPU that your containers will use to run. **Requests** are the **minimum** amount of CPU and memory resources your container needs to run. This helps the Kubernetes schedular determine where there is enough free space in the cluster to run your pods and containers. If no requests are specified, Kubernetes will default to 0, meaning no space will be reserved for each pod. Learn more Learn more

CPU request (i) 0

Examples: 2, 0.5, 500m

Memory request (i) 0

Previous Next

Create a single-image application

Networking

Service

A service is a Kubernetes resource that exposes your application outside of the cluster. The service created here, by default, will use a load balancer to give the set of pods in this application an external IP address.

Expose via a service * ☑

Service name * python-docker-service

Ports

Ports are exposed by your deployment container but are not exposed outside the cluster, and can be used by a service to facilitate communications outside of the cluster.

Name	Port	Port protocol
port80	80	TCP
	80	TCP

Previous Next

Create a single-image application

```
28                ports:
29                  - containerPort: 80
30                resources:
31                  requests:
32                    cpu: '0'
33                    memory: '0'
34                  limits:
35                    cpu: '256'
36                    memory: 11400G
37    - apiVersion: v1
38      kind: Service
39      metadata:
40        name: python-docker-service
41        namespace: default-1666399846961
42      spec:
43        type: LoadBalancer
44        ports:
45          - targetPort: 80
46            name: port80
47            port: 80
48            protocol: TCP
49        selector:
50          app: python-docker
```

Previous | Deploy

Create a single-image application

...

Select a deployment name to see more details, including the individual pods that were created as part of the deployment.

	Resource	Type	Status
✓	default-1666399846961	Namespace	Success
✓	python-docker	Deployment	Success
✓	python-docker-service	Service	Success
↻	python-docker-pods	Pod	⚠ 0/1 pods ready ⓘ

3. Next steps

Here are some actions you can take once your application is deployed.

View the application

View the deployed application by going to the external IP address associated with the frontend service.

Waiting for application to be available.

Previous | Close

Create a single-image application

✓	python-docker-service	Service	Success
✓	python-docker-pods	Pod	1/1 pods ready

3. Next steps

Here are some actions you can take once your application is deployed.

View the application

View the deployed application by going to the external IP address associated with the frontend service.

View application

Learn how to create and deploy custom container images

Set up a GitOps pipeline to automatically deploy your applications from source control to your cluster. Configure GitOps

Give feedback

Help improve this page

Previous | Close

Life expentancy progression of countries per continents

☐ Asia ☐ Europe ☐ Africa ☑ Americas ☑ Oceania