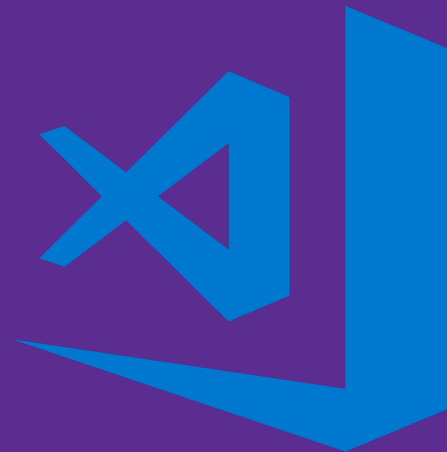




From Zero to Azure with Python, Docker Containers, and VS Code

Dan Taylor

Program Manager
Python Developer Tools



Pre-requisites

Azure Subscription

Visual Studio Code

Git Command Line Tools

Docker

Steps

Create App and Run in Local Docker Container

Deploy to Azure Web Apps for Containers

Create CosmosDB database using Azure Notebooks

Update Code and Redeploy



Azure for Python Developers

Build Python web apps using Docker containers



App Service



Kubernetes Service

Power AI and ML workloads



Azure Notebooks



Machine Learning



Batch AI

Store and Retrieve Data Securely & Reliably



Redis Cache



Storage



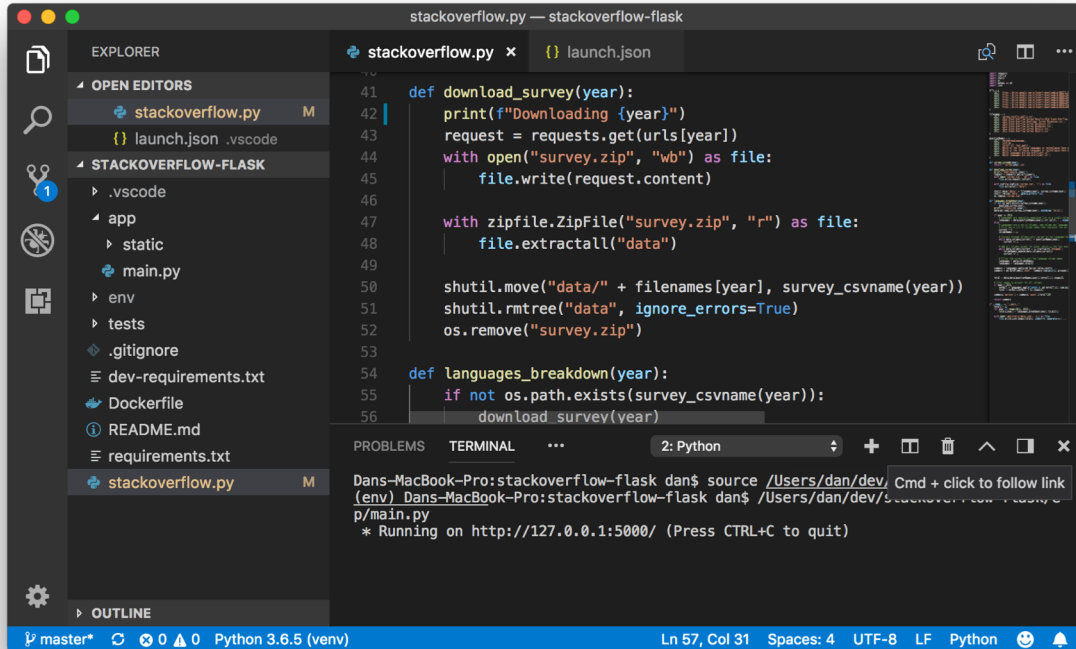
Azure Database
for PostgreSQL



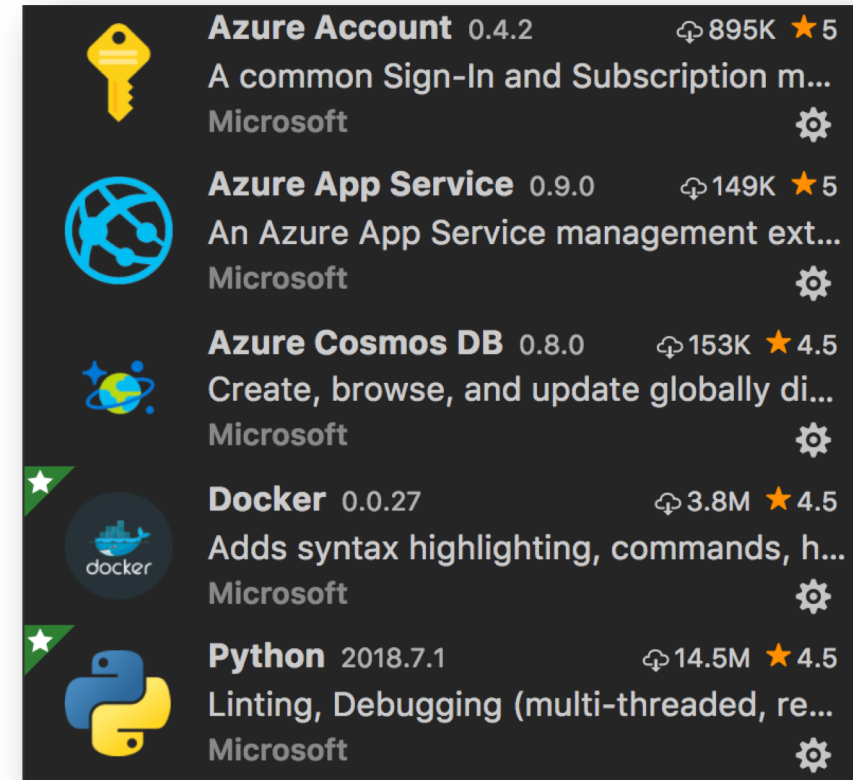
Cosmos DB



Visual Studio Code



+ Extensions:



Free, cross-platform, open source
Fast and lightweight
Rich extension ecosystem

Create App and Run in Local Docker Container

Create app

Make directory

Code .

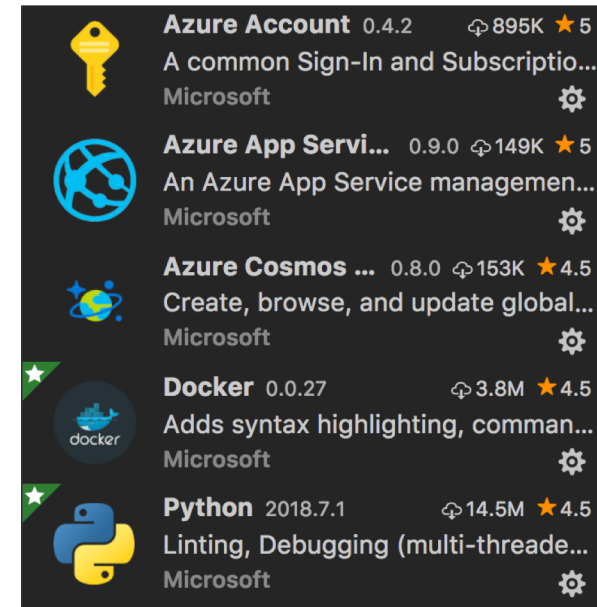
Install Extensions

Write 'hello world' flask app

Add dockerfiles

Change base image

Set listen port



```
>docker: add|
```

Docker: Add Docker files to Workspace

```
FROM tiangolo/uwsgi-nginx-flask:python3|
```

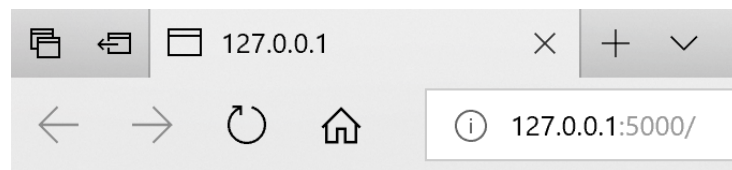
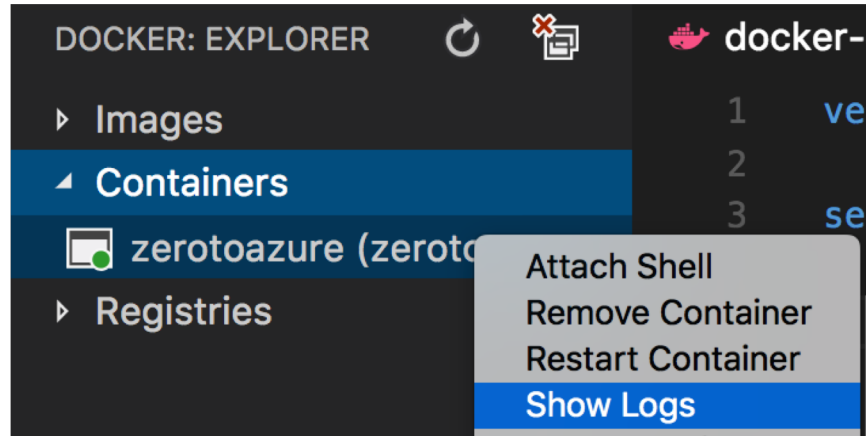
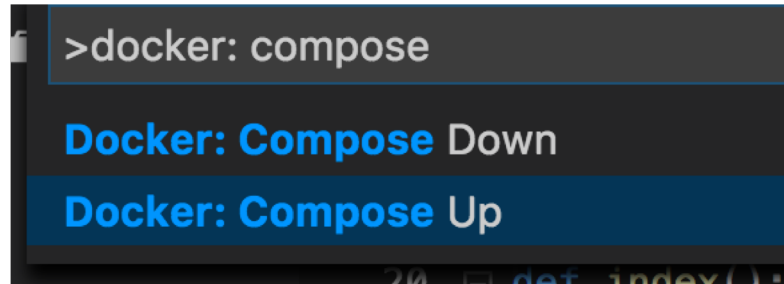
```
python3.5  
# If you prefer miniconda: python3.5-index  
#FROM continuumio/miniconda: python3.6  
python3.6-alpine3.7
```

```
ENV LISTEN_PORT=8000
```

```
WORKDIR /app
```

```
ADD app /app
```

Run docker container



Compose up

Show logs

Browse to localhost

Deploy to Azure Web Apps for Containers

Sign In to Azure

Command: Azure Sign In

Paste code into browser

```
>azure: Sign in
```

Azure: Sign In

Device Login

Enter the code that you received from the application on your device

Visual Studio Code

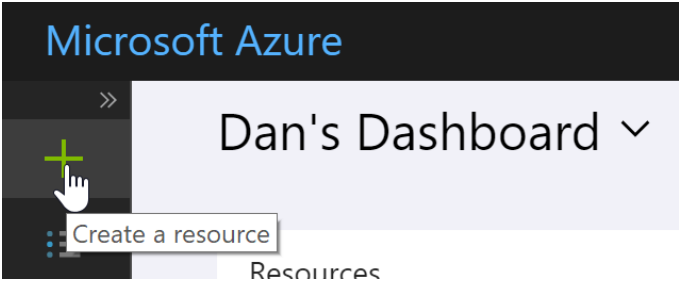
Click Cancel if this isn't the application you were trying to sign in to on your device.

Continue

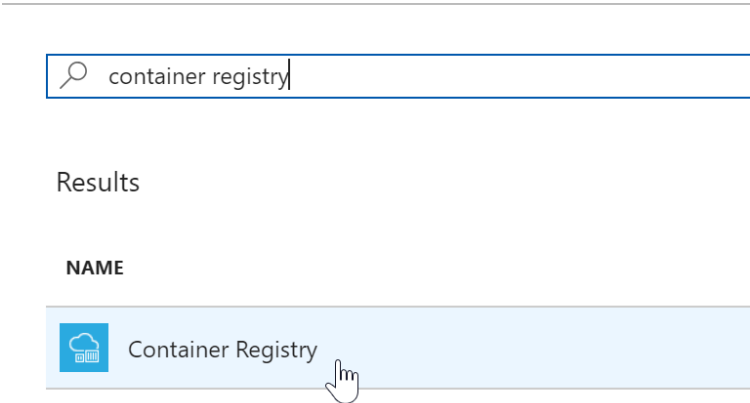
Cancel

Create Azure Resource Group and Container Registry

1.



2.



3.

Create container registry

* Registry name

zerotoazure

.azurecr.io

* Subscription

Dan Taylor - Internal

* Resource group

Create new

Use existing

zerotoazure

* Location

South Central US

* Admin user

Enable

Disable

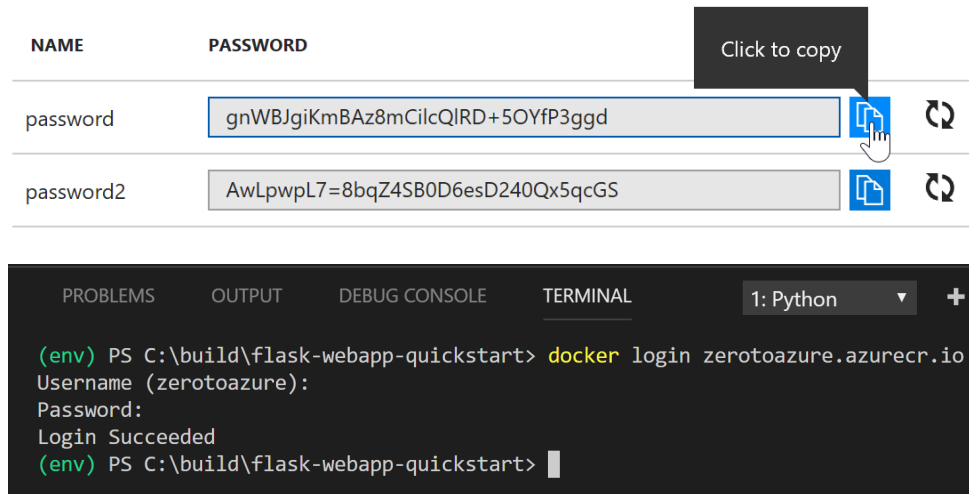
* SKU

Standard

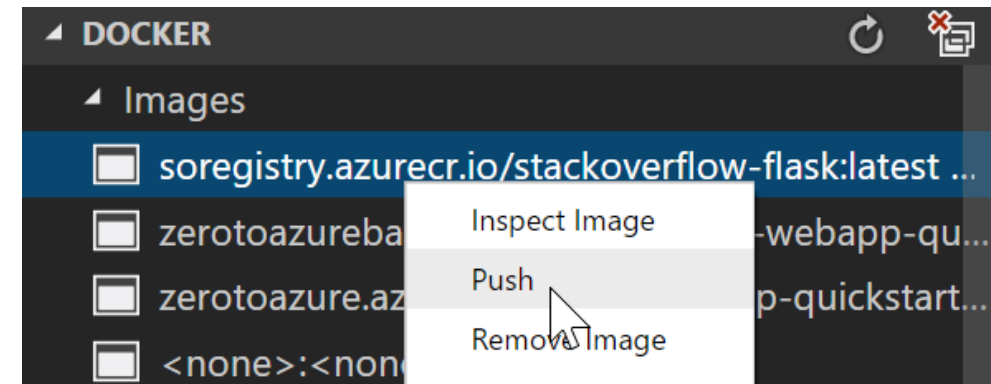
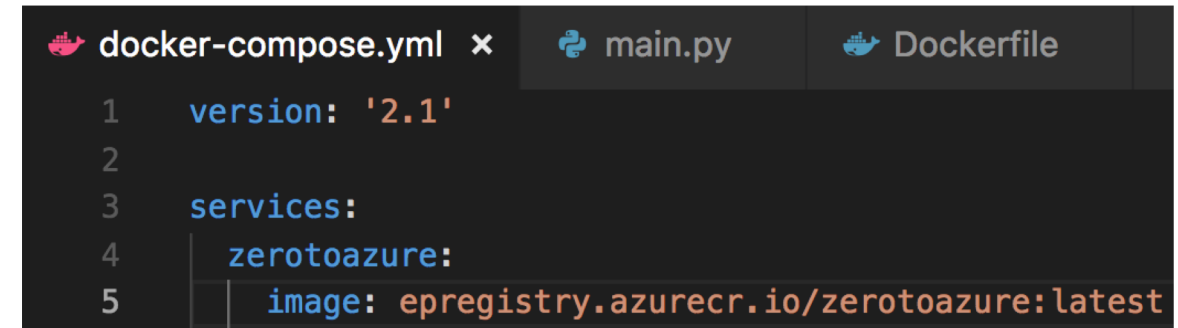
Build Container and Push to Registry

docker login

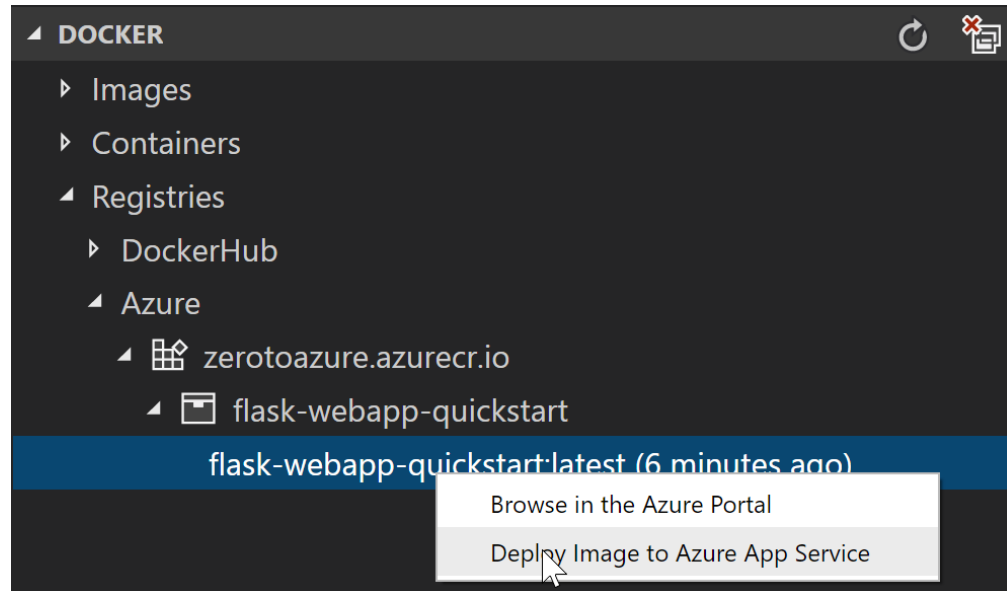
<registry_name>.azurecr.io



Build image and push to Azure
container registry



Deploy Container to New Azure Web App

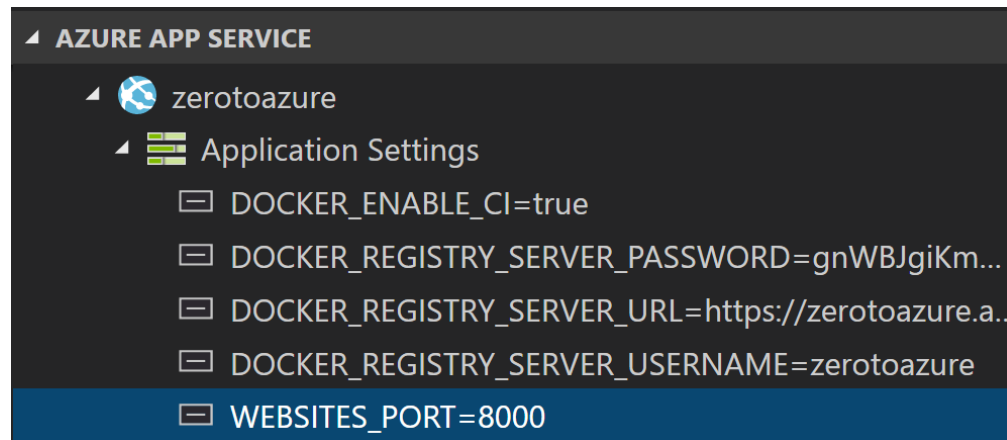


Resource group = epzerotoazure

Plan name: zerotoazureplan

Type: B1

Site name: epzerotoazure



Right-click > Add new setting...

Name: WEBSITES_PORT

Value: 8000

Add data using Azure Notebooks and CosmosDB

Create CosmosDB Account

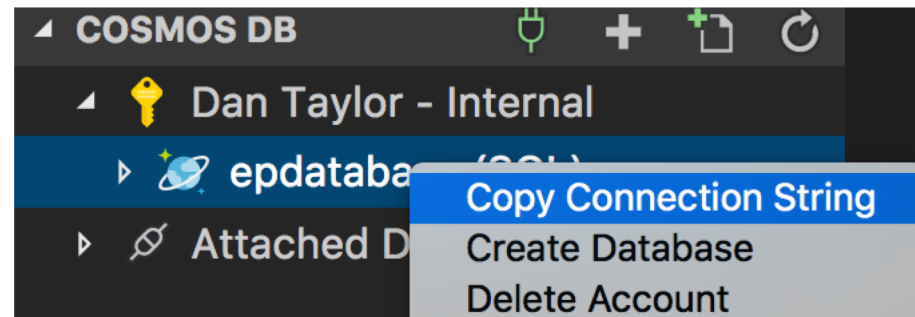
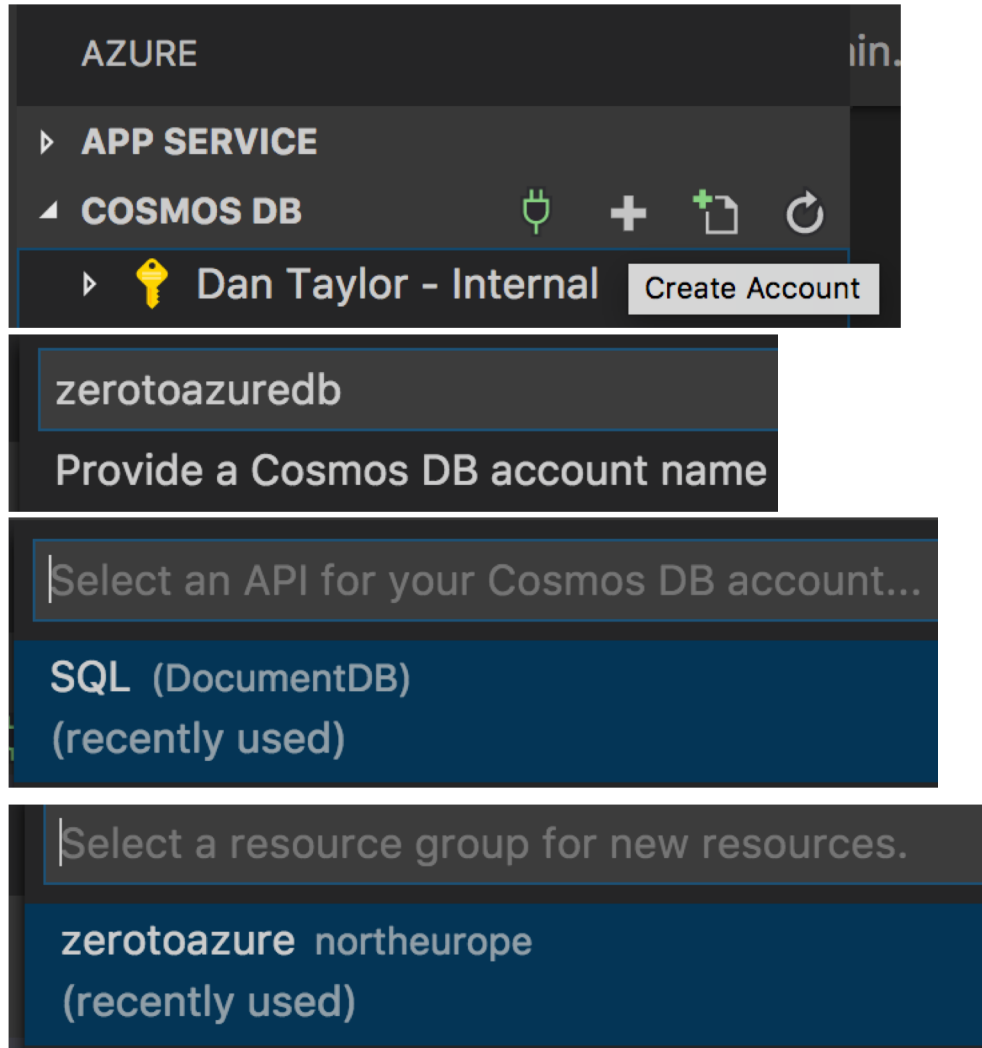
Create CosmosDB from VS Code

Provide a name

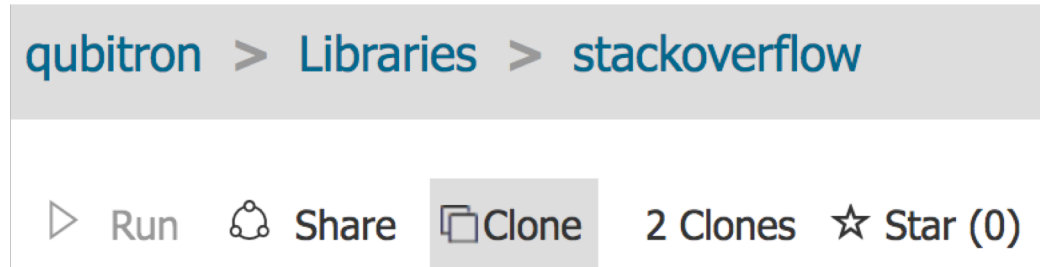
Choose SQL API schema

Choose resource group

Copy connection string



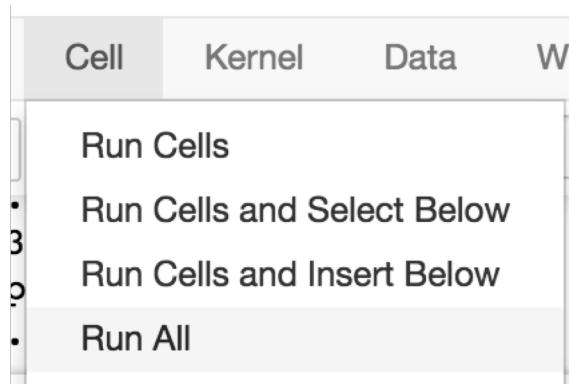
Clone + Run Azure Jupyter Notebook



Clone notebook from:
notebooks.azure.com/qubitron/libraries/stackoverflow

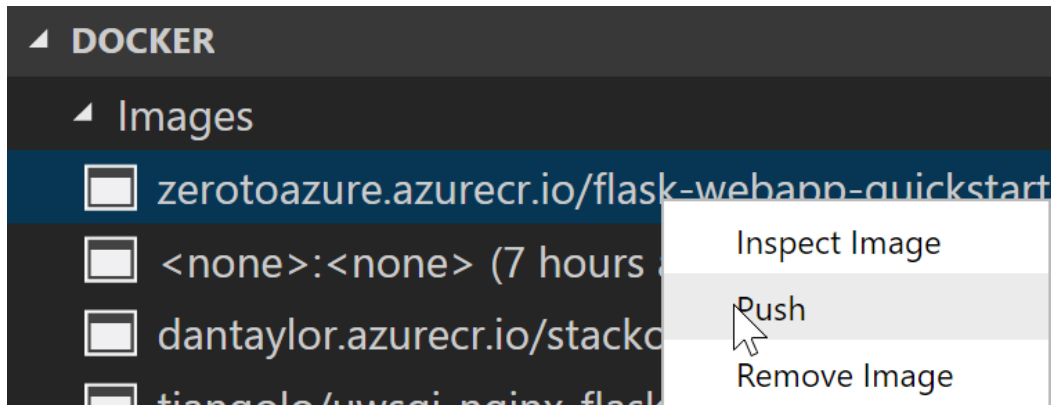
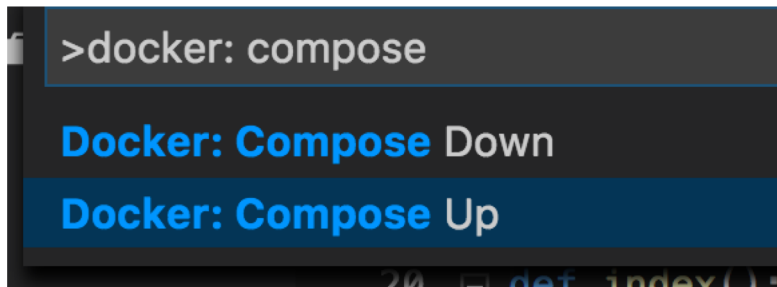
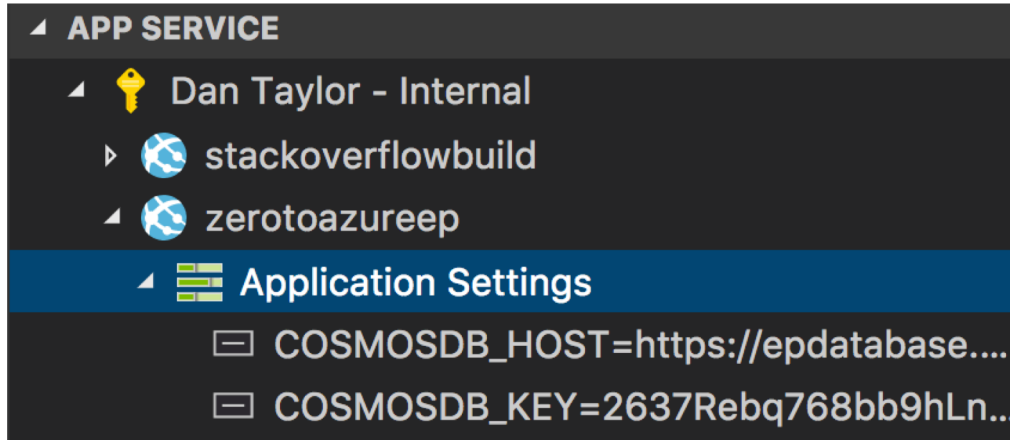
```
In [37]: # Define the settings/constants to use
HOST = 'https://epdatabase.documents.d
MASTER_KEY = '2637Rebq768bb9hLnYaDiyQ
DATABASE_ID = 'stackoverflow'
COLLECTION_ID = 'results'
DOCUMENT_ID = 'languages'
```

Paste connection string into notebook
Run all cells



Update Code and Redeploy

Change Code and Redeploy



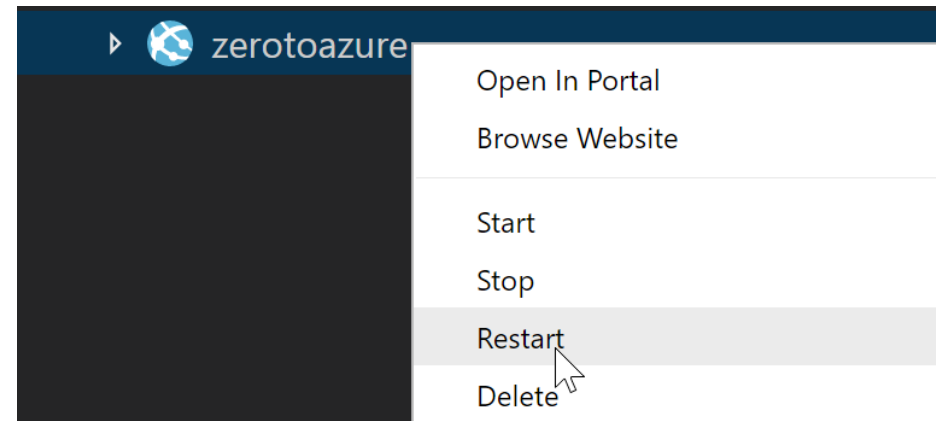
github.com/qubitron/stackoverflow-flask, europython-zero2azure branch

Add connection settings env vars

Compose up

Push

Restart web site



More Information

For information and updates, visit our blog at:

aka.ms/pythonblog

Code: github.com/qubitron/zerotoazure-flaskcosmos

