Cloud Onboarding for Azure

Microsoft Azure

There are many incredible resources for Microsoft.

- Microsoft Learn[46]
- Azure for Education[47]
- Azure Free Trial[48]

Using Github Actions with PyTest and Azure Cloud Shell

Let's also show how an initial cloud-based development environment could work with Azure Cloud Shell[49] and Github Actions[50]. The source code for this example project is here[51].

You can watch a screencast here of this workflow here.

Video Link: https://www.youtube.com/watch?v=rXXtJpcVems[52]

Steps to run this Azure Github Actions project

- Create a Github Repo (if not created)
- Open Azure Cloud Shell
- Create ssh-keys in Azure Cloud Shell
- Upload ssh-keys to Github
- Create scaffolding for the project (if not created)

- `Makefile`

It should look similar to the file below.

```
install:
    pip install --upgrade pip &&\
    pip install -r requirements.txt

test:
    python -m pytest -vv test_hello.py


lint:
    pylint --disable=R,C hello.py

all: install lint test
```

- `requirements.txt`

The requirements.txt should include:

```
pylint
pytest
```

- Create a python virtual environment and source it if not created

```
python3 -m venv ~/.myrepo
source ~/.myrepo/bin/activate
```

- Create initial `hello.py` and `test_hello.py`

hello.py

```
def toyou(x):
    return "hi %s" % x


def add(x):
    return x + 1


def subtract(x):
    return x - 1
```

test_hello.py

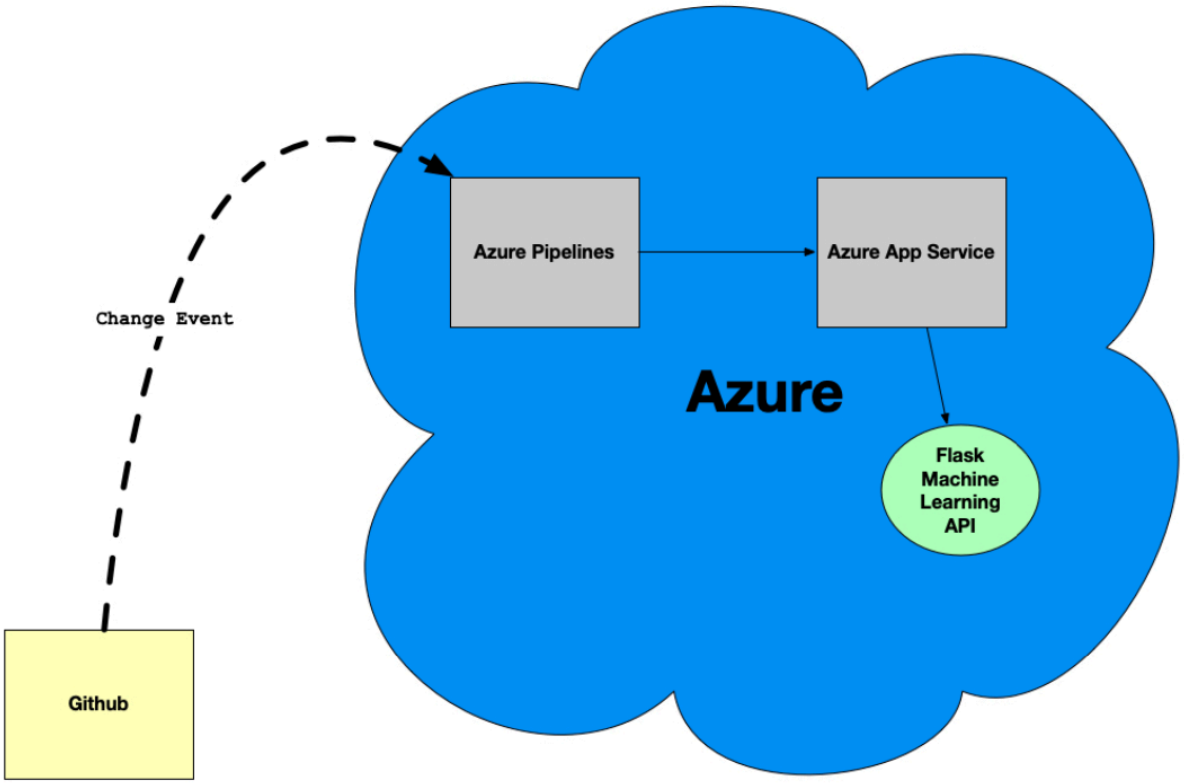
```
from hello import toyou, add, subtract


def setup_function(function):
    print("Running Setup: %s" % {function.__name__})
    function.x = 10


def teardown_function(function):
    print("Running Teardown: %s" % {function.__name__})
    del function.x


### Run to see failed test
#def test_hello_add():
# assert add(test_hello_add.x) == 12

def test_hello_subtract():
    assert subtract(test_hello_subtract.x) == 9
```

- Run `make all`, which will install, lint, and test code.
- Setup Github Actions in `pythonapp.yml`

```
name: Azure Python 3.5
on: [push]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v2
    - name: Set up Python 3.5.10
      uses: actions/setup-python@v1
      with:
        python-version: 3.5.10
    - name: Install dependencies
      run: |
        make install
    - name: Lint
      run: |
        make lint
    - name: Test
      run: |
        make test
```

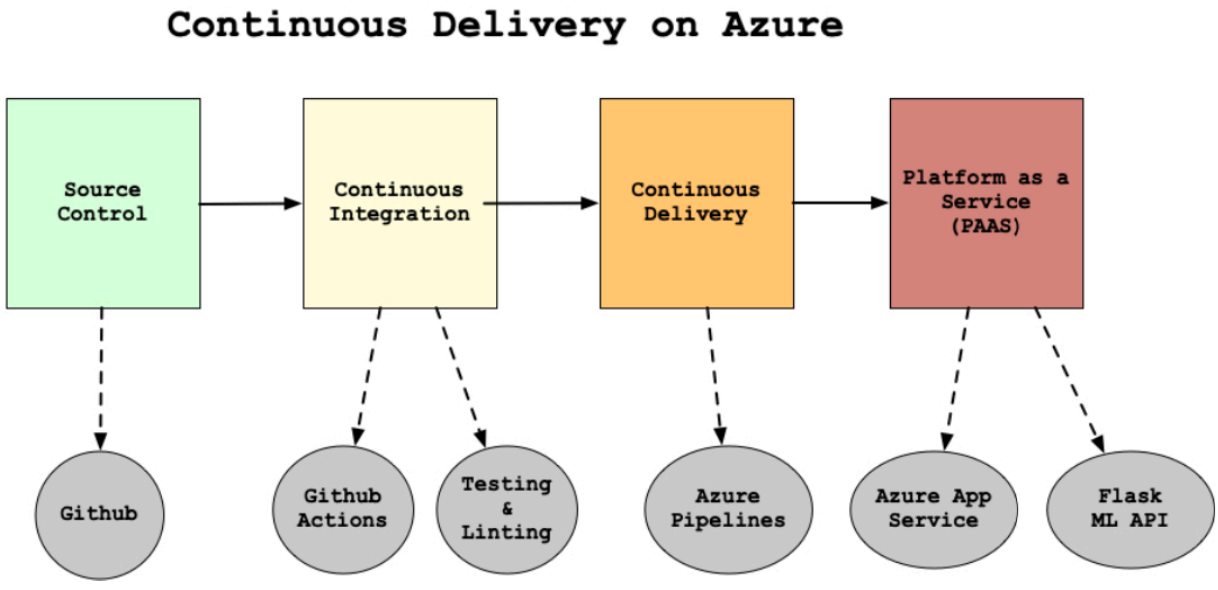
- Commit changes and push to Github
- Verify Github Actions Test Software
- Run project in Azure Shell

Later you could expand this initial setup to allow for an exact continuous delivery workflow. This initial project could be the starter kit to deploy the code to an Azure PaaS.



continuous-delivery

One way to imagine this is a sequence of steps with branches, as shown.



continuous-delivery-project-azure