

Accelerate your workflow from local Python prototype to the cloud

Savannah Ostrowski
PyGotham TV

👋 Hey! I'm Savannah.

Senior Product Manager @ Microsoft

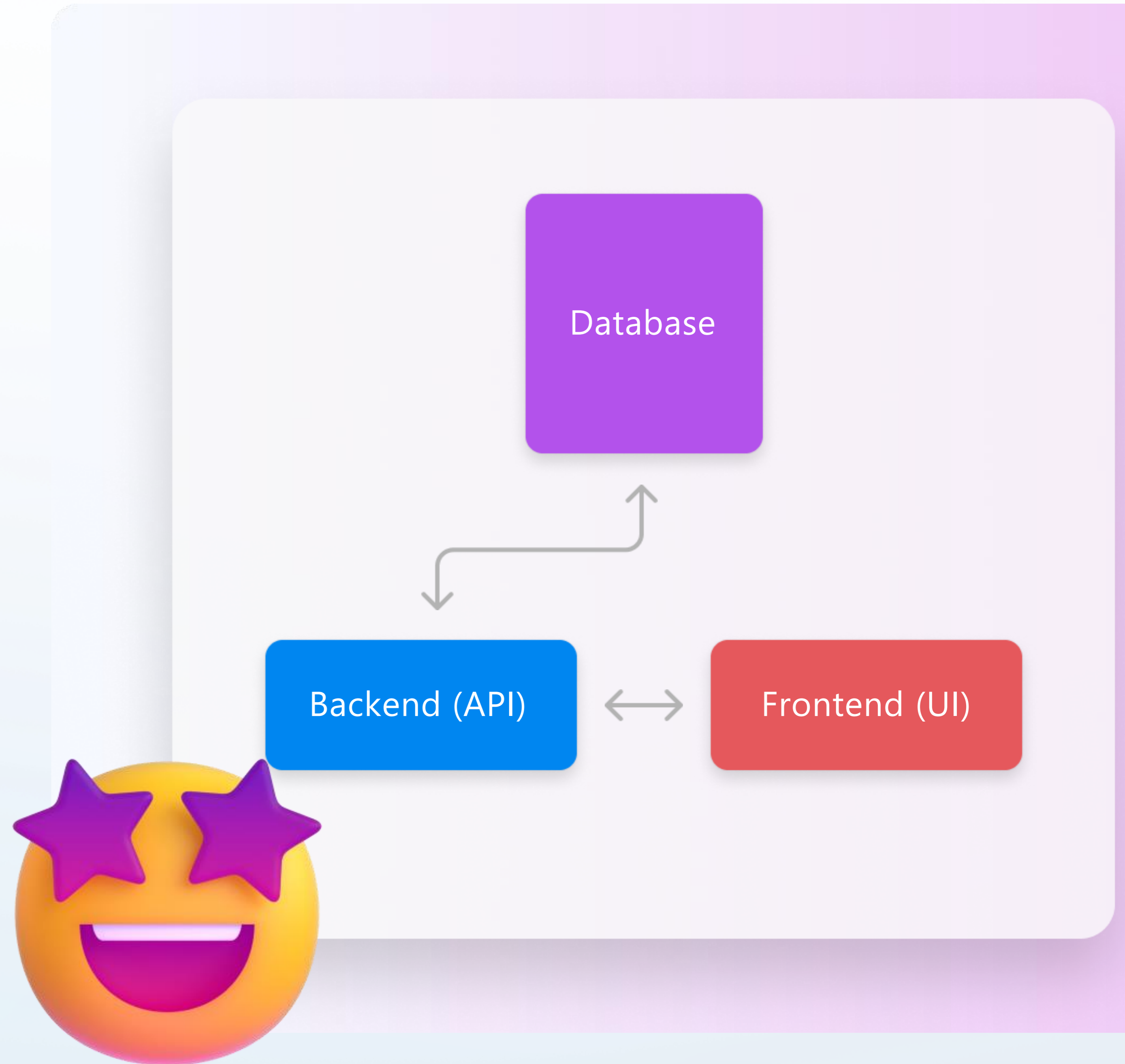
Currently: Product Lead for the Azure Developer CLI (azd)

Previously: Product Manager for Pylance/Python dev experience

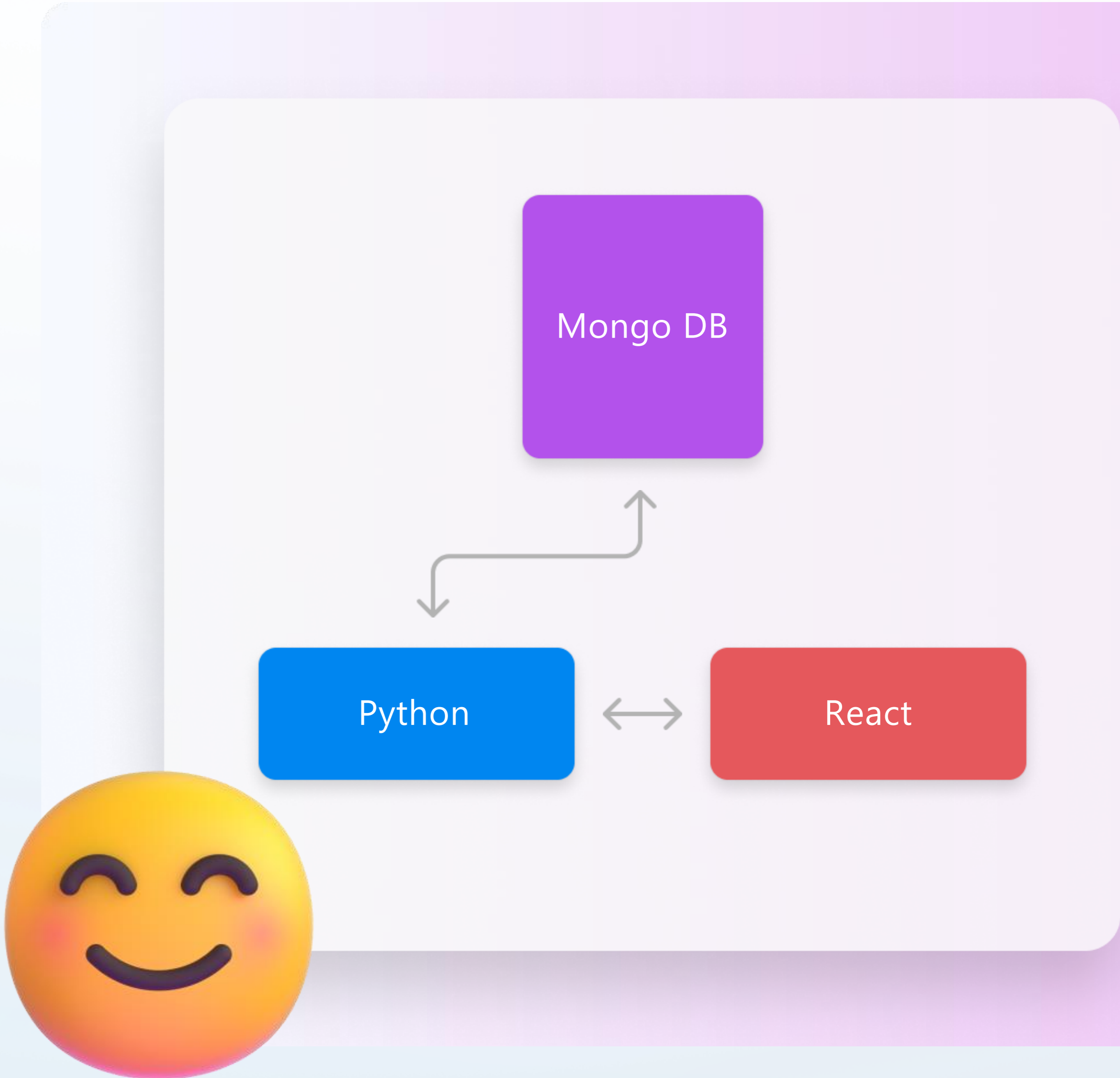
Before that: Software Engineer at startups/consultancies



From idea...



**...to local
development**



**...to the
cloud???**



What does your app look like on Azure?

A flavor of doing things the “right” way

Build — Resources and Considerations

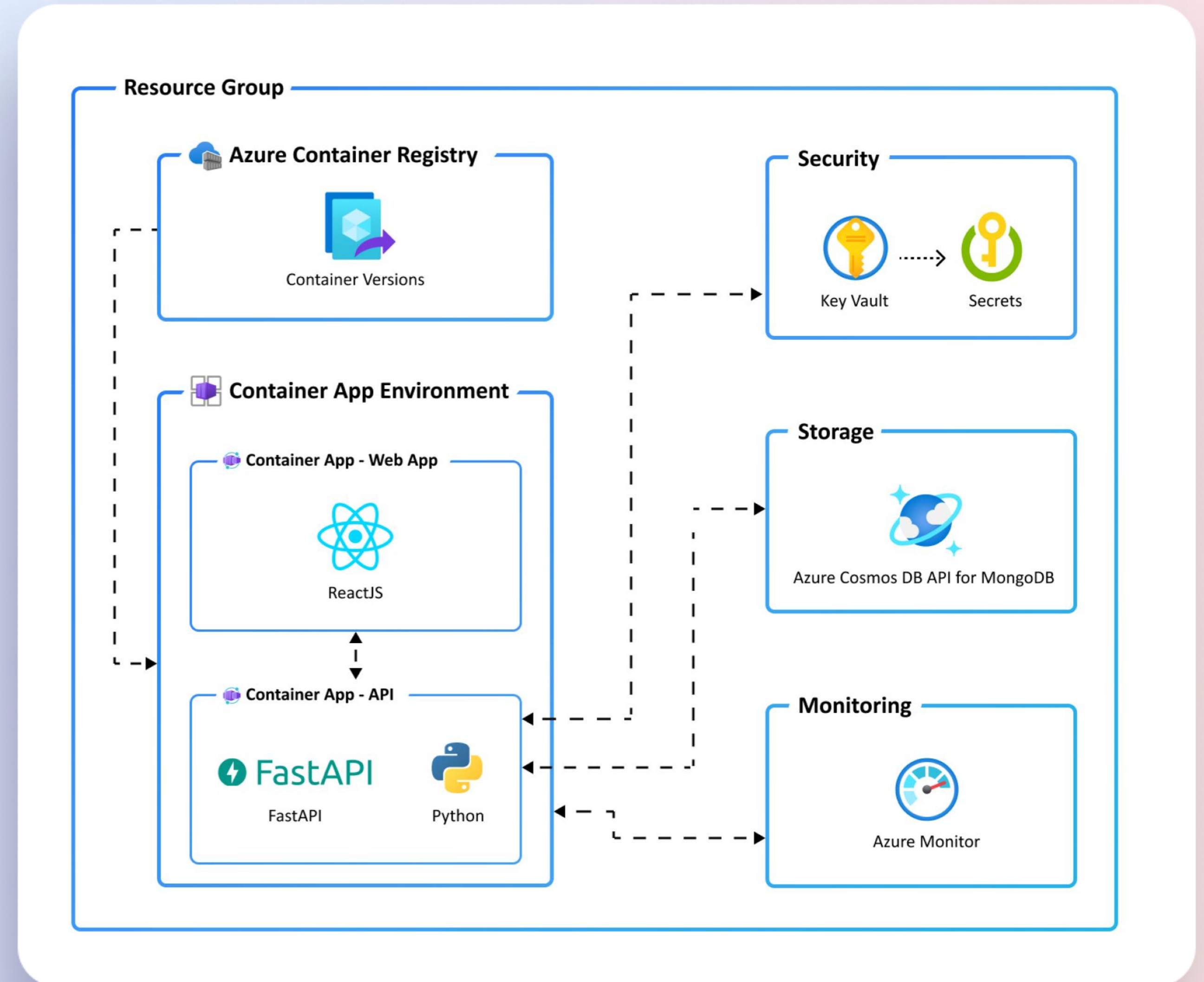
- Container Apps Environment (1)
- Container Registry (1)
- Container App (2)
- Key Vault (1)
- Azure Cosmos DB for MongoDB (1)
- Configuration of infrastructure services
- Configuration of permissions and roles
- Local dev support for working with resources

Monitoring

- Log Analytics Workspace (1)
- Application Insights (1)
- Portal dashboard (1)

Test and Release

- CI/CD pipeline via GitHub Actions



**That's a lot to think
about and manage!**

Do cloud things the right way with the Azure Developer CLI (azd)

From local development environment to the cloud in a single step

```
azd init  
azd up  
azd monitor  
azd pipeline config
```

azd commands



Your app



azd up



The cloud

Build on application templates infused with best practices

Idiomatic and extensible templates make going code to cloud easy

aka.ms/awesome-azd

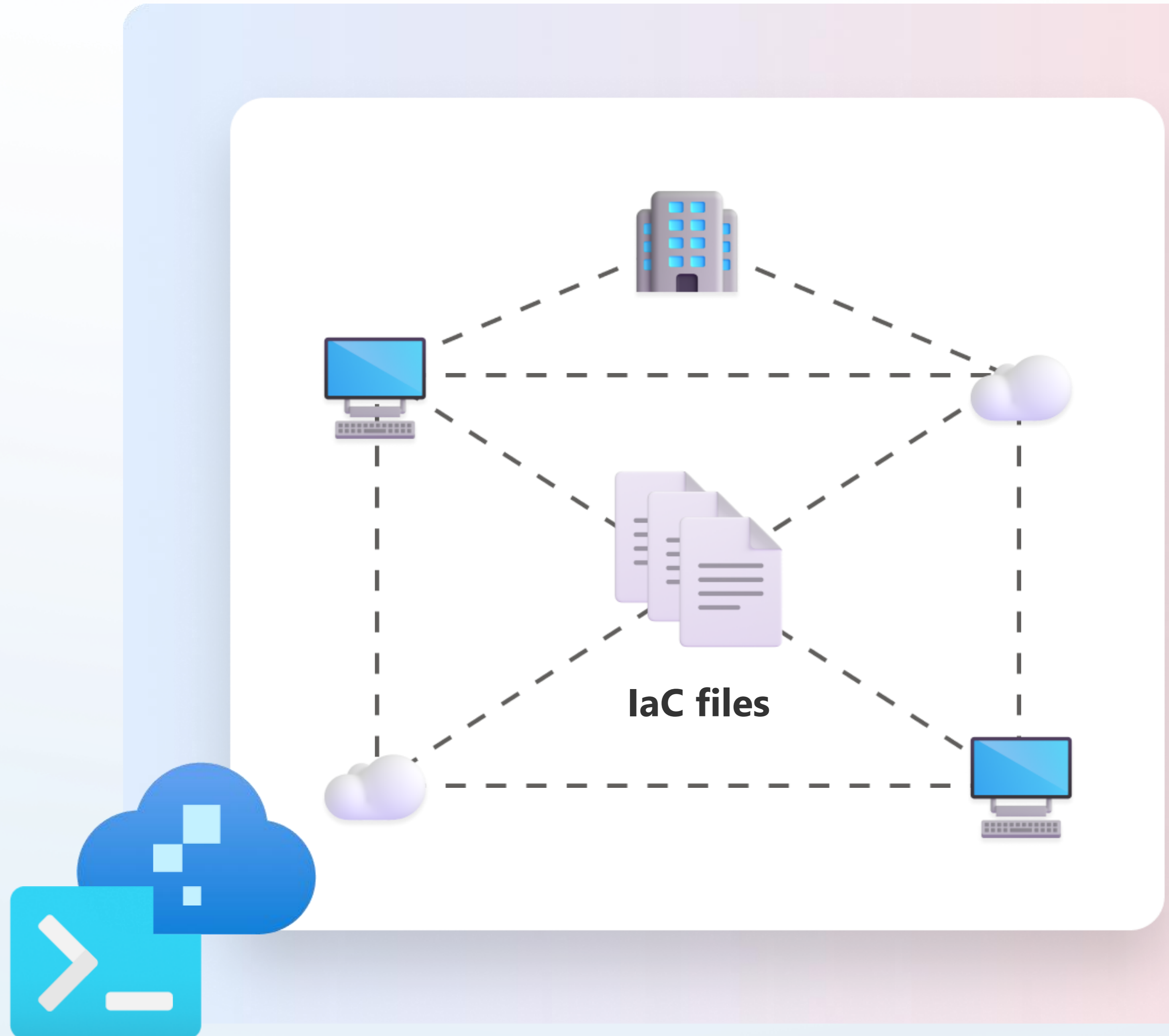
Checkout Awesome azd, a community supported library of azd compatible templates.

The screenshot shows the 'awesome-azd' website's 'Template Library'. The page features a search bar and a grid of templates. A modal window is open for the 'ToDo Application - Java MongoDB AppService' template, which is Microsoft Authored. The modal displays the following information:

- Author:** Azure Dev
- Description:** A complete sample To Do application that demonstrates how to build an Azure solution using React.js for the Web application, Java for the API,...
- Tags:** .NET, C#, Azure App Insights, Azure Functions, Tag, Azure Log Analytics, Azure Health Data Service, FHIR Service, Bicep, Additional Tag, CosmosDB
- Quick Use:** `azd init -t Azure-Samples/azure-health`

Infrastructure as Code-centric

Use declarative IaC for repeatability and reusability



todo-python-mongo-aca Public template

forked from Azure-Samples/todo-python-mongo-aca

main 1 branch 0 tags Use this template

This branch is up to date with Azure-Samples/todo-python-mongo-aca:main. Contribute Sync fork

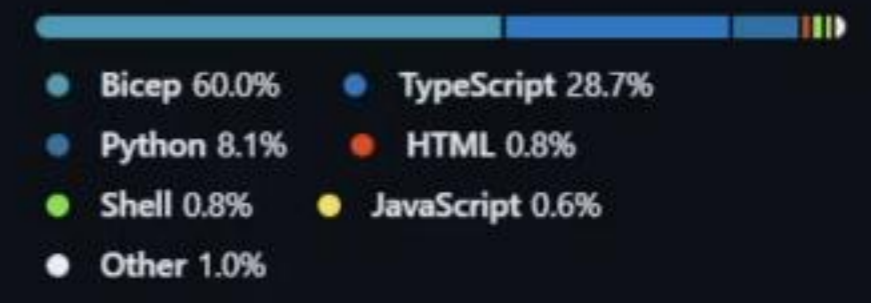
azure-sdk Synchronize repo from Repoman ae8e502 last week 31 commits

.azdo/pipelines	Synchronize repo from Repoman	3 months ago
.devcontainer	Synchronize repo from Repoman	last month
.github/workflows	Synchronize repo from Repoman	last month
.vscode	Synchronize repo from Repoman	3 months ago
assets	Synchronize repo from Repoman	7 months ago
infra	Synchronize repo from Repoman	last week
src	Synchronize repo from Repoman	last week
tests	Synchronize repo from Repoman	4 months ago
.gitattributes	Synchronize repo from Repoman	last year
.gitignore	Create template Azure-Samples/todo-python-mongo-aca	last year
LICENSE	Create template Azure-Samples/todo-python-mongo-aca	last year
NOTICE.txt	Create template Azure-Samples/todo-python-mongo-aca	last year
OPTIONAL_FEATURES.md	Synchronize repo from Repoman	3 months ago
README.md	Synchronize repo from Repoman	last month
azure.yaml	Synchronize repo from Repoman	3 months ago
openapi.yaml	Synchronize repo from Repoman	4 months ago

A blueprint for getting a React.js web app with Python (FastAPI) API and a MongoDB API in Cosmos database onto Azure. The frontend, currently a ToDo application, is designed as a placeholder that can easily be removed and replaced with your own frontend code.

- MIT license
- Activity
- 0 stars
- 0 watching
- 13 forks

Languages



Suggested Workflows

Based on your tech stack

Actions Importer Set up

Automatically convert CI/CD files to YAML for GitHub Actions.

Work

EXPLORER

OPEN EDITORS

env azure/pygotham

TODO-PYTHON-MONGO-ACA [...]

.azdo

.azure

pygotham

env

config.json

config.json

.devcontainer

.github

.vscode

assets

infra

src

tests

.gitattributes

.gitignore

azure.yaml

LICENSE

NOTICE.txt

openapi.yaml

OPTIONAL_FEATURES.md

README.md

.azure > pygotham > .env

```
1 API_CORS_ACA_URL="https://ca-web-jmlf2efuczbcu.gentlesmoke-060d99a6.eastus2.azurecontainerapps.io"
2 APPLICATIONINSIGHTS_CONNECTION_STRING="InstrumentationKey=6e80909c-b438-4632-84f6-324528610c7f;IngestionEndpoint=https://eastus2-3.in.applicationinsights.
3 APPLICATIONINSIGHTS_NAME="appi-jmlf2efuczbcu"
4 AZURE_CONTAINER_ENVIRONMENT_NAME="cae-jmlf2efuczbcu"
5 AZURE_CONTAINER_REGISTRY_ENDPOINT="crjmlf2efuczbcu.azurecr.io"
6 AZURE_CONTAINER_REGISTRY_NAME="crjmlf2efuczbcu"
7 AZURE_COSMOS_CONNECTION_STRING_KEY="AZURE-COSMOS-CONNECTION-STRING"
8 AZURE_COSMOS_DATABASE_NAME="Todo"
9 AZURE_ENV_NAME="pygotham"
10 AZURE_KEY_VAULT_ENDPOINT="https://kv-jmlf2efuczbcu.vault.azure.net/"
11 AZURE_KEY_VAULT_NAME="kv-jmlf2efuczbcu"
12 AZURE_LOCATION="eastus2"
13 AZURE_SUBSCRIPTION_ID="a7e458e8-b7b0-4662-b79e-512ca957fadf"
14 AZURE_TENANT_ID="52e77deb-6eca-45cb-872c-7f01fa1b0947"
15 REACT_APP_API_BASE_URL="https://ca-api-jmlf2efuczbcu.gentlesmoke-060d99a6.eastus2.azurecontainerapps.io"
16 REACT_APP_APPLICATIONINSIGHTS_CONNECTION_STRING="InstrumentationKey=6e80909c-b438-4632-84f6-324528610c7f;IngestionEndpoint=https://eastus2-3.in.applicatio
17 REACT_APP_WEB_BASE_URL="https://ca-web-jmlf2efuczbcu.gentlesmoke-060d99a6.eastus2.azurecontainerapps.io"
18 SERVICE_API_ENDPOINTS="[]"
19 SERVICE_API_IMAGE_NAME="crjmlf2efuczbcu.azurecr.io/todo-python-mongo-aca/api-pygotham:azd-deploy-1692150475"
20 SERVICE_API_NAME="ca-api-jmlf2efuczbcu"
21 SERVICE_API_RESOURCE_EXISTS="false"
22 SERVICE_WEB_IMAGE_NAME="crjmlf2efuczbcu.azurecr.io/todo-python-mongo-aca/web-pygotham:azd-deploy-1692150477"
23 SERVICE_WEB_NAME="ca-web-jmlf2efuczbcu"
24 SERVICE_WEB_RESOURCE_EXISTS="false"
25 USE_APIM="false"
26
```

OUTLINE

TIMELINE

EXPLORER azure-dev.yml 2

OPEN EDITORS

.github > workflows > azure-dev.yml

azure-dev.yml .github... 2

TODO-PYTHON-MONGO-ACA [WSL...]

.azdo

.azure

.devcontainer

.github / workflows

azure-dev.yml 2

.vscode

assets

infra

src

tests

.gitattributes

.gitignore

! azure.yaml

LICENSE

NOTICE.txt

! openapi.yaml

OPTIONAL_FEATURES.md

README.md

```

1  on:
2    workflow_dispatch:
3    push:
4      # Run when commits are pushed to mainline branch (main or master)
5      # Set this to the mainline branch you are using
6      branches:
7        - main
8        - master
9
10 # GitHub Actions workflow to deploy to Azure using azd
11 # To configure required secrets for connecting to Azure, simply run `azd pipeline config`
12
13 # Set up permissions for deploying with secretless Azure federated credentials
14 # https://learn.microsoft.com/en-us/azure/developer/github/connect-from-azure?tabs=azure-portal%2CLinux#set-up-azure-login-with-openid-connect-auth
15 permissions:
16   id-token: write
17   contents: read
18
19 jobs:
20   build:
21     runs-on: ubuntu-latest
22     env:
23       AZURE_CLIENT_ID: ${vars.AZURE_CLIENT_ID}
24       AZURE_TENANT_ID: ${vars.AZURE_TENANT_ID}
25       AZURE_SUBSCRIPTION_ID: ${vars.AZURE_SUBSCRIPTION_ID}
26       AZURE_CREDENTIALS: ${secrets.AZURE_CREDENTIALS}
27     steps:
28     - name: Checkout
29       uses: actions/checkout@v3
30
31     - name: Install azd
32       uses: Azure/setup-azd@v0.1.0
33
34     - name: Log in with Azure (Federated Credentials)
35       if: ${env.AZURE_CLIENT_ID != ''}
36       run: |
37         azd auth login `
38         --client-id "$Env:AZURE_CLIENT_ID" `
39         --federated-credential-provider "github" `
40         --tenant-id "$Env:AZURE_TENANT_ID"
41     shell: pwsh

```

Code to Cloud with Azure Developer CLI

All-in-one with azd up!

Package, provision and deploy as a single step with the azd up command!



DISCOVER

What do I want to build?

Initialize your own repository or leverage an existing idiomatic template to get started.

Each template includes real app code, local development support, Infrastructure as Code assets, and more!

Find tons of azd compatible templates at: <https://azure.github.io/awesome-azd/>

```
$ azd template list
$ azd init
```



INFRASTRUCTURE

How do I create & connect services for my application?

Provision the right resources to run your application on Azure.

Every template includes IaC files written in Bicep or Terraform.

```
$ azd provision
```



DEPLOY

How do I get my application running in the cloud?

Package and deploy your application code to Azure in minutes.

```
$ azd deploy
```



BEST PRACTICES

How do I infuse best practices into my workflow?

Monitor your app's health with usage, performance and reliability metrics and dashboards.

Set up CI/CD to run against real Azure resources on every commit to the repo.

```
$ azd monitor
$ azd pipeline config
```

```
$ azd up
```

So maybe I've convinced you that

the cloud is cool!

(when you have the right tools to support your workflow)



Ready to get started?

Resources for this talk

github.com/savannahostrowski/pygotham

Azure Developer CLI links

- Install: aka.ms/azd-install
- Docs: aka.ms/azd
- GitHub: github.com/azure/azure-dev
- Community standups: aka.ms/azd-standups
- Template gallery: aka.ms/azd-python-templates

