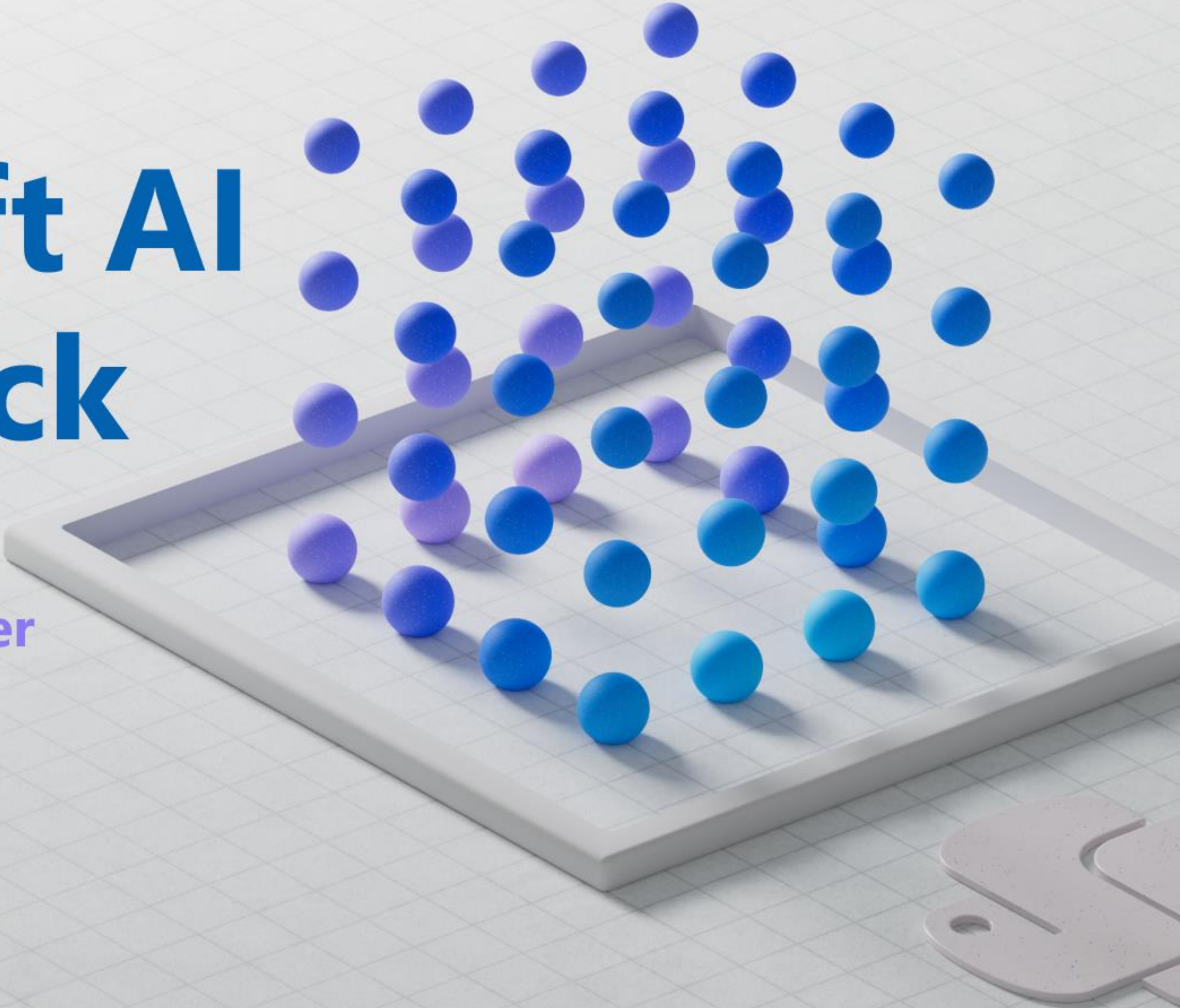
January 29th - February 12th

The Microsoft AI Chat App Hack



Build, innovate, and #HackTogether

aka.ms/hacktogether/chatapp

The AI Chat App Hack

Microsoft

January 29th - February 12th

29th ▶
Building a RAG Chat App in Python

30th ▶
Customizing your RAG Chat App

31st ▶
Azure AI Search Best Practices

1st ▶
GPT-4 with Vision

2nd
HACK
HACK
HACK

3rd
HACK
HACK

4th
HACK
HACK
HACK

5th ▶
AM: RAG Chat Web Components
PM: Access Control in RAG Chat Apps

6th ▶
Evaluating a RAG Chat App

7th ▶
RAG Chat Special Topic

8th ▶
Continuous Deployment of your Chat App

9th
HACK
HACK

10th
HACK

11th
HACK
HACK

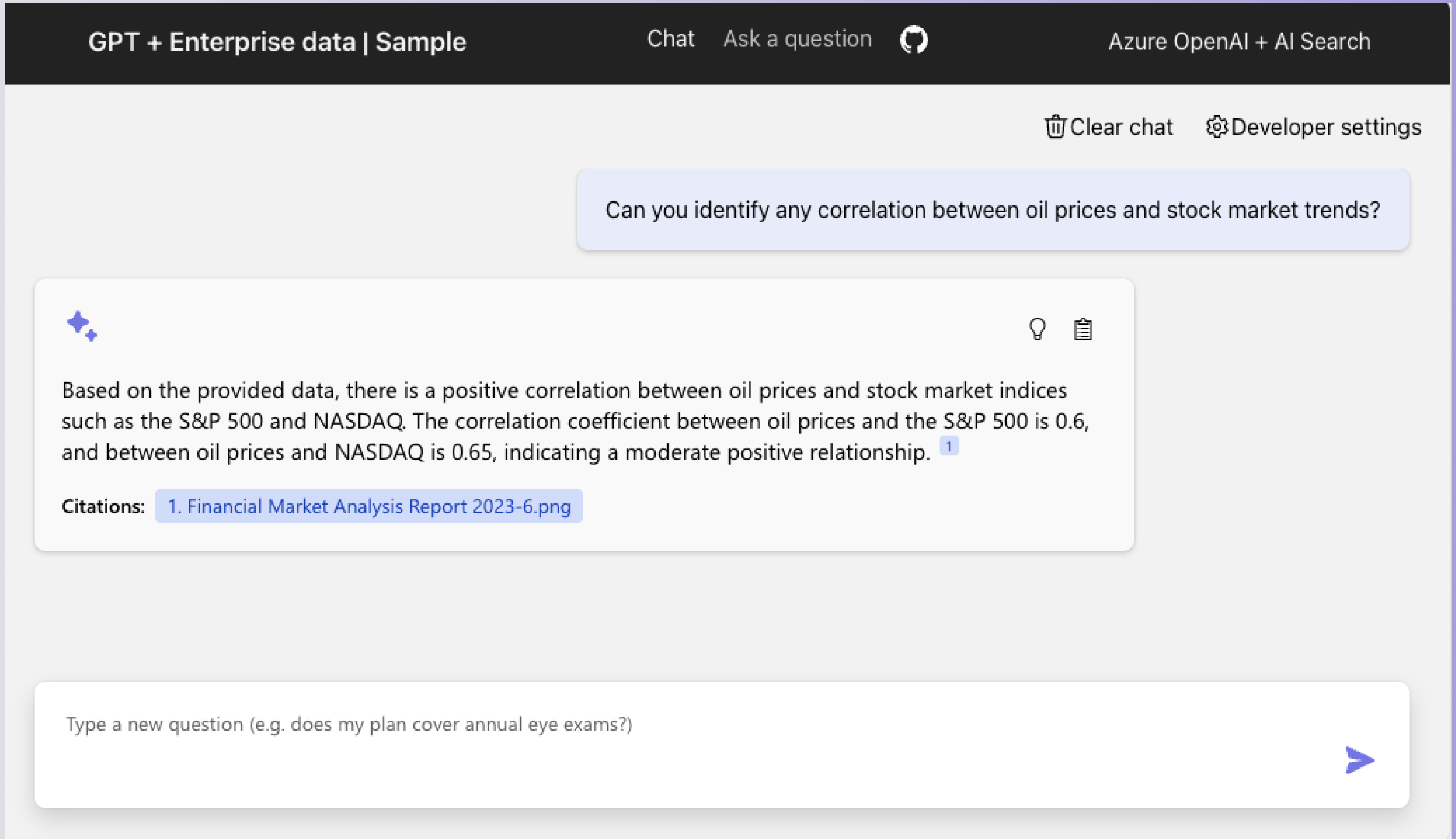
12th
SUBMIT YOUR PROJECT

Build, innovate, and #HackTogether

Continuous Deployment of RAG Chat Apps

Microsoft

RAG chat app



Can you identify any correlation between oil prices and stock market trends?

Based on the provided data, there is a positive correlation between oil prices and stock market indices such as the S&P 500 and NASDAQ. The correlation coefficient between oil prices and the S&P 500 is 0.6, and between oil prices and NASDAQ is 0.65, indicating a moderate positive relationship. [1]

Citations: 1. Financial Market Analysis Report 2023-6.png

Type a new question (e.g. does my plan cover annual eye exams?)

Search results

+

Chat Completion

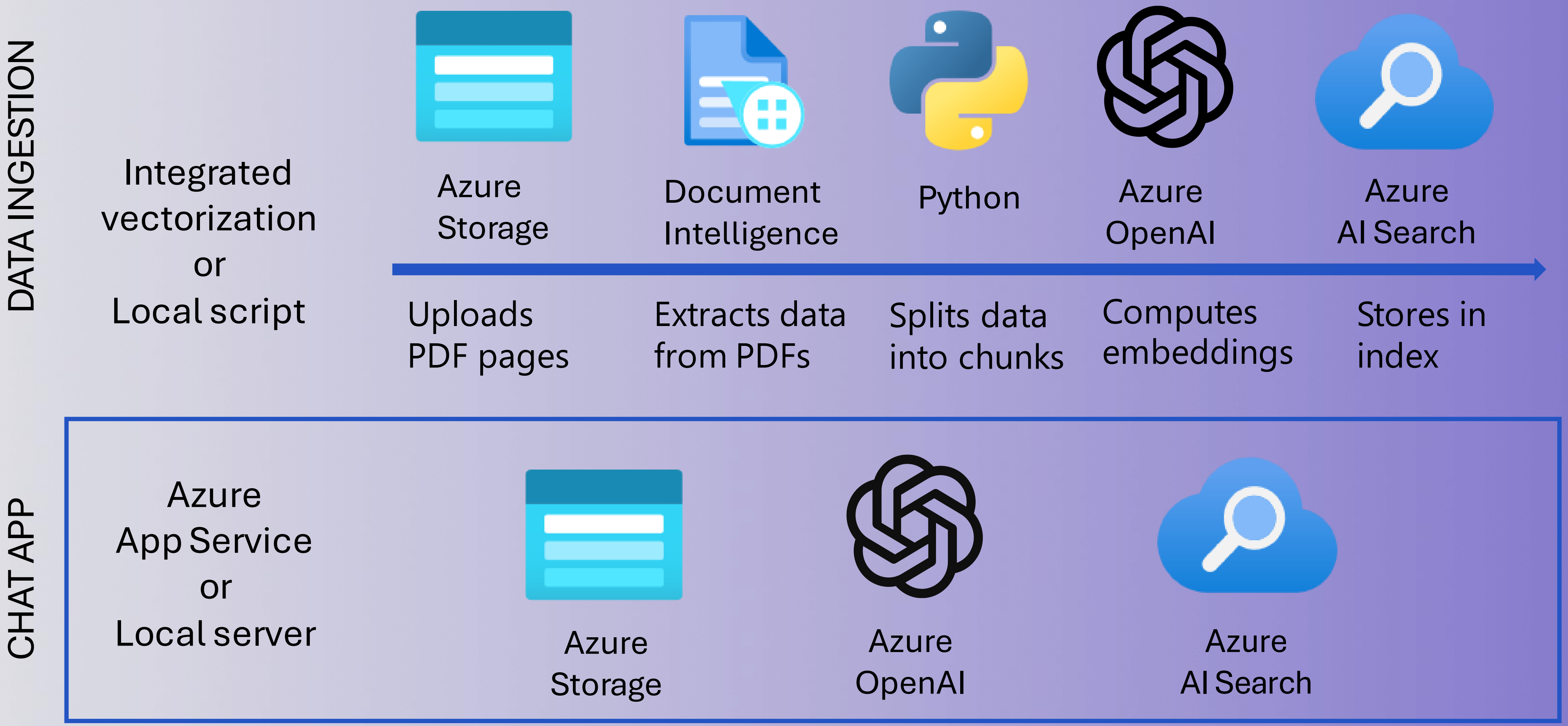
=

Answer with citations

https://github.com/Azure-Samples/azure-search-openai-demo/ aka.ms/ragchat

Microsoft

Azure architecture for RAG chat app



DATA INGESTION

Integrated vectorization
or
Local script

Azure Storage — Uploads PDF pages

Document Intelligence — Extracts data from PDFs

Python — Splits data into chunks

Azure OpenAI — Computes embeddings

Azure AI Search — Stores in index

CHAT APP

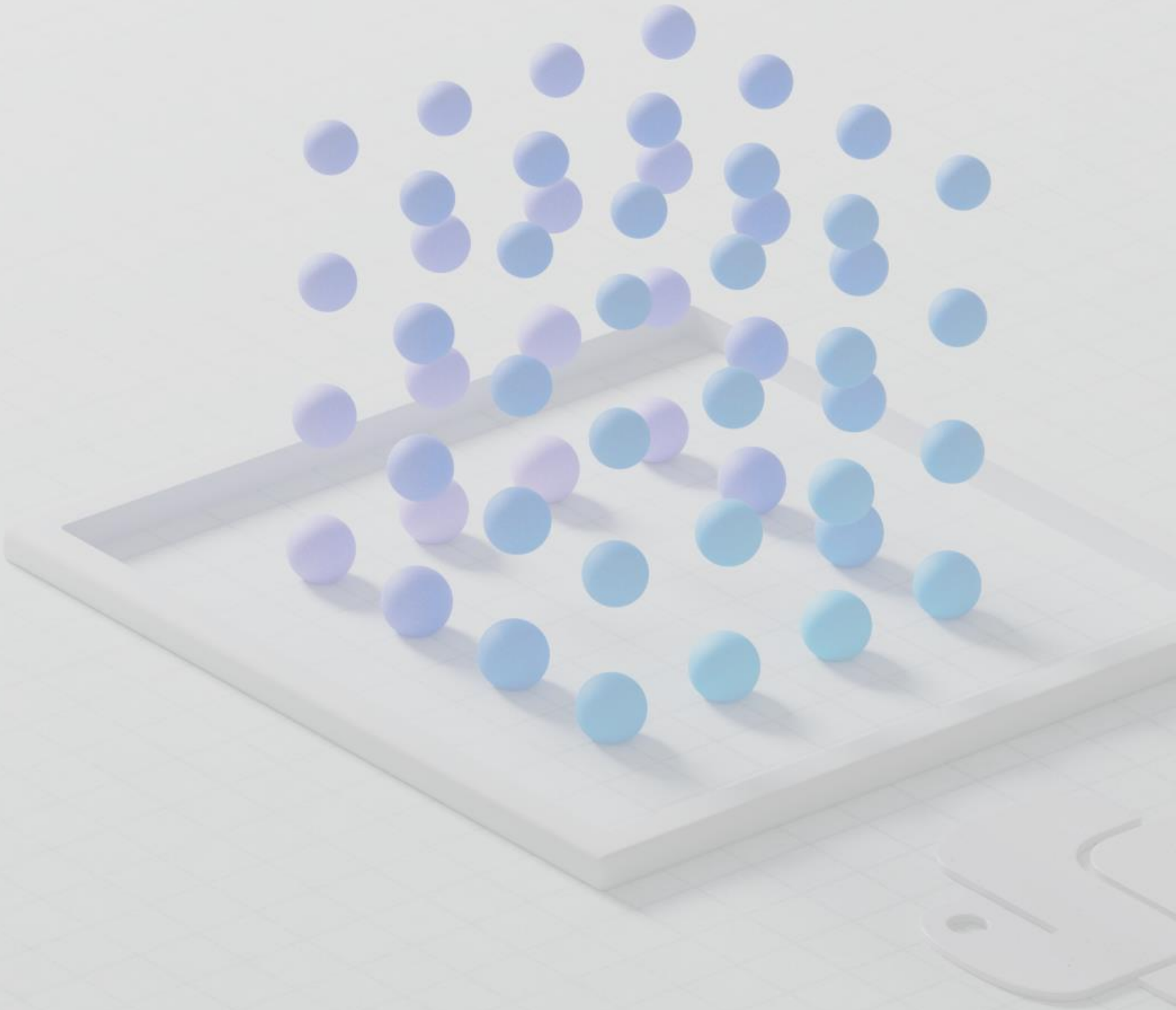
Azure App Service
or
Local server

Azure Storage

Azure OpenAI

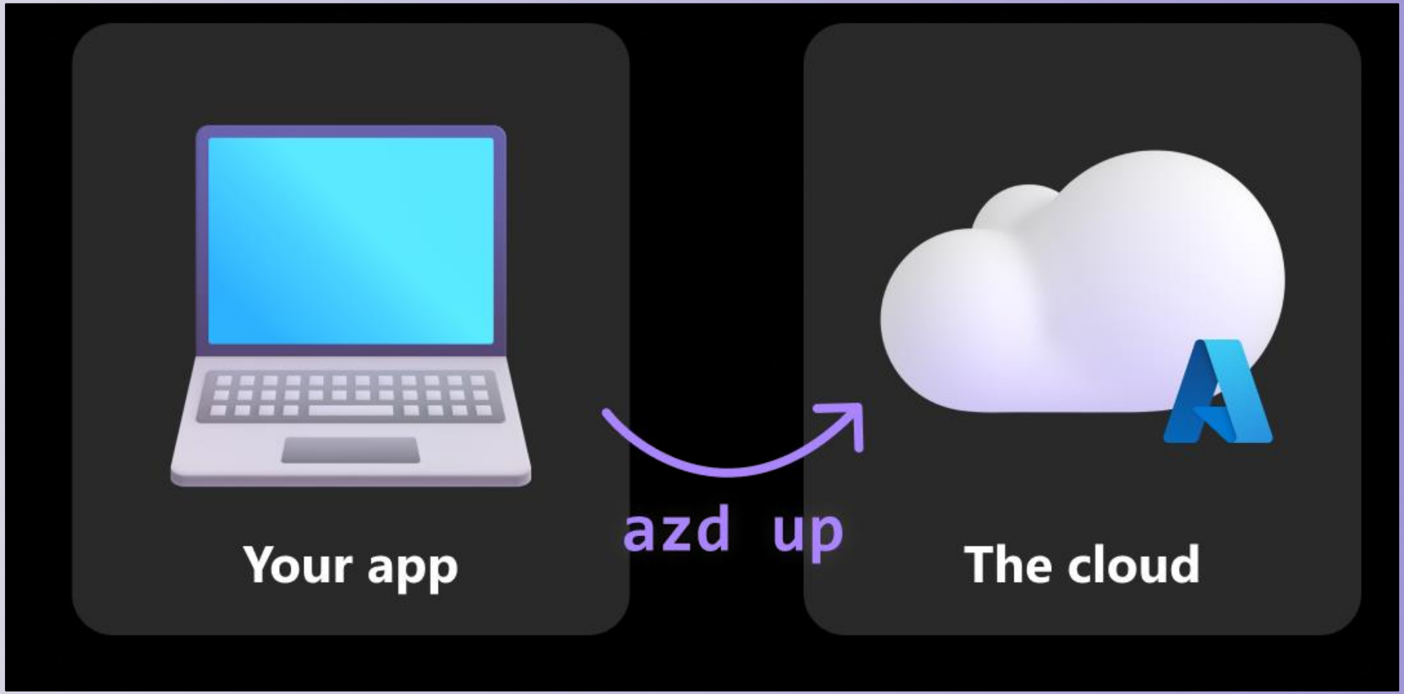
Azure AI Search

Manual deployment



Deploying with azd

Use **`azd`** command (Azure Developer CLI) to easily get an app up on Azure.

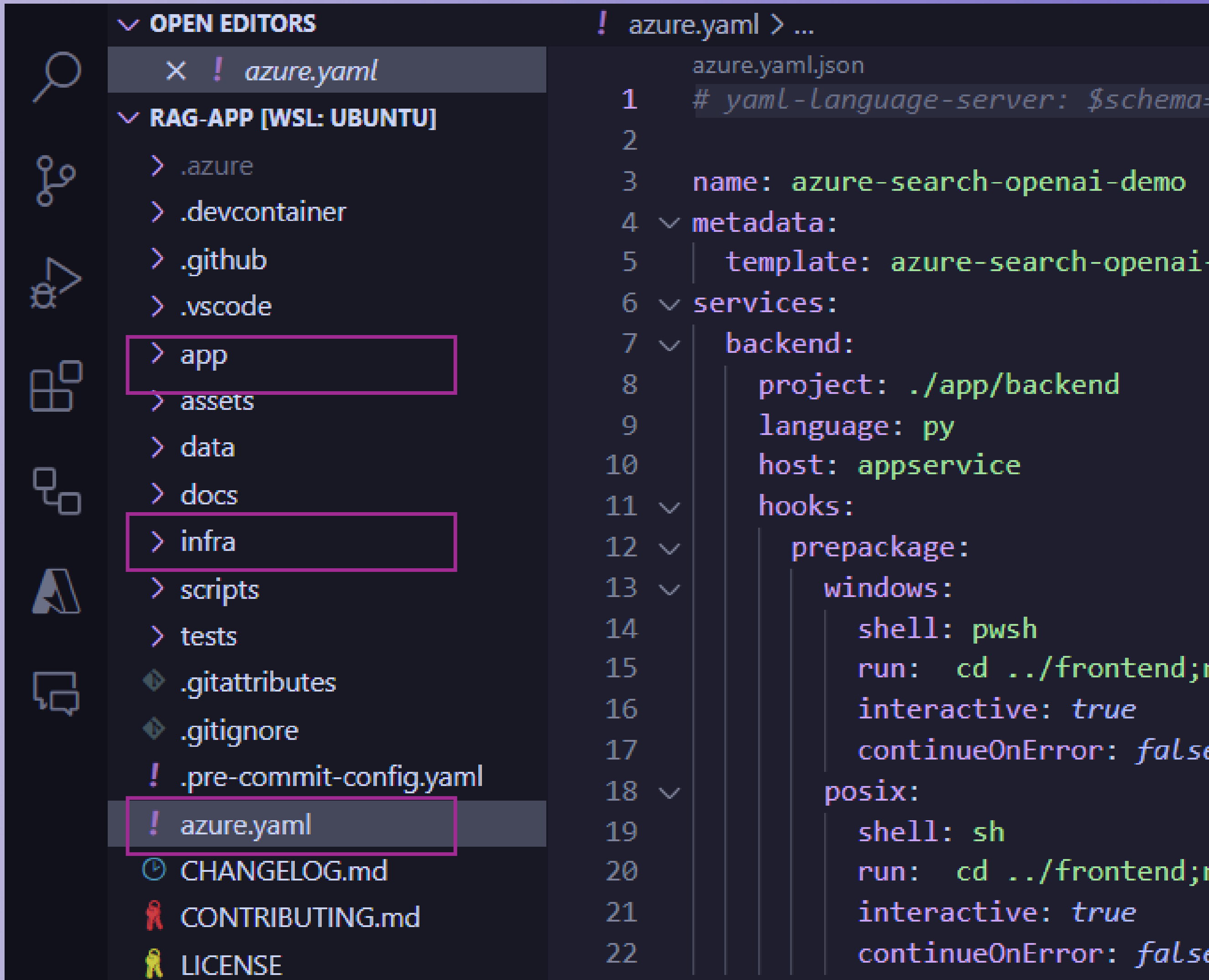


Project setup for azd

app/ with web application code

Bicep files to provision Azure resources

azure.yaml to specify the azd project and app metadata



```yaml
# yaml-language-server: $schema=

name: azure-search-openai-demo
metadata:
  template: azure-search-openai-
services:
  backend:
    project: ./app/backend
    language: py
    host: appservice
    hooks:
      prepackage:
        windows:
          shell: pwsh
          run:  cd ../frontend;
          interactive: true
          continueOnError: fals
        posix:
          shell: sh
          run:  cd ../frontend;
          interactive: true
          continueOnError: fals
```

azd commands

Provision resources

```
azd provision
```

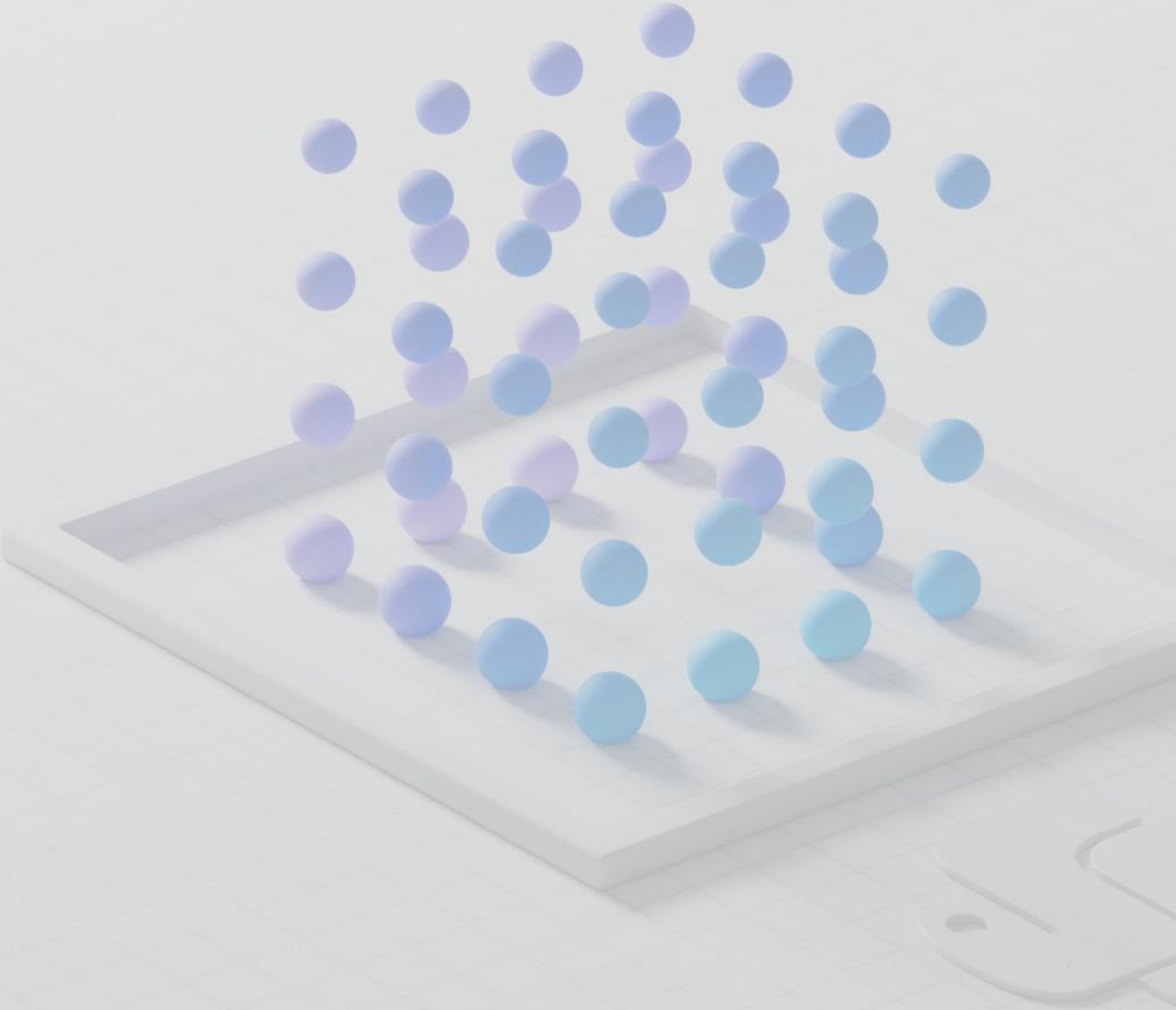
Deploy application code:

```
azd deploy
```

Do both!

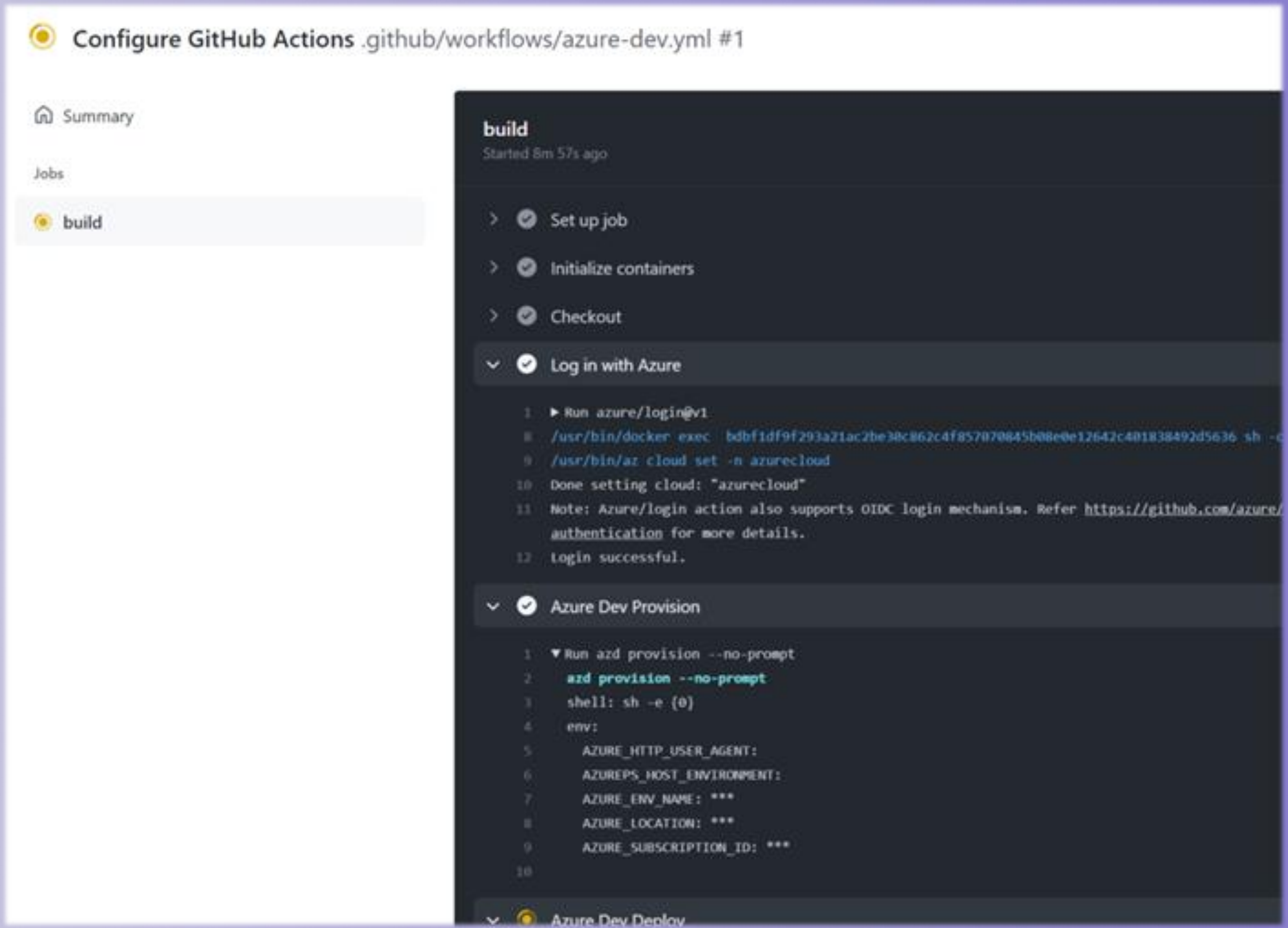
```
azd up
```

Continuous deployment



Goal

Deploy to Azure on push,
using a CI platform
like GitHub Actions
or Azure DevOps



Configure GitHub Actions .github/workflows/azure-dev.yml #1

```
▶ Run azure/login@v1
/usr/bin/docker exec  bdbf1df9f293a21ac2be30c862c4f857070845b08e0e12642c401838492d5636 sh -c
/usr/bin/az cloud set -n azurecloud
Done setting cloud: "azurecloud"
Note: Azure/login action also supports OIDC login mechanism. Refer https://github.com/azure/
authentication for more details.
Login successful.
```

```
▼ Run azd provision --no-prompt
  azd provision --no-prompt
  shell: sh -e {0}
  env:
    AZURE_HTTP_USER_AGENT:
    AZUREPS_HOST_ENVIRONMENT:
    AZURE_ENV_NAME: ***
    AZURE_LOCATION: ***
    AZURE_SUBSCRIPTION_ID: ***
```

https://github.com/Azure-Samples/azure-search-openai-demo/pull/1083

aka.ms/ragchat/workflow

Setting up a GitHub workflow

Add workflow file for GitHub actions:

```
.github/workflows/azure-dev.yaml
```

Make sure all necessary variables/secrets are declared in azure.yaml:

```
pipeline:
  variables:
      - AZURE_OPENAI_SERVICE …
```

Run command to configure all necessary Actions variables:

```
azd pipeline config
```

Customizing azd environment variables

If you previously did…

```
azd env set VAR_NAME value
```

azure.yaml knows which ones to sync and azure-dev.yaml knows about existing environment variables.

When you set any azd env variables, re-run to sync new values:

```
azd pipeline config
```

Adding new azd environment variables

Add them to your workflow env:

```
env:
      AZURE_SUBSCRIPTION_ID: $(AZURE_SUBSCRIPTION_ID)
      NEW_VAR_NAME: $(NEW_VAR_NAME)
```

Add to azure.yaml:

```
pipeline:
  variables:
      - NEW_VAR_NAME …
```

When you change any azd env variables, re-run:

```
azd pipeline config
```

Using continuous deployment today

Either merge this PR or wait for it to get merged next week:

https://github.com/Azure-Samples/azure-search-openai-demo/pull/1083

aka.ms/ragchat/workflow

Then configure the workflow:

```
azd pipeline config
```

Next steps

- Register for the hackathon → aka.ms/hacktogether/chatapp
- Introduce yourself in our discussion forum
- Deploy the repo with the sample data
  - See steps on low cost deployment → aka.ms/ragchat/free
- Hack, hack, hack, hack! 👨🏿 💻 🧑🏻‍💻 👩🏼‍💻
- Post in forum if you have any questions. 🙋🏼‍♀️ 🙋🏾‍♀️ 🙋🏻‍♀️
- Submit your project before February 12th to win prizes! 💸 💰 💵
- Join tomorrow’s sessions from MVPs!