Empowering Youth with Career Recommender using Azure Kubernetes Service, Azure ML, Azure Service Bus, Azure SQL

Problem Statement

We will build a **Career Recommendation engine using Text Data and Azure Kubernetes Service**. To demonstrate this, we would use a case study approach and build a recommendation engine for a **non-profit organization** Career Village**.**

CareerVillage.org is a non-profit that crowdsources career advice for **underserved youth**. The U.S. has almost 500 students for every guidance counsellor. *Underserved youth lack the network to find their career role models*, making CareerVillage.org the only option for millions of young people in America and around the globe with nowhere else to turn.

To date, 25,000 volunteers have created profiles and opted in to receive emails when a career question is a good fit for them. To help students get the advice they need, the team at CareerVillage.org needs to be able to send the right questions to the right volunteers. The notifications sent to volunteers seem to have the greatest impact on how many questions are answered.

We will use the following

- Questions asked by the students
- Answers provided by the professionals and the professionals details

When a student asks a question, we would find similar questions which have been answered. Then we would connect the student question with the professional so that the professional can answer the question. In the user interface, we would also **display the top ten questions and answers** which have the highest similarity with the question asked.

Question:

Recommendations

I want to be a data scientist

Question	Answer	Similarity
I want to be a data scientist, what online courses should I take ? #datascience	You should search for Algorithm videos. Usually when studying data, you would need to know about databases structure, analytics skills, and some other logics. Another thing you could do would be start analyzing some small real cases like how long does it take to go from your house to the supermarket and what you could do to reduce the time? or how often do you drink water (time gap between each occurence). How could you track that? and how could you improve it? is it good? these are a few examples on how you could analyze stuff.	0.838
I want to be a data scientist, what online courses should I take	Hello Chong G. I am not a data scientist, but I think I can give you some advice on this. Nowadays, an increasing number of professions are requiring analytics capabilities. There are some core things you should learn to handle great amount of data, like:	0.838

Question:

Recommendations

I want to be a carpenter

Question	Answer	Similarity
#college What would I have to do over there? What would I learn?	Congratulations on being interested in becoming a carpenter. It takes a special person to enter this field and meet the demands which this career area presents. The first step is to get to know yourself to see if you share the personality traits which make carpenters successful. The next step is doing networking to meet and talk to and possibly shadow carpenters to see if this is something that you really want to do, as a career area could look much different on the inside than it looks from the outside.	0.718
Hi , My name is Angela and i go to job corps . i was	<p>Hi Angela, </p><p> </p><p>Congratulations on taking a look at going into the trades as a rewarding and fulfilling career! The trades are experiencing a critical shortage of talent entering those fields and have largely been under-valued in our society over the last several decades. The	0.604

Question:

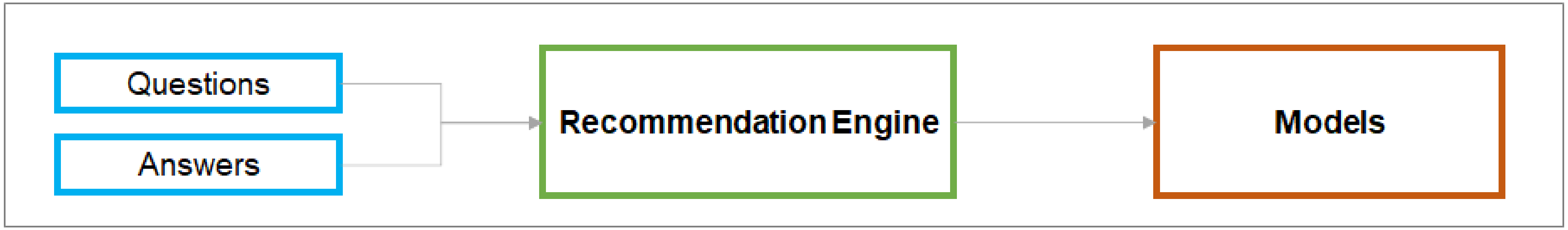
Recommendations

I want to be a nice person

Question	Answer	Similarity
do you get a long with others #social-work	Hi Brandie, I assume you are asking in the professional sense. Getting along with colleagues is something people have to work on everyday. Someone is more easy-going, while other people are more formal. What I'm trying to say is, that you can't get along with everyone around you 100% of the time. A conflict can be beneficial if handled in a constructive way. I personally learnt a lot in the field of professional tension and conflict. Even working relationships are built on compromise. The thing I work hard to remember every time, is to stay as cool as possible, analyze, <u>communicate </u>and learn for the next time. Have great day Zuzana	0.596
I want to become the best person i	Self-reflection is the best way to grow as a person. First, identify what it is that you want to improve. Then ask yourself why do you want to improve or change in that area. After you answer the 'what' and 'why', you'll be able to	0.36

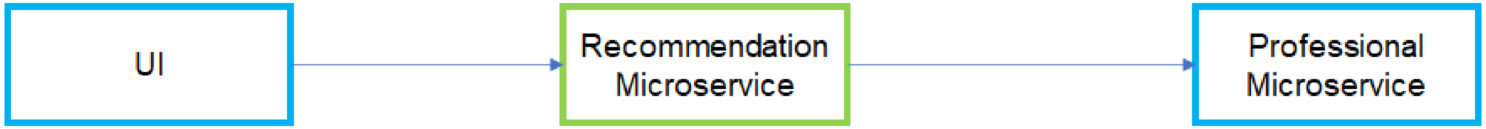
Architecture

The recommender models are being built in **Azure ML**. The input to the Recommendation Engine is Questions and Answers for Careers and the Recommendation Engine produces the Models.



The solution is deployed with **3** main components

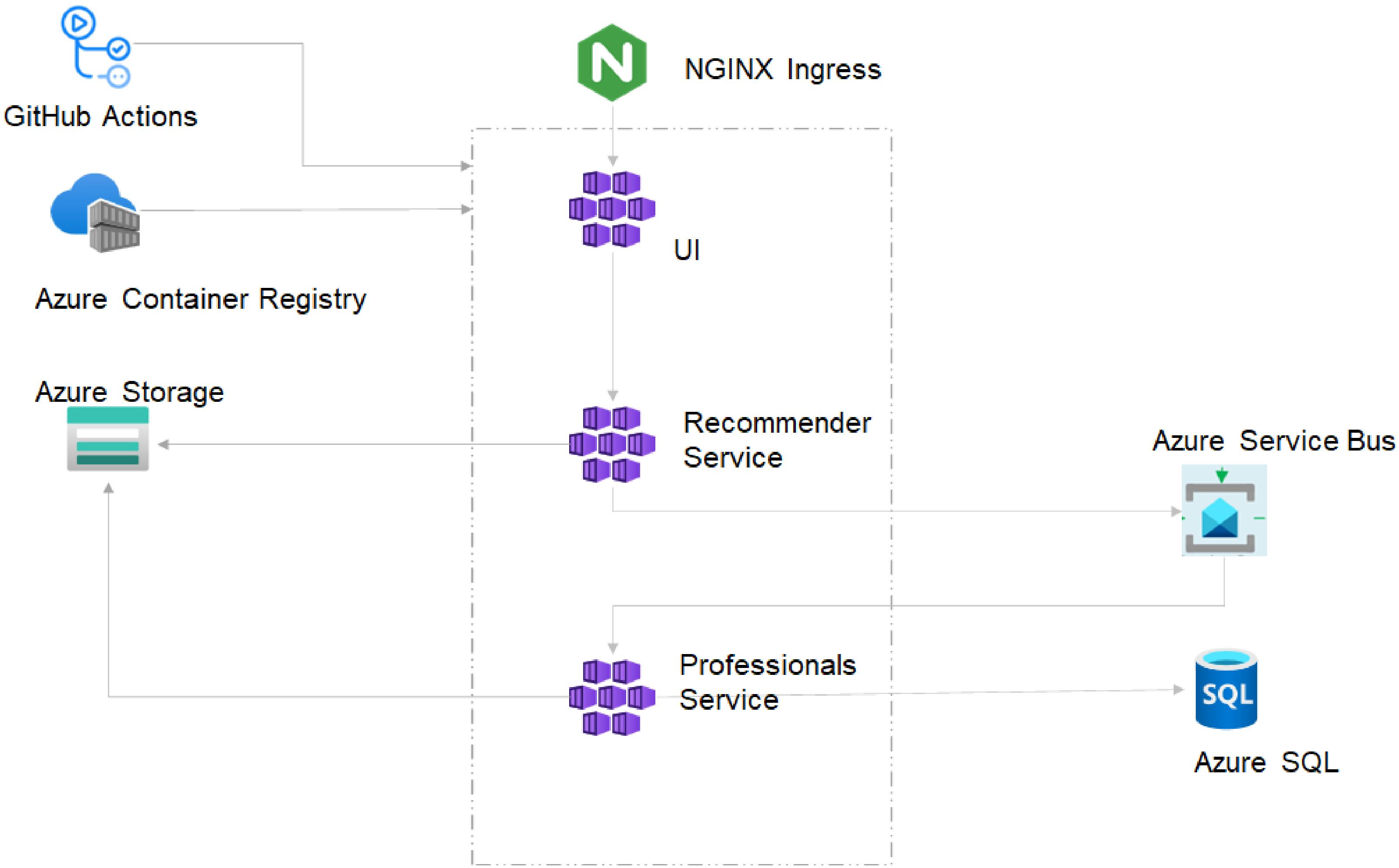
1. **UI**
2. **Recommendation Microservice** using the **Models**
3. **Professionals Microservice** connecting the Recommendations with the Professionals



Data Flow

1. The student puts the career question in the UI
2. The question is used by the Recommendation microservice to generate the recommendations and pass back the previously generated recommended answers to the UI.
3. The Professional microservice finds the appropriate professionals to answer the questions.

Architecture in Azure



1. Ingress implemented with **NGINX**
2. The Front end, the Back End [the Recommender Service, the Professionals Service] is implemented as **Azure Kubernetes Service**
3. **Azure Storage** stores the questions, answers, professionals and the recommender models
4. The **Azure Service Bus** connects the Recommender Service and Professionals Service and the recommendations are exchanged thru this
5. The **Azure SQL** stores the recommendations and also the professionals who are assigned the recommendations
6. The **Azure Container Registry** stores the container images
7. **GitHub Actions** are used for Continuous Deployment

Technical Details and Implementation of solution

Recommender Model Details

Steps:

1. The questions have body and title. We make a consolidated column combining *body and the title* .
2. We make a **TF-IDF [Term Frequency Inverse Document Frequency]** vector for each of the questions text column and also of the question asked by the student.
3. We calculate the **cosine similarity** between the question asked and the consolidated list of questions. This would enable us select the top ten similarities and recommend the question to the professionals who have answered it.

TF-IDF

This defines how important a word is in a set of documents. Example for your young child , the most important word is **mom**. Example for a bar tender , important words would be related to **drinks.** A document in this case is the question text.

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)

IDF(t) = log(Total number of documents / Number of documents with term t in it).

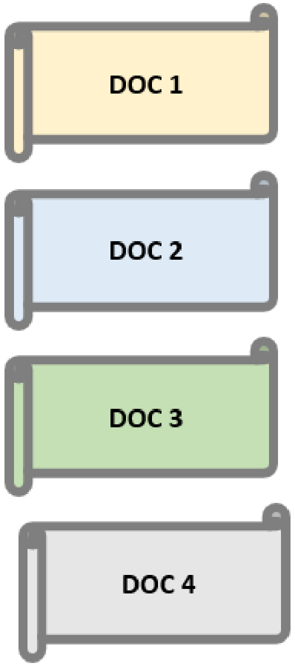
TF-IDF = TF * IDF

Example

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)

IDF(t) = log_e(Total number of documents / Number of documents with term t in it).

Value = TF * IDF



This is the first document.

This document is the second document

And this is the third one

Is this the first document?

TF(first in the DOC 1) = 1 / 5

IDF (first in the DOC 1) = log_e (4 / 2)

TF(the in the DOC 1) = 1 / 5

IDF (the in the DOC 1) = log_e (4 / 4)

TF – IDF = 0

A commonly occurring word [**the**] has a TF-IDF of zero whereas the word [**first**] has a non-zero TF-IDF.

Cosine similarity

If we have 2 vectors A and B, cosine similarity is the cosine of the angle between them. If A and B are very similar, the value is closer to 1 and if they are very dissimilar, the value is closer to zero.

Here we represent the question as vectors. The values of the vector are the TFIDF value of the various words in the question text.

Building the model in Azure ML has the following steps:

1. Create the Azure ML workspace
2. Upload data into the Azure ML Workspace
3. Create the code folder
4. Create the Compute Cluster
5. Create the Model
6. Create the Compute Environment
7. Create the Estimator
8. Create the Experiment and Run
9. Register the Model

Front End Kubernetes Service

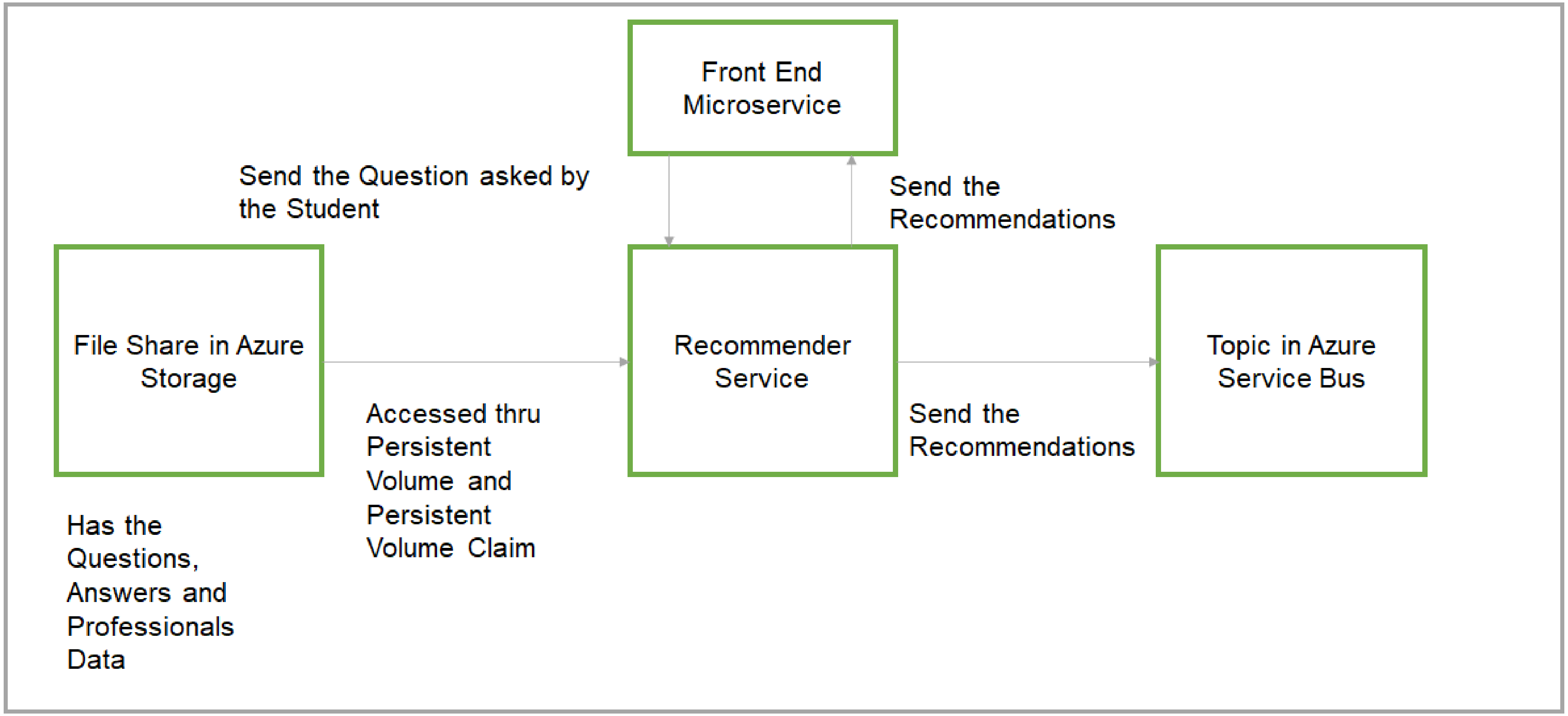
The Front-End Kubernetes Service has the UI and calls the Recommender Service. The Recommender Service URL is implemented as an Environment Variable and the Environment Variable refers to a secret.

Front End Service calling the Recommender Service

```python
url = os.environ["KUBERNETES_RECO_URL"]

@bp.route('/', methods=('GET', 'POST'))
def predict():
        if request.method == 'POST':
            q_new =  request.form.get("question")

            # defining a params dict for the parameters to be sent to the API
            PARAMS = {'questions':q_new}

            r = requests.post(url,params= PARAMS)
            # Convert JSON to DataFrame Using read_json()
            results = pd.read_json(r.text)

            return render_template('recommendations/index.html',
            allitems = list(results.values.tolist()),
            recos = True,
            question_new = q_new)

        return render_template('recommendations/index.html',
```

Recommender Kubernetes Service

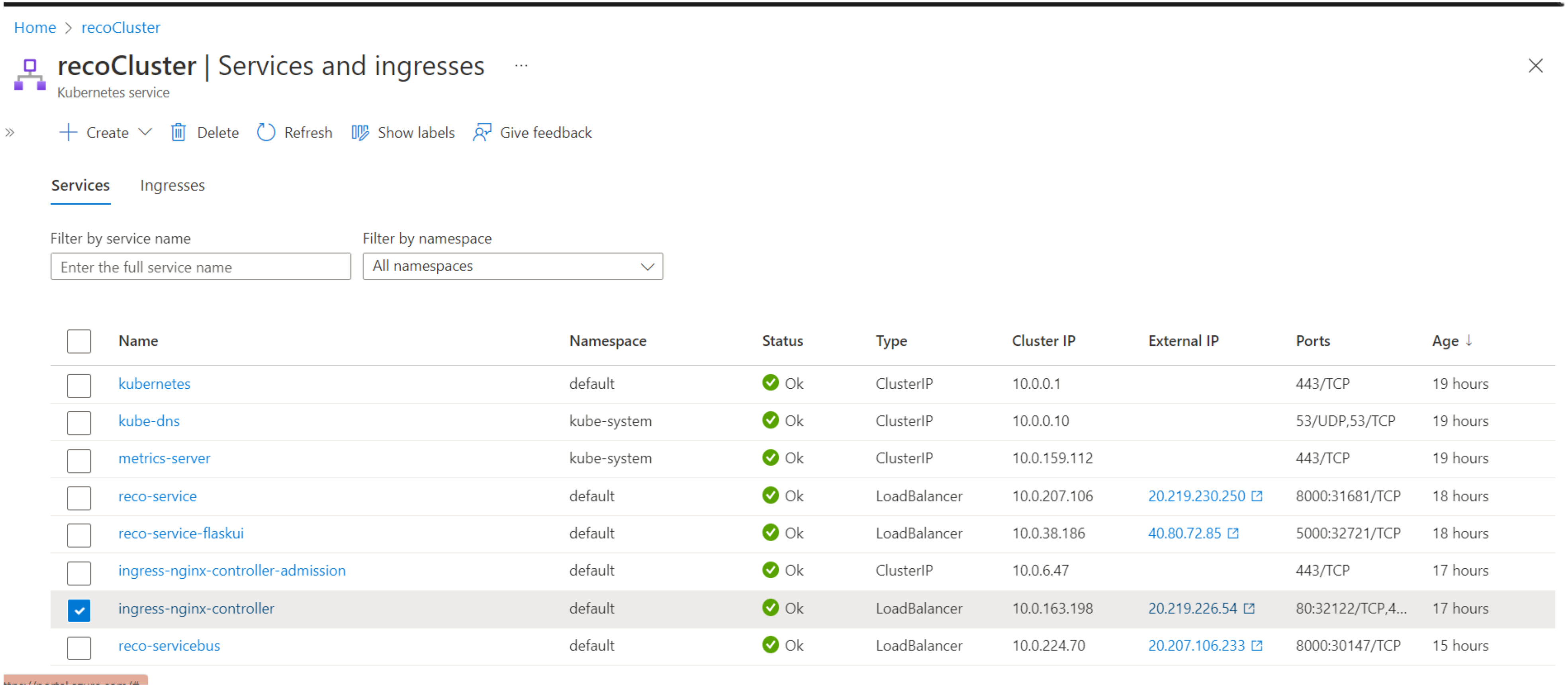
1. The service receives the Questions from the Front End Microservice
2. It uses the Models and Questions , Answers stored in the Azure Storage File Share to create the **recommendations**. The Recommender accesses the File Share with **Persistent Storage Volume and Persistent Storage Claim**.
3. The recommendations are sent to the Topic in the Azure Service Bus for the Professionals Service. The service bus URL used by the microservice is stored as a Kubernetes Secret.
4. The recommendations obtained are sent to the Front End Microservice so that they can be displayed.


Front End Microservice
Send the Question asked by the Student
Send the Recommendations
File Share in Azure Storage
Recommender Service
Topic in Azure Service Bus
Accessed thru Persistent Volume and Persistent Volume Claim
Send the Recommendations
Has the Questions, Answers and Professionals Data

Recommender service using cosine similarity for recommendations

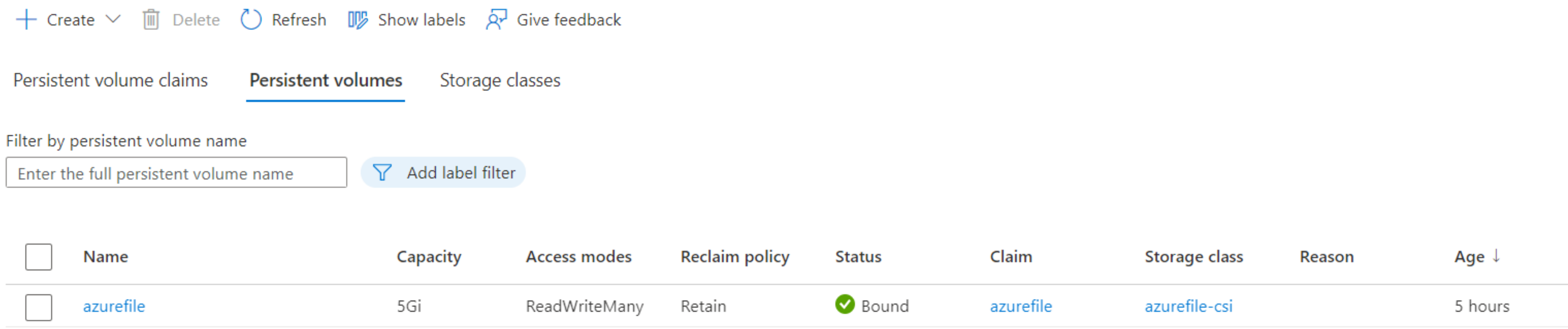
```
def get_top_n_answers(q_new):

    q_new1 = q_new
    q_new = [q_new]

    with open(CAREER_VILLAGE_PATH + 'tfidf_vectorizer.pkl', 'rb') as f:
        tfidf_vectorizer = pickle.load(f)
    with open(CAREER_VILLAGE_PATH + 'q_tfidf.pkl', 'rb') as f:
        q_tfidf = pickle.load(f)

    q_new_tfidf = tfidf_vectorizer.transform(q_new)

    result = cosine_similarity(q_new_tfidf,q_tfidf)
    result_df = pd.DataFrame(result[0], columns = ['sim'])
    q = pd.concat([questions,result_df],axis = 1)
    q = q.sort_values(by="sim",ascending = False)
```

Services

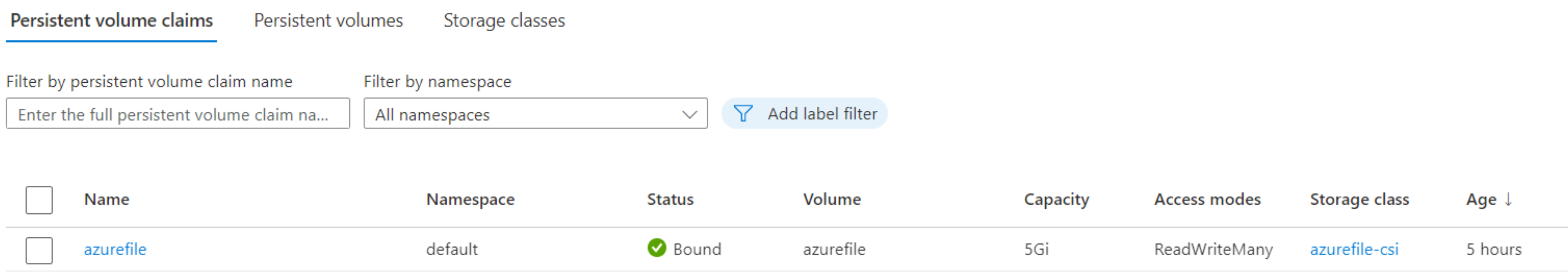


Name	Namespace	Status	Type	Cluster IP	External IP	Ports	Age ↓
kubernetes	default	Ok	ClusterIP	10.0.0.1		443/TCP	19 hours
kube-dns	kube-system	Ok	ClusterIP	10.0.0.10		53/UDP,53/TCP	19 hours
metrics-server	kube-system	Ok	ClusterIP	10.0.159.112		443/TCP	19 hours
reco-service	default	Ok	LoadBalancer	10.0.207.106	20.219.230.250	8000:31681/TCP	18 hours
reco-service-flaskui	default	Ok	LoadBalancer	10.0.38.186	40.80.72.85	5000:32721/TCP	18 hours
ingress-nginx-controller-admission	default	Ok	ClusterIP	10.0.6.47		443/TCP	17 hours
ingress-nginx-controller	default	Ok	LoadBalancer	10.0.163.198	20.219.226.54	80:32122/TCP,4...	17 hours
reco-servicebus	default	Ok	LoadBalancer	10.0.224.70	20.207.106.233	8000:30147/TCP	15 hours

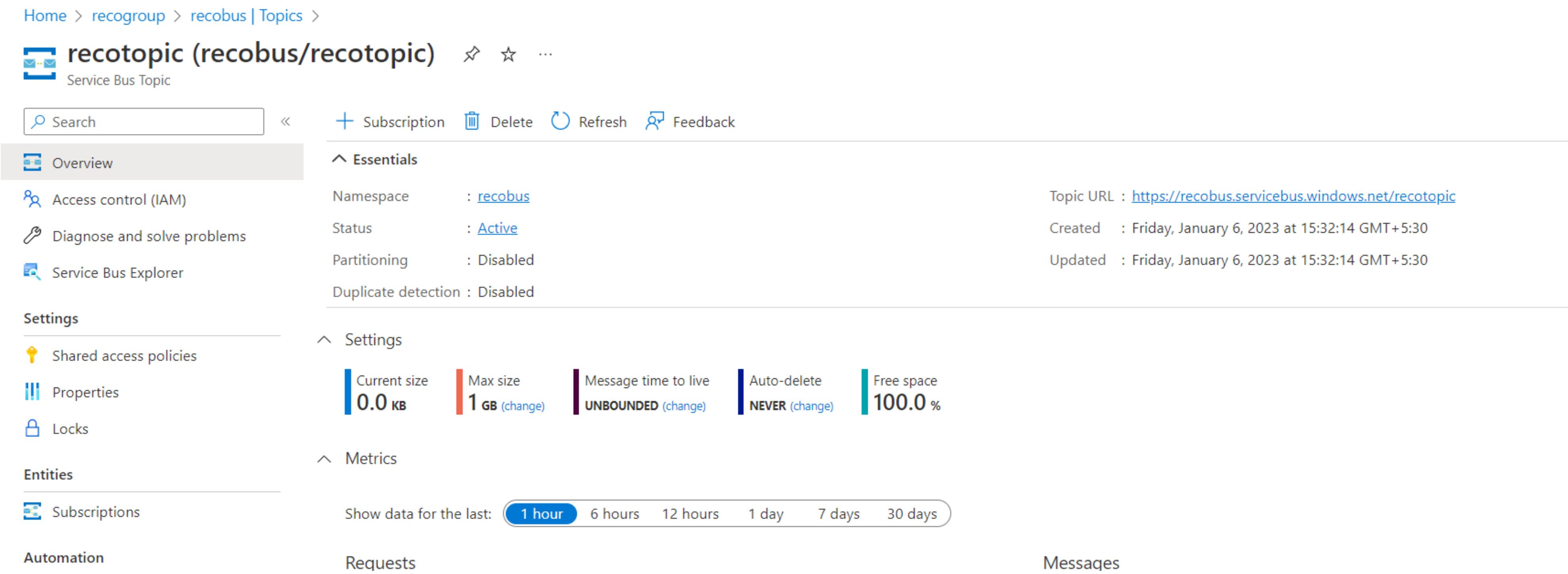
Persistent Volume



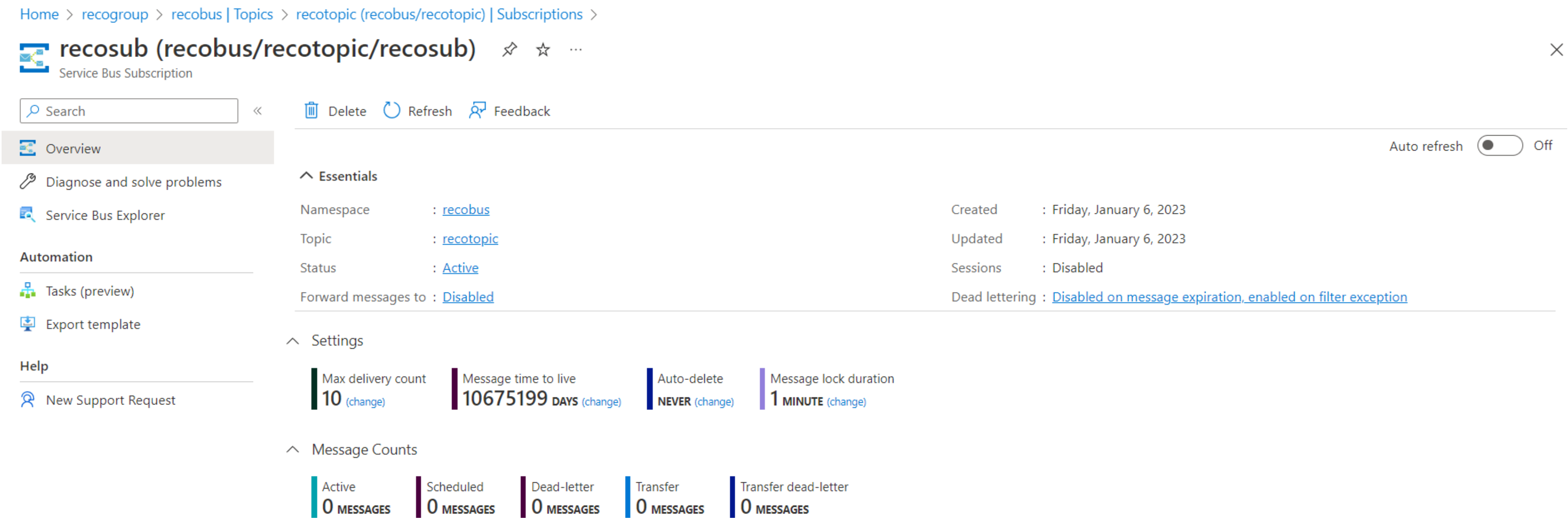
Persistent Volume Claim



Service Bus Topic



Service Bus Subscription

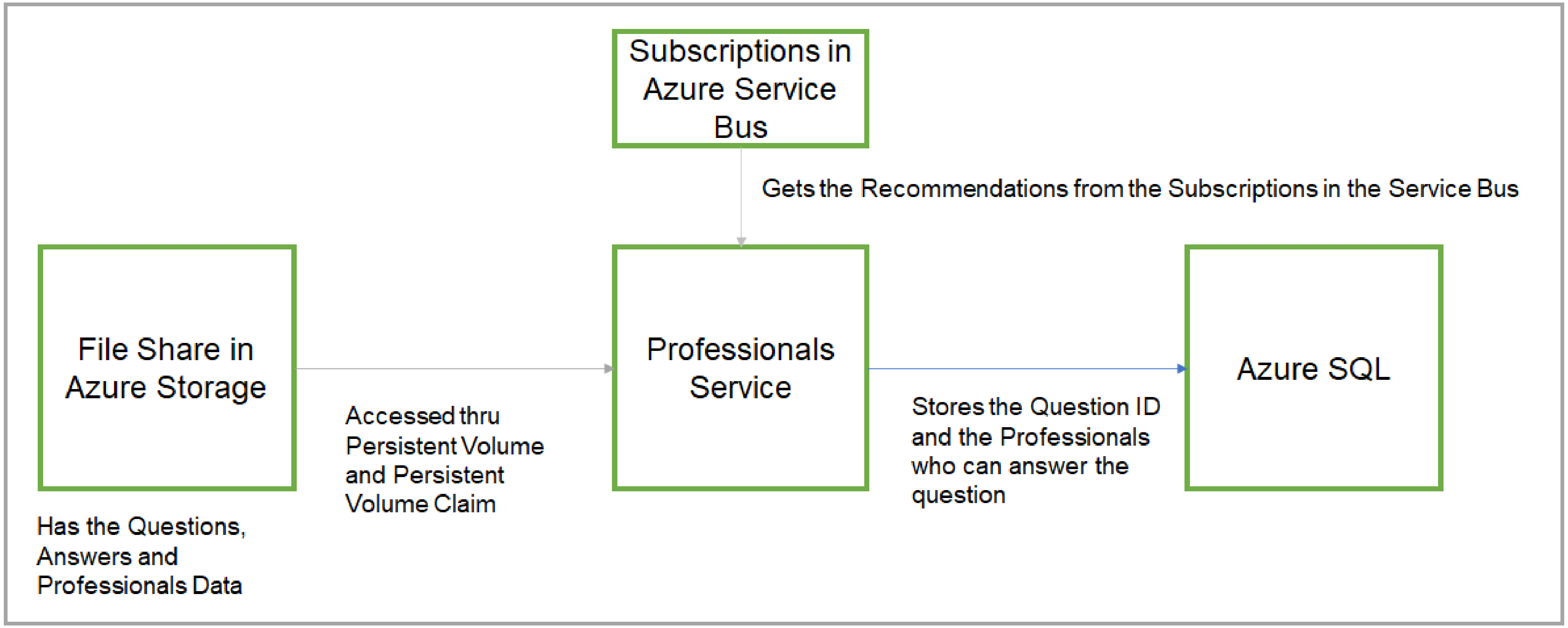


Services and Pods running in AKS

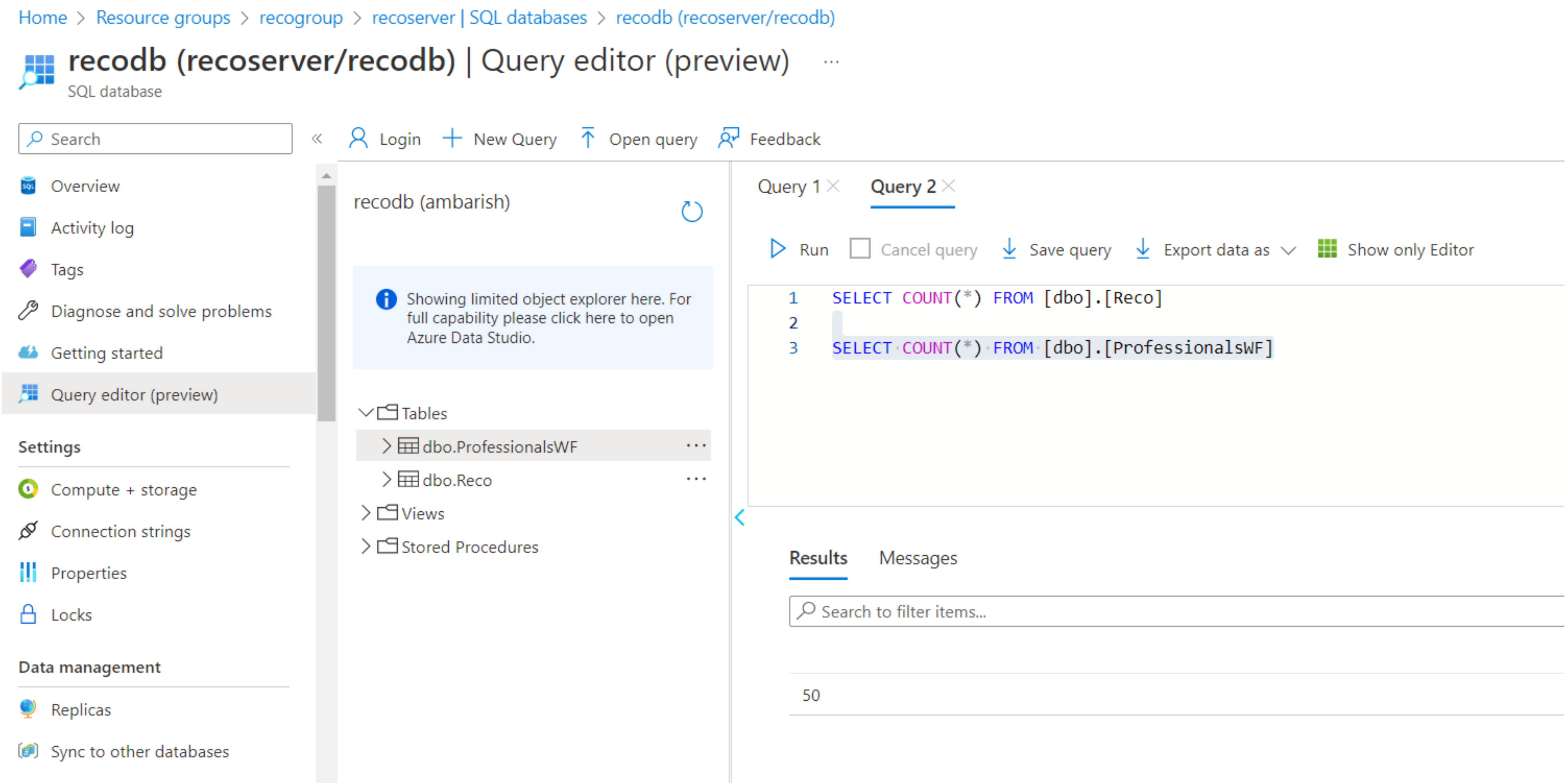
```
 /mnt/c/Am/career-reco-b/servicebusapi  master !1  kubectl get pods
NAME                                          READY   STATUS    RESTARTS   AGE
ingress-nginx-controller-6f7bd4bcfb-9z6c6    1/1     Running   0          101m
recodeploy-7df8d966c8-7q5tc                   1/1     Running   0          3h24m
recodeploy-7df8d966c8-c4bsz                   1/1     Running   0          3h24m
recodeploy-flaskui-87c555b89-4qm2m            1/1     Running   0          166m
recodeploy-flaskui-87c555b89-b5rhp            1/1     Running   0          166m
recoservicebusdeploy-689c8b6d86-cnrzk         1/1     Running   0          8s
recoservicebusdeploy-689c8b6d86-f928g         1/1     Running   0          8s
 /mnt/c/Am/career-reco-b/servicebusapi  master !1  kubectl get service
NAME                                   TYPE           CLUSTER-IP     EXTERNAL-IP      PORT(S)
                   AGE
ingress-nginx-controller               LoadBalancer   10.0.163.198   20.219.226.54    80:32122/T
CP,443:31339/TCP   101m
ingress-nginx-controller-admission     ClusterIP      10.0.6.47      <none>           443/TCP
                   101m
kubernetes                             ClusterIP      10.0.0.1       <none>           443/TCP
                   3h56m
reco-service                           LoadBalancer   10.0.207.106   20.219.230.250   8000:31681
/TCP               3h25m
reco-service-flaskui                   LoadBalancer   10.0.38.186    40.80.72.85      5000:32721
/TCP               167m
reco-servicebus                        LoadBalancer   10.0.224.70    20.207.106.233   8000:30147
/TCP               11m
```

Professionals Kubernetes Service

1. The service reads the Recommendations from the subscription in the Azure Service Bus
2. The recommendations are combined with the Professionals data stored in Azure Storage File Share accessed [using **Persistent Storage Volume and Persistent Storage Claim]** to get the Professionals who can answer the question.
3. The recommendations are stored in Azure SQL
4. The professionals who can answer the question are stored in a table in Azure SQL along with the question. This table can act as a repository for questions to be answered



Databases and Tables



Challenges in implementing the solution

We explored the use of **Sentence Transformers** which is a State of Art technique for NLP problems. The container image for this technique was huge in size compared to the TF-IDF technique and the performance was similar. Therefore, we used the TF-IDF technique.

The solution makes use of several Azure services such as Azure Kubernetes Service, Azure ML,Azure Service Bus, Azure SQL and Azure Storage. Integrating it required considerable planning. The seamless integration between the Azure services helped to make the implementation easier

Business Benefit

This project can be used all over the world as a tool of career recommendations for underserved youth for the betterment of their careers by qualified professionals. This technique can be extended to several other fields such as Question Answer solving for Tickets in the IT Service Industry, Knowledge base enhancer for new joiners in a field where the access to skilled professionals is difficult.

GitHub link

https://github.com/ambarishg/CareerRecommender has all the code and the deployment steps