Using Js & Python Developed Enterprise Application (Azure DevOps and Azure Kubernetes Service)

Hey, I have a fabulous idea to connect hawkers and turn them into enterprise

Problem Statement

Our aim is to connect hawkers and give them online platforms where they can register and showcase their products and share the outlet location. So they can keep connected with customers. Also, they can take bulk orders and list new items. Direct communication between customer and business owner. No intermediator. Platforms allow them to use their own payment method. No charges for registration. No Geometric restrictions. Anyone from around the globe can place an order and direct the mode of communication. Logistics are managed by the owner. All hawkers' registration is verified. Motivating them to register as a proprietorship. Plus we provide training in digital transformation and digital marketing.

How to start it?

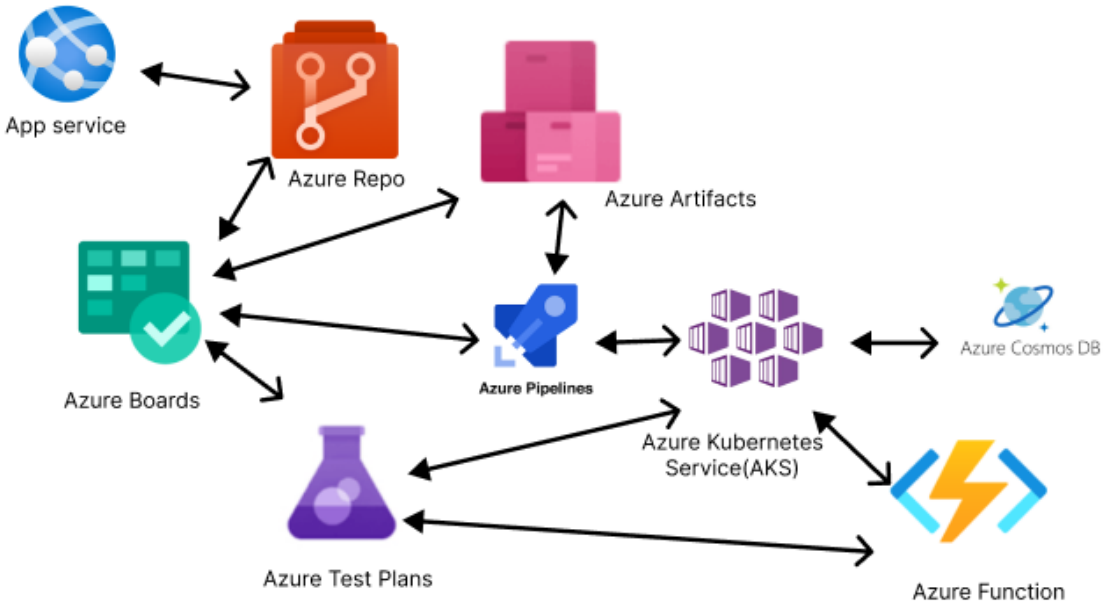
As software engineers, we learn many methodologies of software development but we never implement them. Most of the time we directly write the code. Hey, what's wrong with it? We know the great people who write the code to develop and deploy the application and that change the world. My dear friends, you are right but it is half part of the story. Before developing the applications lots of processes, research, and feasibility checks go on after the report of all these the concept or idea stays without any abbreviations, the response of user adaptability, and making a profit, then only it go to further steps like development and deployment. It is best practice to start small and evolved slowly and with periodic growth. For that, you need proper project management tools like Jira or Trello boards. Also, you need contentious development, testing, and deployment. For that, we need to use a common repo for that we need git. tools for building CI/CD pipelines using Jenkins, ansible for infrastructure configuration Terraform for infrastructure monitoring and deployment through the containerized applications using docker, and orchestration using Kubernetes but all in different tools

can we have a common platform where all tools are pre-configure so we can focus on business development instead of managing infrastructure more. The answer to this question is **Yes,** We have azure DevOps which has all these features.

Why Azure?

One of the benefits it all tools you need are already in a few clicks with basic configuration, without installation and messy configuration unlike other platforms like AWS. In Azure DevOps is a service, so you have just implemented it. You can use these tools individually as well. Azure DevOps comes with Azure Boards, Azure PipLine, Azure Repos, Azure Test Plans, Azure Artifacts, and lots of extensions for customization. Another benefit is you can pay in Rupees(INR). Azure Allows you to pay in Indian Ruppe(INR) which is another major benefit from a business owner's perspective. Apart from that Azure has many managed and serverless services where you don’t need to keep managing the infrastructure.

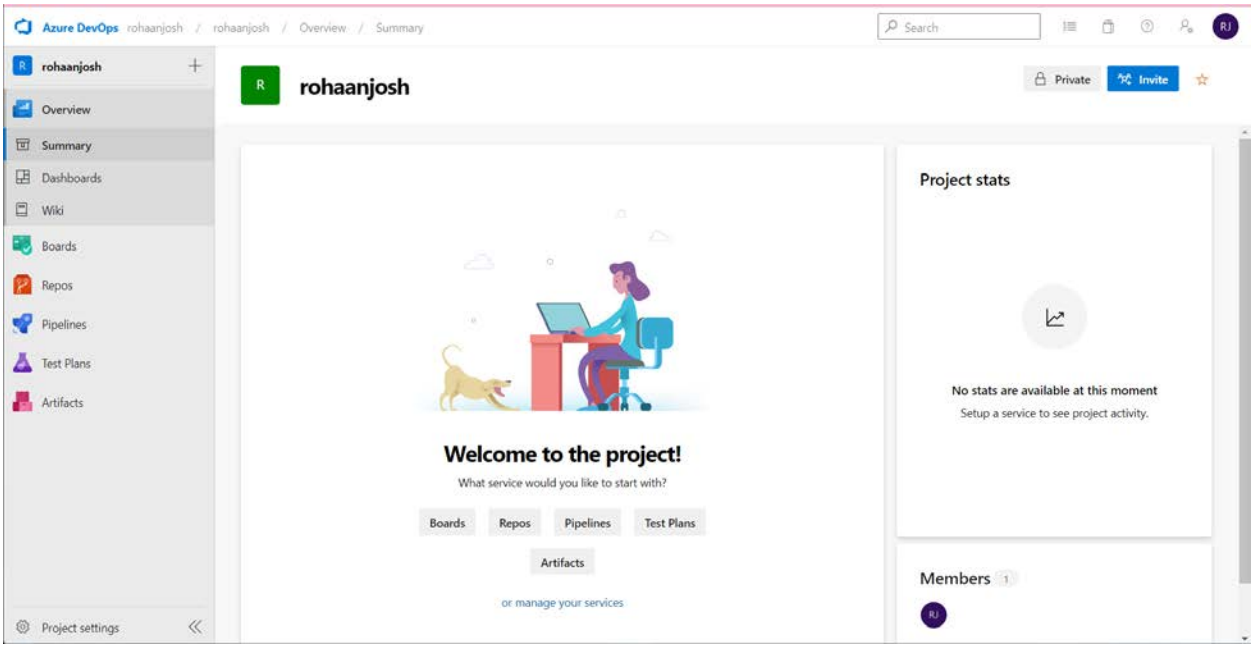
Solution Architecture



What is Azure DevOps?

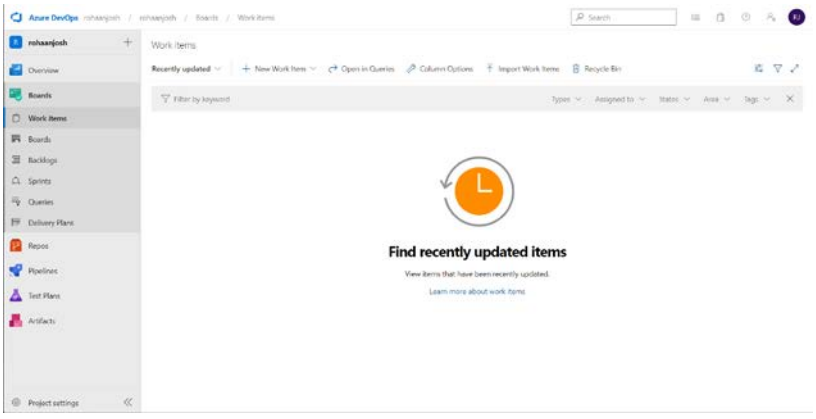
It is a software-as-a-service platform from Microsoft. It implements all your DevOps processes. Earlier it was known as Team Foundation Server then Visual Studio Team Services now it is called Azure DevOps. DevOps is the combination of Concepts, tools, and practices to develop applications faster in an automated and high-quality way. The aim of DevOps is to make the software development lifecycle more efficient and automated. Azure DevOps is the technological implementation of the DevOps process of the Software development life cycle.

Azure DevOps Overview

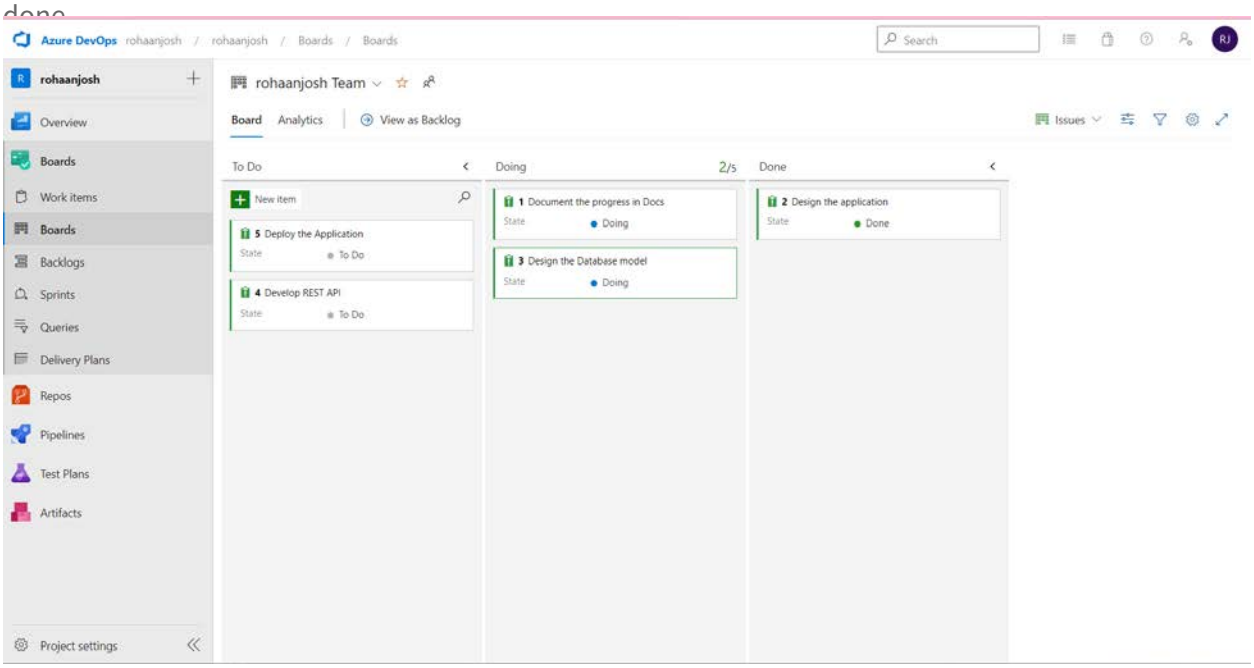


When you sign in to Azure DevOps it is the first screen you will get. It shows the summary of the project by default as well as you can see there is a Dashboard section where you can monitor your application and process. Wiki is the markdown-based page it helps to bring the team on the same page. It documents the current version or shares repos or code.

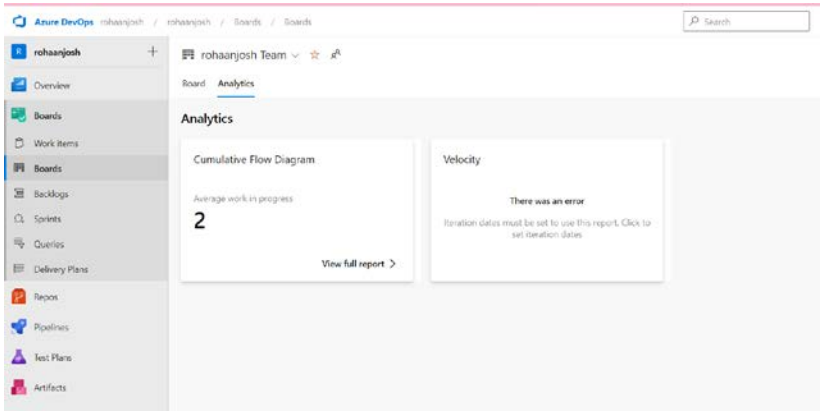
Azure Boards



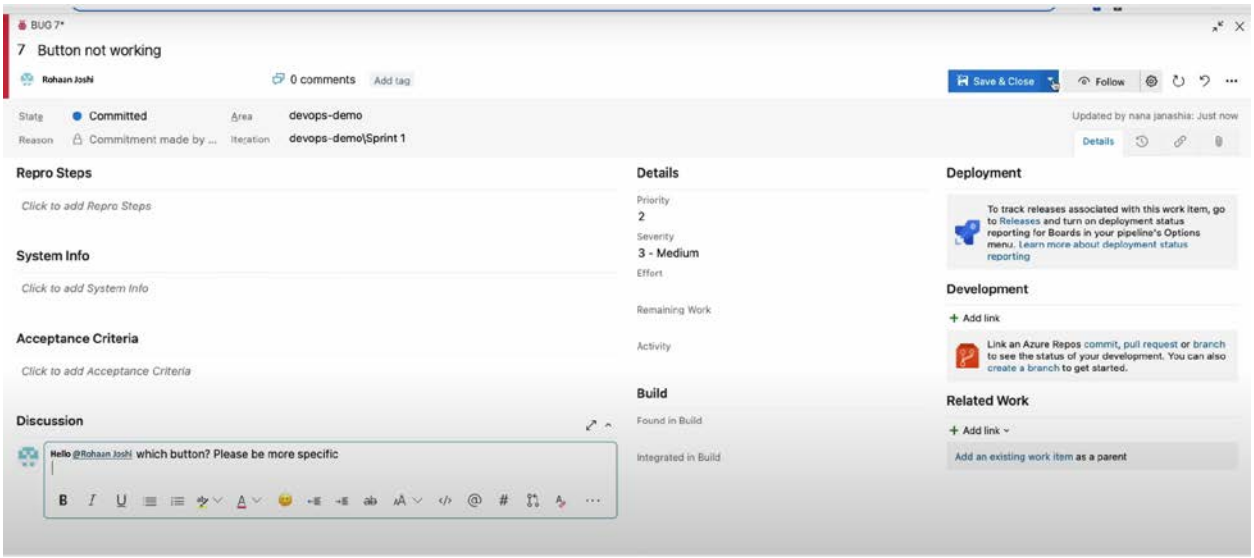
Software development life cycle includes more phases than coding, testing, and deployment. It includes the Planning phase before writing the single line of code. It can be done by the product manager. It is telling that what we are developing? Why we are deploying it? What's the business value? And it also includes the UX Design at the initial state after the UX Designer report then the manager decided how to move ahead because UX design gives the reality and feasibility checkers. There are serval workflows and processes but **Agile** and **Scrum** are the most famous. It is used by managers to decide how to divide the task or split the teams on which roles and many more. It is a simple workflow where we can mention What task we have to do, what we are doing, and what is done.



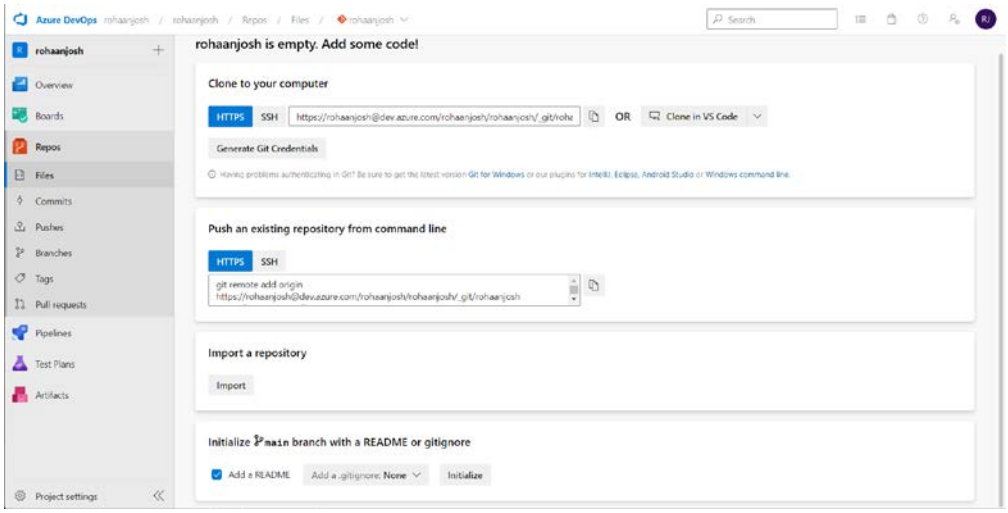
The Todo is allocated by the manager or you can add your own as well. You just drag and drop it as you make the process in the task for example I am writing the blog as well as I am designing the Databases model. I already design the application. And I have to Develop and deploy the REST API.



You can see the analytics as well. Apart from that, you can create tickets, and tasks for features, improvement, or bug fixes as a part of the agile or scrum process. And you can assign it to the team members and you can track the progress of this task. Also, it leverages communication between developers if any one of them has questions on the task they can discuss it as well. If you are an office 365 user then you can connect Teams with Azure Boards and directly communicate with each other and solve the problem. Plus you have an overview and transparency as well as the status of all the tasks. You will get to know who is developing on it, and what's the status of the task. It is perfect for team project as well as a solo full-stack developer it also helps you keep the track of your progress and motivate you to accomplish the further task in the project.

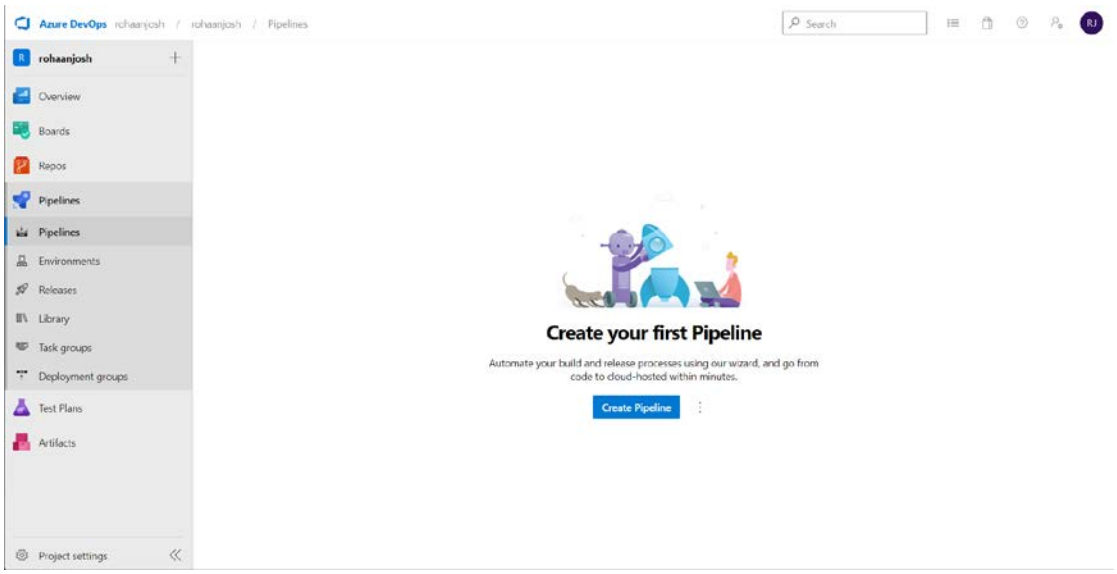


Azure Repos



It is similar to the Private Github where you can manage the code with an advanced file management tool. Here developer pushes the code through the branches to the main in collaborative may in a repo. The code is hosted in Azure Repo and changes are pushed. It is similar to GitHub based on GIT. It also allows developers to create a pull request and communicate, collaborate and merge into the main branch. It also linked with Task on Boards so you can see them there under the release work.

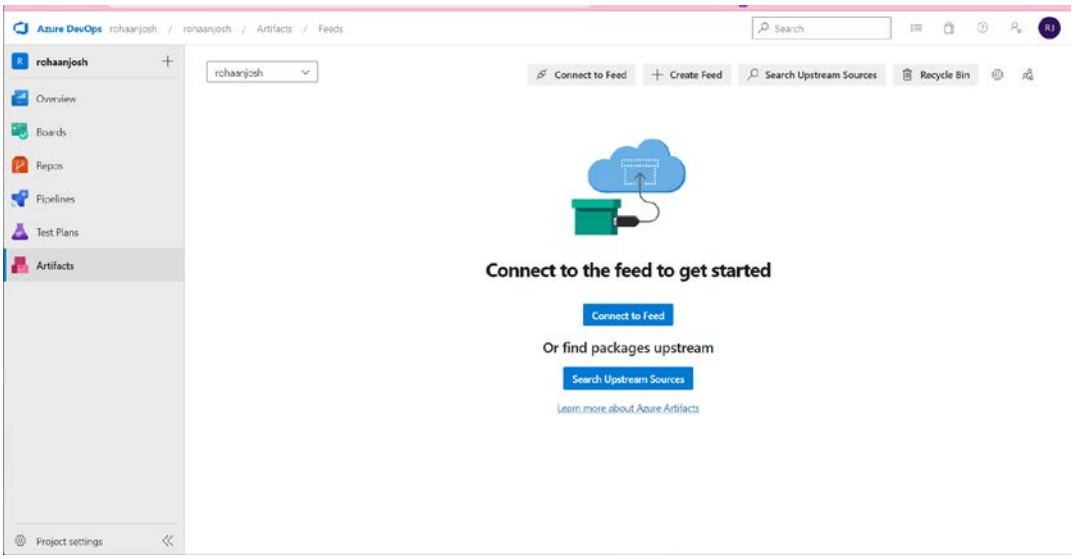
Azure pipelines



After the feature is developed and merged into the main branch then it is ready to release before it goes to the production environment it must be tested and packaged into an artifact which is a deliverable or deplorable format like Java projects creating jar files. This process of continuous Testing and packaging application automatically called the Continuous Integration(CI) process. Building this process pipeline, Azure Pipeline services for it. You can write the pipeline scripts using a YAML file. You can host this pipeline config with the rest of your app code in your git project. You have to define the steps to run the test, package the application, create a docker image, and push the docker image. Each step executes a command. To execute commands in script attribute. Instead of writing the scripts by own you can defiant the task. There are predefined tasks available for different usescase. Tasks are pre-created scripts. You can use it as per your convenience. You can select and configure using the user interface. After configuring and saving the task you will get the code into your Yaml file for that particular task. You can edit, adjust or add additional configurations to the code as you want. It helps you very much because you don't have to know the command exactly without memorizing the syntax for the pipeline and don't write to script from scratch. It is a very high-level simple approach. You can define the multiple jobs in a pipeline. The job contains multiple steps. In the case of multiple jobs, we need to define on jobs attribute. Also, each job has the pool attribute which is the environment where you want to run or test your code, for example, my frontend is Android App, and the backend REST API is in a Linux-server. The database is in Windows-server. You can specify the version of the OS as well. You can run parallel jobs as well. It supports any language, platform, or cloud. Establish a connection to GitHub or another Git service to continuously deploy. To develop a

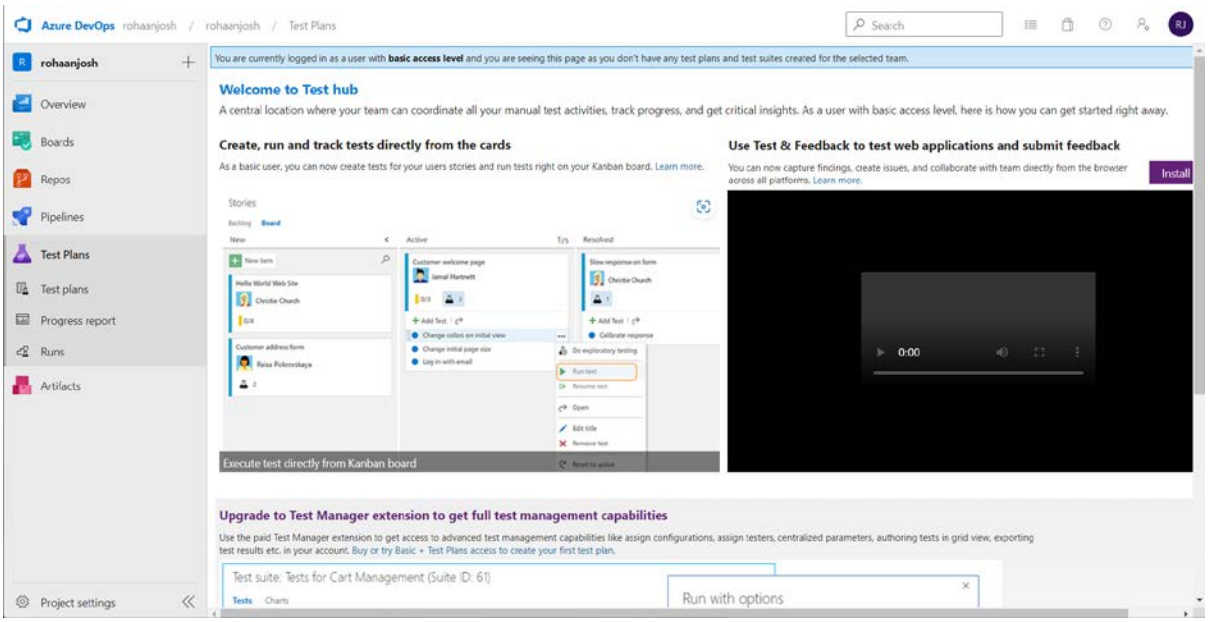
complete DevOps pipeline there are stages as well which you defined in the YAML file, for example, the build stage or deploy stage, test stage, and develop stage. As per your project you can define those. Also for complex projects, you define the templates as well.

Azure Artifacts



Artifacts are different based on the programming language used in writing the application. For example Java Applications Jar files, and .NET Application for Nuget package. For storing this kind of artifact in Azure DevOps we have Azure Artifacts. It currently supports four types of packages as of now Maven, Nuget, NPM, and Python packages. The source can be public or private. But in the modern world, we did not produce these artifacts instead of it we produce the docker image as the artifacts. No matter what langue, or what tools you are using, Artifact remains the same as the container image type.

Azure Test plans



Testing is the most important part of application development. Before going into the production environment the application must be tested manually or automatically using tools. Azure allows you to create manual test cases also automated test cases. You have centralized places to overview all the test plans. Also, it connected with Azure Boards.

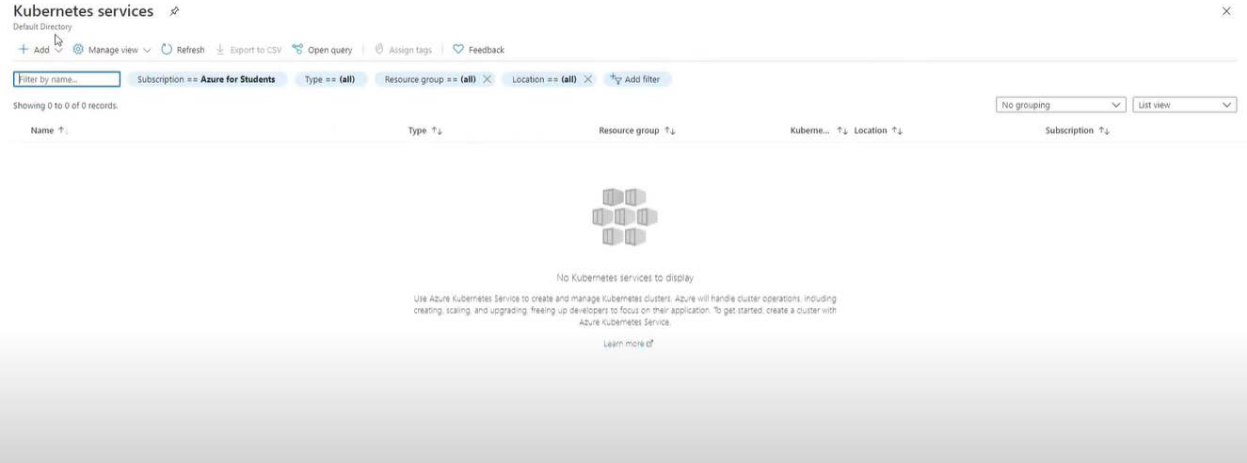
Azure DevOps is a managed service so you have to pay for it but we have free tiers for it. For pricing refer [Azure DevOps Services Pricing | Microsoft Azure](). Azure allows external services under the project settings in the Service connection. Also, you can use a self-hosted agent. In Azure DevOps has an independent account. It came under the same common Azure ecosystem. For deployment, you have to connect it to Azure or a different cloud provider which is the major benefit of it. Because you can manage multi-cloud platforms very efficiently using Azure DevOps.

For our project, we are going with the Free Tier. Let’s build it!!!

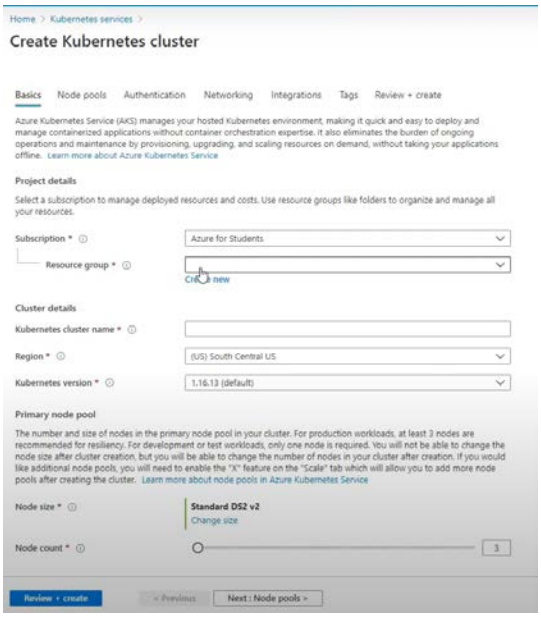
In our case, we are using Azure Boards, Azure Repos, and Azure Artifacts because it comes in the Free Tier. We deployed our backend at AKS Azure Kubernetes Service. And for the database we are using

Azure Kubernetes Services(AKS)

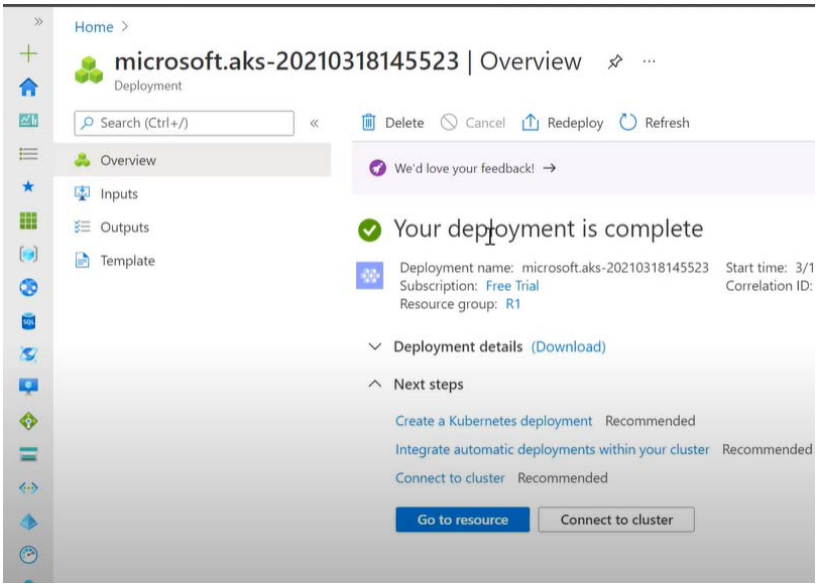
It is managed service provided by azure. It helps to deploy apps quickly and managed the cluster very easily and we have just configured it. We are deploying our backend in Kubernetes because we already containerized our application. For that, We are using **Azure Kubernetes Services(AKS).** Using Azure pipeline we have another option to deploy our app directly by doing certain configurations.



Click on Add to create a Kubernetes cluster



Select the resource group, give the name of the cluster, and select region and I am keeping other things as default.



Now open the command prompt and login on to the Azure portal using the az login command.

```
az aks get-credentials – resource-group R1 –name myaks
```

Now Upload the YAML file in the Kubernetes cluster to deploy the app.

```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: sample
 labels:
  app: sample
spec:
 replicas: 1
 template:
  metadata:
   name: sample
   labels:
    app: sample
  spec:
   nodeSelector:
    "kubernetes.io/os": windows
   containers:
   - name: sample
```

```
      image: mcr.microsoft.com/dotnet/framework/samples:aspnetapp
      resources:
        limits:
          cpu: 1
          memory: 800M
      ports:
        - containerPort: 80
  selector:
    matchLabels:
      app: sample
---
apiVersion: v1
kind: Service
metadata:
  name: sample
spec:
  type: LoadBalancer
  ports:
  - protocol: TCP
    port: 80
```

```
  selector:
    app: sample
```

And cmd of deploying YAML file.

```
kubectl apply -f manifest.yaml
```

Azure App Service

For the frontend of the application we are using App Service to deploy the app on a Serverless environment. It is a fully managed service by Azure. And I am using this service over AKS because I am deploying the app on Jam Stack where I am doing client-side rendering more than server-side rendering. I have optimized my app accordingly. For that, I am using React Js-Based Frameworks. I create another repo for it. For this I used VS-Code Plugin for the same and using this I am deploy the app just by connecting it to the Azure Repo because I already build the app in the Azure Artifacts.

As of now, our project is in the beta development phase where we allow people directly add hawkers but we are verifying their Adhar card details of them if they are not matching then they can't be instead as well as we give them a call and use video calls we verify it again.

#ShopforEveryone

Sujay Fireworks

All kinds of fireworks available

LBS Road, Near Ram bagh, mulund

Ramesh Electric

All kinds of lights, focused blub, LED available

APMC Market, near sham bank, vashi

Manoj Pottery

We having Flask, Lantens, Panti,Deep, Pan, Matka

cosos bank lane kumbhar ali, kalayn(w)

Kumar Decorators

We having many different decoration items

saras bagh, near chitale bandhu shop, pune

#ShopforEveryone

Registration As Shopper

Full Name

Business Name

Adhar Card No

Address

Mobile No

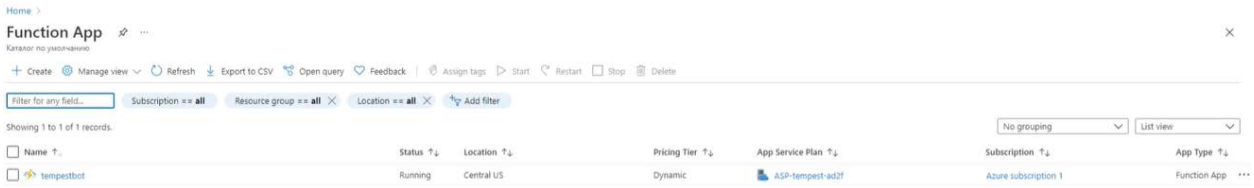
Location of Stall

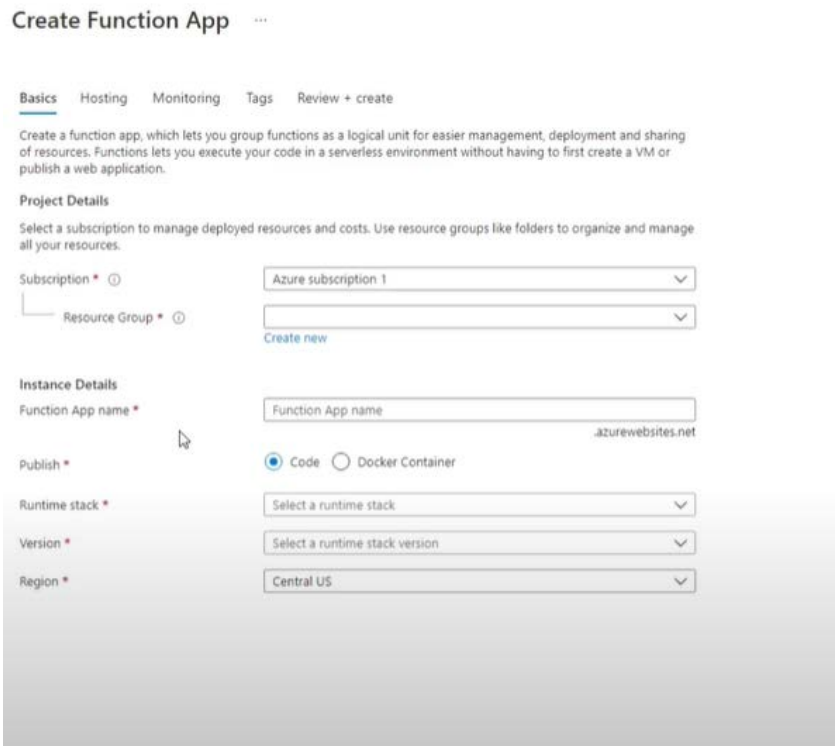
Submit

Azure Functions

To do the verification process we are using Azure functions. Serverless solutions like Azure Functions let you write less code, maintain less infrastructure, and spend less money. The cloud infrastructure offers all the modern resources required to keep your applications running, so you don't have to bother about setting up and managing servers.

Azure Functions takes care of the rest while you concentrate on the code that matters most to you in the language that is most productive for you.





Add resource group the same as AKS, give the name to the function, and select the stack I am selecting Python to minimize the code. The version of python as well as the select region near to you. If you don't have a storage account then you have to create a storage account for function. And the rest of the configuration I am setting as a default. And Hit on review and create function.

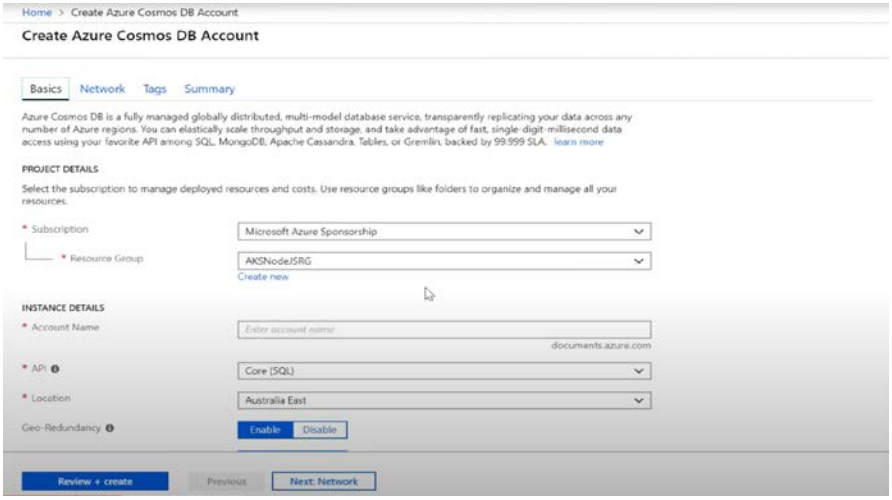
Code:

```
num = input("Please enter a 12 digit mobile number: ")
status=None
# Check if number length is greater or less then 12
if len(num) > 12 or len(num) < 12:
    print("Aadhar number is not valid (Enter a 12 digit number)")
else:
    # Check if first number is 0 or 1
    if num[0] == '0' or num[0] == '1':
        print("Aadhar number is not valid (Aadhar card number cannot
start with 0 or 1)")
    else:
        try:
            num = int(num)
            print("Aadhar number is valid")
        except:
            print('Aadhar number is not valid (Aadhar number should
not contain any characters)')

# Check if number length is greater or less then 12
if len(num) > 12 or len(num) < 12:
    print("Aadhar number is not valid (Enter a 12 digit number)")
else:

# Check if first number is 0 or 1
if num[0] == '0' or num[0] == '1':
    print("Aadhar number is not valid (Aadhar card number cannot
start with 0 or 1)")
else:

try:
     num = int(num)
     print("Aadhar number is valid")
     status="Verified"
     return status
```

```
 except:
     print('Aadhar number is not valid (Aadhar number should not
contain any characters)')
```

Azure Cosmos DB

It is NoSQL based database used for a fully managed, serverless distributed database that supports open-source, creates high-performance applications of any size or scale. At present my model is so simple I am just doing base CRUD REST API. We were taking input as name, mobile number, adhar number, address, phone number, business name, location of stall, and Images which are stored in Cosmos DB as well.



Select the same resource group through the project for API I used MongoDB location which is near to you select it keeping other things as default. Go ahead click on Review and create.

Using Azure CLI we are creating a Database inside the cosmo DB

```
az cosmodb create –namemycsomdb –resource-group myresourcegrp –kind MongoDB
```

Now the database is created go ahead on CosmosDB Tab and allow or set the while listed IP of your backend. It is working similarly to the MongoDB Atlas. Just configure it and connect both the backend and the Database. That's it. Here we finished the Beta-1.0.0 version of our first enterprise application. Congratulations!!!

Knowledge Sharing and Best Practices:

- Chose technology stack wisely like I used Jam Stack
- Be minimalistic while developing the enterprise applications
- Keep things simple and clean Architecture of application.

Challenges Faced:

- Integrating multiple components is a very skilled job. That makes me smile many times on me. Azure DevOps and Azure Cloud services are two separate entities in the same ecosystem. It is not necessary to use Azure Cloud for all Azure DevOps projects. Integration is fun.
- After Integration check the workflow and find & fixing bugs. it's a really fun movement for me.

Business Benefits

- Helps to improve the economy of the country
- Hawkers come under monetization and will get the same benefit as registered business owners.
- It helps them to connect with their customer without hesitation of change of place, or season.
- As a project owner I am getting satisfaction helping them to establish the business to the next level.

By Rohaan Joshi