**Deployment of 10-Tier Microservices Application on Azure Kubernetes Cluster:**

**Introduction:**

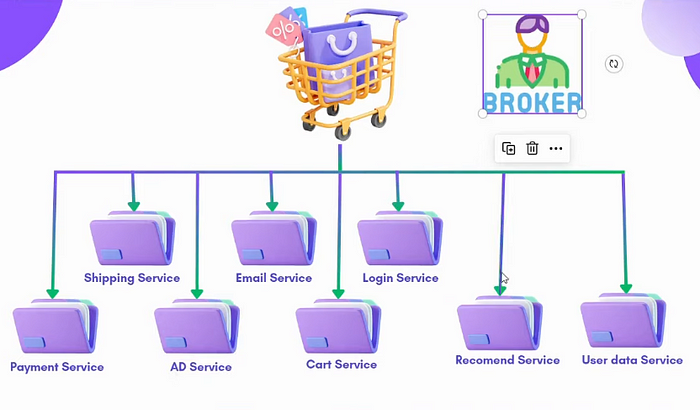
In this project we will use online shopping application which creates around 12 pods and same number of services when we creates the apps, so it likely to create 24 resources inside the kubernetes cluster when we deploy this application.

**What is microservices application and why we need it?**

Let consider the e commerce application like amazon where we do not write whole source code in a single folder but we write different source code in different folder for each service like payment, cart, shipping etc these all have different source code by keeping everything separate whenever need to change/update the source code of application it will easy to update the specific folder it will not effect the entire application. The best part of microservices application is that they can be written in different programming languages for each service.

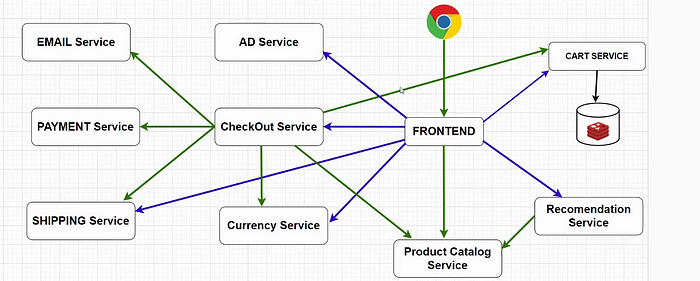
**How the microservices communicate among each other?**

There are multiple ways for that one of the way is known as “Message Broker” any service want to communicate with other service first it communicate with the message broker then it forward the request to other service.



Some of the common knows message broker is “RabbitMQ” and “RediCache”, Another way is like GRPC (Google remote procedure call), GRPC support too many variety of languages so we will also use it in our project.

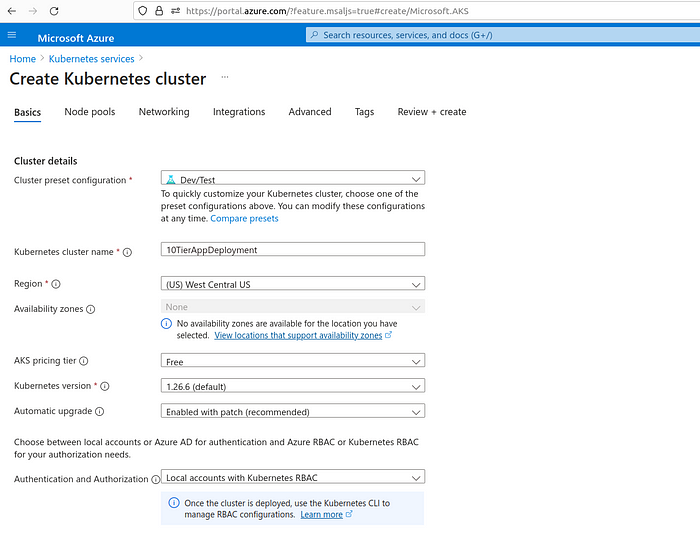
**Architecture of our 10 Tier Application:**



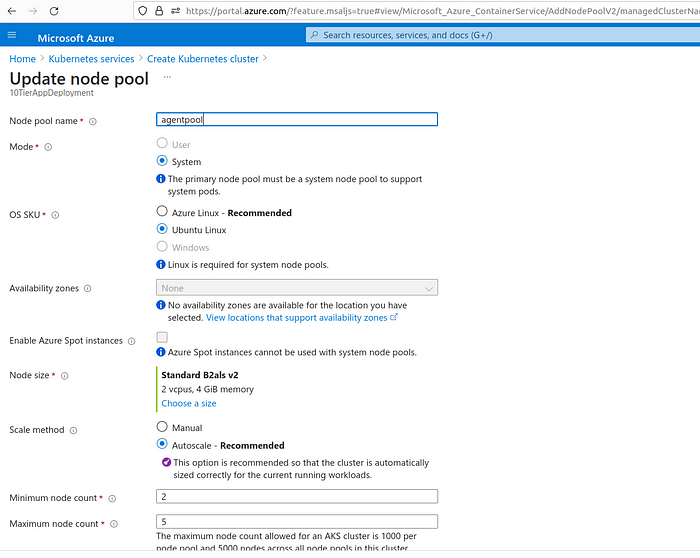
Through browser we will access the application and **front end** will be the entry point of this application Then there will be service called **product catalog service**which is used when you search for different products on the application. **Recommendation service**will recommend you the items based on your search. **Cart service**will hold the item that we wish to buy. We will use **redis database**to store the cart item in it. **AD service**will manage the Ads runs on the application. **Checkout service**is used when you go to buy the item stored in cart service, this service also link with **currency service**which help us to select our desired payment currency, From checkout service we also have other three services that are link to it first is **email service**that is used to get the invoice or receipt of the shopping, payment will be managed bo another micro service called **payment service**and at last there is **shipping service**that is use for shipment fo the purchased items.

These all services are written in different programming languages that is the beauty of microservice applications.

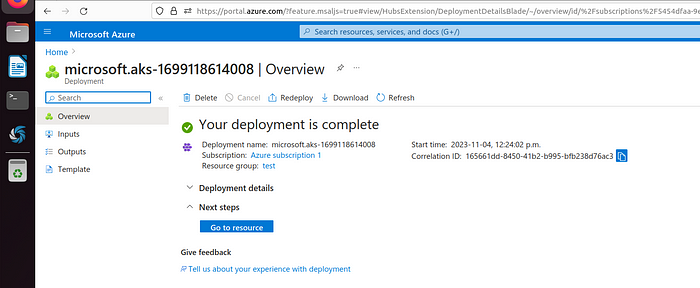
Now we will create kubernetes cluster on azure cloud called (AKS azure kubernetes service)to deploy this application, we will use Dev/Test environment because it creates small cluster and very cheap in pricing as compare to production environment:



Then define the node pool for your cluster, as we need just 2 cpus and 4gb of ram we will go with B2S node:

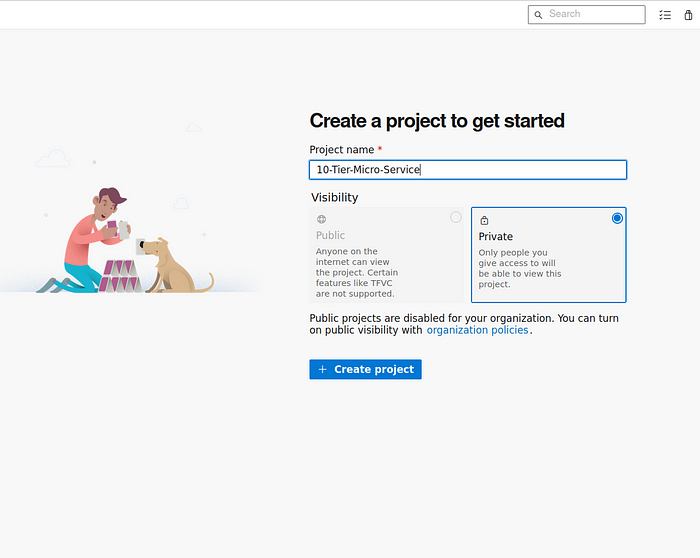


Go with all default setting in networking and integration, advanced, security tab and click on create cluster:

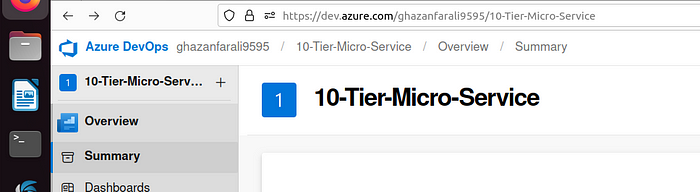


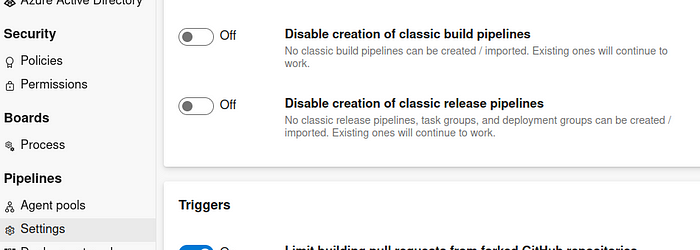
Once cluster is created we will create **Azure Devops Platform**for deployment of the application, we can also use jenkins CICD but we will go with this azure service, Azure DevOps is a set of development tools and services provided by Microsoft for software development, including CI/CD (Continuous Integration and Continuous Deployment) capabilities. It offers a comprehensive platform for managing the entire software development lifecycle, from source code management to build and release automation.

In azure devops environment we first creates the organization then create a project in it:

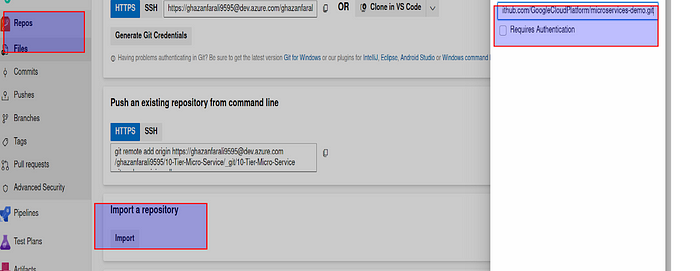


Once project is created go to organization level(ghazanfarali9595) and enable pipeline build process there so that we will use azure devops pipeline service in our project:





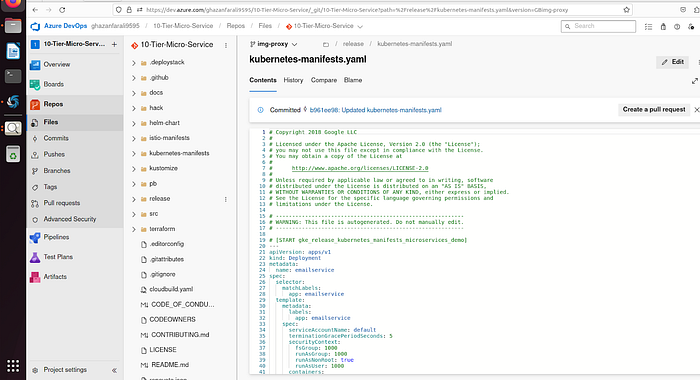
Now go back inside our project and go in Repos option and import a github repo there:



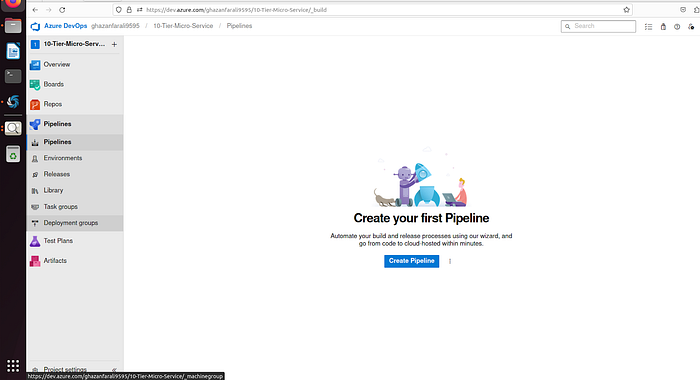
Repo:

<https://github.com/GoogleCloudPlatform/microservices-demo>

It have successfully import the github repo to our project actually this repo has all the deployments and service files for each microservice application that will deploy on AKS:

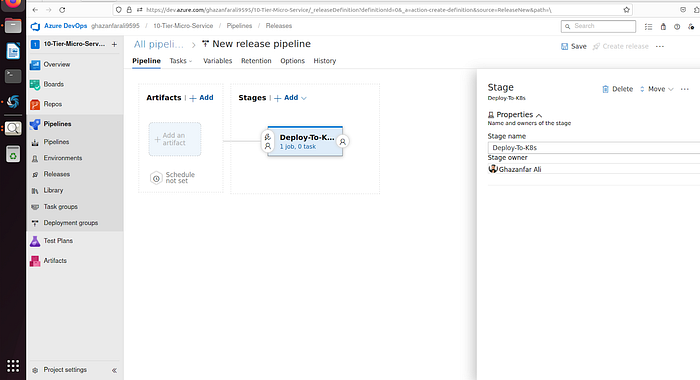


Now we will create pipeline for our project:

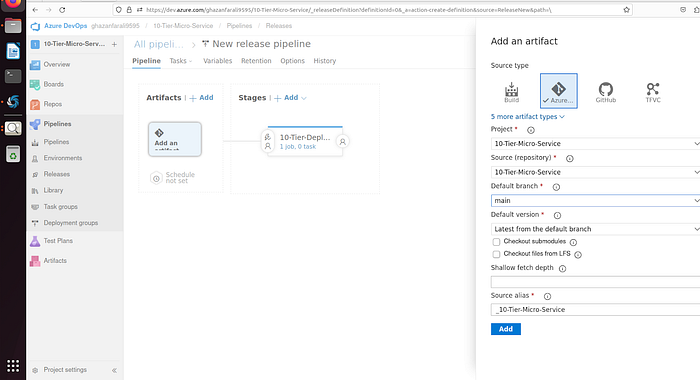


Since we have yaml file ready and we are not building the docker files from scratch so we will chose **releases** option in pipeline so that we can directly deploy to kubernetes cluster:

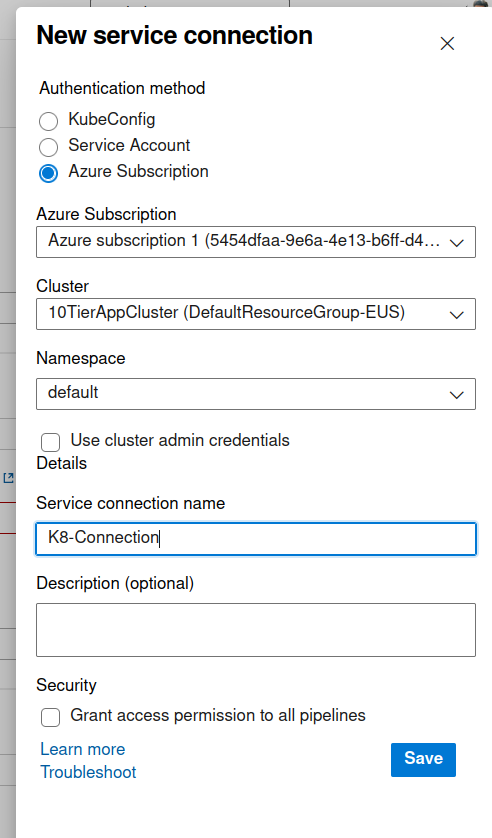




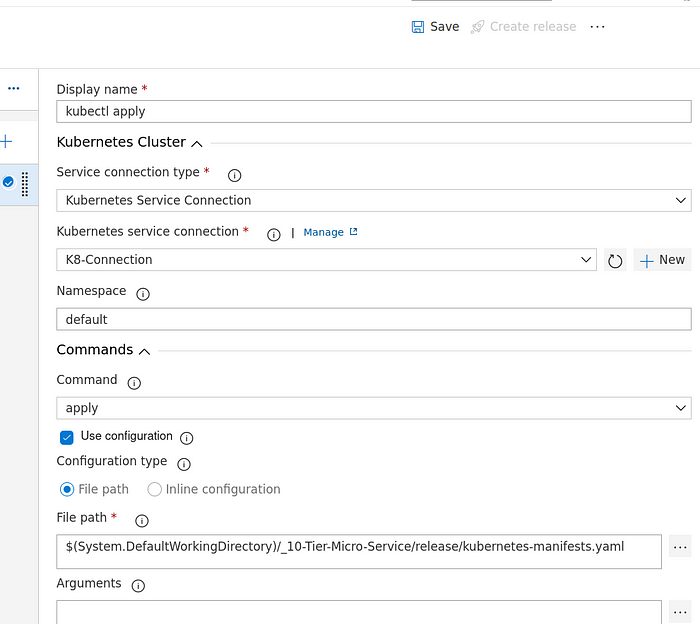
Here we created a new job then click on add artifact and and fetch the file from azure repo:



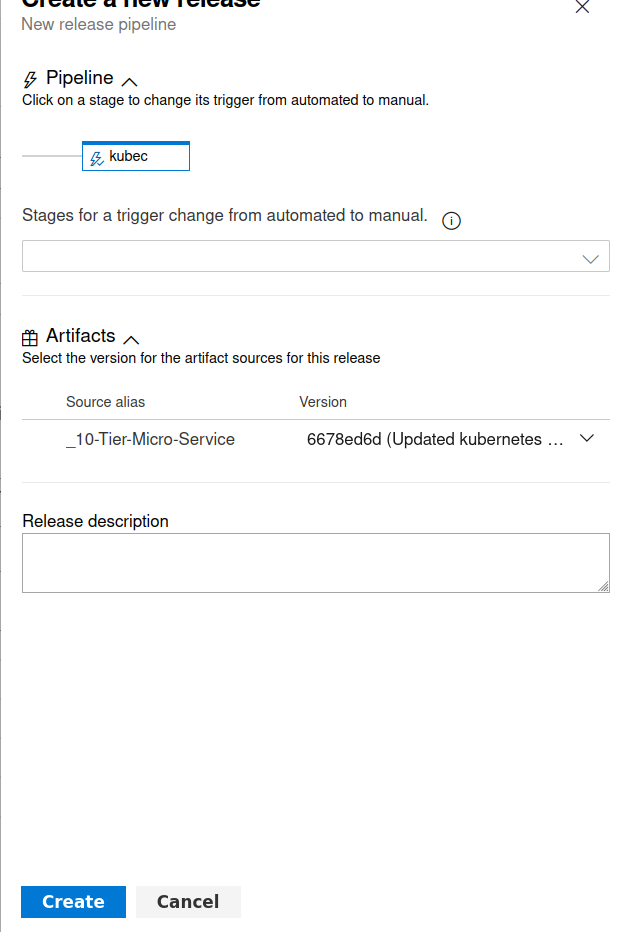
We will use kubectl for to communicate with cluster so that add kubectl in agent section of your pipeline job and connect it with your newly created kubernetes cluster.



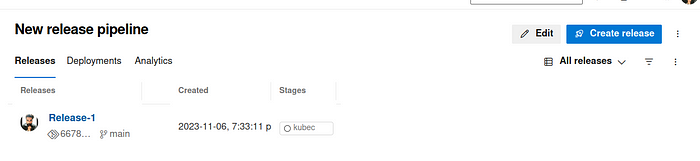
Use this connection in agent task, and in command tab use command “apply” and give the file path of kubernetes yaml file:



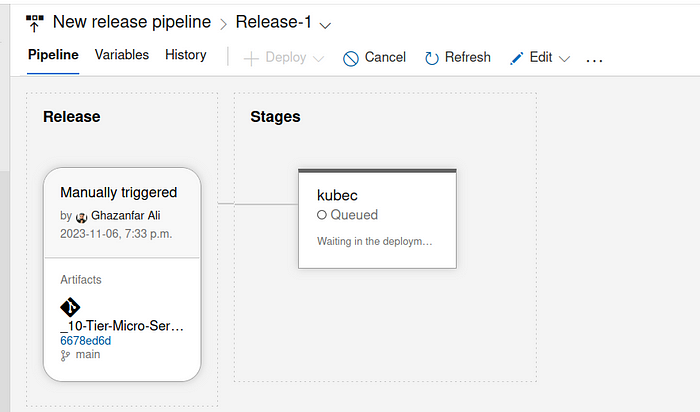
Once it saved, click on “create release” to do the deployment.



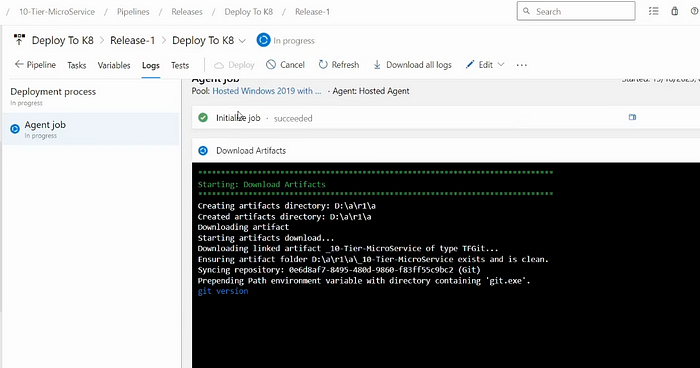
It has created the new release:



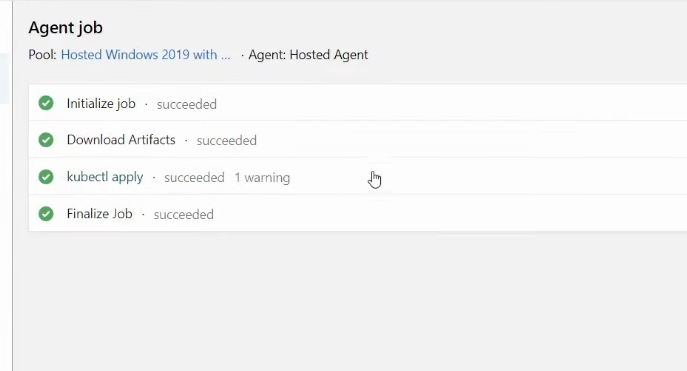
Now we will deploy the released pipeline:



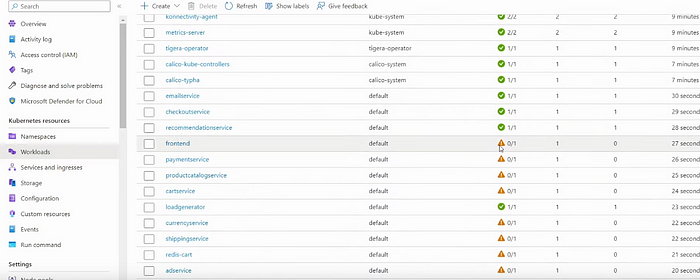
It has started the deployment:



This deployment can take little bit of time as it have to deployed 10 to 12 pods in our cluster:



Coming back to your cluster you will start seeing the deployed pods on your cluster:

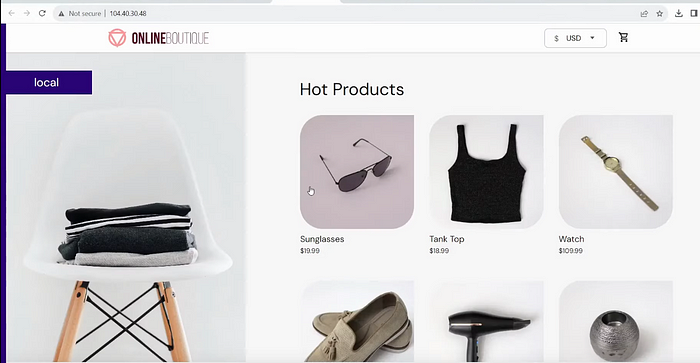


Once these all pods created we will be able to access our application through services and ingress tab on our azure cluster:

There will be a service with name “front-external” in which we have used loadbalancer service type so that we can access our application from the outside world:

https://miro.medium.com/v2/resize:fit:700/0*8TIbisrkbwryFQQ4

Click on the service to check the application deployed status:



Our app is deployed successfully on azure kubernetes cluster by using azure devops organization.

Thats all in this project!