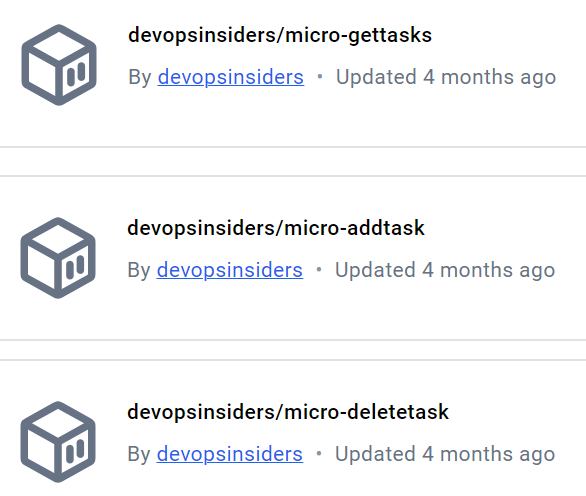
**Three tier application Microservices Deployment**

**Prerequisite:**

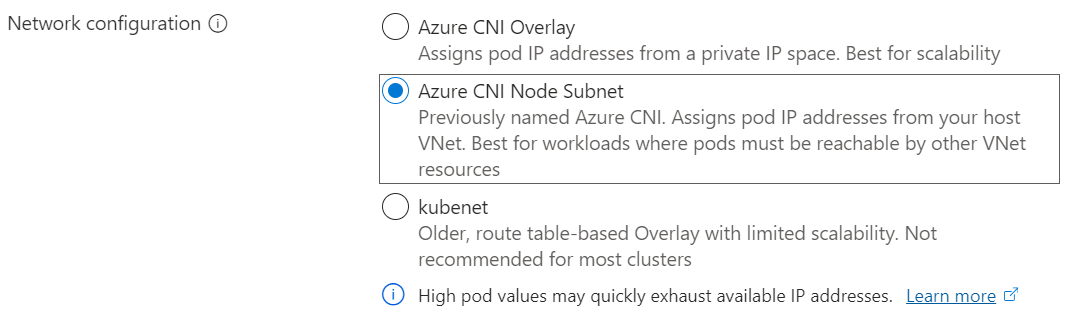
1. One Databased Creation Needed
2. For Best Practices, Need to store Database connection string in key vault
3. One Key Vault Creation Needed
4. Frontend, Backend and Database
5. Need one Docker System for build Frontend image
6. AKS creation needed
7. Need docker repositories details for backend deployment for docker images.



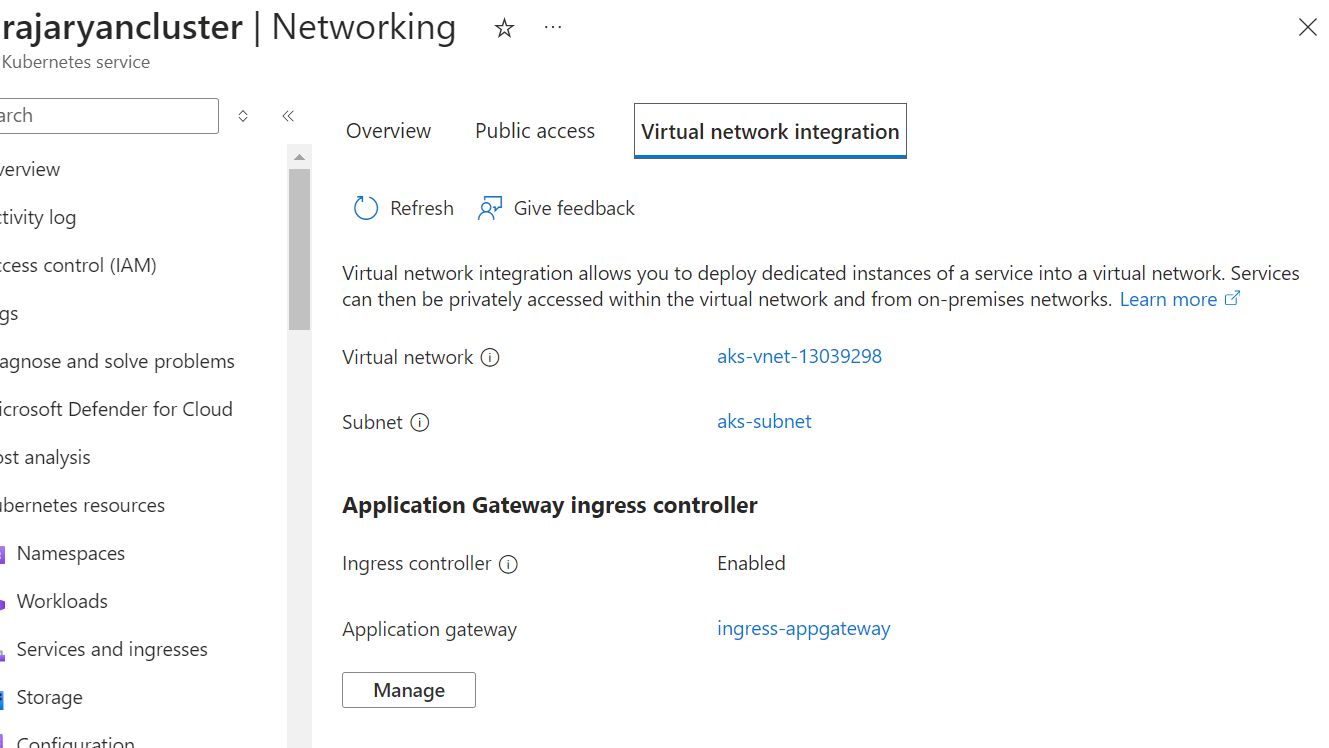
1. Need git repositories details for frontend deployment

**https://github.com/satishranjan7183/MicroTodoUI.git**

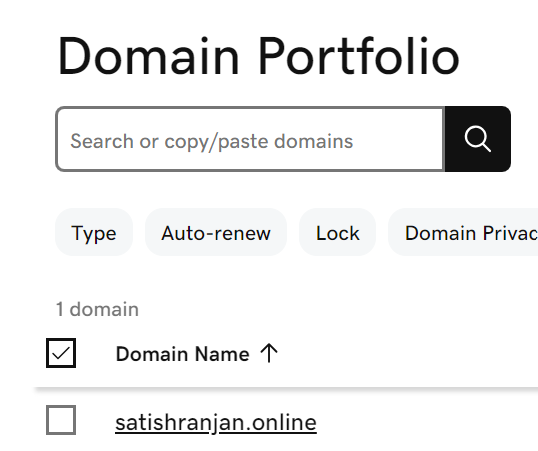
1. Need to enable Azure CNI Node Subnet in network configuration

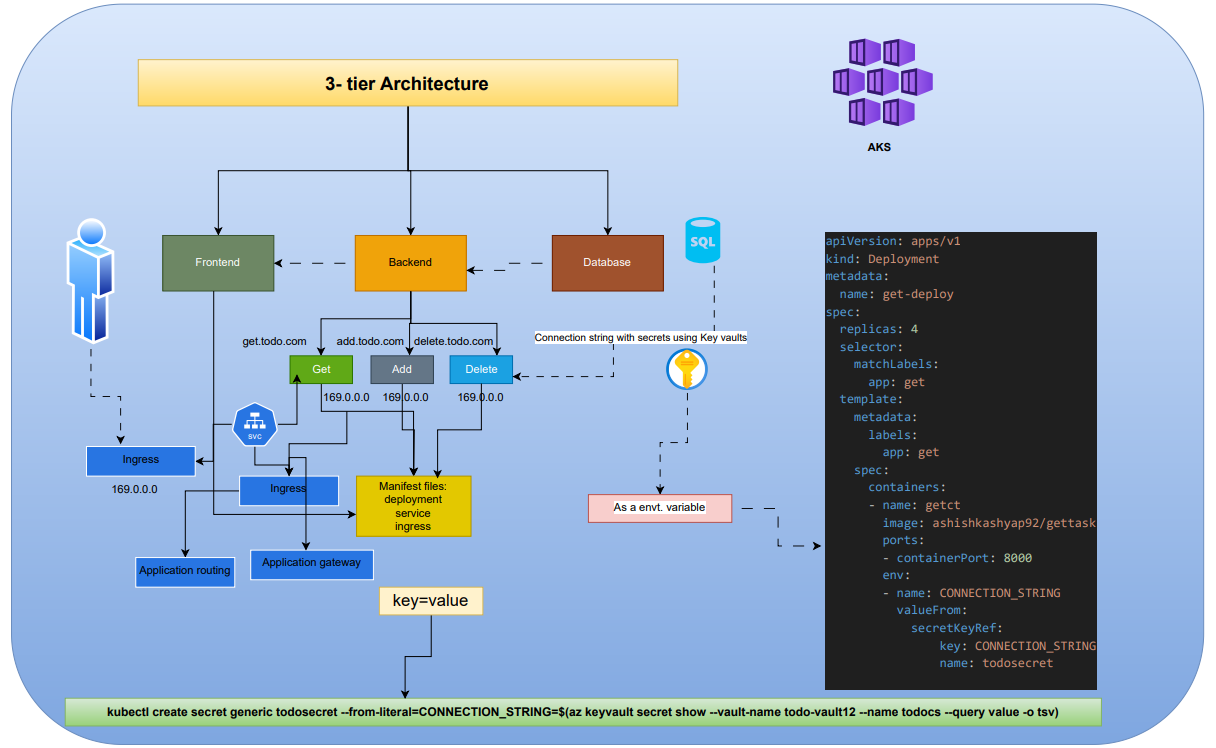


1. After Creating AKS cluster, need to enable Ingress Controller



1. DNS needed for this from godaddy.





**Deployment Steps:**

Step1: Need to create 2 folder in our Laptop :

1. Backend
   * 1. Gettask
        1. Deployment.yaml
        2. Ingress.yaml
        3. Service.yaml
     2. Addtask
        1. Deployment.yaml
        2. Ingress.yaml
        3. Service.yaml
     3. Deletetask
        1. Deployment.yaml
        2. Ingress.yaml
        3. Service.yaml
2. frontend
   * + 1. Deployment.yaml
       2. Ingress.yaml
       3. Service.yaml

 A screen shot of a computer code

Description automatically generatedA screenshot of a computer program

Description automatically generated

Need to write deployment.yaml , ingress.yaml and services.yaml for gettask, addtask and deletetask.

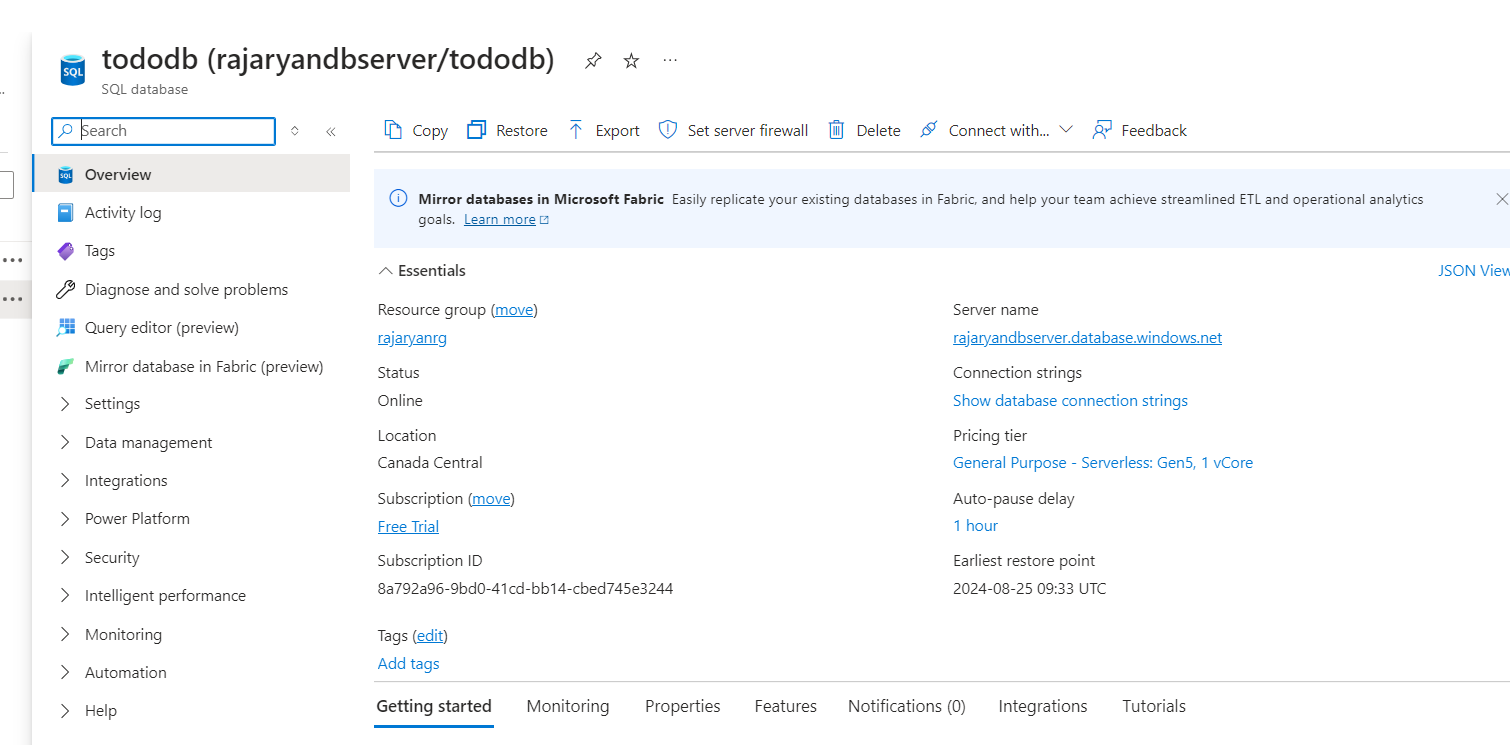
Step2: Create one database for application

* Search sqldatabase in azure portal - > Create database
* while creating database need to enable below parameter :

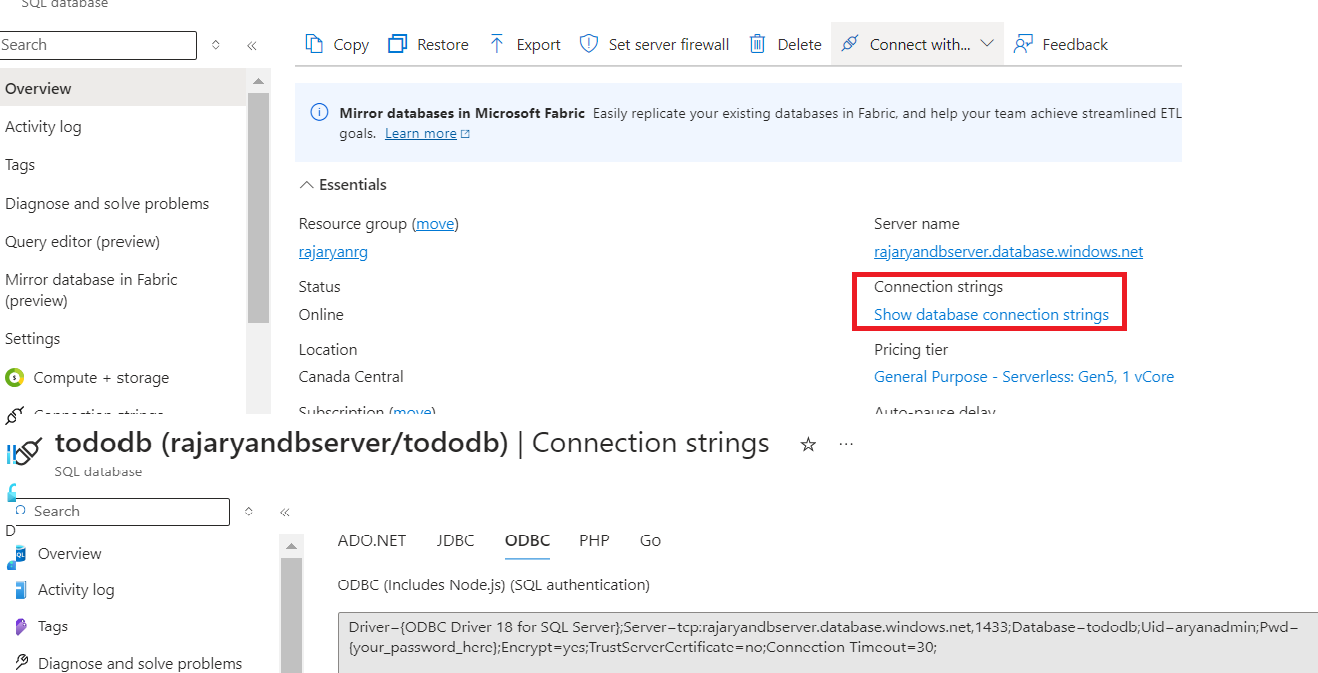
Allow Azure services and resources to access this server – Yes

Add current client IP address – Yes

* Put any name of database



Step3: Once database created, then copy connection string



Driver={ODBC Driver 18 for SQL Server};Server=tcp:rajaryandbserver.database.windows.net,1433;Database=tododb;Uid=aryanadmin;Pwd={your\_password\_here};Encrypt=yes;TrustServerCertificate=no;Connection Timeout=30;

**Updated ODBC driver version and password and then copy:**

Driver={ODBC Driver 17 for SQL Server};Server=tcp:rajaryandbserver.database.windows.net,1433;Database=tododb;Uid=aryanadmin;Pwd=Aryan1215@;Encrypt=yes;TrustServerCertificate=no;Connection Timeout=30;

Step4: Now Need to create Key vault and Secret

Key Vault Creation

* Search Key vault 🡪 create Key Vault -> choose Resource group and enter key vault name and then create

(Put any Key Vault name “todokeyvault11”)

Secret Creation

* Then go inside created keyvault and then got to object 🡪 click Secret 🡪 Click on Generate/import
* Fill below parameter:
  + Name: <Put any secret name “todocs”>
  + SecretValue : Paste connection string

“Driver={ODBC Driver 17 for SQL Server};Server=tcp:rajaryandbserver.database.windows.net,1433;Database=tododb;Uid=aryanadmin;Pwd=Aryan1215@;Encrypt=yes;TrustServerCertificate=no;Connection Timeout=30;”

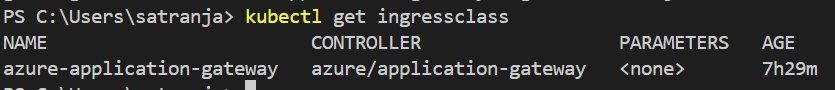
* And then create Secret.

Step5: Now Need to create secret Map in Kubernetes aks

* kubectl create secret generic todosecret --from-literal=CONNECTION\_STRING=$(az keyvault secret show --vault-name todovault11 --name todocs --query value -o tsv)
* Kubectl get secret
* Now this secret key and value will use in deployment in environment

Step6: Now Need to update yaml

* Update deployment yaml, ingress yaml and service yaml for all backend



* Deploy deployment, ingress and service for all backend application (add task, get task and delete task)
* In deployment yaml update image name which need to take from devopsinsider docker hub repo
* Kubectl apply -f . (Run this command from inside add task , get task and delete task folder where all deployment.yaml, services.yaml and ingress.yaml is present)
* Kubectl get pods
* Kubectl get all

Step7: Now Need to make one folder with name frontend and then git clone the code

* Clone the frontend code from below git hub link:

[**https://github.com/satishranjan7183/MicroTodoUI.git**](https://github.com/satishranjan7183/MicroTodoUI.git)

* Once clone successfully, then will go inside src folder and open **TodoApp.js** and update url of microservice:

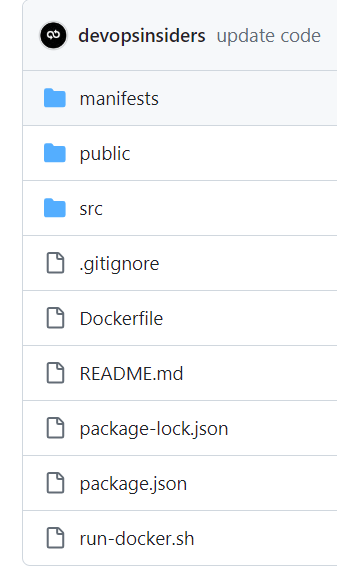


* Now login to docker machine:

Docker login

* Go inside the [MicroTodoUI](https://github.com/satishranjan7183/MicroTodoUI) where all frontend code present, then run docker build command to prepare frontend image:

Docker build . -t satishranjan7183/frontend



Docker images (to check)

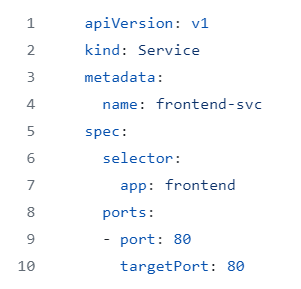
* Once image is ready, then push image to our docker registry

Docker push satishranjan7183/frontend

* Now go to docker hub and check image pushed in public registry

Step8: Now update the manifest file of frontend

* Now Update the frontend manifest yaml file (deployment.yaml, ingress.yaml and service.yaml)
* In service.yaml need to put port and target port 80 (both port same 80)

 A screen shot of a computer code

Description automatically generatedA screen shot of a computer code

Description automatically generated

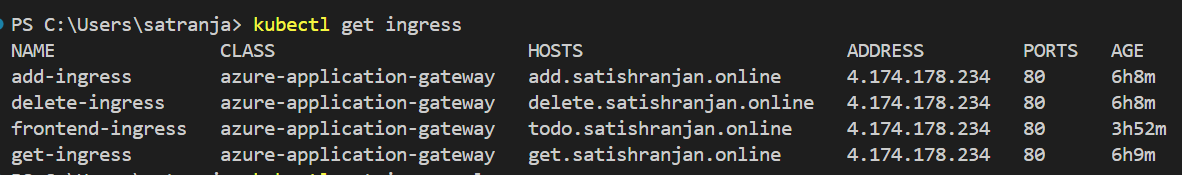
* Now go inside the frontend manifest folder and deploy:

Kubectl apply -f .

Kubectl get pod

Kubectl get all

Step9: Now check ingress IP address and then add host in godaddy



* Now need to login in go daddy and add host

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* Wait some time and then try below link in browser:

add.satishranjan.online

delete.satishranjan.online

todo.satishranjan.online

get.satishranjan.online

Step10: Now successfully deployed our Microservices in kubernetes