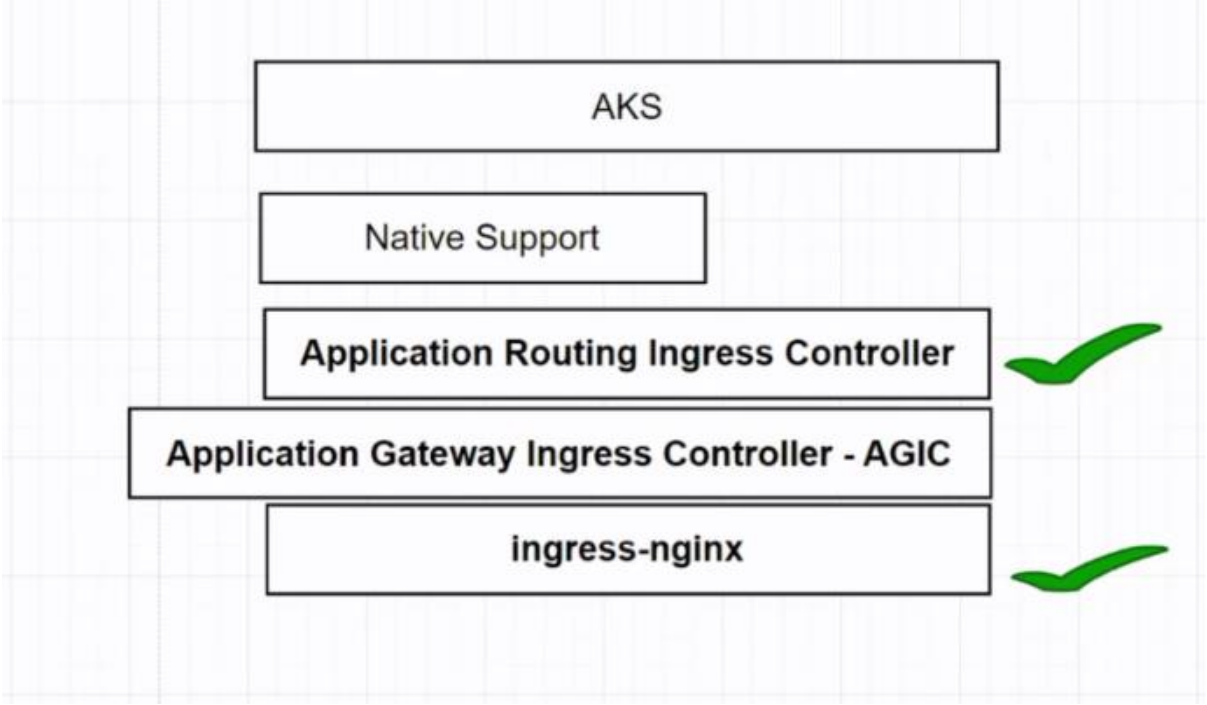
17 Nov 2024

AGENDA -



1) Create folder “9) 17 November Kubernetes” and open with vs code and connect to cluster.

az login

az account set --subscription 48f88df7-0d53-4866-a66f-82eb0ac469e3

az aks get-credentials --resource-group rgcloud --name k8scloud --overwrite-existing

2) ENABLE - enable AGIC in aks

https://learn.microsoft.com/en-us/azure/application-gateway/tutorial-ingress-controller-add-on-existing

Enable the AGIC add-on in existing AKS cluster through Azure CLI

If you'd like to continue using Azure CLI, you can continue to enable the AGIC add-on in the AKS cluster you created, **myCluster**, and specify the AGIC add-on to use the existing application gateway you created, **myApplicationGateway**.

Azure CLI | Copy | Open Cloud Shell

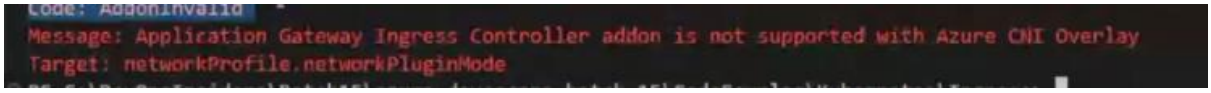
```
appgwId=$(az network application-gateway show --name myApplicationGateway --res
az aks enable-addons --name myCluster --resource-group myResourceGroup --addon
```

---

az aks enable-addons --name k8scloud --resource-group rgcloud --addon ingress-appgw --appgw-id $appgwId

az aks enable-addons --name k8scloud --resource-group rgcloud --addon ingress-appgw --appgw-subnet-cidr 10.224.0.0/16

az aks enable-addons --name k8scloud --resource-group rgcloud --addon ingress-appgw --appgw-subnet-cidr 10.225.0.0/16



So we have create new cluster as this cluster is not supported.

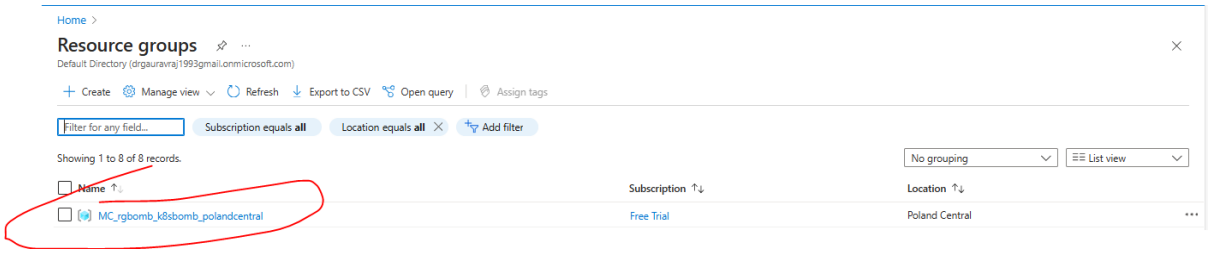
3) SEARCH – azure aks networking plugins

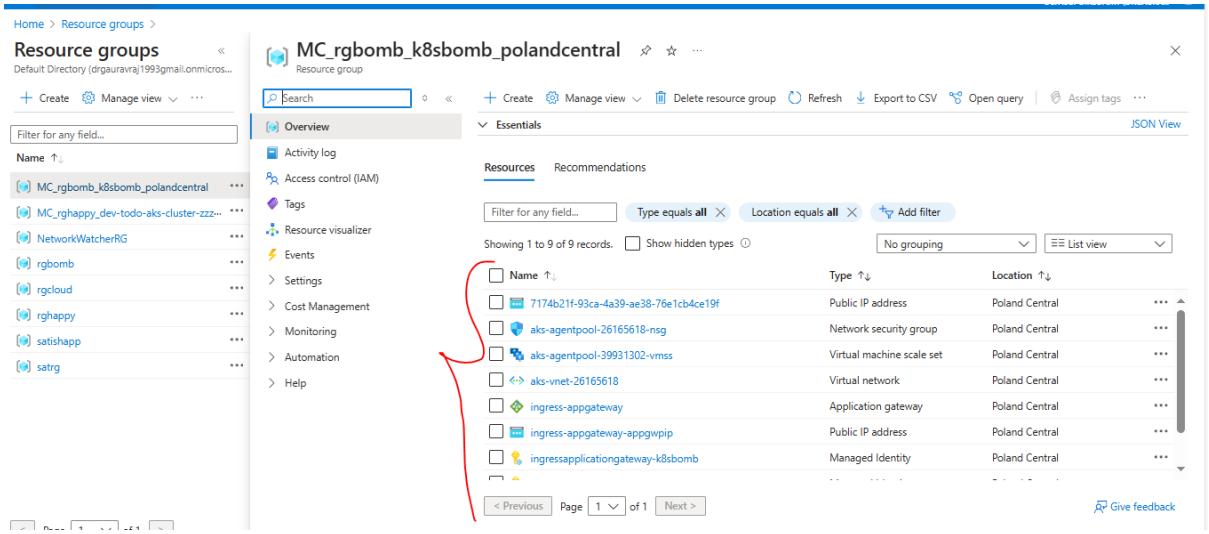
NOTE: 1) How do we expose anything in cluster = through loadbalancer service

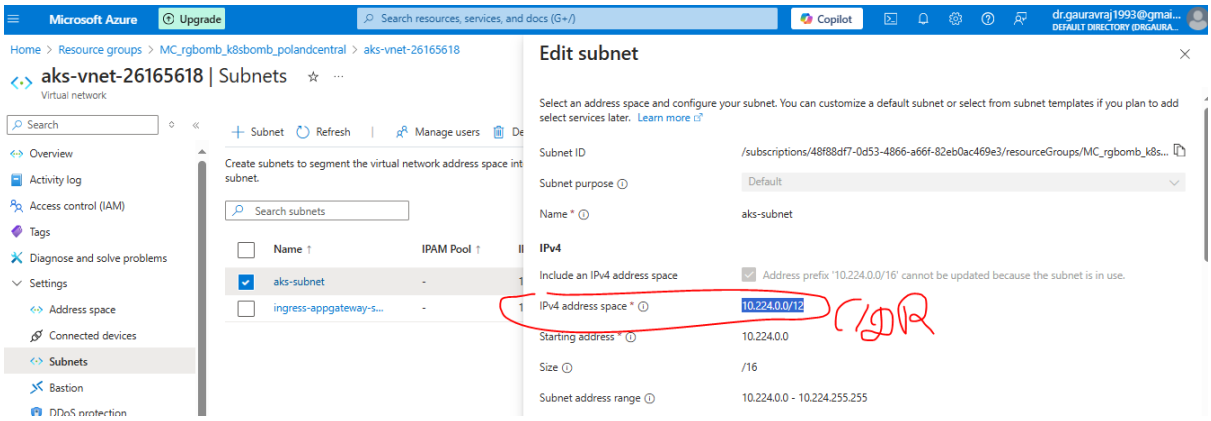
2) Limitations of loadbalancer service = for an application only one ip can be used. So for resolving this issue we had brought ingress controller

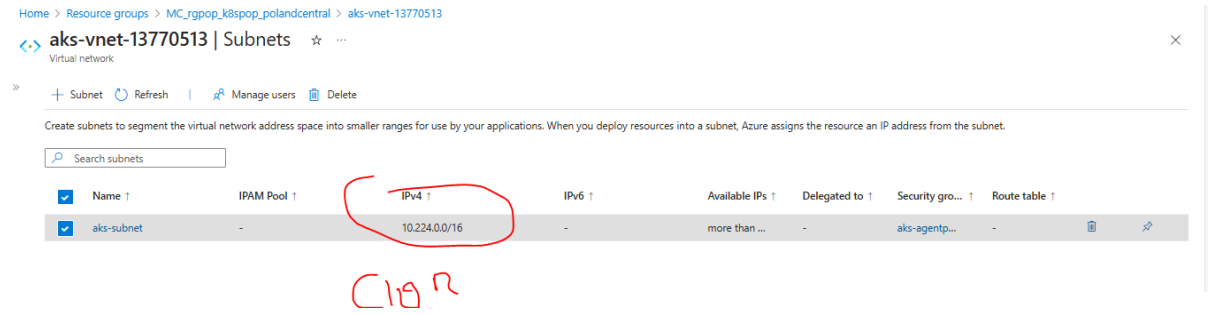
3) Calico = supports network policy

4) With our cluster an extra rg gets created









5) **az aks enable-addons --name k8spop --resource-group rgpop --addon ingress-appgw --appgw-subnet-cidr 10.224.0.0/16** = run to add application gateway

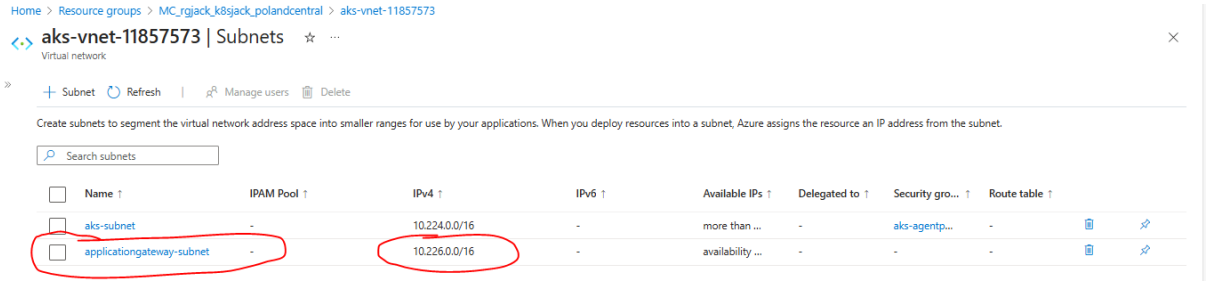
Ab isme dusra cide banalo i.e. 224 ko 226 krdo

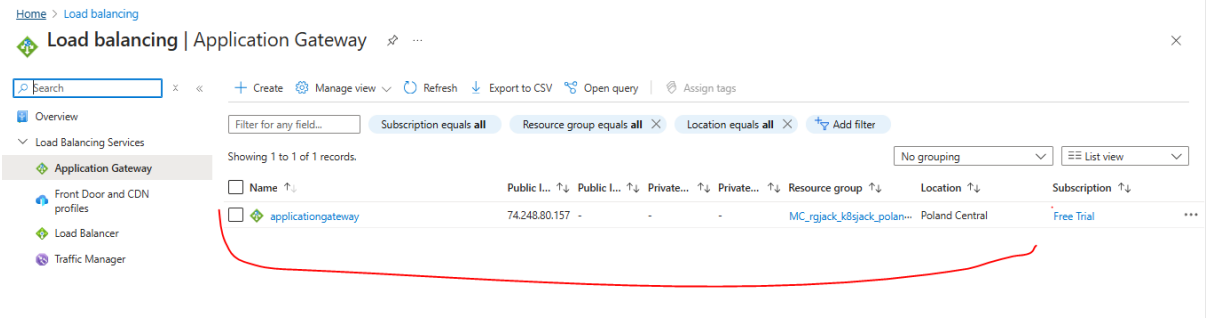
az aks enable-addons --name k8sjack --resource-group rgjack --addon ingress-appgw --appgw-subnet-cidr 10.226.0.0/16

```
PS C:\4) KUBERNETES\9) 17 November Kubernetes> az aks enable-addons --name k8sjack --resource-group rgjack --addon ingress-appgw --appgw-subnet-cidr 10.226.0.0/16
- Running ..
```

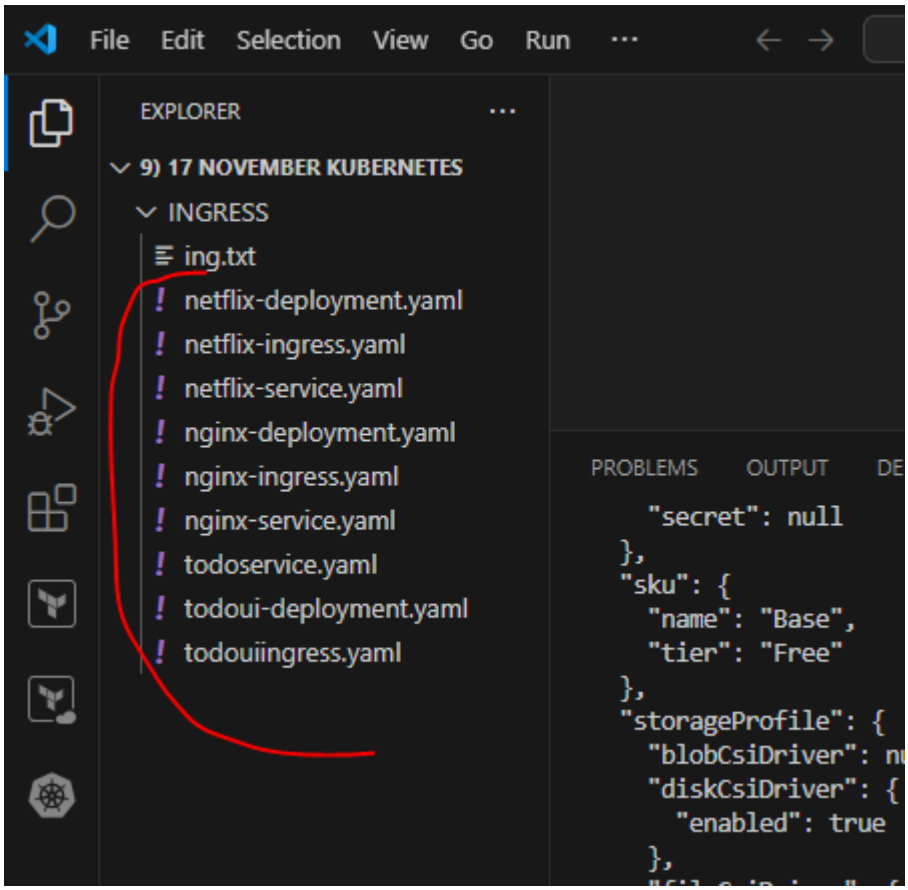
NOTE: Application gateway gets created on its personal subnet only

6) So now application gateway got created





7) Copy last class content in today's class



++

AGENDA – DO NETFLIX DEPLOYMENT

8) **kubectl apply -f netflix-deployment.yaml**

```
PS C:\4) KUBERNETES\9) 17 November Kubernetes\INGRESS> kubectl apply -f netflix-deployment.yaml
deployment.apps/netflix created
PS C:\4) KUBERNETES\9) 17 November Kubernetes\INGRESS>
```

9) **kubectl apply -f netflix-ingress.yaml**

```
PS C:\4) KUBERNETES\9) 17 November Kubernetes\INGRESS> kubectl apply -f netflix-ingress.yaml
ingress.networking.k8s.io/netflix-rule created
PS C:\4) KUBERNETES\9) 17 November Kubernetes\INGRESS>
```

10) **kubectl apply -f netflix-service.yaml**

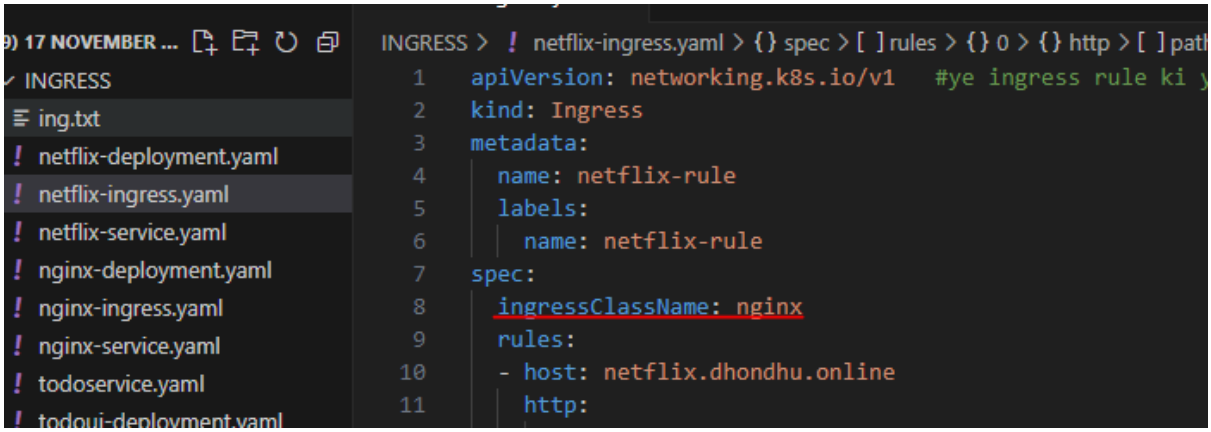
```
PS C:\4) KUBERNETES\9) 17 November Kubernetes\INGRESS> kubectl apply -f netflix-service.yaml
service/netflix created
```

11) **kubectl get ingressclass** =

```
PS C:\4) KUBERNETES\9) 17 November Kubernetes\INGRESS> kubectl get ingressclass
NAME                        CONTROLLER                  PARAMETERS   AGE
azure-application-gateway   azure/application-gateway   <none>       75m
PS C:\4) KUBERNETES\9) 17 November Kubernetes\INGRESS>
```

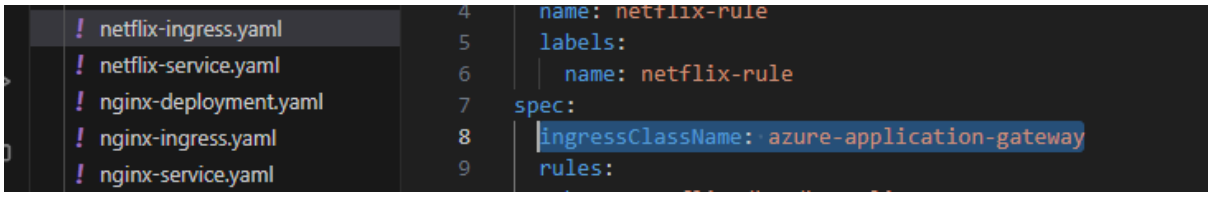
12) **kubectl get ingress** = isme ip address nhi hai

```
PS C:\4) KUBERNETES\9) 17 November Kubernetes\INGRESS> kubectl get ingress
NAME           CLASS   HOSTS                    ADDRESS   PORTS   AGE
netflix-rule   nginx   netflix.dhondhu.online             80      3m33s
PS C:\4) KUBERNETES\9) 17 November Kubernetes\INGRESS>
```



So upar dono commands se pata chala ki humne code me class name – nginx pass kiya hai but "**kubectl get ingressclass**" se pata chala ki real me class ka name "**azure-application-gateway**" hai, to hum is name ko code me pass krenge else purana class rakhne se wo kisi controller se connect nhi hoga to wo faltu pada rhega.

13) So pass **ingressClassName** as **azure-application-gateway** in code



14) **kubectl delete -f netflix-ingress.yaml** = delete old ingress rule of Netflix

```
PS C:\4) KUBERNETES\9) 17 November Kubernetes\INGRESS> kubectl delete -f netflix-ingress.yaml
ingress.networking.k8s.io "netflix-rule" deleted
PS C:\4) KUBERNETES\9) 17 November Kubernetes\INGRESS>
```

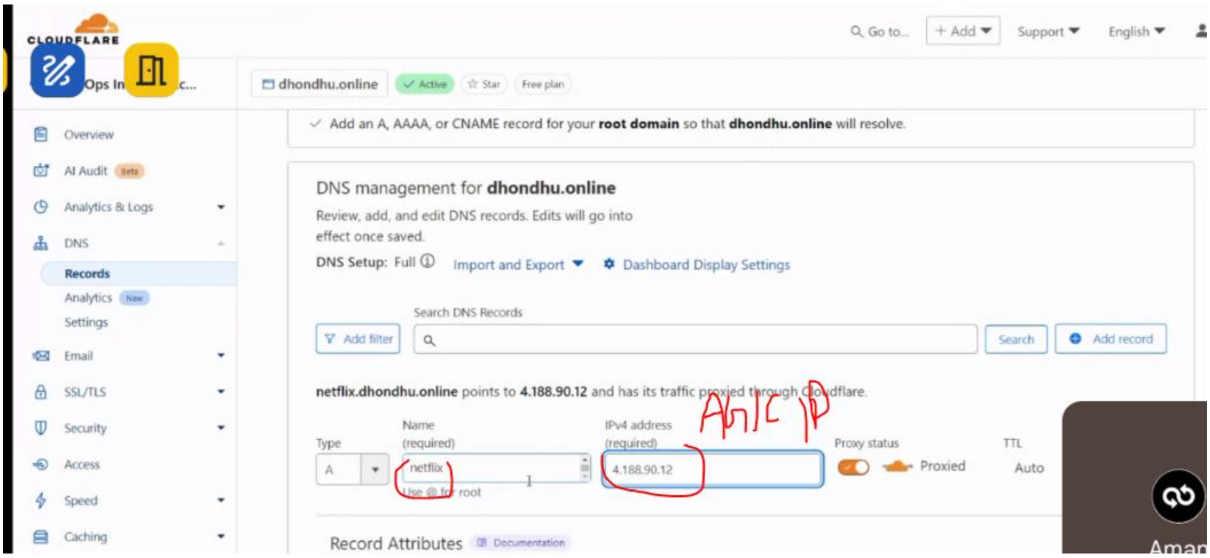
15) **kubectl apply -f netflix-ingress.yaml** = again create with **ingressClassName** as **azure-application-gateway** in code

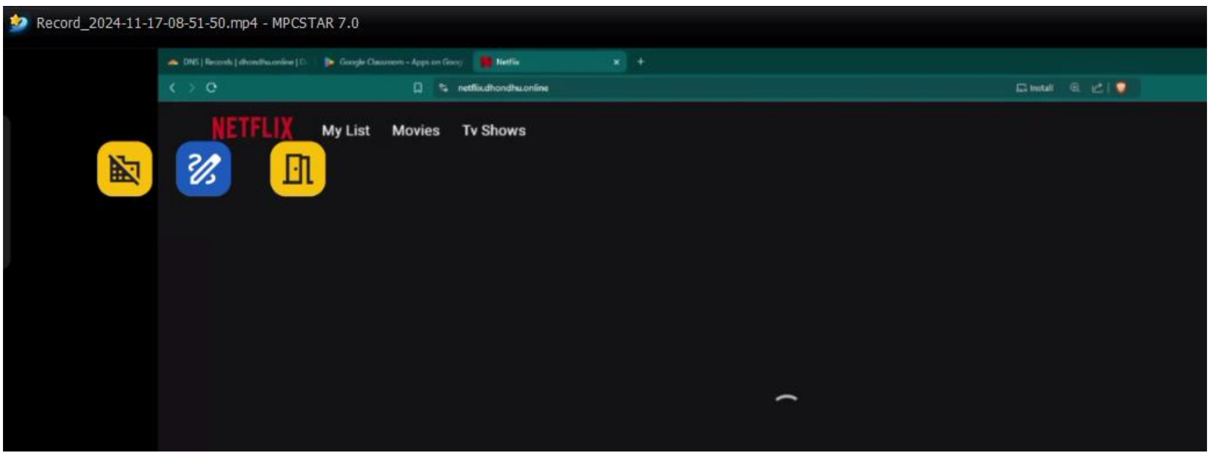
```
PS C:\4) KUBERNETES\9) 17 November Kubernetes\INGRESS> kubectl apply -f netflix-ingress.yaml
ingress.networking.k8s.io/netflix-rule created
PS C:\4) KUBERNETES\9) 17 November Kubernetes\INGRESS>
```

16) **kubectl get ingress** = so ab ip address mil gaya hai humare Netflix rule ko

```
PS C:\4) KUBERNETES\9) 17 November Kubernetes\INGRESS> kubectl get ingress
NAME           CLASS                       HOSTS                    ADDRESS         PORTS   AGE
netflix-rule   azure-application-gateway   netflix.dhondhu.online   74.248.80.157   80      66s
PS C:\4) KUBERNETES\9) 17 November Kubernetes\INGRESS>
```

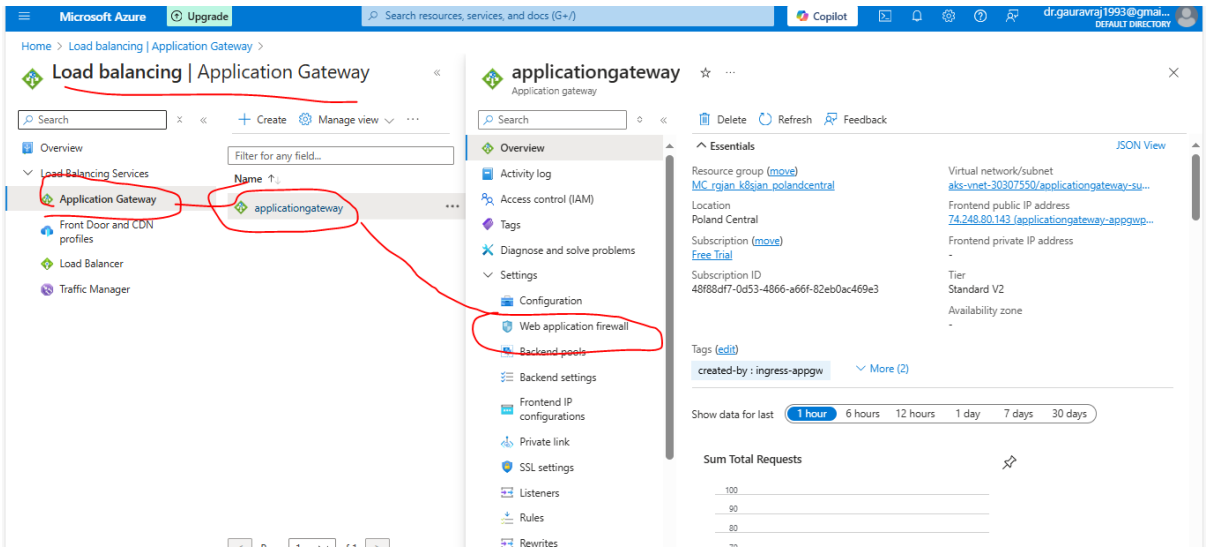
17) Go to cloud flare and we can see that AGIC is running

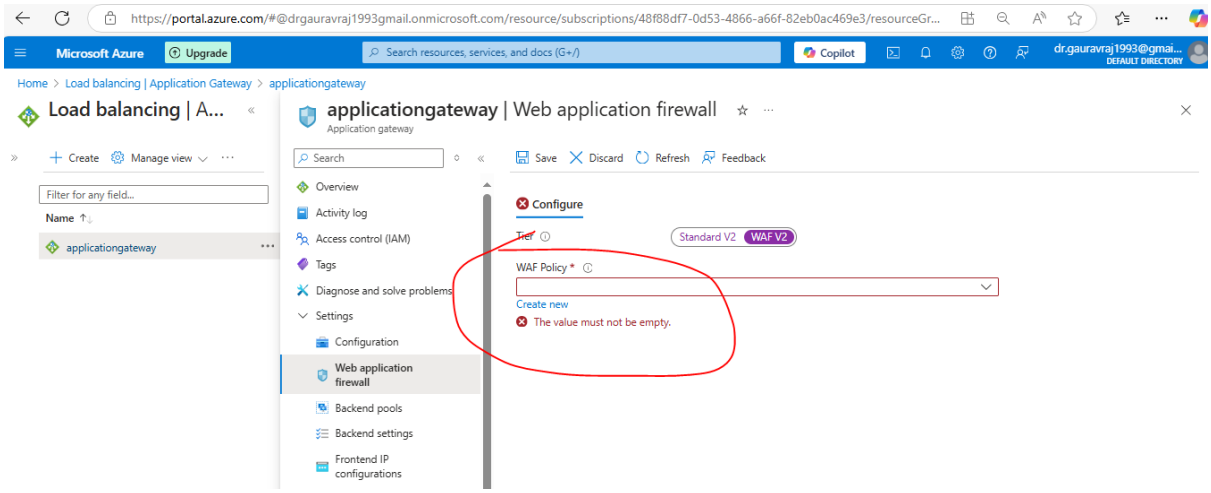




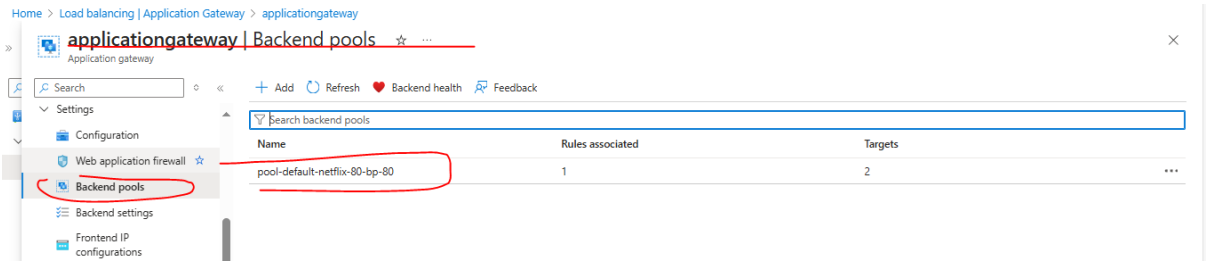
++

AGENDA = How to enable WAF on application gateway loadbalancer

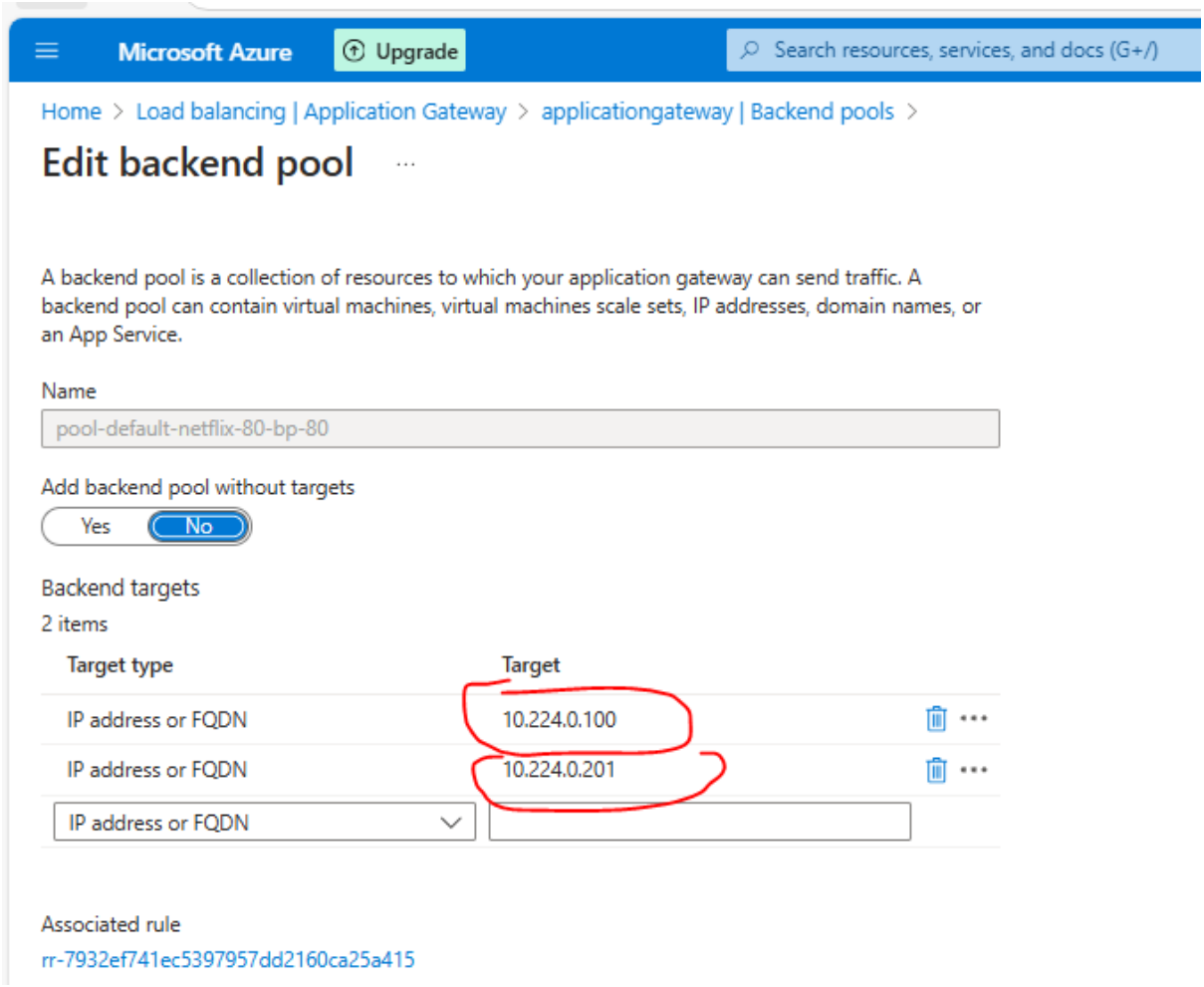




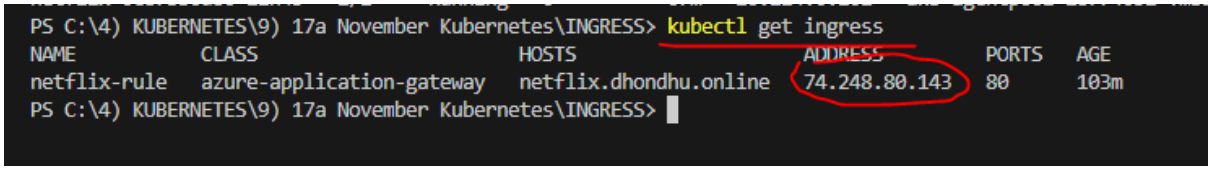
1) In Backend pools section, we have pod ip addresses contained in it

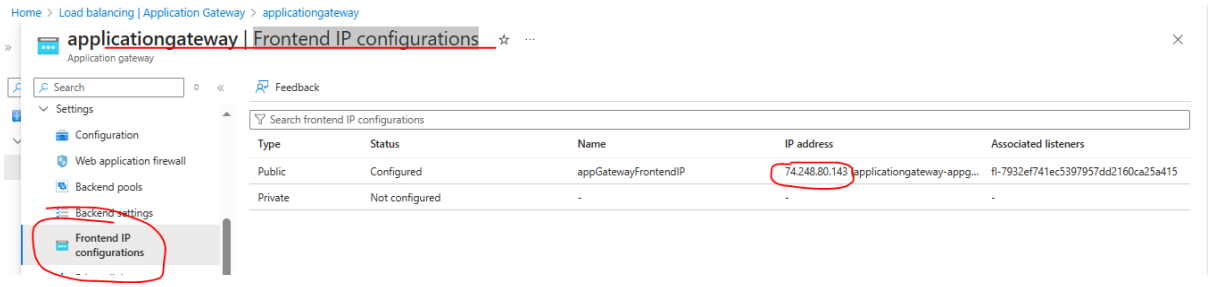



```
PS C:\4) KUBERNETES\9) 17a November Kubernetes\INGRESS> kubectl get pods -o wide
NAME                       READY   STATUS    RESTARTS   AGE   IP             NODE                                NOMINATED NODE   READINESS GATES
netflix-6c87bb6d86-v4tr9   1/1     Running   0          67m   10.224.0.100   aks-agentpool-18774052-vmss000003   <none>           <none>
netflix-6c87bb6d86-z2h45   1/1     Running   0          67m   10.224.0.201   aks-agentpool-18774052-vmss000002   <none>           <none>
PS C:\4) KUBERNETES\9) 17a November Kubernetes\INGRESS>
```

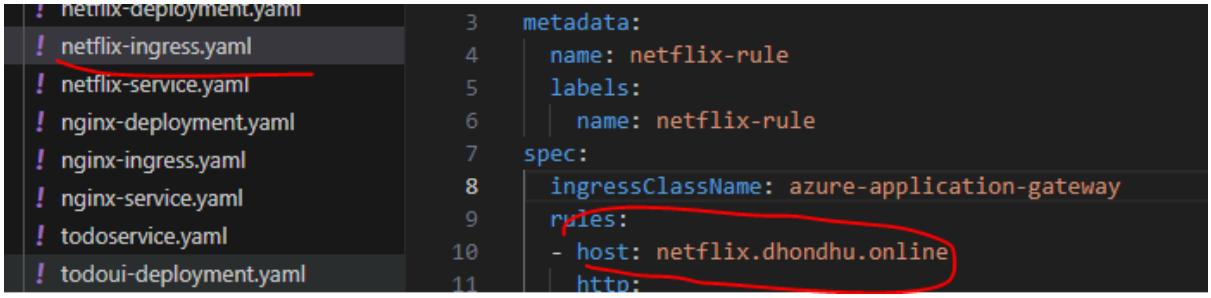


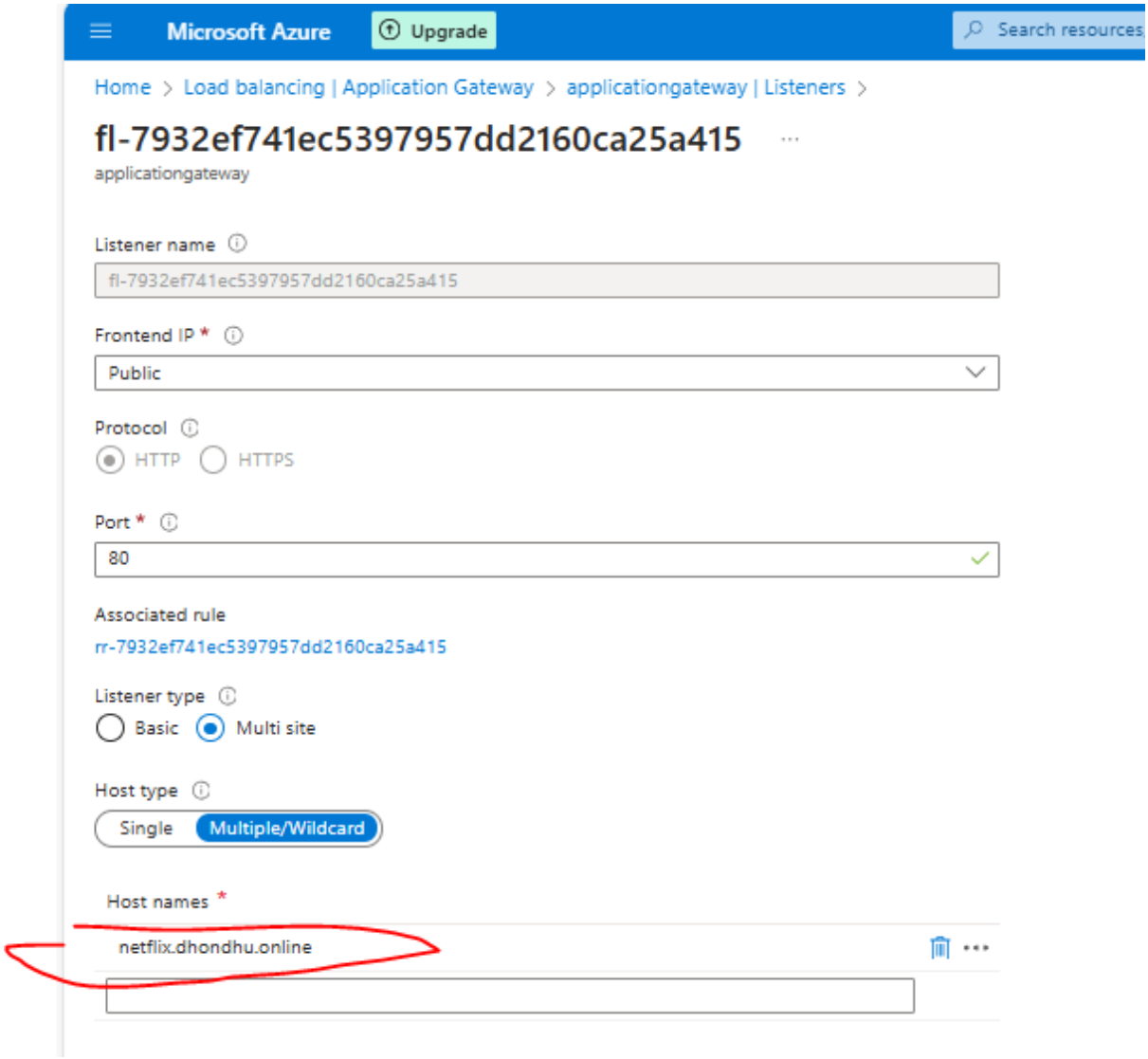
2) In “Frontend IP configurations” section, ingress ip is configured





3) In “Listeners” section host url is configured





++

AGENDA = todoui application deployment

1) **kubectl apply -f todoui-deployment.yaml**

```
PS C:\4) KUBERNETES\9) 17a November Kubernetes\INGRESS> kubectl apply -f todoui-deployment.yaml
deployment.apps/todoui created
PS C:\4) KUBERNETES\9) 17a November Kubernetes\INGRESS>
```

2) **kubectl apply -f todoservice.yaml**

```
PS C:\4) KUBERNETES\9) 17a November Kubernetes\INGRESS> kubectl apply -f todoservice.yaml
service/todoui created
PS C:\4) KUBERNETES\9) 17a November Kubernetes\INGRESS>
```

3) **kubectl apply -f todouiingress.yaml**

```
PS C:\4) KUBERNETES\9) 17a November Kubernetes\INGRESS> kubectl apply -f todouiingress.yaml
ingress.networking.k8s.io/todoui-ingress created
PS C:\4) KUBERNETES\9) 17a November Kubernetes\INGRESS>
```

4) **kubectl get ingressclass**

```
PS C:\4) KUBERNETES\9) 17a November Kubernetes\INGRESS> kubectl get ingressclass
NAME                        CONTROLLER                  PARAMETERS   AGE
azure-application-gateway   azure/application-gateway   <none>       136m
PS C:\4) KUBERNETES\9) 17a November Kubernetes\INGRESS>
```

5) **Kubectl get ingress** = So no ip is assigned as "**kubectl get ingressclass**" has different class name so pass that in yaml code i.e. "azure-application-gateway"

```
NAME            CLASS                                HOSTS                   ADDRESS         PORTS   AGE
netflix-rule    azure-application-gateway            netflix.dhondhu.online  74.248.80.143   80      123m
todoui-ingress  webapprouting.kubernetes.azure.com   todoui.dhondhu.online                   80      4m46s
```

```
! nginx-ingress.yaml
! nginx-service.yaml
! todoservice.yaml
! todoui-deployment.yaml
! todouiingress.yaml

spec:
  ingressClassName: webapprouting.kubernetes.azure.com
  rules:
  - host: todoui.dhondhu.online
    http:
      paths:
      - pathType: Prefix
```

Change it as below

```
! nginx-deployment.yaml
! nginx-ingress.yaml
! nginx-service.yaml
! todoservice.yaml
! todoui-deployment.yaml
! todouiingress.yaml

    name: todoui-ingress
spec:
  ingressClassName: azure-application-gateway
  rules:
  - host: todoui.dhondhu.online
    http:
      paths:
      - pathType: Prefix
        path: "/"
```

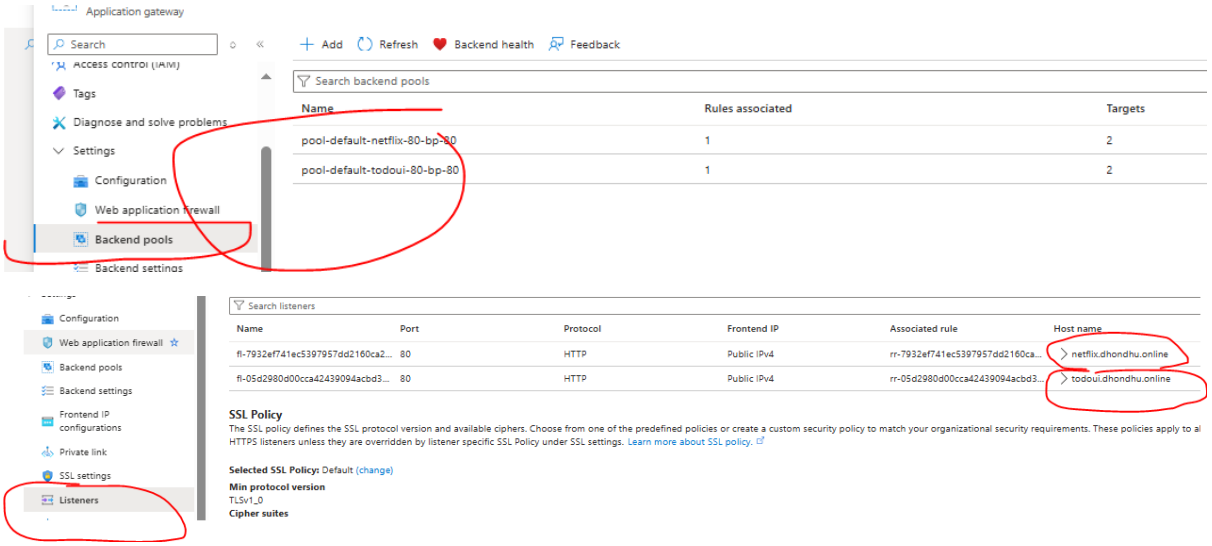
6) **kubectl apply -f todouiingress.yaml**

```
PS C:\4) KUBERNETES\9) 17a November Kubernetes\INGRESS> kubectl apply -f todouiingress.yaml
ingress.networking.k8s.io/todoui-ingress created
```

7) **kubectl get ingress**

```
PS C:\4) KUBERNETES\9) 17a November Kubernetes\INGRESS> kubectl get ingress
NAME            CLASS                       HOSTS                   ADDRESS         PORTS   AGE
netflix-rule    azure-application-gateway   netflix.dhondhu.online  74.248.80.143   80      133m
todoui-ingress  azure-application-gateway   todoui.dhondhu.online   74.248.80.143   80      9s
PS C:\4) KUBERNETES\9) 17a November Kubernetes\INGRESS>
```

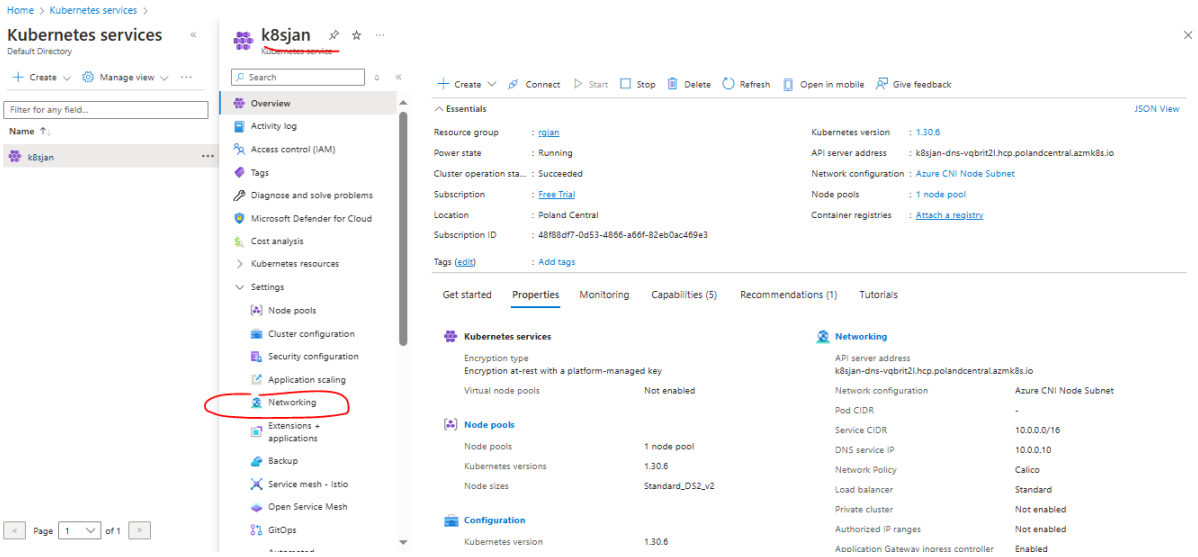
8) So now we can see in portal we have got backend pool, listener, Frontend IP configurations for todoui application also

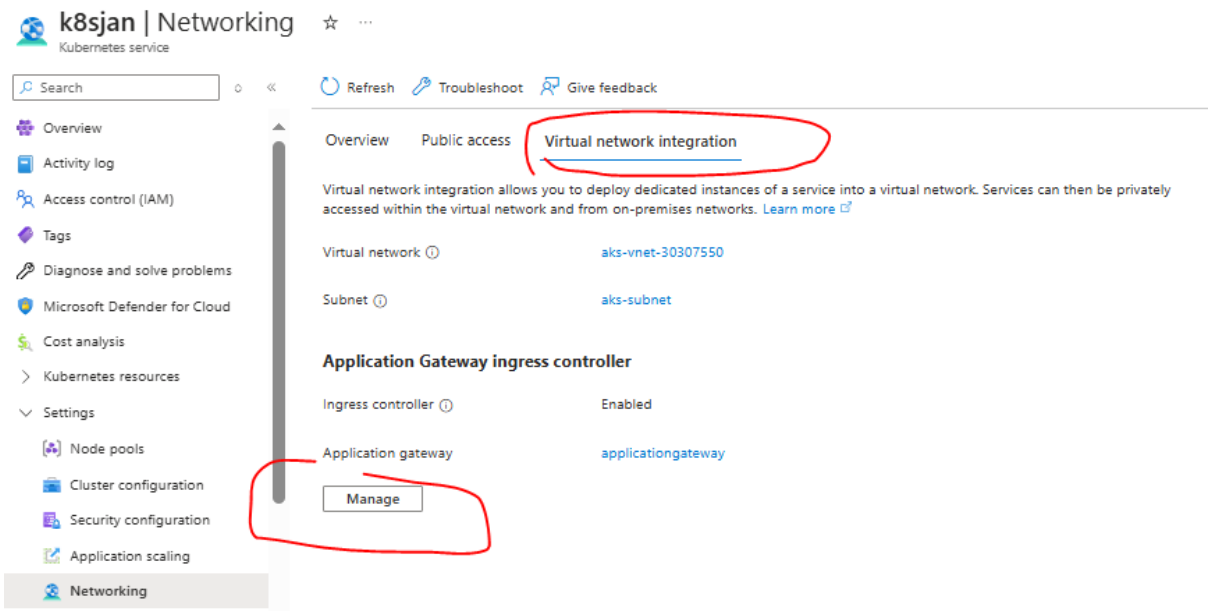


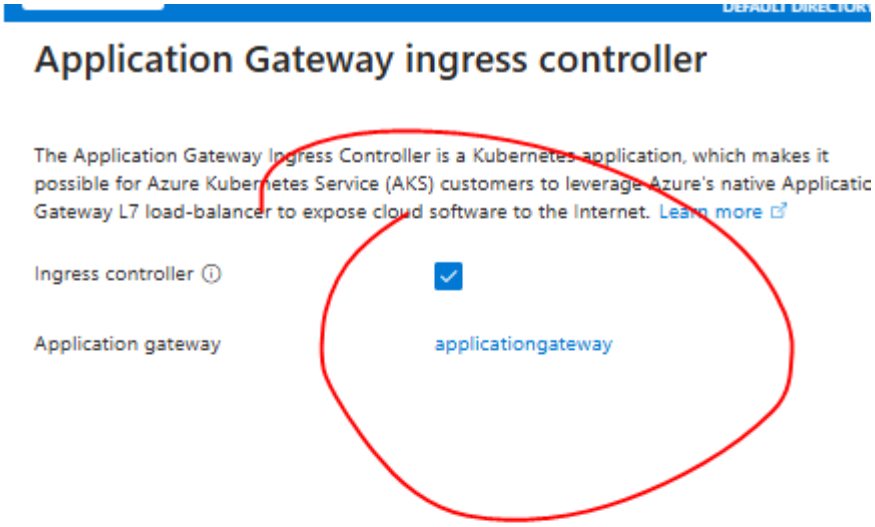
++

AGENDA – ADDING AGIC INGRESS CONTROLLER THROUGH PORTAL MANUALLY

1) Go to aks cluster -> Networking -> virtual networking configuration -> manage -> tick ingress controller option -> save




k8sjan | Networking
Kubernetes service
Search
Refresh
Troubleshoot
Give feedback
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Microsoft Defender for Cloud
Cost analysis
Kubernetes resources
Settings
Node pools
Cluster configuration
Security configuration
Application scaling
Networking
Overview
Public access
Virtual network integration
Virtual network integration allows you to deploy dedicated instances of a service into a virtual network. Services can then be privately accessed within the virtual network and from on-premises networks. Learn more
Virtual network
aks-vnet-30307550
Subnet
aks-subnet
Application Gateway ingress controller
Ingress controller
Enabled
Application gateway
applicationgateway
Manage



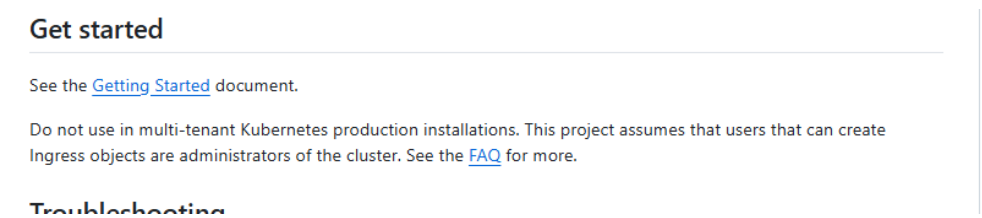
++

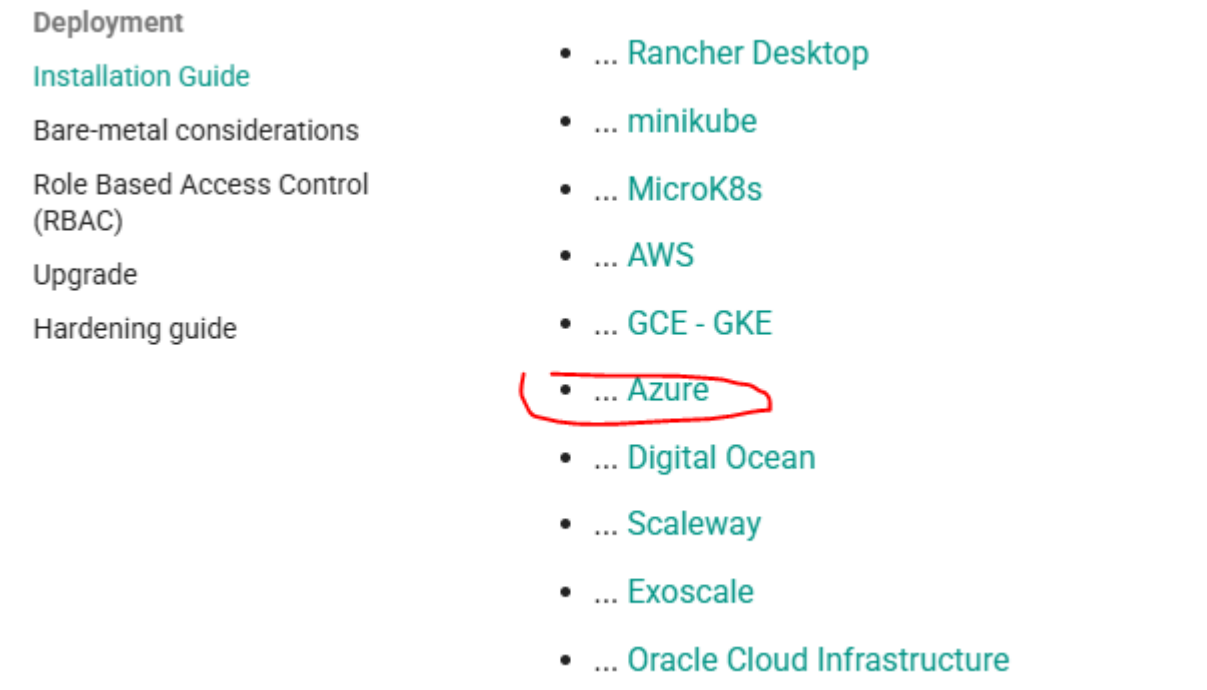
AGENDA – RAN NGINX CONTROLLER ALSO

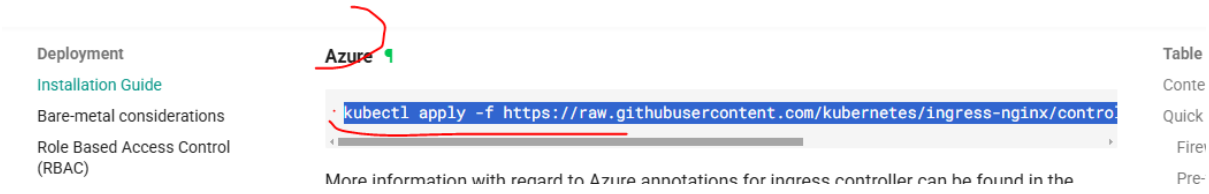
1) SEARCH – ingress- nginx

https://github.com/kubernetes/ingress-nginx

scroll down

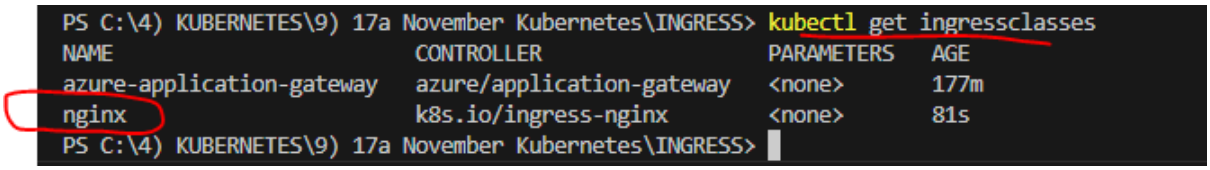






`kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.12.0/deploy/static/provider/cloud/deploy.yaml = this command runs our nginx controller`

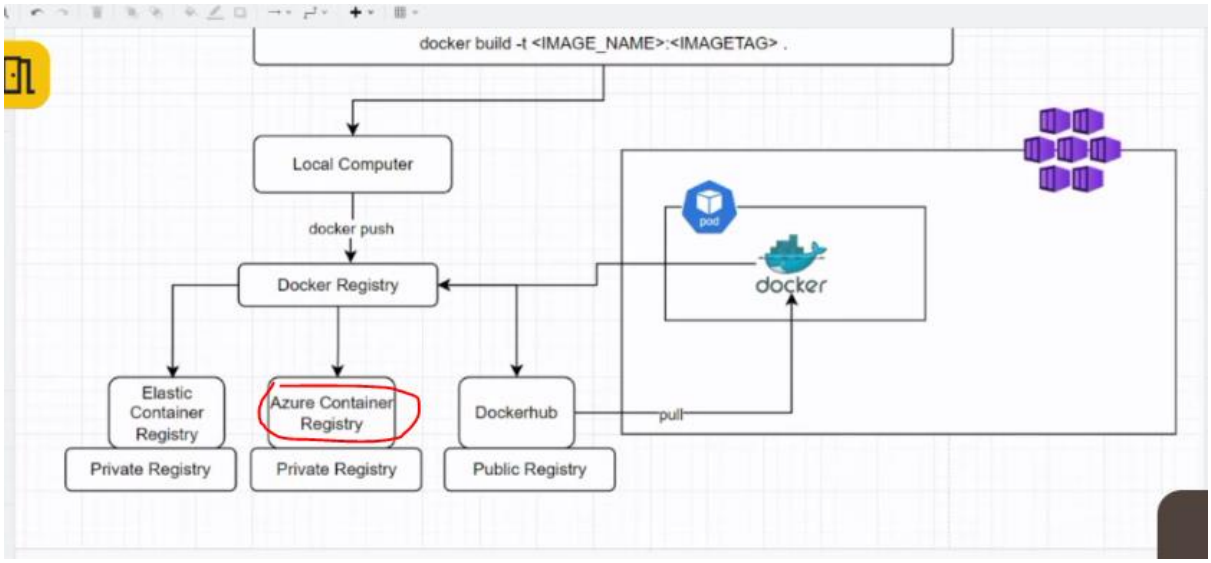
`2) kubectl get ingressclasses`



++

AGENDA – docker

1) If our docker image is kept in local computer then we have to push that into different docker registries



INTERVIEW – We never have to say that we push our image to docker hub as it is public registry. Instead we have to say that we push our image to "Azure container registry" which is a private registry.