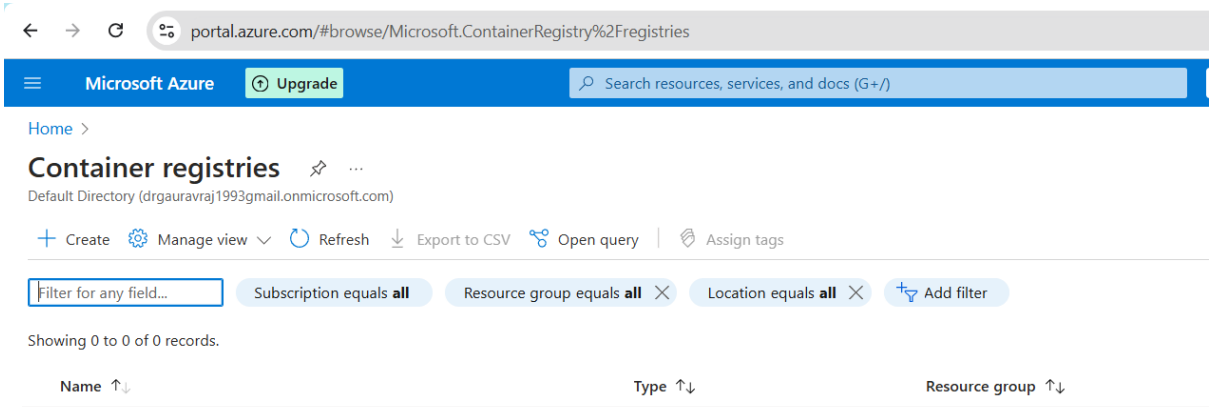
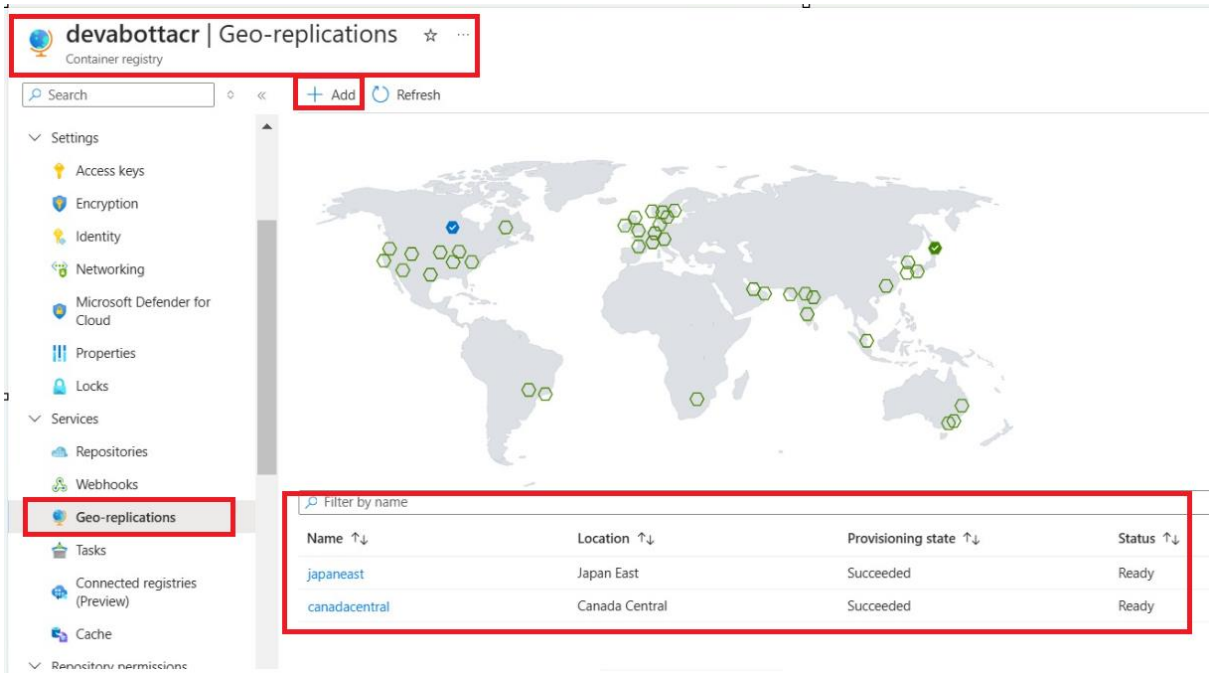
DR Implementation In AKS Cluster

Step1: Create Two AKS Cluster in Different Region

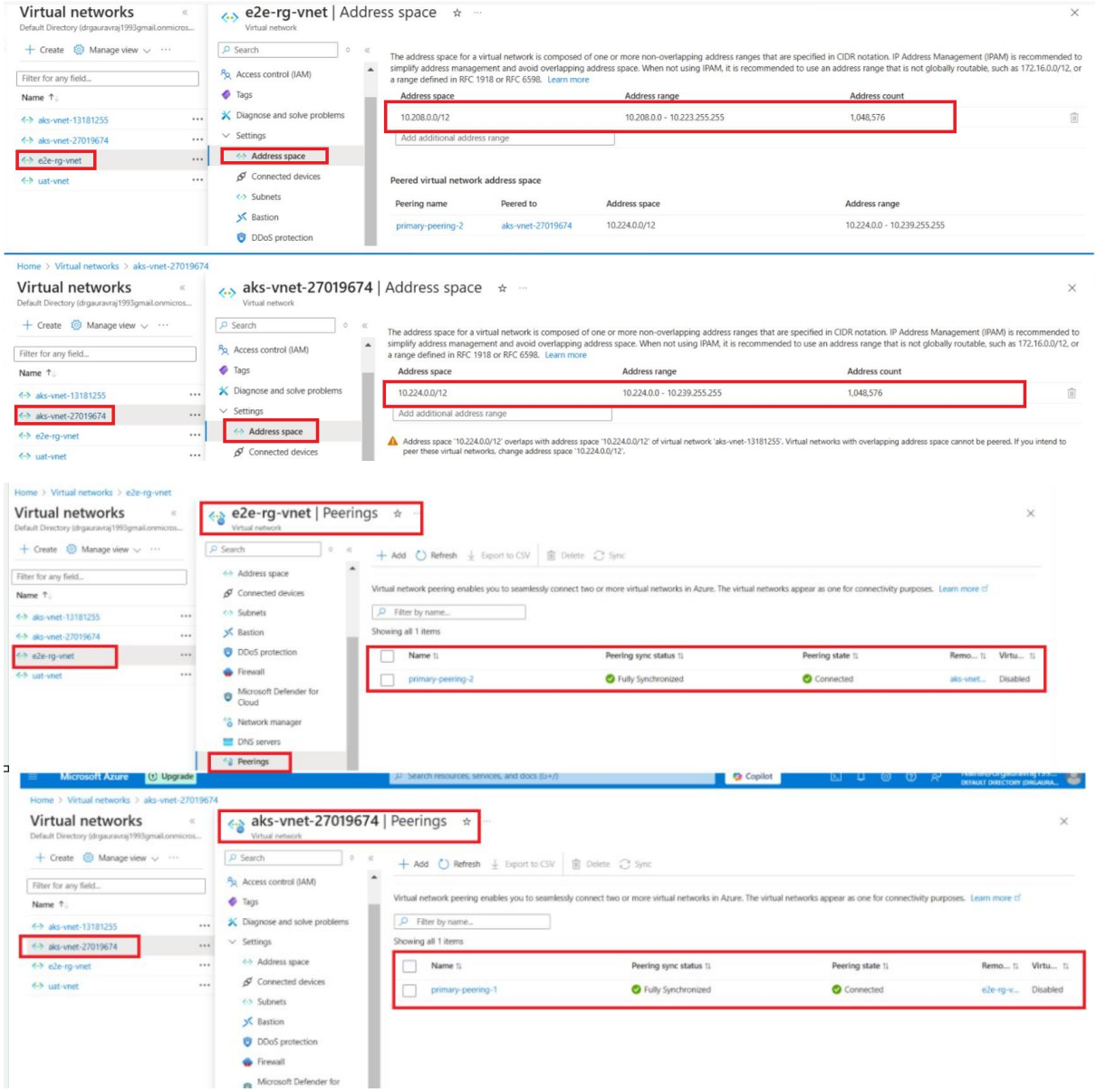
a. **Primary-DR-aks with 1 zone and region is Canada Central**
b. **Secondary-DR-aks with 1 zone and region is Japan East**

Step2: Create on ACR in region Canada Central and add Geo-Geographical location for Japan East.





Step3: Perform VNET Peering with both in different cluster.



Step4: Enable ingress controller in both AKS

Step5: Run Below command In both AKS to make public IP Static

Note: Run Below Step in Both AKS Cluster.

a. (Optional) First of all, check public ip of both aks (1 by 1) with below commands
az network public-ip list --resource-group MC_e2e-rg_primary-aks-cluster_japaneast --query "[].{Name:name, IPAddress:ipAddress}" -o table

b. Then,make static public ip and dns name with below command:
az network public-ip update --resource-group MC_e2e-rg_primary-aks-cluster_japaneast --name ingress-appgateway-appgwpip --dns-name ingress-appgateway --allocation-method Static --sku Standard

c. Then check dns name by below command:

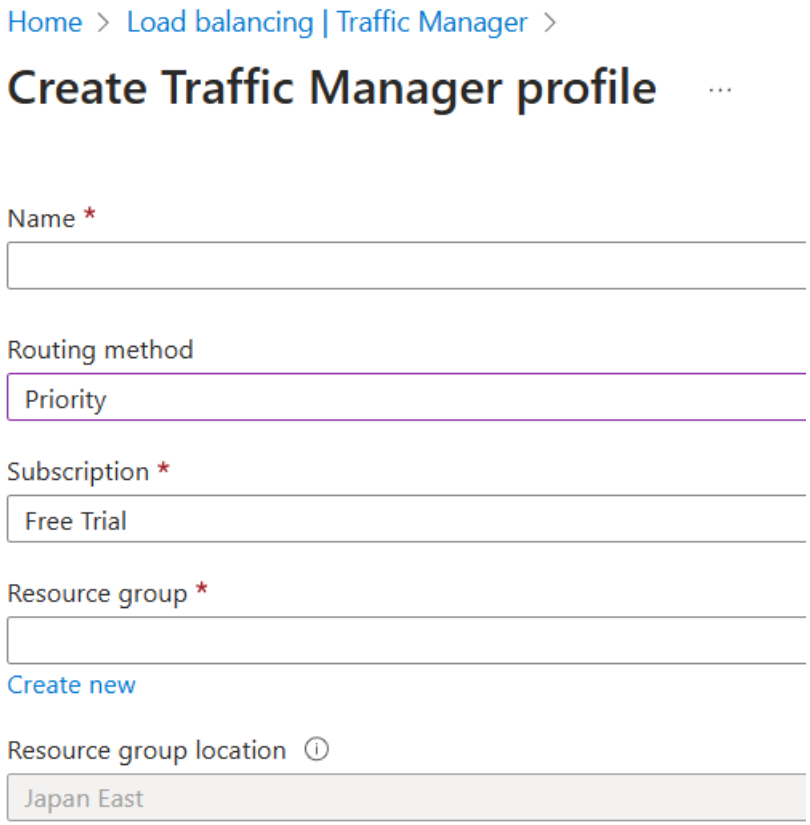
az network public-ip show --resource-group MC_e2e-rg_primary-aks-cluster_japaneast --name ingress-appgateway-appgwpip --query "dnsSettings.fqdn" -o tsv

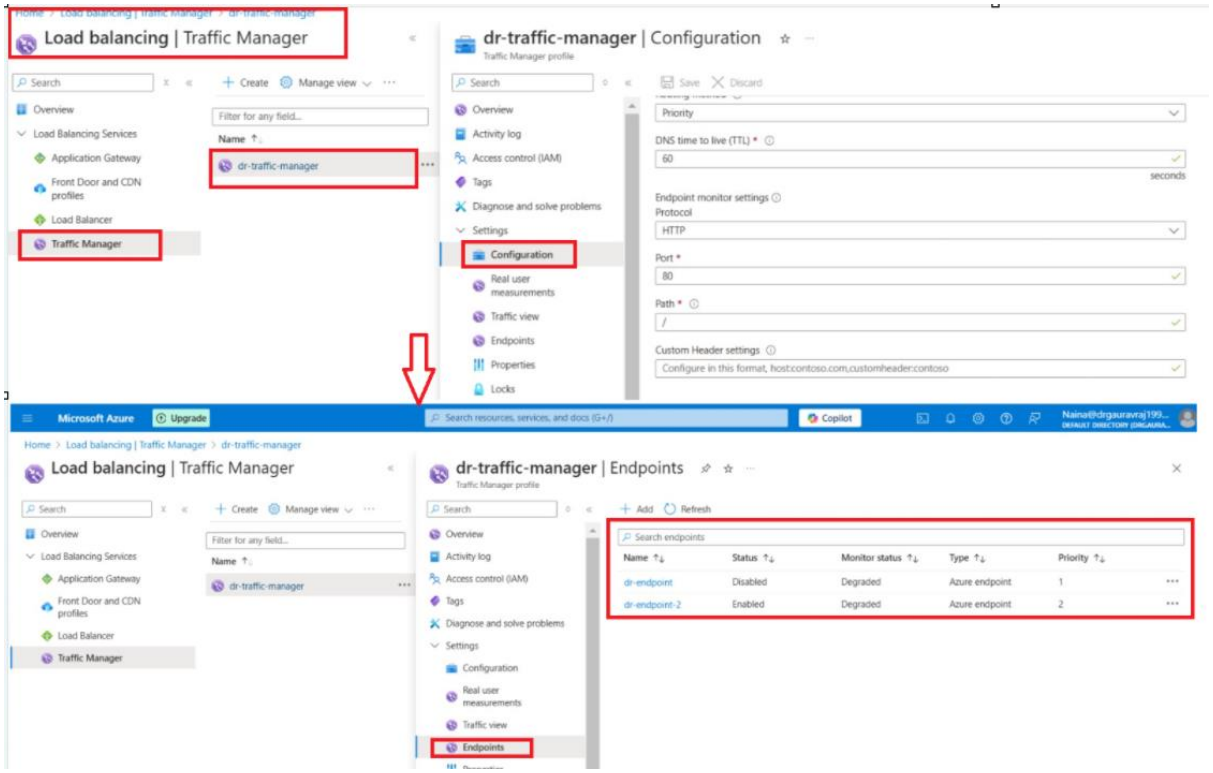
d. (optional)Then again verify public ips list
az network public-ip list --resource-group MC_e2e-rg_primary-aks-cluster_japaneast --query "[].{Name:name, IPAddress:ipAddress}" -o table

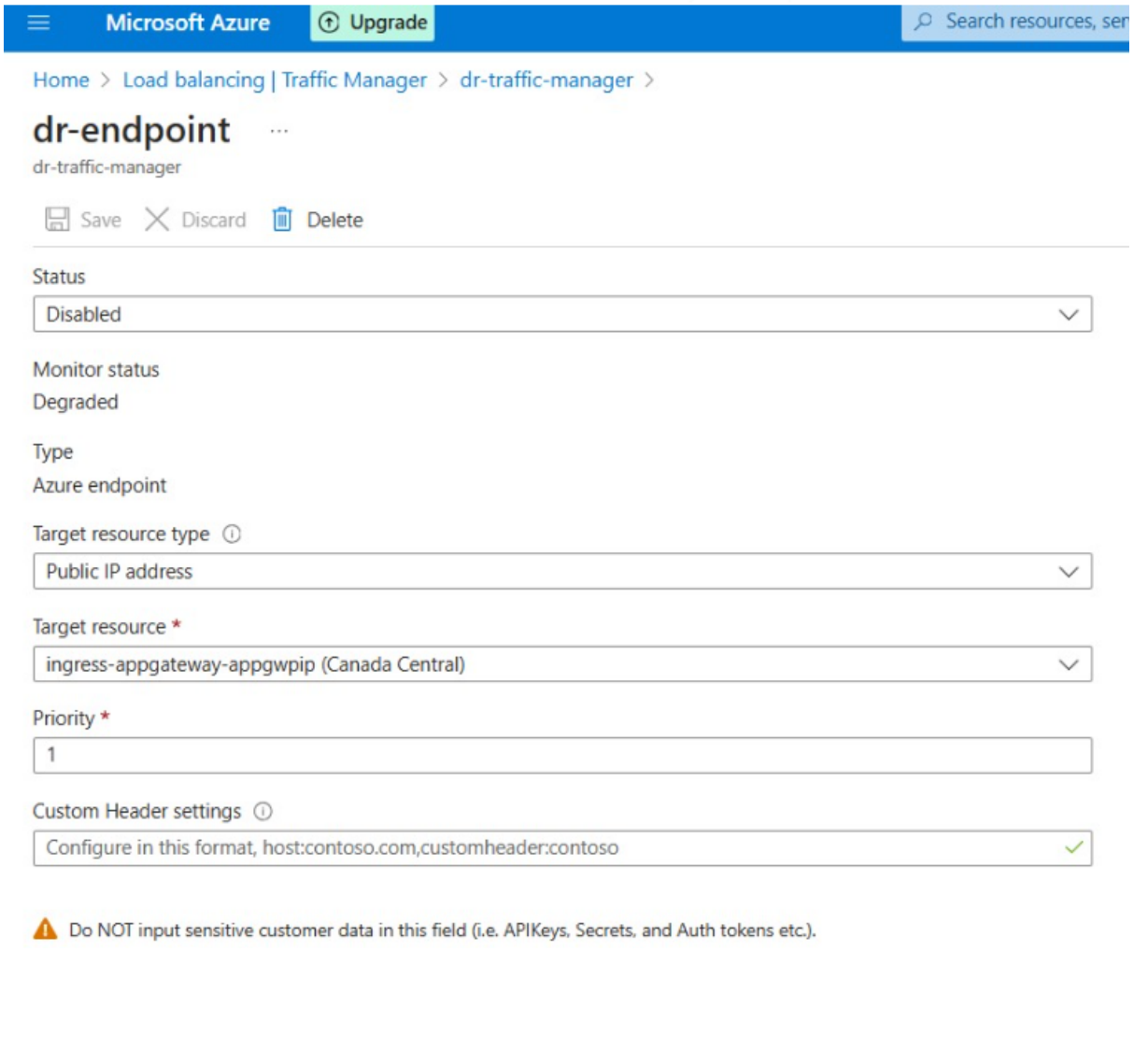
```
nancy@DESKTOP-7M9SJH4:~/DEVOPS/KUBERNETES/DR$ code .
nancy@DESKTOP-7M9SJH4:~/DEVOPS/KUBERNETES/DR$ az network public-ip list --resource-group MC_e2e-rg_primary-aks-cluster_japaneast --qu
ery "[].{Name:name, IPAddress:ipAddress}" -o table
Name                                  IPAddress
------------------------------------  -------------
5dc4c2ff-7dca-4cc6-8594-e4f4d7afd32e  74.176.25.158
ingress-appgateway-appgwpip           74.176.152.13
nancy@DESKTOP-7M9SJH4:~/DEVOPS/KUBERNETES/DR$ az network public-ip show   --resource-group MC_e2e-rg_primary-aks-cluster_japaneast
--name ingress-appgateway-appgwpip   --query "dnsSettings.fqdn" -o tsv
ingress-appgateway.japaneast.cloudapp.azure.com
nancy@DESKTOP-7M9SJH4:~/DEVOPS/KUBERNETES/DR$
az network public-ip show   --resource-group MC_e2e-rg_primary-aks-cluster_japaneast   --name ingress-appgateway-appgwpip   --query "
dnsSettings.fqdn" -o tsv
ingress-appgateway.japaneast.cloudapp.azure.com
nancy@DESKTOP-7M9SJH4:~/DEVOPS/KUBERNETES/DR$ az network public-ip list --resource-group MC_e2e-rg_primary-aks-cluster_japaneast --qu
ery "[].{Name:name, IPAddress:ipAddress}" -o table
Name                                  IPAddress
------------------------------------  -------------
5dc4c2ff-7dca-4cc6-8594-e4f4d7afd32e  74.176.25.158
ingress-appgateway-appgwpip           74.176.152.13
nancy@DESKTOP-7M9SJH4:~/DEVOPS/KUBERNETES/DR$
```

Step6: Perform Below Step for Creating traffic Manager:

a. Create Traffic Manager with Name, RG, location, Routing(Priority) etc.
b. Add endpoint: name, add 1st dns name(generate above), give priority 1, add
c. Add endpoint: name, add 2nd dns name(generate above), give priority 2, add
d. In configuration > add path (if applicable)




Load balancing | Traffic Manager
dr-traffic-manager | Configuration
Traffic Manager profile
Overview
Load Balancing Services
Application Gateway
Front Door and CDN profiles
Load Balancer
Traffic Manager
Create
Manage view
Filter for any field...
Name
dr-traffic-manager
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Settings
Configuration
Real user measurements
Traffic view
Endpoints
Properties
Locks
Save
Discard
Priority
DNS time to live (TTL) *
60
seconds
Endpoint monitor settings
Protocol
HTTP
Port *
80
Path *
/
Custom Header settings
Configure in this format, host:contoso.com,customheader:contoso
Microsoft Azure
Upgrade
Copilot
Home > Load balancing | Traffic Manager > dr-traffic-manager
dr-traffic-manager | Endpoints
Add
Refresh
Search endpoints
Name
Status
Monitor status
Type
Priority
dr-endpoint
Disabled
Degraded
Azure endpoint
1
dr-endpoint-2
Enabled
Degraded
Azure endpoint
2


Microsoft Azure
Upgrade
Search resources, ser
Home > Load balancing | Traffic Manager > dr-traffic-manager >
dr-endpoint
dr-traffic-manager
Save
Discard
Delete
Status
Disabled
Monitor status
Degraded
Type
Azure endpoint
Target resource type
Public IP address
Target resource *
ingress-appgateway-appgwpip (Canada Central)
Priority *
1
Custom Header settings
Configure in this format, host:contoso.com,customheader:contoso
Do NOT input sensitive customer data in this field (i.e. APIKeys, Secrets, and Auth tokens etc.).

Step7: traffic-manager dns name in godaddy with cname like “dr-traffic-manager.trafficmanager.net”

Step8: check by nslookup nslookup dr-traffic-manager.trafficmanager.net, if it is 1st aks ip or not

```
nancy@DESKTOP-7M9SJH4:~/DEVOPS/KUBERNETES/DR$ nslookup dr-traffic-manager.trafficmanager.net
Server:         10.255.255.254
Address:        10.255.255.254#53

Non-authoritative answer:
dr-traffic-manager.trafficmanager.net   canonical name = ingress-appgateway.japaneast.cloudapp.azure.com.
Name:   ingress-appgateway.japaneast.cloudapp.azure.com
Address: 74.176.152.13

nancy@DESKTOP-7M9SJH4:~/DEVOPS/KUBERNETES/DR$
```

Step9: Now Deploy Application in both AKS cluster, run deployment.yaml, services.yaml and ingress.yaml

```
EXPLORER                                  nginx-ingress.yaml    ngin-deploymnet.yaml >
DR [WSL: UBUNTU]                          ngin-deploymnet.yaml > {} spec > {} selector
  ngin-deploymnet.yaml                    1  apiVersion: apps/v1
  nginx-ingress.yaml                      2  kind: Deployment
  nginx-svc.yaml                          3  metadata:
                                          4    name: nginx-deployment
                                          5    labels:
                                          6      app: nginx
                                          7  spec:
                                          8    replicas: 1
                                          9    selector:
                                          10     matchLabels:
                                          11       app: nginx
                                          12   template:
                                          13     metadata:
                                          14       labels:
                                          15         app: nginx
                                          16     spec:
                                          17       containers:
                                          18       - name: nginx
                                          19         image: nginx:latest
                                          20         ports:
                                          21         - containerPort: 80
```

```yaml
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"networking.k8s.io/v1","kind":"Ingress","metadata":{"annotations":{},"nam
lt"},"spec":{"ingressClassName":"azure-application-gateway","rules":[{"host":"nginxdr.satishr
":{"service":{"name":"nginx-service-dr","port":{"number":80}}},"path":"/","pathType":"Prefix"
  creationTimestamp: "2024-12-31T09:18:42Z"
  generation: 4
  name: nginx-ingress-dr
  namespace: default
  resourceVersion: "161485"
  uid: 56e79e61-956a-4ea1-bc8c-618bdd059165
spec:
  ingressClassName: azure-application-gateway
  rules:
  - host: dr-traffic-manager.trafficmanager.net
    http:
      paths:
      - backend:
          service:
            name: nginx-service-dr
            port:
              number: 80
        path: /
        pathType: Prefix
status:
"/tmp/kubectl-edit-1586752864.yaml" 33L, 1227B
```

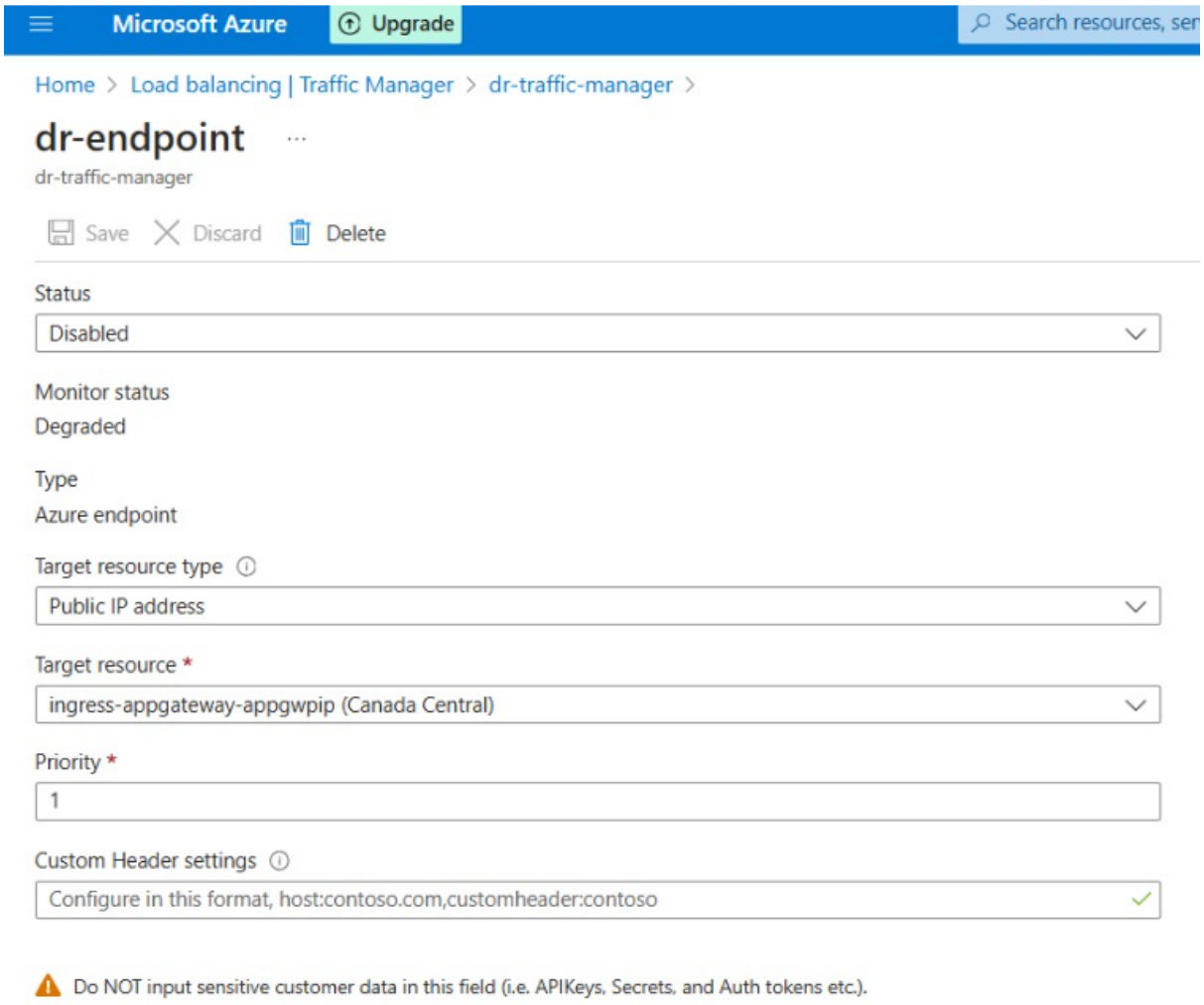
```
DR [WSL: UBUNTU]
! ngin-deploymnet.yaml
! nginx-ingress.yaml
! nginx-svc.yaml
```

```yaml
! nginx-svc.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: nginx-service
5    labels:
6      app: nginx
7  spec:
8    selector:
9      app: nginx  # This should match the label in the pod spec
10   ports:
11     - protocol: TCP
12       port: 80          # The port exposed by the service
13       targetPort: 80     # The port on the pod the service should route traffic to
14   type: ClusterIP      # For internal access within the cluster
```

Step10: Now In 2 AKS Cluster, go to 2nd nginx pod > /usr/share/nginx/html, edit index.html

Step11: Now, test aplication, in browser we can paster host name "dr-traffic-manager.trafficmanager.net", Primary DR nginx will open.

Step12: Now test failover by Disaling 1 endpoint in Traffic manager



Browse dr-traffic-manager.trafficmanager.net, in Browser it will give u conut of 2 cluster deployment