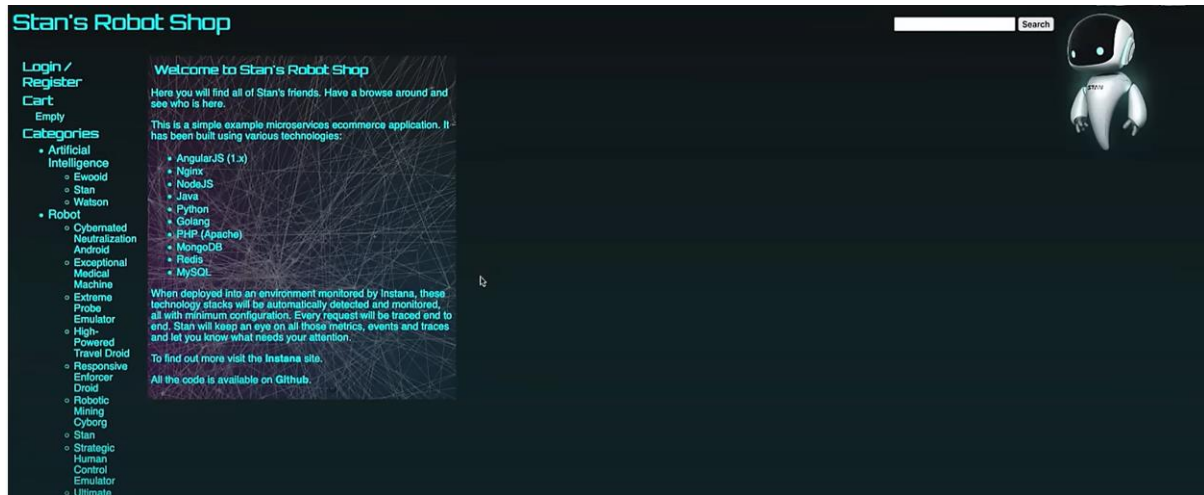


Day 17:

Deploying a Robot Shop App on AKS

Robot Shop Source Code: <https://github.com/instana/robot-shop>

Prefer: Abhishek Veeramalla Repo (With modifications to manifest files): <https://github.com/iam-veeramalla/three-tier-architecture-demo/tree/master>



Each microservice is written in different language.

Step – 1: Create a Resource Group

Basics Tags Review + create

Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. [Learn more](#)

Project details

Subscription *

Resource group *

Resource details

Region *

Step – 2: Create an AKS Cluster - prefer to use 1.27.7 K8s version

Basics Node pools Networking Integrations Monitoring Security Advanced Tags Review + create

Subscription *

Resource group *

[Create new](#)

Cluster details

Cluster preset configuration *

To quickly customize your Kubernetes cluster, choose one of the preset configurations above. You can modify these configurations at any time. [Compare presets](#)

Kubernetes cluster name *

Region *

Availability zones

AKS pricing tier

Kubernetes version *

Automatic upgrade

Automatic upgrade scheduler

Start on: Fri Nov 01 2024 00:00 +00:00 (Coordinated Universal Time)

[Edit schedule](#)

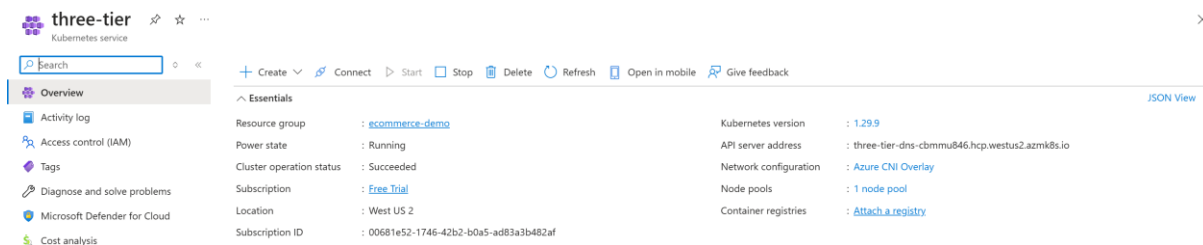
Node pools

In addition to the required primary node pool configured on the Basics tab, you can also add optional node pools to handle a variety of workloads. [Learn more](#)

[+ Add node pool](#) [Delete](#)

Node size	OS SKU	Node count	Availability zones	Max pods / node
Standard_DS2_v2 (...)	Ubuntu	2 - 5	None	110

Click on Review and Create.



Docker file for any MS to create containers:

```

1 FROM node:14
2
3 ENV INSTANA_AUTO_PROFILE true
4
5 EXPOSE 8080
6
7 WORKDIR /opt/server
8
9 COPY package.json /opt/server/
10
11 RUN npm install
12
13 COPY server.js /opt/server/
14
15 CMD ["node", "server.js"]

```

Starts with instructions like **From** and **Env**.

Expose: Port number on which the current Microservice application will run.

Workdir: This is where we want to perform all our next actions on. Its like a preferred directory. Next commands in the dockerfile will be run from this folder.

Copy: For python we will have requirements.txt to know the dependencies required to be installed for the project. Similarly for Nodes JS application we will have package.json also to some extent we can compare with pom.xml for java application. For Go lang application we can compare with go.mod like that. For DotNet .csproj file.

Run: It will install all the dependencies that are mentioned inside dependency file.

Copy: For this application the entire source code is inside the server.js

CMD: To run the application, we will run this command.

K8s Manifest files for these docker images:

We can use helm charts also for deploying applications into K8s cluster. Helm charts given in source code repo is not working when we try to deploying in AKS or EKS.

Helm charts have three main components: chart.yaml (Will have metadata (Chart version) of the chart), templates (Deployment and Service YAMl), values.yml (the values that are dynamic needs to be mentioned here and they will be passed to the manifest files in the template folder.)

In general if we write manifest files for 8 MS application then we will have 16 deployment and service YAM files. Additional manifest files for db and others. Overall we will have many files.

Consider we have 26 manifest files in total for one application. Then if we have 5 environments like Dev, SIT, UAT, Pre-Prod and Prod then $26 \times 5 = 130$ Files.

It is very difficult to write those many files. Because in SIT we may need our container to take max ram of 500MB but for prod we can allow it to occupy till 1 GB as well. So, we need to identify such parameters and pass them as dynamic parameters and then in values.yaml we can specify the values.



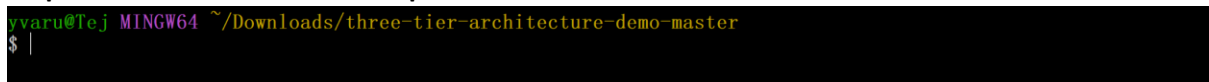
chart.yaml:



values.yaml



Step 3: Clone the Source Code Repo from Abhishek Veeramalla.



We had a redis statefulset yaml file along with deployment and service yaml files, where we mentioned we need a pv of 1 GB. Storage class is default. For azure if we mention storage class is default then it will allocate Azure disk of 1 gb storage.

Azure will have storage classes like Azure files (EFS in amazon); Azure disks (EBS in amazon).

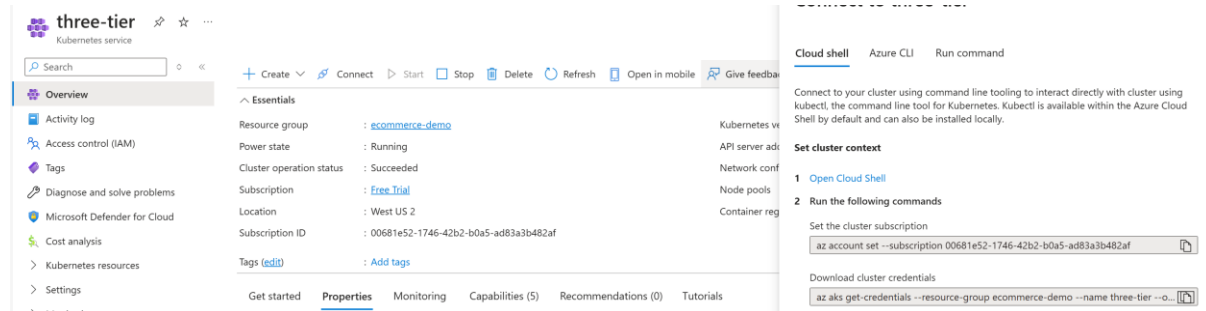
When to use Azure files and Azure disks?

If the PV is used by one container only then go for Azure Disks/EBS based on cloud.

If the PV is accessed by multiple containers from different nodes then use Azure files/EFS.

Step 4: Azure CLI, Kubectl and Helm to be installed

Step 5: Connect to AKS cluster using cluster credentials in local terminal



```
yvaru@Tej MINGW64 ~/Downloads/three-tier-architecture-demo-master
$ az aks get-credentials --resource-group ecommerce-demo --name three-tier --overwrite-existing
Merged "three-tier" as current context in C:\Users\yvaru\.kube\config

yvaru@Tej MINGW64 ~/Downloads/three-tier-architecture-demo-master
$

yvaru@Tej MINGW64 ~/Downloads/three-tier-architecture-demo-master
$ kubectl config current-context
three-tier

yvaru@Tej MINGW64 ~/Downloads/three-tier-architecture-demo-master
$ kubectl get pods
No resources found in default namespace.
```

Steps 6: Install Helm charts of AKS

CD to AKS/Helm folder:

```
yvaru@Tej MINGW64 ~/Downloads/three-tier-architecture-demo-master
$ ls
AKS/  EKS/  K8s/  OpenShift/  Swarm/  catalogue/  docker-compose-load.yaml  fluentd/  mongo/  payment/  ratings/  user/
DCOS/  GKE/  LICENSE  README.md  cart/  dispatch/  docker-compose.yaml  load-gen/  mysql/  pullbaseimages.sh*  shipping/  web/

yvaru@Tej MINGW64 ~/Downloads/three-tier-architecture-demo-master
$ cd AKS/

yvaru@Tej MINGW64 ~/Downloads/three-tier-architecture-demo-master/AKS
$ ls
helm/

yvaru@Tej MINGW64 ~/Downloads/three-tier-architecture-demo-master/AKS
$ cd helm/

yvaru@Tej MINGW64 ~/Downloads/three-tier-architecture-demo-master/AKS/helm
$ ls
Chart.yaml  README.md  ingress.yaml  templates/  values.yaml
```

Helm charts are already written. Create a namespace and Install the chart in that.

```
yvaru@Tej MINGW64 ~/Downloads/three-tier-architecture-demo-master/AKS/helm
$ kubectl create ns robot-shop
namespace/robot-shop created

yvaru@Tej MINGW64 ~/Downloads/three-tier-architecture-demo-master/AKS/helm
$ helm install robot-shop --namespace robot-shop .
NAME: robot-shop
LAST DEPLOYED: Fri Nov 1 00:17:08 2024
NAMESPACE: robot-shop
STATUS: deployed
REVISION: 1
TEST SUITE: None

yvaru@Tej MINGW64 ~/Downloads/three-tier-architecture-demo-master/AKS/helm
$ kubectl get pods -n robot-shop
NAME                                READY   STATUS    RESTARTS   AGE
cart-64c54bd6d7-jn65c              1/1     Running   0           118s
catalogue-577d55dcc8-xqfxh         1/1     Running   0           118s
dispatch-886d77bc8-7xjfg           1/1     Running   0           118s
mongodb-7474db6fdf-qnh7l           1/1     Running   0           118s
mysql-5fb849d788-8v22p             1/1     Running   0           118s
payment-6cc6b544db-hrl4g           1/1     Running   0           118s
rabbitmq-f688fb85f-hnqxm            1/1     Running   0           118s
ratings-56954c898b-k5krp            0/1     Running   0           118s
redis-0                             1/1     Running   0           118s
shipping-84c69c4c94-lw2wv           0/1     Running   0           118s
user-554bfcf4f8-x6zlp              1/1     Running   0           118s
web-dc75f5d68-rejjh                1/1     Running   0           118s
```

Lets check the PVC we gave in redis statfulset yaml is also created or not.

```
yvaru@Tej MINGW64 ~/Downloads/three-tier-architecture-demo-master/AKS/helm
$ kubectl get storageclass
NAME                                PROVISIONER             RECLAIMPOLICY   VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION   AGE
azurefile                           file.csi.azure.com      Delete          Immediate            true                   142m
azurefile-csi                       file.csi.azure.com      Delete          Immediate            true                   142m
azurefile-csi-premium               file.csi.azure.com      Delete          Immediate            true                   142m
azurefile-premium                   file.csi.azure.com      Delete          Immediate            true                   142m
default (default)                   disk.csi.azure.com      Delete          WaitForFirstConsumer true                   142m
managed                             disk.csi.azure.com      Delete          WaitForFirstConsumer true                   142m
managed-csi                         disk.csi.azure.com      Delete          WaitForFirstConsumer true                   142m
managed-csi-premium                 disk.csi.azure.com      Delete          WaitForFirstConsumer true                   142m
managed-premium                     disk.csi.azure.com      Delete          WaitForFirstConsumer true                   142m
```

```
yvaru@Tej MINGW64 ~/Downloads/three-tier-architecture-demo-master/AKS/helm
$ kubectl describe pod redis-0 -n robot-shop
Name:         redis-0
Namespace:    robot-shop
Events:
  Type        Reason              Age   From                        Message
  ----        -
  Normal      Scheduled           3m55s default-scheduler          Successfully assigned robot-shop/redis-0 to aks-agentpool-30053109-vmss000001
  Normal      SuccessfulAttachVolume 3m43s attachdetach-controller    AttachVolume.Attach succeeded for volume "pvc-ea3056e8-1bcd-475b-b38e-5d31adb5c46"
  Normal      Pulling             3m39s kubelet                    Pulling image "redis:4.0.6"
  Normal      Pulled              2m30s kubelet                    Successfully pulled image "redis:4.0.6" in 5.386s (1m8.794s including waiting)
  Normal      Created             2m30s kubelet                    Created container redis
  Normal      Started             2m30s kubelet                    Started container redis
```

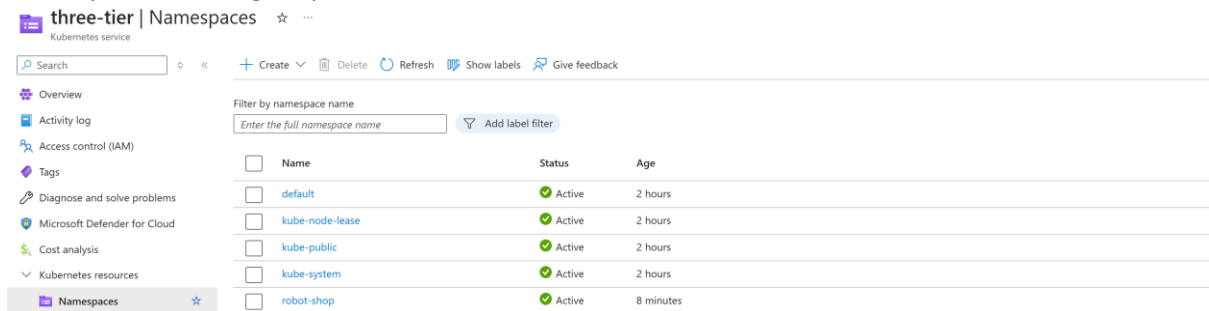
PV is attached to the Redis Pod.

```
yvaru@Tej MINGW64 ~/Downloads/three-tier-architecture-demo-master/AKS/helm
$ kubectl get pvc -n robot-shop
NAME          STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
data-redis-0  Bound   pvc-ea3056e8-1bcd-475b-b38e-5d31adb5c46  1Gi        RWO            default        <unset>               5m34s
```

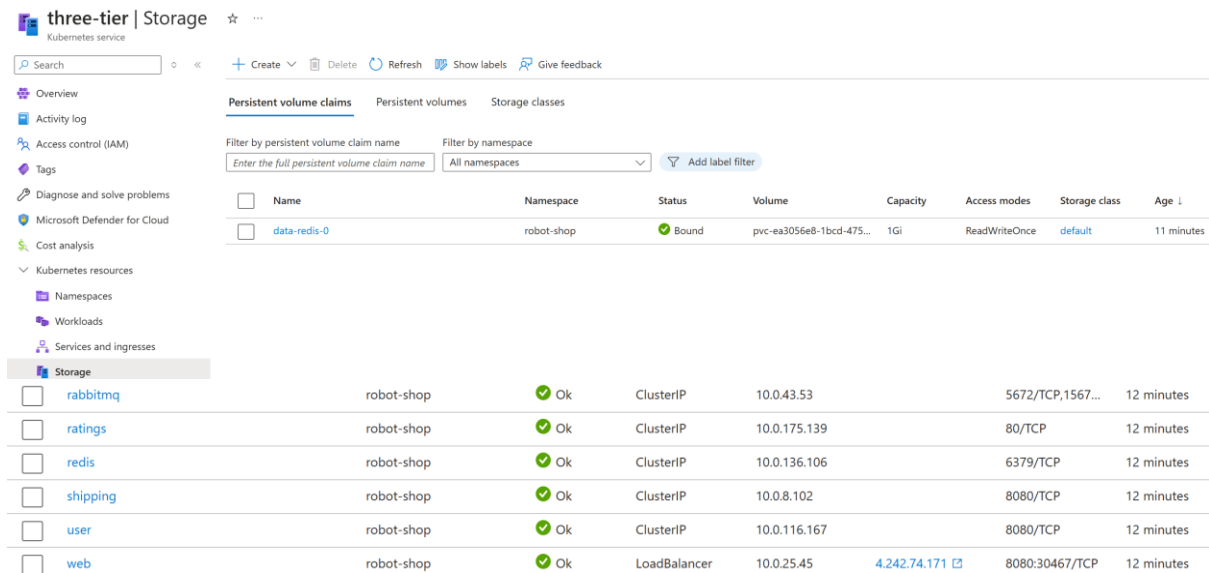
All pods are up and running now:

```
yvaru@Tej MINGW64 ~/Downloads/three-tier-architecture-demo-master/AKS/helm
$ kubectl get pods -n robot-shop -w
NAME                                READY   STATUS    RESTARTS   AGE
cart-64c54bd6d7-jn65c              1/1     Running   0           6m17s
catalogue-577d55dcc8-xqfxh         1/1     Running   0           6m17s
dispatch-886d77bc8-7xjfg           1/1     Running   0           6m17s
mongodb-7474db6fdf-qnh7l           1/1     Running   0           6m17s
mysql-5fb849d788-8v22p             1/1     Running   0           6m17s
payment-6cc6b544db-hr14g           1/1     Running   0           6m17s
rabbitmq-f688fb85f-hnqxm            1/1     Running   0           6m17s
ratings-56954c898b-k5krp            1/1     Running   0           6m17s
redis-0                             1/1     Running   0           6m17s
shipping-84c69c4c94-lw2wv           1/1     Running   0           6m17s
user-554bfcf4f8-x6zlp              1/1     Running   0           6m17s
web-dc75f5d68-rcjjh                1/1     Running   0           6m17s
```

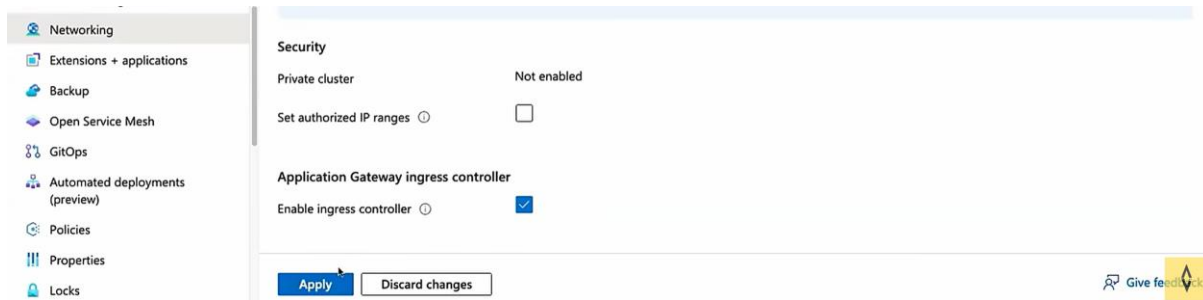
Namespace created got updated in Azure AKS UI:



You can view Pods, conatiners, YAMI files, services, Ingress, PV and PVC everything from here as well.

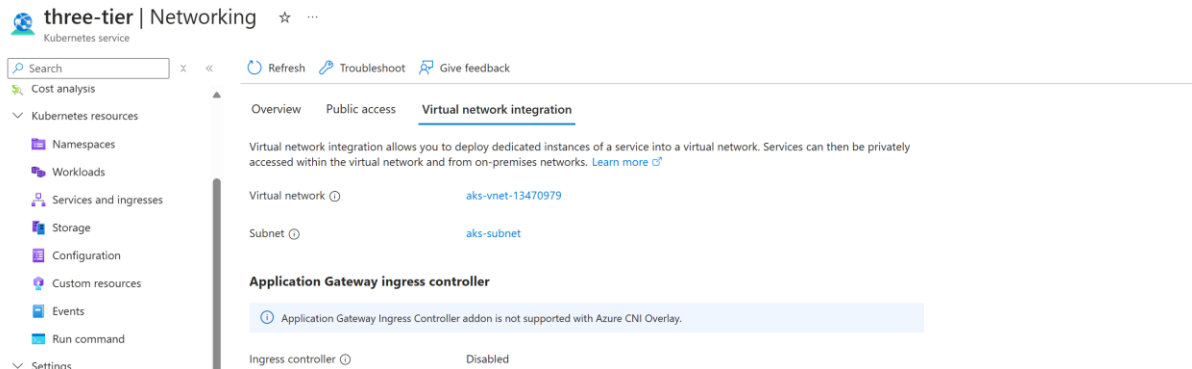


Web is created using Load Balancer, we can access it using that External IP. But we will not get additional capabilities that ingress controller can offer like path based or host based routing. So, lets can configure Ingress. To do that got AKS cluster and click on Networking:



We need to enable this 'Enable Ingress Controller'

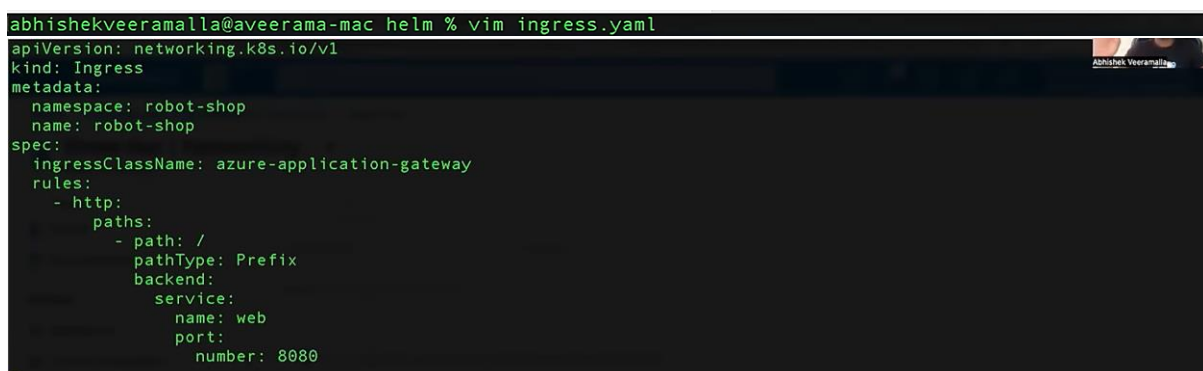
But we are getting like this:



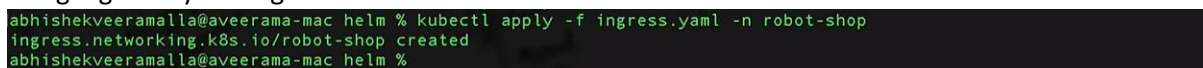
It will be a big process to enable it now since everything is created. Prefer to create K8s cluster with 1.27.7 version.



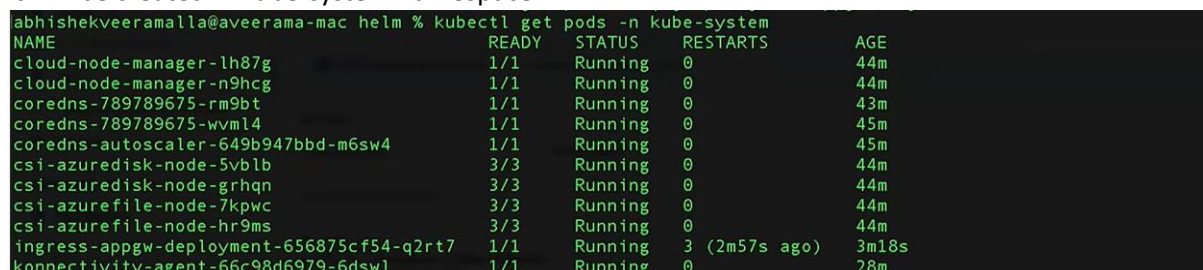
We need to have Ingress YAMI file:



If each namespace is using different Ingress controllers like Nginx and azure application gateway Ingress then it is required to mention ingress class name as it will help in identifying the resources using ingress by the ingress controller.



It will be created in kube-system namespace:




```

abhishekveeramalla@aveerama-mac helm % kubectl get ing -n robot-shop
NAME          CLASS                      HOSTS      ADDRESS      PORTS      AGE
robot-shop    azure-application-gateway   *          4.149.200.4  80         3m8s
abhishekveeramalla@aveerama-mac helm %

```

Address is not yet available right now: check logs

```

abhishekveeramalla@aveerama-mac helm % kubectl logs deploy/ingress-appgw-deployment -n kube-system

```

Some issue causing ingress controller to fail:

```

I0216 12:37:35.795698    1 reflector.go:255] Listing and watching *v1beta1.AzureApplicationGatewayRewrite from
mod/k8s.io/client-go/v0.20.0-beta.1/tools/cache/reflector.go:167
E0216 12:37:35.797609    1 reflector.go:138] pkg/mod/k8s.io/client-go/v0.20.0-beta.1/tools/cache/reflector.go:
Failed to watch *v1beta1.AzureApplicationGatewayRewrite: failed to list *v1beta1.AzureApplicationGatewayRewrite:
erver could not find the requested resource (get azureapplicationgatewayrewrites.appgw.ingress.azure.io)
I0216 12:37:38.656364    1 mutate_app_gateway.go:174] Applied generated Application Gateway configuration
I0216 12:37:38.656452    1 mutate_app_gateway.go:189] cache: Updated with latest applied config.
I0216 12:37:38.657209    1 mutate_app_gateway.go:193] END AppGateway deployment
I0216 12:37:38.657246    1 controller.go:152] Completed last event loop run in: 7.16317451s
abhishekveeramalla@aveerama-mac helm %

```

This is not an issue from our end. Delete the Ingress pod:

```

abhishekveeramalla@aveerama-mac helm % kubectl delete pod ingress-appgw-deployment-656875cf54-q2rt7 -n kube-system
pod "ingress-appgw-deployment-656875cf54-q2rt7" deleted
abhishekveeramalla@aveerama-mac helm %

```

New controller is created in sometime:

```

abhishekveeramalla@aveerama-mac helm % kubectl get pods -n kube-system
NAME                                READY   STATUS    RESTARTS   AGE
cloud-node-manager-lh87g            1/1     Running   0           49m
cloud-node-manager-n9hcg            1/1     Running   0           49m
coredns-789789675-rm9bt             1/1     Running   0           48m
coredns-789789675-wvml4             1/1     Running   0           49m
coredns-autoscaler-649b947bbd-m6sw4 1/1     Running   0           49m
csi-azuredisk-node-5vblb             3/3     Running   0           49m
csi-azuredisk-node-grhqn             3/3     Running   0           49m
csi-azurefile-node-7kpwc             3/3     Running   0           49m
csi-azurefile-node-hr9ms             3/3     Running   0           49m
ingress-appgw-deployment-656875cf54-qsjtd 1/1     Running   0           15s
connectivity-agent-66c98d6979-6dswl  1/1     Running   0           33m
abhishekveeramalla@aveerama-mac helm % kubectl get ing -n robot-shop
NAME          CLASS                      HOSTS      ADDRESS      PORTS      AGE
robot-shop    azure-application-gateway   *          4.149.200.4  80         7m6s
abhishekveeramalla@aveerama-mac helm %

```

