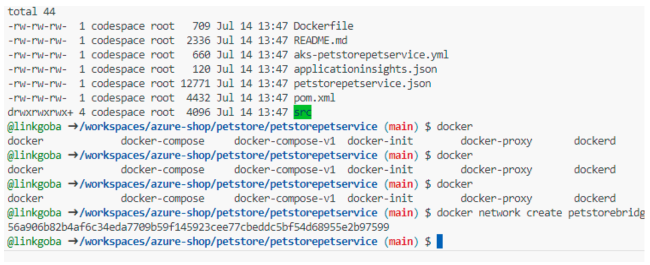
Azure Shop

Friday 23 June 2023 15:11

DOCKER

Setup local bridge network so our Docker containers (PetStoreApp, PetStorePetService, PetStoreProductService & PetStoreOrderService can all communicate)

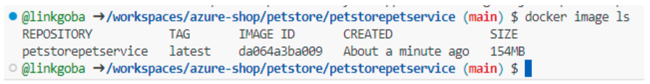


```
total 44
-rw-rw-rw-  1 codespace root   709 Jul 14 13:47 Dockerfile
-rw-rw-rw-  1 codespace root  2336 Jul 14 13:47 README.md
-rw-rw-rw-  1 codespace root   660 Jul 14 13:47 aks-petstorepetservice.yml
-rw-rw-rw-  1 codespace root   120 Jul 14 13:47 applicationinsights.json
-rw-rw-rw-  1 codespace root 12771 Jul 14 13:47 petstorepetservice.json
-rw-rw-rw-  1 codespace root  4432 Jul 14 13:47 pom.xml
drwxrwxrwx+ 4 codespace root  4096 Jul 14 13:47 src
@linkgoba →/workspaces/azure-shop/petstore/petstorepetservice (main) $ docker
docker             docker-compose     docker-compose-v1  docker-init        docker-proxy       dockerd
@linkgoba →/workspaces/azure-shop/petstore/petstorepetservice (main) $ docker
docker             docker-compose     docker-compose-v1  docker-init        docker-proxy       dockerd
@linkgoba →/workspaces/azure-shop/petstore/petstorepetservice (main) $ docker
docker             docker-compose     docker-compose-v1  docker-init        docker-proxy       dockerd
@linkgoba →/workspaces/azure-shop/petstore/petstorepetservice (main) $ docker network create petstorebridg
56a906b82b4af6c34eda7709b59f145923cee77cbeddc5bf54d68955e2b97599
@linkgoba →/workspaces/azure-shop/petstore/petstorepetservice (main) $
```

1.

Have Docker build our PetStorePetService Docker Image. (This is a multi stage Docker build, it will compile the PetStorePetService code and build our Docker Image containing this Spring Boot jar and all of the dependencies.)

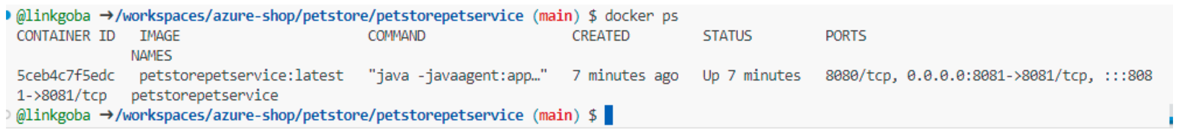
```
docker build -t petstorepetservice .
```



```
@linkgoba →/workspaces/azure-shop/petstore/petstorepetservice (main) $ docker image ls
REPOSITORY           TAG       IMAGE ID       CREATED              SIZE
petstorepetservice   latest    da064a3ba009   About a minute ago   154MB
@linkgoba →/workspaces/azure-shop/petstore/petstorepetservice (main) $
```

This will instruct Docker to start a running container with the following petstorepetservice:latest image, forwarding port 8081 to the Spring Boot App running on 8081. The PETSTOREPETSERVICE_SERVER_PORT is one of several environment variables

docker run --rm --net petstorebridge --name petstorepetservice -p 8081:8081 –e PETSTOREPETSERVICE_SERVER_PORT=8081 -d petstorepetservice:latest



```
@linkgoba →/workspaces/azure-shop/petstore/petstorepetservice (main) $ docker ps
CONTAINER ID   IMAGE                       COMMAND                  CREATED         STATUS         PORTS
               NAMES
5ceb4c7f5edc   petstorepetservice:latest   "java -javaagent:app…"   7 minutes ago   Up 7 minutes   8080/tcp, 0.0.0.0:8081->8081/tcp, :::808
1->8081/tcp    petstorepetservice
@linkgoba →/workspaces/azure-shop/petstore/petstorepetservice (main) $
```

2.

Have Docker build our PetStoreOrderService Docker Image. (This is a multi stage Docker build, it will compile the PetStoreOrderService code and build our Docker Image containing this Spring Boot jar and all of the dependencies.

```
docker build -t petstorepetservice .
```

This will instruct Docker to start a running container with the following petstoreproductservice:latest image, forwarding port 8082 to the Spring Boot App running on 8082. The PETSTOREPRODUCTSERVICE_SERVER_PORT is one of several environment variables that we will introduce over the course of these guides.

docker run --rm --net petstorebridge --name petstoreproductservice -p 8082:8082 -e PETSTOREPRODUCTSERVICE_SERVER_PORT=8082 -d petstoreproductservice:latest

3.

Have Docker build our PetStoreOrderService Docker Image. (This is a multi stage Docker build, it will compile the PetStoreOrderService code and build our Docker Image containing this Spring Boot jar and all of the dependencies.

docker build -t petstoreorderservice .

Instruct Docker to start a running container with the following petstoreorderservice:latest image, forwarding port 8083 to the Spring Boot App running on 8083. The PETSTOREORDERSERVICE_SERVER_PORT is one of several environment variables that we will introduce over the course of these guides.

```
docker run --rm --net petstorebridge --name petstoreorderservice -p 8083:8083 -e
PETSTOREORDERSERVICE_SERVER_PORT=8083 -d petstoreorderservice:latest
```

4.

Build store app docker image

cd to azure-cloud/petstore/petstoreapp

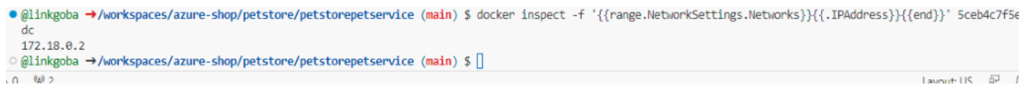
```
docker build -t petstoreapp .
```

```
docker image ls
```

Lets get the IP Addresses of the running Pet Store Services, we will need to pass them to the PetStoreApp to ensure communication can be made at runtime.

run the following command 3 times (substituting the <container_name_or_id> for petstorepetservice:latest & petstoreproductservice:latest & petstoreorderservice:latest that appeared in the latest docker ps command above) and capture the ip address that is displayed for each, for example 172.18.0.2 is the ip address for the petstorepetservice on my machine.

```
docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' <container_name_or_id>
```



```
@linkgoba ➜/workspaces/azure-shop/petstore/petstorepetservice (main) $ docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' 5ceb4c7f5e
dc
172.18.0.2
@linkgoba ➜/workspaces/azure-shop/petstore/petstorepetservice (main) $ 
```

```
@linkgoba ➜ /workspaces/azure-shop/petstore/petstoreapp (main) $ docker ps
CONTAINER ID   IMAGE                       COMMAND                CREATED         STATUS
PORTS                                    NAMES
c5f3292601cd   petstoreorderservice:latest    "java -javaagent:app…"   5 minutes ago    Up 5 minutes    8080/tcp,
0.0.0.0:8083->8083/tcp, :::8083->8083/tcp   petstoreorderservice
e041a7c6212d   petstoreproductservice:latest   "java -javaagent:app…"   11 minutes ago   Up 11 minutes
8080/tcp, 0.0.0.0:8082->8082/tcp, :::8082->8082/tcp   petstoreproductservice
adf1c9b1d63d   petstorepetservice:latest      "java -javaagent:app…"   32 minutes ago   Up 32 minutes   8080/tcp,
0.0.0.0:8081->8081/tcp, :::8081->8081/tcp   petstorepetservice
```

```
docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' c5f3292601cd
172.18.0.4
docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' e041a7c6212d
172.18.0.3
docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' adf1c9b1d63d
172.18.0.2
```

This will instruct Docker to start a running container with the following petstoreapp:latest image, forwarding port 8080 to the Spring Boot App running on 8080 (default Spring Boot Port). The PETSTOREAPP_SERVER_PORT is one of several environment variables that we will introduce over the course of these guides.

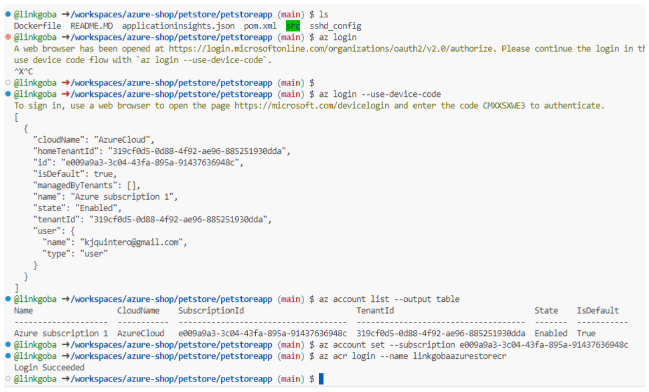
```
docker run --rm --net petstorebridge --name petstoreapp -p 8080:8080 -e PETSTOREAPP_SERVER_PORT=8080 -e
PETSTOREPETSERVICE_URL=http://172.18.0.2:8081 -e PETSTOREPRODUCTSERVICE_URL=http://172.18.0.3:8082 -e
PETSTOREORDERSERVICE_URL=http://172.18.0.4:8083 -d petstoreapp:latest
```

```
@linkgoba ➜ /workspaces/azure-shop/petstore/petstoreapp (main) $ docker ps
CONTAINER ID   IMAGE                       COMMAND                CREATED           STATUS
PORTS                                    NAMES
33f0324d23bf   petstoreapp:latest            "/bin/bash -c '/usr/…"   About a minute ago   Up 58 seconds   2222/tcp,
0.0.0.0:8080->8080/tcp, :::8080->8080/tcp   petstoreapp
c5f3292601cd   petstoreorderservice:latest    "java -javaagent:app…"   19 minutes ago      Up 19 minutes
8080/tcp, 0.0.0.0:8083->8083/tcp, :::8083->8083/tcp   petstoreorderservice
e041a7c6212d   petstoreproductservice:latest   "java -javaagent:app…"   25 minutes ago      Up 25 minutes
8080/tcp, 0.0.0.0:8082->8082/tcp, :::8082->8082/tcp   petstoreproductservice
adf1c9b1d63d   petstorepetservice:latest      "java -javaagent:app…"   46 minutes ago      Up 46 minutes   8080/tcp,
0.0.0.0:8081->8081/tcp, :::8081->8081/tcp   petstorepetservice
```

To kill all running containers with Docker run the following command:

```
docker kill $(docker ps -q)
```

PUSH DOCKER TO AZURE CONTAINER REGISTRY



```
● @linkgoba ➜/workspaces/azure-shop/petstore/petstoreapp (main) $ ls
  Dockerfile  README.MD  applicationinsights.json  pom.xml  src  sshd_config
⊗ @linkgoba ➜/workspaces/azure-shop/petstore/petstoreapp (main) $ az login
  A web browser has been opened at https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize. Please continue the login in th
  use device code flow with `az login --use-device-code`.
  ^X^C
○ @linkgoba ➜/workspaces/azure-shop/petstore/petstoreapp (main) $
● @linkgoba ➜/workspaces/azure-shop/petstore/petstoreapp (main) $ az login --use-device-code
  To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code CMXXSXWE3 to authenticate.
  [
    {
      "cloudName": "AzureCloud",
      "homeTenantId": "319cf0d5-0d88-4f92-ae96-885251930dda",
      "id": "e009a9a3-3c04-43fa-895a-91437636948c",
      "isDefault": true,
      "managedByTenants": [],
      "name": "Azure subscription 1",
      "state": "Enabled",
      "tenantId": "319cf0d5-0d88-4f92-ae96-885251930dda",
      "user": {
        "name": "kjquintero@gmail.com",
        "type": "user"
      }
    }
  ]
● @linkgoba ➜/workspaces/azure-shop/petstore/petstoreapp (main) $ az account list --output table
  Name                  CloudName    SubscriptionId                        TenantId                              State    IsDefault
  --------------------  -----------  ------------------------------------  ------------------------------------  -------  -----------
  Azure subscription 1  AzureCloud   e009a9a3-3c04-43fa-895a-91437636948c  319cf0d5-0d88-4f92-ae96-885251930dda  Enabled  True
● @linkgoba ➜/workspaces/azure-shop/petstore/petstoreapp (main) $ az account set --subscription e009a9a3-3c04-43fa-895a-91437636948c
● @linkgoba ➜/workspaces/azure-shop/petstore/petstoreapp (main) $ az acr login --name linkgobaazurestorecr
  Login Succeeded
○ @linkgoba ➜/workspaces/azure-shop/petstore/petstoreapp (main) $
```

az account set --subscription <your subscription>

az acr login --name linkgobaazurestorecr

az acr update -n linkgobaazurestorecr -g Azure_Store --admin-enabled true

tagging your local Docker image built in the previous guide so that we can push it to Azure Container Registry then push it

docker image tag petstoreapp:latest linkgobaazurestorecr.azurecr.io/petstoreapp:latest

docker push linkgobaazurestorecr.azurecr.io/petstoreapp:latest

Push the Pet Store Pet Service Docker Image to Azure Container Registry

cd to azure-cloud/petstore/petstorepetservice

```
docker image tag petstorepetservice:latest
linkgobaazurestorecr.azurecr.io/petstorepetservice:latest
docker push linkgobaazurestorecr.azurecr.io/petstorepetservice:latest
```

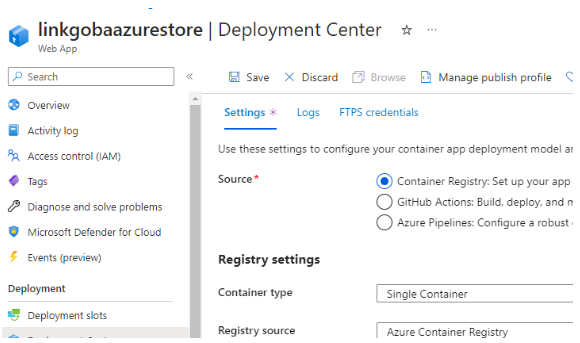
Push the Pet Store Product Service Docker Image to Azure Container Registry

```
docker image tag petstoreproductservice:latest
linkgobaazurestorecr.azurecr.io/petstoreproductservice:latest
docker push linkgobaazurestorecr.azurecr.io/petstoreproductservice:latest
```

Push the Pet Store Order Product Service Docker Image to Azure Container Registry

```
docker image tag petstoreorderservice:latest
linkgobaazurestorecr.azurecr.io/petstoreorderservice:latest
docker push linkgobaazurestorecr.azurecr.io/petstoreorderservice:latest
```

Link Web App Service with Azure Container Registry



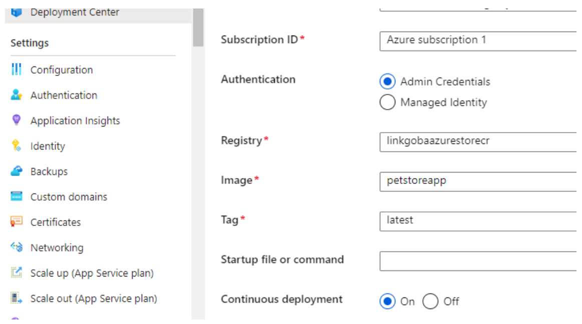
Use these settings to configure your container app deployment model ar

Source * [✓ Container Registry: Set up your app] GitHub Actions: Build, deploy, and n Azure Pipelines: Configure a robust

Registry settings

Container type: Single Container

Registry source: Azure Container Registry



Subscription ID* Azure subscription 1

Authentication [✓ Admin Credentials] Managed Identity

Registry* linkgobaazurestorecr

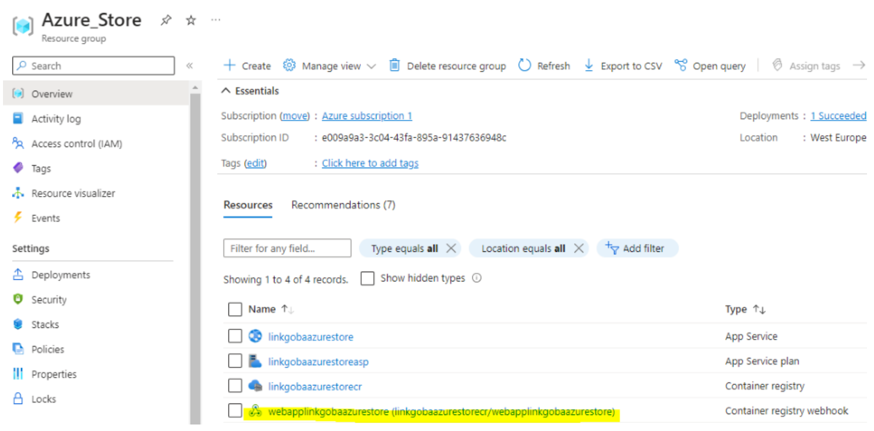
Image* petstoreapp

Tag* latest

Startup file or command

Continuous deployment [✓ On] Off

Linking will create a new resource of type Container Registry webhook



Subscription (move) : Azure subscription 1

Subscription ID : e009a9a3-3c04-43fa-895a-91437636948c

Tags (edit) : Click here to add tags

Deployments : 1 Succeeded

Location : West Europe

Showing 1 to 4 of 4 records.

Name	Type
linkgobaazurestore	App Service
linkgobaazurestoreasp	App Service plan
linkgobaazurestorecr	Container registry
webapplinkgobaazurestore (linkgobaazurestorecr/webapplinkgobaazurestore)	Container registry webhook

Well, an Azure Function is a different beast than an App Service. An Azure function is triggered by an external event or a timer. It then executes the code of the function. When hosted on a consumption plan this execution is allowed to run for 5 or 10 minutes max. When you need a longer execution time you need to run it on an App Service Plan.

An App Service can host any app you've created. Like a website (that runs continuously and doesn't need to be triggered before it starts doing something) or an api for example.

Create Azure Kubernetes

```
az login --use-device-code
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code CYW38ZKNT to authenticate.
[
 {
  "cloudName": "AzureCloud",
  "homeTenantId": "319cf0d5-0d88-4f92-ae96-885251930dda",
  "id": "e009a9a3-3c04-43fa-895a-91437636948c",
  "isDefault": true,
  "managedByTenants": [],
  "name": "Azure subscription 1",
  "state": "Enabled",
  "tenantId": "319cf0d5-0d88-4f92-ae96-885251930dda",
  "user": {
   "name": "kjquintero@gmail.com",
   "type": "user"
  }
 }
]
@linkgoba ➜ /workspaces/azure-shop (main) $ az account list --output table
Name                CloudName    SubscriptionId                        TenantId                              State    IsDefault
------------------  -----------  ------------------------------------  ------------------------------------  -------  -----------
Azure subscription 1  AzureCloud   e009a9a3-3c04-43fa-895a-91437636948c  319cf0d5-0d88-4f92-ae96-885251930dda
Enabled  True
@linkgoba ➜ /workspaces/azure-shop (main) $ az account set -s e009a9a3-3c04-43fa-895a-91437636948c
@linkgoba ➜ /workspaces/azure-shop (main) $ az configure --defaults acr=linkgobaazurestorecr.azurecr.io
@linkgoba ➜ /workspaces/azure-shop (main) $ az acr login -n linkgobaazurestorecr
Login Succeeded
@linkgoba ➜ /workspaces/azure-shop (main) $
```

Create a Kubernetes service:

```
az aks create --resource-group=Azure_Store --name=linkgobaazurepetstore-akscluster --attach-acr linkgobaazurestorecr --dns-name-prefix=linkgobaazurepetstoreserviceaks --generate-ssh-keys
```

Install `kubectl` using the Azure CLI

```
az aks install-cli
az aks get-credentials --resource-group=Azure_Store --name=linkgobaazurepetstore-akscluster
```

Install and configure Ingress controller

An Ingress controller abstracts away the complexity of Kubernetes application traffic routing and provides a bridge between Kubernetes services and external ones.

```
NAMESPACE=ingress-petstoreservices
```

helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx

helm repo update

helm install ingress-nginx ingress-nginx/ingress-nginx --create-namespace --namespace $NAMESPACE --set controller.service.annotations."service.beta.kubernetes.io/azure-load-balancer-health-probe-request-path"=/healthz f--> Do not excecute, run the one below

```
RESOURCE_GROUP=Azure_Store
ACR_URL=linkgobaazurestorecr.azurecr.io
REGISTRY_NAME=linkgobaazurestorecr
SOURCE_REGISTRY=k8s.gcr.io
CONTROLLER_IMAGE=ingress-nginx/controller
CONTROLLER_TAG=v1.0.4

PATCH_IMAGE=ingress-nginx/kube-webhook-certgen
PATCH_TAG=v1.1.1
DEFAULTBACKEND_IMAGE=defaultbackend-amd64
DEFAULTBACKEND_TAG=1.5
```

```
helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
helm repo update
```

Import the Ingress controller and required images into your ACR (Helm will use them on the install below)

```
az acr import --resource-group=$RESOURCE_GROUP --name $REGISTRY_NAME --source
$SOURCE_REGISTRY/$CONTROLLER_IMAGE:$CONTROLLER_TAG --image $CONTROLLER_IMAGE:$CONTROLLER_TAG
az acr import --resource-group=$RESOURCE_GROUP --name $REGISTRY_NAME --source
$SOURCE_REGISTRY/$PATCH_IMAGE:$PATCH_TAG --image $PATCH_IMAGE:$PATCH_TAG
az acr import --resource-group=$RESOURCE_GROUP --name $REGISTRY_NAME --source
$SOURCE_REGISTRY/$DEFAULTBACKEND_IMAGE:$DEFAULTBACKEND_TAG --image
$DEFAULTBACKEND_IMAGE:$DEFAULTBACKEND_TAG
```

If you head over to the Azure Portal > Azure Container Registry > Repositories you can view the recently imported images

Instruct Helm to install and configure the Ingress controller with the images

```
helm install ingress-nginx ingress-nginx/ingress-nginx --namespace $NAMESPACE --
create-namespace --set controller.replicaCount=2 --set controller.nodeSelecto
r."kubernetes\.io/os"=linux --set controller.image.registry=$ACR_URL --set
controller.image.image=$CONTROLLER_IMAGE --set controller.image.tag=$CONTROLLER_TAG --
set controller.image.digest="" --set con
```

```
troller.admissionWebhooks.patch.nodeSelector."kubernetes\.io/os"=linux --set
controller.admissionWebhooks.patch.image.registry=$ACR_URL --set
controller.admissionWebhooks.patch.image.image=$PATCH_IMAGE
--set controller.admissionWebhooks.patch.image.tag=$PATCH_TAG --set
controller.admissionWebhooks.patch.image.digest="" --set
defaultBackend.nodeSelector."kubernetes\.io/os"=linux --set defaultBackend.
image.registry=$ACR_URL --set defaultBackend.image.image=$DEFAULTBACKEND_IMAGE --set
defaultBackend.image.tag=$DEFAULTBACKEND_TAG --set defaultBackend.image.digest=""
```

Verify it is up and running

```
kubectl --namespace $NAMESPACE get services -o wide -w ingress-nginx-controller
```

```
NAME                        TYPE           CLUSTER-IP   EXTERNAL-IP
PORT(S)                       AGE   SELECTOR
ingress-nginx-controller   LoadBalancer   10.0.39.29   20.103.32.220
80:30325/TCP,443:31903/TCP   78s
app.kubernetes.io/component=controller,app.kubernetes.io/instance=ingress-
nginx,app.kubernetes.io/name=ingress-nginx
```

Deploy Pet Store Services to AKS

1. Add a user nodepool for the petstore services, the deployment yam's will use the nodeSelector `agentpool: petstorenp2` to deploy to this pool
```
   --resource-group azurepetstorerg \
    --cluster-name azurepetstore-akscluster \
    --name petstorenp2 \
    --node-count 3
```

2. Deploy petstorepetservice to AKS

   cd to azure-cloud/petstore/petstorepetservice

```
vi petstorepetservice-deployment.yml
```

   update the image path to that of your container registry, save and exit
```
image: azurepetstorecr.azurecr.io/petstorepetservice:latest
```

   run the deployment
```
@linkgoba ➜ /workspaces/azure-shop/petstore/petstorepetservice (main) $ kubectl apply -f aks-
petstorepetservice.yml --namespace $NAMESPACE
deployment.apps/aks-petstorepetservice created
service/aks-petstorepetservice created
@linkgoba ➜ /workspaces/azure-shop/petstore/petstorepetservice (main) $
```

verify the deployment

```
@linkgoba ➜ /workspaces/azure-shop/petstore/petstorepetservice (main) $ kubectl get
all --namespace $NAMESPACE
NAME                                            READY   STATUS             RESTARTS
AGE
pod/aks-petstorepetservice-649475ccb4-r2kqs     0/1     ImagePullBackOff   0
50s
pod/ingress-nginx-controller-846885cf97-4c4g8   1/1     Running            0
32m
pod/ingress-nginx-controller-846885cf97-6qdwr   1/1     Running            0
32m

NAME                                           TYPE           CLUSTER-IP     EXTERNAL-
IP     PORT(S)                      AGE
service/aks-petstorepetservice                 ClusterIP      10.0.192.219
<none>          80/TCP                      50s
service/ingress-nginx-controller               LoadBalancer   10.0.39.29
20.103.32.220   80:30325/TCP,443:31903/TCP   32m
service/ingress-nginx-controller-admission     ClusterIP      10.0.15.116
<none>          443/TCP                     32m
```

```
NAME                                       READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/aks-petstorepetservice     0/1     1            0           50s
deployment.apps/ingress-nginx-controller   2/2     2            2           32m

NAME                                                  DESIRED   CURRENT   READY   AGE
replicaset.apps/aks-petstorepetservice-649475ccb4     1         1         0       50s
replicaset.apps/ingress-nginx-controller-846885cf97   2         2         2       32m
@linkgoba ➜ /workspaces/azure-shop/petstore/petstorepetservice (main) $
```

Deploy petstoreproductservice:

```
cd to azure-cloud/petstore/petstoreproductservice
vi petstoreproductservice-deployment.yml
update the image path to that of your container registry, save and exit
image: azurepetstorecr.azurecr.io/petstoreproductservice:latest
```

```
@linkgoba ➜ /workspaces/azure-shop/petstore/petstoreproductservice (main) $ kubectl
apply -f aks-petstoreproductservice.yml --namespace $NAMESPACE
deployment.apps/aks-petstoreproductservice created
service/aks-petstoreproductservice created
@linkgoba ➜ /workspaces/azure-shop/petstore/petstoreproductservice (main) $
```

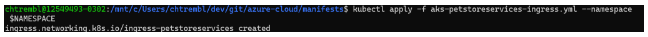
Deploy petstoreorderservice:

```
$ kubectl apply -f aks-petstoreorderservice.yml --namespace $NAMESPACE
deployment.apps/aks-petstoreorderservice created
service/aks-petstoreorderservice created
@linkgoba ➜ /workspaces/azure-shop/petstore/petstoreorderservice (main) $
```

Deploy Ingress controller configuration

```
cd to azure-cloud/manifests
kubectl apply -f aks-petstoreservices-ingress.yml --namespace $NAMESPACE
```

You should see something similar to the below image:



```
chtrembl@12549493-0302:/mnt/c/Users/chtrembl/dev/git/azure-cloud/manifests$ kubectl apply -f aks-petstoreservices-ingress.yml --namespace
 $NAMESPACE
ingress.networking.k8s.io/ingress-petstoreservices created
```

```
kubectl --namespace $NAMESPACE get services -o wide -w ingress-nginx-controller
NAME                       TYPE           CLUSTER-IP    EXTERNAL-IP
PORT(S)                      AGE   SELECTOR
ingress-nginx-controller   LoadBalancer   10.0.39.29    20.103.32.220
80:30325/TCP,443:31903/TCP   38m
app.kubernetes.io/component=controller,app.kubernetes.io/instance=ingress-
nginx,app.kubernetes.io/name=ingress-nginx
^C@linkgoba ➜ /workspaces/azure-shop/manifests (main) $ curl
http://20.103.32.220/petstorepetservice/v2/pet/info | json_pp
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:--  0:00:11 --:--:--     0
  0     0    0     0    0     0      0      0 --:--:--  0:00:12 --:--:--     0
  0     0    0     0    0     0      0      0 --:--:--  0:01:24 --:--:--     0^C
```

```
Error:
NAME                                           READY   STATUS
RESTARTS   AGE
pod/aks-petstoreorderservice-6ff6cdd56d-qgcj7     0/1     ImagePullBackOff
0          7m21s
```

```
pod/aks-petstorepetservice-649475ccb4-r2kqs       0/1     ImagePullBackOff
0          11m
pod/aks-petstoreproductservice-79f5f9bd88-tk4gb   0/1     ImagePullBackOff
0          8m38s
pod/ingress-nginx-controller-846885cf97-4c4g8     1/1     Running
0
```