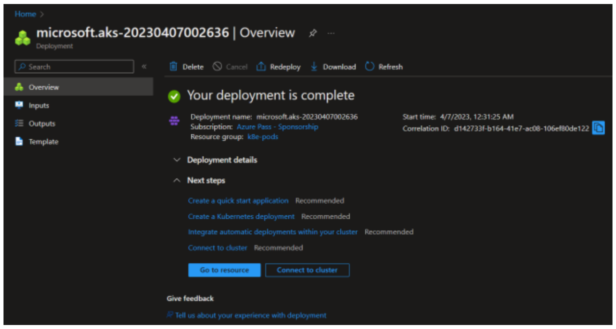
Kubernetes Pods

Practice1: Simple pods operations

1. Login to Azure and connect to your AKS cluster.
   - We first need to create AKS cluster. We do that by going to Kubernetes services in Azure a go to + Create and Create Kubernetes Cluster. And we input the requested fields. After that we click on Review + create. We wait for successful validation and then we click create. If the deployment of the AKS cluster is successful, we get the following:



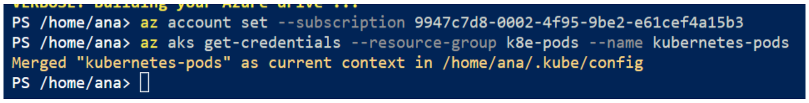
Your deployment is complete

Deployment name: microsoft.aks-20230407002636
Subscription: Azure Pass - Sponsorship
Resource group: k8e-pods
Start time: 4/7/2023, 12:31:25 AM
Correlation ID: d142733f-b164-41e7-ac08-106ef80de122

Create a quick start application Recommended
Create a Kubernetes deployment Recommended
Integrate automatic deployments within your cluster Recommended
Connect to cluster Recommended

Tell us about your experience with deployment

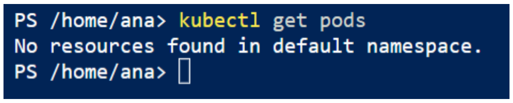
   After the cluster is created, we need to connect to that cluster by using Azure CLI. We use the following commands:



```
VERBOSE: Building your Azure drive ...
PS /home/ana> az account set --subscription 9947c7d8-0002-4f95-9be2-e61cef4a15b3
PS /home/ana> az aks get-credentials --resource-group k8e-pods --name kubernetes-pods
Merged "kubernetes-pods" as current context in /home/ana/.kube/config
PS /home/ana> 
```

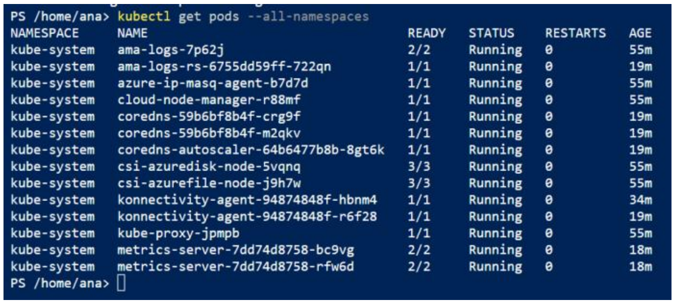
   Once we have run the commands above to connect to the cluster, we can run any kubectl commands.

2. Check how many pods run under the default namespace. Run kubectl get pods.



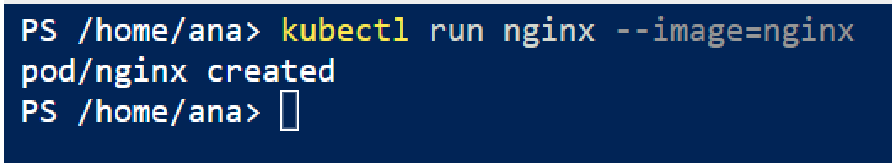
```
PS /home/ana> kubectl get pods
No resources found in default namespace.
PS /home/ana> 
```

   After we run the command, we can see that there are no pods.
3. You should not see any pod under the default namespace. Now check all namespaces. Run kubectl get pods –all-namespace.
   - When we run the command, we can see that there are several pods automatically created.



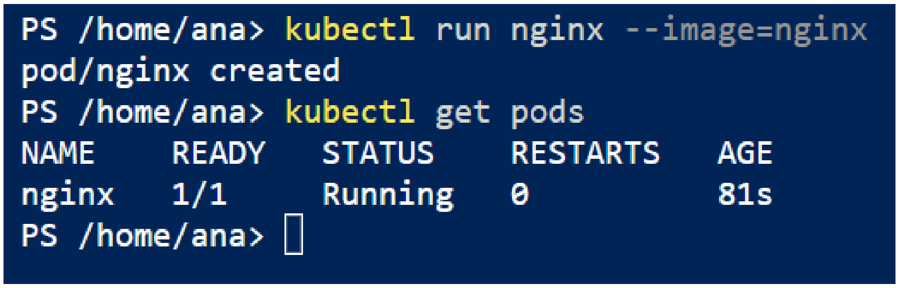
```
PS /home/ana> kubectl get pods --all-namespaces
NAMESPACE     NAME                                  READY   STATUS    RESTARTS   AGE
kube-system   ama-logs-7p62j                        2/2     Running   0          55m
kube-system   ama-logs-rs-6755dd59ff-722qn          1/1     Running   0          19m
kube-system   azure-ip-masq-agent-b7d7d             1/1     Running   0          55m
kube-system   cloud-node-manager-r88mf              1/1     Running   0          55m
kube-system   coredns-59b6bf8b4f-crg9f              1/1     Running   0          19m
kube-system   coredns-59b6bf8b4f-m2qkv              1/1     Running   0          19m
kube-system   coredns-autoscaler-64b6477b8b-8gt6k   1/1     Running   0          19m
kube-system   csi-azuredisk-node-5vqnq              3/3     Running   0          55m
kube-system   csi-azurefile-node-j9h7w              3/3     Running   0          55m
kube-system   konnectivity-agent-94874848f-hbnm4    1/1     Running   0          34m
kube-system   konnectivity-agent-94874848f-r6f28    1/1     Running   0          19m
kube-system   kube-proxy-jpmpb                      1/1     Running   0          55m
kube-system   metrics-server-7dd74d8758-bc9vg       2/2     Running   0          18m
kube-system   metrics-server-7dd74d8758-rfw6d       2/2     Running   0          18m
PS /home/ana>
```

4. How many pods do you see? Who deployed these pods? Why are they deployed?
   - We see 14 pods here, but they can vary. The number will depend on the specific Kubernetes cluster and what applications and components are running in it. The kube-system namespace in Kubernetes is reserved for system-level components that are critical to the operation of the Kubernetes cluster. The reason for deploying kube-system pods is to provide essential cluster functionality, including Control Plane Components, Networking, Logging and Monitoring, Security.
5. Now deploy your first pod using the imperative approach. Run kubectl run nginx --image=nginx.



```
PS /home/ana> kubectl run nginx --image=nginx
pod/nginx created
PS /home/ana>
```

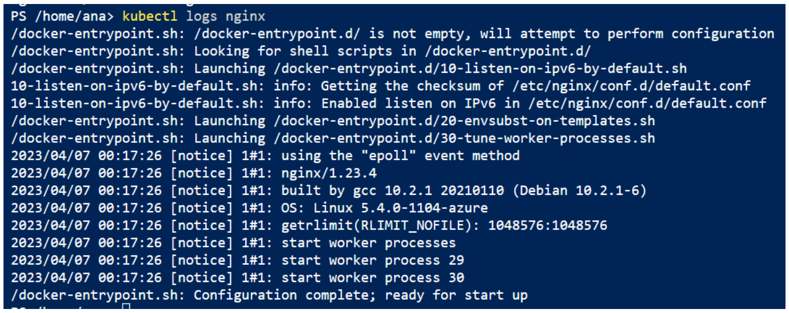
6. Validate if the pods have been created. What is the status of your pod?



```
PS /home/ana> kubectl run nginx --image=nginx
pod/nginx created
PS /home/ana> kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          81s
PS /home/ana>
```

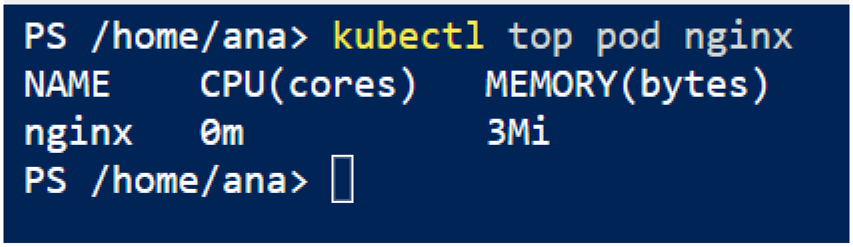
We validate them with the command **kubectl run nginx –image=nginx.** And we can see that the status of the pod created is **running.**

7. Check the logs coming out of your pod. Run kubectl logs nginx.



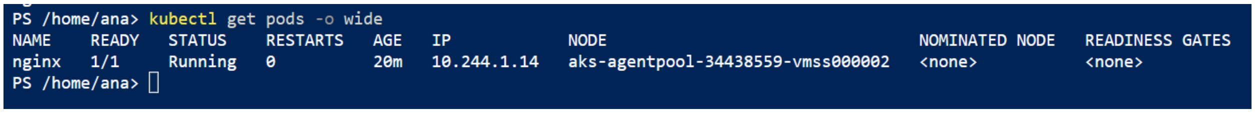
```
PS /home/ana> kubectl logs nginx
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
2023/04/07 00:17:26 [notice] 1#1: using the "epoll" event method
2023/04/07 00:17:26 [notice] 1#1: nginx/1.23.4
2023/04/07 00:17:26 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/04/07 00:17:26 [notice] 1#1: OS: Linux 5.4.0-1104-azure
2023/04/07 00:17:26 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/04/07 00:17:26 [notice] 1#1: start worker processes
2023/04/07 00:17:26 [notice] 1#1: start worker process 29
2023/04/07 00:17:26 [notice] 1#1: start worker process 30
/docker-entrypoint.sh: Configuration complete; ready for start up
```

8. Run following command to check current resource consumption of your pod: kubectl top pod nginx.



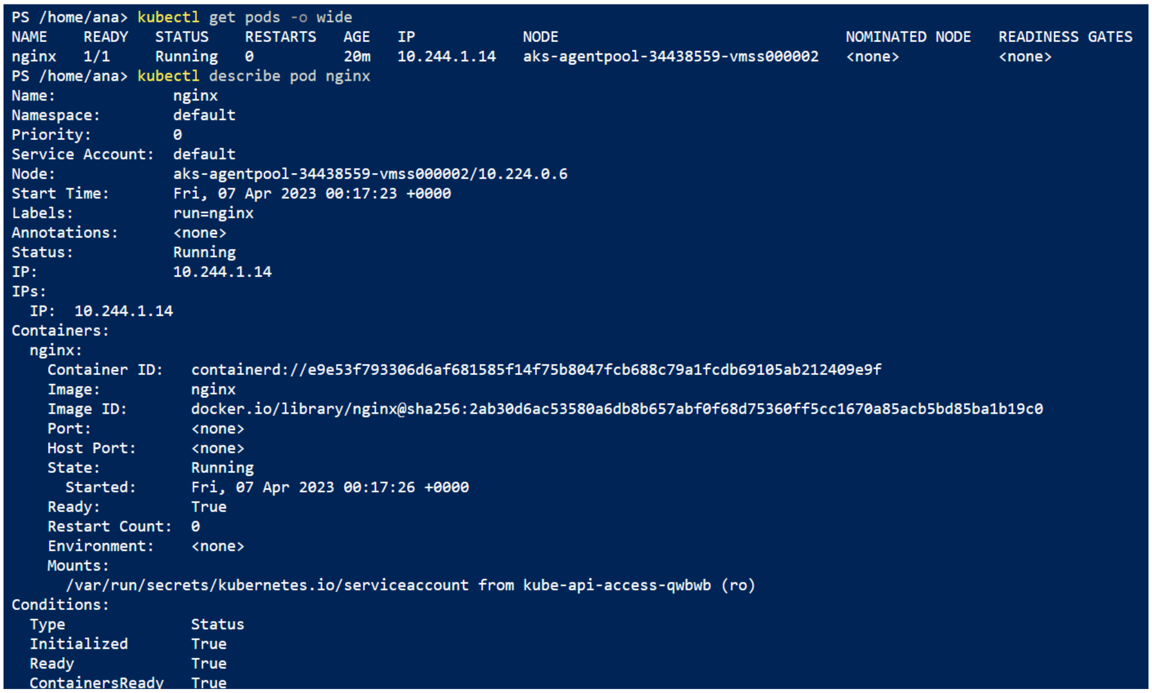
```
PS /home/ana> kubectl top pod nginx
NAME    CPU(cores)   MEMORY(bytes)
nginx   0m           3Mi
PS /home/ana> 
```

9. Check on which Node your pods have been scheduled. Run kubectl get pods –o wide.



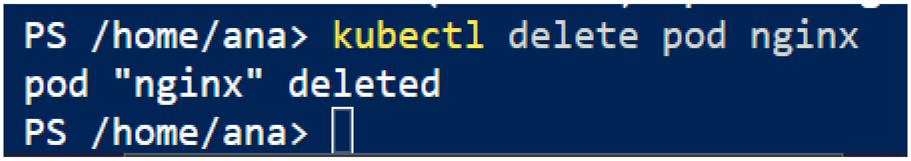
```
PS /home/ana> kubectl get pods -o wide
NAME    READY   STATUS    RESTARTS   AGE   IP            NODE                                NOMINATED NODE   READINESS GATES
nginx   1/1     Running   0          20m   10.244.1.14   aks-agentpool-34438559-vmss000002   <none>           <none>
PS /home/ana> 
```

10. Try to find the same information but this time running kubectl describe pod nginx.



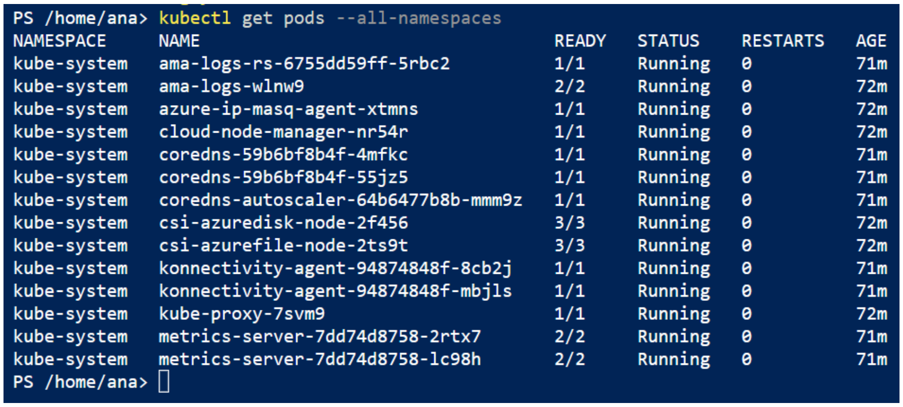
```
PS /home/ana> kubectl get pods -o wide
NAME    READY   STATUS    RESTARTS   AGE   IP            NODE                                NOMINATED NODE   READINESS GATES
nginx   1/1     Running   0          20m   10.244.1.14   aks-agentpool-34438559-vmss000002   <none>           <none>
PS /home/ana> kubectl describe pod nginx
Name:             nginx
Namespace:        default
Priority:         0
Service Account:  default
Node:             aks-agentpool-34438559-vmss000002/10.224.0.6
Start Time:       Fri, 07 Apr 2023 00:17:23 +0000
Labels:           run=nginx
Annotations:      <none>
Status:           Running
IP:               10.244.1.14
IPs:
  IP:  10.244.1.14
Containers:
  nginx:
    Container ID:   containerd://e9e53f793306d6af681585f14f75b8047fcb688c79a1fcdb69105ab212409e9f
    Image:          nginx
    Image ID:       docker.io/library/nginx@sha256:2ab30d6ac53580a6db8b657abf0f68d75360ff5cc1670a85acb5bd85ba1b19c0
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Fri, 07 Apr 2023 00:17:26 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-qwbwb (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
```

11. Delete your pod using kubectl delete pod nginx.



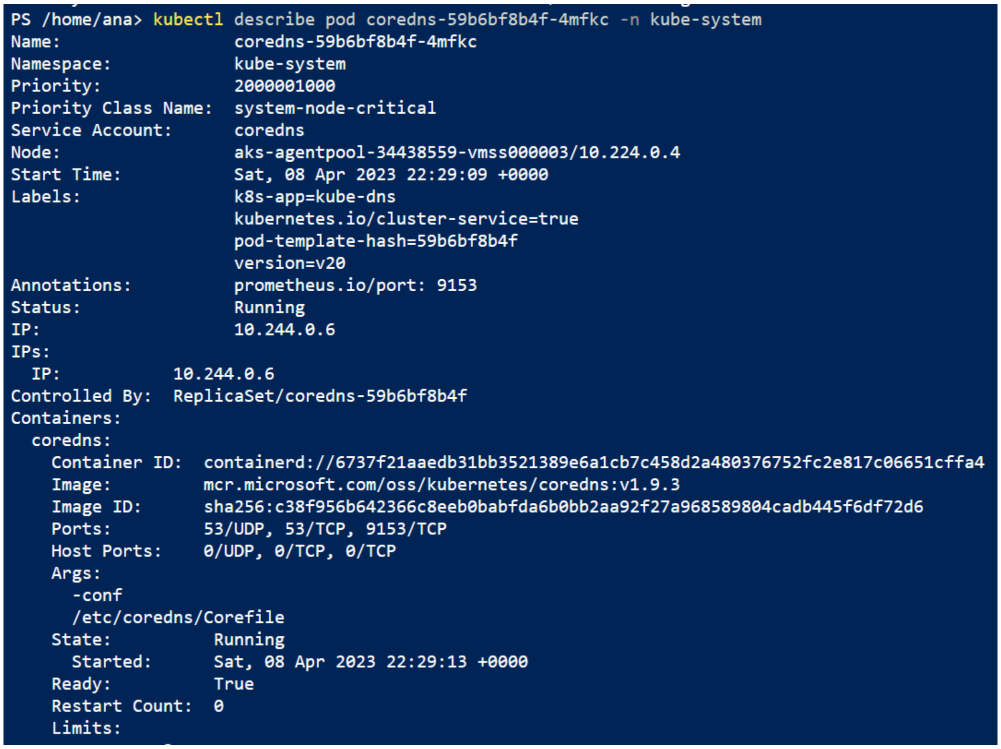
```
PS /home/ana> kubectl delete pod nginx
pod "nginx" deleted
PS /home/ana> 
```

12. Let's find the image used on one of the coredns pods under the kube-system namespace.
13. Once again list all pods under all namespaces.

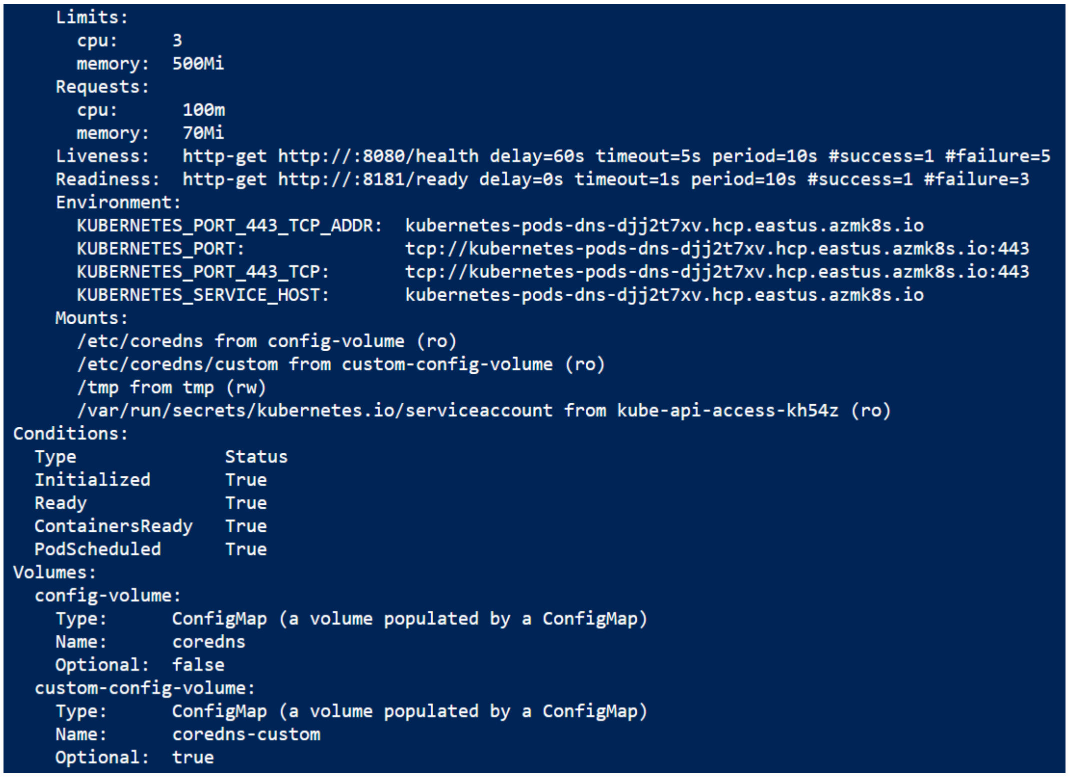


```
PS /home/ana> kubectl get pods --all-namespaces
NAMESPACE     NAME                                   READY   STATUS    RESTARTS   AGE
kube-system   ama-logs-rs-6755dd59ff-5rbc2           1/1     Running   0          71m
kube-system   ama-logs-wlnw9                         2/2     Running   0          72m
kube-system   azure-ip-masq-agent-xtmns              1/1     Running   0          72m
kube-system   cloud-node-manager-nr54r               1/1     Running   0          72m
kube-system   coredns-59b6bf8b4f-4mfkc               1/1     Running   0          71m
kube-system   coredns-59b6bf8b4f-55jz5               1/1     Running   0          71m
kube-system   coredns-autoscaler-64b6477b8b-mmm9z    1/1     Running   0          71m
kube-system   csi-azuredisk-node-2f456               3/3     Running   0          72m
kube-system   csi-azurefile-node-2ts9t               3/3     Running   0          72m
kube-system   konnectivity-agent-94874848f-8cb2j     1/1     Running   0          71m
kube-system   konnectivity-agent-94874848f-mbjls     1/1     Running   0          71m
kube-system   kube-proxy-7svm9                       1/1     Running   0          72m
kube-system   metrics-server-7dd74d8758-2rtx7        2/2     Running   0          71m
kube-system   metrics-server-7dd74d8758-lc98h        2/2     Running   0          71m
PS /home/ana> 
```

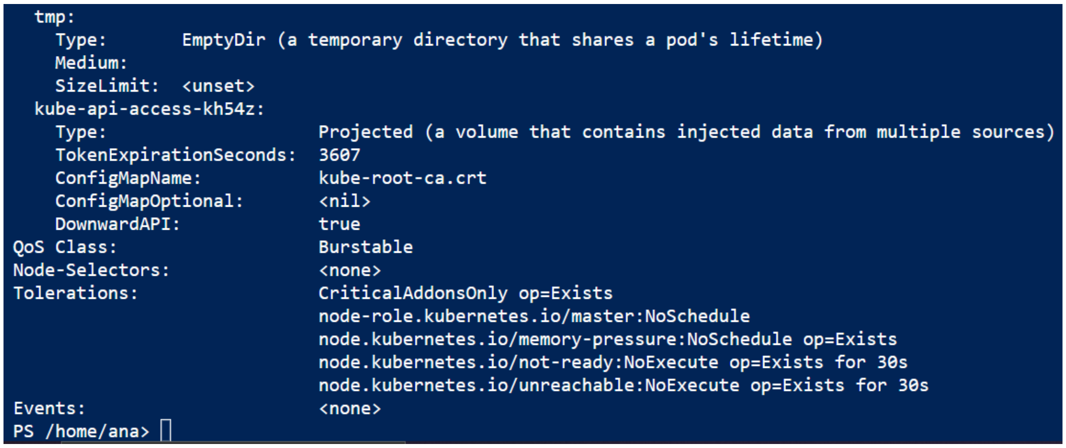
14. Note one of the coredns pods. Now run kubectl describe pod <coredns-name> -n kube-system. Replace the <coredns-name> place holder with noted name.
    We will use this coredns pod: **coredns-59b6bf8b4f-4mfkc**



```
PS /home/ana> kubectl describe pod coredns-59b6bf8b4f-4mfkc -n kube-system
Name:                 coredns-59b6bf8b4f-4mfkc
Namespace:            kube-system
Priority:             2000001000
Priority Class Name:  system-node-critical
Service Account:      coredns
Node:                 aks-agentpool-34438559-vmss000003/10.224.0.4
Start Time:           Sat, 08 Apr 2023 22:29:09 +0000
Labels:               k8s-app=kube-dns
                      kubernetes.io/cluster-service=true
                      pod-template-hash=59b6bf8b4f
                      version=v20
Annotations:          prometheus.io/port: 9153
Status:               Running
IP:                   10.244.0.6
IPs:
  IP:           10.244.0.6
Controlled By:  ReplicaSet/coredns-59b6bf8b4f
Containers:
  coredns:
    Container ID:  containerd://6737f21aaedb31bb3521389e6a1cb7c458d2a480376752fc2e817c06651cffa4
    Image:         mcr.microsoft.com/oss/kubernetes/coredns:v1.9.3
    Image ID:      sha256:c38f956b642366c8eeb0babfda6b0bb2aa92f27a968589804cadb445f6df72d6
    Ports:         53/UDP, 53/TCP, 9153/TCP
    Host Ports:    0/UDP, 0/TCP, 0/TCP
    Args:
      -conf
      /etc/coredns/Corefile
    State:          Running
      Started:      Sat, 08 Apr 2023 22:29:13 +0000
    Ready:          True
    Restart Count:  0
    Limits:
```

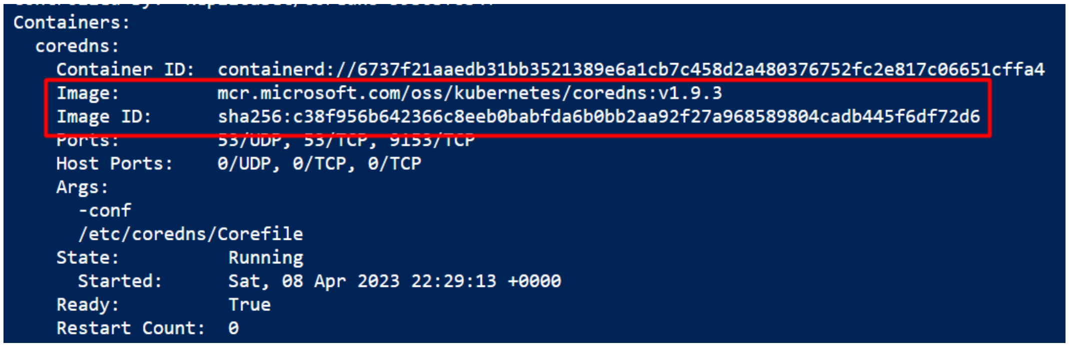


```
    Limits:
      cpu:     3
      memory:  500Mi
    Requests:
      cpu:      100m
      memory:   70Mi
    Liveness:   http-get http://:8080/health delay=60s timeout=5s period=10s #success=1 #failure=5
    Readiness:  http-get http://:8181/ready delay=0s timeout=1s period=10s #success=1 #failure=3
    Environment:
      KUBERNETES_PORT_443_TCP_ADDR:  kubernetes-pods-dns-djj2t7xv.hcp.eastus.azmk8s.io
      KUBERNETES_PORT:               tcp://kubernetes-pods-dns-djj2t7xv.hcp.eastus.azmk8s.io:443
      KUBERNETES_PORT_443_TCP:       tcp://kubernetes-pods-dns-djj2t7xv.hcp.eastus.azmk8s.io:443
      KUBERNETES_SERVICE_HOST:       kubernetes-pods-dns-djj2t7xv.hcp.eastus.azmk8s.io
    Mounts:
      /etc/coredns from config-volume (ro)
      /etc/coredns/custom from custom-config-volume (ro)
      /tmp from tmp (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-kh54z (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  config-volume:
    Type:      ConfigMap (a volume populated by a ConfigMap)
    Name:      coredns
    Optional:  false
  custom-config-volume:
    Type:      ConfigMap (a volume populated by a ConfigMap)
    Name:      coredns-custom
    Optional:  true
```



```
  tmp:
    Type:       EmptyDir (a temporary directory that shares a pod's lifetime)
    Medium:
    SizeLimit:  <unset>
  kube-api-access-kh54z:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   Burstable
Node-Selectors:              <none>
Tolerations:                 CriticalAddonsOnly op=Exists
                             node-role.kubernetes.io/master:NoSchedule
                             node.kubernetes.io/memory-pressure:NoSchedule op=Exists
                             node.kubernetes.io/not-ready:NoExecute op=Exists for 30s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 30s
Events:                      <none>
PS /home/ana>
```

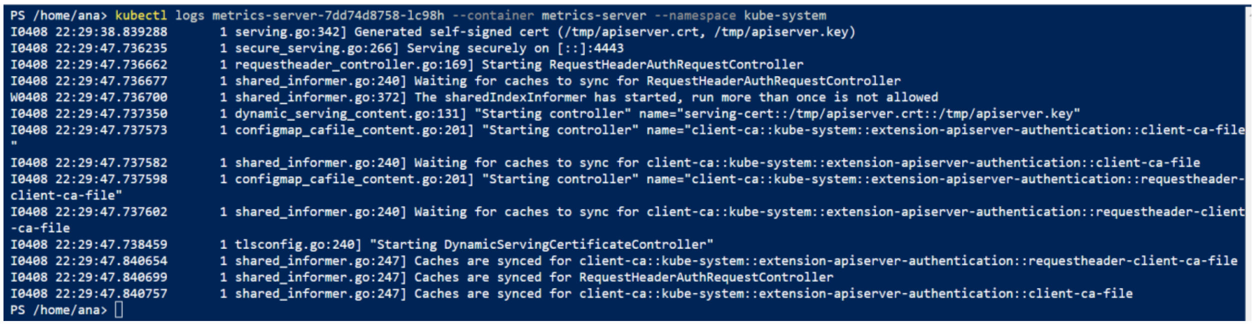
15. Inspect the output and locate the image information.



```
Containers:
  coredns:
    Container ID:  containerd://6737f21aaedb31bb3521389e6a1cb7c458d2a480376752fc2e817c06651cffa4
    Image:         mcr.microsoft.com/oss/kubernetes/coredns:v1.9.3
    Image ID:      sha256:c38f956b642366c8eeb0babfda6b0bb2aa92f27a968589804cadb445f6df72d6
    Ports:         53/UDP, 53/TCP, 9153/TCP
    Host Ports:    0/UDP, 0/TCP, 0/TCP
    Args:
      -conf
      /etc/coredns/Corefile
    State:          Running
      Started:      Sat, 08 Apr 2023 22:29:13 +0000
    Ready:          True
    Restart Count:  0
```

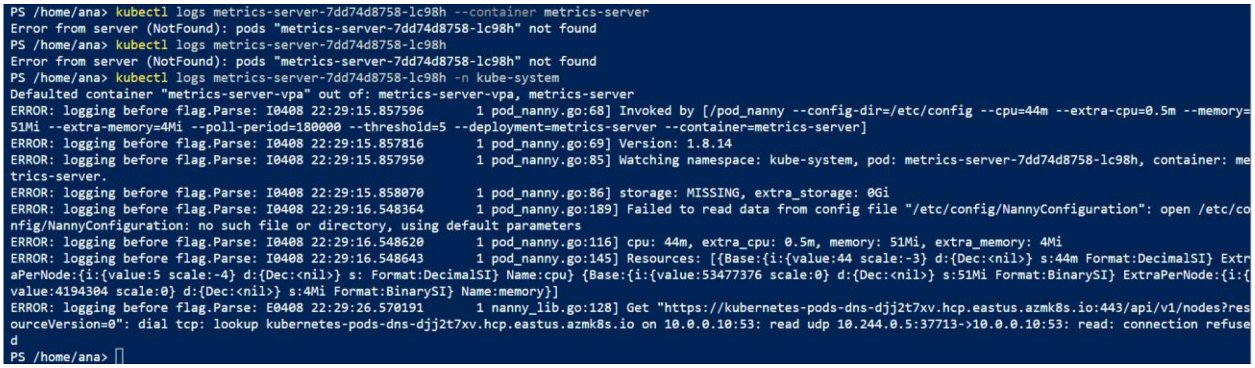
16. Now let us check the logs of the metrics-server pod. Run the same command as in step 7 but don’t forget to add the namespace in which this pod is created.

    We use the command **kubectl logs metrics-server-7dd74d8758-lc98h --container metrics-server --namespace kube-system** where we need to specify the container name as well as the namespace, otherwise if we don’t specify the namespace and the container we would receive an error.



```
PS /home/ana> kubectl logs metrics-server-7dd74d8758-lc98h --container metrics-server --namespace kube-system
I0408 22:29:38.839288       1 serving.go:342] Generated self-signed cert (/tmp/apiserver.crt, /tmp/apiserver.key)
I0408 22:29:47.736235       1 secure_serving.go:266] Serving securely on [::]:4443
I0408 22:29:47.736662       1 requestheader_controller.go:169] Starting RequestHeaderAuthRequestController
I0408 22:29:47.736677       1 shared_informer.go:240] Waiting for caches to sync for RequestHeaderAuthRequestController
W0408 22:29:47.736700       1 shared_informer.go:372] The sharedIndexInformer has started, run more than once is not allowed
I0408 22:29:47.737350       1 dynamic_serving_content.go:131] "Starting controller" name="serving-cert::/tmp/apiserver.crt::/tmp/apiserver.key"
I0408 22:29:47.737573       1 configmap_cafile_content.go:201] "Starting controller" name="client-ca::kube-system::extension-apiserver-authentication::client-ca-file
"
I0408 22:29:47.737582       1 shared_informer.go:240] Waiting for caches to sync for client-ca::kube-system::extension-apiserver-authentication::client-ca-file
I0408 22:29:47.737598       1 configmap_cafile_content.go:201] "Starting controller" name="client-ca::kube-system::extension-apiserver-authentication::requestheader-
client-ca-file"
I0408 22:29:47.737602       1 shared_informer.go:240] Waiting for caches to sync for client-ca::kube-system::extension-apiserver-authentication::requestheader-client
-ca-file
I0408 22:29:47.738459       1 tlsconfig.go:240] "Starting DynamicServingCertificateController"
I0408 22:29:47.840654       1 shared_informer.go:247] Caches are synced for client-ca::kube-system::extension-apiserver-authentication::requestheader-client-ca-file
I0408 22:29:47.840699       1 shared_informer.go:247] Caches are synced for RequestHeaderAuthRequestController
I0408 22:29:47.840757       1 shared_informer.go:247] Caches are synced for client-ca::kube-system::extension-apiserver-authentication::client-ca-file
PS /home/ana> 
```

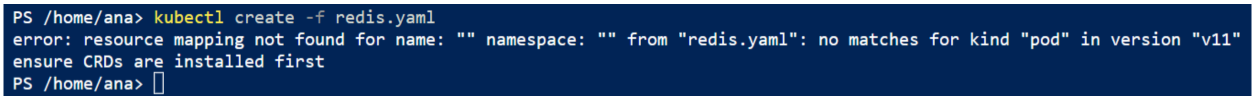
    Errors when either of the container or the namespace are not specified:



```
PS /home/ana> kubectl logs metrics-server-7dd74d8758-lc98h --container metrics-server
Error from server (NotFound): pods "metrics-server-7dd74d8758-lc98h" not found
PS /home/ana> kubectl logs metrics-server-7dd74d8758-lc98h
Error from server (NotFound): pods "metrics-server-7dd74d8758-lc98h" not found
PS /home/ana> kubectl logs metrics-server-7dd74d8758-lc98h -n kube-system
Defaulted container "metrics-server-vpa" out of: metrics-server-vpa, metrics-server
ERROR: logging before flag.Parse: I0408 22:29:15.857596       1 pod_nanny.go:68] Invoked by [/pod_nanny --config-dir=/etc/config --cpu=44m --extra-cpu=0.5m --memory=
51Mi --extra-memory=4Mi --poll-period=180000 --threshold=5 --deployment=metrics-server --container=metrics-server]
ERROR: logging before flag.Parse: I0408 22:29:15.857816       1 pod_nanny.go:69] Version: 1.8.14
ERROR: logging before flag.Parse: I0408 22:29:15.857950       1 pod_nanny.go:85] Watching namespace: kube-system, pod: metrics-server-7dd74d8758-lc98h, container: me
trics-server.
ERROR: logging before flag.Parse: I0408 22:29:15.858070       1 pod_nanny.go:86] storage: MISSING, extra_storage: 0Gi
ERROR: logging before flag.Parse: I0408 22:29:16.548364       1 pod_nanny.go:189] Failed to read data from config file "/etc/config/NannyConfiguration": open /etc/co
nfig/NannyConfiguration: no such file or directory, using default parameters
ERROR: logging before flag.Parse: I0408 22:29:16.548620       1 pod_nanny.go:116] cpu: 44m, extra_cpu: 0.5m, memory: 51Mi, extra_memory: 4Mi
ERROR: logging before flag.Parse: I0408 22:29:16.548643       1 pod_nanny.go:145] Resources: [{Base:{i:{value:44 scale:-3} d:{Dec:<nil>} s:44m Format:DecimalSI} Extr
aPerNode:{i:{value:5 scale:-4} d:{Dec:<nil>} s: Format:DecimalSI} Name:cpu} {Base:{i:{value:53477376 scale:0} d:{Dec:<nil>} s:51Mi Format:BinarySI} ExtraPerNode:{i:{
value:4194304 scale:0} d:{Dec:<nil>} s:4Mi Format:BinarySI} Name:memory}]
ERROR: logging before flag.Parse: E0408 22:29:26.570191       1 nanny_lib.go:128] Get "https://kubernetes-pods-dns-djj2t7xv.hcp.eastus.azmk8s.io:443/api/v1/nodes?res
ourceVersion=0": dial tcp: lookup kubernetes-pods-dns-djj2t7xv.hcp.eastus.azmk8s.io on 10.0.0.10:53: read udp 10.244.0.5:37713->10.0.0.10:53: read: connection refuse
d
PS /home/ana> 
```

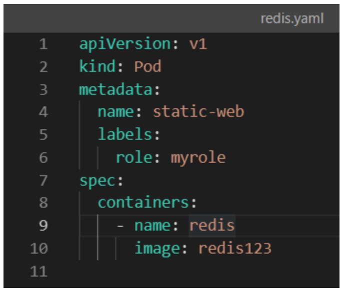
Practice2: Working with pod manifest files

1. Now it is time to deploy pod using manifest file (declarative approach). Copy the following code block on your local computer in a file called redis.yaml:
2. Try to deploy the pod defined in redis.yaml. Run **kubectl create –f redis.yaml**.



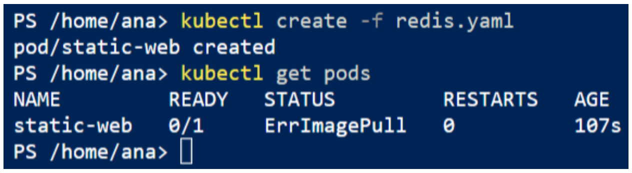
```
PS /home/ana> kubectl create -f redis.yaml
error: resource mapping not found for name: "" namespace: "" from "redis.yaml": no matches for kind "pod" in version "v11"
ensure CRDs are installed first
PS /home/ana> 
```

3. You will receive errors on your screen. Your next task will be to correct the syntax of the code you just copied. You can use the online Kubernetes documentation, or you can search the internet in general.
   - This is the corrected syntax of the redis.yaml file.
     In yaml files everything is indented. In the given code there wasn't any indentation. Next the apiVersion was wrong, it should have been v1, because v11 doesn't exist. In kind, pod needed to be capitalized.



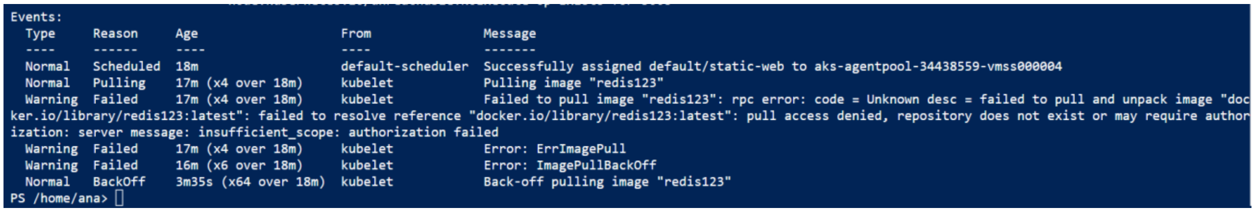
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: static-web
  labels:
    role: myrole
spec:
  containers:
    - name: redis
      image: redis123

```

4. When you solve all the syntax errors your pod should be deployed but is it running? What is the status of your pod?
   - The status of the pod here is ErrImagePull. This means that kubernetes is unable to locate the image.



```
PS /home/ana> kubectl create -f redis.yaml
pod/static-web created
PS /home/ana> kubectl get pods
NAME         READY   STATUS         RESTARTS   AGE
static-web   0/1     ErrImagePull   0          107s
PS /home/ana> 
```

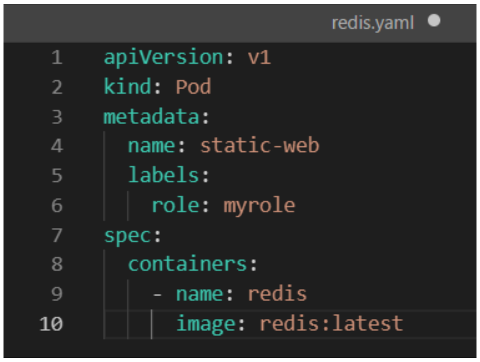
5. Check the events associated with this pod. Run the **kubectl describe pod static-web command**. What are the events showing? Why is your pod not running?
   (This is just the events part)



```
Events:
  Type     Reason     Age                  From               Message
  ----     ------     ----                 ----               -------
  Normal   Scheduled  18m                  default-scheduler  Successfully assigned default/static-web to aks-agentpool-34438559-vmss000004
  Normal   Pulling    17m (x4 over 18m)    kubelet            Pulling image "redis123"
  Warning  Failed     17m (x4 over 18m)    kubelet            Failed to pull image "redis123": rpc error: code = Unknown desc = failed to pull and unpack image "doc
ker.io/library/redis123:latest": failed to resolve reference "docker.io/library/redis123:latest": pull access denied, repository does not exist or may require author
ization: server message: insufficient_scope: authorization failed
  Warning  Failed     17m (x4 over 18m)    kubelet            Error: ErrImagePull
  Warning  Failed     16m (x6 over 18m)    kubelet            Error: ImagePullBackOff
  Normal   BackOff    3m35s (x64 over 18m) kubelet            Back-off pulling image "redis123"
PS /home/ana> 
```

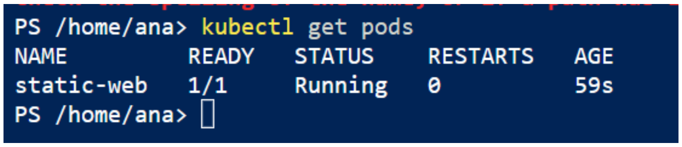
   - The status of the pod here is ErrImagePull. This means that kubernetes is unable to locate the image.

6. Find the correct image (check the Docker hub page) and correct it in the manifest.
   - The correct image was redis:latest instead redis123.



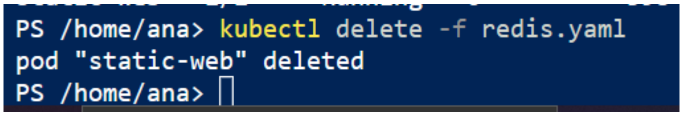
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: static-web
  labels:
    role: myrole
spec:
  containers:
    - name: redis
      image: redis:latest
```

7. Locate the image information and put the correct image name. Redeploy the pod (first run **kubectl delete pod static-web** to delete the pod, then run kubectl create once again).
8. Check the status of your pod. It should be running now.



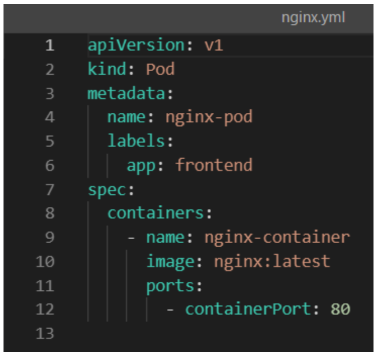
```
PS /home/ana> kubectl get pods
NAME         READY   STATUS    RESTARTS   AGE
static-web   1/1     Running   0          59s
PS /home/ana> 
```

9. Now you can delete the pod. Try to delete it using the **kubectl delete –f redis.yaml**.



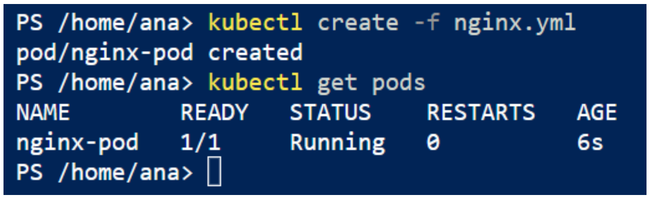
```
PS /home/ana> kubectl delete -f redis.yaml
pod "static-web" deleted
PS /home/ana> 
```

10. Your next task is to create and test nginx pod definition. Your definition should use the nginx official image, should use label named app with value frontend and should publish port 80. Make sure you complete this task because we will use this template in our next Labs. Your nginx pod should be running without any issues.



nginx.yml

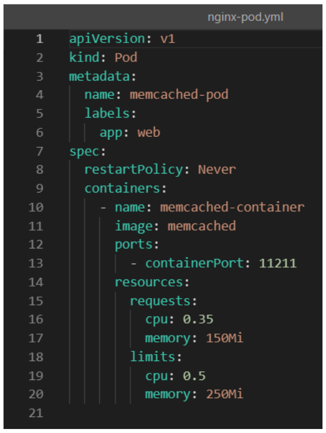
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: frontend
spec:
  containers:
    - name: nginx-container
      image: nginx:latest
      ports:
        - containerPort: 80
```



```
PS /home/ana> kubectl create -f nginx.yml
pod/nginx-pod created
PS /home/ana> kubectl get pods
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          6s
PS /home/ana> 
```

11. Final task of this practice will be to define pod definition with following details:

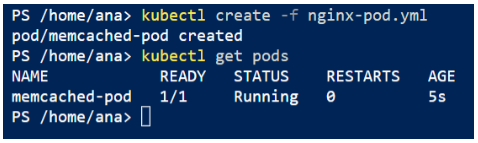
- Image=memcached
- Port= 11211
- Label app=web
- CPU request=0.35 cores
- RAM request=0.15 GB
- CPU limit=0.5 cores
- Ram limit=0.25 GB
- Restart policy=Never



nginx-pod.yml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: memcached-pod
  labels:
    app: web
spec:
  restartPolicy: Never
  containers:
    - name: memcached-container
      image: memcached
      ports:
        - containerPort: 11211
      resources:
        requests:
          cpu: 0.35
          memory: 150Mi
        limits:
          cpu: 0.5
          memory: 250Mi
```

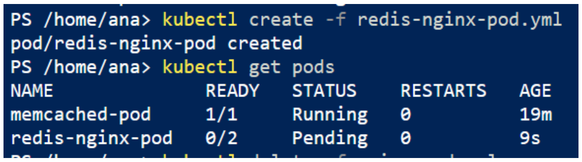
12. Don’t forget to try your pod definition.



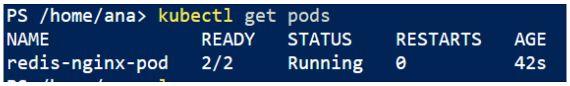
```
PS /home/ana> kubectl create -f nginx-pod.yml
pod/memcached-pod created
PS /home/ana> kubectl get pods
NAME            READY   STATUS    RESTARTS   AGE
memcached-pod   1/1     Running   0          5s
PS /home/ana> 
```

Practice3: Multi-container pods

1. Once finished you can try to create multi-container pod definition. Your multi-container pod should use redis and nginx containers with port 6379 and 80 published respectively. Label name should be app with value web.
2. Note that in reality there is no sense to put the redis and nginx under the same pod but it can be done for the purpose of learning.
3. Deploy your multi-container pod. It should have running status.
   What is written under Ready column when you kubectl get the pods? – 0/2 in the first running of **kubectl get pods**, and 2/2 when we run **kubectl get pods** for the second time.
   Why your pod displays different values for ready? – When the command is ran for the first time the containers were still starting up, but for the second time the command is ran the containers had finished starting up.

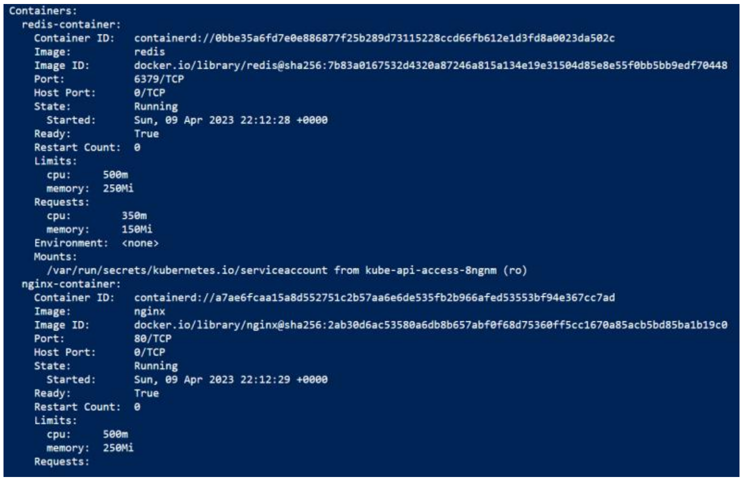


```
PS /home/ana> kubectl create -f redis-nginx-pod.yml
pod/redis-nginx-pod created
PS /home/ana> kubectl get pods
NAME              READY   STATUS    RESTARTS   AGE
memcached-pod     1/1     Running   0          19m
redis-nginx-pod   0/2     Pending   0          9s
```



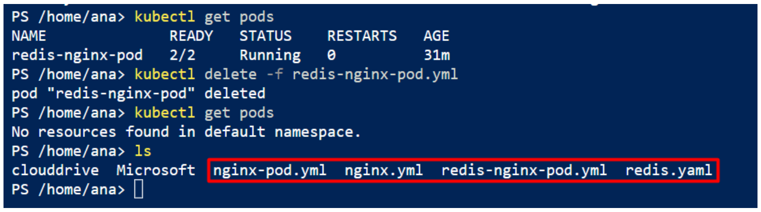
```
PS /home/ana> kubectl get pods
NAME              READY   STATUS    RESTARTS   AGE
redis-nginx-pod   2/2     Running   0          42s
```

4. Kubectl describe your new pod, and locate the containers section. How many containers are listed?
   - To describe our new pod we use **kubectl describe pod redis-nginx-pod**. And there are 2 containers.



```
Containers:
  redis-container:
    Container ID:   containerd://0bbe35a6fd7e0e886877f25b289d73115228ccd66fb612e1d3fd8a0023da502c
    Image:          redis
    Image ID:       docker.io/library/redis@sha256:7b83a0167532d4320a87246a815a134e19e31504d85e8e55f0bb5bb9edf70448
    Port:           6379/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Sun, 09 Apr 2023 22:12:28 +0000
    Ready:          True
    Restart Count:  0
    Limits:
      cpu:     500m
      memory:  250Mi
    Requests:
      cpu:        350m
      memory:     150Mi
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-8ngnm (ro)
  nginx-container:
    Container ID:   containerd://a7ae6fcaa15a8d552751c2b57aa6e6de535fb2b966afed53553bf94e367cc7ad
    Image:          nginx
    Image ID:       docker.io/library/nginx@sha256:2ab30d6ac53580a6db8b657abf0f68d75360ff5cc1670a85acb5bd85ba1b19c0
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Sun, 09 Apr 2023 22:12:29 +0000
    Ready:          True
    Restart Count:  0
    Limits:
      cpu:     500m
      memory:  250Mi
    Requests:
```

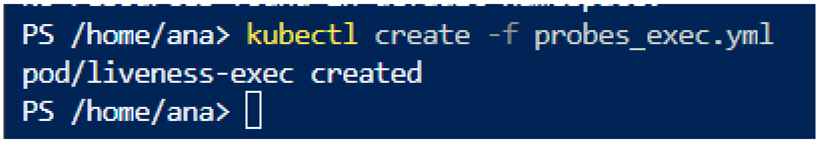
5. Delete all the pods under the default namespace.
6. Don’t delete any of the manifest files you have created so far.
   - Before deleting the pod we first list all the pods in the default namespace, and confirm that the redis-nginx-pod pod is present. After we confirm that the redis-nginx-pod pod is present we delete it.
   - **kubectl delete -f redis-nginx-pod.yml** command does not delete the manifest file we created.



```
PS /home/ana> kubectl get pods
NAME              READY   STATUS    RESTARTS   AGE
redis-nginx-pod   2/2     Running   0          31m
PS /home/ana> kubectl delete -f redis-nginx-pod.yml
pod "redis-nginx-pod" deleted
PS /home/ana> kubectl get pods
No resources found in default namespace.
PS /home/ana> ls
clouddrive  Microsoft  nginx-pod.yml  nginx.yml  redis-nginx-pod.yml  redis.yaml
PS /home/ana> 
```

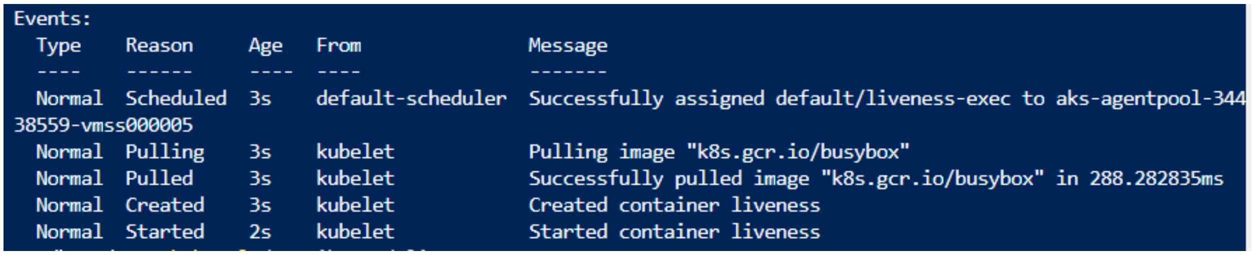
Practice4: Probes

3. Run kubectl create –f probes_exec.yaml.



```
PS /home/ana> kubectl create -f probes_exec.yml
pod/liveness-exec created
PS /home/ana> 
```

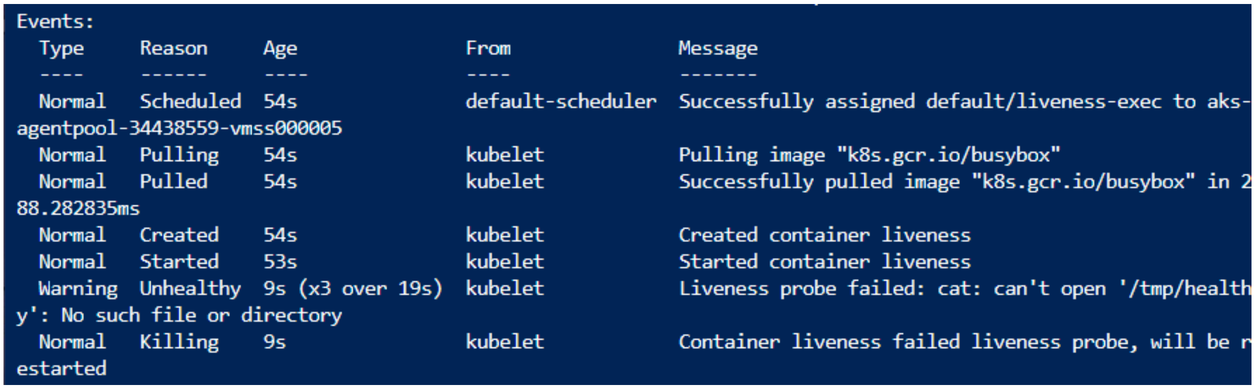
4. Run **kubectl describe pod liveness-exec** immediately after you deploy the pod. The output should indicate that no liveness probes have failed yet.



```
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  3s    default-scheduler  Successfully assigned default/liveness-exec to aks-agentpool-344
38559-vmss000005
  Normal  Pulling    3s    kubelet            Pulling image "k8s.gcr.io/busybox"
  Normal  Pulled     3s    kubelet            Successfully pulled image "k8s.gcr.io/busybox" in 288.282835ms
  Normal  Created    3s    kubelet            Created container liveness
  Normal  Started    2s    kubelet            Started container liveness
```

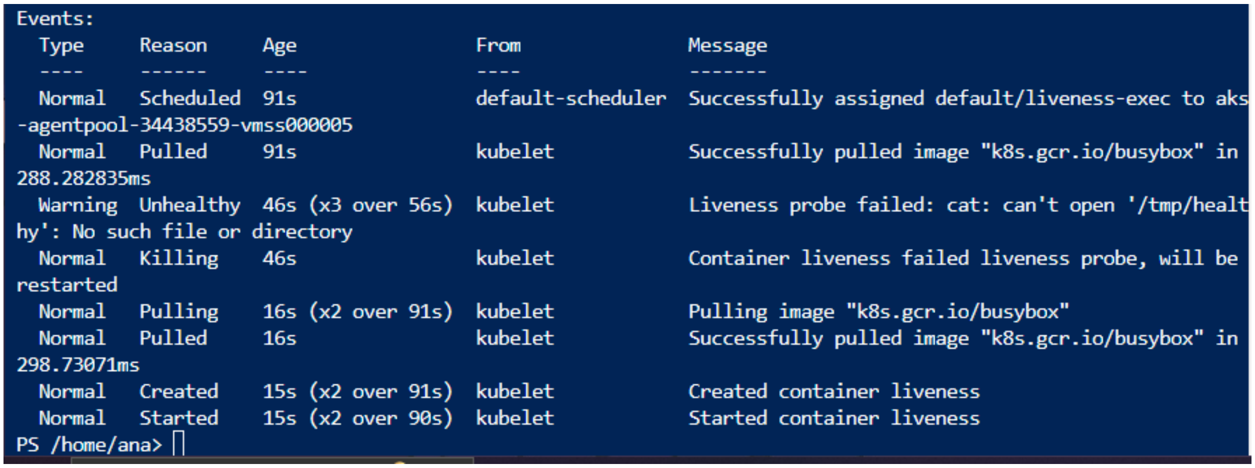
5. After 35 seconds, view the Pod events again. Run **kubectl describe pod liveness-exec**.

6. At the bottom of the output, there should be a messages indicating that the liveness probes have failed, and the containers have been killed and recreated.

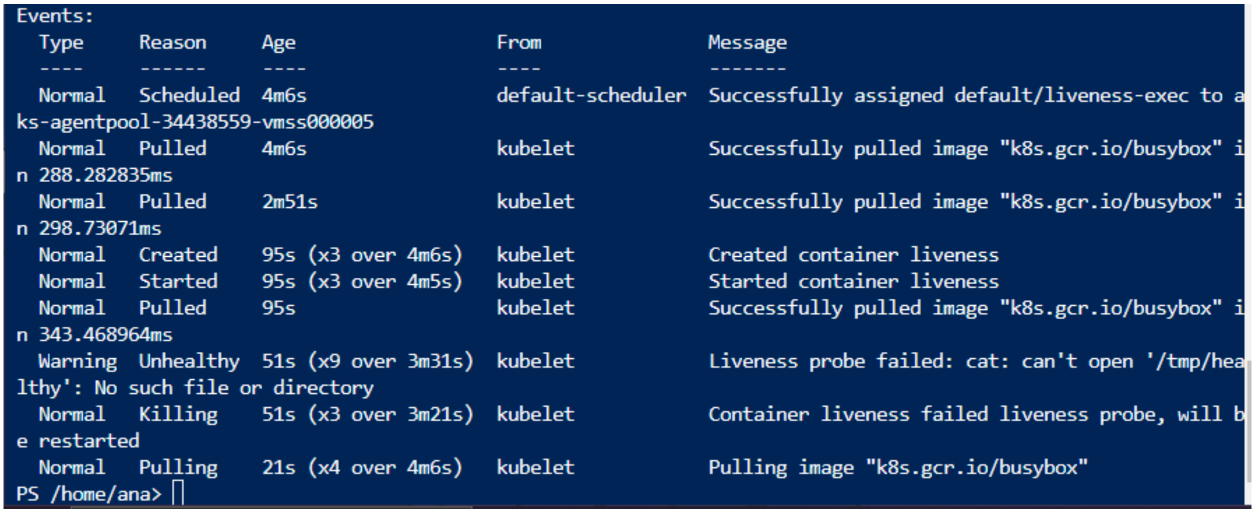


```
Events:
  Type     Reason     Age                From               Message
  ----     ------     ----               ----               -------
  Normal   Scheduled  54s                default-scheduler  Successfully assigned default/liveness-exec to aks-
agentpool-34438559-vmss000005
  Normal   Pulling    54s                kubelet            Pulling image "k8s.gcr.io/busybox"
  Normal   Pulled     54s                kubelet            Successfully pulled image "k8s.gcr.io/busybox" in 2
88.282835ms
  Normal   Created    54s                kubelet            Created container liveness
  Normal   Started    53s                kubelet            Started container liveness
  Warning  Unhealthy  9s (x3 over 19s)   kubelet            Liveness probe failed: cat: can't open '/tmp/health
y': No such file or directory
  Normal   Killing    9s                 kubelet            Container liveness failed liveness probe, will be r
estarted
```

7. Wait another 30 seconds, and verify that the container has been restarted. Run **kubectl get pod liveness-exec**.

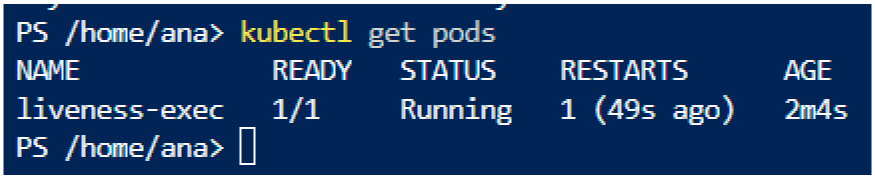


```
Events:
  Type     Reason     Age                From               Message
  ----     ------     ----               ----               -------
  Normal   Scheduled  91s                default-scheduler  Successfully assigned default/liveness-exec to aks
-agentpool-34438559-vmss000005
  Normal   Pulled     91s                kubelet            Successfully pulled image "k8s.gcr.io/busybox" in
288.282835ms
  Warning  Unhealthy  46s (x3 over 56s)  kubelet            Liveness probe failed: cat: can't open '/tmp/healt
hy': No such file or directory
  Normal   Killing    46s                kubelet            Container liveness failed liveness probe, will be
restarted
  Normal   Pulling    16s (x2 over 91s)  kubelet            Pulling image "k8s.gcr.io/busybox"
  Normal   Pulled     16s                kubelet            Successfully pulled image "k8s.gcr.io/busybox" in
298.73071ms
  Normal   Created    15s (x2 over 91s)  kubelet            Created container liveness
  Normal   Started    15s (x2 over 90s)  kubelet            Started container liveness
PS /home/ana> 
```



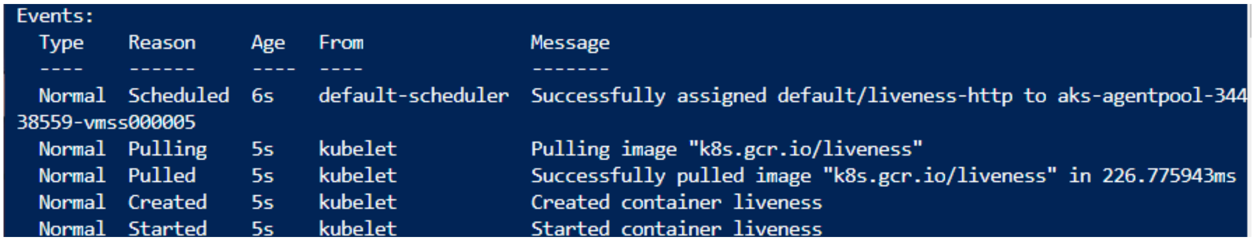
```
Events:
  Type     Reason     Age                  From               Message
  ----     ------     ----                 ----               -------
  Normal   Scheduled  4m6s                 default-scheduler  Successfully assigned default/liveness-exec to a
ks-agentpool-34438559-vmss000005
  Normal   Pulled     4m6s                 kubelet            Successfully pulled image "k8s.gcr.io/busybox" i
n 288.282835ms
  Normal   Pulled     2m51s                kubelet            Successfully pulled image "k8s.gcr.io/busybox" i
n 298.73071ms
  Normal   Created    95s (x3 over 4m6s)   kubelet            Created container liveness
  Normal   Started    95s (x3 over 4m5s)   kubelet            Started container liveness
  Normal   Pulled     95s                  kubelet            Successfully pulled image "k8s.gcr.io/busybox" i
n 343.468964ms
  Warning  Unhealthy  51s (x9 over 3m31s)  kubelet            Liveness probe failed: cat: can't open '/tmp/hea
lthy': No such file or directory
  Normal   Killing    51s (x3 over 3m21s)  kubelet            Container liveness failed liveness probe, will b
e restarted
  Normal   Pulling    21s (x4 over 4m6s)   kubelet            Pulling image "k8s.gcr.io/busybox"
PS /home/ana> 
```

8. The output should show that RESTARTS has been incremented.



```
PS /home/ana> kubectl get pods
NAME            READY   STATUS    RESTARTS      AGE
liveness-exec   1/1     Running   1 (49s ago)   2m4s
PS /home/ana> 
```

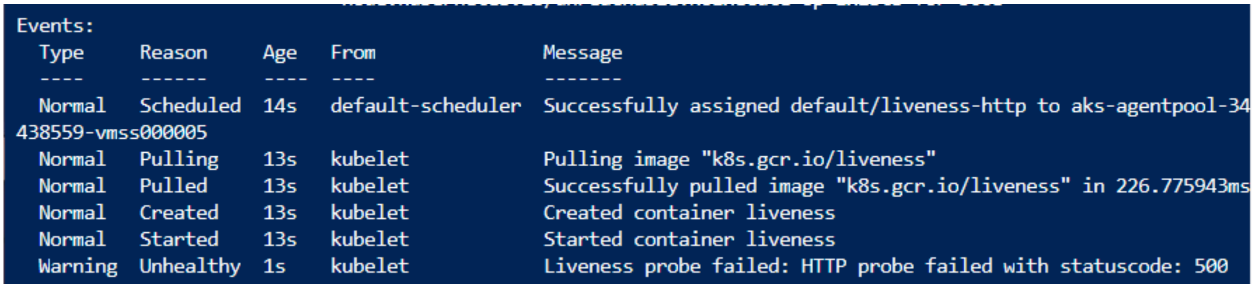
11. For the first 10 seconds that the container is alive, the /healthz handler returns a status of 200. After that, the handler returns a status of 500.



```
Events:
  Type    Reason     Age  From               Message
  ----    ------     ---- ----               -------
  Normal  Scheduled  6s   default-scheduler  Successfully assigned default/liveness-http to aks-agentpool-344
38559-vmss000005
  Normal  Pulling    5s   kubelet            Pulling image "k8s.gcr.io/liveness"
  Normal  Pulled     5s   kubelet            Successfully pulled image "k8s.gcr.io/liveness" in 226.775943ms
  Normal  Created    5s   kubelet            Created container liveness
  Normal  Started    5s   kubelet            Started container liveness
```

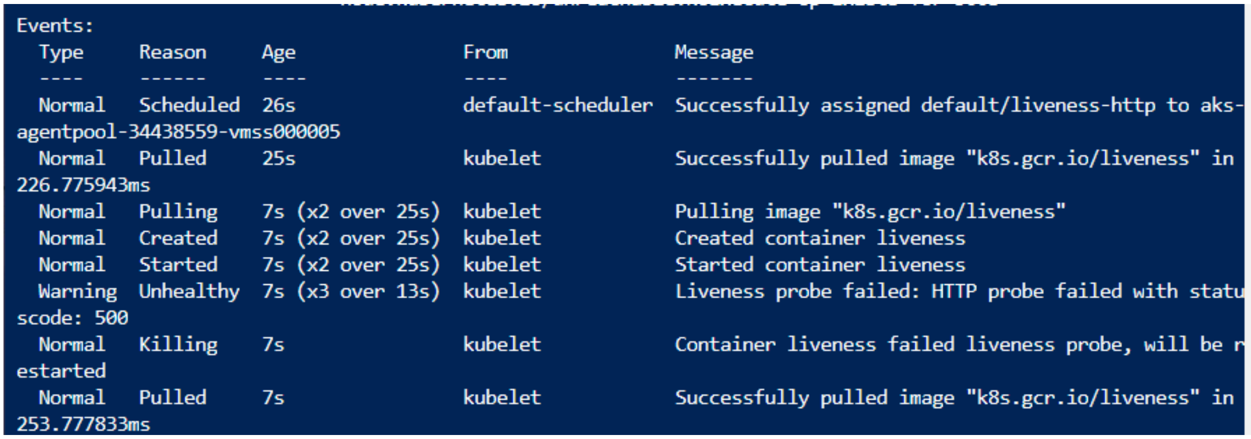
13. Immediately run (you only have 10 secs to run this command) kubectl describe pod liveness-http.

14. Your pod should be live and running.



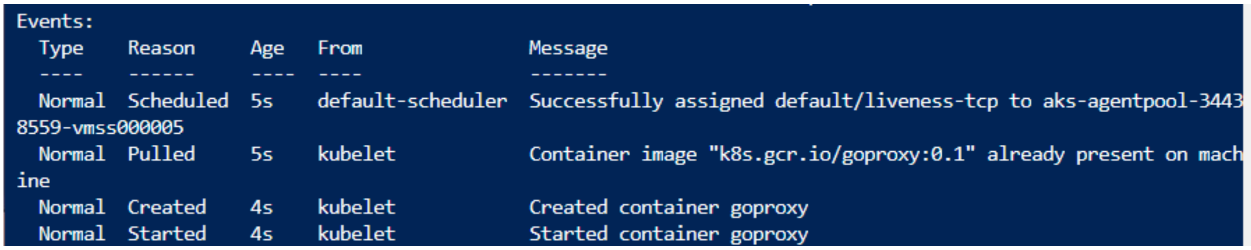
```
Events:
  Type     Reason     Age   From               Message
  ----     ------     ----  ----               -------
  Normal   Scheduled  14s   default-scheduler  Successfully assigned default/liveness-http to aks-agentpool-34
438559-vmss000005
  Normal   Pulling    13s   kubelet            Pulling image "k8s.gcr.io/liveness"
  Normal   Pulled     13s   kubelet            Successfully pulled image "k8s.gcr.io/liveness" in 226.775943ms
  Normal   Created    13s   kubelet            Created container liveness
  Normal   Started    13s   kubelet            Started container liveness
  Warning  Unhealthy  1s    kubelet            Liveness probe failed: HTTP probe failed with statuscode: 500
```

15. After 10 seconds, view Pod events to verify that liveness probes have failed and the container has been restarted. Run again **kubectl describe pod liveness-http**.



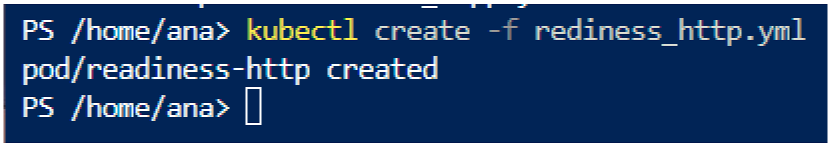
```
Events:
  Type     Reason     Age                From               Message
  ----     ------     ----               ----               -------
  Normal   Scheduled  26s                default-scheduler  Successfully assigned default/liveness-http to aks-
agentpool-34438559-vmss000005
  Normal   Pulled     25s                kubelet            Successfully pulled image "k8s.gcr.io/liveness" in
226.775943ms
  Normal   Pulling    7s (x2 over 25s)   kubelet            Pulling image "k8s.gcr.io/liveness"
  Normal   Created    7s (x2 over 25s)   kubelet            Created container liveness
  Normal   Started    7s (x2 over 25s)   kubelet            Started container liveness
  Warning  Unhealthy  7s (x3 over 13s)   kubelet            Liveness probe failed: HTTP probe failed with statu
scode: 500
  Normal   Killing    7s                 kubelet            Container liveness failed liveness probe, will be r
estarted
  Normal   Pulled     7s                 kubelet            Successfully pulled image "k8s.gcr.io/liveness" in
253.777833ms
```

16. You should see the same output as in step 7. Kubelet will reboot the container.
17.
18. Run **kubectl create –f probes_tcp.yaml**.



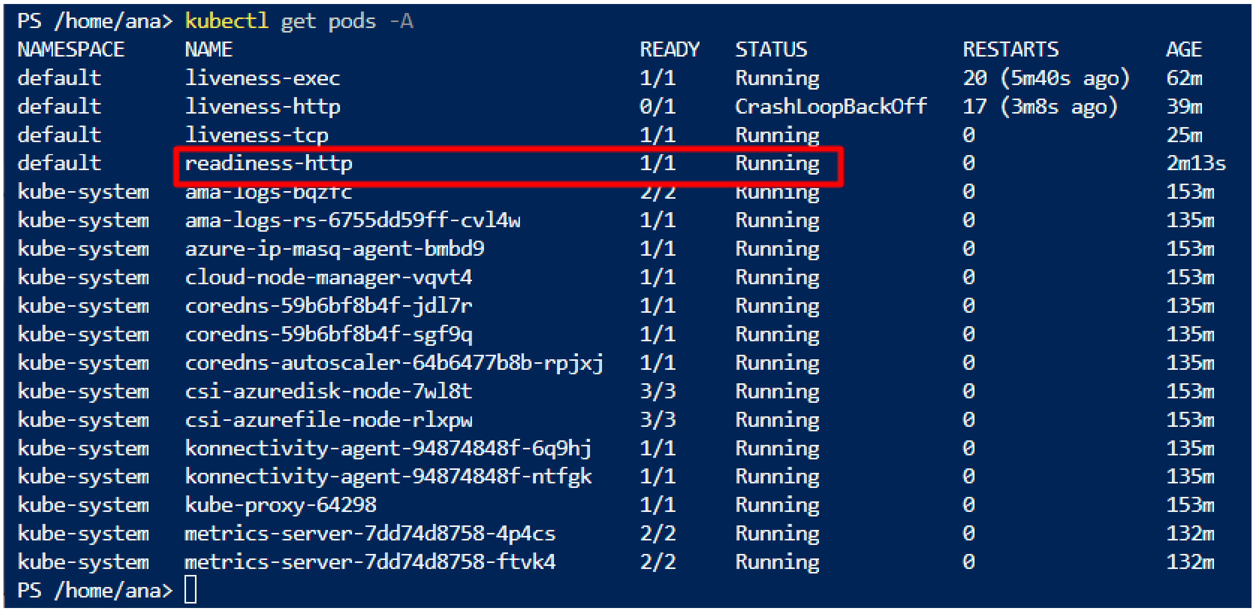
```
Events:
  Type     Reason     Age   From               Message
  ----     ------     ----  ----               -------
  Normal   Scheduled  5s    default-scheduler  Successfully assigned default/liveness-tcp to aks-agentpool-3443
8559-vmss000005
  Normal   Pulled     5s    kubelet            Container image "k8s.gcr.io/goproxy:0.1" already present on mach
ine
  Normal   Created    4s    kubelet            Created container goproxy
  Normal   Started    4s    kubelet            Started container goproxy
```

19. Immediately run (you only have 10 secs to run this command) **kubectl describe pod liveness-tcp**.
20.
21. After 10 seconds, view Pod events to verify that liveness probes have failed and the container has been restarted. Run again **kubectl describe pod liveness-tcp**.
22.
23.
24.
25. Run **kubectl create –f readiness_http.yam**l.



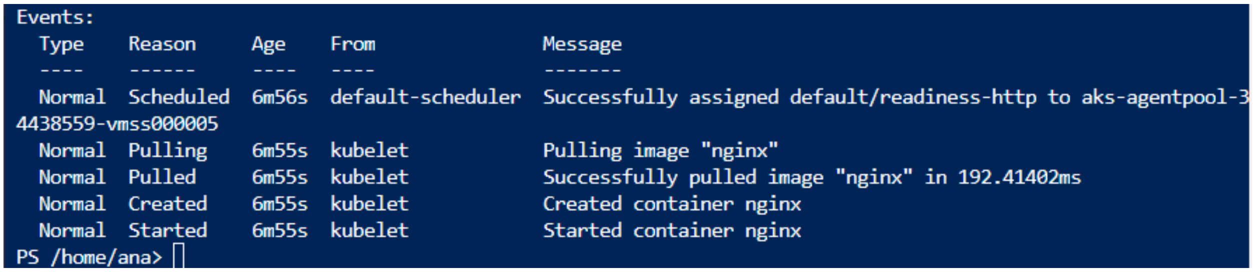
```
PS /home/ana> kubectl create -f rediness_http.yml
pod/readiness-http created
PS /home/ana> 
```

26. Run **kubectl get pods –A** to see the status of your pod.
27. Pods and their status and ready states will be displayed; our pod should be in running state.



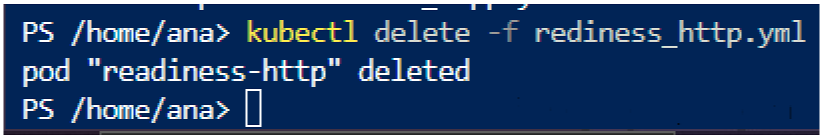
```
PS /home/ana> kubectl get pods -A
NAMESPACE     NAME                                   READY   STATUS             RESTARTS       AGE
default       liveness-exec                          1/1     Running            20 (5m40s ago) 62m
default       liveness-http                          0/1     CrashLoopBackOff   17 (3m8s ago)  39m
default       liveness-tcp                           1/1     Running            0              25m
default       readiness-http                         1/1     Running            0              2m13s
kube-system   ama-logs-bqzfc                         2/2     Running            0              153m
kube-system   ama-logs-rs-6755dd59ff-cvl4w           1/1     Running            0              135m
kube-system   azure-ip-masq-agent-bmbd9              1/1     Running            0              153m
kube-system   cloud-node-manager-vqvt4               1/1     Running            0              153m
kube-system   coredns-59b6bf8b4f-jdl7r               1/1     Running            0              135m
kube-system   coredns-59b6bf8b4f-sgf9q               1/1     Running            0              135m
kube-system   coredns-autoscaler-64b6477b8b-rpjxj    1/1     Running            0              135m
kube-system   csi-azuredisk-node-7wl8t               3/3     Running            0              153m
kube-system   csi-azurefile-node-rlxpw               3/3     Running            0              153m
kube-system   konnectivity-agent-94874848f-6q9hj     1/1     Running            0              135m
kube-system   konnectivity-agent-94874848f-ntfgk     1/1     Running            0              135m
kube-system   kube-proxy-64298                       1/1     Running            0              153m
kube-system   metrics-server-7dd74d8758-4p4cs        2/2     Running            0              132m
kube-system   metrics-server-7dd74d8758-ftvk4        2/2     Running            0              132m
PS /home/ana> 
```

28. Run **kubectl describe pod readiness-http**. Examine the events for this pod. Everything should be OK.



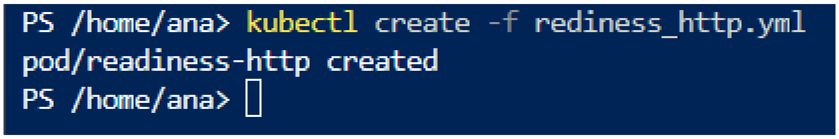
```
Events:
  Type    Reason     Age    From               Message
  ----    ------     ----   ----               -------
  Normal  Scheduled  6m56s  default-scheduler  Successfully assigned default/readiness-http to aks-agentpool-3
4438559-vmss000005
  Normal  Pulling    6m55s  kubelet            Pulling image "nginx"
  Normal  Pulled     6m55s  kubelet            Successfully pulled image "nginx" in 192.41402ms
  Normal  Created    6m55s  kubelet            Created container nginx
  Normal  Started    6m55s  kubelet            Started container nginx
PS /home/ana> 
```

29. Now delete the pod and edit the readiness_http.yaml so that the port parameter has 81 value.



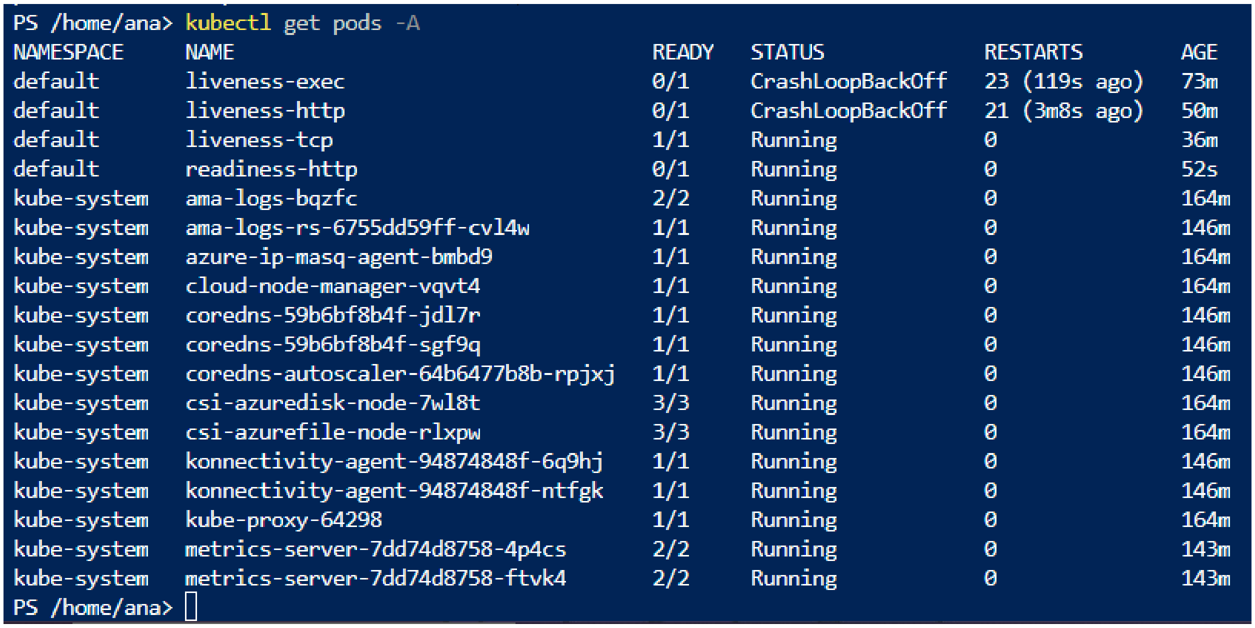
```
PS /home/ana> kubectl delete -f rediness_http.yml
pod "readiness-http" deleted
PS /home/ana> 
```

30. Run again **kubectl create –f readiness_http.yaml**.



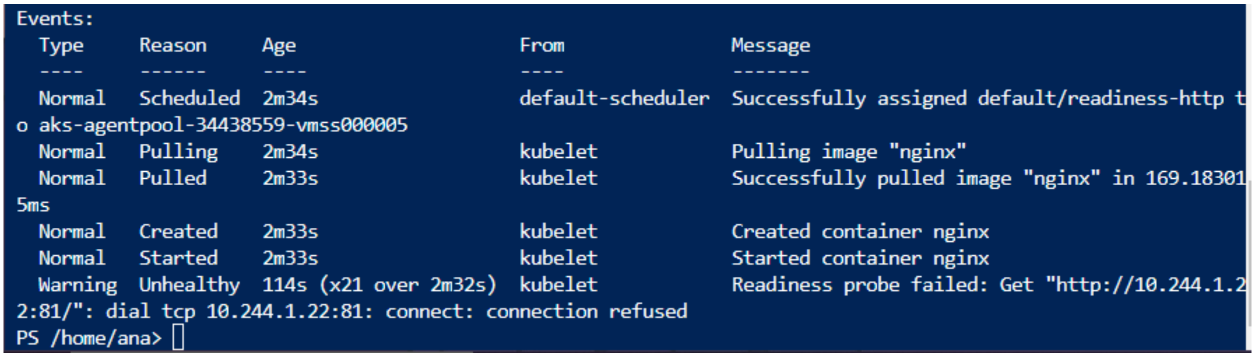
```
PS /home/ana> kubectl create -f rediness_http.yml
pod/readiness-http created
PS /home/ana> 
```

31. Run **kubectl get pods –A** to see the status of your pod. You should see that the pod is running but it is not in ready state.



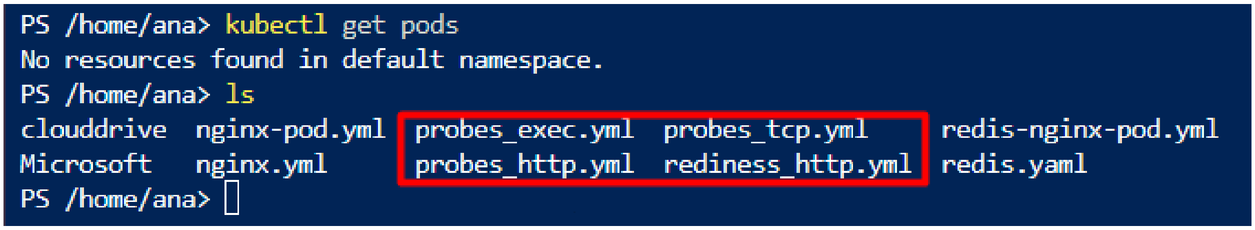
```
PS /home/ana> kubectl get pods -A
NAMESPACE     NAME                                  READY   STATUS             RESTARTS       AGE
default       liveness-exec                         0/1     CrashLoopBackOff   23 (119s ago)  73m
default       liveness-http                         0/1     CrashLoopBackOff   21 (3m8s ago)  50m
default       liveness-tcp                          1/1     Running            0              36m
default       readiness-http                        0/1     Running            0              52s
kube-system   ama-logs-bqzfc                        2/2     Running            0              164m
kube-system   ama-logs-rs-6755dd59ff-cvl4w          1/1     Running            0              146m
kube-system   azure-ip-masq-agent-bmbd9             1/1     Running            0              164m
kube-system   cloud-node-manager-vqvt4              1/1     Running            0              164m
kube-system   coredns-59b6bf8b4f-jdl7r              1/1     Running            0              146m
kube-system   coredns-59b6bf8b4f-sgf9q              1/1     Running            0              146m
kube-system   coredns-autoscaler-64b6477b8b-rpjxj   1/1     Running            0              146m
kube-system   csi-azuredisk-node-7wl8t              3/3     Running            0              164m
kube-system   csi-azurefile-node-rlxpw              3/3     Running            0              164m
kube-system   konnectivity-agent-94874848f-6q9hj    1/1     Running            0              146m
kube-system   konnectivity-agent-94874848f-ntfgk    1/1     Running            0              146m
kube-system   kube-proxy-64298                      1/1     Running            0              164m
kube-system   metrics-server-7dd74d8758-4p4cs       2/2     Running            0              143m
kube-system   metrics-server-7dd74d8758-ftvk4       2/2     Running            0              143m
PS /home/ana> 
```

32. Describe the pod. Run **kubectl describe pod readiness-http**.



```
Events:
  Type     Reason     Age                   From               Message
  ----     ------     ----                  ----               -------
  Normal   Scheduled  2m34s                 default-scheduler  Successfully assigned default/readiness-http t
o aks-agentpool-34438559-vmss000005
  Normal   Pulling    2m34s                 kubelet            Pulling image "nginx"
  Normal   Pulled     2m33s                 kubelet            Successfully pulled image "nginx" in 169.18301
5ms
  Normal   Created    2m33s                 kubelet            Created container nginx
  Normal   Started    2m33s                 kubelet            Started container nginx
  Warning  Unhealthy  114s (x21 over 2m32s)  kubelet           Readiness probe failed: Get "http://10.244.1.2
2:81/": dial tcp 10.244.1.22:81: connect: connection refused
PS /home/ana> 
```

33. From the events we can see that readiness probe failed due to the connection being refused therefore pod will not receive any traffic.
34. Delete all pods under the default namespace.
35. Don't delete any manifest files created so far.



```
PS /home/ana> kubectl get pods
No resources found in default namespace.
PS /home/ana> ls
clouddrive   nginx-pod.yml   probes_exec.yml   probes_tcp.yml       redis-nginx-pod.yml
Microsoft    nginx.yml       probes_http.yml   rediness_http.yml    redis.yaml
PS /home/ana> 
```