Kubernetes

Notes

Issues with local ARM based cluster - non comptabile images

Runing on AKS, with one slight issue in probes http

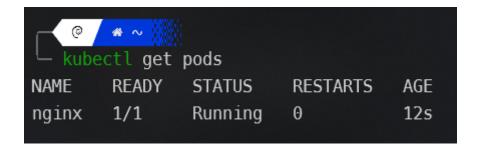
1. Simple pods operations:

No pods available after querying

While using --all-namespaces we see pods(21) under the kube-system namespace. These pods are made and owned by the cluster - basically services that make using k8s possible

```
NAMESPACE
                NAME
                                                                                 RESTARTS
kube-system ama-logs-qxsfd
                                                                                              17m
kube-system ama-logs-rs-6d99f5c8dd-d4476
                                                                     Running
                                                                                              11h
kube-system azure-ip-masq-agent-l6h57
kube-system cloud-node-manager-9gsvr
                                                                     Running
                                                                                              17m
                                                                     Running
                                                                                              17m
kube-system coredns-59b6bf8b4f-4ptgd
kube-system coredns-59b6bf8b4f-ntb9n
                                                                     Running
kube-system coredns-autoscaler-5f9cb57949-sqfqz
kube-system csi-azuredisk-node-7hr9k
kube-system csi-azurefile-node-4c577
kube-system konnectivity-agent-75f989679d-rbbbg
kube-system konnectivity-agent-75f989679d-wtzdg
                                                                     Running
kube-system kube-proxy-5qt2v
                                                                     Running
                                                                                              17m
kube-system metrics-server-7dd74d8758-j4vrz
                                                                     Running
                                                                                              16m
kube-system metrics-server-7dd74d8758-rfk6v
                                                                     Running
                                                                                              16m
```

Afte executing kubectl run nginx --image=nginx and querying the pod is runing and well.



Runing logs gives information about the pod - version, process status etc

```
└─ 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
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/04/05 05:49:00 [notice] 1#1: using the "epoll" event method
2023/04/05 05:49:00 [notice] 1#1: nginx/1.23.4
2023/04/05 05:49:00 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/04/05 05:49:00 [notice] 1#1: 0S: Linux 5.4.0-1104-azure
2023/04/05 05:49:00 [notice] 1#1: qetrlimit(RLIMIT NOFILE): 1048576:1048576
2023/04/05 05:49:00 [notice] 1#1: start worker processes
2023/04/05 05:49:00 [notice] 1#1: start worker process 29
2023/04/05 05:49:00 [notice] 1#1: start worker process 30
```

Using -o wide gives us more information including the node where the pod is running



Using describe gives us a detailed state of a pod

kubectl describe pod nginx Name: nginx Namespace: default Θ Priority: Service Account: default aks-agentpool-41905252-vmss000003/10.224.0.4 Node: Start lime: Wed, 05 Apr 2023 08:48:56 +0300 Labels: run=nginx Annotations: <none> Status: Running IP: 10.224.0.80

Container Image is pulled from a microsoft repository or better know as Microsoft Artifact Registry

- kubectl describe pod coredns-59b6bf8b4f-4ptqd -n kube-system

Name: coredns-59b6bf8b4f-4ptgd

Namespace: kube-system Priority: 2000001000

Priority Class Name: system-node-critical

Service Account: coredns

Node: aks-agentpool-41905252-vmss000003/10.224.0.4

Start Time: Wed, 05 Apr 2023 08:23:38 +0300

Labels: k8s-app=kube-dns

kubernetes.io/cluster-service=true

pod-template-hash=59b6bf8b4f

version=v20

Annotations: prometheus.io/port: 9153

Status: Running
IP: 10.224.0.22

IPs:

IPs:

IP: 10.224.0.22

Controlled By: ReplicaSet/coredns-59b6bf8b4f

Containers: coredns:

Container ID: containerd://cdf5833b7f3f0ae28efd85f195604337b2f92b1377af1d1aba5b0e95d22f949a

Image: mcr.microsoft.com/oss/kubernetes/coredns:v1.9.3

1mage 1D: snazbe:cადუყენის42ანნდანცისთისთისით აგანადეგის ართან ართანის ართანი

Ports: 53/UDP, 53/TCP, 9153/TCP Host Ports: 0/UDP, 0/TCP, 0/TCP

ToDo:

What is a pod_nany

2. Working with pod manifest files:

For the sake of being brief I'll compare both files - wrong and correct versions.

- apiVersion is wrong for pods should be v1 not v11
- Formatting isn't according to yaml standard
- it's spec, not specs typo
- image redis123 doesn't exist(we get a pull image error), we should check the image repo(Docker hub) for proper naming/version

```
apiVersion:v1
                              1+ apiVersion: v1
                              2+ kind: Pod
kind:pod
metadata:
                                 metadata:
name: static-web
                                   name: static-web
                                   labels:
labels:
role: myrole
                                     role: myrole
specs:
                              7+ spec:
containers:
                                   containers:
name: redis
                                   - name: redis
image: redis123
                                     image: redis:latest
                             10+
                                     ports:
                             12-
                                     containerPort: 6379
```

First run and we get a parsing error

After the file has been corrected we get another error this time because of the image name

```
~/Homework/Homework-13/manifets
                                              ■ № master !1 ?2
   kubectl get pods
NAME
             READY
                      STATUS
                                 RESTARTS
                                            AGE
static-web
              1/1
                      Running
                                            12s
└ kubectl get pods
NAME
             READY
                      STATUS
                                      RESTARTS
                                                  AGE
static-web
             \theta/1
                      ErrImagePull
                                                  10s
```

And here we have a healthy pod with a container.

nginx manifest

apiVersion: v1
kind: Pod
metadata:

name: nginx-pod

labels:

app: frontend

spec:

containers:

- name: nginx-container

image: nginx

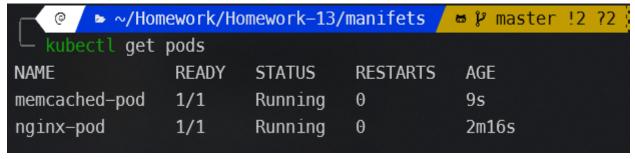
ports:

- containerPort: 80

Memecache manifest

```
apiVersion: v1
kind: Pod
metadata:
  name: memcached-pod
  labels:
    app: web
spec:
  containers:
  - name: memcached-container
    image: memcached
    ports:
    - containerPort: 11211
    resources:
      requests:
        cpu: 350m
        memory: 150Mi
      limits:
        cpu: 500m
        memory: 250Mi
    restartPolicy: Never
```

Both nignx and memcache running

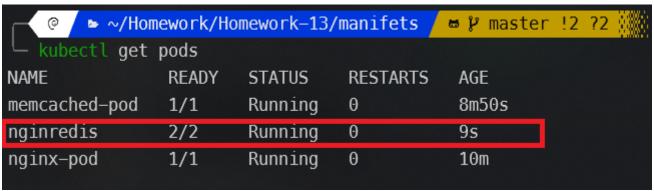


3. Multicontainer pods

Definition of our multicontainer pod

```
apiVersion: v1
kind: Pod
metadata:
  name: nginredis
  labels:
    app: frontend-backend
spec:
  containers:
  - name: nginx-container
    image: nginx:1.22.1
    ports:
    - containerPort: 80
  - name: redis-container
    image: redis:latest
    ports:
    - containerPort: 6379
```

The pod running with two containers inside



Both containers are pulled from their respective image adresses

Containers: nginx-container: Container ID: containerd://723b00ce71de88010272b6322debf7bb1e82b1d320b64872dfcf46728e62a94e Image: nginx:1.22.1 Image ID: docker.io/library/nginx@sha256:fc5f5fb7574755c306aaf88456ebfbe0b006420a184d52b923d2f0197108f6b7 Port: Host Port: 0/ТСР State: Running Started: Wed, 05 Apr 2023 11:40:25 +0300 True Ready: Restart Count: 0 Environment: <none> Mounts: /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-dkt9g (ro) redis-container: Container ID: containerd://7467493cc297e9da70c8a2df9a8f700a6648a69aeb7d070b4624b648e86bf104 Image: redis:latest Image ID: docker.io/library/redis@sha256:7b83a0167532d4320a87246a815a134e19e31504d85e8e55f0bb5bb9edf70448 Port: 6379/TCP 0/TCP Runnir Host Port: State: Running Started: Wed, 05 Apr 2023 11:40:25 +0300 Ready: True Restart Count: 0 Environment: <none> Mounts: /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-dkt9g (ro)

4. Probes

Liveness probe

Edited yaml file for probes exec

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    test: liveness
  name: liveness-exec
spec:
  containers:
    - name: liveness
      image: k8s.gcr.io/busybox
      args:
        - /bin/sh
        - -c
        - touch /tmp/healthy; sleep 30; rm -rf /tmp/healthy; sleep 600
      livenessProbe:
        exec:
          command:
            - cat
            - /tmp/healthy
        initialDelaySeconds: 5
        periodSeconds: 5
```

Running the describe command - no failure yet

Events:			IIVaci IIabol III coo I	To an eachast connected of Enters to 5000
Type	Reason	Age	From	Message
Normal	Scheduled	5 s	default-scheduler	Successfully assigned default/liveness-exec to aks-agentpool-41905252-vmss00000
Normal	Pulling	4s	kubelet	Pulling image "busybox"
Normal	Pulled	4s	kubelet	Successfully pulled image "busybox" in 156.36596ms
Normal	Created	4s	kubelet	Created container liveness
Normal	Started	4s	kubelet	Started container liveness

Getting the failure after 35 seconds.

```
Events:
 Type
          Reason
                    Age From
                                            Message
 Normal Scheduled 38s default-scheduler Successfully assigned default/liveness-exec to aks-agentpool-41905252-vmss000007
 Normal
         Pulling
                    37s kubelet
                                            Pulling image "busybox"
         Pulled
 Normal
                    37s kubelet
                                            Successfully pulled image "busybox" in 156.36596ms
 Normal
         Created
                          kubelet
                                            Created container liveness
 Normal
          Started
                          kubelet
                                            Started container liveness
 Warning Unhealthy 2s
                          kubelet
                                            Liveness probe failed: cat: can't open '/tmp/healthy': No such file or directory
```

```
~/Homework/Homework-13/manifets / # $ master wip !2 ?1
                                                                                                          Kluster-01 *
    get pods -w
                READY
                        STATUS
                                  RESTARTS
                                                 AGE
liveness-exec
                1/1
                        Running
                                  1 (26s ago)
                                                 102s
liveness-exec
                1/1
                        Running
                                  2 (1s ago)
                                                 2m32s
                                                 3m47s
liveness-exec
                1/1
                        Running
                                  3 (1s ago)
```

liveness_http

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    test: liveness
  name: liveness-http
spec:
  containers:
    - name: liveness
      image: k8s.gcr.io/liveness
      args:
        - /server
      livenessProbe:
        httpGet:
          path: /healthz
          port: 8080
          httpHeaders:
            - name: Custom-Header
              value: Awesome
        initialDelaySeconds: 3
        periodSeconds: 3
```

healthz is implemented in Go

Some issues related to running the healthz, might be related to incorrectly configured AKS

```
Events:
                                                           Message
 Type
         Reason
                    Age
         Scheduled 2m7s
                                        default-scheduler Successfully assigned default/liveness-http to aks-agentpool-41905252-vmss000009
 Normal
                                                           Successfully pulled image "k8s.gcr.io/liveness" in 282.344487ms
 Normal
         Pulled
                     2m7s
                                        kubelet
                                                           Successfully pulled image "k8s.gcr.io/liveness" in 254.231974ms
         Pulled
                                        kubelet
 Normal
                    109s
         Created
                    91s (x3 over 2m7s)
                                        kubelet
                                                           Created container liveness
 Normal
 Normal
         Started
                    91s (x3 over 2m7s)
                                        kubelet
                                                           Started container liveness
 Normal
         Pulled
                                         kubelet
                                                           Successfully pulled image "k8s.gcr.io/liveness" in 259.054744ms
                                                           Pulling image "k8s.gcr.io/liveness"
 Normal
         Pulling
                    73s (x4 over 2m7s)
                                        kubelet
                                                           Liveness probe failed: HTTP probe failed with statuscode: 500
 Warning Unhealthy 73s (x9 over 115s)
                                        kubelet
         Killing
                    73s (x3 over 109s) kubelet
                                                           Container liveness failed liveness probe, will be restarted
 Normal
```

ToDo:

fix this

probes_tcp

```
apiVersion: v1
kind: Pod
metadata:
  name: liveness-tcp
labels:
  app: goproxy
spec:
  containers:
  - name: goproxy
    image: k8s.gcr.io/goproxy:0.1
    ports:
    - containerPort: 8080
    livenessProbe:
      tcpSocket:
        port: 9999 # 8080 is a valid port
      initialDelaySeconds: 15
      periodSeconds: 20
```

liveness-tcp is up and running

```
@ / ➤ ~/Homework/Homework-13/manifets / ➡ 🎖 master wip !2 ?4 
     get pods -A
NAMESPACE
             NAME
                                                                               READY
                                                                                       STATUS
                                                                                                          RESTARTS
                                                                                                                        AGE
default
              liveness-http
                                                                               0/1
                                                                                       CrashLoopBackOff
                                                                                                          7 (48s ago)
                                                                                                                        7m52s
default
              liveness-tcp
                                                                               1/1
                                                                                       Running
                                                                                                                        29s
```

Probe failure due to using port 9999

	node, kuber netes, 10/ um eachab te, Nobaecute up-Exists for 3005					
Events:						
Туре	Reason	Age	From	Message		
Normal	Scheduled	37 s	default-scheduler	Successfully assigned default/liveness-tcp to aks-agentpool-41905252-vmss000009		
Normal	Pulled	37 s	kubelet	Container image "k8s.gcr.io/goproxy:0.1" already present on machine		
Normal	Created	37s	kubelet	Created container goproxy		
Normal	Started	37 s	kubelet	Started container goproxy		
Warning	Unhealthy	17s	kubelet	Liveness probe failed: dial tcp 10.224.0.10:9999: connect: connection refused		

• readiness_htpp

```
apiVersion: v1
kind: Pod
metadata:
  name: readiness-http
  labels:
    app: test
spec:
  containers:
  - name: nginx
    image: nginx
    ports:
    - containerPort: 80
    readinessProbe:
      httpGet:
        path: /
        port: 80
        httpHeaders:
        - name: Host
          value: myapplication1.com
      initialDelaySeconds: 1
      periodSeconds: 2
      timeoutSeconds: 1
      successThreshold: 1
      failureThreshold:
```

Pod is up and running

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
default	liveness-http	0/1	CrashLoopBackOff	9 (2m30s ago)	15m
default	liveness-tcp	1/1	Running	6 (115s ago)	7m56s
default	readiness-http	1/1	Running	0	4s

No issues when running the describe command.

Events:				
Type	Reason	Age	From	Message
Normal	Scheduled	91s	default-scheduler	Successfully assigned default/readiness-http to aks-agentpool-41905252-vmss000009
Normal	Pulling	91s	kubelet	Pulling image "nginx"
Normal	Pulled	91s	kubelet	Successfully pulled image "nginx" in 244.77341ms
Normal	Created	90s	kubelet	Created container nginx
Normal	Started	90s	kubelet	Started container nginx

The pod is running but it's not ready

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
default	liveness-http	0/1	CrashLoopBackOff	9 (39s ago)	13m
default	liveness-tcp	0/1	CrashLoopBackOff	5 (4s ago)	6m5s
default	readiness-http	0/1	Running	Θ	32s

Due to the change the probe fails.

Events:				
Type	Reason	Age	From	Message
Normal	Scheduled	72s	default-scheduler	Successfully assigned default/readiness-http to aks-agentpool-41905252-vmss000009
Normal	Pulling	72s	kubelet	Pulling image "nginx"
Normal	Pulled	72s	kubelet	Successfully pulled image "nginx" in 147.784144ms
Normal	Created	72s	kubelet	Created container nginx
Normal	Started	72s	kubelet	Started container nginx
Warning	Unhealthy	34s (x21 over 71s)	kubelet	Readiness probe failed: Get "http://10.224.0.23:81/": dial tcp 10.224.0.23:81: connect: connection refused