



Multi-Container Pattern Side car

[Edition 2]

[Last Update 211323]

For any issues/help contact: support@k21academy.com





Contents

I Introduction		3
2 Documentation		4
2.1 Kubernetes Documentation		
Multi-Container Pattern – Sidecar Container	<u> </u>	5
3.1 Creating Multi-Container Pod with Shared Volume		
3.2 Clean-up resources created in this lab exercise		
- 4 Summary		





1 INTRODUCTION

There are three common design patterns and use-cases for combining multiple containers into a single pod. We'll walk through the **sidecar pattern**, the **adapter pattern**, and the **ambassador pattern**. Look to the end of the post for example YAML files for each of these.

The sidecar pattern consists of a main application—i.e. your web application—plus a helper container with a responsibility that is essential to your application, but is not necessarily part of the application itself.

The ambassador pattern is a useful way to connect containers with the outside world. An ambassador container is essentially a proxy that allows other containers to connect to a port on localhost while the ambassador container can proxy these connections to different environments depending on the cluster's needs.

This guide Covers:

Multi-Container Pattern - Sidecar Container

• Creating Multi-Container Pod with Shared Volume







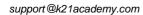
2 DOCUMENTATION

2.1 Kubernetes Documentation

1 Communicate Between Containers

https://kubernetes.io/docs/tasks/access-application-cluster/communicate-containers-same-pod-shared-volume/

- 2 Patterns for Composite Containers https://kubernetes.io/blog/2015/06/the-distributed-system-toolkit-patterns/
- 3 Pods https://kubernetes.io/docs/concepts/workloads/pods/







3 MULTI-CONTAINER PATTERN – SIDECAR CONTAINER

Note: In below Sections we are going to use YAML files no need write complete yaml file because in CKA exam you can official Kubernetes documentation use Below GIT url to clone repo and use yaml files

\$ git clone https://github.com/k21academyuk/Kubernetes

\$ cd Kubernetes

root@master:~# ls Kubernetes root@master:~# cd Kubernetes/ root@master:~/Kubernetes# ls Dockerfile RFADMF.md pycache adapter-configmap.yaml adapter-pod.yaml app.py apple.yaml banana.yaml config-map.yaml configmap-pod.yaml counter-pod.yaml cron.yami daemonset.yaml demo-pod.yaml docker-compose.yaml docker-registry-secret.yaml dockerfile-mg elasticsearch-rbac.yaml elasticsearch-stfullset-oci.yaml elasticsearch-stfullset.yaml elasticsearch-svc.yaml elasticsearch.yaml example-ingress.yaml filebeat-agent.yaml fluentd.yaml root@master:~/Kubernetes#

foo-allow-to-hello.yaml guestbook-frontend-svc.yaml questbook-frontend.vaml headlessservice.yaml hello-allow-from-foo.yaml ingress-app1.yaml ingress-app2.yaml ingress-route.yaml initcontainer.yaml job-mq.yaml job-tmpl.yaml job.yaml kibana-elk.yaml kibana.yaml label-deployment.yaml liveness-pod.yaml logstash-configmap.yaml logstash-deployment.yaml logstash-svc.yaml eťrics-server yami multi-container.yaml multi-pod-configmap.y multi-pod-nginx.yaml multi-prod-consumer.yaml

network-policy.yaml nfs-pv.yaml nfs-pvc.yaml nfspv-pod.yaml nginx-deployment.yaml nginx-hpa.yaml nginx-svc.yaml nodeaffinity-deployment.yaml nodeaffinity1-deployment.yaml nodeanti-affinity-deployment.yaml nodeanti-affinity1-deployment.yaml oke-admin-service-account.yaml pod-dynamicpv-oci.yaml pod-dynamicpv.yaml podaffinity-deployment.yaml podaffinity1-deployment.yaml podanti-affinity-deployment.yaml podanti-affinity1-deployment.yaml priv-reg-pod.yaml pvc-oci.yaml pvc.yaml quota-pod.yaml quota-pod1.yaml quota.yaml rabbitmq-deployment.yaml

rabbitmq-service.yaml readiness-pod.yaml readiness-svc.yaml redis-cm.yaml redis-master-svc.vaml redis-master.yaml
redis-pod.yaml redis-slave-svc.yaml redis-slave.yaml requirements.txt role-dev.yaml rolebind.yaml script.sh security-cxt-nonroot.yaml security-cxt-priv.yaml security-cxt-readonly.yaml security-cxt-rmcap.yamí security-cxt-time.yaml security-cxt.yaml statefulset1.yaml tt-pod.vaml tt-pod1.yaml web.yaml worker.py

3.1 Creating Multi-Container Pod with Shared Volume

- 1. Create a multi container pod with sidecar container pattern
- 2. Viewing the contents of multi-container yaml file which has two container definition

\$ vim multi-container.yaml





```
apiVersion: v1
kind: Pod
metadata:
  name: mc1
spec:
  volumes:
  - name: html
    emptyDir: {}
  containers:
   - name: 1st
    image: nginx
    volumeMounts:
    - name: html
      mountPath: /usr/share/nginx/html
  - name: 2nd
    image: debian
    volumeMounts:
    - name: html
      mountPath: /html
    command: ["/bin/sh", "-c"]
    args:
      - while true; do
          date >> /html/index.html;
          sleep 1;
        done
```

3. Deploying both the containers from multi-container.yaml file

\$ kubectl create -f multi-container.yaml

```
root@kubeadm-master-01:~/Kubernetes#
root@kubeadm-master-01:~/Kubernetes#
root@kubeadm-master-01:~/Kubernetes# kubectl create -f multi-container.yaml
pod/mc1 created
root@kubeadm-master-01:~/Kubernetes#
```

4. Verify the pod status

```
$ kubectl get pods - w
```

\$ Kubectl describe pod mc1

```
root@kubeadm-master-01:~/Kubernetes# kubectl get pods -w
NAME READY STATUS RESTARTS AGE
mc1 2/2 Running 0 23s
^Z
[1]+ Stopped kubectl get pods -w
```





```
[root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl describe pod mc1
Namespace:
              default
Priority:
             0
             worker1/10.0.0.5
Node:
Start Time: Fri, 02 Oct 2020 17:16:14 +0000
Labels:
              <none>
Annotations:
             <none>
Status:
             Running
TP:
             10.36.0.3
TPs:
 IP: 10.36.0.3
Containers:
 1st:
   Container ID: docker://92e66344e0ba3f325870720a1bbb9772187d7e91e716526a6caabcb254487582
   Image:
                  nginx
   Image ID:
                   docker-pullable://nginx@sha256:c628b67d21744fce822d22fdcc0389f6bd763daac23a6b77147d0712ea7102d0
   Port:
                   <none>
   Host Port:
                   <none>
   State:
                   Running
                   Fri, 02 Oct 2020 17:16:16 +0000
     Started:
   Ready:
                    True
   Restart Count: 0
    Environment:
                   <none>
   Mounts:
      /usr/share/nginx/html from html (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-fq86n (ro)
   Container ID: docker://f8304750c1e1ce96cda17094ae430009298df4b5848cc76cd4c0d687da8a3af2
                  debian
   Image:
                  docker-pullable://debian@sha256:439a6bae1ef351ba9308fc9a5e69ff7754c14516f6be8ca26975fb564cb7fb76
   Image ID:
   Port:
                  <none>
   Host Port:
   Command:
     /bin/sh
      -0
   Args:
     while true; do date >> /html/index.html; sleep 1; done
    State:
                   Running
                   Fri, 02 Oct 2020 17:16:26 +0000
     Started:
   Ready:
                   True
   Restart Count: 0
    Environment:
      /html from html (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-fq86n (ro)
Conditions:
```

5. Checking the shared directory directly in the containers

\$ kubectl exec mc1 -c 1st -- /bin/cat /usr/share/nginx/html/index.html

```
root@kubeadm-master-01:~/Kubernetes#
root@kubeadm-master-01:~/Kubernetes# kubectl exec mc1 -c 1st -- /bin/cat /usr/share/nginx/html/index.html
Tue Jun 16 17:12:08 UTC 2020
Tue Jun 16 17:12:09 UTC 2020
Tue Jun 16 17:12:10 UTC 2020
Tue Jun 16 17:12:11 UTC 2020
Tue Jun 16 17:12:12 UTC 2020
Tue Jun 16 17:12:13 UTC 2020
Tue Jun 16 17:12:14 UTC 2020
Tue Jun 16 17:12:15 UTC 2020
Tue Jun 16 17:12:16 UTC 2020
Tue Jun 16 17:12:17 UTC 2020
Tue Jun 16 17:12:18 UTC 2020
Tue Jun 16 17:12:19 UTC 2020
Tue Jun 16 17:12:20 UTC 2020
Tue Jun 16 17:12:21 UTC 2020
Tue Jun 16 17:12:22 UTC 2020
```

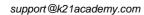




6. We can see that the sidecar container populates the index.html page for the main container serving as web server

3.2 Clean-up resources created in this lab exercise

\$ kubectl delete -f multi-container.yaml







4 SUMMARY

In this guide we Covered:

1. Multi-Container Pattern – Sidecar Container

