

```
Editor  Tab 1  +
controlplane $ vim redis.yaml
controlplane $ cat redis.yaml
apiVersion: v1
kind: Pod
metadata:
  name: redis
  labels:
    app: redis
spec:
  containers:
  - name: redis
    image: redis
  initContainers:
  - name: init-myservice
    image: busybox:1.28
    command: ['sh', '-c', 'sleep20']
controlplane $ k apply -f redis.yaml
pod/redis created
controlplane $ k get po
NAME      READY   STATUS              RESTARTS   AGE
redis     0/1     Init:CrashLoopBackOff 1 (3s ago)  6s
controlplane $
```

```
apiVersion: v1
kind: Pod
metadata:
  name: print-envvars-greeting
  labels:
    app: greeting
spec:
  containers:
    - name: print-env-container
      image: bash
      env:
        - name: GREETING
          value: "welcome to "
        - name: COMPANY
          value: "DevOps"
        - name: GROUP
          value: "industries"
      command: ['sh', '-c', echo "$GREETING $COMPANY $GROUP"]
```

~
~
~
~

```
apiVersion: v1
kind: Pod
metadata:
  name: print-envvars-greeting
  labels:
    name: print-envvars-greeting
spec:
  containers:
    - name: print-env-container
      image: bash
      env:
        - name: GREETING
          value: "Welcome to"
        - name: COMPANY
          value: "Devops"
        - name: GROUP
          value: "Industries"
      command: ["echo"]
      args: ["$(GREETING) $(COMPANY) $(GROUP)"]
```

~
~

```

NAME          READY    STATUS              RESTARTS   AGE
redis         0/1     Init:CrashLoopBackOff 1 (2s ago)  4s
controlplane $ vim greet.yaml
controlplane $ k apply -f greet.yaml
pod/print-envvars-greeting created
controlplane $ k get po
NAME          READY    STATUS              RESTARTS   AGE
print-envvars-greeting 0/1     Completed           2 (16s ago) 29s
redis         0/1     Init:CrashLoopBackOff 4 (17s ago)  2m
controlplane $ kubectl logs print-envvars-greeting
Welcome to Devops Industries
controlplane $

```

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pv-log
  labels:
    app: pv-log
spec:
  accessModes:
    - ReadWriteMany
  capacity:
    storage: 100Mi
  hostPath:
    path: "/pv/log"
  claimRef:
    name: claim-log-1

```

```

~
~
~
~
~
~
~

```

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: claim-log-1
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 50Mi
  selector:
    matchLabels:
      app: pv-log
```

```
~
~
~
~
~
~
~
~
```

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pv-log
spec:
  accessModes:
    - ReadWriteMany
  capacity:
    storage: 100Mi
  hostPath:
    path: /pv/log
    type: Directory
```

```
~
~
```

```

controlplane $ vim webapp.yaml
controlplane $ cat webapp.yaml
apiVersion: v1
kind: Pod
metadata:
  name: webapp
  labels:
    app: nginx
spec:
  containers:
  - name: webapp-pod
    image: nginx
    volumeMounts:
    - name: vol
      mountPath: /var/log/nginx
  volumes:
  - name: vol
    persistentVolumeClaim:
      claimName: claim-log-1
controlplane $ k apply -f webapp.yaml
pod/webapp created
controlplane $

```

```

pod/webapp created
controlplane $ k get DaemonSets --all-namespaces

```

NAMESPACE	NAME	DESIRED	CURRENT	READY	UP-TO-DATE	AVAILABLE	NODE SELECTOR	AGE
kube-system	canal	1	1	1	1	1	kubernetes.io/os=linux	15d
kube-system	kube-proxy	1	1	1	1	1	kubernetes.io/os=linux	15d

```

controlplane $ k get DaemonSets -n kube-system

```

NAME	DESIRED	CURRENT	READY	UP-TO-DATE	AVAILABLE	NODE SELECTOR	AGE
canal	1	1	1	1	1	kubernetes.io/os=linux	15d
kube-proxy	1	1	1	1	1	kubernetes.io/os=linux	15d

```

controlplane $

```

```

controlplane $
controlplane $ kubectl describe pods kube-proxy --namespace=kube-system
Name: kube-proxy-pml4f
Namespace: kube-system
Priority: 2000001000
Priority Class Name: system-node-critical
Service Account: kube-proxy
Node: controlplane/172.30.1.2
Start Time: Thu, 26 Jan 2023 14:25:02 +0000
Labels: controller-revision-hash=78545cdb7d
        k8s-app=kube-proxy
        pod-template-generation=1
Annotations: <none>
Status: Running
IP: 172.30.1.2
IPs:
  IP: 172.30.1.2
Controlled By: DaemonSet/kube-proxy
Containers:
  kube-proxy:
    Container ID: containerd://13b8599416e8e6cbd2e654dad81fa52351462482f345b7f1158dfa288074375e
    Image: registry.k8s.io/kube-proxy:v1.26.0
    Image ID: registry.k8s.io/kube-proxy@sha256:1e9bbe429e4e2b2ad32681c91deb98a334f1bf4135137df5f84f9d03689060fe
    Port: <none>
    Host Port: <none>
    Command:
      /usr/local/bin/kube-proxy
      --config=/var/lib/kube-proxy/config.conf
      --hostname-override=$(NODE_NAME)
    State: Running

```

```

apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: elasticsearch
  namespace: kube-system
  labels:
    k8s-app: fluentd-logging
spec:
  selector:
    matchLabels:
      name: fluentd-elasticsearch
  template:
    metadata:
      labels:
        name: fluentd-elasticsearch
    spec:
      containers:
        - name: fluentd-elasticsearch
          image: k8s.gcr.io/fluentd-elasticsearch:1.20
~
~
~
~
~
~

```

```

controlplane $ vim daemon.yaml
controlplane $ k apply -f daemon.yaml
daemonset.apps/elasticsearch created
controlplane $ k get DaemonSet -n kube system
error: the server doesn't have a resource type "DaemonSet"
controlplane $ k get DaemonSets -n kube system
error: the server doesn't have a resource type "DaemonSets"
controlplane $ kubectl get DaemonSet -n kube-system

```

NAME	DESIRED	CURRENT	READY	UP-TO-DATE	AVAILABLE	NODE SELECTOR	AGE
canal	1	1	1	1	1	kubernetes.io/os=linux	15d
elasticsearch	1	1	1	1	1	<none>	116s
kube-proxy	1	1	1	1	1	kubernetes.io/os=linux	15d

```

controlplane $

```

```

apiVersion: v1
kind: Pod
metadata:
  name: yellow
spec:
  containers:
    - name: lemon
      image: busybox
      tty: true
    - name: gold
      image: redis
~
~
~

```

```
controlplane $ k apply -f ypod.yaml
pod/yellow created
controlplane $ k get po
NAME                                READY   STATUS              RESTARTS   AGE
print-envvars-greeting             0/1     CrashLoopBackOff    10 (3m42s ago)  30m
redis                              0/1     Init:CrashLoopBackOff 10 (4m51s ago)  31m
webapp                             0/1     Pending              0           10m
yellow                             0/2     ContainerCreating    0           6s
controlplane $ k get po
NAME                                READY   STATUS              RESTARTS   AGE
print-envvars-greeting             0/1     CrashLoopBackOff    10 (3m55s ago)  30m
redis                              0/1     Init:Error           11 (5m4s ago)   31m
webapp                             0/1     Pending              0           10m
yellow                             2/2     Running              0           19s
controlplane $
```

```
apiVersion: v1
kind: Pod
metadata:
  name: db-pod
spec:
  containers:
  - name: db-container
    image: mysql:5.7
```

```
~
~
~
~
~
yellow                             2/2     Running              0           19s
controlplane $ vim dppod.yaml
controlplane $ k apply -f dppod.yaml
pod/db-pod created
controlplane $ k get po
NAME                                READY   STATUS              RESTARTS   AGE
db-pod                             0/1     CrashLoopBackOff    2 (27s ago)    55s
print-envvars-greeting             0/1     CrashLoopBackOff    11 (115s ago)  33m
redis                              0/1     Init:CrashLoopBackOff 11 (3m13s ago)  34m
webapp                             0/1     Pending              0           13m
yellow                             2/2     Running              0           3m29s
controlplane $
```

```

Editor  Tab | +
apiVersion: v1
kind: Secret
metadata:
  name: db-secret
data:
  MYSQL_DATABASE: c3FsMDEg
  MYSQL_USER: dXNlcjEg
  MYSQL_PASSWORD: cGFzc3dvcmQgIA==
  MYSQL_ROOT_PASSWORD: cGFzc3dvcmQxMjMgIA==
~
~
~
~
~

apiVersion: v1
kind: Pod
metadata:
  name: db-pod
spec:
  containers:
  - name: db-container
    image: mysql:5.7
    envFrom:
    - secretRef:
        name: db-secret
~
~
~
~
~
~

controlplane $ k get po
NAME      READY   STATUS    RESTARTS
db-pod    1/1     Running   1 (5s ago)

```

12- Why the db-pod status not ready ?

we didn't add database env variables..