1-

Editor Tab 1 +

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: redis
  labels:
    app: MyApp
spec:
  containers:
  - name: redis
    image: redis
  initContainers:
  - name: init-cont
    image: busybox
    command: ['sleep' ,'20']
```

```
controlplane $ vim redis-busybox.yml
controlplane $ kubectl create -f redis-busybox.yml
pod/redis created
```

2-

Editor Tab 1 +

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

```
controlplane $ kubectl create -f greeting.yml
pod/print-envars-greeting created
controlplane $ kubectl logs -f
daemonsets/                   jobs/
deployments/                  pods/
controlplane $ kubectl logs -f print-envars-gr
Welcome to DevOps Industries
```

3-

Editor Tab 1 +

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-log
  labels:
    type: local
spec:
  storageClassName: manual
  capacity:
    storage: 100Mi
  accessModes:
    - ReadWriteMany
  hostPath:
    path: "/pv/log"
```

```
controlplane $ kubectl create -f volume.yml
persistentvolume/pv-log created
```

4-

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

```
controlplane $ kubectl create -f volume-claim.yml
persistentvolumeclaim/claim-log-1 created
```

5-

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: webapp
spec:
  containers:
    - name: myfrontend
      image: nginx
      volumeMounts:
      - mountPath: "/var/log/nginx"
        name: pvc
  volumes:
    - name: pvc
      persistentVolumeClaim:
        claimName: claim-log-1
```

```
controlplane $ kubectl create -f webapp.yml
pod/webapp created
```

6-

```
controlplane $ kubectl get daemonsets.apps --all-namespaces
NAMESPACE     NAME         DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
kube-system   canal        2         2         2       2            2           kubernetes.io/os=linux   11d
kube-system   kube-proxy   2         2         2       2            2           kubernetes.io/os=linux   11d
```

7-canal and kube-proxy

8-registry.k8s.io/kube-proxy:v1.26.0

```
CONTAINERS                IMAGES

calico-node,kube-flannel  docker.io/calico/node:v3.24.1,quay.io/c

kube-proxy                registry.k8s.io/kube-proxy:v1.26.0
```

9-

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

10-

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: yellow
spec:
  containers:
  - name: lemon
    image: busybox
  - name: gold
    image: redis
```

11 & 12-

```
  Normal   Pulled     6s (x3 over 44s)  kubelet            Container image "mysql:5.7" already present on machine
  Warning  BackOff    5s (x5 over 43s)  kubelet            Back-off restarting failed container db-pod in pod db-pod_default(e04c13c8-0f5c-4bb9-8262-09c5c0af4fea)
controlplane $ kubectl get pod db-pod
NAME     READY   STATUS             RESTARTS      AGE
db-pod   0/1     CrashLoopBackOff   3 (24s ago)   74s
controlplane $
```

13-

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: mysecret
type: Opaque
data:
  MYSQL_DATABASE: c3FsMDE=
  MYSQL_USER: dXNlcjE=
  MYSQL_PASSWORD: cGFzc3dvcmQ=
  MYSQL_ROOT_PASSWORD: cGFzc3dvcmQxMjM=
```

14-

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: db-pod
spec:
  containers:
  - name: db
    image: mysql:5.7
    env:
    - name: SECRETS
      valueFrom:
        secretKeyRef:
          name: db-secret
          key: MYSQL_DATABASE
```