

Application Deployment in Kubernetes

Example: deploying a standalone Minio instance with persistent storage and service

minio-deployment.yaml

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  # This name uniquely identifies the Deployment
  name: minio
spec:
  strategy:
    # Strategy used to replace old Pods by new ones
    type: Recreate
  template:
    metadata:
      labels:
        # Label used as selector in Service definition
        app: minio
    spec:
      # Volumes used by this deployment
      volumes:
        - name: data
          # This volume is based on PVC
          persistentVolumeClaim:
            # Name of the PVC created earlier
            claimName: minio-pv-claim
      containers:
        - name: minio
          # Volume mounts for this container
          volumeMounts:
            # Volume 'data' is mounted to path '/data'
            - name: data
              mountPath: "/data"
          # Pulls the Minio image from Docker Hub
          image: minio/minio
          args:
            - server
            - /data
          env:
            # MinIO access key and secret key
            - name: MINIO_ACCESS_KEY
              value: "minio"
            - name: MINIO_SECRET_KEY
              value: "minio123"
          ports:
            - containerPort: 9000
            # Readiness probe detects when MinIO server instance
            # is not ready to accept traffic.
            readinessProbe:
              httpGet:
                path: /minio/health/ready
                port: 9000
                initialDelaySeconds: 120
                periodSeconds: 20
            # Liveness probe detects when MinIO server instance
            # is not working properly and needs restart.
            livenessProbe:
              httpGet:
                path: /minio/health/live
                port: 9000
                initialDelaySeconds: 120
                periodSeconds: 20
```

Commands

- Applying a YAML file:
kubectl apply -f yaml-file.yml
- Getting info of a resource:
kubectl get|describe resource|all
-o wide
- Executing a shell in a container:
kubectl exec -ti <pod> -- /bin/bash
- Getting logs of a pod:
kubectl logs <pod>
- Deleting a resource:
kubectl delete <kind> <name>

minio-pvc.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  # A unique name for the the PVC
  name: minio-pv-claim
spec:
  storageClassName: manual
  accessModes:
    # Read-write on a single node
    - ReadWriteOnce
  resources:
    # The request for storage
    requests:
      storage: 20Gi
```

Tools

- **Kompose**: automatic conversion of Docker Compose files into Kubernetes YAML deployments
- **DevSpace CLI**: deployment of apps in Kubernetes Cluster
- **MS Draft**: development & deployment of apps

minio-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  # A uniquely name for the service
  name: minio-service
spec:
  type: LoadBalancer
  ports:
    - port: 9000
      targetPort: 9000
      protocol: TCP
  selector:
    # Looks for labels app:minio
    app: minio
```

Source of the Minio Standalone Server Deployment: <https://docs.min.io/docs/deploy-minio-on-kubernetes.html>
Home of this poster file and much more: <https://gitlab.mpcdf.mpg.de/thomz/serverless>