

Steps to replicate our mongo db stateful workflow

- 1) After creating our yaml files, we then applied each of them

```
base ~/Documents/GitHub/Exam-FullStack/q2 (1m 7.86s)
minikube start
```

```
base ~/Documents/GitHub/Exam-FullStack/q2 (0.706s)
minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

```
base ~/Documents/GitHub/Exam-FullStack/q2 * minikube (0.185s)
kubectl apply -f pv.yaml
persistentvolume/mongodb-pv created
```

```
base ~/Documents/GitHub/Exam-FullStack/q2 * minikube (0.266s)
kubectl apply -f pvc.yaml
persistentvolumeclaim/mongodb-pvc created
```

```
base ~/Documents/GitHub/Exam-FullStack/q2 * minikube (0.115s)
kubectl apply -f deploy.yaml
deployment.apps/mongodb created
```

```
base ~/Documents/GitHub/Exam-FullStack/q2 * minikube (0.078s)
kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
mongodb-86d986b75d-tdfw	0/1	ContainerCreating	0	12s

```
base ~/Documents/GitHub/Exam-FullStack/q2 * minikube (0.097s)
kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
mongodb-pvc	Bound	mongodb-pv	1Gi	RWO	manual	27s

```
base ~/Documents/GitHub/Exam-FullStack/q2 * minikube (0.083s)
kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
mongodb-pvc	Bound	mongodb-pv	1Gi	RWO	manual	2m15s

- 2) Afterwards, notice how our pvc status is set to bound. We can further clarify that our pod changed to a running state and our DB had mounted on the pod

```
base ~/Documents/GitHub/Exam-FullStack/q2 * minikube (0.167s)
kubectrl exec mongodb-86d986b75d-tdfwm -- df -h /data/db
```

Filesystem	Size	Used	Avail	Use%	Mounted on
overlay	59G	9.5G	46G	18%	/data/db

```
base ~/Documents/GitHub/Exam-FullStack/q2 * minikube (0.079s)
```

```
kubectrl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
<code>mongodb-86d986b75d-tdfwm</code>	1/1	Running	0	5m19s

```
base ~/Documents/GitHub/Exam-FullStack/q2 * minikube (0.108s)
```

```
kubectrl describe pod mongodb-86d986b75d-tdfwm
```

```
Name:          mongodb-86d986b75d-tdfwm
Namespace:     default
Priority:       0
Service Account: default
Node:          minikube/192.168.49.2
Start Time:    Fri, 22 Mar 2024 19:09:04 -0400
Labels:        app=mongodb
               pod-template-hash=86d986b75d
Annotations:   <none>
Status:        Running
IP:            10.244.0.3
IPs:           IP: 10.244.0.3
Controlled By: ReplicaSet/mongodb-86d986b75d
Containers:
  mongodb:
    Container ID:  docker://c3b4cfc186a569aa95371f195a9777b12799914de5508aac4ea014f191cf2c2d
    Image:         mongo
    Image ID:      docker-pullable://mongo@sha256:0e145625e78b94224d16222ff2609c4621ff6e2c390300e4e6bf6
98305596792
    Port:         27017/TCP
    Host Port:    0/TCP
    State:        Running
      Started:    Fri, 22 Mar 2024 19:09:42 -0400
    Ready:        True
    Restart Count: 0
    Environment:  <none>
    Mounts:
      /data/db from mongodb-persistent-storage (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-sxb9w (ro)
Conditions:
  Type            Status
  Initialized      True
  Ready           True
  ContainersReady  True
  PodScheduled     True
Volumes:
```

```

Volumes:
  mongodb-persistent-storage:
    Type: PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName: mongodb-pvc
    ReadOnly: false
  kube-api-access-sxb9w:
    Type: Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName: kube-root-ca.crt
    ConfigMapOptional: <nil>
    DownwardAPI: true
QoS Class: BestEffort
Node-Selectors: <none>
Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
              node.kubernetes.io/unreachable:NoExecute op=Exists for 300s

Events:
  Type        Reason      Age   From          Message
  ----        -
  Normal      Scheduled   5m31s default-scheduler Successfully assigned default/mongodb-86d986b75d-tdfwm to minikube
  Normal      Pulling     5m30s kubelet        Pulling image "mongo"
  Normal      Pulled      4m54s kubelet        Successfully pulled image "mongo" in 35.728s (35.728s including waiting)
  Normal      Created     4m53s kubelet        Created container mongodb
  Normal      Started     4m53s kubelet        Started container mongodb

```