How does a StatefulSet differ from a ReplicaSet? Give Example where they are used.

ReplicaSet = for **stateless** apps (fast, interchangeable pods).

StatefulSet = for **stateful** apps needing persistent data & stable identity.

Scenarios Where ReplicaSet is Used:

1. **Web Servers**
    - E.g., Nginx, Apache
    - Serve static content or act as reverse proxies
    - No need for data persistence between restarts
2. **Frontend Applications**
    - React, Angular, or Vue apps served via containerized environments
    - Pods are interchangeable
3. **API Servers / Microservices**
    - Stateless services (Node.js, Spring Boot, Flask APIs)
    - Can be scaled horizontally without issues
4. **Load-balanced Services**
    - ReplicaSet ensures enough backend pods are available for load balancing

5. **Worker Pods**
    - Jobs like sending emails, notifications, or serving requests where each instance behaves the same

Scenarios Where StatefulSet is Used:

1. **Databases**
    - MySQL, PostgreSQL, MongoDB, Cassandra, etc.
    - Each pod needs its **own volume** to store data.
    - Example: `mysql-0`, `mysql-1`, `mysql-2`
2. **Message Queues**
    - Like **Kafka** or **RabbitMQ**, which need persistent data and pod identity for broker IDs.
3. **Distributed File Systems**
    - e.g., GlusterFS, Ceph — where nodes rely on specific identities and storage.
4. **Leader-based systems**
    - Systems like Zookeeper or etcd require ordered startup and identity for leader election.

TEST

kubectl port-forward pod/nginx-0 8080:80
curl http://localhost:8080

Demo

We are setting up a **StatefulSet of 3 NGINX web servers**. Each server (pod) will:

1. Have its **own identity**: `nginx-0`, `nginx-1`, `nginx-2`
2. Have its **own disk**: like its own hard drive
3. Show a **custom web page** that says:

```
<h1>This is pod nginx-0</h1>
```

So each one tells you who it is.

Even if the pod crashes and restarts, it **remembers** who it is and what page it showed — because its disk (PVC) is attached permanently.

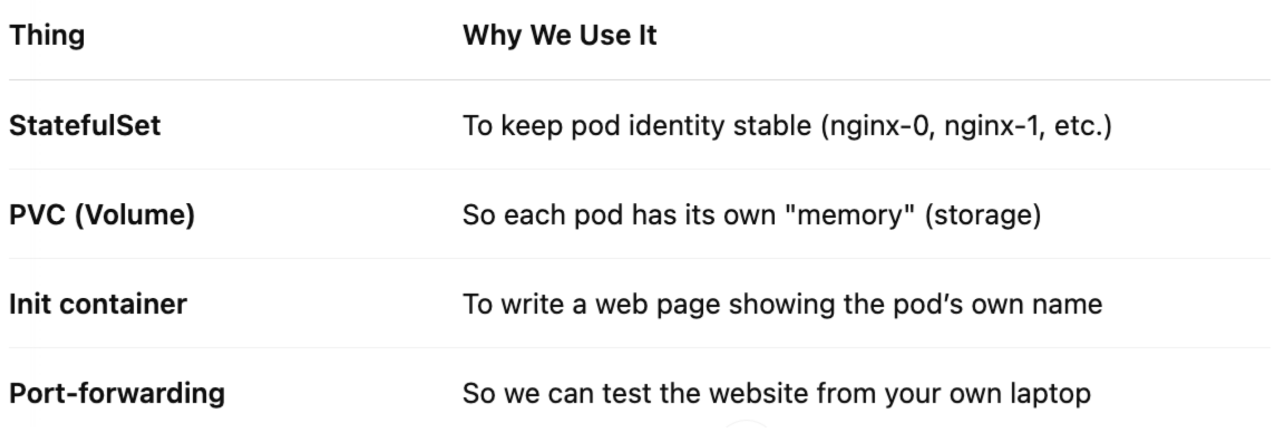
Why Not Use ReplicaSet?

Because if you used a ReplicaSet:

- All pods would be named randomly
- They wouldn't keep their own memory
- You wouldn't know which pod is which — they're all clones!

kubectl get pvc

Check the Pods and their Order: **kubectl get pods -l app=nginx**



Thing	Why We Use It
StatefulSet	To keep pod identity stable (nginx-0, nginx-1, etc.)
PVC (Volume)	So each pod has its own "memory" (storage)
Init container	To write a web page showing the pod's own name
Port-forwarding	So we can test the website from your own laptop

A

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: nginx
spec:
  serviceName: "nginx"
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      initContainers:
        - name: init-html
          image: busybox
          command:
            - sh
            - -c
        - >
          echo "<h1>This is pod $(hostname)</h1>" > /usr/share/nginx/html/index.html
        volumeMounts:
        - name: www
          mountPath: /usr/share/nginx/html
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 80
          name: web
        volumeMounts:
        - name: www
          mountPath: /usr/share/nginx/html
  volumeClaimTemplates:
  - metadata:
      name: www
    spec:
      accessModes: ["ReadWriteOnce"]
      resources:
        requests:
          storage: 1Gi
      storageClassName: standard
```

Q: What is a key difference between a StatefulSet and a ReplicaSet?

A. StatefulSet gives each pod a unique identity

B. ReplicaSet supports stable storage

C. StatefulSet is used for stateless apps

D. ReplicaSet guarantees pod order

Answer: A