```
======================================
Kubernetes ReplicaSet + Service (2025 style)
======================================

# NOTE: ReplicaSets are almost never created manually.
# They are automatically created by Deployments.
# But here is how to do it directly if you really need to.

# 1. Create a ReplicaSet directly (3 nginx replicas)

kubectl create replicaset nginx-rs \
  --image=nginx:alpine \
  --replicas=3
```

```
# Verify
kubectl get rs
kubectl get pods -l app=nginx

# 2. Expose the ReplicaSet as a Service

# Internal only
kubectl expose rs nginx-rs --port=80 --name=nginx-rs-svc

# External via NodePort
kubectl expose rs nginx-rs --port=80 --type=NodePort --name=nginx-rs-svc

# External via LoadBalancer (cloud only)
kubectl expose rs nginx-rs --port=80 --type=LoadBalancer --name=nginx-rs-lb
```

```
# Check
kubectl get svc nginx-rs-svc
========================================
BEST PRACTICE: Use proper YAML files
========================================
# File: replicaset.yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: nginx-rs
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:alpine
        ports:
        - containerPort: 80

# File: service.yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-rs-service
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: NodePort          # Change to ClusterIP or LoadBalancer as needed

# Apply both
kubectl apply -f replicaset.yaml
kubectl apply -f service.yaml

# Scale anytime
kubectl scale replicaset nginx-rs --replicas=10

=========================================
# QUICK ONE-LINERS (for labs / testing)
=========================================
# Create ReplicaSet + expose in 2 commands
kubectl create rs nginx-rs --image=nginx:alpine --replicas=4
kubectl expose rs nginx-rs --type=NodePort --port=80 --name=nginx-rs-svc

# See the assigned port
kubectl get svc nginx-rs-svc
# Example output:
# NAME           TYPE       CLUSTER-IP     PORT(S)        AGE
# nginx-rs-svc   NodePort   10.96.123.45   80:32001/TCP   10s

# Access from browser/machine:
# http://<any-node-ip>:32001
```

```
=====================================
 CLEANUP
=====================================

kubectl delete rs nginx-rs
kubectl delete svc nginx-rs-service
# or
kubectl delete -f replicaset.yaml -f service.yaml

=====================================
# REAL-WORLD ADVICE (2025)
=====================================
# Never use ReplicaSet directly in production!
# Always use Deployment instead (it creates and manages the ReplicaSet for you):
```

```
kubectl create deployment nginx --image=nginx:alpine --replicas=3
kubectl expose deployment nginx --type=NodePort --port=80

# This is the modern, correct way.
```

Done! You now have a fully working ReplicaSet exposed via a Service.