

container-d

```
3  kubectl run nginx
4  kubectl run nginx --image=nginx
5  kubectl get pods
6  kubectl get pods
7  kubectl get pods
8  kubectl get pods
9  kubectl get pods
10 kubectl get pods
11 kubectl get pods
12 kubectl get pods
13 kubectl get pods
14 kubectl get pods
15 kubectl get pods
16 kubectl get pods
17 kubectl run nginx --image=nginx
18 kubectl get pods
19     kubectl describe pod nginx
20 kubectl edit pod nginx
21 kubectl run nginx --image=nginx
22 kubectl edit pod nginx
23 kubectl delete pod nginx
24 kubectl run nginx --image=nginx --restart=Never
25 kubectl get pods
26 kubectl describe pod nginx
27 kubectl logs nginx
28 history
```

ontrolplane ~ ➔

```
Warning   Failed      9m33s (x4 over 11m)  kubelet      Error: ErrImagePull
Warning   Failed      9m20s (x6 over 11m)  kubelet      Error: ImagePullBackOff
Normal    BackOff     60s (x43 over 11m)   kubelet      Back-off pulling image
```

```
controlplane ~ → kubectl edit pod nginx
Edit cancelled, no changes made.
```

```
controlplane ~ → kubectl run nginx --image=nginx
Error from server (AlreadyExists): pods "nginx" already exists
```

```
controlplane ~ ✗ kubectl edit pod nginx
pod/nginx edited
```

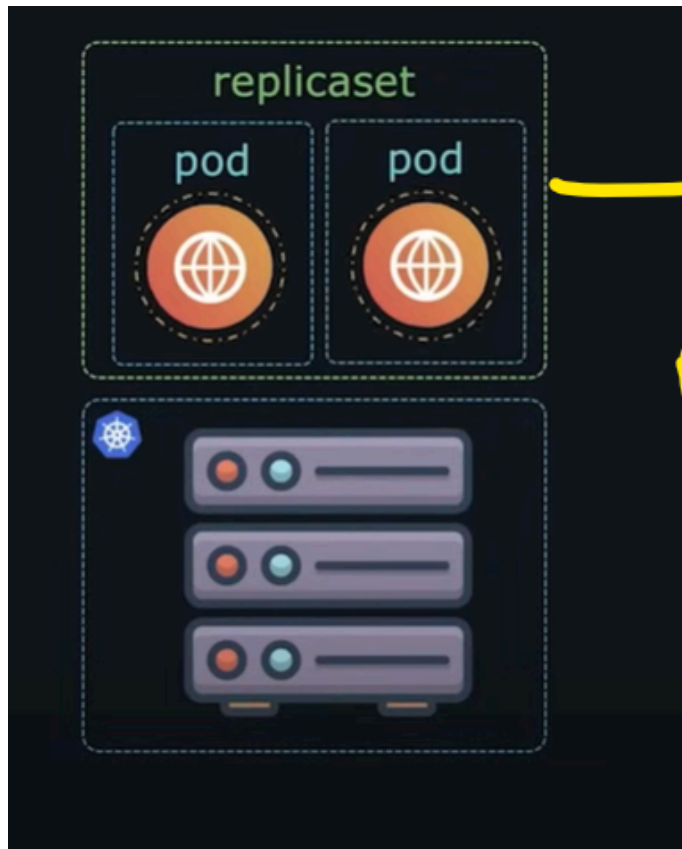
```
controlplane ~ → kubectl delete pod nginx
kubectl run nginx --image=nginx --restart=Never
^C
```

```
controlplane ~ ✗ kubectl delete pod nginx
pod "nginx" deleted
```

```
controlplane ~ → kubectl run nginx --image=nginx --restart=Never
pod/nginx created
```

```
controlplane ~ → kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
nginx     1/1     Running   0           10s
```

## Replica sets:



helps  
multiple  
instances

Describe:

```
controlplane ~ → kubectl describe pod myapp-pod
Name:          myapp-pod
Namespace:     default
Priority:       0
Service Account: default
Node:          controlplane/192.168.50.9
Start Time:    Wed, 26 Feb 2025 15:49:46 +0000
Labels:        app=myapp
               type=front-end
Annotations:   <none>
Status:        Running
IP:            10.42.0.11
IPs:
  IP: 10.42.0.11
Containers:
  nginx-container:
    Container ID:  containerd://2d8a968a565aa5823ca0f6222867923e3509d756614de9454eb122e8c18a
    Image:         nginx
    Image ID:      docker.io/library/nginx@sha256:9d6b58feebd2dbd3c56ab5853333d627cc6e281011
4bcf2072c9496
    Port:          <none>
    Host Port:     <none>
    State:         Running
      Started:     Wed, 26 Feb 2025 15:49:47 +0000
    Ready:         True
    Restart Count: 0
    Environment:   <none>
```

```
- echo Hello Kubernetes! && sleep 3600
```

```
controlplane ~ → kubectl describe pod busybox-pod  
Error from server (NotFound): pods "busybox-pod" not found
```

```
controlplane ~ ✗ vi pod-definition.yml
```

```
controlplane ~ → kubectl describe pod myapp-pod  
Error from server (NotFound): pods "myapp-pod" not found
```

```
controlplane ~ ✗ kubectl apply -f pod-definition.yml  
error: error parsing pod-definition.yml: error converting YAML to JSON: yaml: line 7:  
mapping values are not allowed in this context
```

```
controlplane ~ ✗ vi pod-definition.yml
```

```
controlplane ~ → kubectl apply -f pod-definition.yml  
pod/myapp-pod created
```

```
controlplane ~ → kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
nginx	1/1	Running	0	19m
myapp-pod	1/1	Running	0	28s

```
controlplane ~ →
```

yaml

Copy Edit

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
    app: myapp
    type: front-end
spec:
  containers:
    - name: nginx-container
      image: nginx
```

### Steps to Apply the Fix:

1. Save the corrected YAML as `pod-definition.yaml`.
2. Apply it using:

sh

Copy Edit

```
kubectl apply -f pod-definition.yaml
```

3. Verify that the pod is created successfully:

sh

Copy Edit

```
kubectl get pods
```



```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: myapp-replicaset
  labels:
    app: myapp
    type: front-end
spec:
  template:
    metadata:
      name: myapp-pod
      labels:
        app: myapp
        type: front-end
    spec:
      containers:
        - name: nginx-container
          image: nginx
  replicas: 3
  selector:
    matchLabels:
      type: front-end
```

same