Kubernetes Workload Types

Learn about the seven primary workload types supported by Kubernetes.

Pod

- Smallest deployable unit in Kubernetes
- Runs a single container (or tightly coupled containers)
- Useful for testing or basic deployment

```
apiVersion: v1
kind: Pod
metadata:
  name: busybox-pod
spec:
  containers:
  - name: busybox
    image: busybox:musl
    command: ["sleep", "3600"]
...
```

ReplicaSet

- Ensures a specified number of pod replicas are running
- Used primarily by Deployments
- Rarely created directly

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: busybox-replicaset
spec:
  replicas: 2
  selector:
    matchLabels:
      app: busybox
  template:
    metadata:
      labels:
        app: busybox
    spec:
      containers:
      - name: busybox
        image: busybox:musl
        command: ["sleep", "360"]
...
```

Deployment

- Manages ReplicaSets
- Supports rolling updates and rollbacks
- Ideal for stateless applications

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: busybox-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: busybox
  template:
    metadata:
      labels:
        app: busybox
    spec:
      containers:
      - name: busybox
        image: busybox:musl
        command: ["sleep", "240"]
...
```

StatefulSet

- Manages stateful applications
- Guarantees stable pod identity and storage
- Pods start in order and maintain unique names

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: busybox-statefulset
spec:
  serviceName: "busybox"
  replicas: 2
  selector:
    matchLabels:
      app: busybox
  template:
    metadata:
      labels:
        app: busybox
    spec:
      containers:
      - name: busybox
        image: busybox:musl
        command: ["sleep", "300"]
...
```

DaemonSet

- Runs one pod per node
- Ideal for node-level services like log collectors or network agents

```yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: busybox-daemonset
spec:
  selector:
    matchLabels:
      app: busybox
  template:
    metadata:
      labels:
        app: busybox
    spec:
      containers:
      - name: busybox
        image: busybox:musl
        command: ["sleep", "180"]
...
```

Job

- Runs a task to completion
- Suitable for batch processing or one-off tasks

```
apiVersion: batch/v1
kind: Job
metadata:
  name: busybox-job
spec:
  template:
    metadata:
      name: busybox
    spec:
      restartPolicy: Never
      containers:
      - name: busybox
        image: busybox:musl
        command: ["sleep", "120"]
...
```

CronJob

- Runs Jobs on a schedule
- Useful for recurring tasks (e.g., backups)

```
apiVersion: batch/v1
kind: CronJob
metadata:
  name: busybox-cronjob
spec:
  schedule: "* * * * *"
  jobTemplate:
    spec:
      template:
        spec:
          restartPolicy: OnFailure
          containers:
          - name: busybox
            image: busybox:musl
            command: ["sleep", "15"]
...
```

Summary Table

Workload	Replica Control	Restart	Use Case
Pod	❌	Manual	Basic, single use
ReplicaSet	✅	Yes	Scaling pods
Deployment	✅	Yes	Rolling updates
StatefulSet	✅	Yes	Stateful services
DaemonSet	1 per Node	Yes	Node-wide services
Job	❌	No	Batch/one-time tasks
CronJob	❌	No	Recurring tasks

✅ Ready for Practice?

Next: Let’s try out these workloads hands-on in the lab!