[1. ReplicaSet](https://note.youdao.com/ynoteshare1/iframe.html#6234-1576251593276)

[1.1 创建ReplicaSet](https://note.youdao.com/ynoteshare1/iframe.html#6020-1576286932313)

[1.2 查看ReplicaSet资源](https://note.youdao.com/ynoteshare1/iframe.html#8126-1576286983142)

[1.3 调整ReplicaSet副本数](https://note.youdao.com/ynoteshare1/iframe.html#1140-1576287266297)

[2. Deployment](https://note.youdao.com/ynoteshare1/iframe.html#2110-1576287492991)

[2.1 创建Deployment](https://note.youdao.com/ynoteshare1/iframe.html#7219-1576287507441)

[2.2 升级image](https://note.youdao.com/ynoteshare1/iframe.html#4724-1576291339221)

[2.3 回滚](https://note.youdao.com/ynoteshare1/iframe.html#6134-1576289563714)

[3. DaemonSet](https://note.youdao.com/ynoteshare1/iframe.html#6492-1576287508709)

[3.1 创建DaemonSet](https://note.youdao.com/ynoteshare1/iframe.html#8013-1576291127969)

[3.2 查看DaemonSet资源](https://note.youdao.com/ynoteshare1/iframe.html#8798-1576291574076)

[4. StatefulSet](https://note.youdao.com/ynoteshare1/iframe.html#6280-1576291872817)

[4.1 创建StatefulSet](https://note.youdao.com/ynoteshare1/iframe.html#6832-1576291953819)

[4.2 查看StatefulSet资源](https://note.youdao.com/ynoteshare1/iframe.html#3077-1576298315448)

[5. Job and CronJob](https://note.youdao.com/ynoteshare1/iframe.html#5020-1576287509297)

[5.1 创建 Job](https://note.youdao.com/ynoteshare1/iframe.html#2060-1576298740609)

[5.2 查看Job资源](https://note.youdao.com/ynoteshare1/iframe.html#9848-1576298953992)

[5.3 创建CronJob](https://note.youdao.com/ynoteshare1/iframe.html#1496-1576299060130)

[5.4 查看CronJob](https://note.youdao.com/ynoteshare1/iframe.html#8845-1576299338339)

**1. ReplicaSet**

**1.1 创建ReplicaSet**

<https://kubernetes.io/zh/docs/concepts/workloads/controllers/replicaset/>

cat << EOF > replicaset.yaml

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: nginx

labels:

app: rs-nginx

spec:

# modify replicas according to your case

replicas: 3

selector:

matchLabels:

app: pod-nginx

template:

metadata:

labels:

app: pod-nginx

spec:

containers:

- name: nginx

image: nginx

EOF

kubectl create -f replicaset.yaml

**1.2 查看ReplicaSet资源**

kubectl get replicasets.apps

DESIRED: 请求的副本数

CURRENT：实际运行的副本数

READY：副本数为READY的数量

AGE：生命周期

**1.3 调整ReplicaSet副本数**

# 方法一：修改本地yaml文件的replicas: 3

# 方法二：编辑已经运行资源的yaml文件：kubectl edit replicaset <replicaset\_name>

# 方法三：通过命令行：kubectl scale replicaset --replicas=1 <replicaset\_name>

**2. Deployment**

**2.1 创建Deployment**

<https://kubernetes.io/zh/docs/concepts/workloads/controllers/deployment/>

# 方法一：命令行创建

kubectl run nginx-app --image=nginx:1.9.0 --image-pull-policy=IfNotPresent --replicas=2

# 方法二：yaml创建

cat << EOF > nginx-deployment.yaml

apiVersion: apps/v1 kind: Deployment metadata: name: nginx-deployment labels: app: nginx-deploy spec: replicas: 1 selector: matchLabels: app: nginx-pod template: metadata: labels: app: nginx-pod spec: containers: - name: nginx image: nginx:1.7.6 imagePullPolicy: IfNotPresent ports: - containerPort: 80

EOF

kubectl create -f nginx-deployment.yaml

# 查看创建的Deployment资源

kubectl get deployments.apps

READY：Pod READY的数量

UP-TO-DATE： 升级最新的Pod数量

AVAILABLE：可用的Pod数量

AGE：资源的生命周期

**2.2 升级image**

# 方法一：更改本地yaml，并使用apply升级

# 方法二：使用edit在线更改运行的Deployment，修改container的image

# 方法三：通过命令行升级并记录升级信息：kubectl set image deployment nginx-deployment nginx=nginx:latest --record

# 查看deployment升级的状态

kubectl rollout status deployment nginx-deployment

# 在升级过程可以暂停部署

kubectl rollout pause deployment nginx-deployment

**2.3 回滚**

# 查看历史记录

kubectl rollout history deployment nginx-deployment

# 查看某一个记录的详细信息

kubectl rollout history deployment nginx-deployment --revision=1

# 执行回滚操作

kubectl rollout undo deployment nginx-deployment --to-revision=1

**3. DaemonSet**

**3.1 创建DaemonSet**

<https://kubernetes.io/zh/docs/concepts/workloads/controllers/daemonset/>

cat << EOF > daemonset.yaml

apiVersion: apps/v1

kind: DaemonSet

metadata:

name: fluentd-elasticsearch

labels:

k8s-app: fluentd-logging

spec:

selector:

matchLabels:

name: fluentd-elasticsearch

template:

metadata:

labels:

name: fluentd-elasticsearch

spec:

containers:

- name: fluentd-elasticsearch

image: quay.io/fluentd\_elasticsearch/fluentd:v2.5.2

EOF

kubectl create -f daemonset.yaml

部署不到master上？可能是master本身状态有误

**3.2 查看DaemonSet资源**

DESIRED: 请求的副本数

CURRENT：实际运行的副本数

READY：副本数为READY的数量

UP-TO-DATE： 升级最新的Pod数量

AVAILABLE：可用的Pod数量

NODE SELECTOR: 节点选择器

AGE：生命周期

**4. StatefulSet**

**4.1 创建StatefulSet**

cat << EOF > statefulset.yaml

apiVersion: apps/v1

kind: StatefulSet

metadata:

name: web

spec:

selector:

matchLabels:

app: nginx # has to match .spec.template.metadata.labels

serviceName: "nginx" # \$(podname).(headless server name).namespace.svc.cluster.local

replicas: 3 # by default is 1

template:

metadata:

labels:

app: nginx # has to match .spec.selector.matchLabels

spec:

terminationGracePeriodSeconds: 10

containers:

- name: nginx

image: nginx

ports:

- containerPort: 80

name: web

EOF

kubectl create -f statefulset.yaml

**4.2 查看StatefulSet资源**

# 查看Pod资源

**5. Job and CronJob**

**5.1 创建 Job**

cat << EOF > job.yaml

apiVersion: batch/v1

kind: Job

metadata:

name: pi

spec:

completions: 10

parallelism: 2

template:

spec:

containers:

- name: pi

image: perl

command: ["perl", "-Mbignum=bpi", "-wle", "print bpi(2000)"]

restartPolicy: Never

backoffLimit: 4

EOF

kubectl create -f job.yaml

**5.2 查看Job资源**

NAME： Job的名字

COMPLETIONS：完成的数量/总数量

DURATION：持续时间

AGE：生命周期

**5.3 创建CronJob**

cat << EOF > cronjob.yaml

apiVersion: batch/v1beta1

kind: CronJob

metadata:

name: hello

spec:

schedule: "\*/1 \* \* \* \*"

jobTemplate:

spec:

template:

spec:

containers:

- name: hello

image: busybox

args:

- /bin/sh

- -c

- date; echo Hello from the Kubernetes cluster

restartPolicy: OnFailure

EOF

kubectl create -f cronjob.yaml

**5.4 查看CronJob**

SCHEDULE:时间表

SUSPEND：暂停

ACTIVE：激活的任务

LAST SCHEDULE：最后执行的时间

yaml字段解析：

* .spec.schedule：时间表，必需字段，指定任务运行周期，格式同 [Cron](https://en.wikipedia.org/wiki/Cron)
* .spec.jobTemplate：Job 模板，必需字段，指定需要运行的任务，格式同 [Job](https://jimmysong.io/kubernetes-handbook/concepts/job.html)
* .spec.startingDeadlineSeconds ：启动 Job 的期限（秒级别），该字段是可选的。如果因为任何原因而错过了被调度的时间，那么错过执行时间的 Job 将被认为是失败的。如果没有指定，则没有期限
* .spec.concurrencyPolicy：并发策略
  + Allow（默认）：允许并发运行 Job
  + Forbid：禁止并发运行，如果前一个还没有完成，则直接跳过下一个
  + Replace：取消当前正在运行的 Job，用一个新的来替换