## Replication Controller和ReplicaSet

Replication Controller（复制控制器，RC）和ReplicaSet（复制集，RS）是两种简单部署Pod的方式。因为在生产环境中，主要使用更高级的Deployment等方式进行Pod的管理和部署，所以本节只对Replication Controller和Replica Set的部署方式进行简单介绍。

### Replication Controller

Replication Controller（简称RC）可确保Pod副本数达到期望值，也就是RC定义的数量。换句话说，Replication Controller可确保一个Pod或一组同类Pod总是可用。

如果存在的Pod大于设定的值，则Replication Controller将终止额外的Pod。如果太小，Replication Controller将启动更多的Pod用于保证达到期望值。与手动创建Pod不同的是，用Replication Controller维护的Pod在失败、删除或终止时会自动替换。因此即使应用程序只需要一个Pod，也应该使用Replication Controller或其他方式管理。Replication Controller类似于进程管理程序，但是Replication Controller不是监视单个节点上的各个进程，而是监视多个节点上的多个Pod。

定义一个Replication Controller的示例如下。

apiVersion: v1

kind: **ReplicationController**

metadata:

name: nginx

spec:

replicas: 3

selector:

app: nginx

template:

metadata:

name: nginx

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx

ports:

- containerPort: 80

### ReplicaSet

ReplicaSet是支持基于集合的标签选择器的下一代Replication Controller，它主要用作Deployment协调创建、删除和更新Pod，和Replication Controller唯一的区别是，ReplicaSet支持标签选择器。在实际应用中，虽然ReplicaSet可以单独使用，但是一般建议使用Deployment来自动管理ReplicaSet，除非自定义的Pod不需要更新或有其他编排等。

定义一个ReplicaSet的示例如下：

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: frontend

labels:

app: guestbook

tier: frontend

spec:

# modify replicas according to your case

replicas: 3

selector:

matchLabels:

tier: frontend

matchExpressions:

- {key: tier, operator: In, values: [frontend]}

template:

metadata:

labels:

app: guestbook

tier: frontend

spec:

containers:

- name: php-redis

image: gcr.io/google\_samples/gb-frontend:v3

resources:

requests:

cpu: 100m

memory: 100Mi

env:

- name: GET\_HOSTS\_FROM

value: dns

# If your cluster config does not include a dns service, then to

# instead access environment variables to find service host

# info, comment out the 'value: dns' line above, and uncomment the

# line below.

# value: env

ports:

- containerPort: 80

Replication Controller和ReplicaSet的创建删除和Pod并无太大区别，Replication Controller目前几乎已经不在生产环境中使用，ReplicaSet也很少单独被使用，都是使用更高级的资源Deployment、DaemonSet、StatefulSet进行管理Pod。