

# DEPLOYMENT vs DEPLOYMENT CONFIG

- DC invoca 1 ó + RC  
RC contiene info en un momento dado del estado de DC como plantilla del pod
- D invoca 1 ó + RS  
RS: sucesor de RC
- DC y D: objetos nativos de API Kubernetes
- RC y RS: building blocks

No se deben tocar: pods, RC, RS, se gestionan con DC y/o D.

## Replication Controller (RC)

- Se asegura de que un n° de réplicas del pod esté corriendo en todo momento, actualizando pods (levantando o tirando pods) para llegar a ese número.
- Configuración del recurso:
  - N° réplicas
  - Definición del pod a crear
  - Selector: para identificar los pods que debe “vigilar”

NOTA: Un selector es un conjunto de etiquetas que llevan los pods en sus metadatos.
- No hacen auto-scaling basado en carga o tráfico, no monitoriza eso. Para eso está el recurso externo auto-scaler

## Replica Sets (RS)

- Se asegura de que existe el n° de réplicas deseado en todo momento
- La diferencia con RC es que RS soporta un conjunto basado en un selector y el RC sólo etiquetas. Puede usar expresiones regulares en el selector

## Deployment Config (DC)

- Crea un RC (con la definición del pod) y arranca pods. Si cambia DC → nuevo RC
- Puede llevar hooks de ciclo de vida: antes/después del despliegue del pod “hace cosas”.
- Capacidades:
  - Estrategia de despliegue de versión a una nueva
  - Hooks de ciclo de vida
  - Versionado de apps para poder volver a versión anterior (a mano o autom.) en caso de fallo
  - Escalado manual o automático

- Definición de recurso:
  - Elementos de definición del RC
  - Triggers para crear nuevos despliegues
  - Estrategia de despliegue
  - Hooks de ciclo de vida
- El despliegue se hace con un pod de despliegue que retiene el RC anterior por si falla el nuevo despliegue
- El rollout de cambio de configuración está activado por defecto pero se puede desactivar
- No puede ser escalado si existe un rollout en marcha
- No se puede pausar el pod del deploy

### Deployment (D)

- Descendiente de los DC de OpenShift
- Describen el estado deseado como plantilla de pod
- Crean RS para gestionar los ciclos de vida de los pods
- No soportan rollbacks automáticos en caso de fallo de un nuevo despliegue
- No soportan hooks de ciclo de vida
- No tiene estrategias custom de despliegue
- El rollout de cambio de configuración es implícito, si no se quiere, hay que pausar el deployment: `oc rollout pause deployment/<nombre_deployment>`
- Pueden ser escalados mientras existe un rollout en marcha
- Más rápidos en el despliegue

### NOTAS:

- Recomendación: usar D salvo que se necesite algo específico de los DC.
- DC y D: ambos soportados por OCP
- DC: prefieren consistencia y D: disponibilidad
- En caso de caída de un nodo donde corre un pod:
  - Con DC: no se sustituye el pod por otro (se espera que el nodo “vuelva”), salvo que se borre el nodo porque el pod se gestiona con kubelet. Es kubelet el encargado de borrar el pod.
  - Con D: los pod los gestiona el controller manager que corre en los master, luego se levanta otro pod.

**Refs.:**

[https://docs.openshift.com/container-platform/4.2/applications/deployments/what-deployments-are.html#deployments-comparing-deploymentconfigs\\_what-deployments-are](https://docs.openshift.com/container-platform/4.2/applications/deployments/what-deployments-are.html#deployments-comparing-deploymentconfigs_what-deployments-are)