问题 1. OpenShift 中一直在用 DeploymentsConfig（简称 dc），为啥不用 K8S 中的 Deployments?

OpenShift3（简称 OCP3）主要是用 dc。K8S 1.0 在刚推出的时候，缺乏很多企业客户需要的组件，如 Ingress、bc、dc。OCP 为了满足客户的需求，红帽自行写代码进行补充。所以是红帽先开发了 bc、dc。
OCP 的开发团队尝试将 OCP 的代码提交到 K8S 社区，有的被采纳了。有的虽然未必采纳，但 K8S 采纳了对应的思路，开发了自己的组件，如 Deployments。

所以说，这世上是先有的 dc，后有的 Deployments。

OCP4 开始全面兼容 deployments，OCP4 自身的组件，也都是用 deployments 而非 dc（为了和 K8S 统一）。红帽的建议是：如果能使用 deployments，就使用 deployments（建议首选它）。如果有的应用还想使用 dc，也可以再使用 dc。在 OCP 中部署应用，可以选择自以 dc 还是 deployments 方式部署：

问题 2：dc 和 depoyements 设计上有什么区别?

从 dc 到 pod 的关系是：
deploymentconfig → replicationcontroller → pod。

K8S 社区的容器部署管理使用 deployment。从 deployment 到 pod 的关系是：

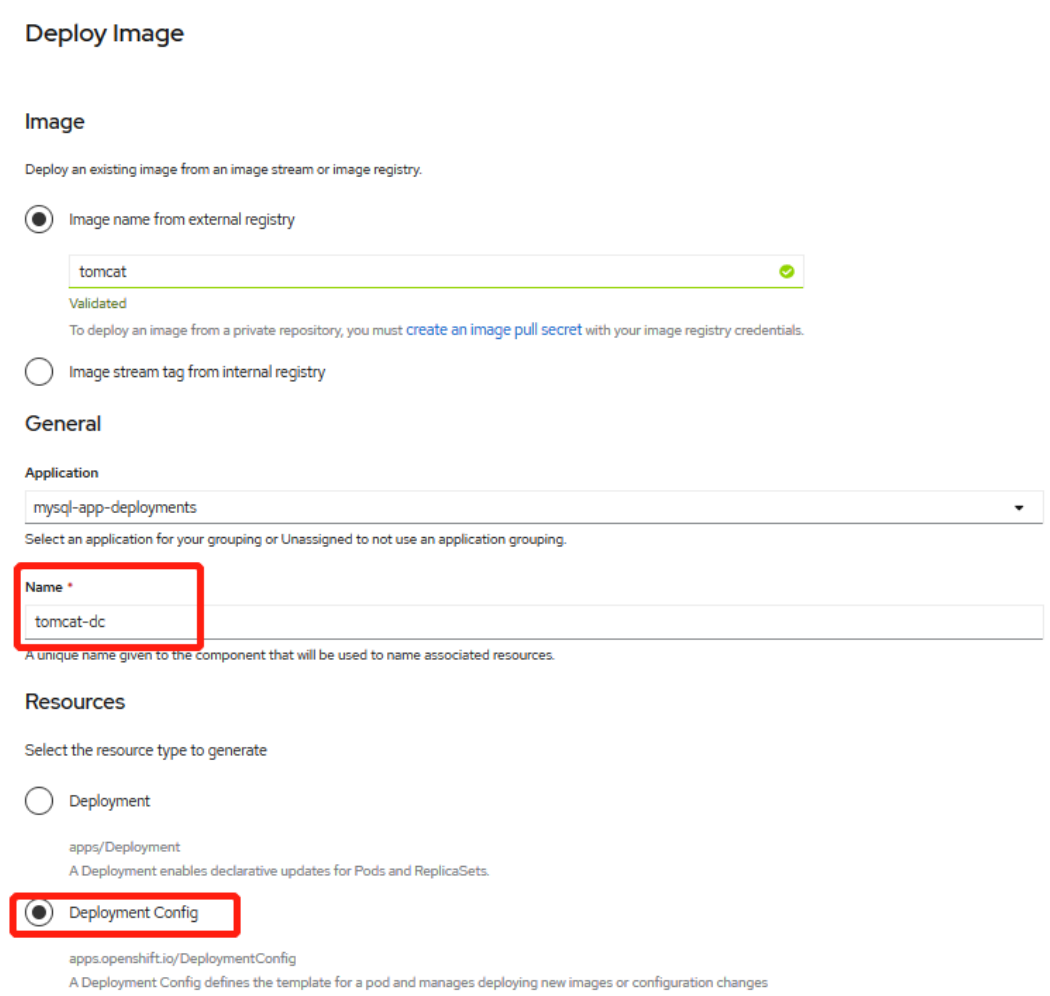
deployment → replicaset → pod
所以说，dc 里 pod 副本控制是 rc，deployments 的副本控制是 rc。

从设计架构上说：

根据 CAP 定理的属性，DeploymentConfigs 首选一致性，而 deployment 则将可用性置于一致性之上。
对于 dc：如果运行 deployer Pod（dc 在执行部署时，会有个 deployment pod）的 OCP 节点发生故障，deployer Pod 不会被再其它 OCP 节点重启，他会一直等待本 node 恢复或者 node 手工被删除。手工删除 node 也会把连带的 deployer Pod 删除。这意味着我们不能删除 deployer Pod 来取消 rollout，因为 kubelet 负责删除关联的 Pod。

Deployments rollouts 被 controller manager 驱动，controller manager 以 HA 模式运行在 OCP 的多个 master 上， 并使用领导者选举算法来评估可用性而不是一致性。在发生故障期间，其他主服务器可以同时对同一 Deployment 进行操作，也就是在其他节点重启。

问题 3：听着太绕，直接说两者在应对节点故障时候的反应?
我们做个验证。在 OCP 中分别通过 dc 和 deployments 方式部署 tomcat 应用：
通过 dc 部署 tomcat：



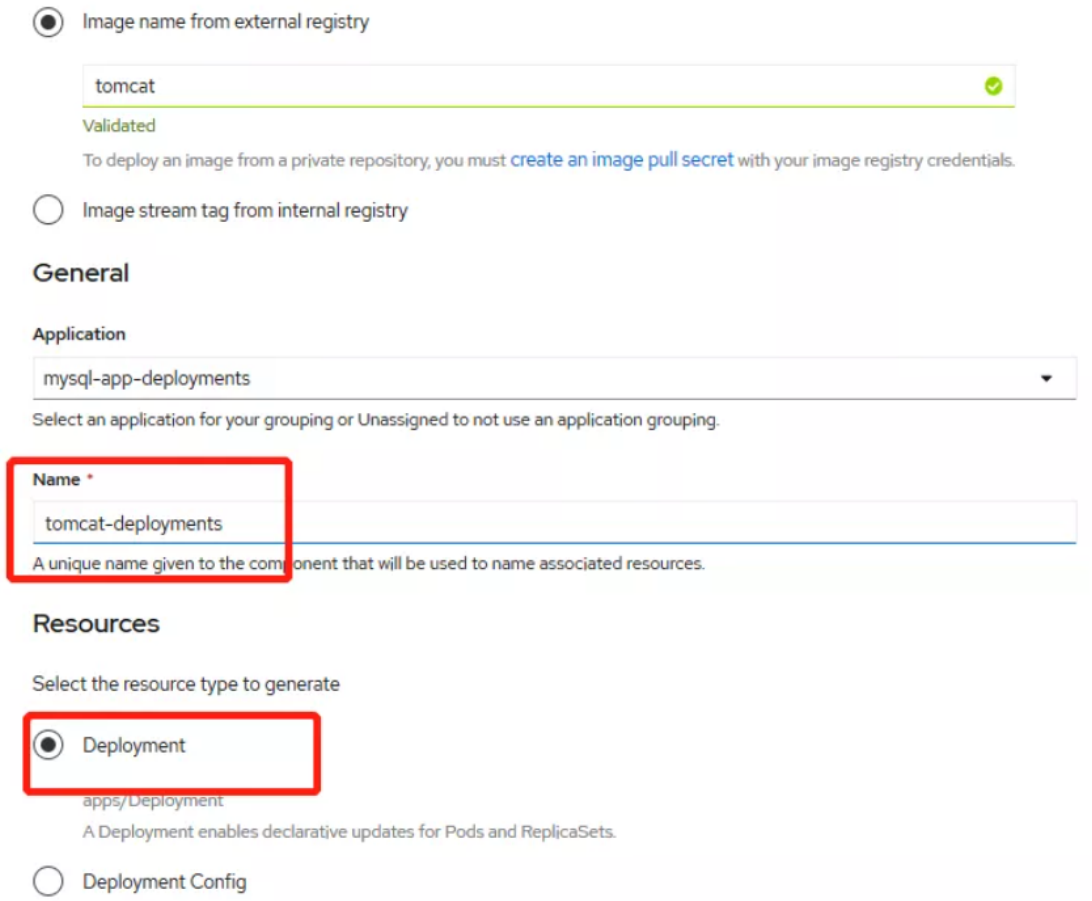
dc 的部署会有 deploy pod：

```
[centos@lb.weixinyucluster ~]$ oc get pods |grep -i tomcat-dc
tomcat-dc-1-7d8mj                  1/1     Running            0          33s
tomcat-dc-1-deploy                 0/1     Completed          0          35s
```

部署完毕后，tomcat-dc 运行在 worker-2 节点：

```
[centos@lb.weixinyucluster ~]$ oc get pods -o wide |grep -i tomcat-dc
tomcat-dc-1-7d8mj                  1/1     Running            0          55s    10.129.2.9   worker-2
ne>           <none>
tomcat-dc-1-deploy                 0/1     Completed          0          57s    10.129.2.8   worker-2
ne>           <none>
```

通过 deployments 方式部署 tomcat：



部署完毕后，我们看到 tomcat-dc 和 tomcat-deployment 都运行在 worker-2:

```
[centos@lb.weixinyucluster ~]$ oc get pods -o wide |grep -i tomcat
tomcat-dc-1-7d8mj                   1/1     Running           0          92s     10.129.2.9   worker-2
one>          <none>
tomcat-dc-1-deploy                  0/1     Completed         0          94s     10.129.2.8   worker-2
one>          <none>
tomcat-deployments-8684b7b879-x2xr5   1/1     Running           0          2m45s   10.129.2.7   worker-2
one>          <none>
```

我们将 worker-2 宕机:

[core@worker-2 ~]$ sudo halt -q

```
[centos@lb.weixinyucluster ~]$ oc get nodes
NAME       STATUS     ROLES    AGE     VERSION
master-0   Ready      master   6d1h    v1.18.3+6025c28
master-1   Ready      master   6d      v1.18.3+6025c28
master-2   Ready      master   6d1h    v1.18.3+6025c28
worker-0   Ready      worker   6d      v1.18.3+6025c28
worker-1   Ready      worker   6d      v1.18.3+6025c28
worker-2   NotReady   worker   5d22h   v1.18.3+6025c28
```

然后观察 tomcat-dc 和 tomcat-deployments 再 node 超时后，都在 worker-0 上重启，时间相同（pod 运行的时间都是 76s）:

```
[centos@lb.weixinyucluster ~]$ oc get pods -o wide |grep -i tomcat
tomcat-dc-1-957mh                   1/1     Running            0          76s     10.131.0.19   worker-0
none>           <none>
tomcat-dc-1-cpmml                   1/1     Terminating        0          8m14s   10.129.2.11   worker-2
none>           <none>
tomcat-deployments-8684b7b879-tb2vj   1/1     Running            0          76s     10.131.0.18   worker-0
none>           <none>
tomcat-deployments-8684b7b879-v9d24   1/1     Terminating        0          8m14s   10.129.2.10   worker-2
none>           <none>
```

接下来，我们再进行一个验证，在触发 dc rollut 的同时。reboot tomcat-dc 所在的节点，我们看到 deploy pod 会一直 pendding：

```
[centos@lb.weixinyucluster ~]$ oc get nodes
NAME       STATUS     ROLES    AGE    VERSION
master-0   Ready      master   6d1h   v1.18.3+6025c28
master-1   Ready      master   6d1h   v1.18.3+6025c28
master-2   Ready      master   6d1h   v1.18.3+6025c28
worker-0   Ready      worker   6d     v1.18.3+6025c28
worker-1   NotReady   worker   6d     v1.18.3+6025c28
[centos@lb.weixinyucluster ~]$  oc get pods -o wide |grep -i tomcat
tomcat-dc-2-deploy                  0/1     Completed          0          6m43s   10.128.2.16   worker-1
<none>          <none>
tomcat-dc-3-deploy                  0/1     Completed          0          3m31s   10.128.2.5    worker-1
<none>          <none>
tomcat-dc-3-mbnf5                   1/1     Running            0          2m50s   10.131.0.23   worker-0
<none>          <none>
tomcat-dc-4-deploy                  0/1     ContainerCreating  0          40s     <none>        worker-1
<none>          <none>
tomcat-deployments-8684b7b879-tb2vj   1/1     Running            0          16m     10.131.0.18   worker-0
<none>          <none>
```

我们对 deployments 进行相同的操作：在触发 deployments rollout 的同时，重启 tomcat-deployment 所在的节点，即 worker-0

```
[centos@lb.weixinyucluster ~]$  oc get pods -o wide |grep -i tomcat
tomcat-dc-2-deploy                  0/1     Completed          0          9m44s   10.128.2.16   worker-1   <
none>           <none>
tomcat-dc-3-deploy                  0/1     Completed          0          6m32s   10.128.2.5    worker-1   <
none>           <none>
tomcat-dc-3-mbnf5                   1/1     Running            0          5m51s   10.131.0.23   worker-0   <
none>           <none>
tomcat-deployments-8684b7b879-tb2vj   1/1     Running            0          19m     10.131.0.18   worker-0   <
none>           <none>
[centos@lb.weixinyucluster ~]$ oc get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
mysql-deployment    0/1     1            0           45m
php-deployments     0/1     1            0           41m
tomcat-deployments  1/1     1            1           38m
[centos@lb.weixinyucluster ~]$ oc patch deployment/tomcat-deployments --patch "{\"spec\":{\"template\":{\"meta
data\":{\"annotations\":{\"last-restart\":\"`date +'%s'`\"}}}}}"
```

tomcat-deployments 很快在 worker-1 节点上成功部署：

```
[centos@lb.weixinyucluster ~]$ oc get nodes
NAME       STATUS     ROLES    AGE    VERSION
master-0   Ready      master   6d1h   v1.18.3+6025c28
master-1   Ready      master   6d1h   v1.18.3+6025c28
master-2   Ready      master   6d1h   v1.18.3+6025c28
worker-0   NotReady   worker   6d     v1.18.3+6025c28
worker-1   Ready      worker   6d     v1.18.3+6025c28
[centos@lb.weixinyucluster ~]$  oc get pods -o wide |grep -i tomcat
tomcat-dc-2-deploy                  0/1     Completed          0          11m     10.128.2.16   worker-1
none>           <none>
tomcat-dc-3-deploy                  0/1     Completed          0          8m41s   10.128.2.5    worker-1
none>           <none>
tomcat-dc-3-mbnf5                   1/1     Running            0          8m      10.131.0.23   worker-0
none>           <none>
tomcat-deployments-65476ff9cc-wqdx5   1/1     Running            0          41s     10.128.2.13   worker-1
none>           <none>
tomcat-deployments-8684b7b879-tb2vj   1/1     Terminating        0          21m     10.131.0.18   worker-0
none>           <none>
```

结论：在应对节点故障时：dc 和 deployments 部署的 pod 没有区别。但如果 dc 和 deployments 正在 rollout 时，或 dc 进行部署时，对应的 OCP 节点发生了重启，dc 会一直

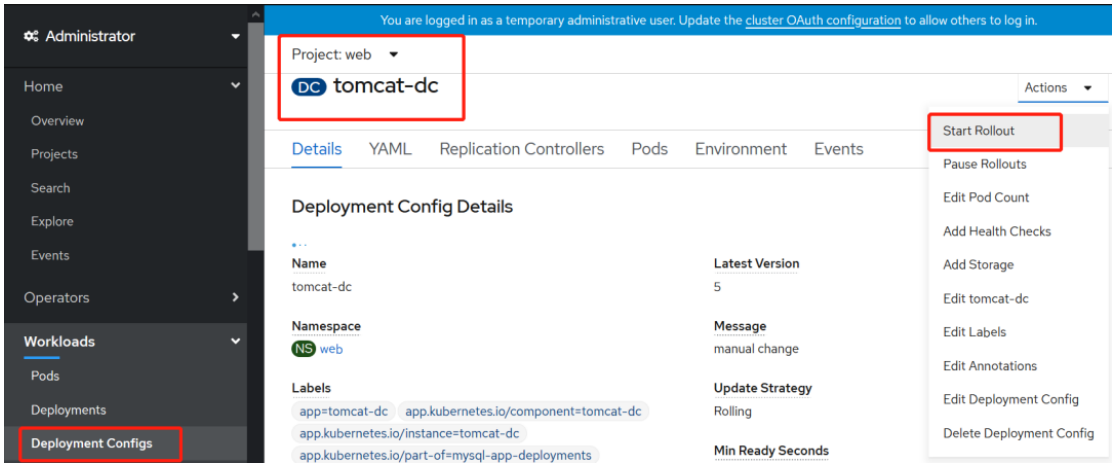
等待节点恢复（主要是有 deployer pod 这个实体存在），而 deployments 则会迅速在其他可用节点重启。

问题 4：在 OCP 上，dc 和 deployment 到底选哪个?

OCP4 默认建议使用 deployments。如果以前 OCP3 的应用用的 dc，迁移到 OCP4 上，还可以继续用 dc。

dc 强在版本控制，rollout 很方便。我们通过命令行 oc rollout latest/dc_name 就能实现，也很方便 rollback：oc rollout history dc/<name> --revision=1

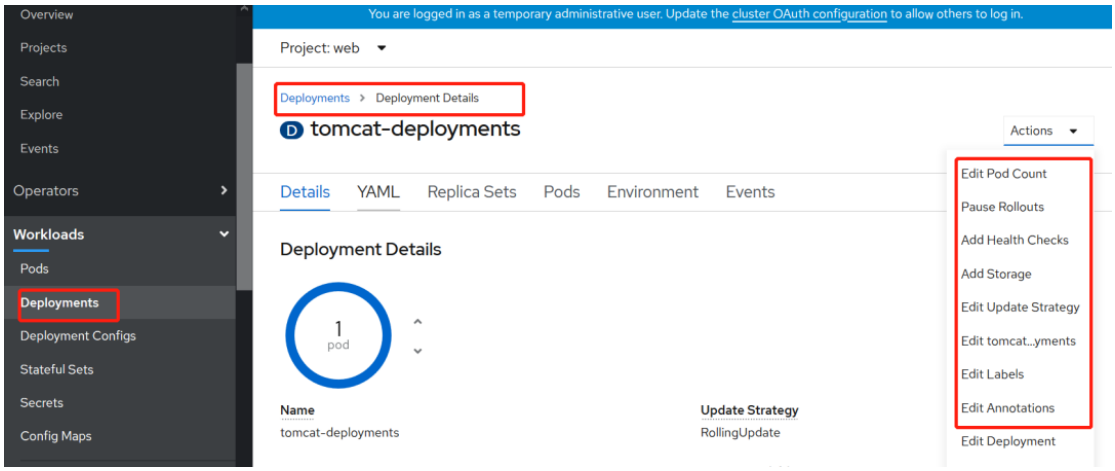
OCP UI 上也有选项：



deployment 由 controller manager 控制，我们会发现 pod 部署速度相比 dc 会快一点。当 pod 部署的时候，如果节点发生故障，pod 会用比较短的时间在其他节点重启（可以简单理解 deployment 的被控制链更短）。

```
[centos@lb.weixinyucluster ~]$ oc get pods -n openshift-controller-manager
NAME                          READY   STATUS    RESTARTS   AGE
controller-manager-4pchw      1/1     Running   0          2d19h
controller-manager-82xqq      1/1     Running   1          2d19h
controller-manager-qjql8      1/1     Running   0          2d19h
```

deployment 默认是无法像 dc 那样通过 cli 或者 ui 进行 rollout 触发的（配置有变化会自动进行）：



如果非要向 dc 那样手工触发 depoyments 的 rollout，那通过修改模板中的注释的值来完成：

#oc patch deployment/tomcat-deployments --patch "{\"spec\":{\"template\":{\"metadata\":{\"annotations\":{\"last-restart\":\"`date +'%s'`\"}}}}}"

最后多说一句，OCP 中应用的 livess 和 readiness，建议都配置上。