kubernetes使用参考文档

一、在kubernetes部署nginx服务

Replication Controller（简称RC）

RC是kubern系统中的核心概念之一，简单来说，它其实定义了一个期望的场景，及声明某种Pod的副本数量在任意时刻都符合某个预期值，所以RC'的定义包括如下几部分：

>> Pod期待的副本数量（replicas）

>> 用于筛选目标Pod的Label Selector

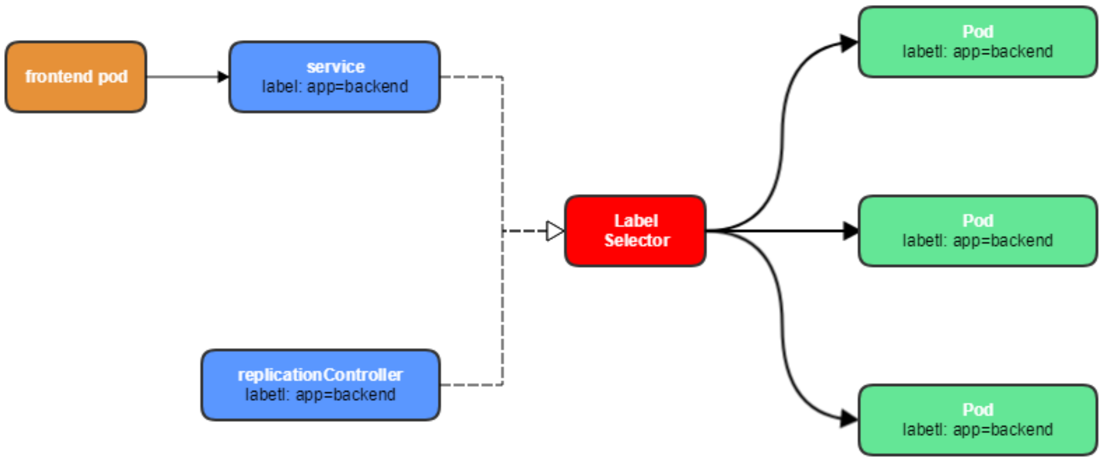
>> 当Pod的副本数量小于预期值数量的时候，用于创建新的Pod的Pod模板

//配置nginx的副本控制器配置文件，也可以通过‘—’将service和副本控制器放在一起啊，为了方便查看，此处不做整合

```
# cat nginx-rc.yaml     # nginx
apiVersion: v1      #
kind: ReplicationController   # RC
metadata:       # name
  name: nginx-controller   # RC
spec:        # ReplicationController
  replicas: 4      # nginx
  selector:
    name: nginx      # "name: nginx"pod" nginx-controller"
  template:
    metadata:
      labels:
        name: nginx     # podRC selector
    spec:
      imagePullSecrets:
      - name: regsecret
      containers:
        - name: nginx    #
          image: 10.22.60.25/test/nginx  # docker
          ports:
            - containerPort: 80  #
          resources:    #
            requests:    #
              memory: "64Mi"  # 64M
              cpu: "200m"   # 0.2CPU
            limits:     #
              memory: "128Mi"  # 128M
              cpu: "300m"   # 0.3CPU
      nodeSelector:     # “disktype: ssdnode”
        disktype: ssd
```

Service（服务）

Service是真实应用服务的抽象，定义了Pod的逻辑集合和访问这个Pod集合的策略。Service将代理Pod对外表现为一个单一接口访问，外部不需要了解后端Pod如何运行，这个给扩展和维护带来了很多的好处，提供了一套简化的服务代理和发现机制。



从图中我们看到，Kubernetes的service定义了一个服务的访问入口地址，前端你的应用Pod通过这个入口地址访问其背后的一组由Pod副本组成的集群实例。

service与其后端Pod副本集群之间则是通过Label
Select来实现“无缝对接”的。而RC的作用实际上是service的服务能力和服务质量始终处于预期的标准

Kubernetes Service定义了外界访问一组特定Pod的方式。Service有自己的IP和端口，Service为Pod提供了负载均衡

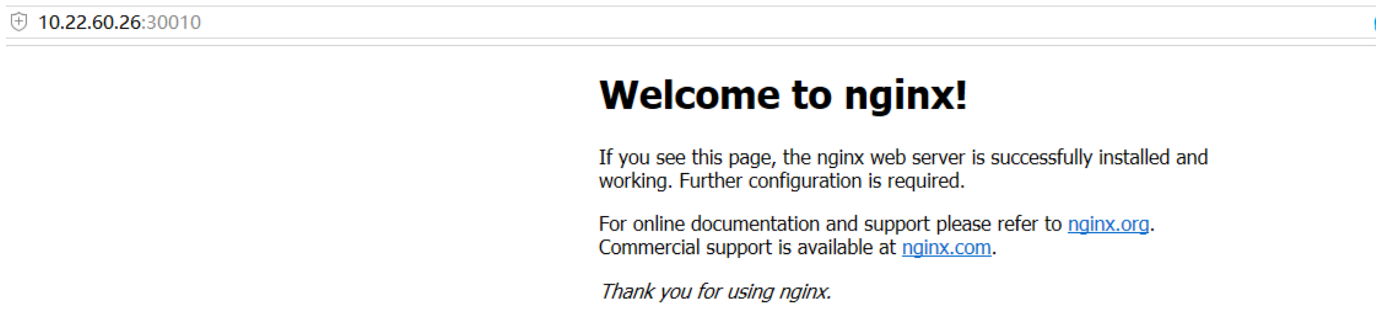
```
# cat nginx-service-nodeport.yaml
apiVersion: v1
kind: Service      # Service
metadata:
  name: nginx-service-nodeport  # Service
spec:
  ports:
    - port: 8000     #
      targetPort: 80
      protocol: TCP     #
      nodePort: 30010    #
  type: NodePort
  selector:       #
    name: nginx
```

使用yaml文件部署服务

```
# kubectl  apply -f nginx-rc.yaml
replicationcontroller "nginx-controller" created
# kubectl  apply -f nginx-service-nodeport.yaml
service "nginx-service-nodeport" created
# kubectl  get pod -o wide
NAME                     READY     STATUS     RESTARTS    AGE       IP
NODE
nginx-controller-58q29   1/1       Running    0           1m
192.168.2.54   odck8sno101
nginx-controller-6nc5h   1/1       Running    0           1m
192.168.2.56   odck8sno101
nginx-controller-88rmn   1/1       Running    0           1m
192.168.2.53   odck8sno101
```

通过浏览器访问任意node节点或master节点的30010端口

http://IP:30010


10.22.60.26:30010

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

二、在kubernetes部署前后台的服务

后台安装

```
# cat cbs-bat.yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: core-cbs-bat-controller
spec:
  replicas: 2
#  selector:
#    name: core
  template:
    metadata:
      labels:
        name: core-cbs-bat
    spec:
      imagePullSecrets:
      - name: cbs-bat-regsecret
      containers:
        - name: cbs-bat
          image: 10.22.60.25/overseas_core/cbs-bat:v1.6.1.0


# cat cbs-onl.yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: core-cbs-onl-controller
spec:
  replicas: 2
#  selector:
#    name: core
  template:
    metadata:
      labels:
        name: core-cbs-onl
    spec:
      imagePullSecrets:
      - name: cbs-onl-regsecret
      containers:
        - name: cbs-onl
          image: 10.22.60.25/overseas_core/cbs-onl:v1.6.1.0
```

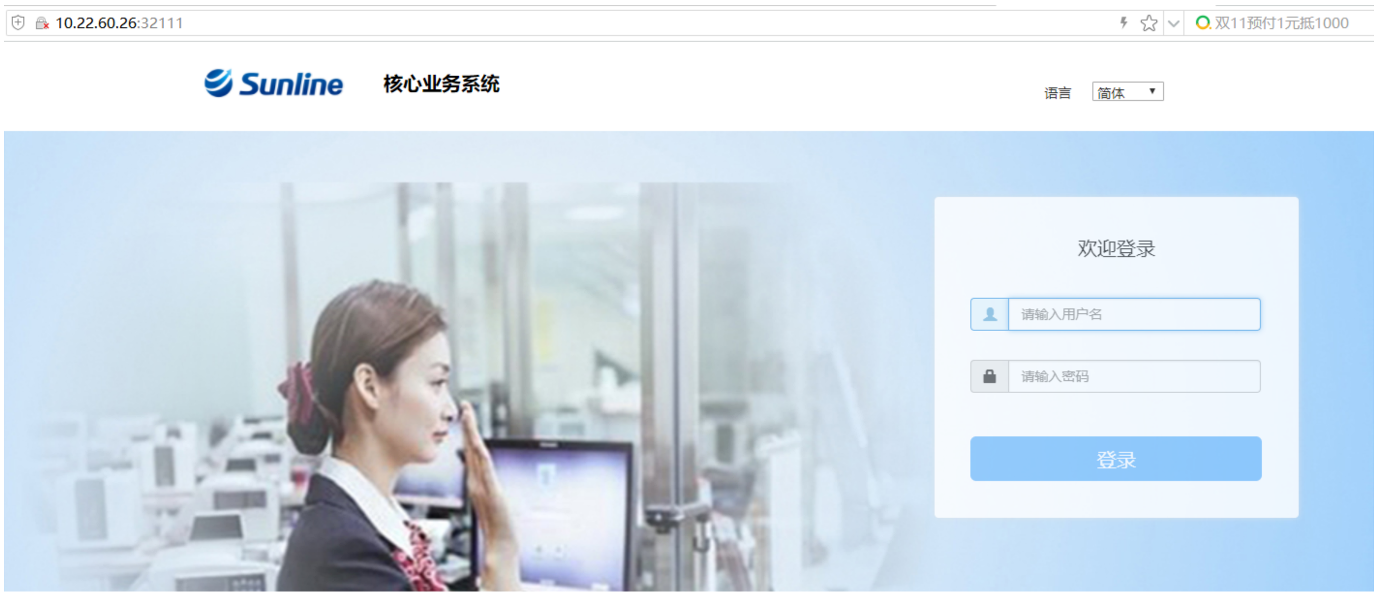
柜面

```
# cat bds.yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: core-bds-controller
spec:
  replicas: 2
  template:
    metadata:
      labels:
        name: core-bds-test
    spec:
      imagePullSecrets:
      - name: core-bds-regsecret
      containers:
        - name: core-bds-test
          image: 10.22.60.25/overseas_core/bds:v1.6.1.0
          ports:
            - containerPort: 8080
---
apiVersion: v1
kind: Service
metadata:
  name: core-bds-svc
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: 8080
      protocol: TCP
      nodePort: 32111   #
  selector:
    name: core-bds-test   #
```

部署服务：

```
# kubectl apply -f  cbs-bat.yaml
replicationcontroller "core-cbs-bat-controller" created
# kubectl apply -f  cbs-onl.yaml
replicationcontroller "core-cbs-onl-controller" created
# kubectl apply -f  bds.yaml
replicationcontroller "core-bds-controller" created
service "core-bds-svc" created
```

通过前端映射端口访问测试



10.22.60.26:32111

双11预付1元抵1000

Sunline 核心业务系统

语言 简体

欢迎登录

请输入用户名

请输入密码

登录