# 容器部署操作手册

进行部署操作时需接入至服务器所在内网，并且根据不同的单位使用不同的namespace

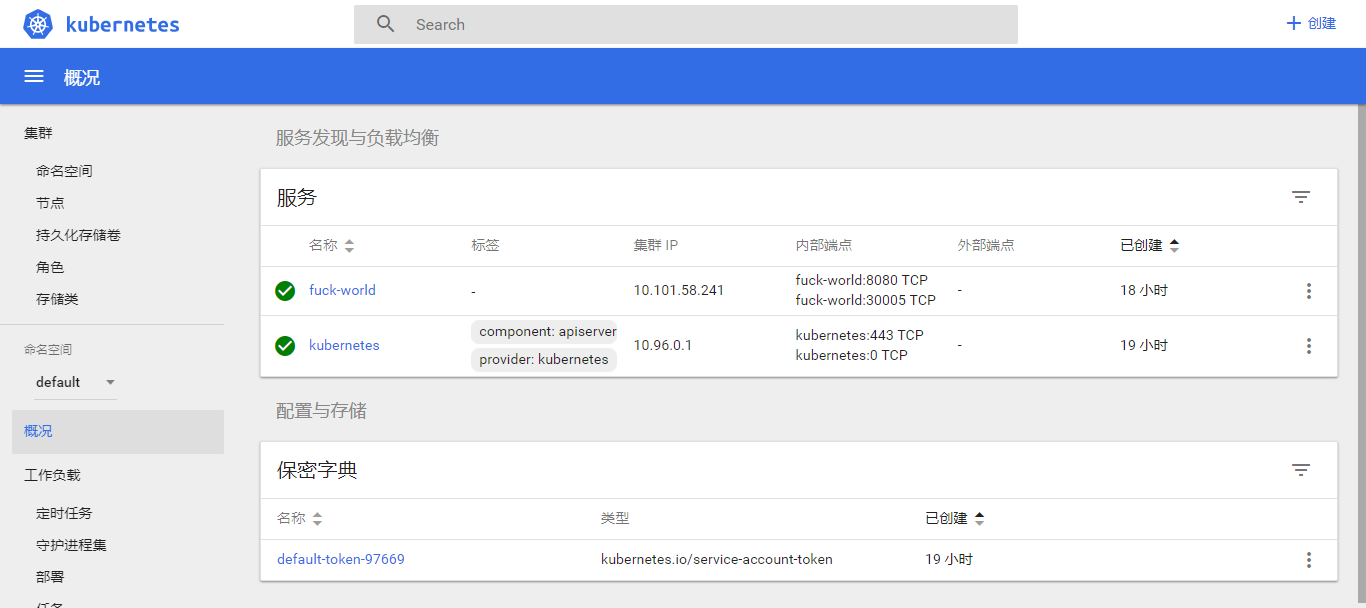
1. **部署模块**

\*若该所属单位第一次部署则参考”namespaces”文件夹中的”cao.yaml”进行修改并namespace创建;之后该单位部署服务应用均使用该namespace。

1. **页面部署**
   1. **页面访问**

浏览器地址栏 http://59.212.147.64:32225

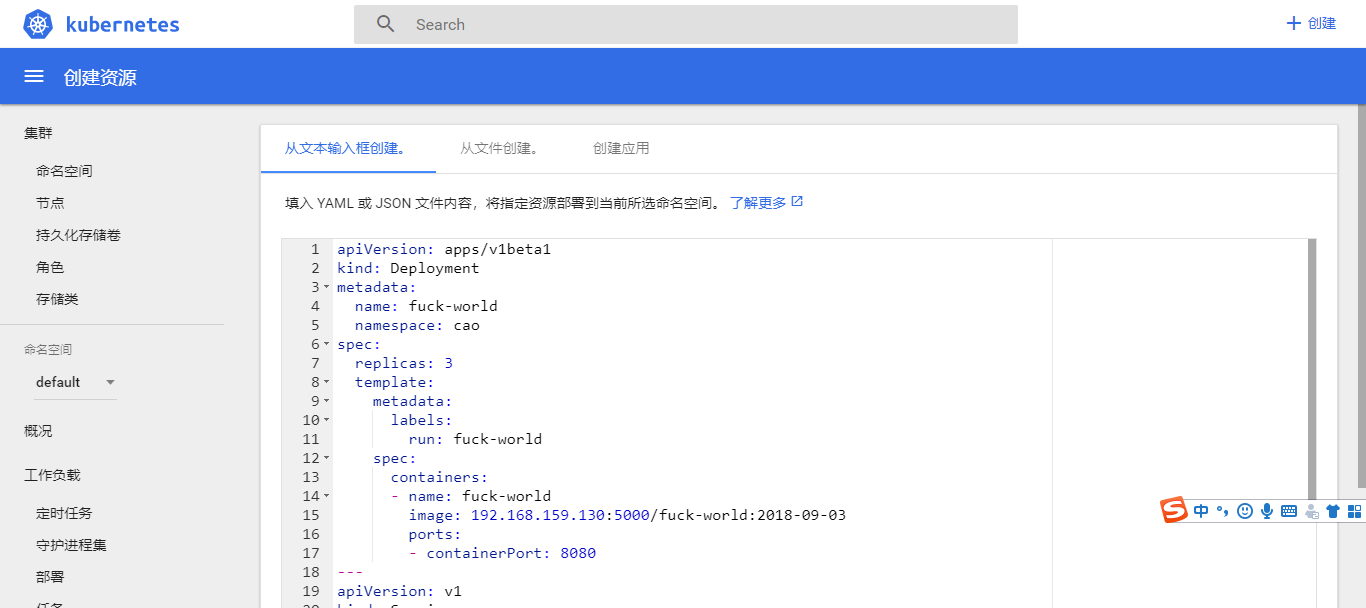
* 1. **顶部右上角点击”创建”**



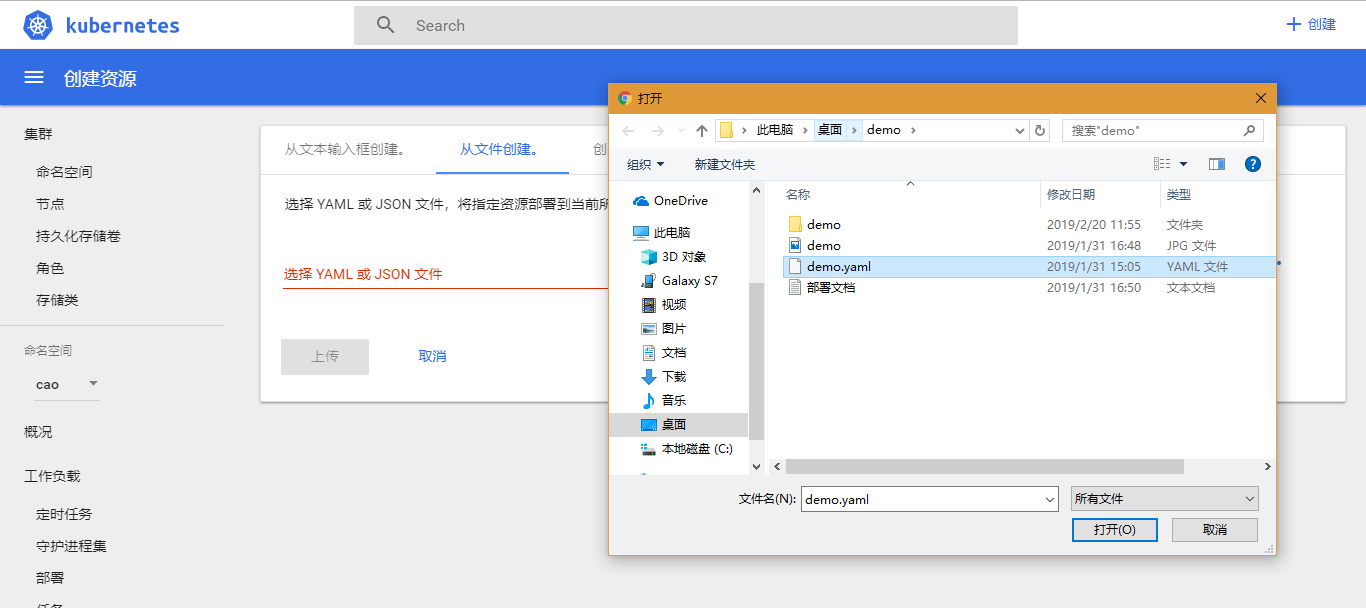
* 1. **选择”从文本框创建”或”从文件创建”**



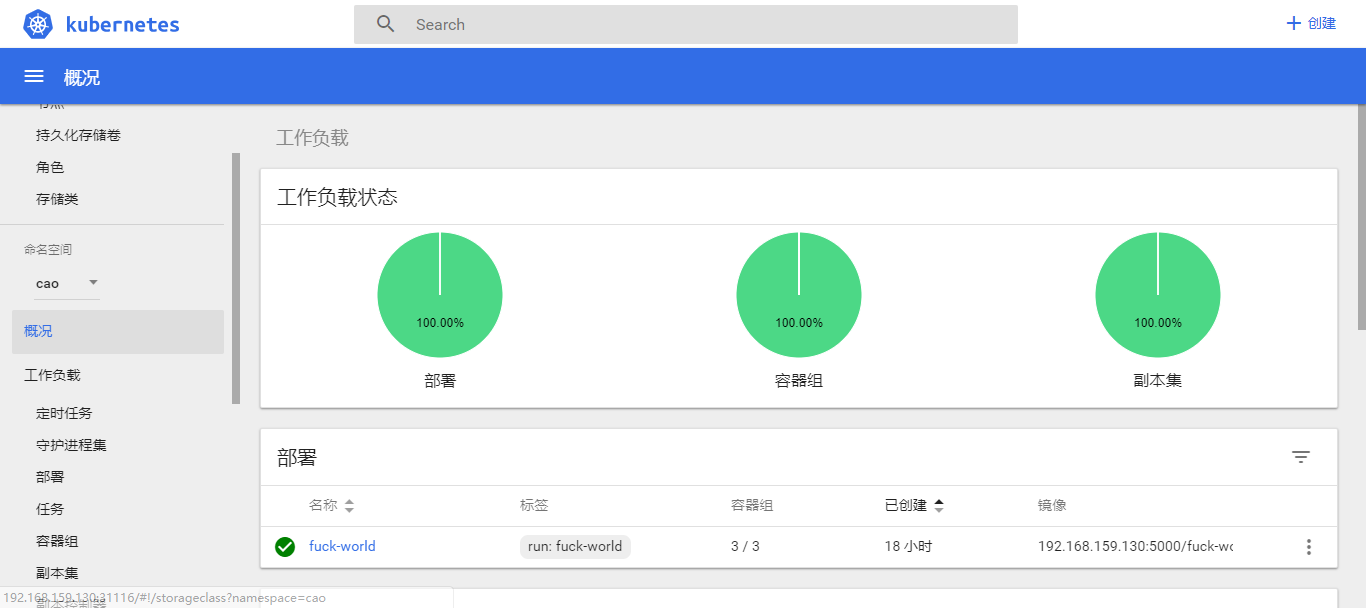
* + 1. **文本框创建**



* + 1. **文件创建**



* 1. **查看**



左侧功能栏在”命名空间”处选择对应部署的命名空间进行相关操作进而查看部署内容。

1. **命令行部署**
   1. **进入主节点服务器**

###### Shell:

ssh [root@59.212.147.64](mailto:root@59.212.147.64)

或用xshell等工具连接主节点服务器

* 1. **向服务器内传入.yaml文件或新建.yaml文件并写入内容**
  2. **创建或更新服务**

###### Shell

kubectl apply -f demo.yaml

注:demo.yaml参考demo文件夹中的”demo.yaml”

###### 打印:

deployment.apps/demo created

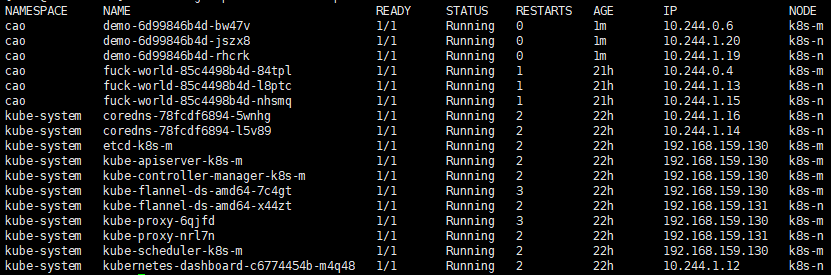
service/demo created

1. **监控模块**
2. **查看pods(容器)**

###### Shell:

kubectl get pods --all-namespaces -o wide

###### 打印:



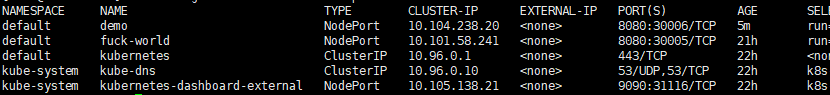
查看到当前状态，running为创建成功并正在运行

1. **查看service(网络端口映射)**

###### Shell:

kubectl get service --all-namespaces -o wide

###### 打印:



1. **调试模块**
2. **删除**
   1. **使用.yaml文件进行删除**

###### Shell:

kubectl delete -f demo.yaml

###### 打印:

deployment.apps "demo" deleted

service "demo" deleted

* 1. **直接删除**

###### Shell:

kubectl delete service --namespace=cao bug

删除service或者pod或者pv/pvc均可以

查看容器组详细信息

###### Shell:

kubectl describe pods --namespace=cao demo

打印内容过多自行操作查看。

1. **进入容器**

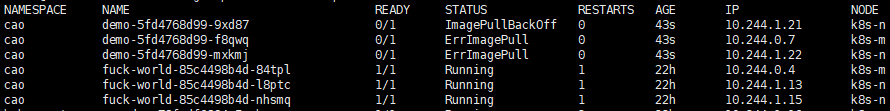
###### Shell:

kubectl --namespace=cao exec -it demo-6d99846b4d-bw47v /bin/bash

注:” demo-6d99846b4d-bw47v”为pods查看命令打印出的pods名。一般用于查看”running”状态容器内部多项日志信息。(在调试时多访问几次再进行进入容器查看日志等操作)

1. **非running容器调试**
   1. **使用describe查看容器组信息**

查看pods时打印出



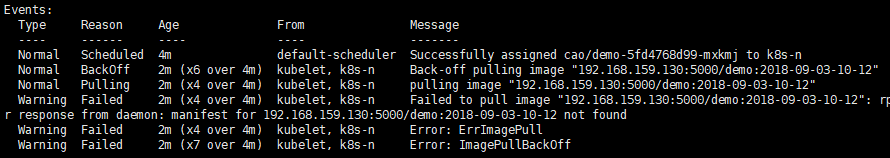
“ImagePullBackOff”、“ErrImagePull”或其他非running状态时

查看容器组详细信息查看容器组问题

###### Shell:

kubectl describe pods --namespace=cao demo

###### 打印结果查看Events事件信息:



即可看到该容器组问题

* 1. **使用logs查看容器信息**

###### Shell:

kubectl logs demo-5fd4768d99-9xd87

###### 打印:

Error from server (BadRequest): container "demo" in pod "demo-5fd4768d99-9xd87" is waiting to start: trying and failing to pull image

注:若容器running也可用该方式查看，例如tomcat报错



* 1. **根据上述3.1和3.2内容对容器进行操作**

例:

修改代码重新打包镜像，上传至私有镜像仓库后进行删除服务应用重新部署。

1. **数据挂载**

若服务运行时有文件生产并有必要进行持久化时，则设计数据挂载。

创建时pv、pvc没有顺序，但最后才能进行服务应用的部署。

删除时顺序为”服务应用→PVC→PV”进行删除。

1. **创建**
   1. **创建pv**

###### Shell:

kubectl apply -f demo-pv.yaml

###### 查看

###### Shell:

kubectl get pv

创建pv后”status”为” Available”

* 1. **创建pvc**

###### Shell:

kubectl apply -f demo-pvc.yaml

###### 查看

kubectl get pvc

创建pvc后”status”为”Pending”

若pv和pvc均没有问题成功创建则pv、pvc状态均为”Bound”

* 1. **部署应用时的模板也要参考”pv-pvc”文件夹中的demo.yaml文件进行修改**

1. **删除**
   1. **删除服务应用**

###### Shell:

kubectl delete -f demo.yaml

* 1. **删除pvc**

###### Shell:

kubectl delete -f demo-pvc.yaml

* 1. **删除pv**

###### Shell:

kubectl delete -f demo-pv.yaml

1. **docker镜像**
2. **加载tar包**

###### Shell:

docker load -I 服务名.tar

1. **查看load后的tar包**

###### Shell:

docker images

* 1. **若不符合命名要求则镜像名**

###### Shell:

docker tag 当前名称:版本号 59.212.147.64:5000/服务名:版本号

若镜像版本号为latest则不需要再tag时写入版本号

例: docker tag 当前名称59.212.147.64:5000/服务名:版本号

1. **上传tar包至私有仓库**

###### Shell:

docker push 59.212.147.64:5000/服务名:版本号

1. **部署模板**

Demo文件夹下各目录中的.yaml文件均为各模块部属配置;.jepg图片则为修改指导。