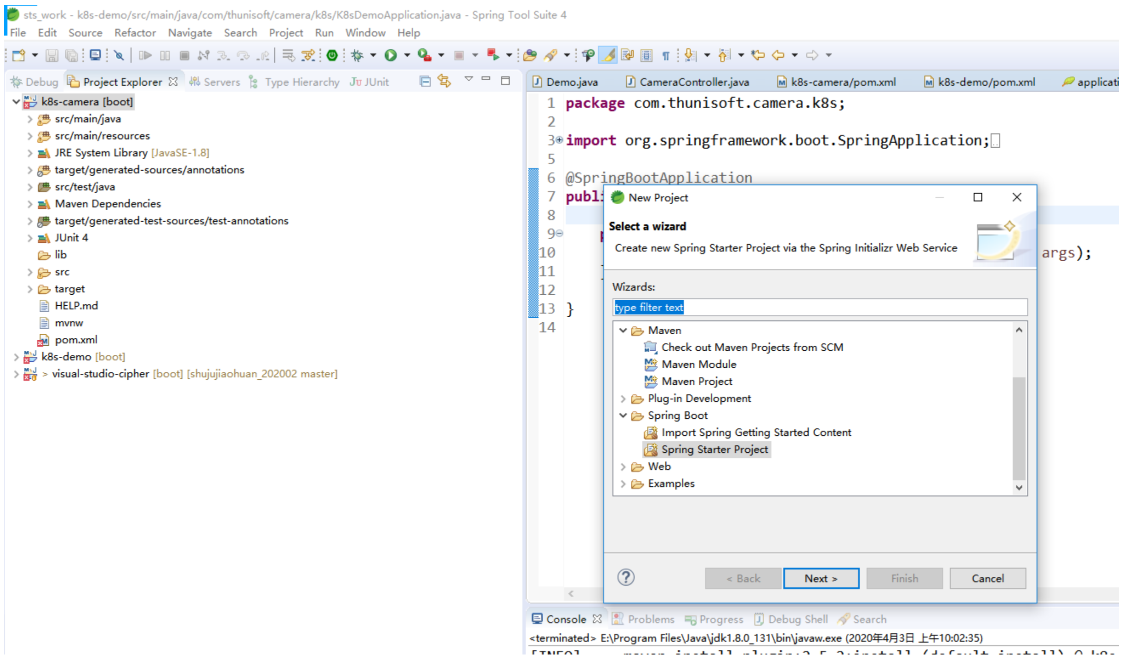
k8s-6、k8s+springBoot项目

笔记本: <Inbox>
创建时间: 2020/4/3 9:50 **更新时间:** 2020/4/3 13:46
作者: 王鹏

（1）创建一个springBoot项目，使用idea或者sts都可以创建一个简单的springBoot项目，我这里使用sts创建项目

（2）点击【file】，点击【new】，点击【project】，点击【spring boot】，点击 【spring starter project】



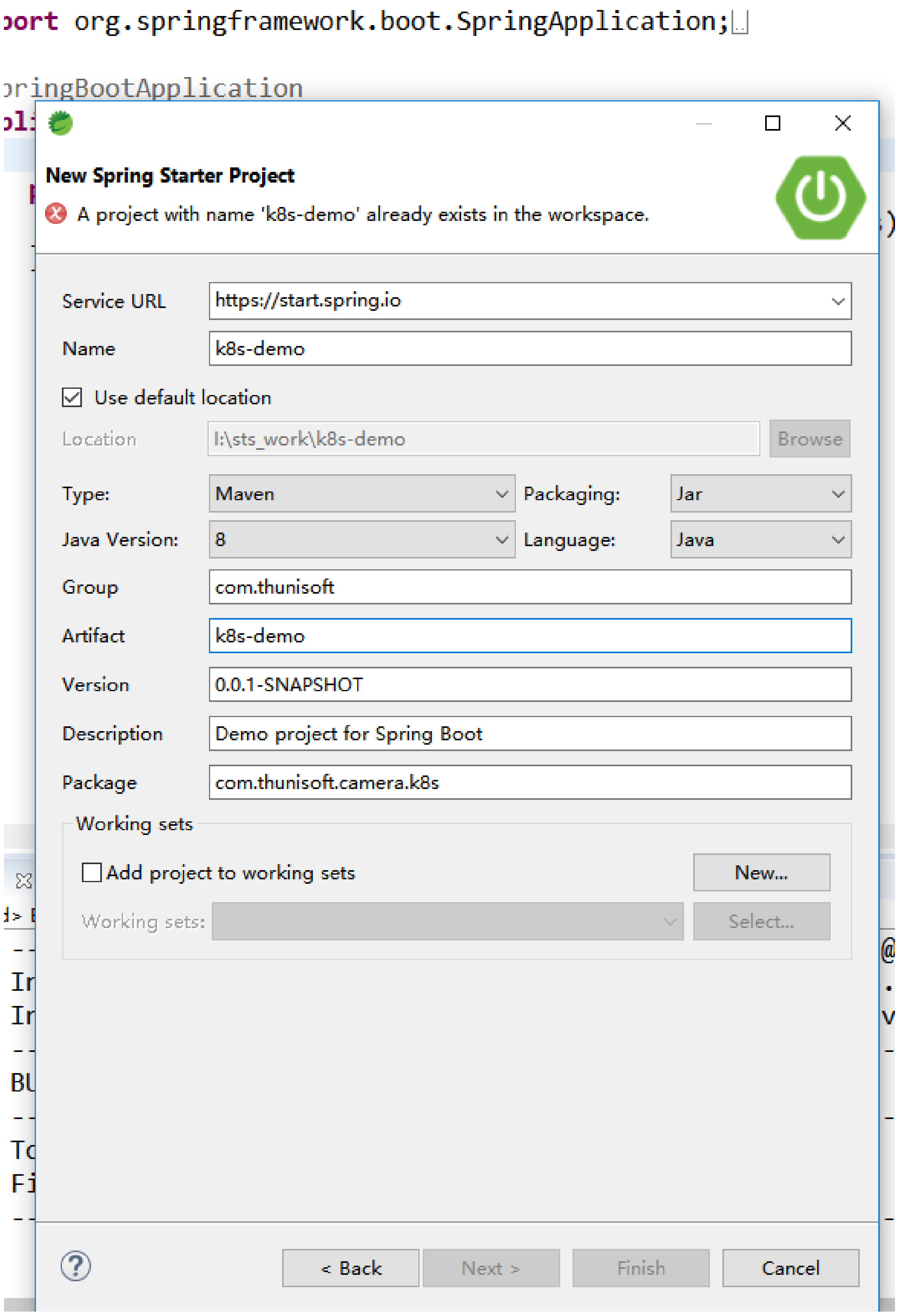
```
package com.thunisoft.camera.k8s;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
publ
                                                   args);

}
```

Select a wizard

Create new Spring Starter Project via the Spring Initializr Web Service

```
<terminated> E:\Program Files\Java\jdk1.8.0_131\bin\javaw.exe (2020年4月3日 上午10:02:35)
```

（3）创建一个SpringBoot demo项目


```
ort org.springframework.boot.SpringApplication;
ringBootApplication
```

New Spring Starter Project

A project with name 'k8s-demo' already exists in the workspace.

Service URL: https://start.spring.io

Name: k8s-demo

[✓] Use default location

Location: I:\sts_work\k8s-demo

Type: Maven

Packaging: Jar

Java Version: 8

Language: Java

Group: com.thunisoft

Artifact: k8s-demo

Version: 0.0.1-SNAPSHOT

Description: Demo project for Spring Boot

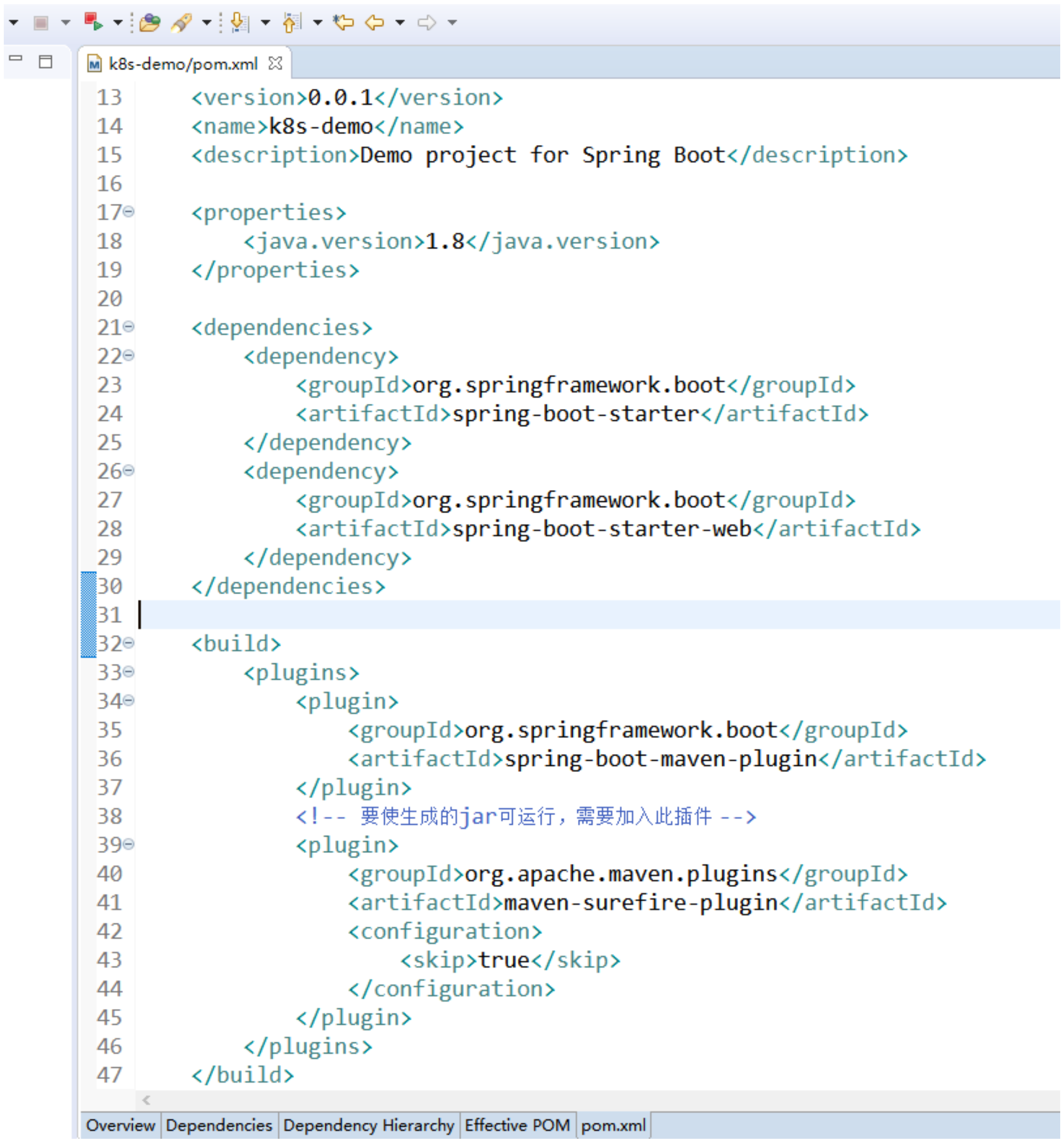
Package: com.thunisoft.camera.k8s

Working sets

[ ] Add project to working sets

Working sets:

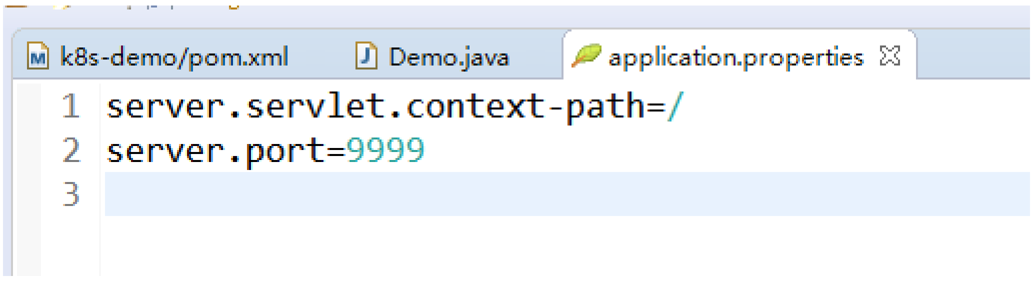
（4）pom文件增加 <`spring-boot-starter-web`>包和<`org.apache.maven.plugins`>包


k8s-demo/pom.xml

```
    <version>0.0.1</version>
    <name>k8s-demo</name>
    <description>Demo project for Spring Boot</description>

    <properties>
        <java.version>1.8</java.version>
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
            <!-- 要使生成的jar可运行，需要加入此插件 -->
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-surefire-plugin</artifactId>
                <configuration>
                    <skip>true</skip>
                </configuration>
            </plugin>
        </plugins>
    </build>
```

```
<dependencies>
          <dependency>
              <groupId>org.springframework.boot</groupId>
              <artifactId>spring-boot-starter</artifactId>
          </dependency>
          <dependency>
              <groupId>org.springframework.boot</groupId>
              <artifactId>spring-boot-starter-web</artifactId>
          </dependency>
</dependencies>
     <build>
          <plugins>
              <plugin>
                   <groupId>org.springframework.boot</groupId>
                   <artifactId>spring-boot-maven-plugin</artifactId>
              </plugin>
              <!-- 要使生成的jar可运行，需要加入此插件 -->
              <plugin>
                   <groupId>org.apache.maven.plugins</groupId>
                   <artifactId>maven-surefire-plugin</artifactId>
                   <configuration>
                        <skip>true</skip>
                   </configuration>
              </plugin>
          </plugins>
     </build>
```

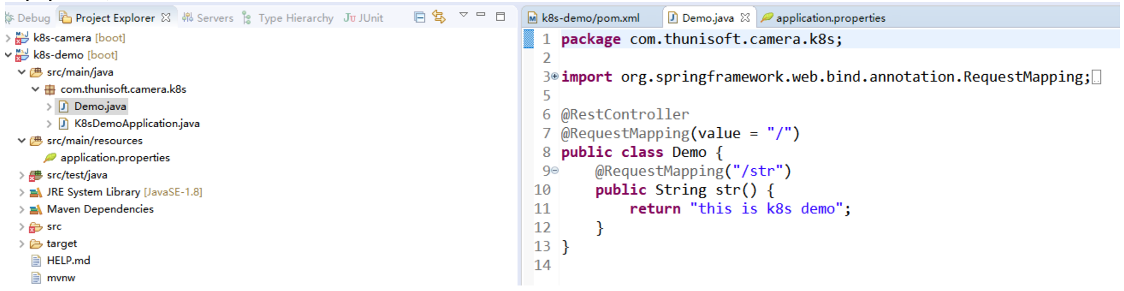
（6） 再application.propertis中配置端口号和项目地址



```
server.servlet.context-path=/
server.port=9999
```

```
server.servlet.context-path=/
server.port=9999
```

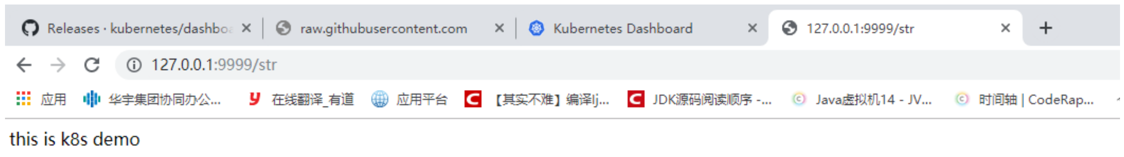
（7）在项目创建一个Demo.class文件



```
package com.thunisoft.camera.k8s;

import org.springframework.web.bind.annotation.RequestMapping;

@RestController
@RequestMapping(value = "/")
public class Demo {
    @RequestMapping("/str")
    public String str() {
        return "this is k8s demo";
    }
}
```

```
package com.thunisoft.camera.k8s;
import  org.springframework.web.bind.annotation.RequestMapping;
import  org.springframework.web.bind.annotation.RestController;
@RestController
@RequestMapping(value = "/")
public class Demo {
     @RequestMapping("/str")
     public String str() {
          return "this is k8s demo";
     }
}
```

（8）运行一下，在浏览器上访问一下

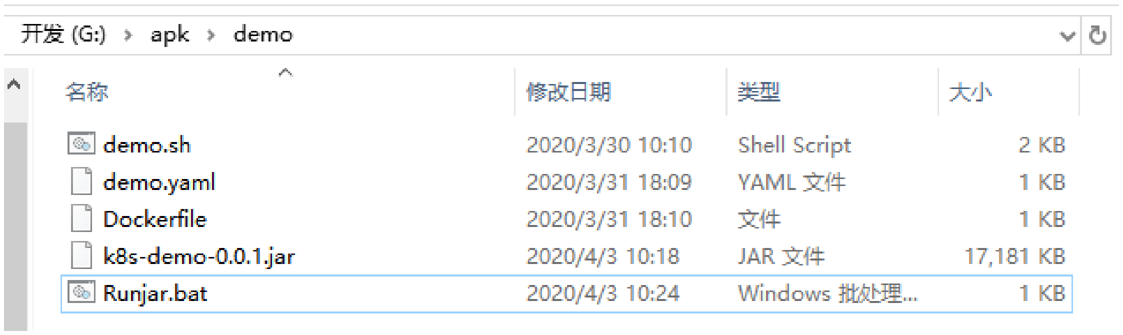


this is k8s demo

（9）项目打成jar包，选中项目【右键】、【Run as】、【4、maven clean】；
选中项目【右键】、【Run as】、【6、maven install】；

```
@ k8s-demo ---
 to E:\apache-maven-3.3.9\repository\com\thunisoft\k8s-demo\0.0.1\k8s-demo-0.0.1.jar
n-3.3.9\repository\com\thunisoft\k8s-demo\0.0.1\k8s-demo-0.0.1.pom
---------------
```

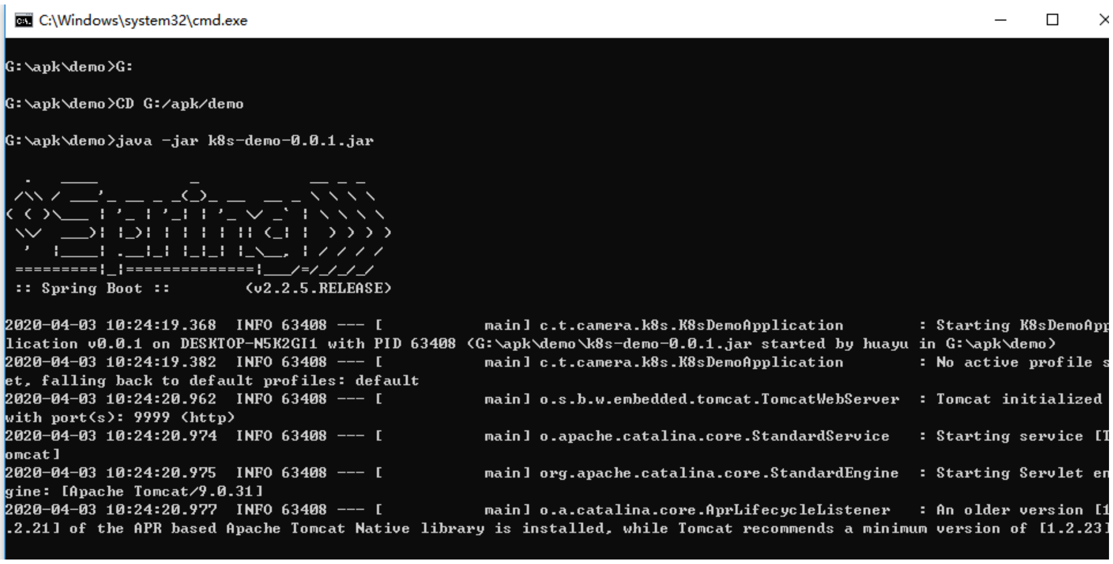
（10）在windows本地测试使用.bat文件，在centOS上运行文件时.sh文件，dockerfile是用于创建镜像的，yaml文件是执行镜像文件的。


开发 (G:) › apk › demo

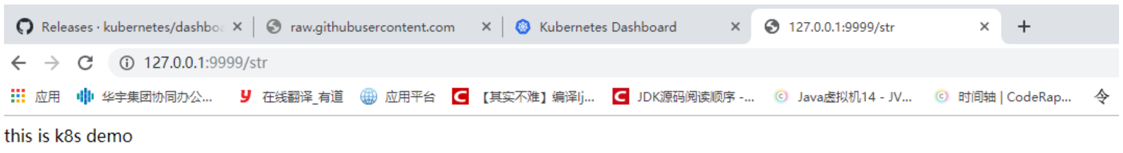
名称	修改日期	类型	大小
demo.sh	2020/3/30 10:10	Shell Script	2 KB
demo.yaml	2020/3/31 18:09	YAML 文件	1 KB
Dockerfile	2020/3/31 18:10	文件	1 KB
k8s-demo-0.0.1.jar	2020/4/3 10:18	JAR 文件	17,181 KB
Runjar.bat	2020/4/3 10:24	Windows 批处理...	1 KB

在windows上运行Runjar.bat访问浏览器.

```
G:
CD G:/apk/demo
java -jar k8s-demo-0.0.1.jar
```


C:\Windows\system32\cmd.exe

```
G:\apk\demo>G:

G:\apk\demo>CD G:/apk/demo

G:\apk\demo>java -jar k8s-demo-0.0.1.jar

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.2.5.RELEASE)

2020-04-03 10:24:19.368  INFO 63408 --- [           main] c.t.camera.k8s.K8sDemoApplication        : Starting K8sDemoApp
lication v0.0.1 on DESKTOP-N5K2GI1 with PID 63408 (G:\apk\demo\k8s-demo-0.0.1.jar started by huayu in G:\apk\demo)
2020-04-03 10:24:19.382  INFO 63408 --- [           main] c.t.camera.k8s.K8sDemoApplication        : No active profile s
et, falling back to default profiles: default
2020-04-03 10:24:20.962  INFO 63408 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized
with port(s): 9999 (http)
2020-04-03 10:24:20.974  INFO 63408 --- [           main] o.apache.catalina.core.StandardService   : Starting service [T
omcat]
2020-04-03 10:24:20.975  INFO 63408 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet en
gine: [Apache Tomcat/9.0.31]
2020-04-03 10:24:20.977  INFO 63408 --- [           main] o.a.catalina.core.AprLifecycleListener   : An older version [1
.2.21] of the APR based Apache Tomcat Native library is installed, while Tomcat recommends a minimum version of [1.2.23]
```



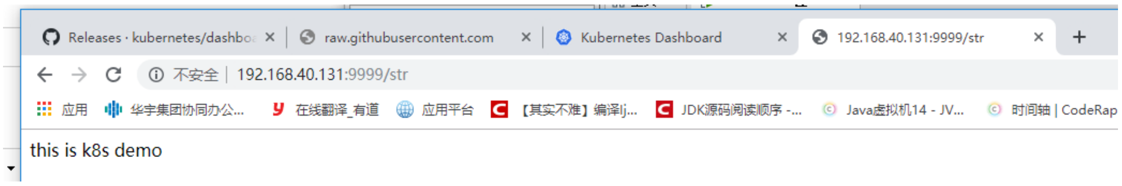
this is k8s demo

(11) 在centOS上创建一个demo文件夹.将jar包放到demo文件夹中,
创建一个demo.sh文件,编辑demo.sh文件,将下方内容复制到demo.sh文件中,红色文字代表项目jar包,启动方式./demo.sh

```
#!/bin/bash
DIR="$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )"
JAVA_OPT="-Xmx4000m"
##执行的应用
APP_NAME=k8s-demo-0.0.1.jar
#使用说明，用来提示输入参数
usage() {
    echo "Usage: sh exchange.sh [start|stop|restart|status]"
    exit 1
}
#检查程序是否在运行
is_exist(){
  pid=`ps -ef|grep $APP_NAME|grep java |grep -v grep|awk '{print $2}'`
  #如果不存在返回1，存在返回0
  if [ -z "${pid}" ]; then
   return 1
  else
    return 0
  fi
}
#启动方法
start(){
  is_exist
  if [ $? -eq 0 ]; then
    echo "${APP_NAME} is already running. pid=${pid}"
  else
    nohup java ${JAVA_OPT} -jar ${APP_NAME} &
  fi
}
#停止方法
stop(){
  is_exist
  if [ $? -eq "0" ]; then
    kill -9 $pid
    echo "${APP_NAME} has stopped successfully"
  else
    echo "${APP_NAME} is not running"
  fi
}
#输出运行状态
status(){
  is_exist
  if [ $? -eq "0" ]; then
    echo "${APP_NAME} is running. Pid is ${pid}"
  else
    echo "${APP_NAME} is NOT running."
  fi
}
#重启
restart(){
  stop
  sleep 5
  start
}
#根据输入参数，选择执行对应方法，不输入则执行使用说明
case "$1" in
  "start")
    start
    ;;
  "stop")
    stop
    ;;
  "status")
    status
    ;;
  "restart")
    restart
    ;;
  *)
    usage
    ;;
esac
```

(12) 设置文件权限,并运行文件,在浏览器上访问,记住我们设置的项目端口号.访问的是虚拟机IP

```
chmod +x demo.sh
然后运行文件
./demo.sh start
```


```
Releases · kubernetes/dashbo  ×   raw.githubusercontent.com  ×   Kubernetes Dashboard  ×   192.168.40.131:9999/str  ×
不安全 | 192.168.40.131:9999/str
应用  华宇集团协同办公...  在线翻译_有道  应用平台  【其实不难】编译j...  JDK源码阅读顺序 -...  Java虚拟机14 - JV...  时间轴 | CodeRap

this is k8s demo
```

(13)关闭项目

```
[root@master demo]# ./demo.sh stop
k8s-demo-0.0.1.jar has stopped successfully
```

(14)修改Dockerfile,先创建一个Dockerfile文件,放到虚拟机demo文件夹内

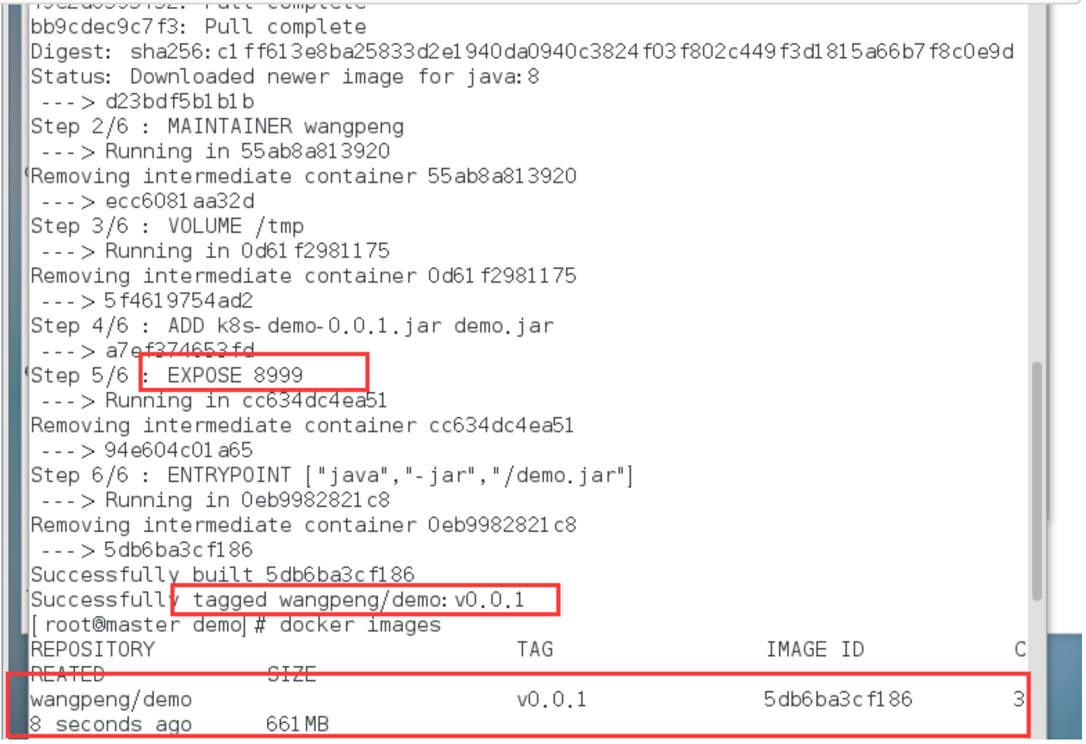
```
#基础镜像jdk
FROM java:8
#指定维护者信息
MAINTAINER wangpeng
#挂载的路径
VOLUME /tmp
#将jar打入镜像之中
ADD k8s-demo-0.0.1.jar demo.jar
#容器向外暴露的端口  此端口没用
#EXPOSE 8999
#入口命令，执行jar
ENTRYPOINT ["java","-jar","/demo.jar"]
```

(15) 通过docker命令创建镜像.docker build 创建的意思 -t (tag)的意思 打成镜像名称wangpeng/demo, 版本号 v0.0.1 后面的. 代表当前目录上.

```
docker build -t wangpeng/demo:v0.0.1 .
```

(16)通过docker命令查看镜像

```
docker iamges
```


```
bb9cdec9c7f3: Pull complete
Digest: sha256:c1ff613e8ba25833d2e1940da0940c3824f03f802c449f3d1815a66b7f8c0e9d
Status: Downloaded newer image for java:8
 ---> d23bdf5b1b1b
Step 2/6 : MAINTAINER wangpeng
 ---> Running in 55ab8a813920
Removing intermediate container 55ab8a813920
 ---> ecc6081aa32d
Step 3/6 : VOLUME /tmp
 ---> Running in 0d61f2981175
Removing intermediate container 0d61f2981175
 ---> 5f4619754ad2
Step 4/6 : ADD k8s-demo-0.0.1.jar demo.jar
 ---> a7ef374653fd
Step 5/6 : EXPOSE 8999
 ---> Running in cc634dc4ea51
Removing intermediate container cc634dc4ea51
 ---> 94e604c01a65
Step 6/6 : ENTRYPOINT ["java","-jar","/demo.jar"]
 ---> Running in 0eb9982821c8
Removing intermediate container 0eb9982821c8
 ---> 5db6ba3cf186
Successfully built 5db6ba3cf186
Successfully tagged wangpeng/demo:v0.0.1
[root@master demo]# docker images
REPOSITORY                          TAG                 IMAGE ID            C
REATED              SIZE
wangpeng/demo                       v0.0.1              5db6ba3cf186        3
8 seconds ago       661MB
```

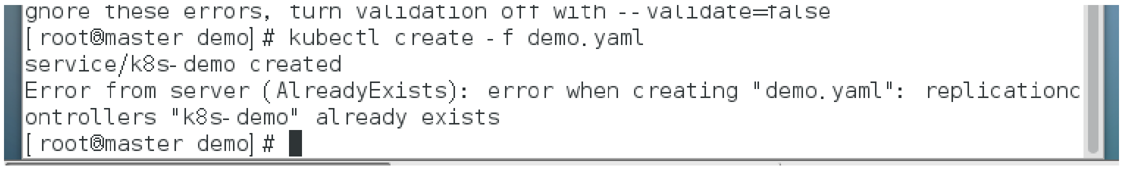
成功生成一个wangpeng/demo:v0.0.1版本的镜像

(17) 编写yaml文件,创建一个demo.yaml文件.注意yml文件格式,层级结构

```
apiVersion: v1
kind: ReplicationController
metadata:
name: k8s-demo
spec:
replicas: 1
selector:
   app: k8s-demo
template:
   metadata:
     labels:
       app: k8s-demo
   spec:
     containers:
     - name: cipher
       #选择镜像文件名称
       image: wangpeng/demo:v0.0.1
       #默认在本机找镜像
       imagePullPolicy: IfNotPresent
---
apiVersion: v1
kind: Service
metadata:
name: k8s-demo
spec:
#使用NodePort端口
type: NodePort
ports:
  #原来项目设置的端口
- port: 9999
   targetPort: 9999
   #设置访问端口为30100 ,可以自己设置
   nodePort: 30100
selector:
   app: k8s-demo
```

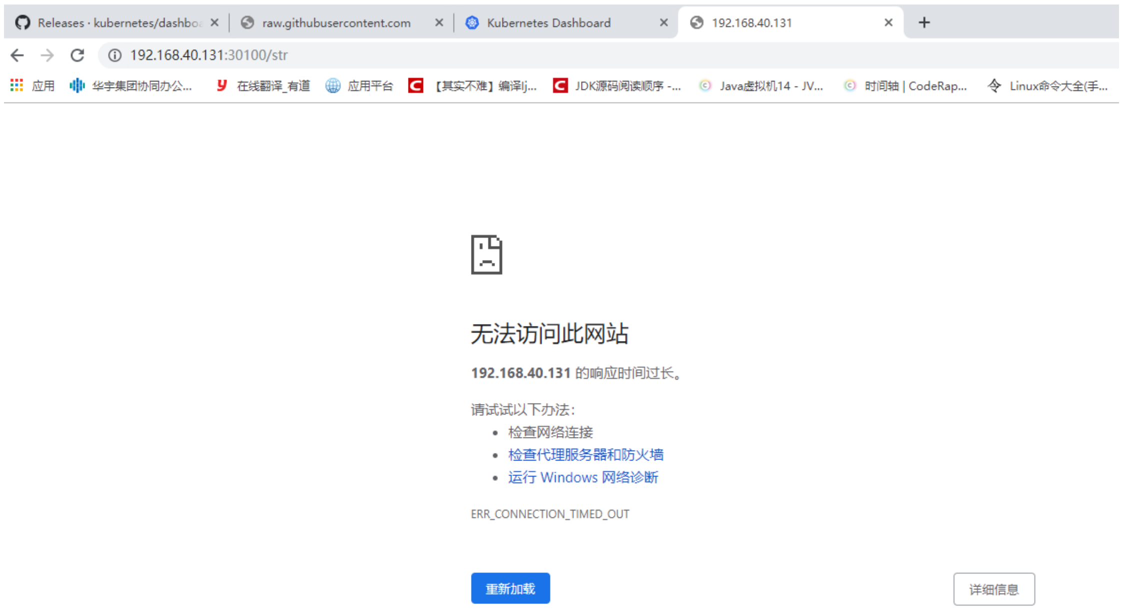
(18)通过docker命令运行yaml文件

```
kubectl create -f demo.yaml
```



```
gnore these errors, turn validation off with --validate=false
[root@master demo]# kubectl create -f demo.yaml
service/k8s-demo created
Error from server (AlreadyExists): error when creating "demo.yaml": replicationc
ontrollers "k8s-demo" already exists
[root@master demo]#
```

访问项目时,无法找到



无法访问此网站

192.168.40.131 的响应时间过长。

请试试以下办法：
- 检查网络连接
- 检查代理服务器和防火墙
- 运行 Windows 网络诊断

ERR_CONNECTION_TIMED_OUT

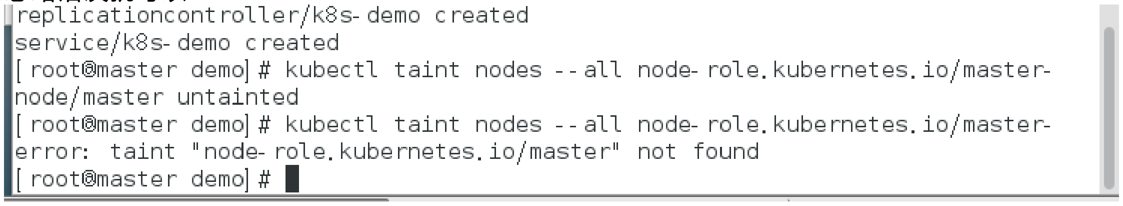
并且项目的demo,pode状态为Pending时,执行以下内容

```
kubectl taint nodes --all node-role.kubernetes.io/master-
```

(19) 是否允许master节点上部署pod

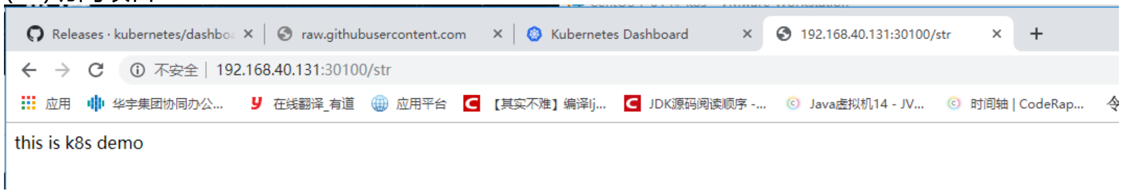
```
允许master节点部署pod
kubectl taint nodes --all node-role.kubernetes.io/master-
如果不允许调度
kubectl taint nodes master1 node-role.kubernetes.io/master=:NoSchedule
污点可选参数
      NoSchedule: 一定不能被调度
      PreferNoSchedule: 尽量不要调度
      NoExecute: 不仅不会调度, 还会驱逐Node上已有的Pod
```

忽略错误就可以



```
replicationcontroller/k8s-demo created
service/k8s-demo created
[root@master demo]# kubectl taint nodes --all node-role.kubernetes.io/master-
node/master untainted
[root@master demo]# kubectl taint nodes --all node-role.kubernetes.io/master-
error: taint "node-role.kubernetes.io/master" not found
[root@master demo]# 
```

(20)访问项目



this is k8s demo

完毕