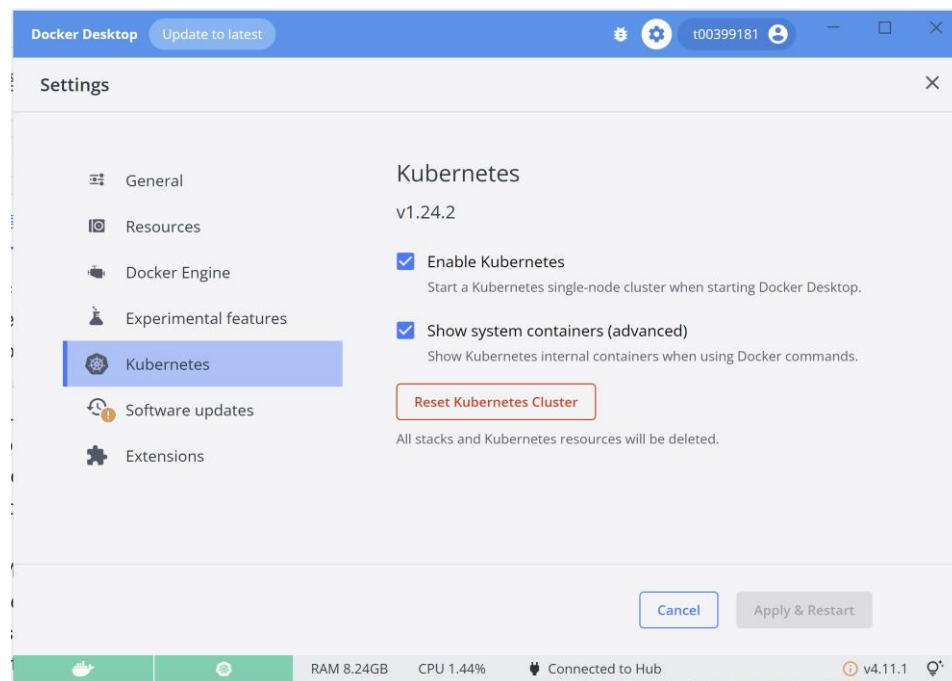


k8s使用

陶召胜

安装

- 请访问链接: <https://github.com/AliyunContainerService/k8s-for-docker-desktop>, 根据说明安装



下载镜像

- 国内下载，然后更名

部署dashboard

- `kubectl apply -f kubernetes-dashboard.yaml`
- 检查 kubernetes-dashboard 应用状态
 - ✓ `kubectl get pod -n kubernetes-dashboard`
- 将k8s server代理到本地端口: `kubectl proxy`
- 配置控制台访问令牌

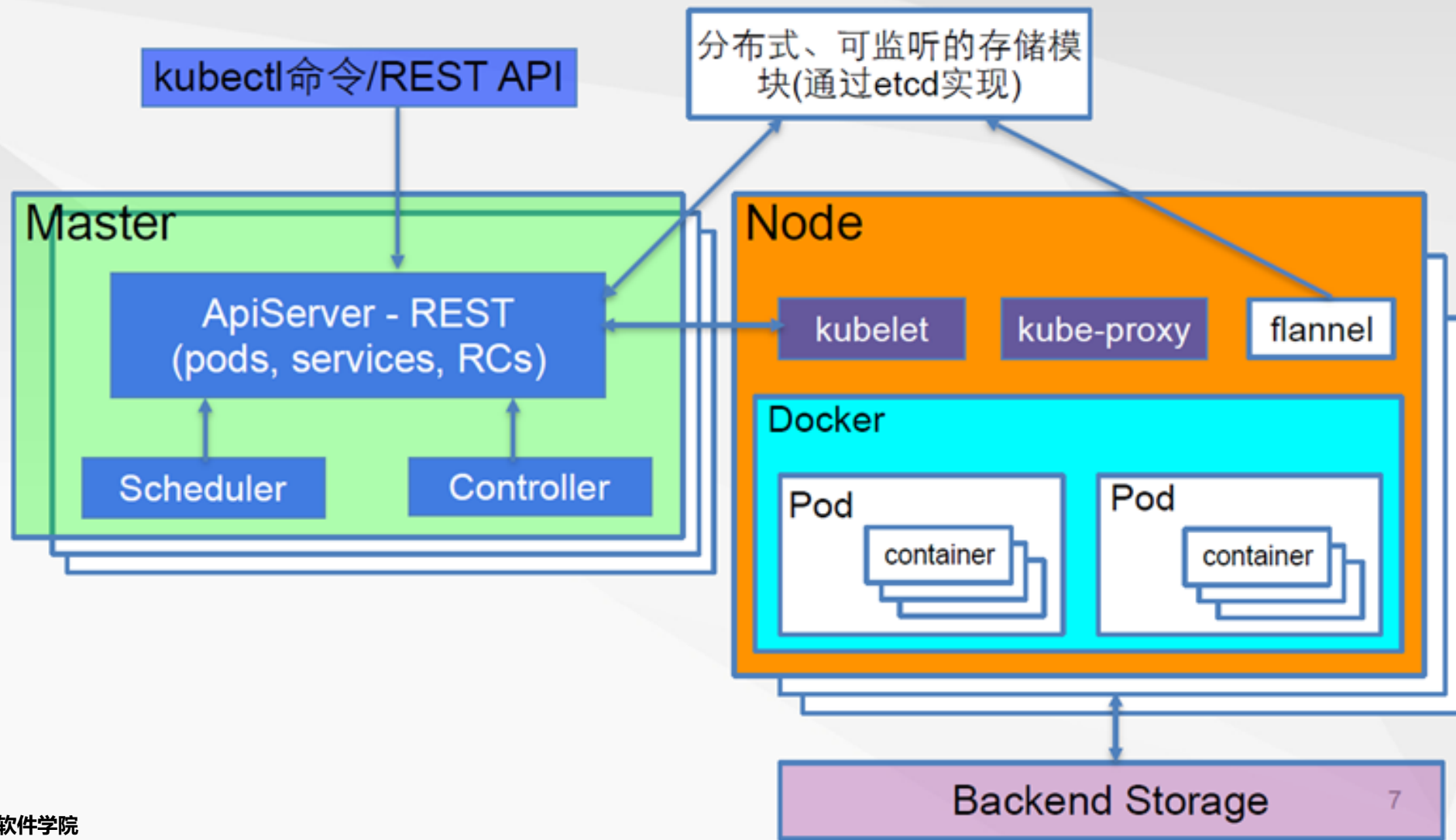
启动ingress controller

- `kubectl apply -f ingress-nginx-controller.yaml`

验证 Kubernetes 集群状态

- `kubectl cluster-info`
- `kubectl get nodes`
- `kubectl get nodes --show-labels`
- 给节点打标签: `kubectl label node docker-desktop disktype=ssd`
 - ✓ 查询满足标签的节点: `kubectl get nodes -l disktype=ssd`
 - ✓ 删除标签: `kubectl label node docker-desktop disktype-`

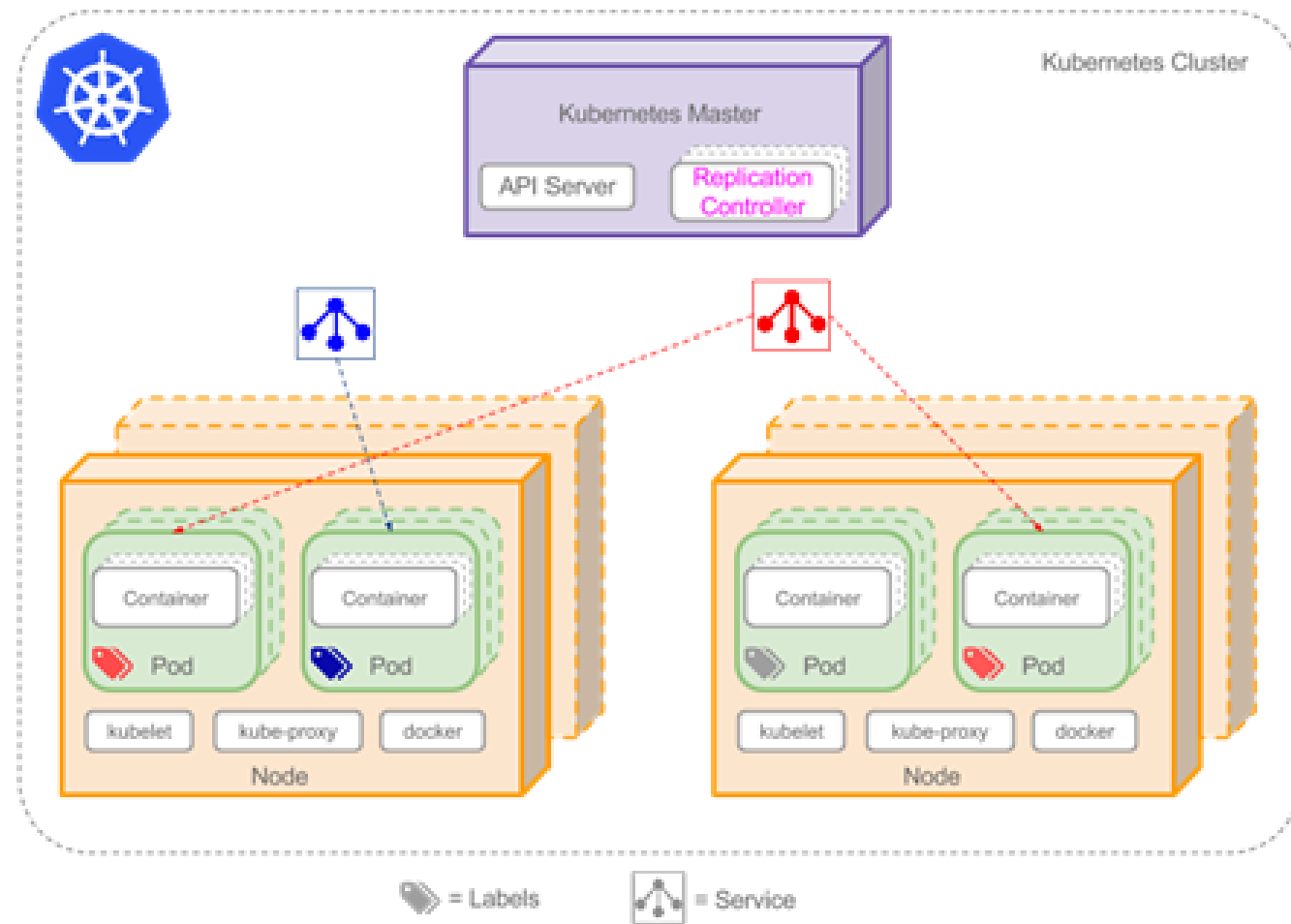
Kubernetes 基本架构



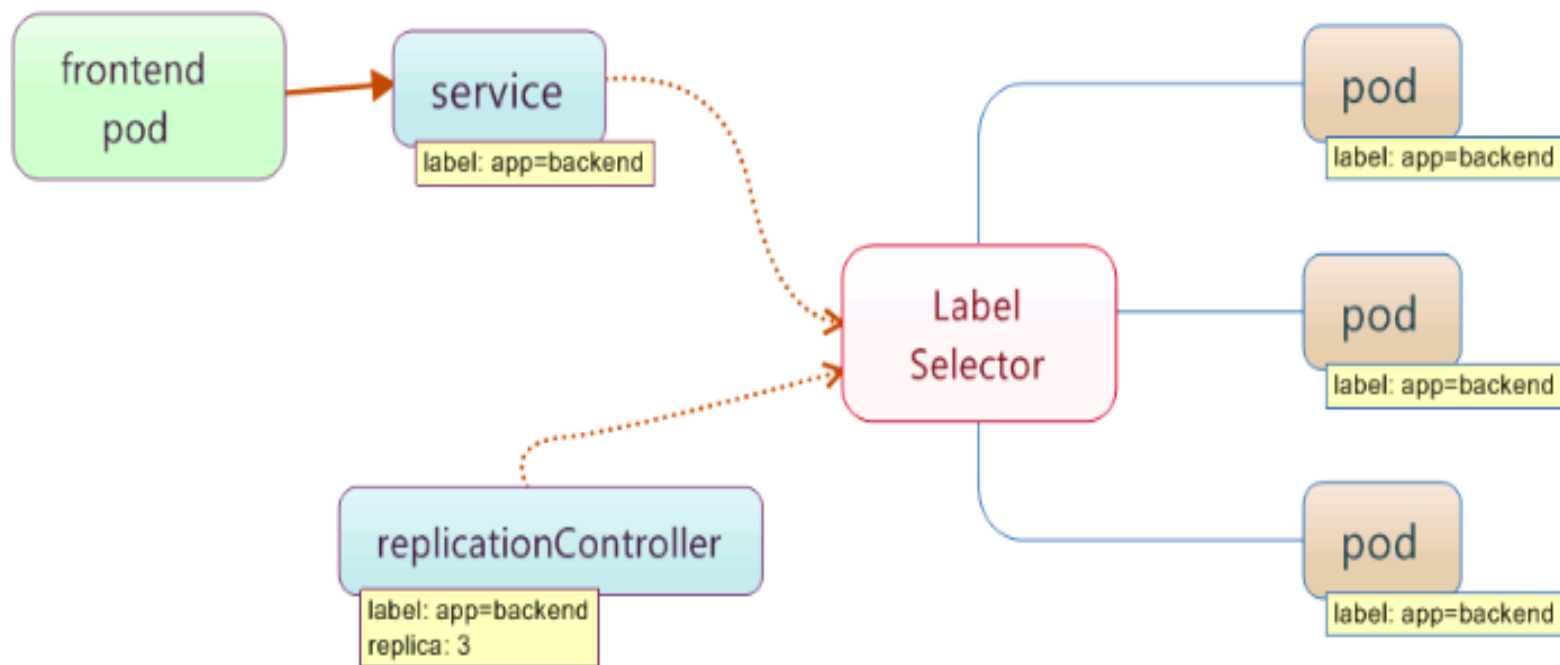
k8s中的资源

- namespaces
- Pods
- ReplicaSet
- Deployment
- Service
- Ingress
- configmap
- secrets
- serviceaccounts
- DaemonSet
-

核心概念



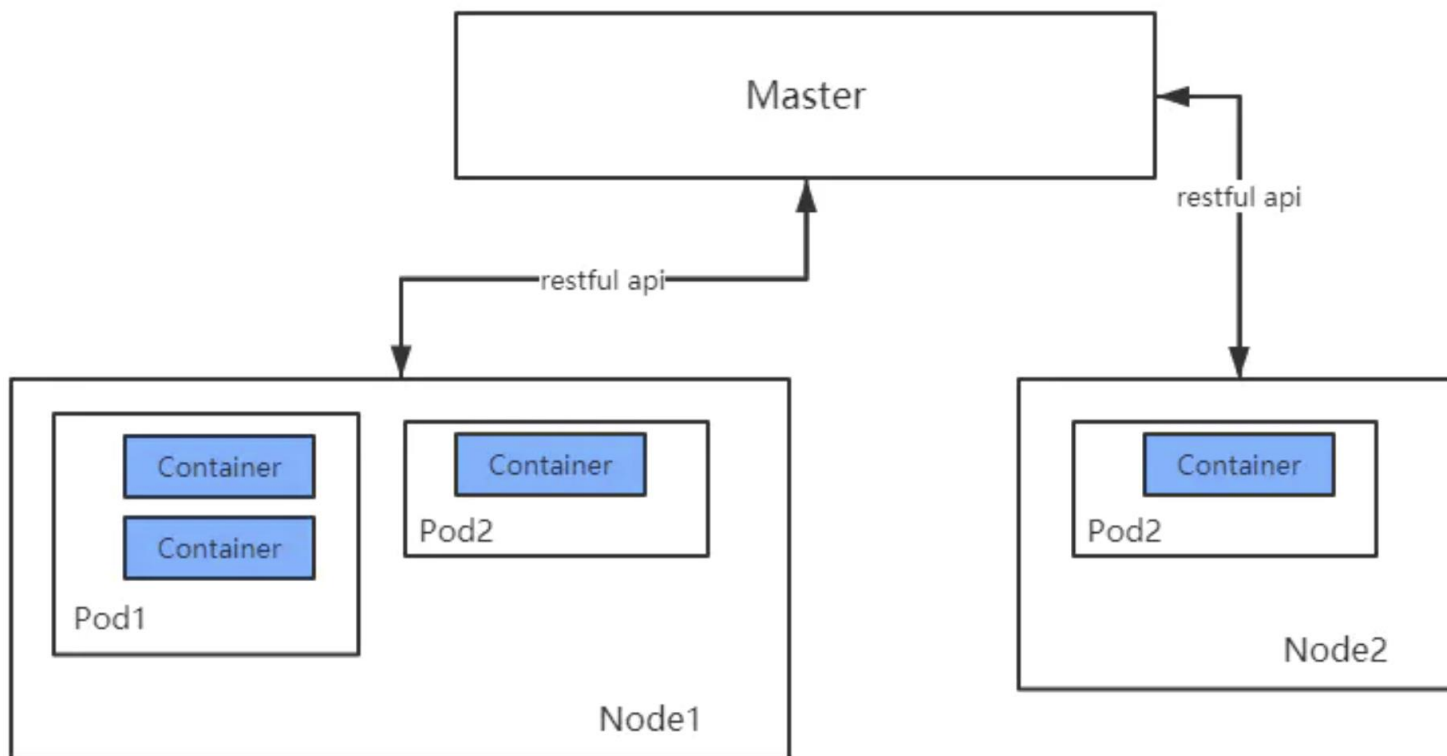
Label



Pod

- 学习链接: <https://www.jianshu.com/p/ce71385e0370>
- Pod是Kubernetes调度的最小单元
- 一个Pod可以包含一个或多个容器, 因此它可以被看作是内部容器的逻辑宿主机。Pod的设计理念是为了支持多个容器在一个Pod中共享网络和文件系统
 - ✓ PID命名空间: Pod中不同的应用程序可以看到其他应用程序的进程ID
 - ✓ network命名空间: Pod中多个容器处于同一个网络命名空间, 因此能够访问的IP和端口范围都是相同的。也可以通过localhost相互访问
 - ✓ IPC命名空间: Pod中的多个容器共享Inner-process Communication命名空间, 因此可以通过SystemV IPC或POSIX进行进程间通信
 - ✓ UTS命名空间: Pod中的多个容器共享同一个主机名
 - ✓ Volumes: Pod中各个容器可以共享在Pod中定义分存储卷 (Volume)
- restartPolicy字段
 - ✓ Always: 只要退出就重启
 - ✓ OnFailure: 失败退出时 (exit code不为0) 才重启
 - ✓ Never: 永远不重启

Pod、Container与Node之间的关系



创建一个pod

- `kubectl run myspittr --image spittr:1.0-SNAPSHOT`
- `kubectl get pods`
- `kubectl logs -f myspittr`
- pod中执行一个命令
 - ✓ `kubectl exec myspittr -- ls /run/secrets/kubernetes.io/serviceaccount`
- `kubectl delete pod myspittr`

如何将pod或服务端口快速映射到本机端口（调试用）

- `kubectl port-forward pod/myspittr 8081:8080`
 - ✓ 访问: `http://localhost:8081/spittr/`
- `kubectl port-forward service/demo 8081:80`

创建服务

- `kubectl expose pod myspittr --port 8080`
- `kubectl get svc`
- `kubectl delete service myspittr`
- #指定镜像，指定暴露端口，会同时创建同名的service
 - ✓ `kubectl run myspittr --image=spittr:1.0-SNAPSHOT --expose --port=8080`

创建ingress

- `kubectl create ingress myspittr --class=nginx --rule=www.license.com/*=myspittr:8080`
 - ✓ 访问: `http://www.license.com/spittr/`
- `kubectl delete ingress myspittr`

如何使用命令行快捷创建deployment、service、ingress

- `kubectl create deployment myspittr --image=spittr:1.0-SNAPSHOT --port=8080`
- `kubectl expose deployment myspittr` 把上面的这个pod的8080端口暴露为myspittr服务
- `kubectl create ingress myspittr --class=nginx --rule=www.license.com/*=myspittr:8080`
- 访问: `http://www.license.com/spittr/`
- 删除:
 - ✓ `kubectl delete ingress myspittr`
 - ✓ `kubectl delete service myspittr`
 - ✓ `kubectl delete deployment myspittr`

使用yaml文件部署

- `kubectl create -f k8s-deploy.yaml`
- `kubectl delete -f k8s-deploy.yaml`

Deployment

- 学习链接: <https://www.jianshu.com/p/029661f38674>
- 重启: `kubectl rollout restart deployment/spittr`
 - ✓ `kubectl rollout restart deployment --selector=app=spittr`
- 更新镜像重部署: `kubectl set image deployment/spittr spittr=spittr:1.0`
- 扩容: `kubectl scale deployment spittr --replicas 2`
- 自动: `kubectl autoscale deployment spittr --min=10 --max=15 --cpu-percent=80`
- 查看历史版本: `kubectl rollout history deployment/spittr`
- 回滚到前一个版本: `kubectl rollout undo deployment/spittr`

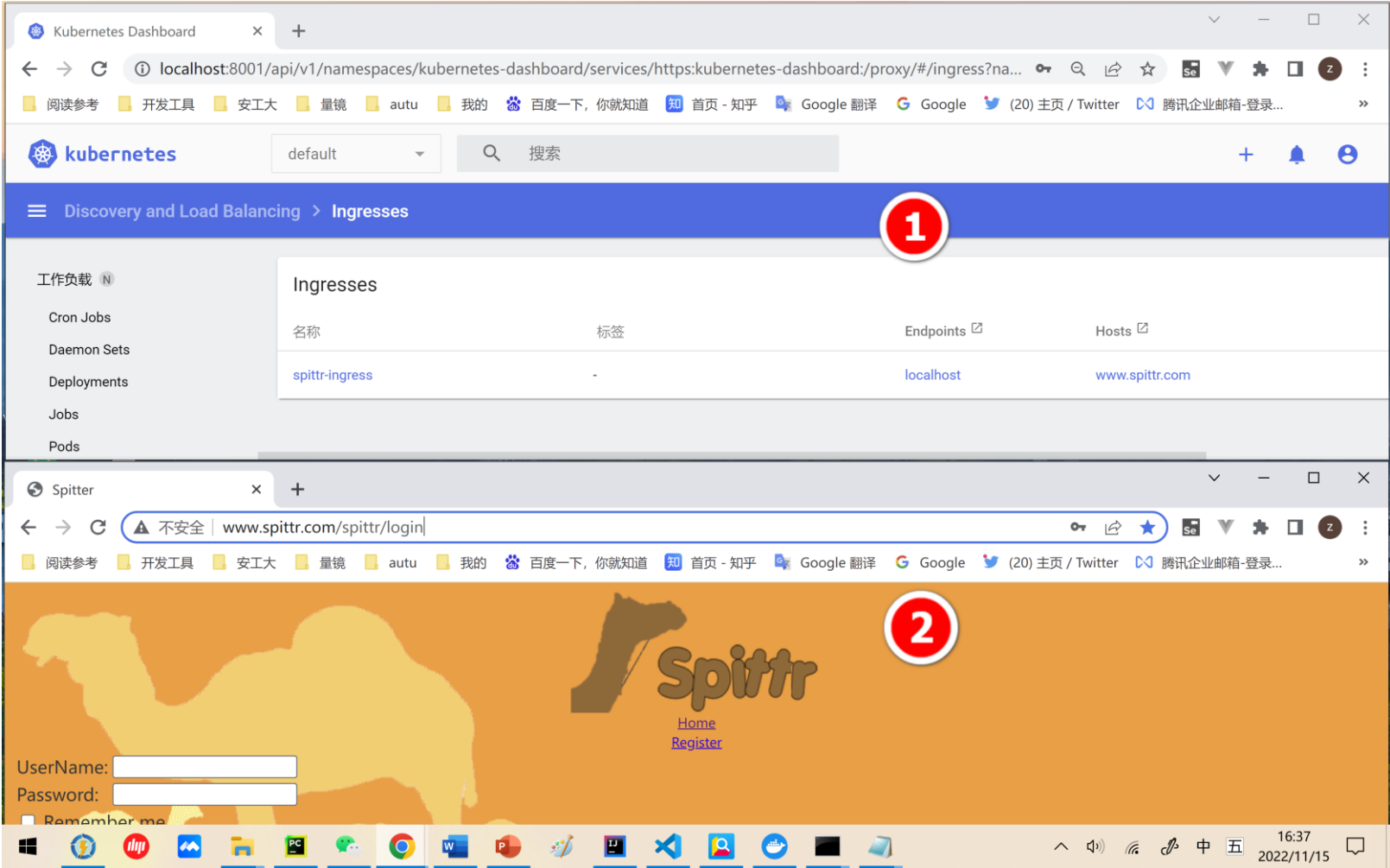
k8s常用命令

- `kubectl get secrets/pods/all [-n namespace]`
- `kubectl get secret mysecret -o yaml`
- `kubectl delete pod pod_name [-n namespace]`
- `kubectl apply -f [json文件或yaml文件-路径]`
- `kubectl delete -f [json文件或yaml文件-路径]`
- `kubectl describe secret mysecret`
- `kubectl logs secret1-pod`
- pod
- deployment
- service
- configmap
- daemonset
- secret
-

练习

- 代码下载
 - ✓ <https://github.com/tzs919/spittr-deploy.git>
- 构建本地镜像：
 - ✓ `mvn clean package -Dmaven.test.skip=true`
 - ✓ `mvn docker:build`
- 部署
 - ✓ `kubectl create -f k8s-deploy.yaml`
- 访问
 - ✓ `http://www.spittr.com/spittr/login`
 - ✓ 用户名: tzs919 密码: 123456
 - ✓ 注意需要在本机的hosts文件配置域名 (C:\Windows\System32\drivers\etc\hosts或/etc/hosts)
 - ✓ 127.0.0.1 www.spittr.com
- 删除
 - ✓ `kubectl delete -f k8s-deploy.yaml`

运行截图



谢谢观看！

