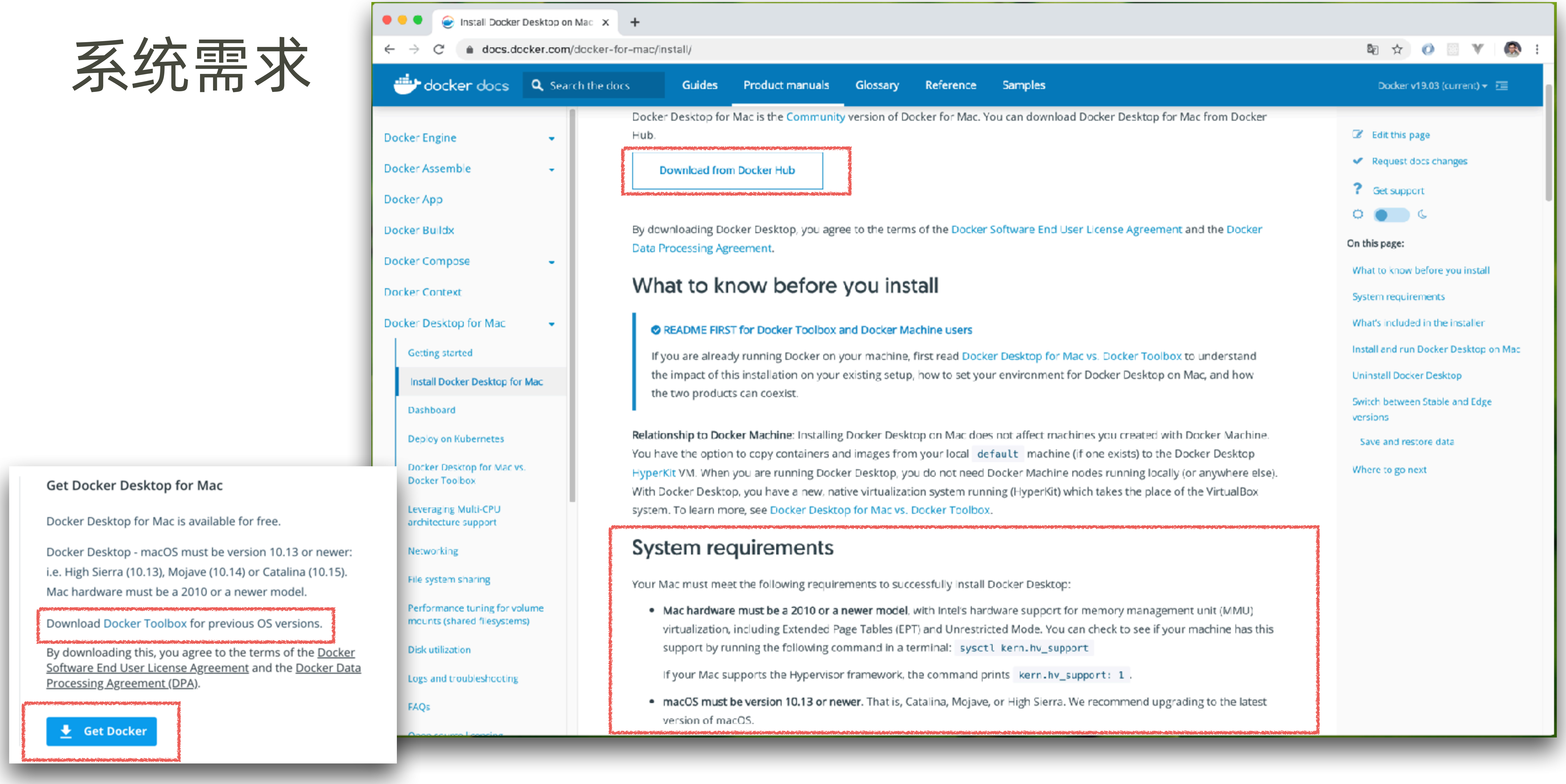
第 1 节

Docker Desktop for Mac安装

本课内容

- 演示本地安装Docker Desktop for macOS Mojave
- 演示启用Docker Desktop中的K8s

系统需求



Docker Desktop for Mac is the Community version of Docker for Mac. You can download Docker Desktop for Mac from Docker Hub.

Download from Docker Hub

By downloading Docker Desktop, you agree to the terms of the Docker Software End User License Agreement and the Docker Data Processing Agreement.

What to know before you install

README FIRST for Docker Toolbox and Docker Machine users

If you are already running Docker on your machine, first read Docker Desktop for Mac vs. Docker Toolbox to understand the impact of this installation on your existing setup, how to set your environment for Docker Desktop on Mac, and how the two products can coexist.

Relationship to Docker Machine: Installing Docker Desktop on Mac does not affect machines you created with Docker Machine. You have the option to copy containers and images from your local `default` machine (if one exists) to the Docker Desktop HyperKit VM. When you are running Docker Desktop, you do not need Docker Machine nodes running locally (or anywhere else). With Docker Desktop, you have a new, native virtualization system running (HyperKit) which takes the place of the VirtualBox system. To learn more, see Docker Desktop for Mac vs. Docker Toolbox.

System requirements

Your Mac must meet the following requirements to successfully install Docker Desktop:

- **Mac hardware must be a 2010 or a newer model**, with Intel's hardware support for memory management unit (MMU) virtualization, including Extended Page Tables (EPT) and Unrestricted Mode. You can check to see if your machine has this support by running the following command in a terminal: `sysctl kern.hv_support`

  If your Mac supports the Hypervisor framework, the command prints `kern.hv_support: 1`.
- **macOS must be version 10.13 or newer.** That is, Catalina, Mojave, or High Sierra. We recommend upgrading to the latest version of macOS.

Get Docker Desktop for Mac

Docker Desktop for Mac is available for free.

Docker Desktop - macOS must be version 10.13 or newer: i.e. High Sierra (10.13), Mojave (10.14) or Catalina (10.15). Mac hardware must be a 2010 or a newer model.

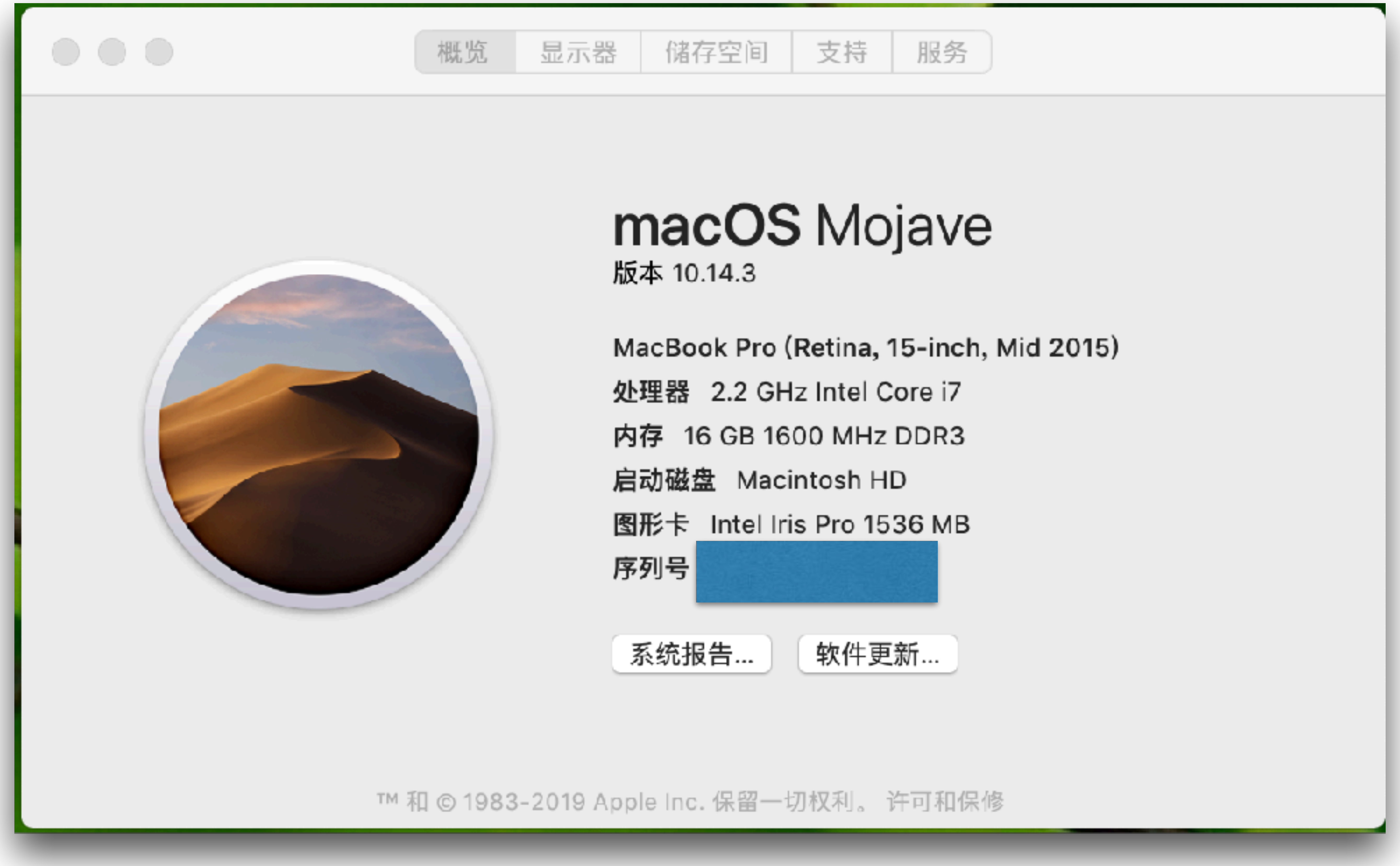
Download Docker Toolbox for previous OS versions.

By downloading this, you agree to the terms of the Docker Software End User License Agreement and the Docker Data Processing Agreement (DPA).

Get Docker

https://docs.docker.com/docker-for-mac/install/

macOS系统版本

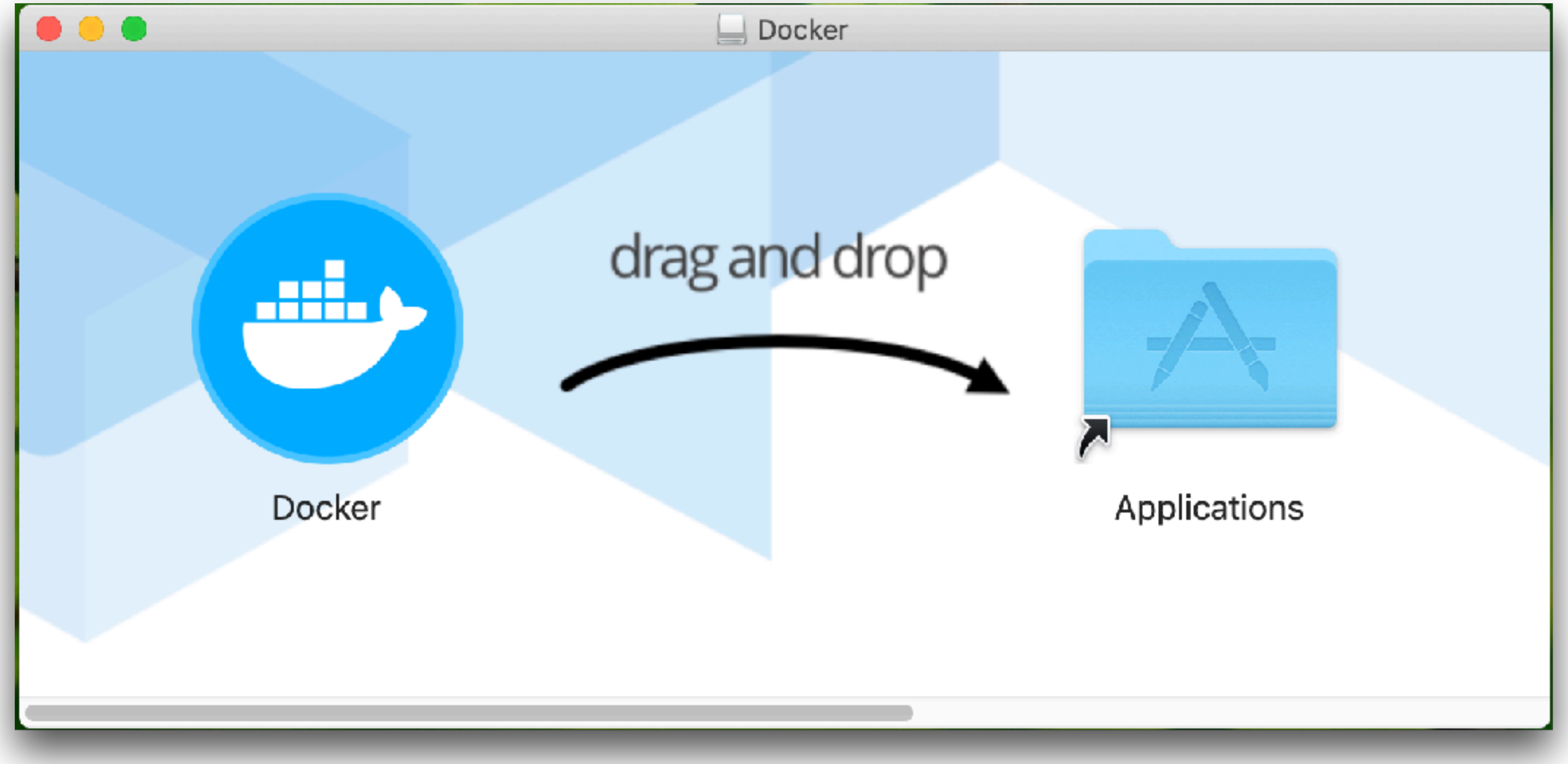


macOS Mojave
版本 10.14.3

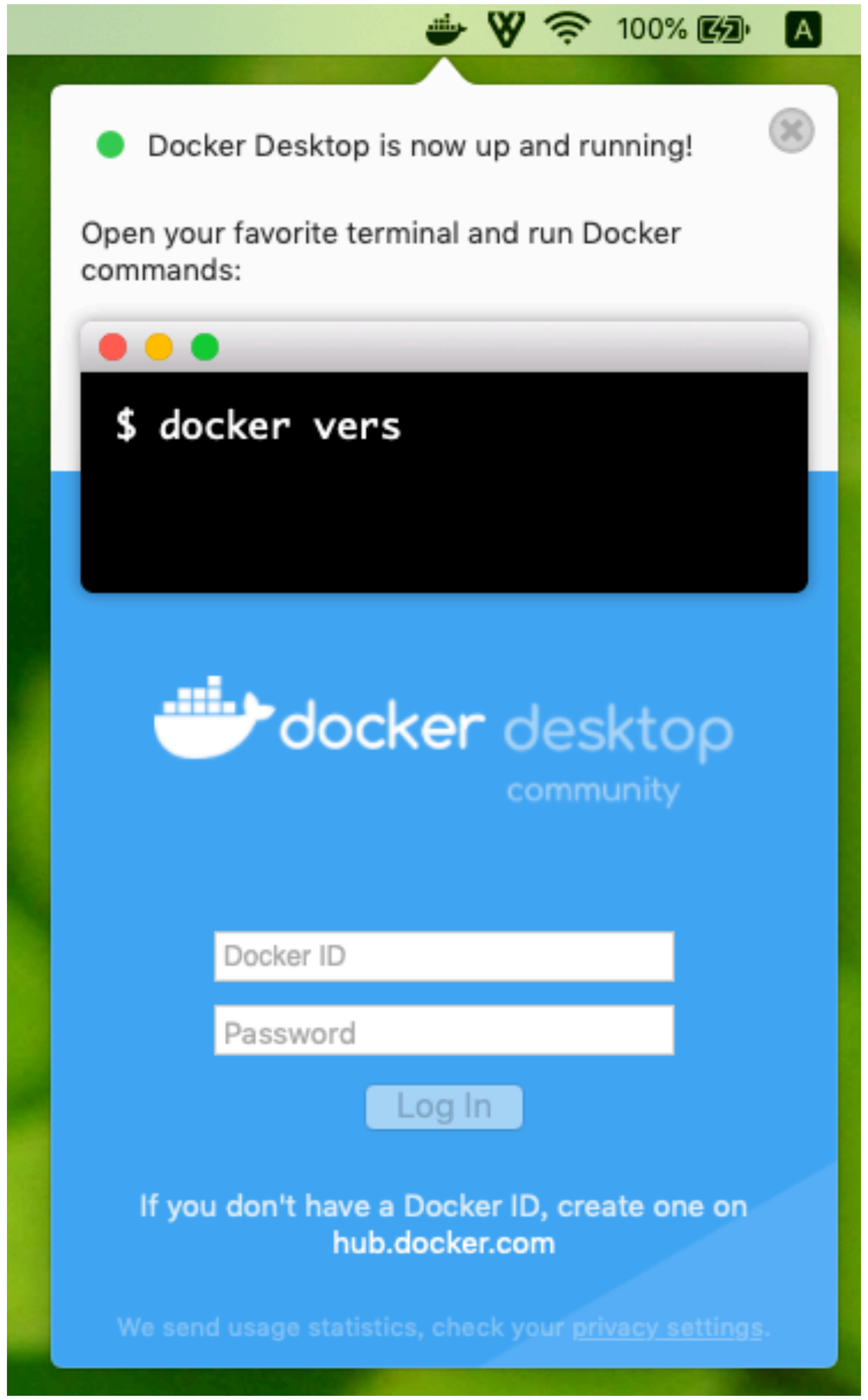
MacBook Pro (Retina, 15-inch, Mid 2015)
处理器 2.2 GHz Intel Core i7
内存 16 GB 1600 MHz DDR3
启动磁盘 Macintosh HD
图形卡 Intel Iris Pro 1536 MB
序列号 [redacted]

™ 和 © 1983-2019 Apple Inc. 保留一切权利。 许可和保修

安装Docker Desktop for Mac



成功安装



Docker Desktop is now up and running!

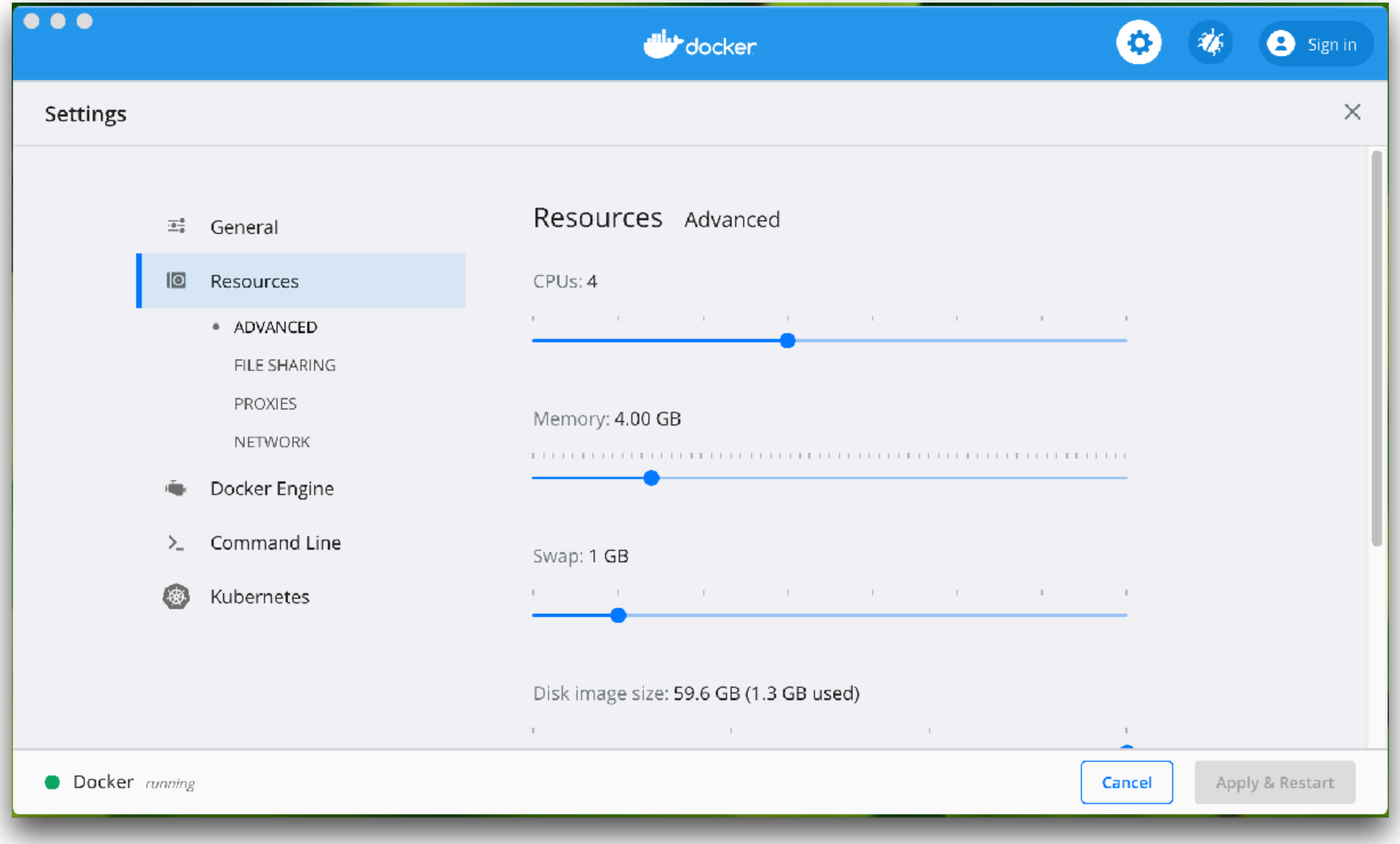
Open your favorite terminal and run Docker commands:

```
$ docker vers
```

If you don't have a Docker ID, create one on hub.docker.com

We send usage statistics, check your privacy settings.

容器资源设置



CPUs: 4

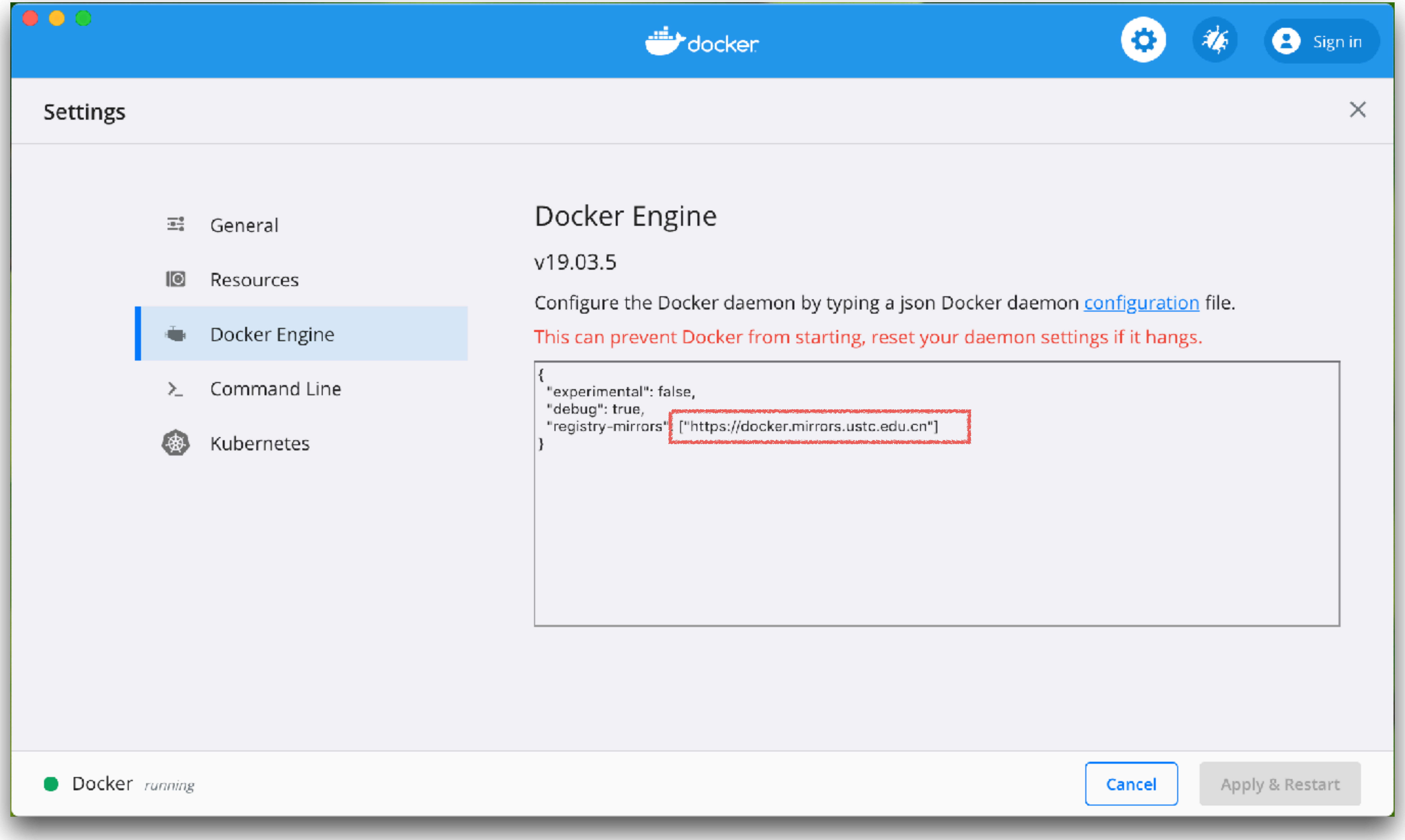
Memory: 4.00 GB

Swap: 1 GB

Disk image size: 59.6 GB (1.3 GB used)

Docker *running*

镜像加速器设置



Docker Engine

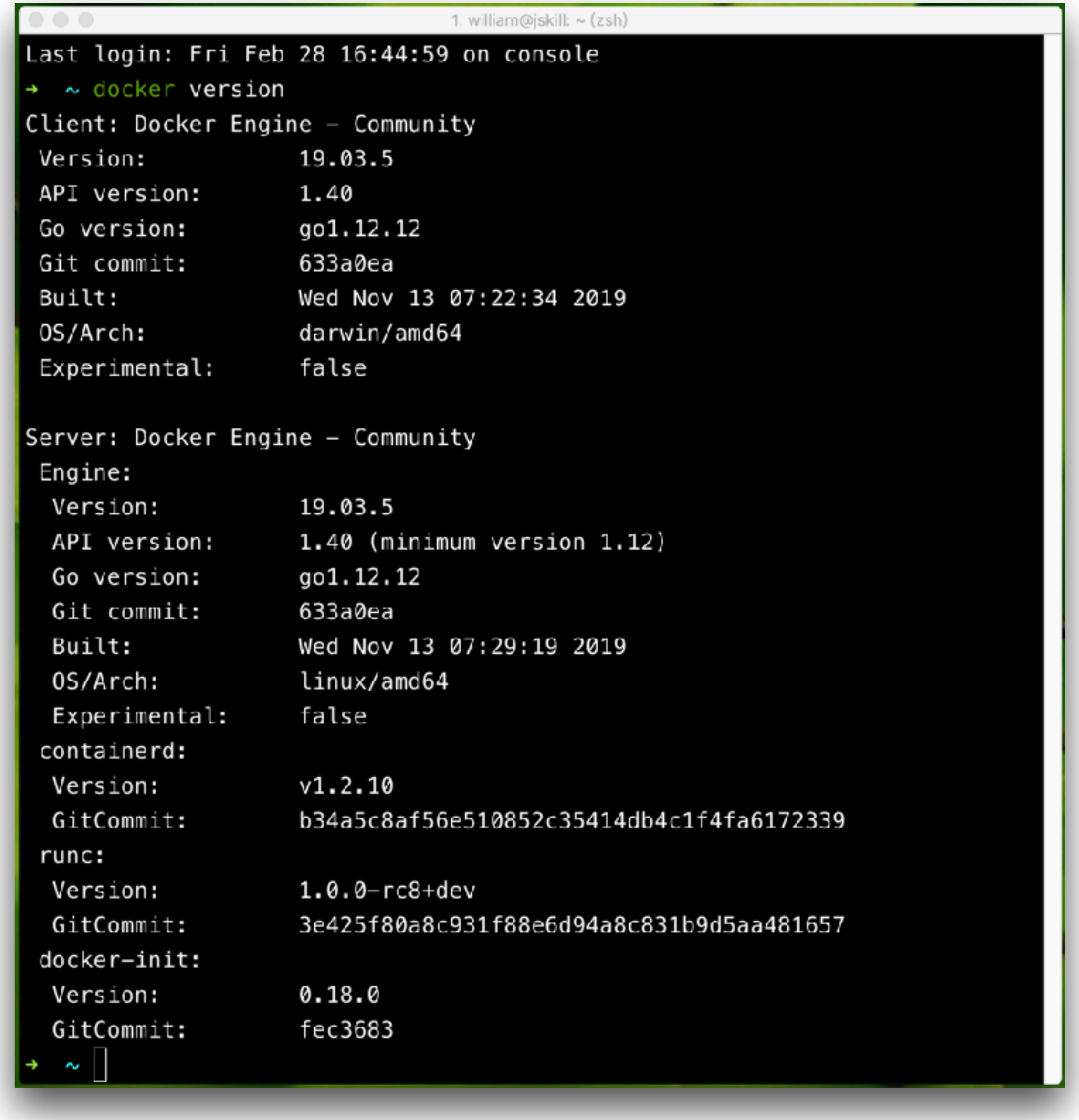
v19.03.5

Configure the Docker daemon by typing a json Docker daemon configuration file.

This can prevent Docker from starting, reset your daemon settings if it hangs.

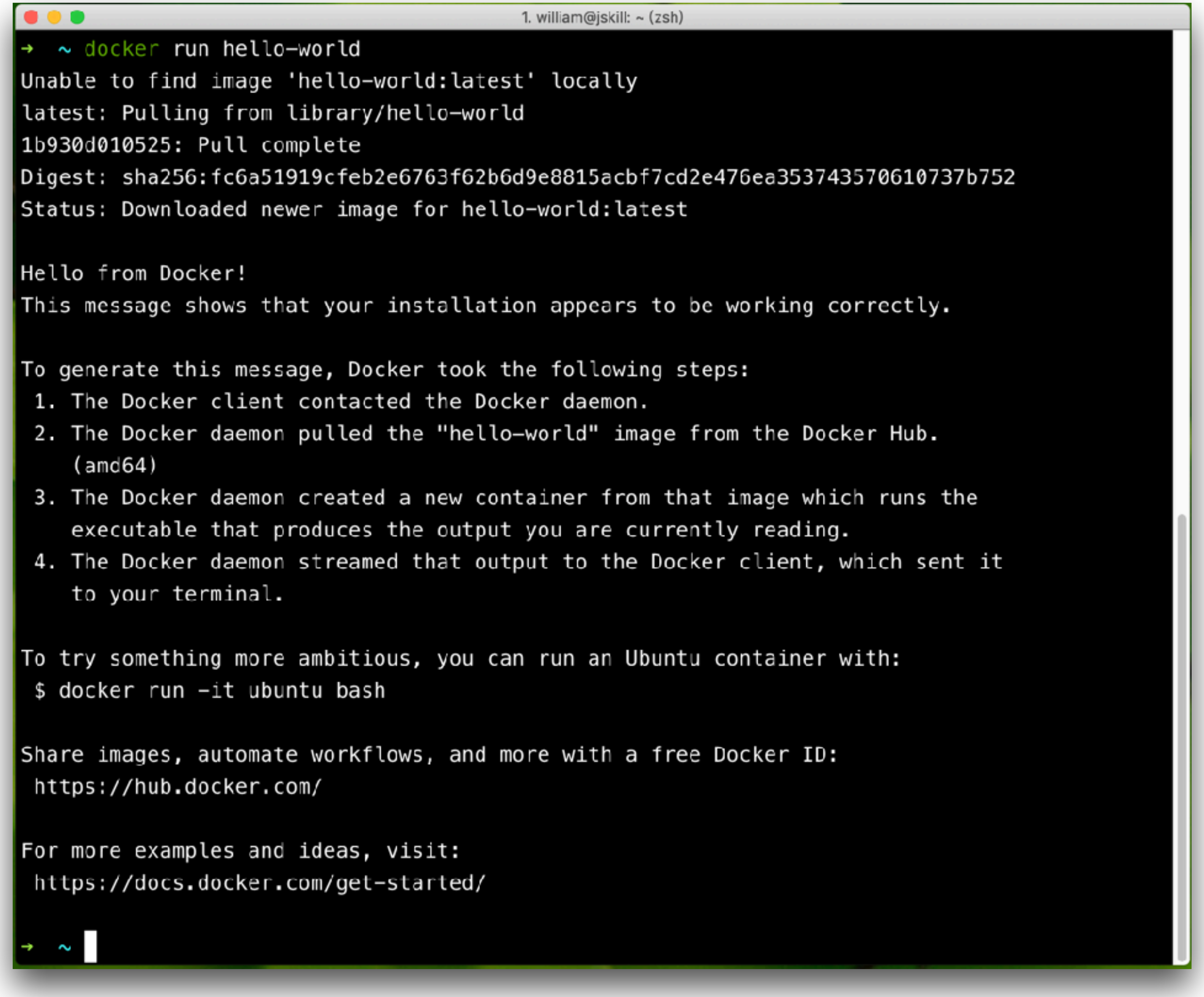
```
{
  "experimental": false,
  "debug": true,
  "registry-mirrors": ["https://docker.mirrors.ustc.edu.cn"]
}
```

校验安装 ~ docker version



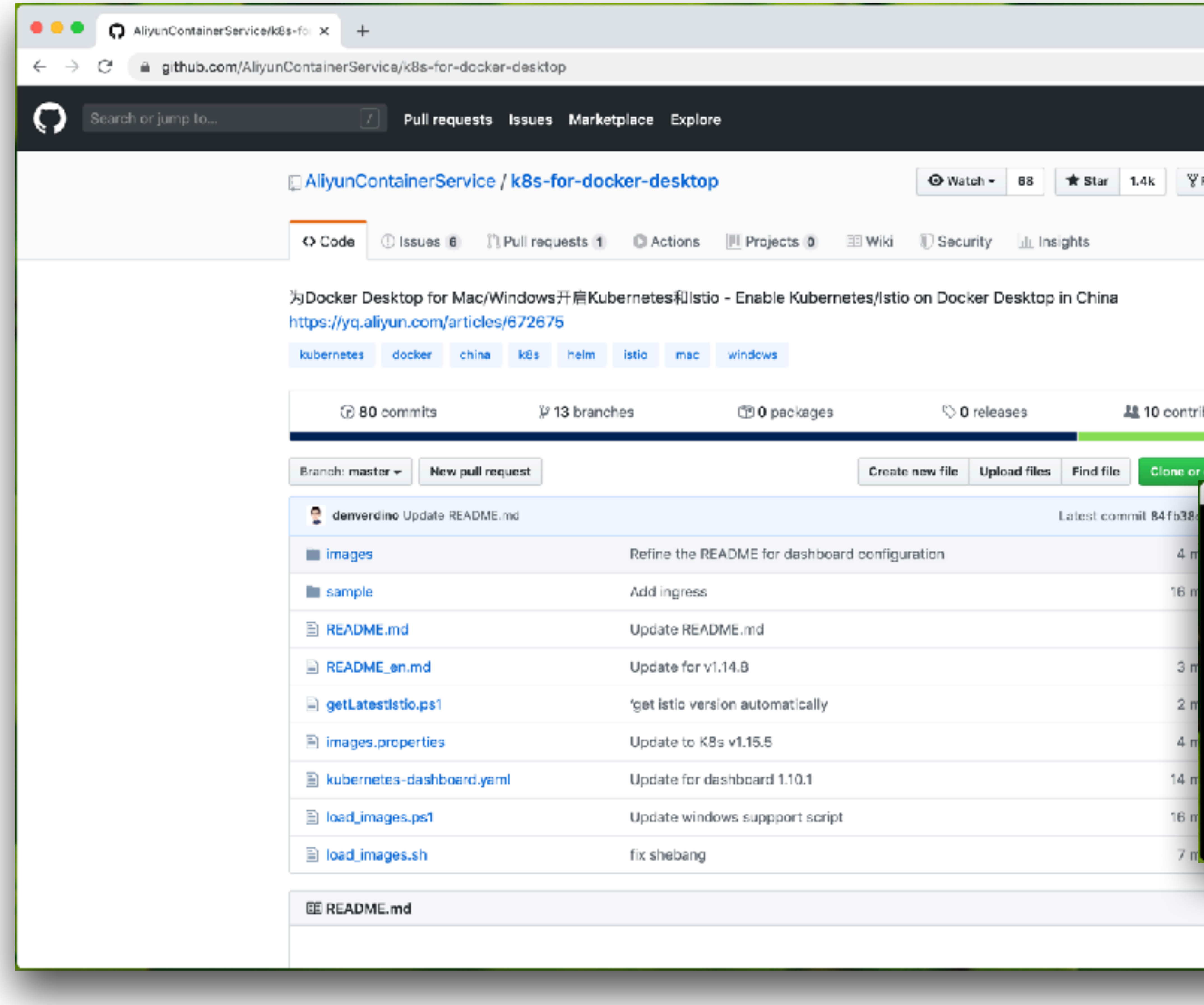
```
Last login: Fri Feb 28 16:44:59 on console
➜  ~ docker version
Client: Docker Engine - Community
 Version:           19.03.5
 API version:       1.40
 Go version:        go1.12.12
 Git commit:        633a0ea
 Built:             Wed Nov 13 07:22:34 2019
 OS/Arch:           darwin/amd64
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:          19.03.5
  API version:      1.40 (minimum version 1.12)
  Go version:       go1.12.12
  Git commit:       633a0ea
  Built:            Wed Nov 13 07:29:19 2019
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          v1.2.10
  GitCommit:        b34a5c8af56e510852c35414db4c1f4fa6172339
 runc:
  Version:          1.0.0-rc8+dev
  GitCommit:        3e425f80a8c931f88e6d94a8c831b9d5aa481657
 docker-init:
  Version:          0.18.0
  GitCommit:        fec3683
➜  ~ 
```

校验安装 ~ docker run hello-world



```
➜  ~ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:fc6a51919cfeb2e6763f62b6d9e8815acbf7cd2e476ea353743570610737b752
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

➜  ~
```

预加载k8s相关镜像



AliyunContainerService/k8s-fo ×
github.com/AliyunContainerService/k8s-for-docker-desktop
Search or jump to...
Pull requests
Issues
Marketplace
Explore

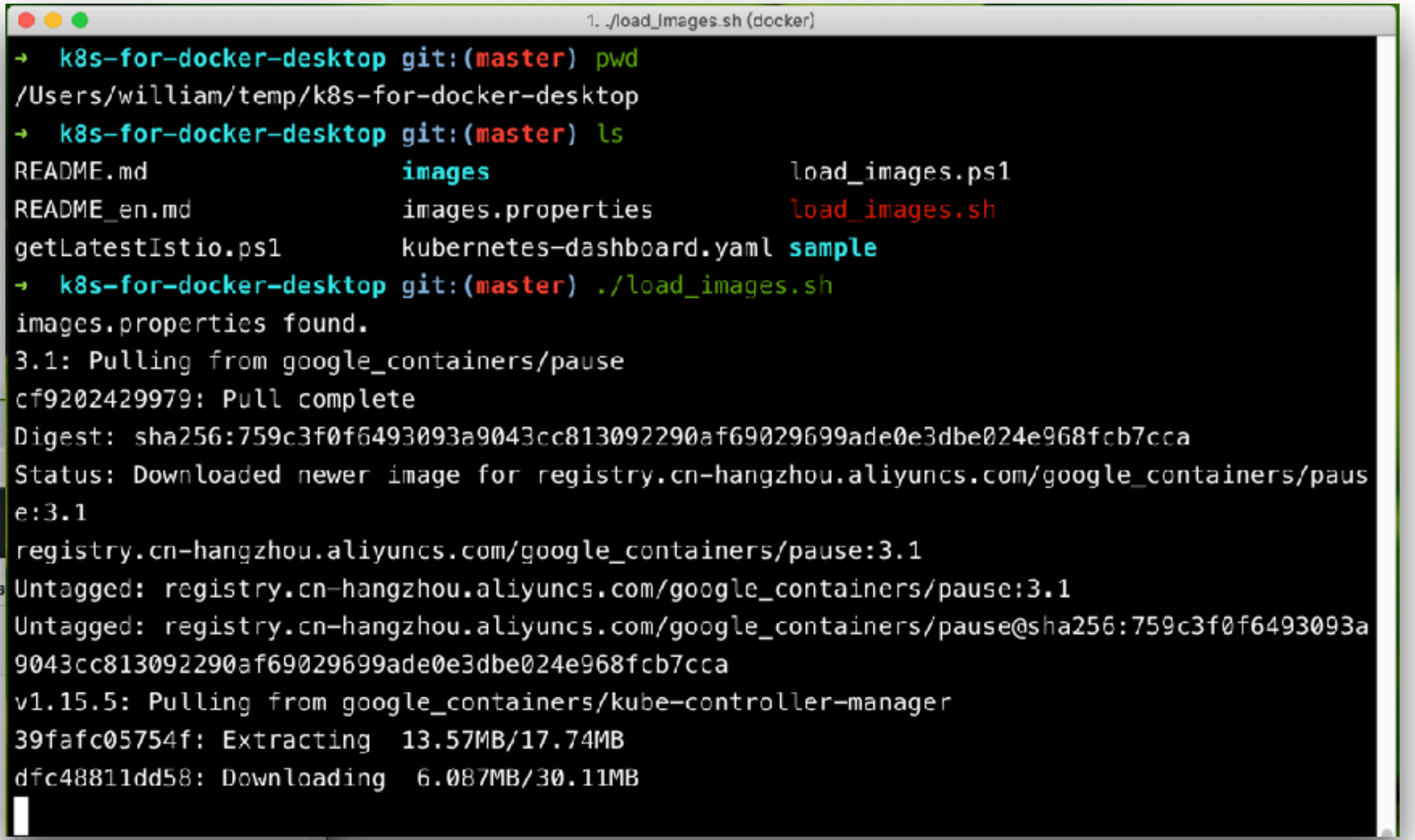
AliyunContainerService / k8s-for-docker-desktop

为Docker Desktop for Mac/Windows开启Kubernetes和Istio - Enable Kubernetes/Istio on Docker Desktop in China
https://yq.aliyun.com/articles/672675

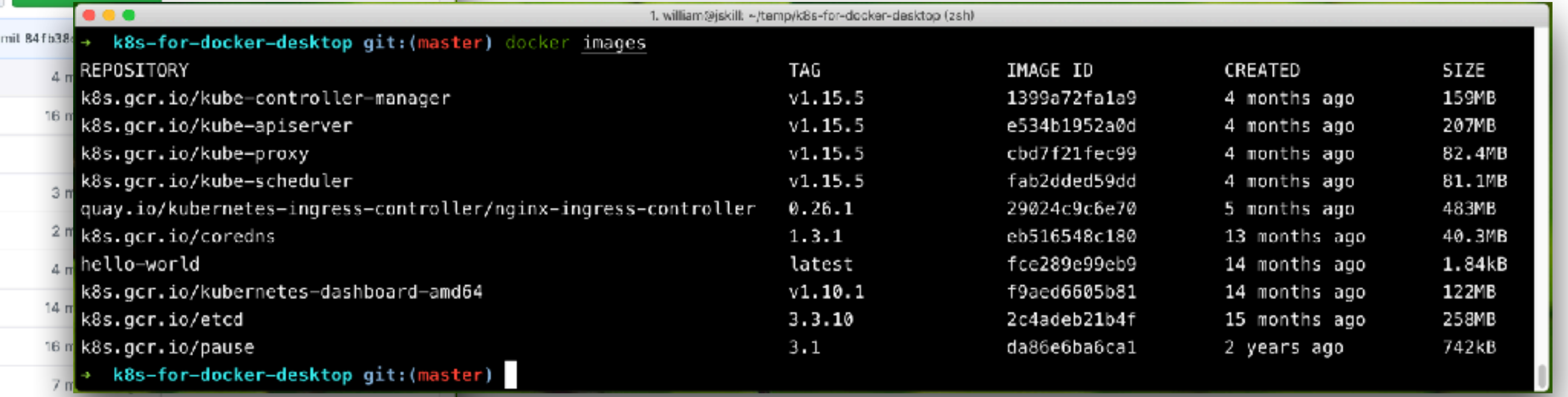
denverdino Update README.md   Latest commit 84fb38

images	Refine the README for dashboard configuration	4 m
sample	Add ingress	16 m
README.md	Update README.md	
README_en.md	Update for v1.14.8	3 m
getLatestIstio.ps1	'get istio version automatically	2 m
images.properties	Update to K8s v1.15.5	4 m
kubernetes-dashboard.yaml	Update for dashboard 1.10.1	14 m
load_images.ps1	Update windows suppport script	16 m
load_images.sh	fix shebang	7 m

README.md



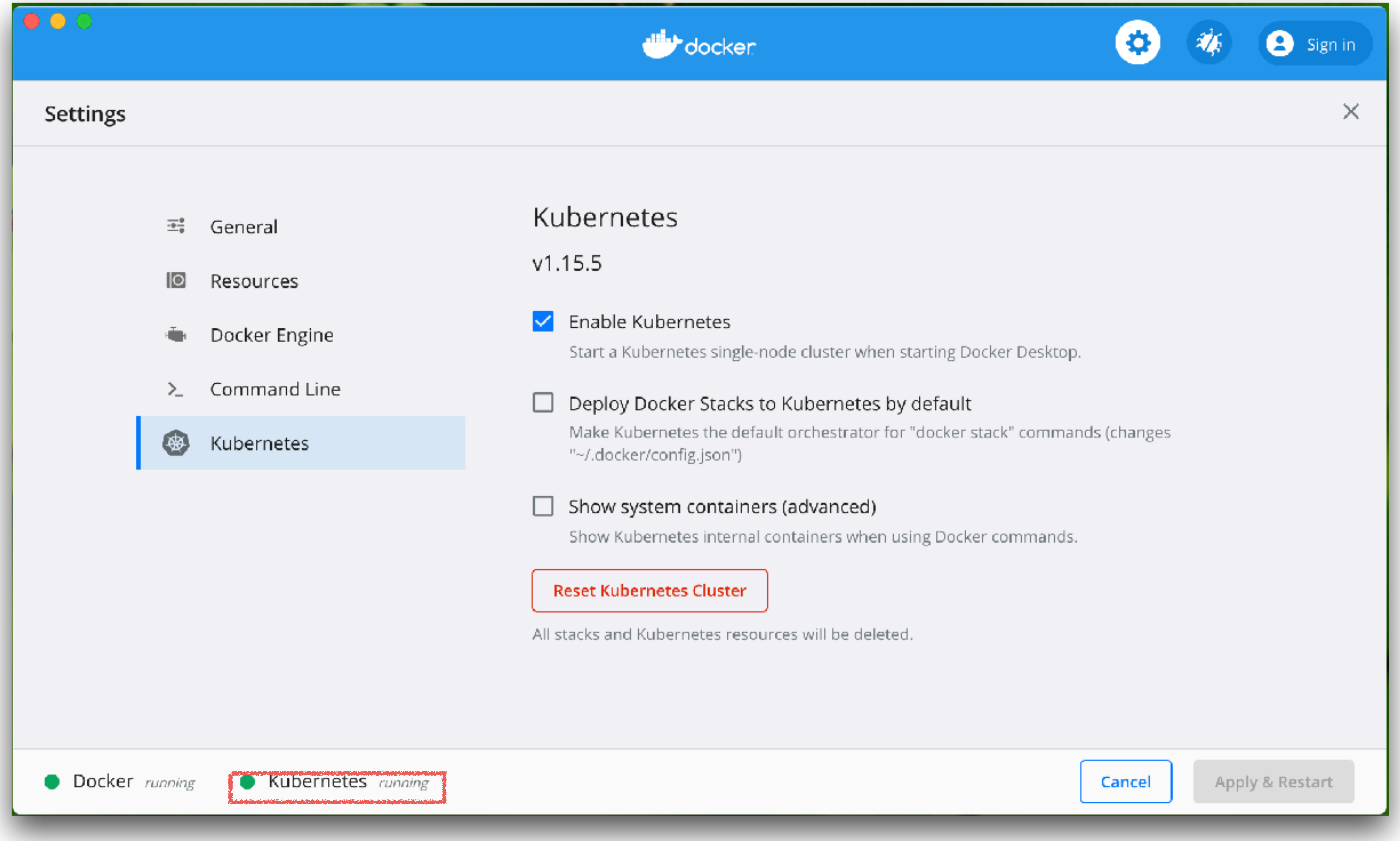
```
➜  k8s-for-docker-desktop git:(master) pwd
/Users/william/temp/k8s-for-docker-desktop
➜  k8s-for-docker-desktop git:(master) ls
README.md                 images                    load_images.ps1
README_en.md              images.properties         load_images.sh
getLatestIstio.ps1        kubernetes-dashboard.yaml sample
➜  k8s-for-docker-desktop git:(master) ./load_images.sh
images.properties found.
3.1: Pulling from google_containers/pause
cf9202429979: Pull complete
Digest: sha256:759c3f0f6493093a9043cc813092290af69029699ade0e3dbe024e968fcb7cca
Status: Downloaded newer image for registry.cn-hangzhou.aliyuncs.com/google_containers/paus
e:3.1
registry.cn-hangzhou.aliyuncs.com/google_containers/pause:3.1
Untagged: registry.cn-hangzhou.aliyuncs.com/google_containers/pause:3.1
Untagged: registry.cn-hangzhou.aliyuncs.com/google_containers/pause@sha256:759c3f0f6493093a
9043cc813092290af69029699ade0e3dbe024e968fcb7cca
v1.15.5: Pulling from google_containers/kube-controller-manager
39fafc05754f: Extracting  13.57MB/17.74MB
dfc48811dd58: Downloading  6.087MB/30.11MB
```



```
➜  k8s-for-docker-desktop git:(master) docker images
REPOSITORY                                               TAG        IMAGE ID       CREATED         SIZE
k8s.gcr.io/kube-controller-manager                       v1.15.5    1399a72fa1a9   4 months ago    159MB
k8s.gcr.io/kube-apiserver                                v1.15.5    e534b1952a0d   4 months ago    207MB
k8s.gcr.io/kube-proxy                                    v1.15.5    cbd7f21fec99   4 months ago    82.4MB
k8s.gcr.io/kube-scheduler                                v1.15.5    fab2dded59dd   4 months ago    81.1MB
quay.io/kubernetes-ingress-controller/nginx-ingress-controller 0.26.1 29024c9c6e70 5 months ago 483MB
k8s.gcr.io/coredns                                       1.3.1      eb516548c180   13 months ago   40.3MB
hello-world                                              latest     fce289e99eb9   14 months ago   1.84kB
k8s.gcr.io/kubernetes-dashboard-amd64                    v1.10.1    f9aed6605b81   14 months ago   122MB
k8s.gcr.io/etcd                                          3.3.10     2c4adeb21b4f   15 months ago   258MB
k8s.gcr.io/pause                                         3.1        da86e6ba6ca1   2 years ago     742kB
➜  k8s-for-docker-desktop git:(master)
```

https://github.com/AliyunContainerService/k8s-for-docker-desktop

启用K8s设置



Kubernetes

v1.15.5

☑ Enable Kubernetes

Start a Kubernetes single-node cluster when starting Docker Desktop.

☐ Deploy Docker Stacks to Kubernetes by default

Make Kubernetes the default orchestrator for "docker stack" commands (changes "~/.docker/config.json")

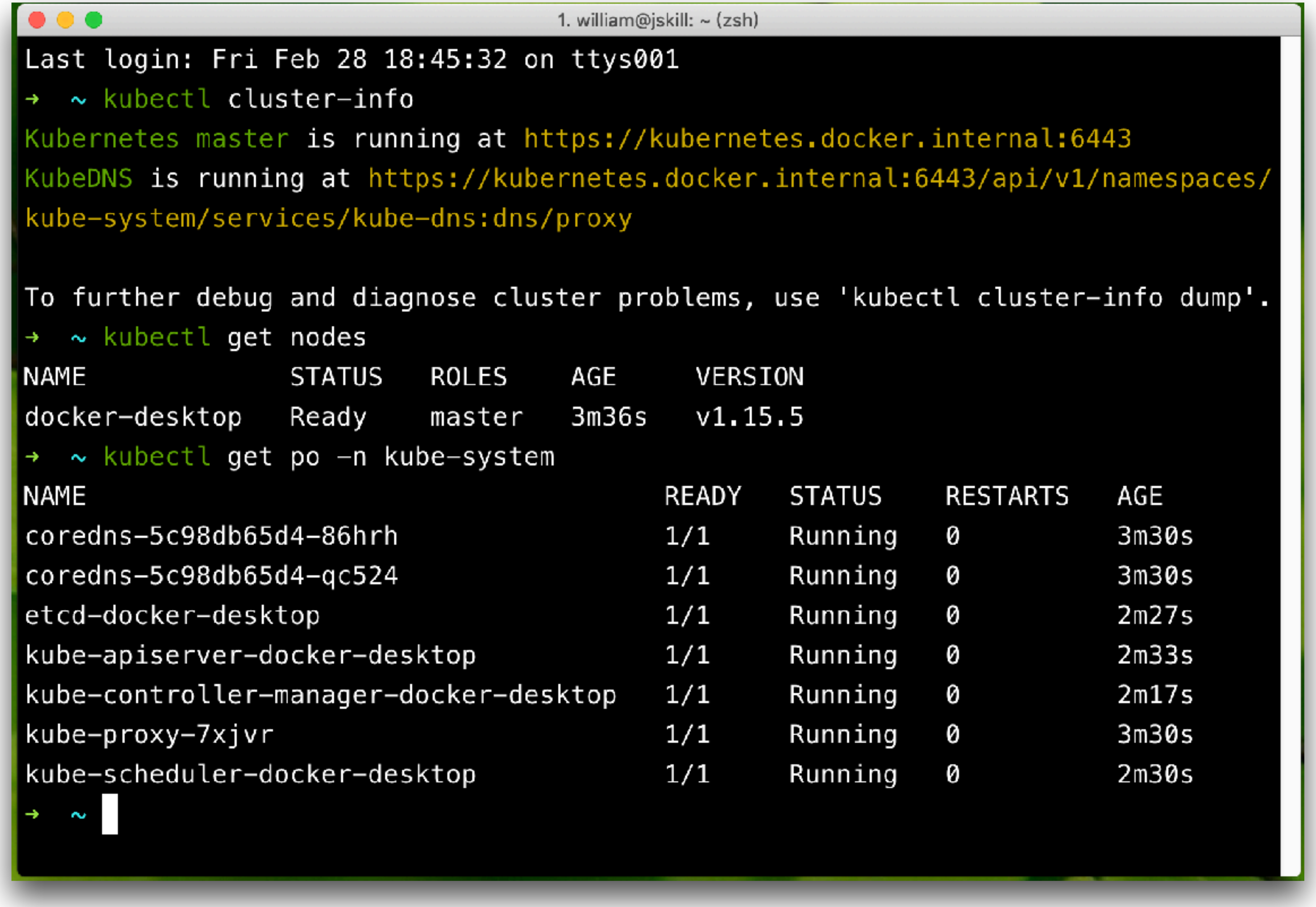
☐ Show system containers (advanced)

Show Kubernetes internal containers when using Docker commands.

All stacks and Kubernetes resources will be deleted.

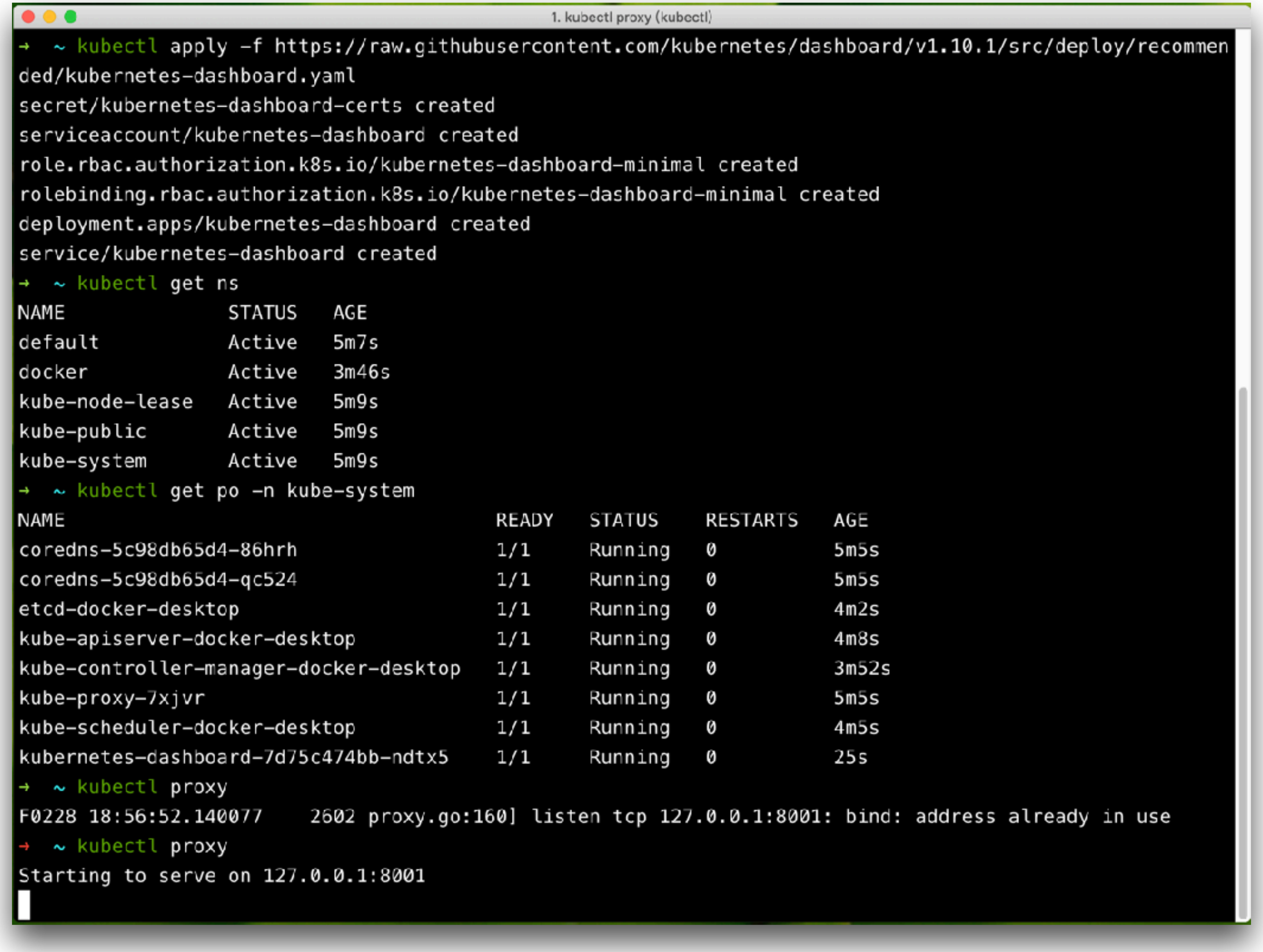
● Docker *running* ● Kubernetes *running*

校验K8s安装



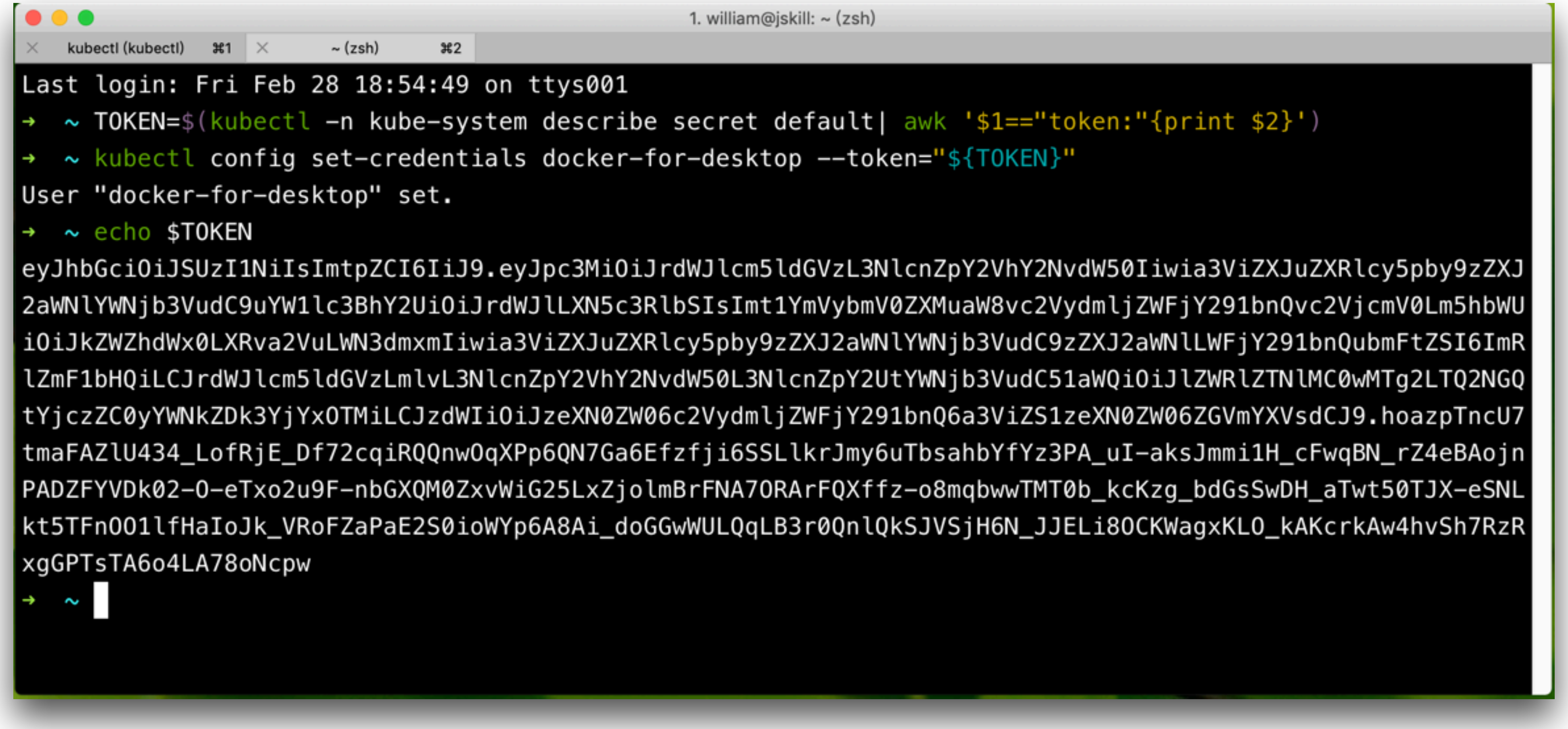
```
Last login: Fri Feb 28 18:45:32 on ttys001
➜  ~ kubectl cluster-info
Kubernetes master is running at https://kubernetes.docker.internal:6443
KubeDNS is running at https://kubernetes.docker.internal:6443/api/v1/namespaces/
kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
➜  ~ kubectl get nodes
NAME             STATUS   ROLES    AGE     VERSION
docker-desktop   Ready    master   3m36s   v1.15.5
➜  ~ kubectl get po -n kube-system
NAME                                     READY   STATUS    RESTARTS   AGE
coredns-5c98db65d4-86hrh                 1/1     Running   0          3m30s
coredns-5c98db65d4-qc524                 1/1     Running   0          3m30s
etcd-docker-desktop                      1/1     Running   0          2m27s
kube-apiserver-docker-desktop            1/1     Running   0          2m33s
kube-controller-manager-docker-desktop   1/1     Running   0          2m17s
kube-proxy-7xjvr                         1/1     Running   0          3m30s
kube-scheduler-docker-desktop            1/1     Running   0          2m30s
➜  ~ 
```

安装k8s dashboard并启动proxy



```
→  ~ kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v1.10.1/src/deploy/recommen
ded/kubernetes-dashboard.yaml
secret/kubernetes-dashboard-certs created
serviceaccount/kubernetes-dashboard created
role.rbac.authorization.k8s.io/kubernetes-dashboard-minimal created
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard-minimal created
deployment.apps/kubernetes-dashboard created
service/kubernetes-dashboard created
→  ~ kubectl get ns
NAME              STATUS   AGE
default           Active   5m7s
docker            Active   3m46s
kube-node-lease   Active   5m9s
kube-public       Active   5m9s
kube-system       Active   5m9s
→  ~ kubectl get po -n kube-system
NAME                                     READY   STATUS    RESTARTS   AGE
coredns-5c98db65d4-86hrh                 1/1     Running   0          5m5s
coredns-5c98db65d4-qc524                 1/1     Running   0          5m5s
etcd-docker-desktop                      1/1     Running   0          4m2s
kube-apiserver-docker-desktop            1/1     Running   0          4m8s
kube-controller-manager-docker-desktop   1/1     Running   0          3m52s
kube-proxy-7xjvr                         1/1     Running   0          5m5s
kube-scheduler-docker-desktop            1/1     Running   0          4m5s
kubernetes-dashboard-7d75c474bb-ndtx5    1/1     Running   0          25s
→  ~ kubectl proxy
F0228 18:56:52.140077    2602 proxy.go:160] listen tcp 127.0.0.1:8001: bind: address already in use
→  ~ kubectl proxy
Starting to serve on 127.0.0.1:8001
```

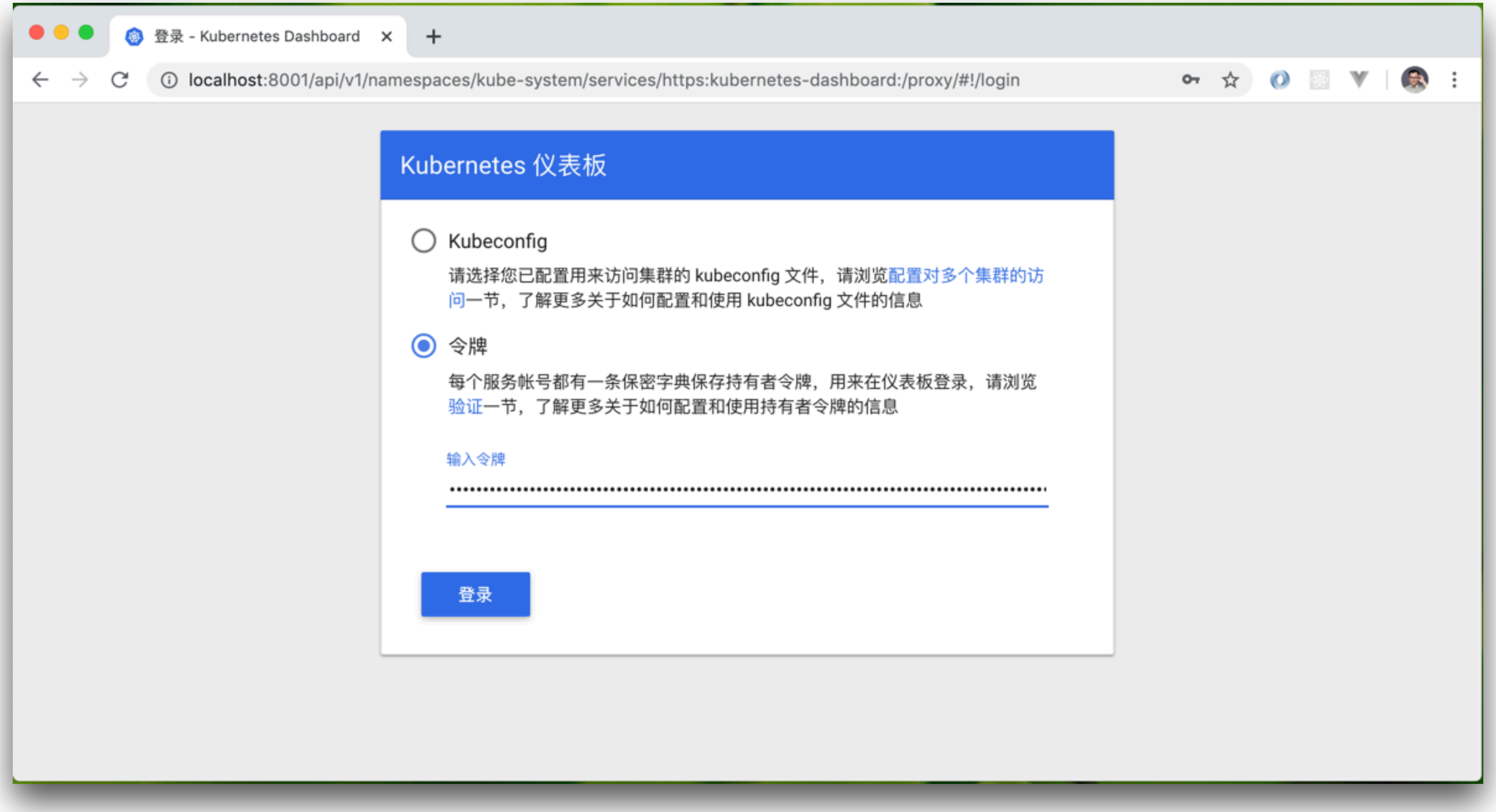
获取dashboard令牌



```
Last login: Fri Feb 28 18:54:49 on ttys001
→  ~ TOKEN=$(kubectl -n kube-system describe secret default| awk '$1=="token:"{print $2}')
→  ~ kubectl config set-credentials docker-for-desktop --token="${TOKEN}"
User "docker-for-desktop" set.
→  ~ echo $TOKEN
eyJhbGciOiJSUzI1NiIsImtpZCI6IiJ9.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ
2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJrdWJlLXN5c3RlbSIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VjcmV0Lm5hbWU
iOiJkZWZhdWx0LXRva2VuLWN3dmxmIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQubmFtZSI6ImR
lZmF1bHQiLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC51aWQiOiJlZWRlZTNlMC0wMTg2LTQ2NGQ
tYjczZC0yYWNkZDk3YjYxOTMiLCJzdWIiOiJzeXN0ZW06c2VydmljZWFjY291bnQ6a3ViZS1zeXN0ZW06ZGVmYXVsdCJ9.hoazpTncU7
tmaFAZlU434_LofRjE_Df72cqiRQQnwOqXPp6QN7Ga6Efzfji6SSLlkrJmy6uTbsahbYfYz3PA_uI-aksJmmi1H_cFwqBN_rZ4eBAojn
PADZFYVDk02-O-eTxo2u9F-nbGXQM0ZxvWiG25LxZjolmBrFNA7ORArFQXffz-o8mqbwwTMT0b_kcKzg_bdGsSwDH_aTwt50TJX-eSNL
kt5TFnOO1lfHaIoJk_VRoFZaPaE2S0ioWYp6A8Ai_doGGwWULQqLB3r0QnlQkSJVSjH6N_JJELi8OCKWagxKLO_kAKcrkAw4hvSh7RzR
xgGPTsTA6o4LA78oNcpw
→  ~
```

```
TOKEN=$(kubectl -n kube-system describe secret default| awk '$1=="token:"{print $2}')
kubectl config set-credentials docker-for-desktop --token="${TOKEN}"
echo $TOKEN
```

登录dashboard


登录 - Kubernetes Dashboard
localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/#!/login

Kubernetes 仪表板

○ Kubeconfig

请选择您已配置用来访问集群的 kubeconfig 文件，请浏览配置对多个集群的访问一节，了解更多关于如何配置和使用 kubeconfig 文件的信息

● 令牌

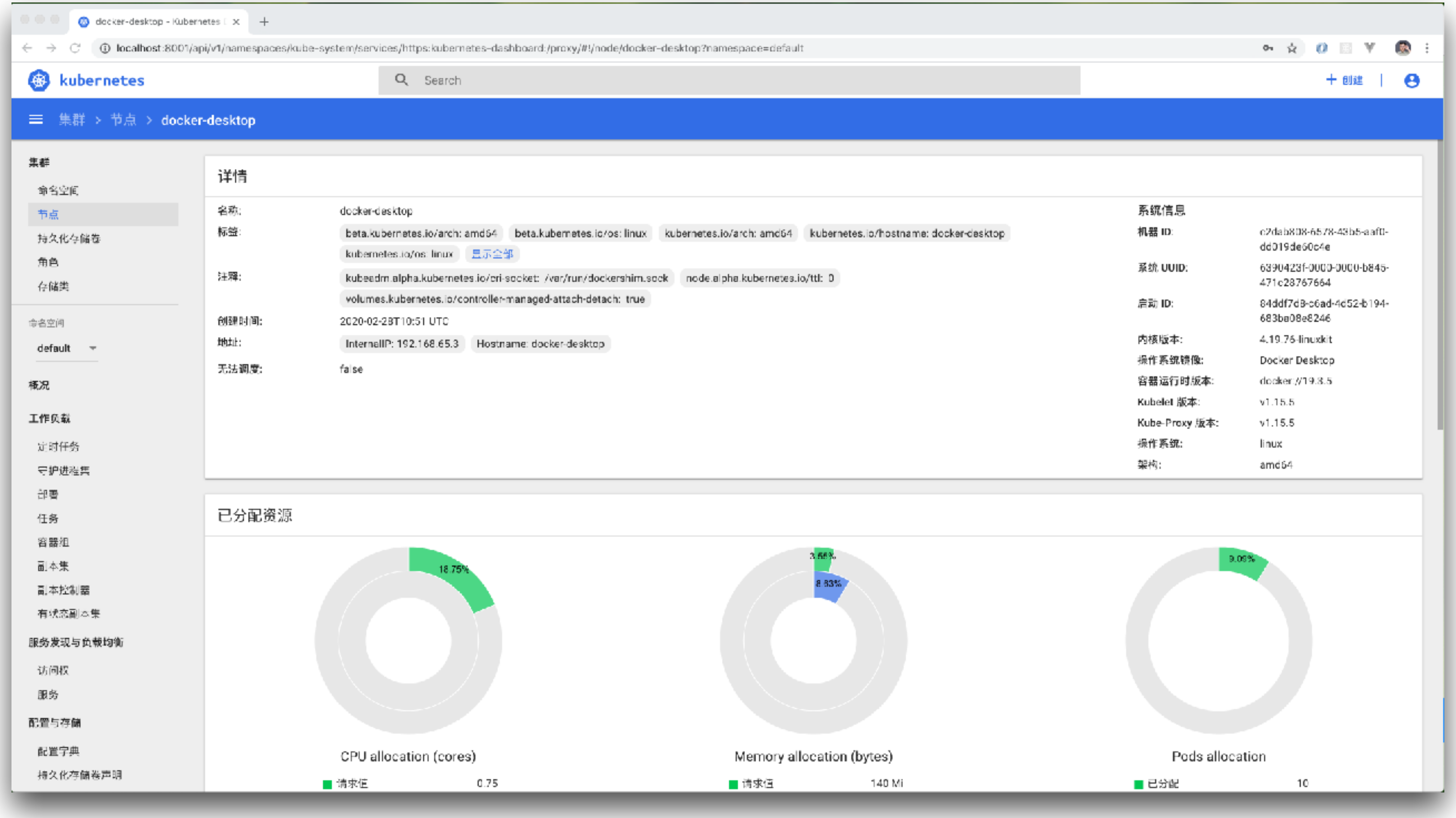
每个服务帐号都有一条保密字典保存持有者令牌，用来在仪表板登录，请浏览验证一节，了解更多关于如何配置和使用持有者令牌的信息

输入令牌

登录

http://localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/#!/overview?namespace=default

访问dashboard



详情

名称: docker-desktop

标签: beta.kubernetes.io/arch: amd64 | beta.kubernetes.io/os: linux | kubernetes.io/arch: amd64 | kubernetes.io/hostname: docker-desktop | kubernetes.io/os: linux | 显示全部

注释: kubeadm.alpha.kubernetes.io/cri-socket: /var/run/dockershim.sock | node.alpha.kubernetes.io/ttl: 0 | volumes.kubernetes.io/controller-managed-attach-detach: true

创建时间: 2020-02-28T10:51 UTC

地址: InternalIP: 192.168.65.3 | Hostname: docker-desktop

无法调度: false

系统信息

机器 ID:	c2dab808-6578-43b5-aaf0-dd019de60c4e
系统 UUID:	6390423f-0000-0000-b845-471c28767664
启动 ID:	84ddf7d8-c6ad-4c52-b194-683ba08e8246
内核版本:	4.19.76-linuxkit
操作系统镜像:	Docker Desktop
容器运行时版本:	docker://19.3.5
Kubelet 版本:	v1.15.5
Kube-Proxy 版本:	v1.15.5
操作系统:	linux
架构:	amd64

已分配资源

CPU allocation (cores)
🟩 请求值 0.75

Memory allocation (bytes)
🟩 请求值 140 Mi

Pods allocation
🟩 已分配 10

本课小结

- 演示本地安装Docker Desktop for Mac + 启用K8s
  - 先决条件：对macOs版本有要求，否则只能考虑Minikube + VirutalBox
  - 设置容器资源(CPU/Memory)
  - 设置镜像加速器(考虑申请阿里云镜像加速器)
  - 阿里云k8s镜像预加载脚本(国内网络)
  - 安装参考： https://github.com/AliyunContainerService/k8s-for-docker-desktop
  - 安装步骤如更新请参考官方安装文档： https://docs.docker.com/docker-for-mac/install/
  - 第8章第2节，演示安装Minikube for Mac