Leader-us系列

从Docker到Kubernetes 第10周

DATAGURU专业数据分析社区

法律声明

【声明】本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

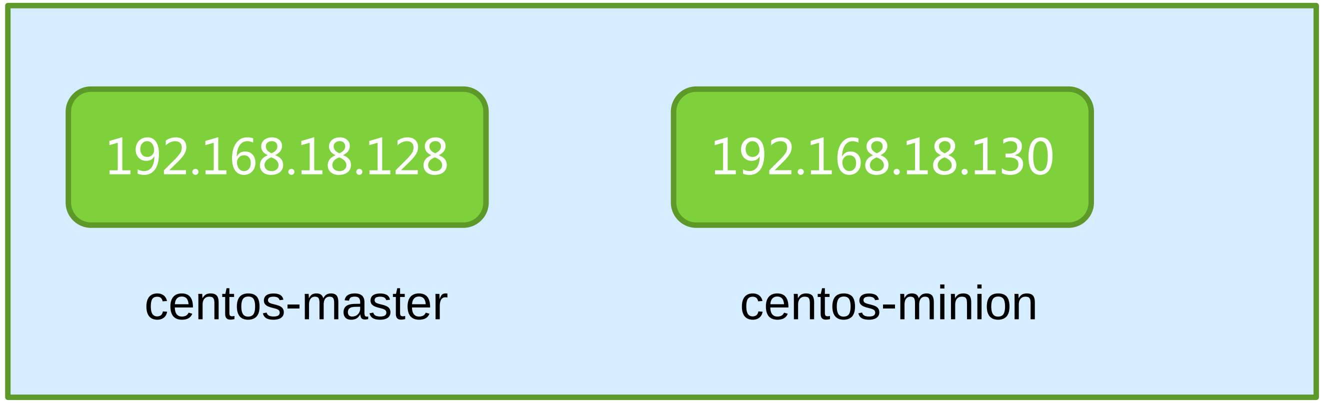
课程详情访问炼数成金培训网站

http://edu.dataguru.cn

- Kubernetes安装
- Kubernetes基本命令
- Hello world

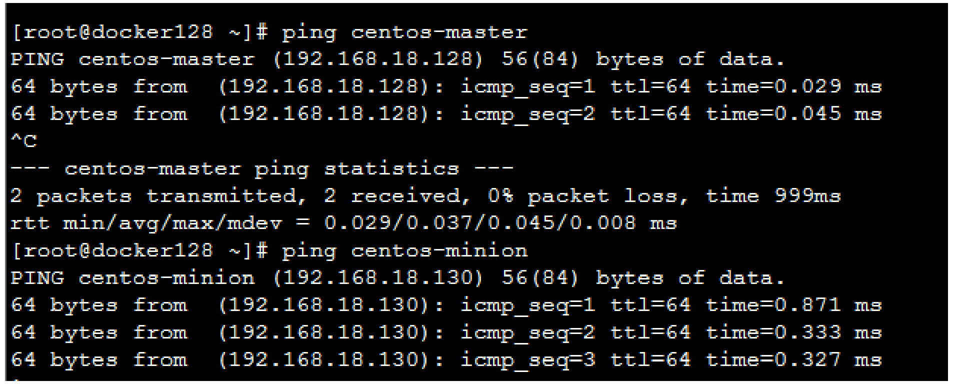
Kubernetes安装

https://github.com/kubernetes/kubernetes/blob/release-1.0/docs/getting-started-guides/centos/centos_manual_config.md



vi /etc/hostname

```
echo "192.168.18.128  centos-master
192.168.18.130   centos-minion
" >> /etc/hosts
```

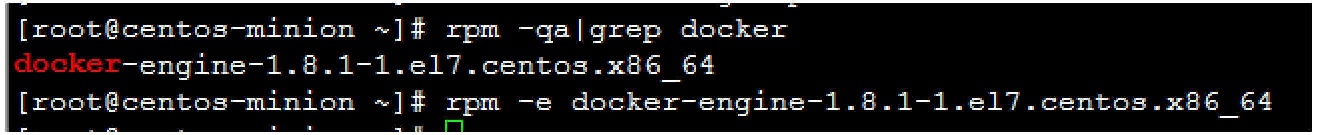


```
[root@docker128 ~]# ping centos-master
PING centos-master (192.168.18.128) 56(84) bytes of data.
64 bytes from  (192.168.18.128): icmp_seq=1 ttl=64 time=0.029 ms
64 bytes from  (192.168.18.128): icmp_seq=2 ttl=64 time=0.045 ms
^C
--- centos-master ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.029/0.037/0.045/0.008 ms
[root@docker128 ~]# ping centos-minion
PING centos-minion (192.168.18.130) 56(84) bytes of data.
64 bytes from  (192.168.18.130): icmp_seq=1 ttl=64 time=0.871 ms
64 bytes from  (192.168.18.130): icmp_seq=2 ttl=64 time=0.333 ms
64 bytes from  (192.168.18.130): icmp_seq=3 ttl=64 time=0.327 ms
```

Kubernetes安装

Create virt7-testing repo on all hosts

```
[virt7-testing]
name=virt7-testing
baseurl=http://cbs.centos.org/repos/virt7-testing/x86_64/os/
gpgcheck=0
```

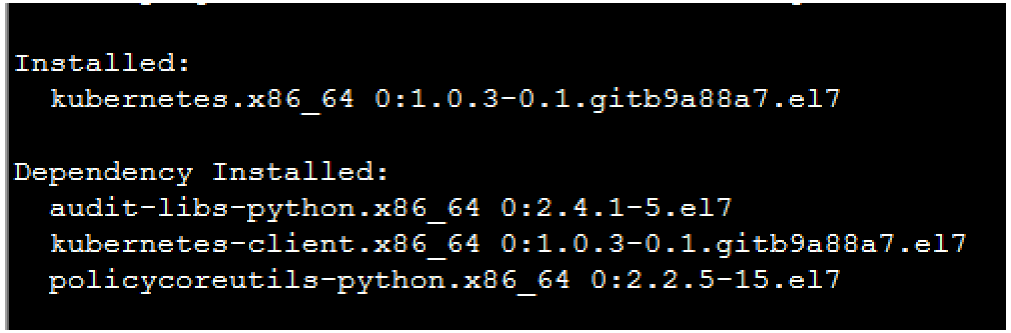


```
[root@centos-minion ~]# rpm -qa|grep docker
docker-engine-1.8.1-1.el7.centos.x86_64
[root@centos-minion ~]# rpm -e docker-engine-1.8.1-1.el7.centos.x86_64
```

```
cd /etc/yum.repo.d
vi virt7-testing.repo
```

Install Kubernetes on all hosts - centos-{master,minion}

```
yum -y install --enablerepo=virt7-testing kubernetes
```

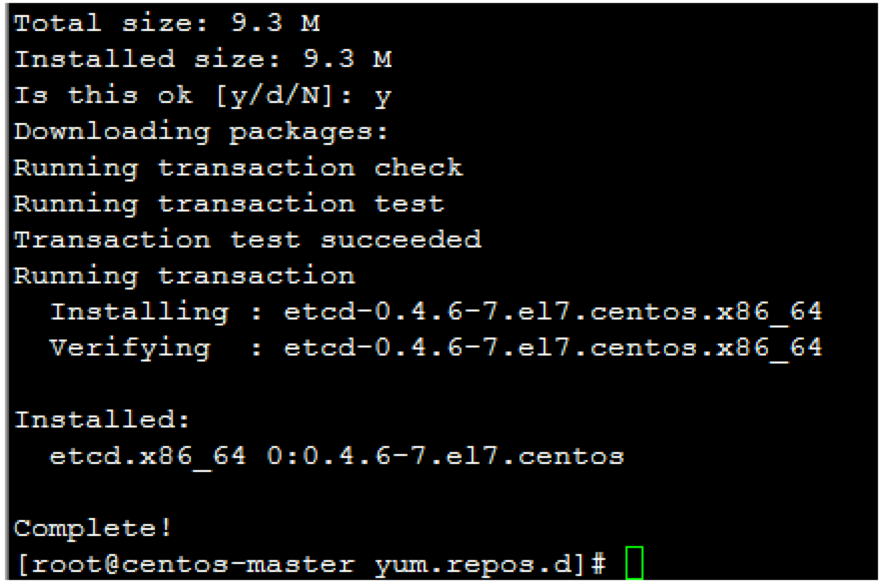


```
Installed:
  kubernetes.x86_64 0:1.0.3-0.1.gitb9a88a7.el7

Dependency Installed:
  audit-libs-python.x86_64 0:2.4.1-5.el7
  kubernetes-client.x86_64 0:1.0.3-0.1.gitb9a88a7.el7
  policycoreutils-python.x86_64 0:2.2.5-15.el7
```

Kubernetes安装

master 节点安装 etcd package

yum install http://cbs.centos.org/kojifiles/packages/etcd/0.4.6/7.el7.centos/x86_64/etcd-0.4.6-7.el7.centos.x86_64.rpm



```
Total size: 9.3 M
Installed size: 9.3 M
Is this ok [y/d/N]: y
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : etcd-0.4.6-7.el7.centos.x86_64
  Verifying  : etcd-0.4.6-7.el7.centos.x86_64

Installed:
  etcd.x86_64 0:0.4.6-7.el7.centos

Complete!
[root@centos-master yum.repos.d]# 
```

```
systemctl disable iptables-services firewalld
systemctl stop iptables-services firewalld
```

Kubernetes安装

每个节点上修改kubernetes配置文件

vi /etc/kubernetes/config

```
# How the controller-manager, scheduler, and proxy find the apiserver
KUBE_MASTER="--master=http://centos-master:8080"
KUBE_ETCD_SERVERS="--etcd_servers=http://centos-master:4001"
```

Master节点上 vi /etc/kubernetes/apiserver

```
# The address on the local server to listen to.
KUBE_API_ADDRESS="--address=0.0.0.0"
KUBE_API_PORT="--port=8080"
# Comma separated list of nodes in the etcd cluster
#KUBE_ETCD_SERVERS="--etcd_servers=http://127.0.0.1:2379"
```

Kubernetes安装

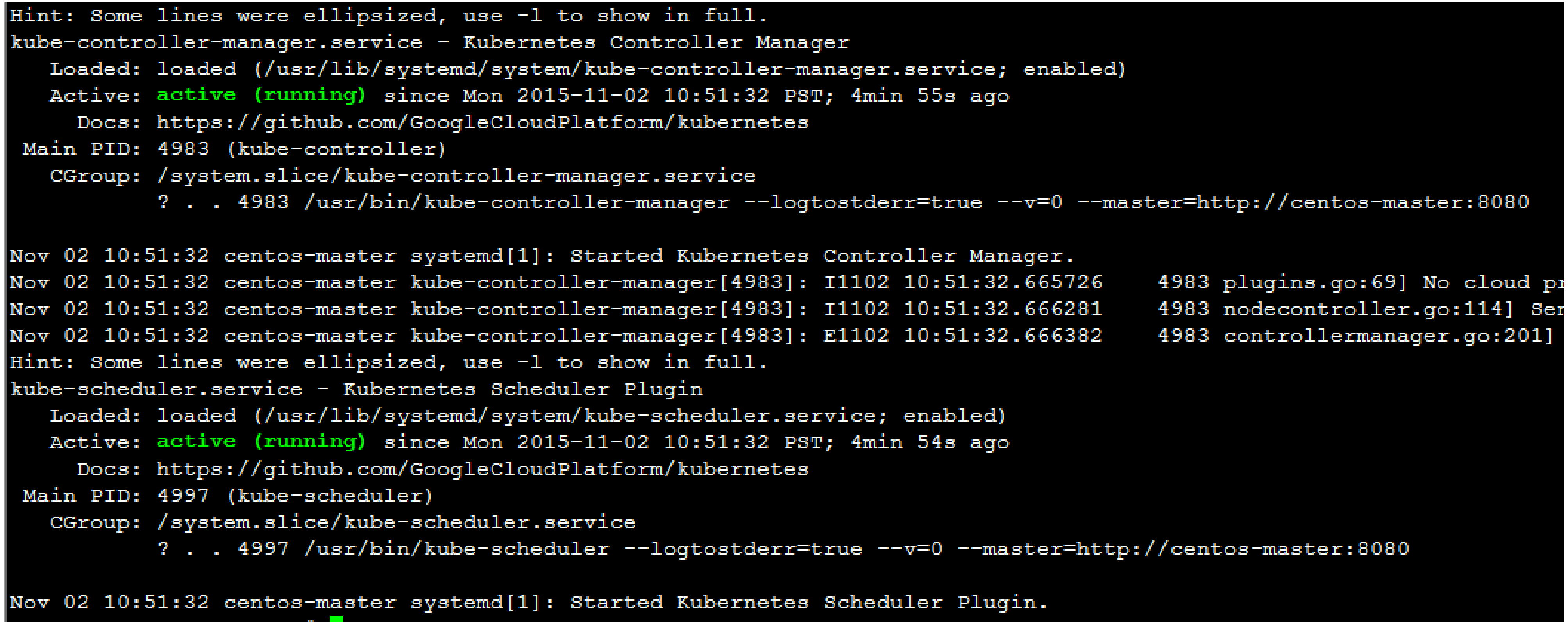
Master节点启动相关kubernetes服务

```
for SERVICES in etcd kube-apiserver kube-controller-manager kube-scheduler; do
    systemctl restart $SERVICES
    systemctl enable $SERVICES
    systemctl status $SERVICES
done
```

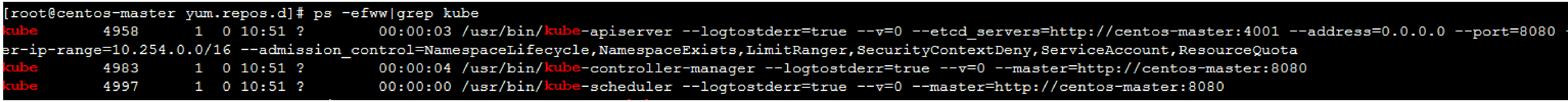
```
for SERVICES in etcd kube-apiserver kube-controller-manager kube-scheduler; do
    systemctl status $SERVICES
done
```

Kubernetes安装

Master节点启动相关kubernetes服务



```
Hint: Some lines were ellipsized, use -l to show in full.
kube-controller-manager.service - Kubernetes Controller Manager
   Loaded: loaded (/usr/lib/systemd/system/kube-controller-manager.service; enabled)
   Active: active (running) since Mon 2015-11-02 10:51:32 PST; 4min 55s ago
     Docs: https://github.com/GoogleCloudPlatform/kubernetes
 Main PID: 4983 (kube-controller)
   CGroup: /system.slice/kube-controller-manager.service
           ? . . 4983 /usr/bin/kube-controller-manager --logtostderr=true --v=0 --master=http://centos-master:8080

Nov 02 10:51:32 centos-master systemd[1]: Started Kubernetes Controller Manager.
Nov 02 10:51:32 centos-master kube-controller-manager[4983]: I1102 10:51:32.665726    4983 plugins.go:69] No cloud pr
Nov 02 10:51:32 centos-master kube-controller-manager[4983]: I1102 10:51:32.666281    4983 nodecontroller.go:114] Ser
Nov 02 10:51:32 centos-master kube-controller-manager[4983]: E1102 10:51:32.666382    4983 controllermanager.go:201]
Hint: Some lines were ellipsized, use -l to show in full.
kube-scheduler.service - Kubernetes Scheduler Plugin
   Loaded: loaded (/usr/lib/systemd/system/kube-scheduler.service; enabled)
   Active: active (running) since Mon 2015-11-02 10:51:32 PST; 4min 54s ago
     Docs: https://github.com/GoogleCloudPlatform/kubernetes
 Main PID: 4997 (kube-scheduler)
   CGroup: /system.slice/kube-scheduler.service
           ? . . 4997 /usr/bin/kube-scheduler --logtostderr=true --v=0 --master=http://centos-master:8080

Nov 02 10:51:32 centos-master systemd[1]: Started Kubernetes Scheduler Plugin.
```



```
[root@centos-master yum.repos.d]# ps -efww|grep kube
kube      4958     1  0 10:51 ?        00:00:03 /usr/bin/kube-apiserver --logtostderr=true --v=0 --etcd_servers=http://centos-master:4001 --address=0.0.0.0 --port=8080 -
er-ip-range=10.254.0.0/16 --admission_control=NamespaceLifecycle,NamespaceExists,LimitRanger,SecurityContextDeny,ServiceAccount,ResourceQuota
kube      4983     1  0 10:51 ?        00:00:04 /usr/bin/kube-controller-manager --logtostderr=true --v=0 --master=http://centos-master:8080
kube      4997     1  0 10:51 ?        00:00:00 /usr/bin/kube-scheduler --logtostderr=true --v=0 --master=http://centos-master:8080
```

Kubernetes安装

Node节点修改/etc/kubernetes/kubelet

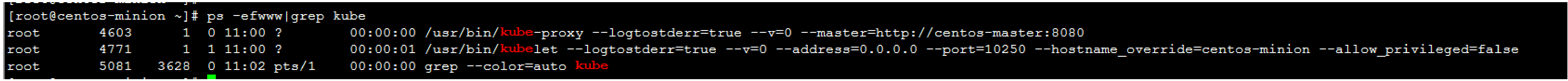
```
###
# kubernetes kubelet (minion) config
KUBELET_ADDRESS="--address=0.0.0.0"
KUBELET_PORT="--port=10250"
KUBELET_HOSTNAME="--hostname_override=centos-minion"
KUBELET_API_SERVER="--api_servers=http://centos-master:8080“
# Add your own!
KUBELET_ARGS=""
```

Kubernetes安装

node节点

```
for SERVICES in kube-proxy kubelet docker; do
    systemctl restart $SERVICES
    systemctl enable $SERVICES
    systemctl status $SERVICES
done
```

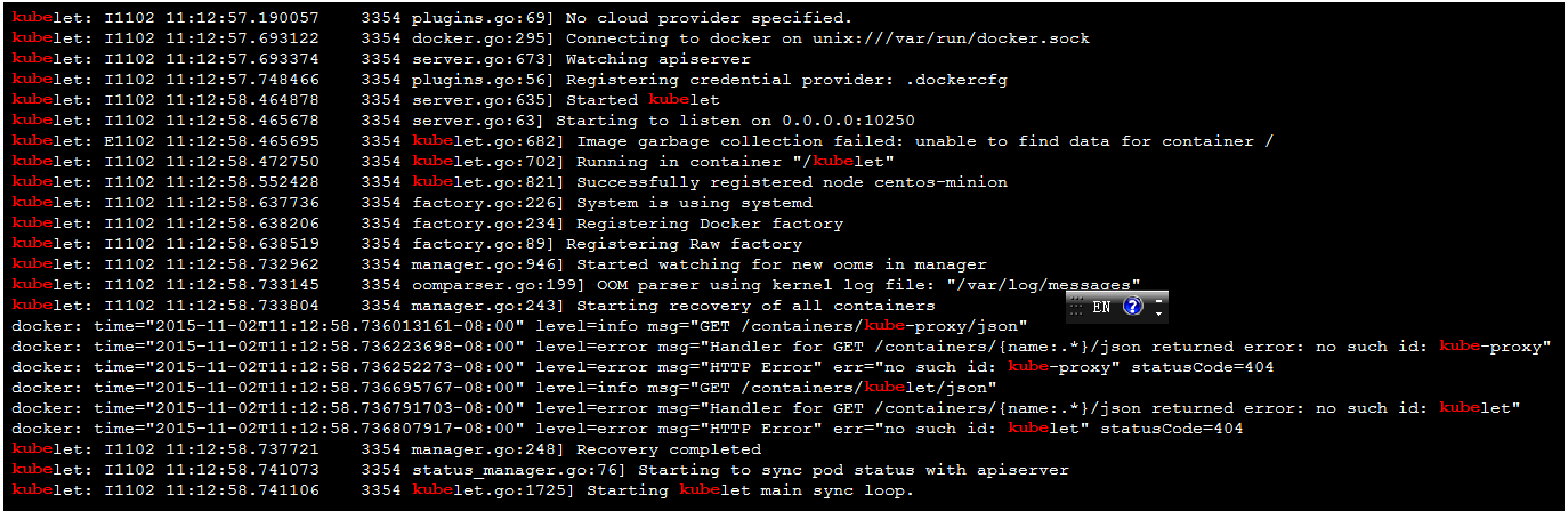
```
for SERVICES in kube-proxy kubelet docker; do
  systemctl status $SERVICES
done
```



```
[root@centos-minion ~]# ps -efwww|grep kube
root      4603     1  0 11:00 ?        00:00:00 /usr/bin/kube-proxy --logtostderr=true --v=0 --master=http://centos-master:8080
root      4771     1  1 11:00 ?        00:00:01 /usr/bin/kubelet --logtostderr=true --v=0 --address=0.0.0.0 --port=10250 --hostname_override=centos-minion --allow_privileged=false
root      5081  3628  0 11:02 pts/1    00:00:00 grep --color=auto kube
```

Kubernetes安装

tail -f /var/log/messages |grep kube



```
kubelet: I1102 11:12:57.190057    3354 plugins.go:69] No cloud provider specified.
kubelet: I1102 11:12:57.693122    3354 docker.go:295] Connecting to docker on unix:///var/run/docker.sock
kubelet: I1102 11:12:57.693374    3354 server.go:673] Watching apiserver
kubelet: I1102 11:12:57.748466    3354 plugins.go:56] Registering credential provider: .dockercfg
kubelet: I1102 11:12:58.464878    3354 server.go:635] Started kubelet
kubelet: I1102 11:12:58.465678    3354 server.go:63] Starting to listen on 0.0.0.0:10250
kubelet: E1102 11:12:58.465695    3354 kubelet.go:682] Image garbage collection failed: unable to find data for container /
kubelet: I1102 11:12:58.472750    3354 kubelet.go:702] Running in container "/kubelet"
kubelet: I1102 11:12:58.552428    3354 kubelet.go:821] Successfully registered node centos-minion
kubelet: I1102 11:12:58.637736    3354 factory.go:226] System is using systemd
kubelet: I1102 11:12:58.638206    3354 factory.go:234] Registering Docker factory
kubelet: I1102 11:12:58.638519    3354 factory.go:89] Registering Raw factory
kubelet: I1102 11:12:58.732962    3354 manager.go:946] Started watching for new ooms in manager
kubelet: I1102 11:12:58.733145    3354 oomparser.go:199] OOM parser using kernel log file: "/var/log/messages"
kubelet: I1102 11:12:58.733804    3354 manager.go:243] Starting recovery of all containers
docker: time="2015-11-02T11:12:58.736013161-08:00" level=info msg="GET /containers/kube-proxy/json"
docker: time="2015-11-02T11:12:58.736223698-08:00" level=error msg="Handler for GET /containers/{name:.*}/json returned error: no such id: kube-proxy"
docker: time="2015-11-02T11:12:58.736252273-08:00" level=error msg="HTTP Error" err="no such id: kube-proxy" statusCode=404
docker: time="2015-11-02T11:12:58.736695767-08:00" level=info msg="GET /containers/kubelet/json"
docker: time="2015-11-02T11:12:58.736791703-08:00" level=error msg="Handler for GET /containers/{name:.*}/json returned error: no such id: kubelet"
docker: time="2015-11-02T11:12:58.736807917-08:00" level=error msg="HTTP Error" err="no such id: kubelet" statusCode=404
kubelet: I1102 11:12:58.737721    3354 manager.go:248] Recovery completed
kubelet: I1102 11:12:58.741073    3354 status_manager.go:76] Starting to sync pod status with apiserver
kubelet: I1102 11:12:58.741106    3354 kubelet.go:1725] Starting kubelet main sync loop.
```

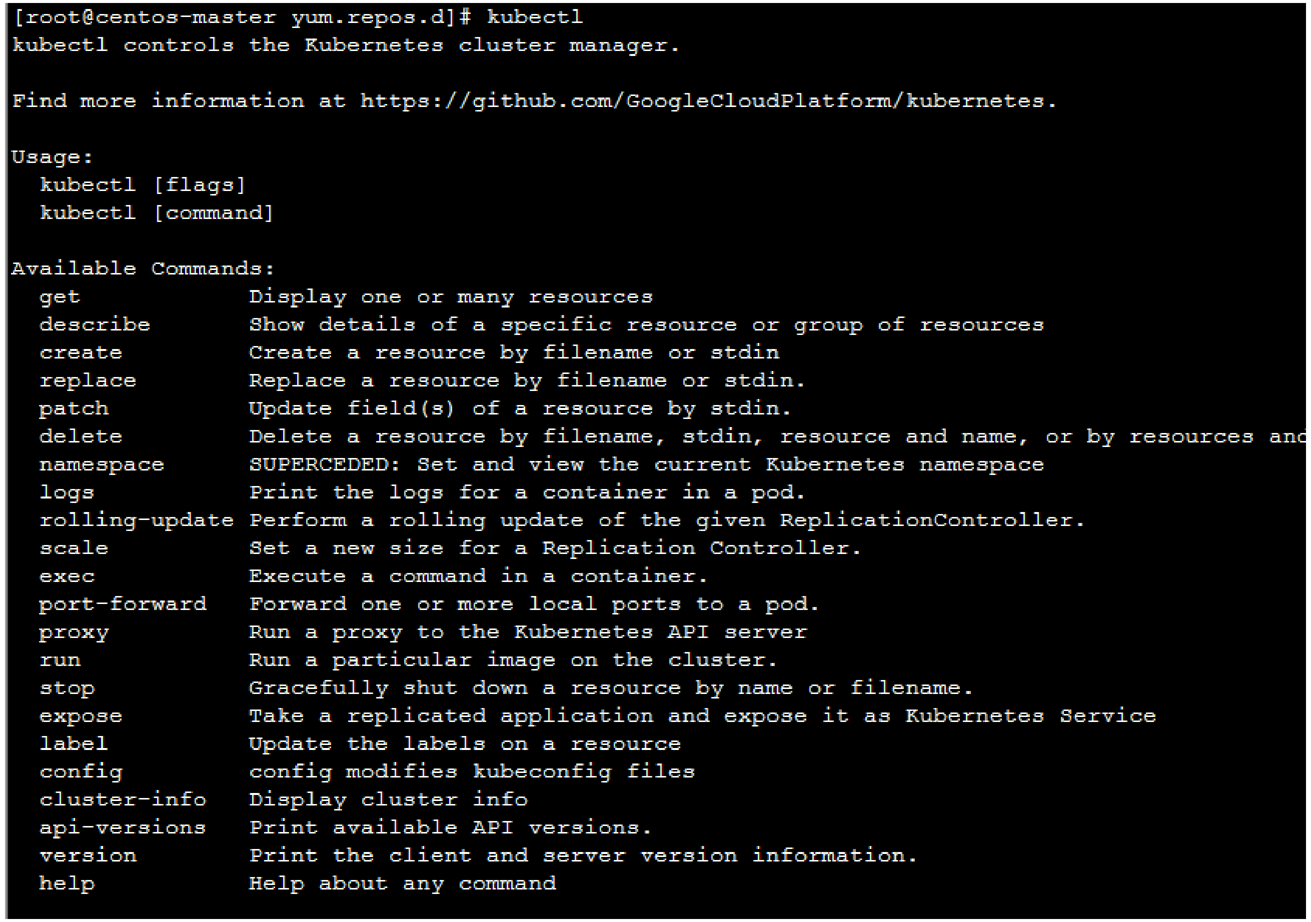
Kubernetes安装

在master节点执行kubectl get nodes
查看到节点注册成功，则表明系统安装正常

```
[root@centos-master yum.repos.d]# kubectl get nodes
NAME            LABELS                                  STATUS
centos-minion   kubernetes.io/hostname=centos-minion   Ready

[root@centos-master yum.repos.d]# kubectl cluster-info
Kubernetes master is running at http://localhost:8080
```

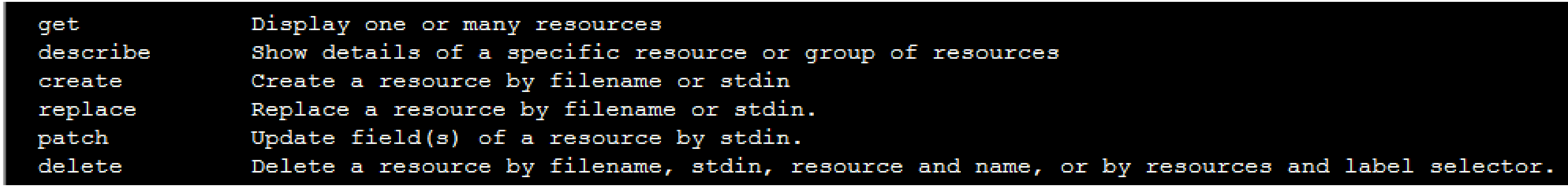
Kubernetes基本命令

Master上的kubectrl命令为管理集群的命令



```
[root@centos-master yum.repos.d]# kubectl
kubectl controls the Kubernetes cluster manager.

Find more information at https://github.com/GoogleCloudPlatform/kubernetes.

Usage:
  kubectl [flags]
  kubectl [command]

Available Commands:
  get            Display one or many resources
  describe       Show details of a specific resource or group of resources
  create         Create a resource by filename or stdin
  replace        Replace a resource by filename or stdin.
  patch          Update field(s) of a resource by stdin.
  delete         Delete a resource by filename, stdin, resource and name, or by resources and
  namespace      SUPERCEDED: Set and view the current Kubernetes namespace
  logs           Print the logs for a container in a pod.
  rolling-update Perform a rolling update of the given ReplicationController.
  scale          Set a new size for a Replication Controller.
  exec           Execute a command in a container.
  port-forward   Forward one or more local ports to a pod.
  proxy          Run a proxy to the Kubernetes API server
  run            Run a particular image on the cluster.
  stop           Gracefully shut down a resource by name or filename.
  expose         Take a replicated application and expose it as Kubernetes Service
  label          Update the labels on a resource
  config         config modifies kubeconfig files
  cluster-info   Display cluster info
  api-versions   Print available API versions.
  version        Print the client and server version information.
  help           Help about any command
```

Kubernetes基本命令

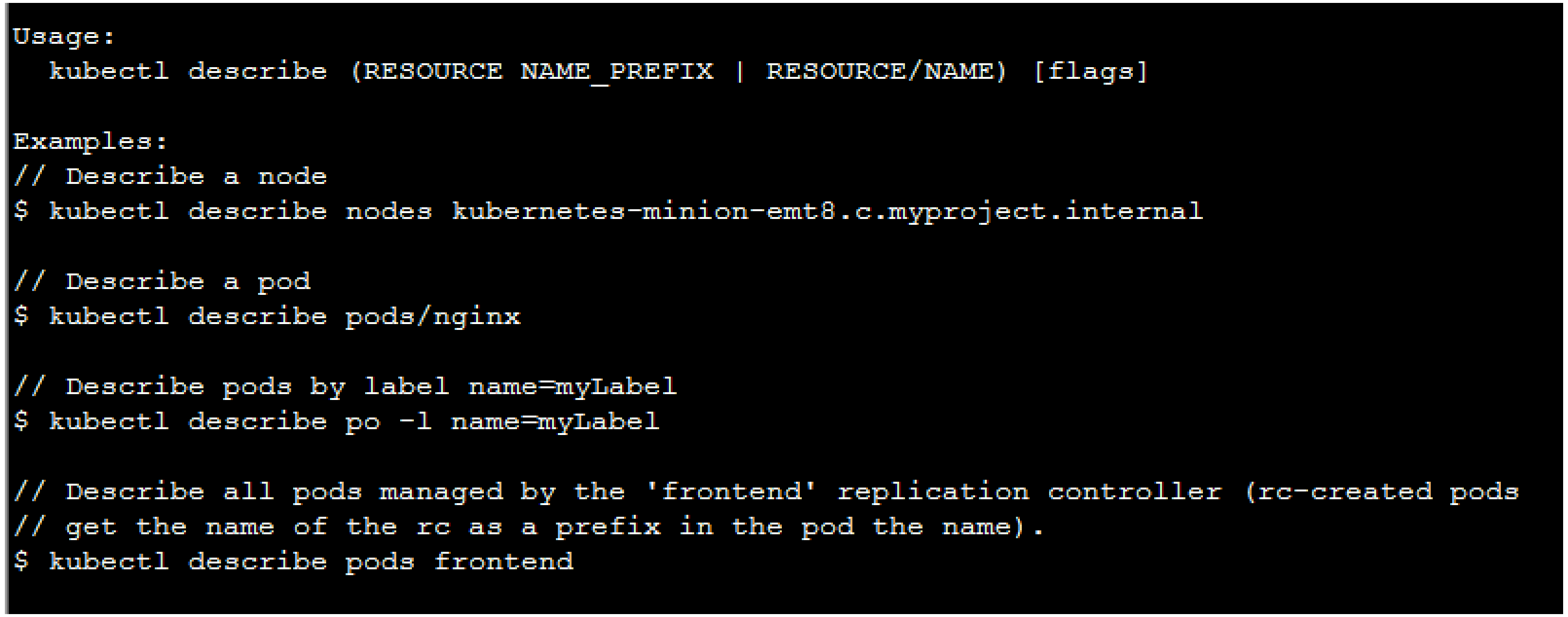


```
get             Display one or many resources
describe        Show details of a specific resource or group of resources
create          Create a resource by filename or stdin
replace         Replace a resource by filename or stdin.
patch           Update field(s) of a resource by stdin.
delete          Delete a resource by filename, stdin, resource and name, or by resources and label selector.
```

资源管理命令

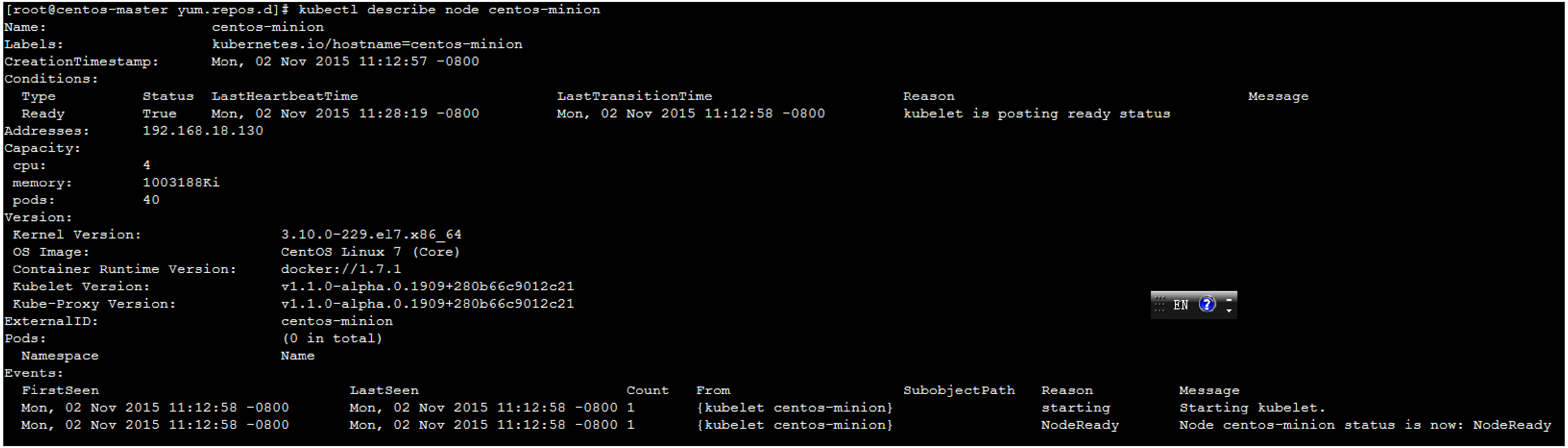
Kubernetes基本命令

kubectl describe --help

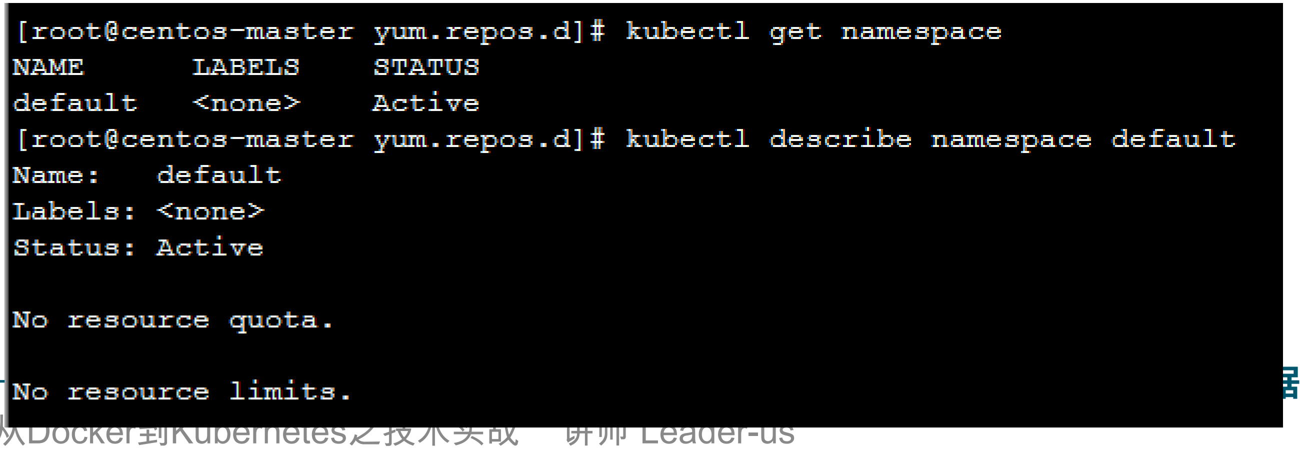


```
Usage:
  kubectl describe (RESOURCE NAME_PREFIX | RESOURCE/NAME) [flags]

Examples:
// Describe a node
$ kubectl describe nodes kubernetes-minion-emt8.c.myproject.internal

// Describe a pod
$ kubectl describe pods/nginx

// Describe pods by label name=myLabel
$ kubectl describe po -l name=myLabel

// Describe all pods managed by the 'frontend' replication controller (rc-created pods
// get the name of the rc as a prefix in the pod the name).
$ kubectl describe pods frontend
```

Kubernetes基本命令



```
[root@centos-master yum.repos.d]# kubectl describe node centos-minion
Name:                   centos-minion
Labels:                 kubernetes.io/hostname=centos-minion
CreationTimestamp:      Mon, 02 Nov 2015 11:12:57 -0800
Conditions:
  Type          Status  LastHeartbeatTime                       LastTransitionTime                      Reason                                  Message
  Ready         True    Mon, 02 Nov 2015 11:28:19 -0800         Mon, 02 Nov 2015 11:12:58 -0800         kubelet is posting ready status
Addresses:      192.168.18.130
Capacity:
 cpu:           4
 memory:        1003188Ki
 pods:          40
Version:
 Kernel Version:                3.10.0-229.el7.x86_64
 OS Image:                      CentOS Linux 7 (Core)
 Container Runtime Version:     docker://1.7.1
 Kubelet Version:               v1.1.0-alpha.0.1909+280b66c9012c21
 Kube-Proxy Version:            v1.1.0-alpha.0.1909+280b66c9012c21
ExternalID:                     centos-minion
Pods:                           (0 in total)
  Namespace                     Name
Events:
  FirstSeen                             LastSeen                        Count   From                    SubobjectPath   Reason          Message
  Mon, 02 Nov 2015 11:12:58 -0800       Mon, 02 Nov 2015 11:12:58 -0800 1       {kubelet centos-minion}                 starting        Starting kubelet.
  Mon, 02 Nov 2015 11:12:58 -0800       Mon, 02 Nov 2015 11:12:58 -0800 1       {kubelet centos-minion}                 NodeReady       Node centos-minion status is now: NodeReady
```



```
[root@centos-master yum.repos.d]# kubectl get namespace
NAME      LABELS    STATUS
default   <none>    Active
[root@centos-master yum.repos.d]# kubectl describe namespace default
Name:   default
Labels: <none>
Status: Active

No resource quota.

No resource limits.
```

Hello world

vi nginx-pod.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx
    ports:
    - containerPort: 80
```

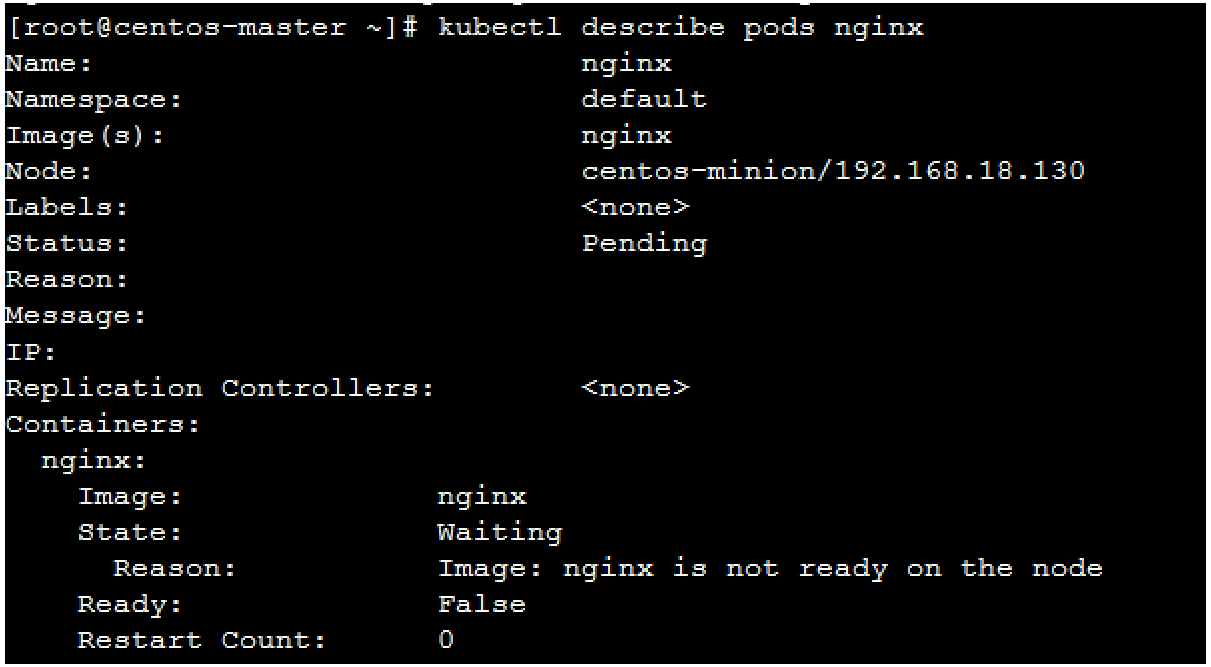
```
kubectl create -f nginx-pod.yaml
```

```
[root@centos-master ~]# kubectl create -f nginx-pod.yaml
Error from server: error when creating "nginx-pod.yaml":
Pod "nginx" is forbidden: no API token found for service
account default/default, retry after the token is
automatically created and added to the service account
```

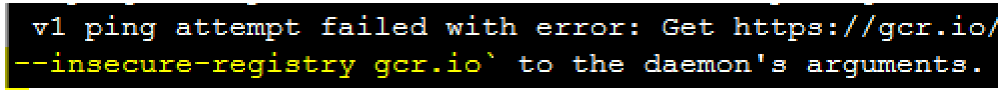
```
KUBE_ADMISSION_CONTROL="--admission_control=NamespaceLifecycle,NamespaceExists,LimitRanger,SecurityContextDeny,ServiceAccount,ResourceQuota"
```

Hello world

```
[root@centos-master ~]# kubectl get pods
NAME      READY     STATUS                               RESTARTS   AGE
nginx     0/1       Image: nginx is not ready on the node   0          56s
```



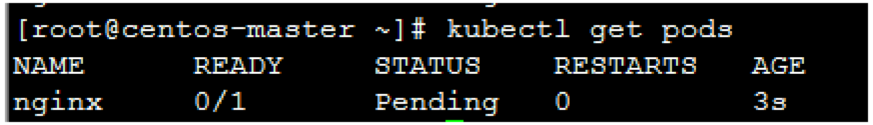
```
[root@centos-master ~]# kubectl describe pods nginx
Name:                           nginx
Namespace:                      default
Image(s):                       nginx
Node:                           centos-minion/192.168.18.130
Labels:                         <none>
Status:                         Pending
Reason:
Message:
IP:
Replication Controllers:        <none>
Containers:
  nginx:
    Image:              nginx
    State:              Waiting
      Reason:           Image: nginx is not ready on the node
    Ready:              False
    Restart Count:      0
```



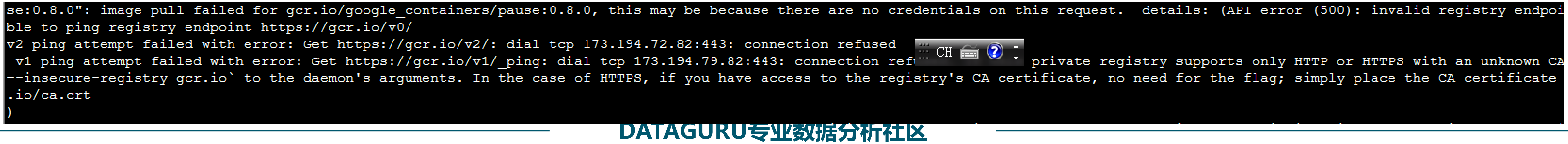
```
 v1 ping attempt failed with error: Get https://gcr.io/
--insecure-registry gcr.io` to the daemon's arguments.
```

/etc/sysconfig/docker

INSECURE_REGISTRY='--insecure-registry gcr.io'

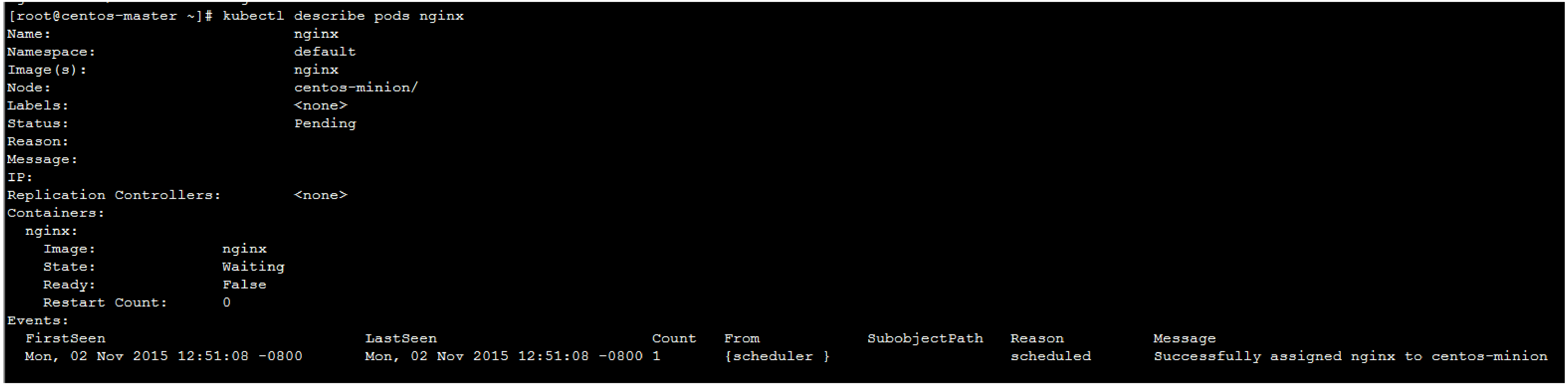


```
[root@centos-master ~]# kubectl get pods
NAME         READY       STATUS      RESTARTS     AGE
nginx        0/1         Pending     0            3s
```

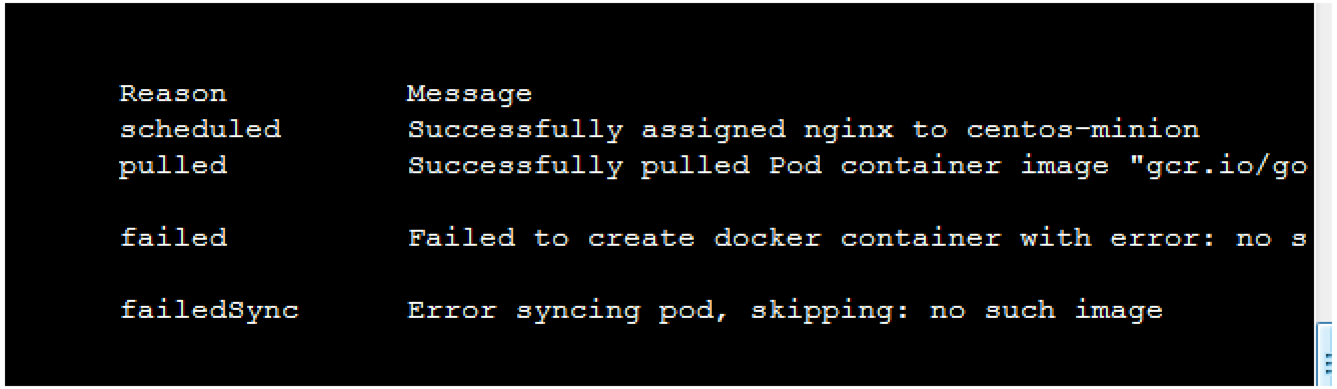


```
se:0.8.0": image pull failed for gcr.io/google_containers/pause:0.8.0, this may be because there are no credentials on this request.  details: (API error (500): invalid registry endpoi
ble to ping registry endpoint https://gcr.io/v0/
v2 ping attempt failed with error: Get https://gcr.io/v2/: dial tcp 173.194.72.82:443: connection refused
 v1 ping attempt failed with error: Get https://gcr.io/v1/_ping: dial tcp 173.194.79.82:443: connection ref          private registry supports only HTTP or HTTPS with an unknown CA
--insecure-registry gcr.io` to the daemon's arguments. In the case of HTTPS, if you have access to the registry's CA certificate, no need for the flag; simply place the CA certificate
.io/ca.crt
)
```

Hello world

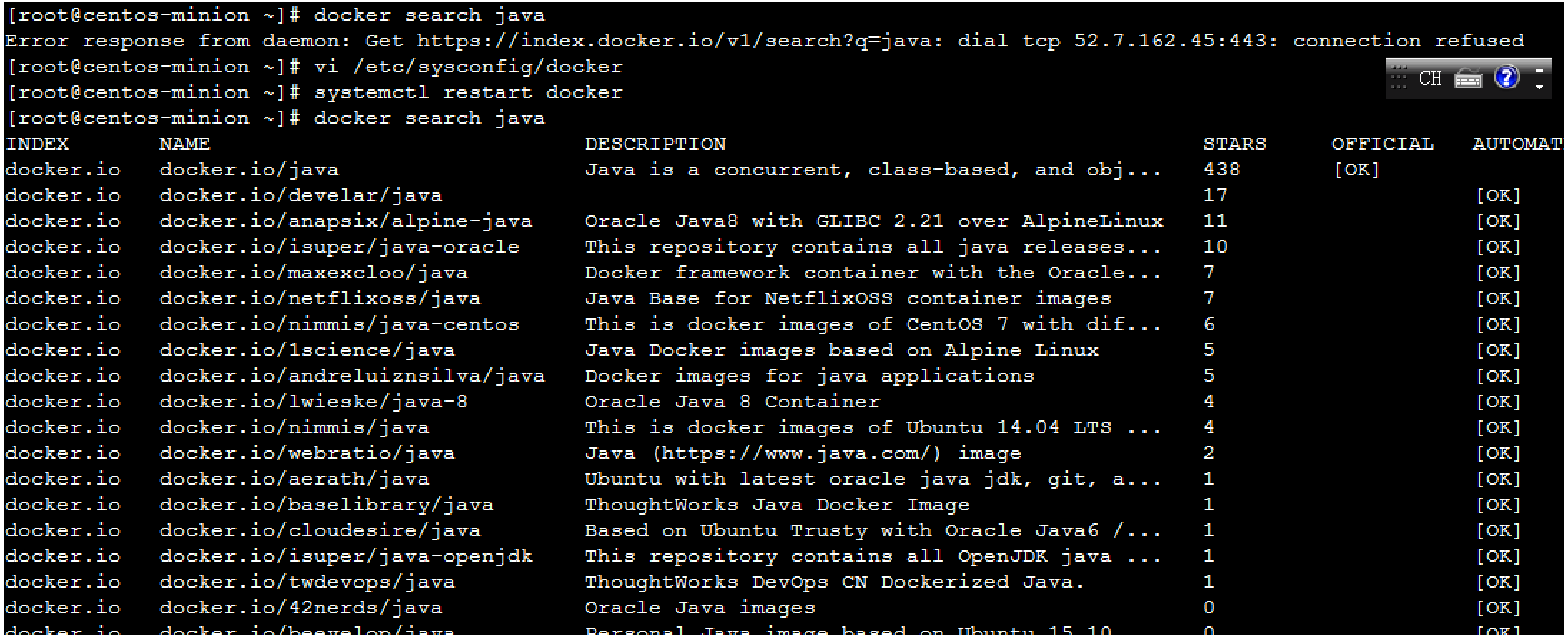


```
[root@centos-master ~]# kubectl describe pods nginx
Name:                           nginx
Namespace:                      default
Image(s):                       nginx
Node:                           centos-minion/
Labels:                         <none>
Status:                         Pending
Reason:
Message:
IP:
Replication Controllers:        <none>
Containers:
  nginx:
    Image:              nginx
    State:              Waiting
    Ready:              False
    Restart Count:      0
Events:
  FirstSeen                             LastSeen                        Count   From            SubobjectPath   Reason          Message
  Mon, 02 Nov 2015 12:51:08 -0800       Mon, 02 Nov 2015 12:51:08 -0800 1       {scheduler }                    scheduled       Successfully assigned nginx to centos-minion
```



```
Reason          Message
scheduled       Successfully assigned nginx to centos-minion
pulled          Successfully pulled Pod container image "gcr.io/go

failed          Failed to create docker container with error: no s

failedSync      Error syncing pod, skipping: no such image
```

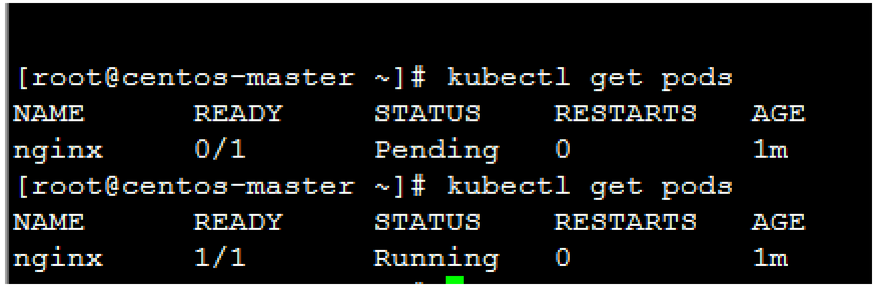
Hello world



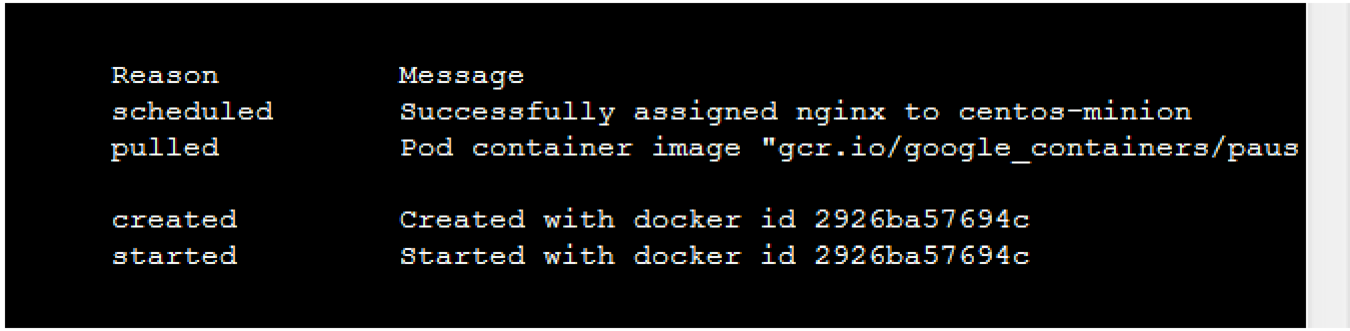
```
[root@centos-minion ~]# docker search java
Error response from daemon: Get https://index.docker.io/v1/search?q=java: dial tcp 52.7.162.45:443: connection refused
[root@centos-minion ~]# vi /etc/sysconfig/docker
[root@centos-minion ~]# systemctl restart docker
[root@centos-minion ~]# docker search java
INDEX       NAME                                 DESCRIPTION                                     STARS     OFFICIAL   AUTOMAT
docker.io   docker.io/java                       Java is a concurrent, class-based, and obj...   438        [OK]
docker.io   docker.io/develar/java                                                               17                    [OK]
docker.io   docker.io/anapsix/alpine-java        Oracle Java8 with GLIBC 2.21 over AlpineLinux   11                    [OK]
docker.io   docker.io/isuper/java-oracle         This repository contains all java releases...   10                    [OK]
docker.io   docker.io/maxexcloo/java             Docker framework container with the Oracle...   7                     [OK]
docker.io   docker.io/netflixoss/java            Java Base for NetflixOSS container images       7                     [OK]
docker.io   docker.io/nimmis/java-centos         This is docker images of CentOS 7 with dif...   6                     [OK]
docker.io   docker.io/1science/java              Java Docker images based on Alpine Linux        5                     [OK]
docker.io   docker.io/andreluiznsilva/java       Docker images for java applications             5                     [OK]
docker.io   docker.io/lwieske/java-8             Oracle Java 8 Container                         4                     [OK]
docker.io   docker.io/nimmis/java                This is docker images of Ubuntu 14.04 LTS ...   4                     [OK]
docker.io   docker.io/webratio/java              Java (https://www.java.com/) image              2                     [OK]
docker.io   docker.io/aerath/java                Ubuntu with latest oracle java jdk, git, a...   1                     [OK]
docker.io   docker.io/baselibrary/java           ThoughtWorks Java Docker Image                  1                     [OK]
docker.io   docker.io/cloudesire/java            Based on Ubuntu Trusty with Oracle Java6 /...   1                     [OK]
docker.io   docker.io/isuper/java-openjdk        This repository contains all OpenJDK java ...   1                     [OK]
docker.io   docker.io/twdevops/java              ThoughtWorks DevOps CN Dockerized Java.         1                     [OK]
docker.io   docker.io/42nerds/java               Oracle Java images                              0                     [OK]
docker.io   docker.io/beevelop/java              Personal Java image based on Ubuntu 15.10       0                     [OK]
```

DATAGURU专业数据分析社区

Hello world



```
[root@centos-master ~]# kubectl get pods
NAME        READY       STATUS      RESTARTS    AGE
nginx       0/1         Pending     0           1m
[root@centos-master ~]# kubectl get pods
NAME        READY       STATUS      RESTARTS    AGE
nginx       1/1         Running     0           1m
```



```
    Reason          Message
    scheduled       Successfully assigned nginx to centos-minion
    pulled          Pod container image "gcr.io/google_containers/paus

    created         Created with docker id 2926ba57694c
    started         Started with docker id 2926ba57694c
```

大功告成

Thanks

FAQ时间