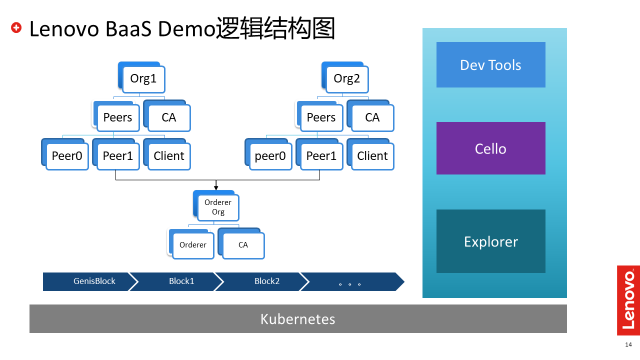
# Lenovo BaaS Demo演示

Lenovo BaaS集群部署采用Kubeadm方式自动部署，我们这里使用二进制包部署，可以屏蔽很多细节，使得对各个Lenovo BaaS模块感知很少，也可以迅速上手。

1. 架构拓扑图



1. 环境规划

|  |  |  |
| --- | --- | --- |
| 角色 | IP | 组件 |
| Master | 10.100.47.49 | etcd  kube-apiserver  kube-controller-manager  kube-scheduler |
| Node1 | 10.100.47.98 | kubelet  kube-proxy  docker |

环境说明：

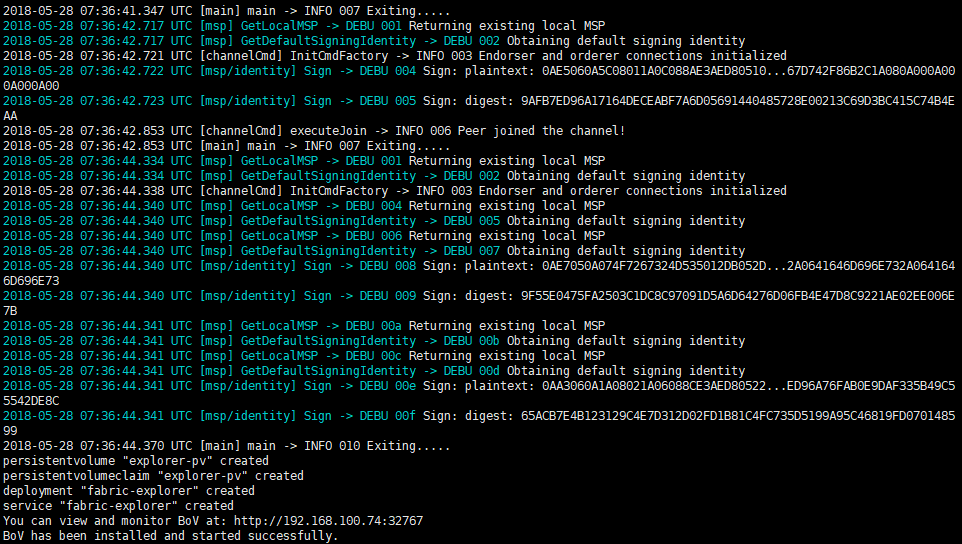
操作系统：Ubuntu16.04

Kubernetes版本：v1.9.1

Docker版本：v17.09-ce

均采用当前最新稳定版本。

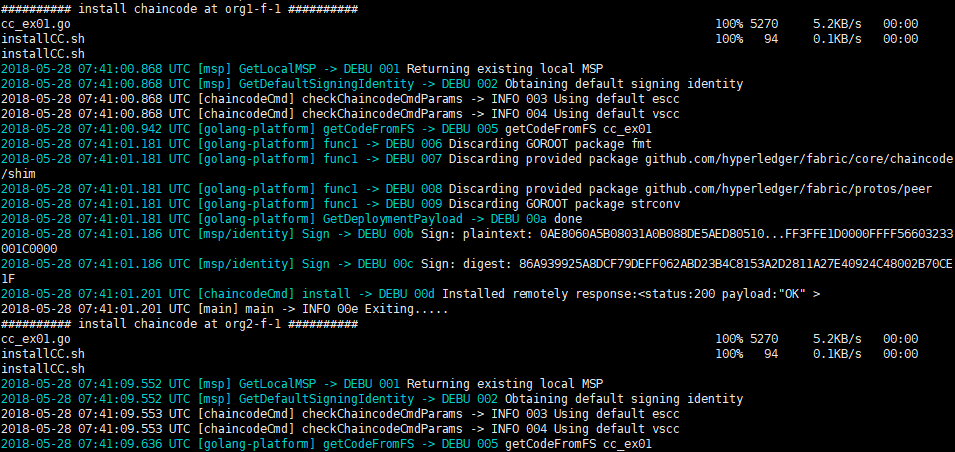
1. 建立Lenovo BaaS网络
2. 登陆到Demo主机（10.100.47.49：root/qwer!@#$）
3. cd /home/ubuntu/baas/setupCluster, 修改cluster-config.yaml,定义组织结构
4. 运行命令： bash install.sh建立网络



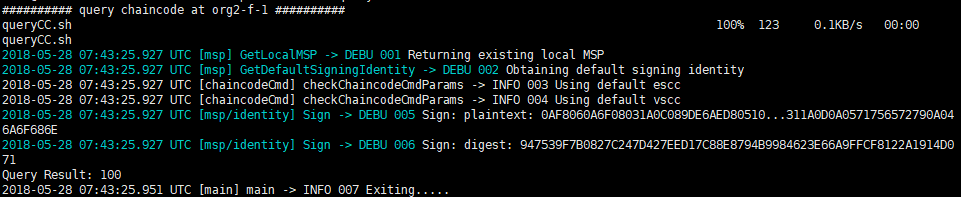
1. chaincode操作

Lenovo BaaS网络环境搭建好后，下面可以开始实验chiancode的操作了，依次采用一下步骤

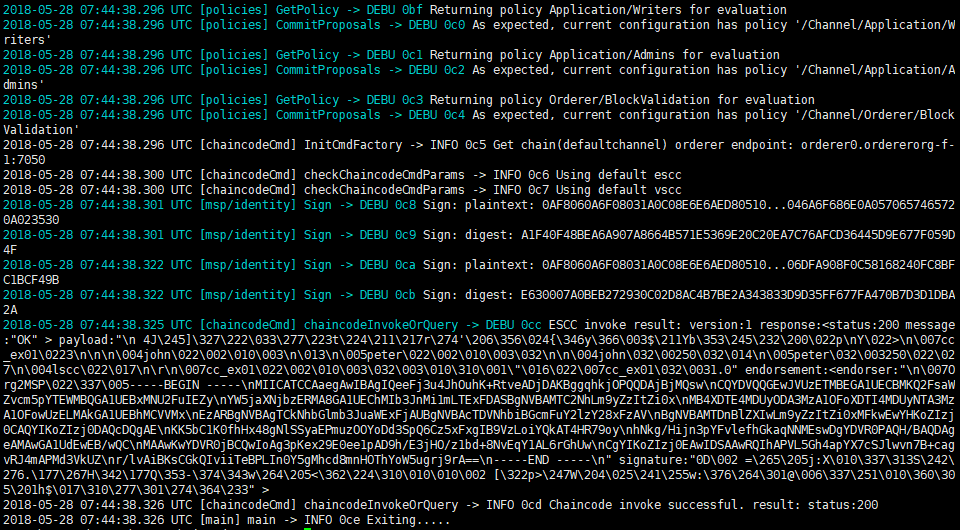
1. 安装
   1. 依然登陆到Demo主机（10.100.47.49：root/123456）
   2. cd /home/ubuntu/chaincodeOperater
   3. bash install.sh



1. 查询
2. cd /home/ubuntu/chaincodeOperater
3. bash query.sh



1. 调用
2. cd /home/ubuntu/chaincodeOperater
3. bash invoke.sh



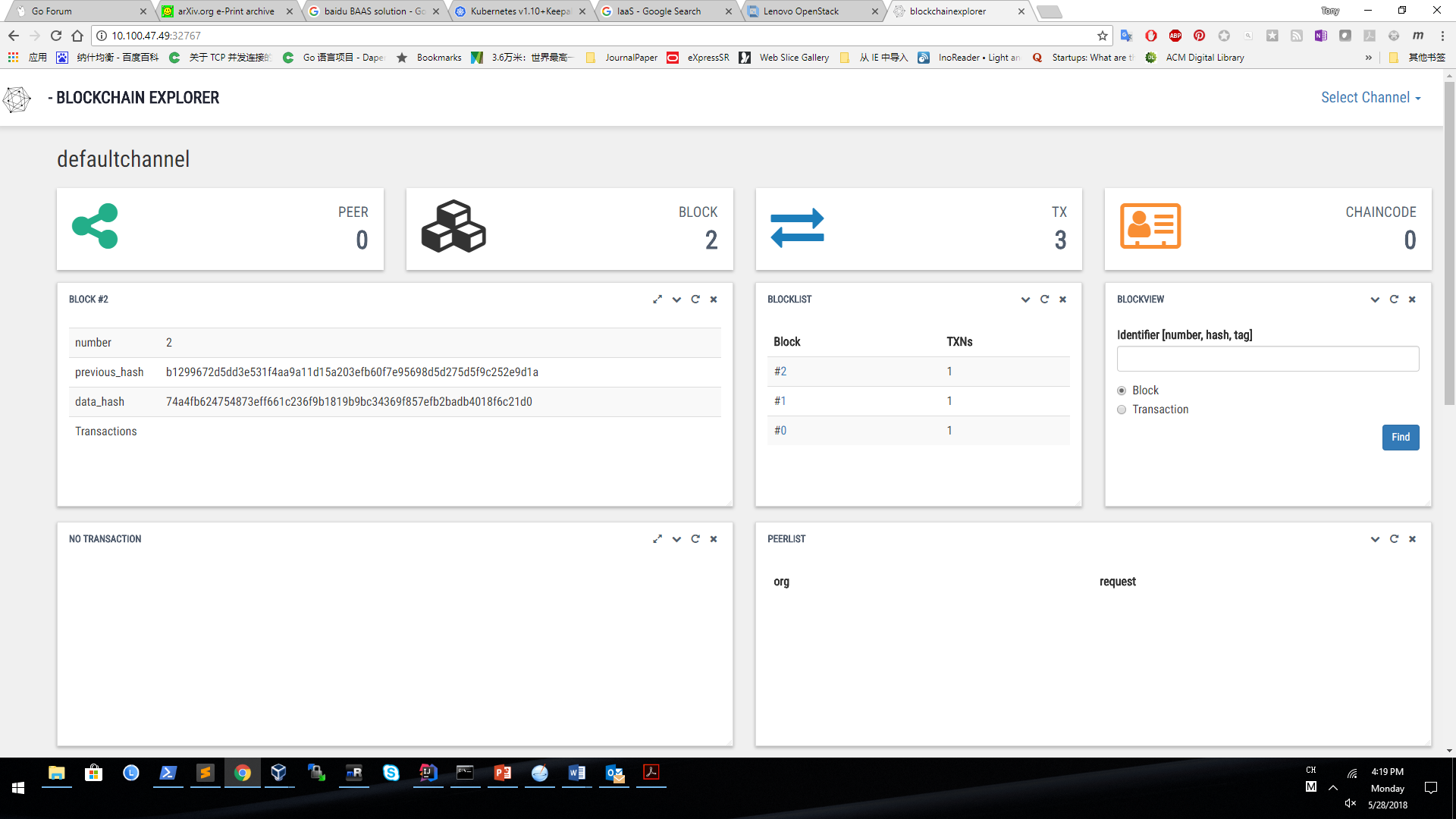
1. 通过explorer查看blockchain相关信息

至此，我们已经对chaincode进行了一轮操作，包括安装，初始化，查询，调用。从第四步开始，我们就可以通过explorer对我们部署的blockchain进行观察了。

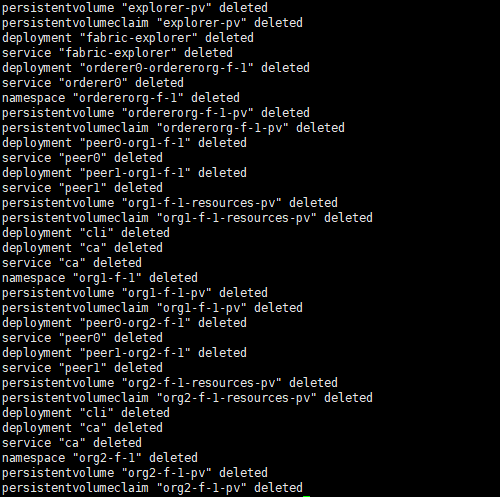
使用第四步屏幕输出最后提示信息：

You can view and monitor BoV at: http://192.168.100.74:32767，将ip地址替换为10.100.47.49。

从浏览器访问http://10.100.47.49:32767/



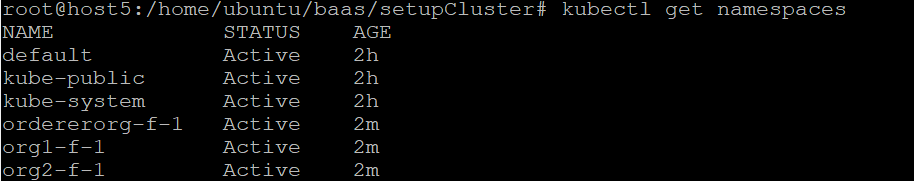
1. 删除已建立的fabric网络
2. 登陆到Demo主机（10.100.47.49：root/123456）
3. cd /home/ubuntu/baas/setupCluster
4. 运行命令： bash stop.sh建立网络



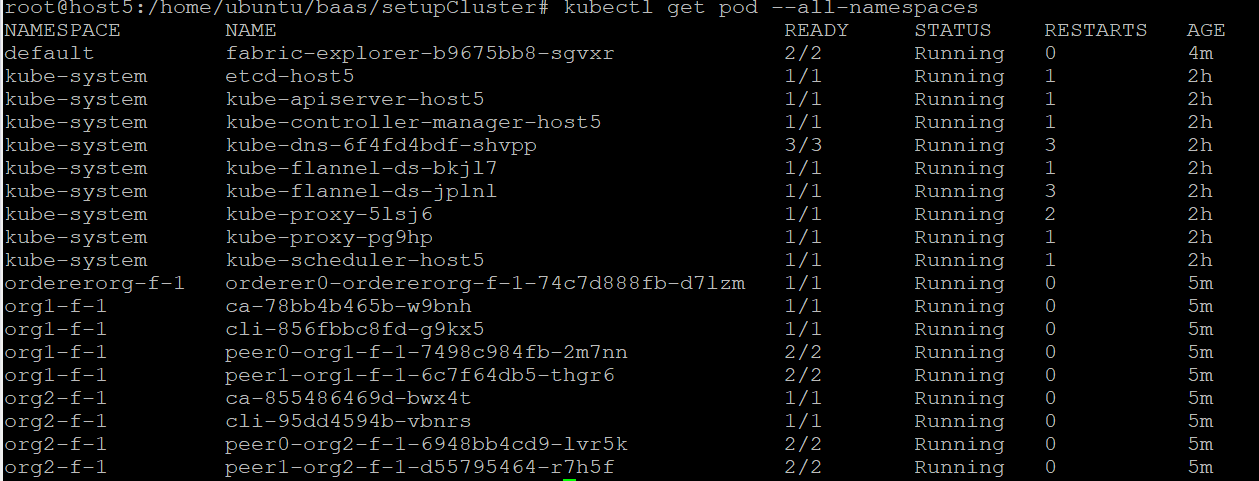
1. 诊断手段

运行Lenovo BaaS过程中，可能会使用到以下诊断手段

1. kubectl get namespaces



1. kubectl get pod --all-namespaces



1. kubeCleaner.sh
2. cd /home/ubuntu/baas/setupCluster
3. bash kubeCleaner.sh 清除没有正常回收的pod资源

### cello

1. 环境规划

|  |  |  |
| --- | --- | --- |
| 角色 | IP | 组件 |
| Master | 10.100.47.98 | cello-master |
| Worker | 10.100.47.49 | cello-worker |

环境说明：

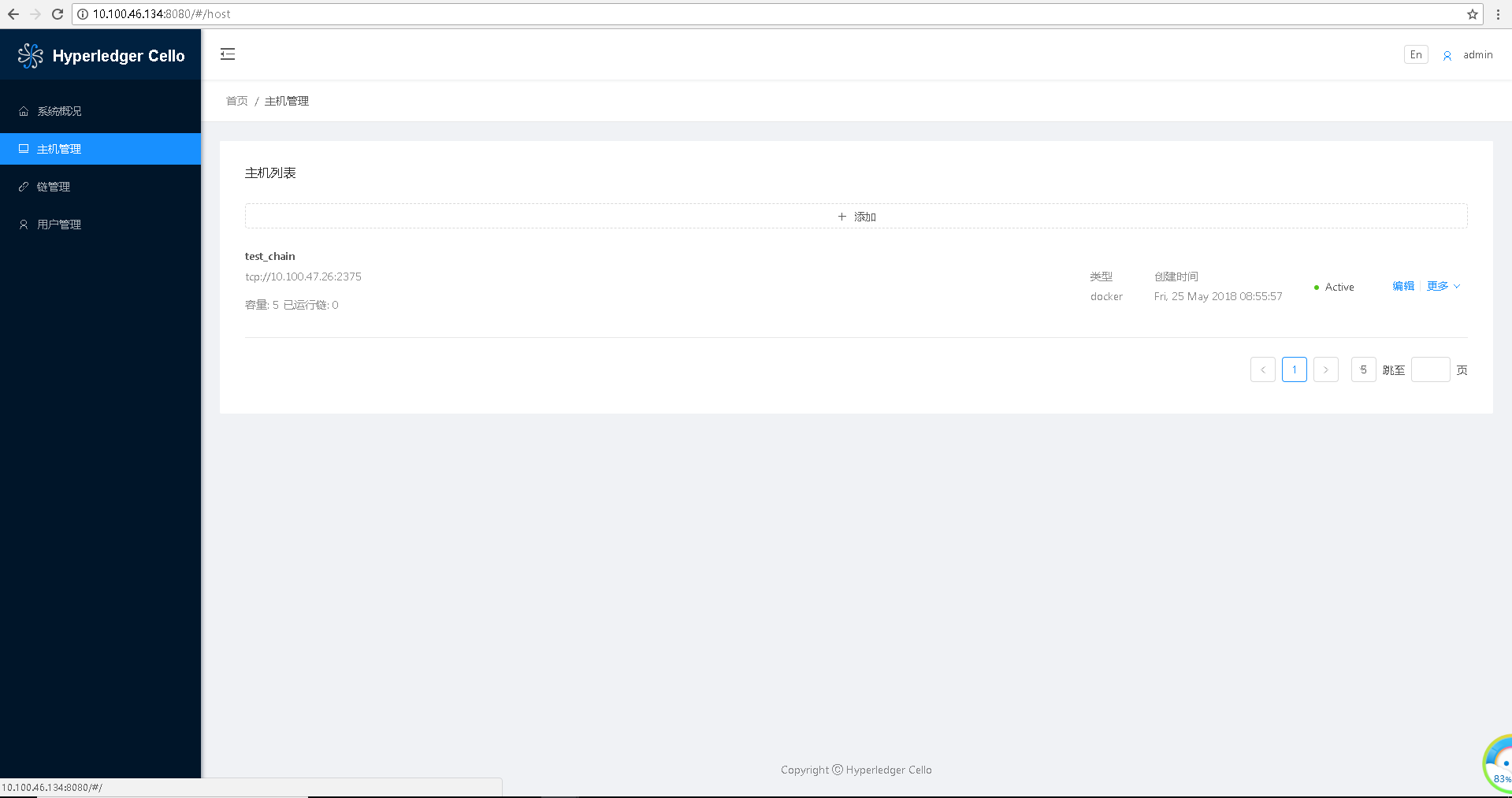
操作系统：Ubuntu16.04

Docker版本：v17.09-ce

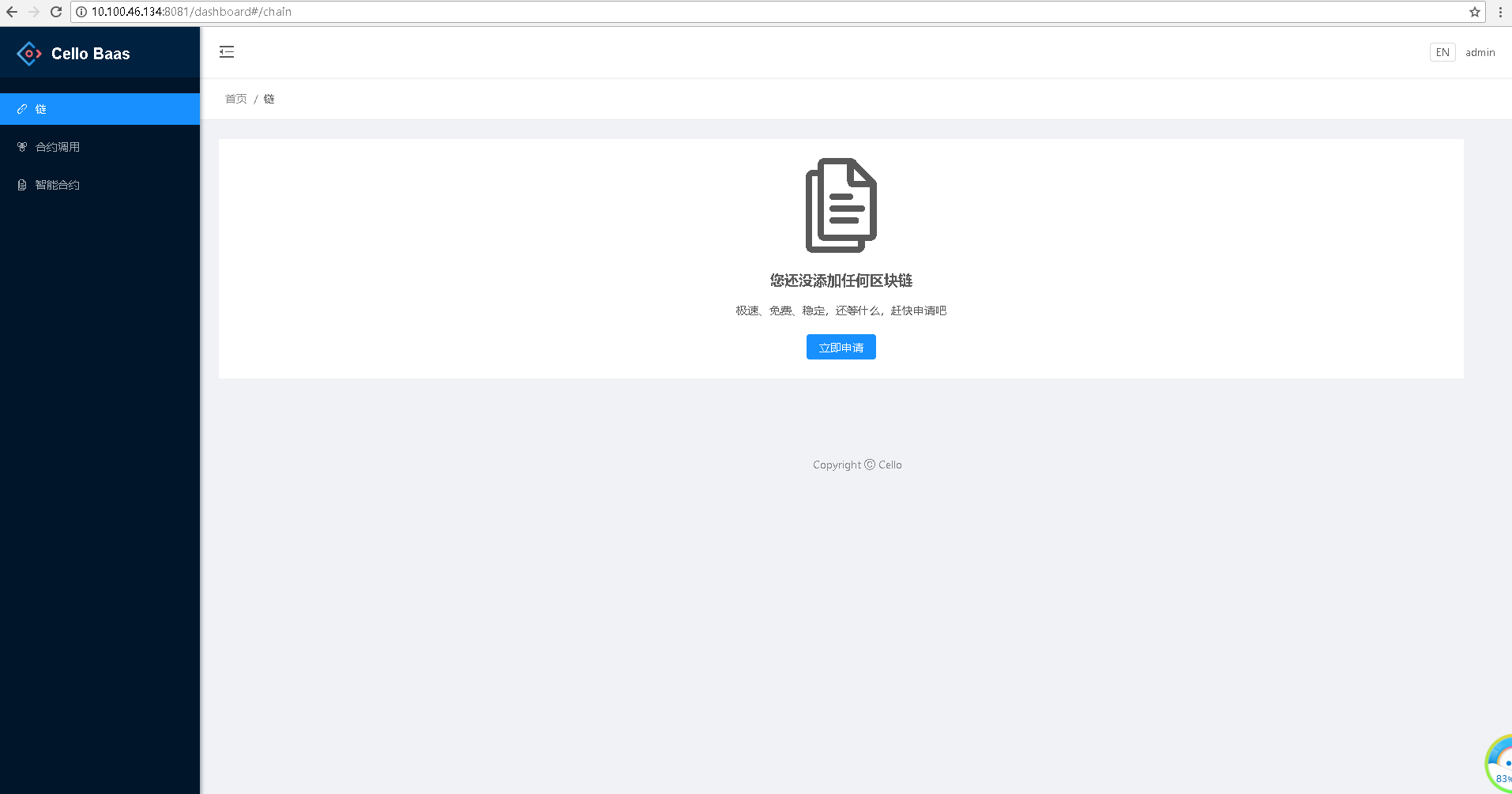
Docke-compose版本：v 1.17.0

2. 通过访问http:// 10.100.47.98:8080 查看管理页面，默认账号密码为：admin/pass

在主机管理页面，添加worker节点。



3. 通过访问http:// 10.100.47.98:8081 查看用户的dashboard



4、常用命令

cd /home/ubuntu/cello/ 进入cello文件夹下

make start *// 启动cello服务*

make restart *// 重启cello服务*

make **stop** *// 停止cello服务*

make logs *// 查看日志*