Kubernetes容器集群部署工作

1. 大概采取的是这个网页上面的方法<http://kubernetes.io/v1.0/docs/getting-started-guides/ubuntu.html>
2. 前提条件：四台机器，ip：202.120.40.177 端口是：10122，10222，10322，10422. 对应的内网ip是192.168.1.101~192.168.1.104  
   KUBE\_VERSION:-"1.0.6"
3. 具体工作：  
   $ git clone <https://github.com/GoogleCloudPlatform/kubernetes.git>  
   $ cd kubernetes/cluster/Ubuntu  
   $ ./build.sh  
   完成上面的指令之后，会在目录下出现binaries文件夹  
     
   接下来要修改/kubernetes/cluster/ubuntu中的config-default.sh文件以完成配置：

export nodes=${nodes:-"administrator@192.168.1.101 administrator@192.168.1.102 administrator@192.168.1.103 administrator@192.168.1.104"}

role=${role:-"a i i i"} /////a表示master，i表示节点，ai表示即是master又是节点。m

export roles=($role)

export NUM\_MINIONS=${NUM\_MINIONS:-3} /////表示节点的数量

export SERVICE\_CLUSTER\_IP\_RANGE=${SERVICE\_CLUSTER\_IP\_RANGE:-192.168.3.0/24}

export FLANNEL\_NET=${FLANNEL\_NET:-172.16.0.0/16}  
下一步是回到cluster目录下，执行命令，来启动kubernetes：

KUBERNETES\_PROVIDER=ubuntu ./kube-up.sh

但是一开始就报错，提示ssh出错，需要启动ssh-agent，使用ssh gen key的方法即可解决

Move on!!!

 重新执行，需要一路的输入密码，但是在过程中还是出现了问题，在启动master节点的etc过程中出现了迷之卡顿，发现是在cluster/saltbase/salt/ generate-cert中的make-ca-cert.sh中需要下载一个包 ：curl -L -O <https://storage.googleapis.com/kubernetes-release/easy-rsa/easy-rsa.tar.gz> > /dev/null 2>&1

然而那个网站会抽筋，所以改用别处下载好包，放在某个地方（/tmp）让它自己去取来用: 在原来curl -L -O <https://storage.googleapis.com/kubernetes-release/easy-rsa/easy-rsa.tar.gz> > /dev/null 2>&1 的地方改成cp /tmp/easy-rsa.tar.gz ${tmpdir}

成功！！

继续执行，结果发现，继续运行发生cp opmitting问题发现是

kubernetes/cluster/ubuntu中的util.sh一句话写错了，文件夹需要的是cp -r 而不是cp 语句是provision-master中的sudo cp ~/kube/master/\* /opt/bin/ 改成sudo cp -r ~/kube/master/\* /opt/bin/

然后迷之卡顿消失，master节点的顺利完成

然而还有一个问题

echo "It looks as if you don't have a compiled kubectl binary"

echo

echo "If you are running from a clone of the git repo, please run"

echo "'./build/run.sh hack/build-cross.sh'. Note that this requires having"

echo "Docker installed."

echo

echo "If you are running from a binary release tarball, something is wrong. "

echo "Look at http://kubernetes.io/ for information on how to contact the "

echo "development team for help."

后来发现是cluster 里面的kubectl.sh中文件配置失误，所以进行修改在第101行插入这句话

"${KUBE\_ROOT}/cluster/ubuntu/binaries/kubectl"

之后顺利执行了~~~

需要记住的是，每次重新执行KUBERNETES\_PROVIDER=ubuntu ./kube-up.sh这一句之前需要把中途下载的kube包，bin包删了，同时关闭etcd 语句是rm -r ~/kube ,sudo rm -r /opt/bin/,sudo service etcd stop

成功运行之后在validating node 的时候出现了问题，经过看了代码发现是node上面没有装docker，依次在三台node上面装完docker之后，重新运行指令，出现以下结果：

Found 3 node(s).

NAME LABELS STATUS

192.168.1.102 kubernetes.io/hostname=192.168.1.102 Ready

192.168.1.103 kubernetes.io/hostname=192.168.1.103 Ready

192.168.1.104 kubernetes.io/hostname=192.168.1.104 Ready

Validate output:

NAME STATUS MESSAGE ERROR

controller-manager Healthy ok nil

scheduler Healthy ok nil

etcd-0 Healthy {"health": "true"} nil

Nice！！！！

集群基本搭设完成~~

后记，这个版本的kubernetes不支持1.8+版本的docker，所以小心

后记2，由于原先的方法：apt-get install docker.io

in –sf /usr/bin/docker/io /usr/local/bin/docker

默认安装的是dockerv1.0.1，会在后面进行server连接的时候不兼容kubernetes，需要使用以下命令进行升级，目前升级到了1.7.0-dev

：add-apt-repository ppa:docker-maint/testing

apt-get update

apt-get install docker.io 然后docker升级成功：

Client version: 1.7.0-dev

Client API version: 1.19

Go version (client): go1.4.2

Git commit (client): 5e06332

OS/Arch (client): linux/amd64

Server version: 1.7.0-dev

Server API version: 1.19

Go version (server): go1.4.2

Git commit (server): 5e06332

OS/Arch (server): linux/amd64