使用acs-engine 部署Kubernetes集群

说明：由于acs-engine产品在不断开发改进中，现有配置文件方法在后续不一定能成功。建议用户参考Github [https://github.com/Azure/acs-engine](https://na01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2FAzure%2Facs-engine&data=02%7C01%7Clqi%40microsoft.com%7Cab82e0d9396249e50f5f08d521bf7e1c%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C636452028751478290&sdata=4uaXw0pomUB8tOXRgr5asNyAywadvBQjxwHtUu%2BTe%2FM%3D&reserved=0) 了解最新进展，交流技术问题。

现阶段关于kebernetes的架构请参考链接：<https://github.com/Azure/acs-engine/blob/master/docs/kubernetes/walkthrough.md>

1. 安装acs-engine

建议在Ubuntu 16.04 中使用acs-engine 0.8.0进行部署。该系统版本是经过测试的较稳定的版本，其他Linux发行版本如CentOS可能在后续Kubernetes管理中有一些问题。若使用acs-engine的其他版本，使用的Kubernetes.json文件格式可能有所不同。

您也可以用源码进行编译安装，本文对该部分不做介绍， 参考<https://github.com/Azure/acs-engine/blob/master/docs/acsengine.md#build-acs-engine-from-source>

acs-engine下载地址：[https://github.com/Azure/acs-engine/releases/tag/v0.8.0](https://na01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2FAzure%2Facs-engine%2Freleases%2Ftag%2Fv0.8.0&data=02%7C01%7Clqi%40microsoft.com%7Cab82e0d9396249e50f5f08d521bf7e1c%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C636452028751478290&sdata=B1c9qMbykJY9lhNDHpQPVng28X26DZgX9V4Mrsk3f9w%3D&reserved=0)

*$wget* [*https://github.com/Azure/acs-engine/releases/download/v0.8.0/acs-engine-v0.8.0-linux-amd64.tar.gz*](https://na01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2FAzure%2Facs-engine%2Freleases%2Fdownload%2Fv0.8.0%2Facs-engine-v0.8.0-linux-amd64.tar.gz&data=02%7C01%7Clqi%40microsoft.com%7Cab82e0d9396249e50f5f08d521bf7e1c%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C636452028751478290&sdata=%2FYnxbeNEukp%2FnbevW%2BNSAybIyLLKkakRGnrn1G6TYYk%3D&reserved=0)

*$tar zxvf acs-engine-v0.8.0-linux-amd64.tar.gz*

*$cd acs-engine-v0.8.0-linux-amd64/*

1. 创建服务主体。

您需要首先安装Azure CLI 2.0。安装方法请参考：<https://docs.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest>

*$az cloud set -n AzureChinaCloud*

*$ az login  -u* [*test@test.partner.onmschina.cn*](mailto:test@test.partner.onmschina.cn) *-p* [*yourpassword*](mailto:lickky.1989@W)

*$ az account set --subscription  "12345678-xxxx-xxxx-1234567890"*

*$az group create -n lqik8s07 -l chinanorth*

*$az ad sp create-for-rbac --role="Contributor" --scopes="/subscriptions/12345678-xxxx-xxxx-1234567890/resourceGroups/lqik8s07"*

{  
  "appId": "1d061005-xxxx-xxxx-xxxx-e17b62daacb0",  
  "displayName": "azure-cli-2017-07-27-06-55-01",  
  "name": "<http://azure-cli-2017-07-27-06-55-01>",  
  "password": "e7efcbff-xxxx-xxxx-xxxx-4d1e5c4ca0a0",  
  "tenant": "89e1b688-xxxx-xxxx-xxxx-54d0a43a4f0d"  
}

这里生成的appid是后面kubernetes.json模板中的ClientID，password为Secret。

注意：为了安全考虑，这里仅将服务主体的权限赋予了特定资源组，该资源组将用于创建Kubernetes的所有Azure资源。如果您需要将kubernetes集群放在其他资源组，您需要额外再进行权限设置，否则在后续服务配置中会遇到权限拒绝等错误。设置方法为：

$az role assignment create --role "Contributor" --assignee "appid" --scope /subscriptions/12345678-xxxx-xxxx-1234567890/resourceGroups/lqik8s08

为了测试方便，您也可以直接将权限赋予整个订阅，但我们不推荐在生产环境中这样做。方法如下：

$az role assignment create --role "Contributor" --assignee "appid" --scope /subscriptions/12345678-xxxx-xxxx-1234567890

1. 编辑模板

在使用该模板前，您应该已经完成了第二步，并已经准备好了SSH 公钥和密钥对，这将用于登录您的Kubernetes master节点。

Kubernetes.json 模板如下，将高亮部分改成您的值，将文件保存在acs-engine命令的目录中。

{

  "apiVersion": "vlabs",

  "location": "yourlocation",

  "properties": {

    "orchestratorProfile": {

      "orchestratorType": "Kubernetes",

      "orchestratorRelease": "1.6"

    },

    "masterProfile": {

      "count": 1,

      "dnsPrefix": "yourprefix",

      "vmSize": "Standard\_F2"

    },

    "agentPoolProfiles": [

      {

        "name": "agentpool1",

        "count": 2,

        "vmSize": "Standard\_F2",

        "availabilityProfile": "AvailabilitySet"

      }

    ],

    "linuxProfile": {

      "adminUsername": "yourusername",

      "ssh": {

        "publicKeys": [

          {

            "keyData": "yourkey"

          }

        ]

      }

    },

    "servicePrincipalProfile": {

      "ClientID": "appid",

      "Secret": "password "

    }

  }

}

              说明：

* location 决定了后续kubernetes master节点的DNS名对应的Azure终结点是否正确，请确保一定要填写正确。在中国，可选值应该为Chinanorth 或者Chinaeast。
* dnsprefix决定了输出的文件夹，会在当前目录的\_output下生成一个同名的文件夹，其中包括部署kubernetes集群的参数文件。
* 您可以调整模板中masterProfile 和agentPoolProfiles，修改其count和VMsize为您实际的需求。
* 不同版本的acs-engine的kubernetes.json模板中具体的参数名可能有所调整，若不确定，建议在Github中查找或提交问题。

1. 生成Azure模板

确认您使用的acs-engine版本是正确的。这将决定生成的模板配置。

*$ ./acs-engine version*

*Version: v0.8.0*

*GitCommit: 79572455*

*GitTreeState: clean*

*$./acs-engine generate kubernetes.json*

*INFO[0000] Generating assets into \_output/lqi07...*

1. 编辑查看模板

在笔者测试阶段，为了在Azure中国区部署成功，部分模板参数需要做调整。在后续的开发中，这部分配置可能会被修正。请参考GitHub的最新进展。

切换到模板目录\_output/yourprefix，编辑azuredeploy.parameters.json。替换其中部分值如下：

$ cd \_output/lqi07

$ vi azuredeploy.parameters.json

                         修改“kubernetesHyperkubeSpec” 的值为：

**crproxy.trafficmanager.net:6000/google\_containers/hyperkube-amd64:v1.6.11**

修改“dockerEngineDownloadRepo” 的值为：

<https://mirror.kaiyuanshe.cn/docker-engine/apt/repo>

修改“kubernetesTillerSpec” 的值为：

**crproxy.trafficmanager.net:6000/kubernetes-helm/tiller:v2.6.1**

1. 部署Kubernetes集群

该过程大约需要20分钟左右，其中包括创建Azure资源如VNET，负载均衡，自定义路由，虚拟机，存储账号等；并配置虚拟机中kubernetes各组件和服务。

$ az group deployment create -g lqik8s07 --template-file azuredeploy.json  --parameters azuredeploy.parameters.json

1. 测试

acs-engine默认会为每个区域生成一个Kubernetes的配置文件。根据您部署的区域， 将该配置赋予您当前的会话，以便进行管理。

$export KUBECONFIG=kubeconfig/kubeconfig.chinanorth.json

$ kubectl get nodes

NAME                        STATUS    ROLES     AGE       VERSION

k8s-agentpool1-30086461-0   Ready     agent     5m        v1.6.11

k8s-agentpool1-30086461-1   Ready     agent     5m        v1.6.11

k8s-master-30086461-0       Ready     master    5m        v1.6.11

排错：

1. IO timeout error:

kubernetes使用apiserver组件用于通信和管理集群服务。若出现该错误，请排查apiserver是否正确启动，以及master节点上端口是否打开。

登录到master节点，看是否有apiserver的docker 容器在运行。

root@k8s-master-30086461-0**:~# docker ps -a | grep apiserver**

fd40fbc2eb90        crproxy.trafficmanager.net:6000/google\_containers/hyperkube-amd64@sha256:f44db072ffd2e0a356268468ea8acddedb35bc45f1a2de38e3f4a4c94ff873dd            "/hyperkube apiserver"   49 minutes ago      Up 49 minutes                           k8s\_kube-apiserve \_kube-apiserver-k8s-master-30086461-0\_kube-system\_dc4225866b646d6de7d22fad11386a75\_0

7ed5d22d047c        crproxy.trafficmanager.net:6000/google\_containers/pause-amd64:3.0                                                                                    "/pause"                 49 minutes ago      Up 49 minutes                           k8s\_POD\_kube-apiserver-k8s-master-30086461-0\_kube-system\_dc4225866b646d6de7d22fad11386a75\_0

查看443端口是否在被监听。

root@k8s-master-30086461-0**:~# netstat -tunlp**

Active Internet connections (only servers)

Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name

tcp        0      0 127.0.0.1:10248         0.0.0.0:\*               LISTEN      1872/hyperkube

tcp        0      0 127.0.0.1:10249         0.0.0.0:\*               LISTEN      2650/hyperkube

tcp        0      0 127.0.0.1:2379          0.0.0.0:\*               LISTEN      1160/etcd

tcp        0      0 10.240.255.5:2379       0.0.0.0:\*               LISTEN      1160/etcd

tcp        0      0 10.240.255.5:2380       0.0.0.0:\*               LISTEN      1160/etcd

tcp        0      0 0.0.0.0:22              0.0.0.0:\*               LISTEN      1223/sshd

**tcp6       0      0 :::443                  :::\*                    LISTEN      2188/hyperkube**

tcp6       0      0 :::4194                 :::\*                    LISTEN      1872/hyperkube

tcp6       0      0 :::31811                :::\*                    LISTEN      2650/hyperkube

tcp6       0      0 :::10250                :::\*                    LISTEN      1872/hyperkube

tcp6       0      0 :::10251                :::\*                    LISTEN      2366/hyperkube

tcp6       0      0 :::10252                :::\*                    LISTEN      2338/hyperkube

tcp6       0      0 :::10255                :::\*                    LISTEN      1872/hyperkube

tcp6       0      0 :::8080                 :::\*                    LISTEN      2188/hyperkube

tcp6       0      0 :::22                   :::\*                    LISTEN      1223/sshd

udp        0      0 0.0.0.0:68              0.0.0.0:\*                           1033/dhclient

              如果没有该容器，docker ps -a 查看运行kuberlet的容器，检查其日志输出是否有相关错误，根据错误进行问题解决。

              root@k8s-master-30086461-0:~# docker ps -a | grep **kubelet**

5135c1bedfa8        crproxy.trafficmanager.net:6000/google\_containers/hyperkube-amd64:v1.6.11                                                                            "/hyperkube kubelet -"   53 minutes ago      Up 53 minutes                           tender\_rosalind

root@k8s-master-30086461-0:~# docker logs 513 > /tmp/kubelet.log 2>&1

如果有容器在运行，但是443端口并没有被监听。查看位于/etc/kubernetes/manifests/下配置文件kube-apiserver.yaml是否有问题。或者查看apiserver容器的日志输出，方法与上一步类似。

测试成功，说明Kubernetes集群搭建完成。您可以开始在集群中部署您应用了。由于Kubernetes技术是开源产品，关于其具体使用中的问题，若与Azure平台无关，建议咨询开源社区或在Github中寻求支持。