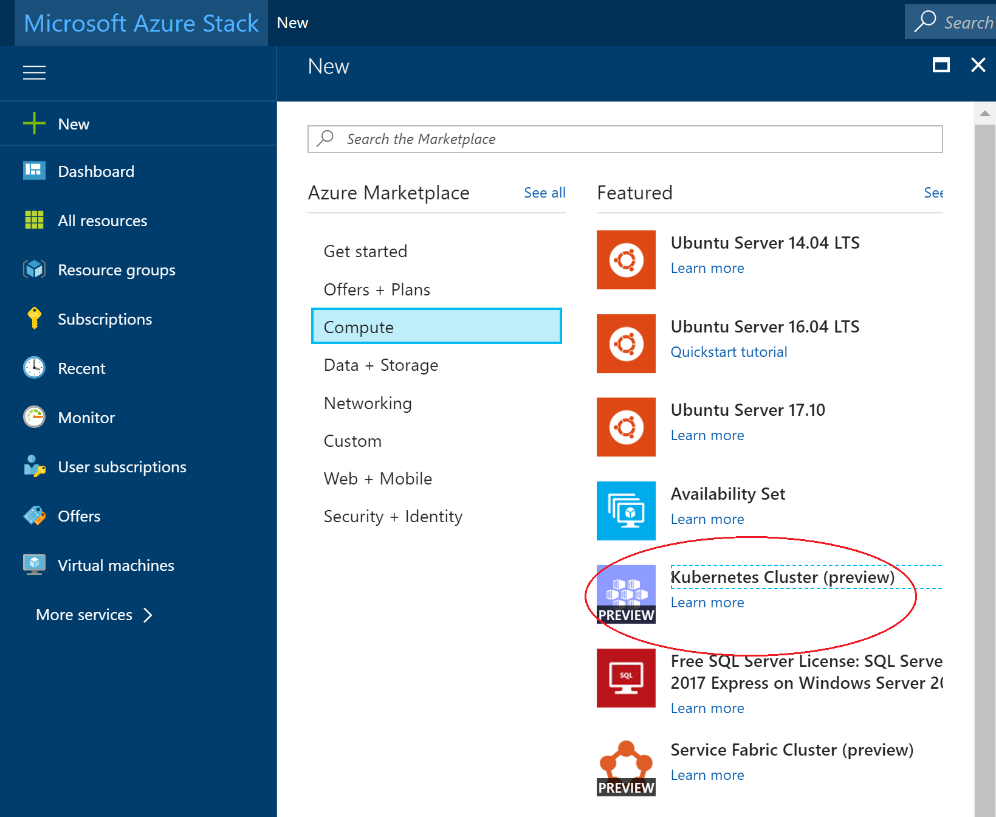
Revision History

说明：

文档中关于在Azure Stack上安装K8S cluster的步骤在Azure Stack连接global Azure的情形下验证

Azure Stack支持Kubernetes的现状

Kubernetes Cluster是Azure Stack plan的标准feature，在Marketplace -> Compute下可以找到Kubernetes Cluster服务。



但是目前还有几个问题：

Kubernets Cluster在Azure Stack中还是private preview，在安装过程中可能经常会遇到问题。因为我们能接触到的配置项相对比较少，往往很难解决，目前只能找产品组协助解决

Kubernetes Cluster的安装，默认会去从global站点拉取Kubernetes服务所需的各种镜像。因为国内防火墙的限制，这往往导致镜像拉取不成功，导致Kubernetes服务进程不能启动，整个部署失败。从很多国外技术论坛的内容看，讨论类似国内防火墙导致的问题并不多，因此global产品组可能还没有重视这个问题。直接使用Kubernetes Cluster进行K8S集群部署，很可能会遇到防火墙问题，而我们能用的配置项较少，很难对Kubernetes cluster服务的镜像源地址进行配置

因此，目前阶段，如果我们想在Azure Stack上部署Kubernetes集群，使用acs-engine应该是比较可行的办法。当然，和使用acs-engine在Azure公有云上部署K8S集群相比，在Azure Stack上部署K8S集群步骤相对多一点，主要是因为每一个Azure Stack的安装instance，它们的内部profile信息不相同。在创建部署K8S集群部署的apiModel文件时，必须引入Azure stack的profile内容，而对于Azure公有云来说这部分是不用反应在apiModel中的。

虽然在Azure Stack上部署K8S相对要麻烦一点，但并不算困难。下面我们就开始部署K8S on Azure Stack！

前期准备

确认Azure stack需要在1803版本以上（否则可能的问题请参考前人的总结：<https://mp.weixin.qq.com/s?__biz=MzA3NTM1MzE4Nw==&mid=2649617746&idx=1&sn=8d110646120244bbab6b55e252fc744e&chksm=8768885eb01f014824a130cb4148cd9c8648e20d155549a1d09df7d88b3dceeec64821ee09dd#rd>）

Azure Stack与公有云Azure连接

确保Azure Marketplace中包含：

* Publisher = "Canonical" Offer = "UbuntuServer" SKU = "16.04-LTS" Version = "16.04.201802220" OSType = "Linux"
* Custom Script for Linux, 2.0.3

获取Azure Stack ERCS访问信息（IP、用户名/密码）

git clone https://github.com/kylercai/K8SOnAzureStack.git

创建SSH密钥对，用于SSH访问K8S集群的节点

运行acs-engine的Linux VM

创建apiModel

从命令”git clone <https://github.com/kylercai/K8SOnAzureStack.git>”，可以获取两个关键文件：

azurestack-default.json

AzureStack.AcsEngine.psm1，我们将要运行这个powershell脚本

说明：

AzureStack.AcsEngine.psm1脚本的原始版本可以从<https://github.com/radhikagupta5/AzureStack-QuickStart-Templates/tree/radhikgu-acs/101-acsengine-kubernetes-1803> 找到。经过多次试验后，对原始的版本进行了简化，去掉了创建Service Principal和授权，创建存储账户和上传apiModel等步骤。这些步骤并不是必须的，且比较容易导致访问权限、api版本不兼容等问题，导致大量的troubleshooting要做。去掉这些步骤后，在后续步骤中采用少量的手工操作进行替代，避免原始版本可能导致的困难的troubleshooting。

启动Powershell ISE，建议以管理员身份启动

依次执行以下命令，并用合适的内容替代highlight部分

Import-Module *d:\MSFT\Azure-Study\AKS\azurestack\mtc\AzureStack.AcsEngine.psm1* -Force -Verbose **// AzureStack.AcsEngine.psm1在本地的完整路径**

$namingSuffix = *23456* **// 希望在K8S master节点的DNS前缀**

$masterDnsPrefix = "k8s-" + $namingSuffix

$tenantSubscriptionId = "*f3bd3-……-bc867b*" **// 当前使用Azure Stack的subscription**

$CloudAdminPass = ConvertTo-SecureString "*xxxxxx*" -AsPlainText -Force **// 访问ERCS的密码**

$cloudAdminCredential = New-Object System.Management.Automation.PSCredential ("*xxxxxxx*", $CloudAdminPass) **// 访问ERCS系统的用户名**

$serviceAdmin = " *xxx@xxxx.xxxx.xxx* " **// 当前访问AzureStack的用户名**

$AdminPass = ConvertTo-SecureString "*xxxxxxx*" -AsPlainText -Force **// 当前访问AzureStack的密码**

$serviceAdminCredential = New-Object System.Management.Automation.PSCredential ($serviceAdmin, $AdminPass)

$TenantAdmin = "*xxx@xxxx.xxxx.xxx*" **// 当前访问AzureStack的用户名**

$TenantAdminPass = ConvertTo-SecureString "*xxxxxxx*" -AsPlainText -Force **// 当前访问AzureStack的密码**

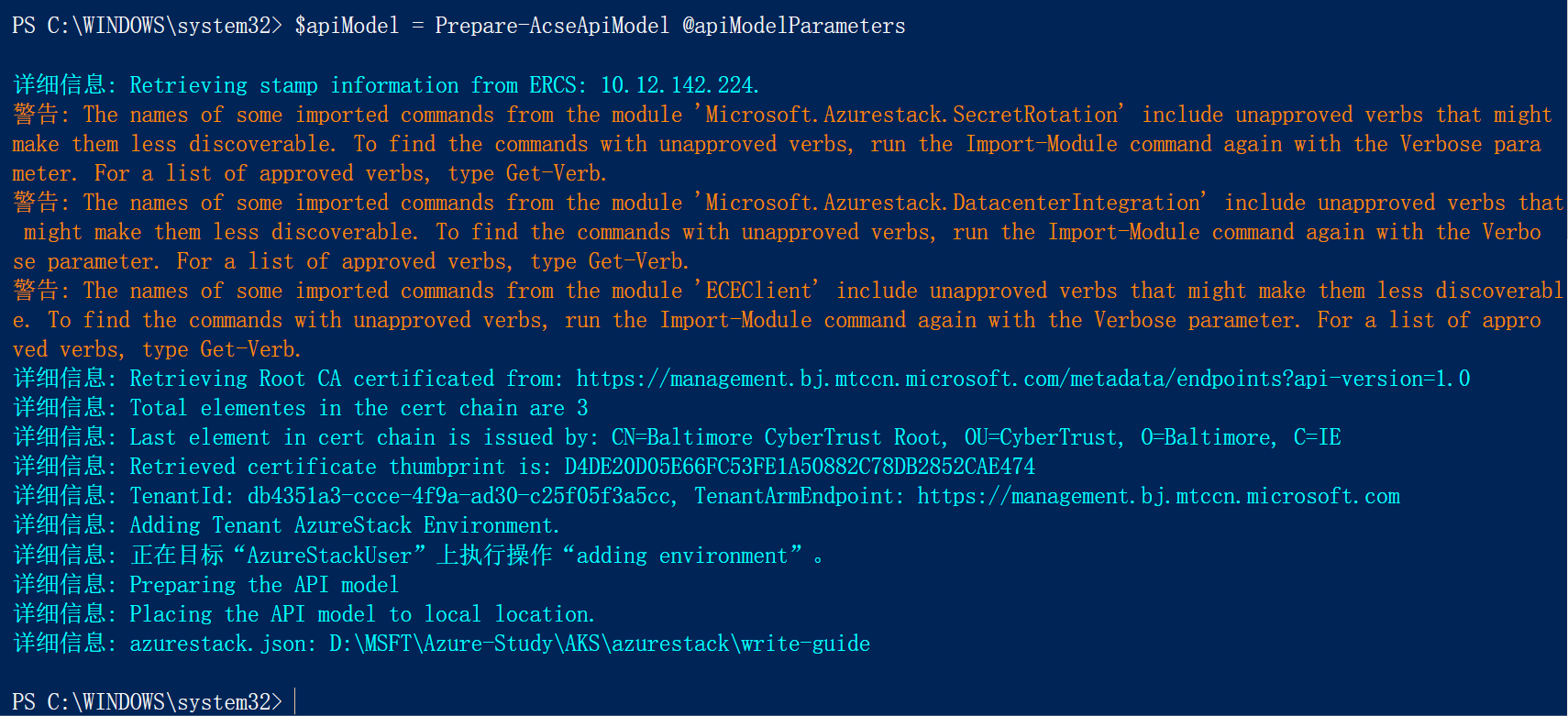
$tenantAdminCredential = New-Object System.Management.Automation.PSCredential ($TenantAdmin, $TenantAdminPass)

$acsSshKey = "*ssh-rsa AAAAB3….voAoeJQw/af azureuser@azconsole*" **// 创建的SSH密钥对的公钥内容**

$apiModelParameters = @{'ErcsComputerName' = "*10.x.x.x*"; 'CloudAdminCredential' = $cloudAdminCredential; 'ServiceAdminCredential' = $serviceAdminCredential; 'TenantAdminCredential' = $tenantAdminCredential; 'TenantSubscriptionId' = $tenantSubscriptionId; 'MasterDnsPrefix' = $masterDnsPrefix; 'LinuxVmSshKey' = $acsSshKey; 'NamingSuffix' = $namingSuffix;} **// ERCS的IP地址**

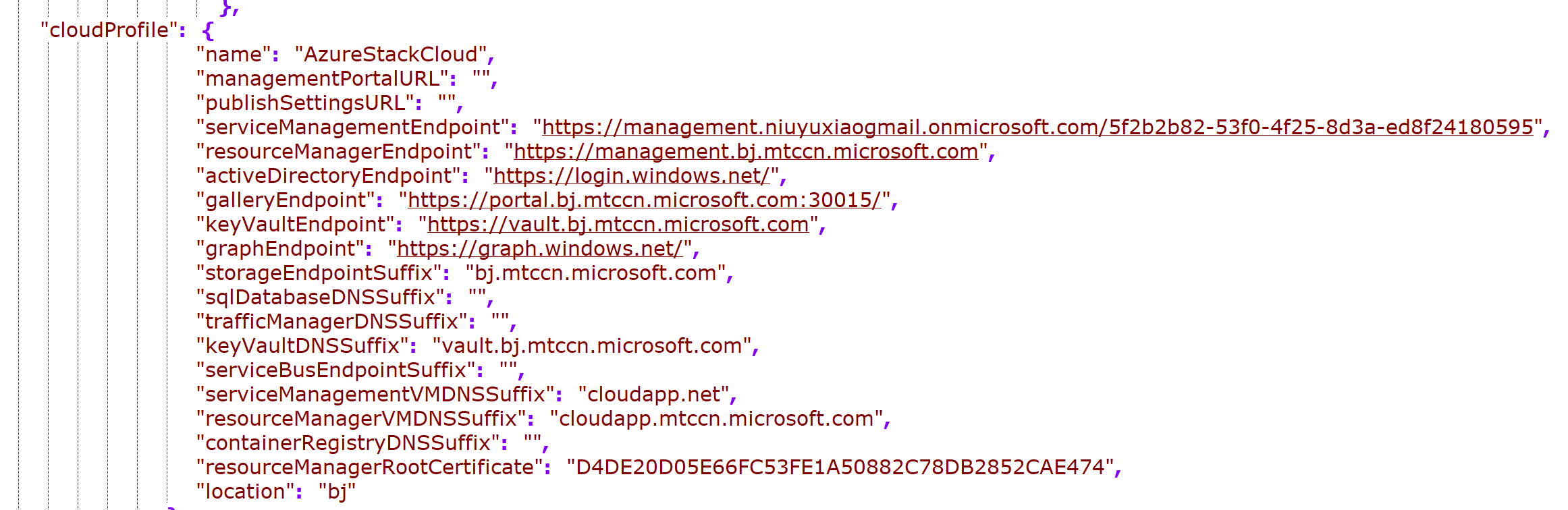
$apiModel = Prepare-AcseApiModel @apiModelParameters

如果顺利，执行的结果如下，在红色方框内，显示生成的apiModel文件（azurestack.json）在本地的目录路径：



说明：

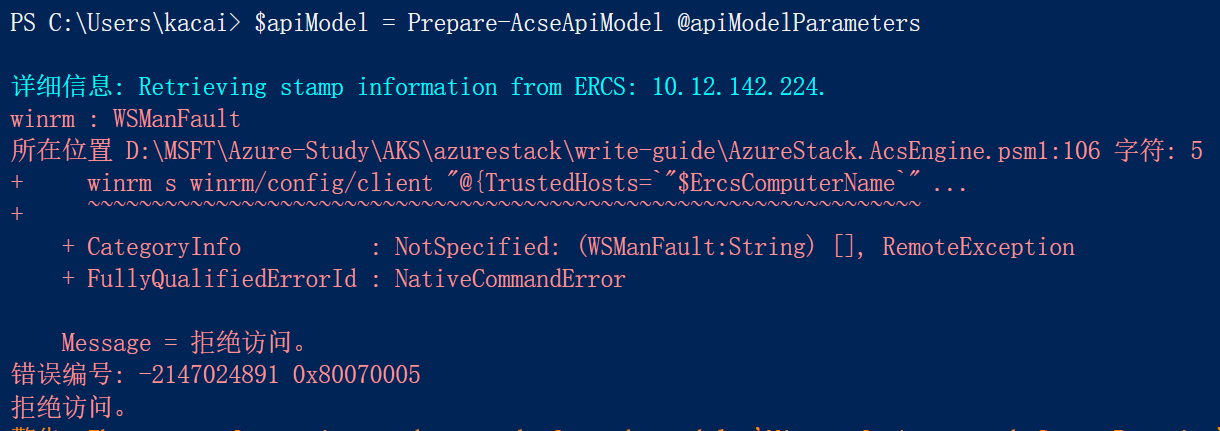
执行过程连接Azure Stack的ERCS，从中抽取Azure Stack Instance的thumbprint、tenantId、endpoint等信息。这些信息被填到生成的apiModel文件中，相应的在cloudProfile部分，大致内容类似：



这些信息在公有云使用acs-engine时是不必须的。

Tip：

建议使用管理员身份启动Powershell ISE来运行以上命令，否则可能会出现本地创建TrustedHosts被拒绝



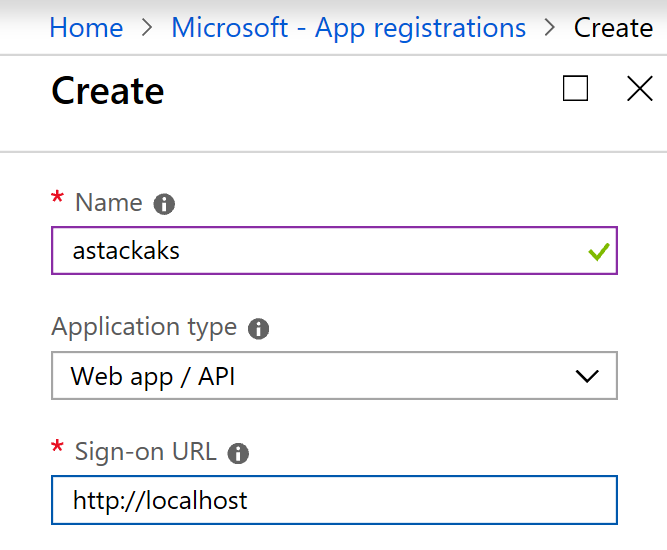
在脚本执行过程中，可能会遇到一些问题，但我们的目标只是生成apiModel文件，因此只要脚本还能继续运行下去，最终生成apiModel文件，我们不一定都需要处理所有的问题。<https://github.com/kylercai/K8SOnAzureStack.git> 这个地址下有一个创建成功生成的apiModel文件（azurestack.json）例子，只要您生成apiModel文件和例子对比内容都完整，就不需要关注脚本运行过程中，遇到的无关我们目标的问题。

创建SPN

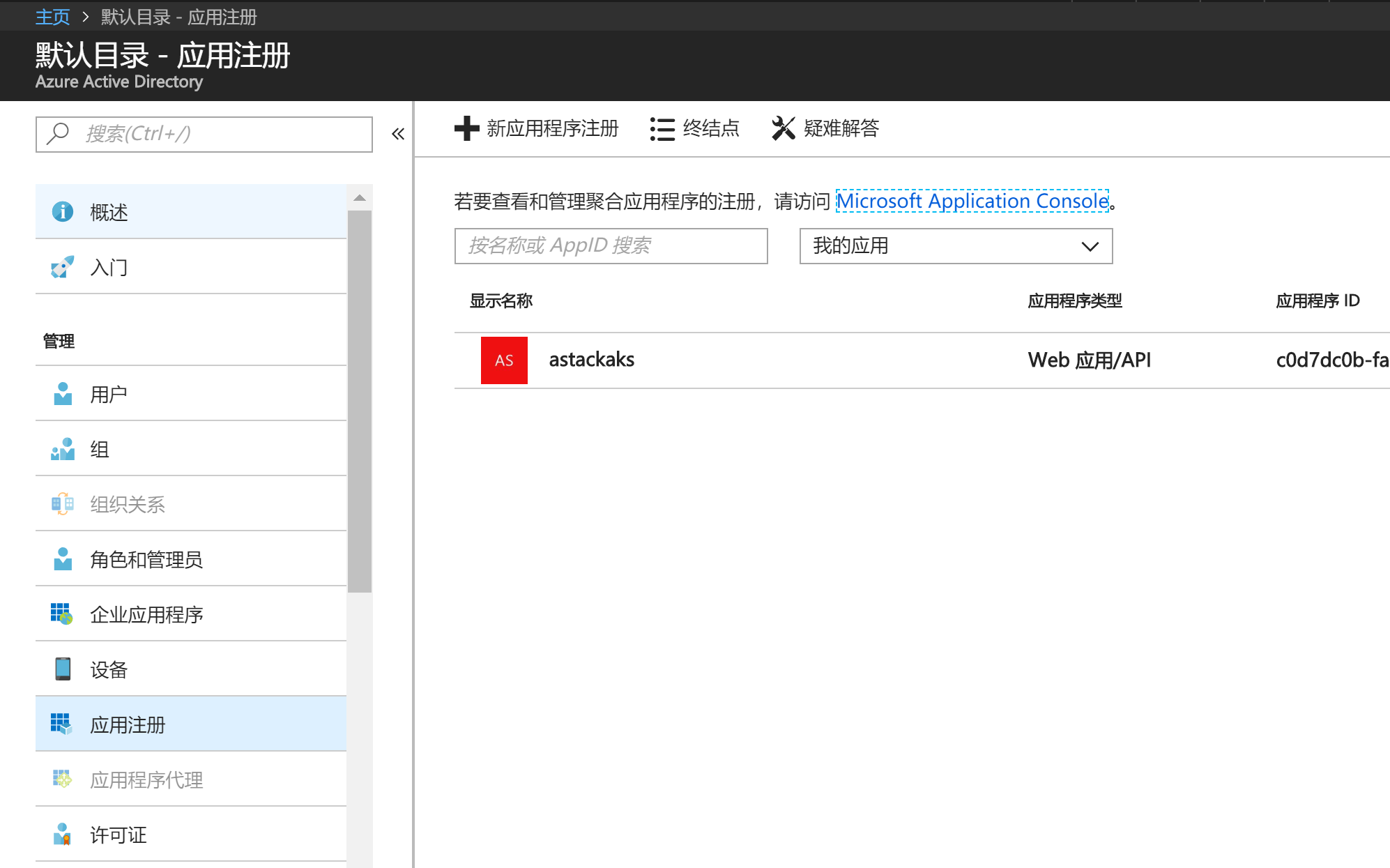
用Azure Stack的登录租户ID，登录global Azure

进入Azure Active Directory -> App registrations -> New application registration

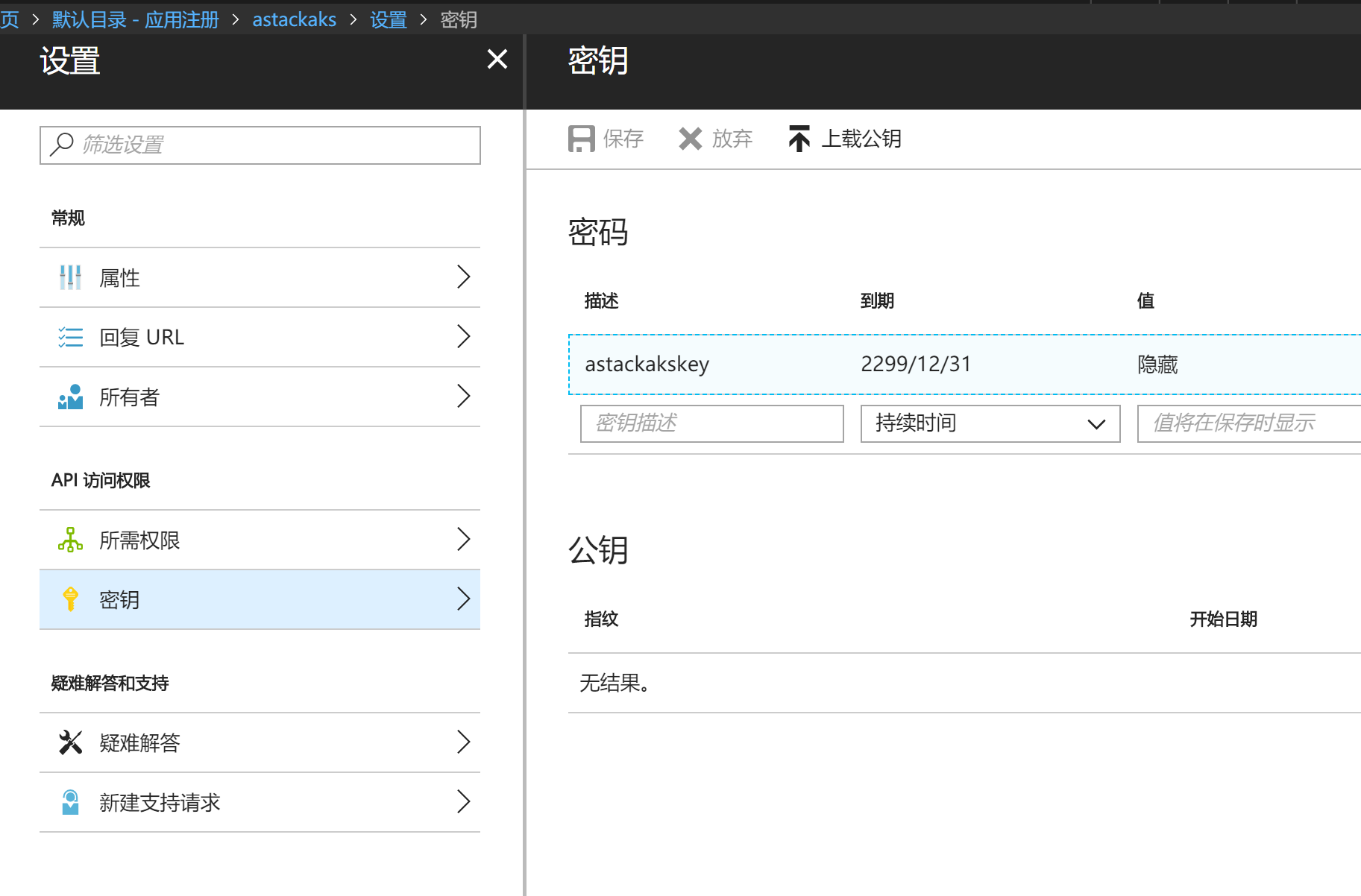
输入应用名称、Application type选择Web app/API、Sign-on URL选择<http://localhost>



记录创建的Application ID，这个作为后续要使用的Service Principal client ID。



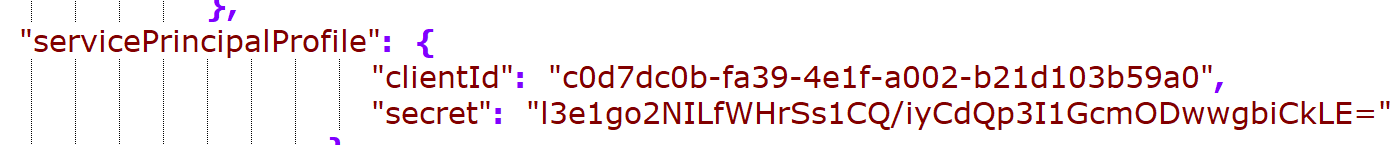
点击创建的应用，进入 Settings -> Keys，填写Description、Expire选择Never expires，点击保存，记录创建的Key值。这个作为后续要使用的Service Principal client secret。



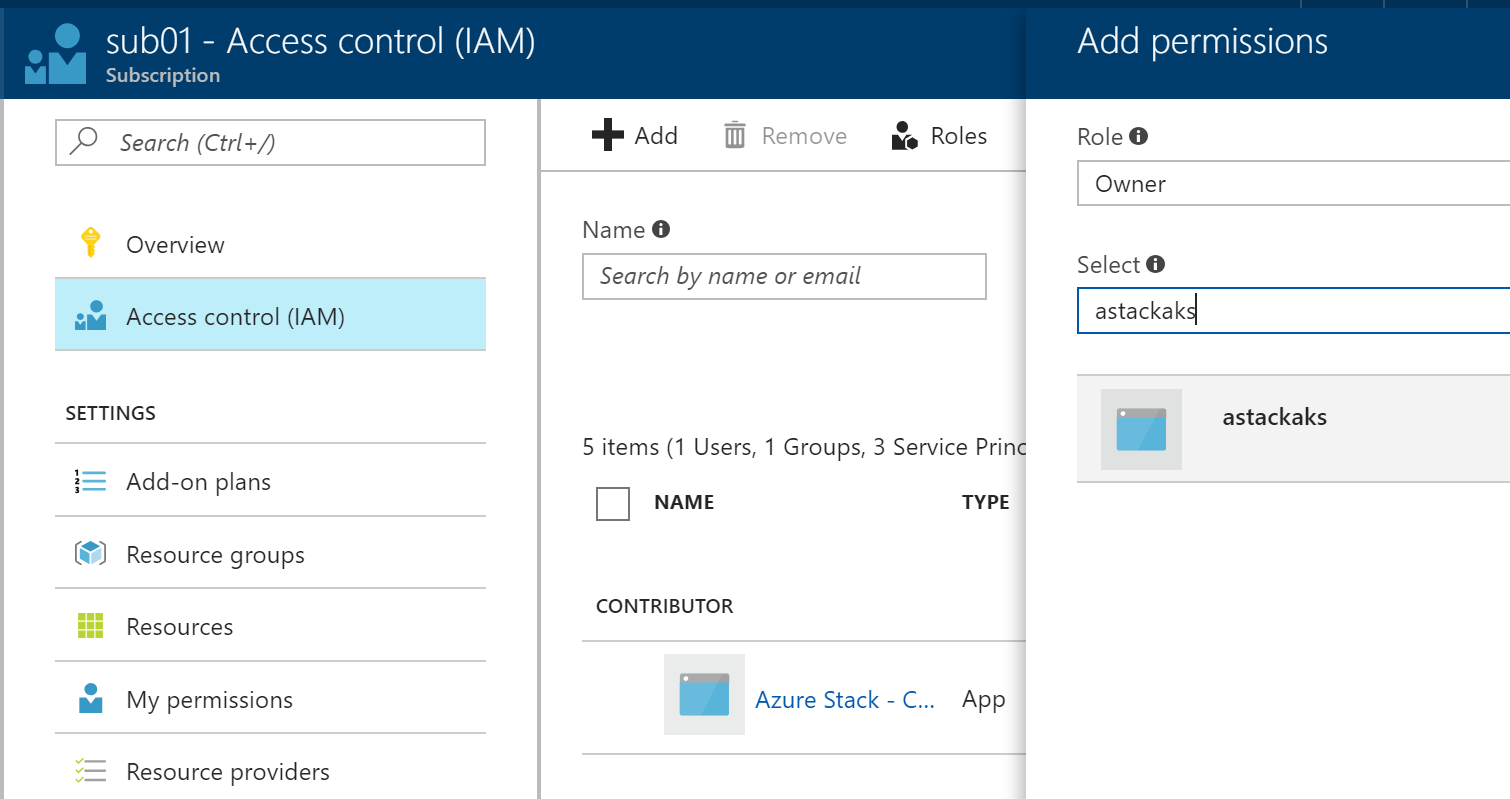
打开之前创建的apiModel（azurestack.json）文件，查找到以下部分：



把上面或的client id和key分别填入”clientId”和”secret”



给Service Principal授权访问当前Azure Stack的subscription。登入Azure Stack portal，进入Subscription，选择Access control(IAM) -> Add，在Role处选择Owner，在Select下输入应用名进行查找，选择应用并保存



说明：

这一步要求Azure Stack必须是和公有云Azure连接的

创建K8S部署ARM template

登入实现准备好的Linux VM，从 <https://github.com/msazurestackworkloads/acs-engine/archive/acs-engine-v0140.zip> 下载acs-engine到本地

解压acs-engine-v-140.zip，在examples/azurestack目录下，可以看到acs-engine可执行文件，这个是支持AzureStack的acs-engine版本

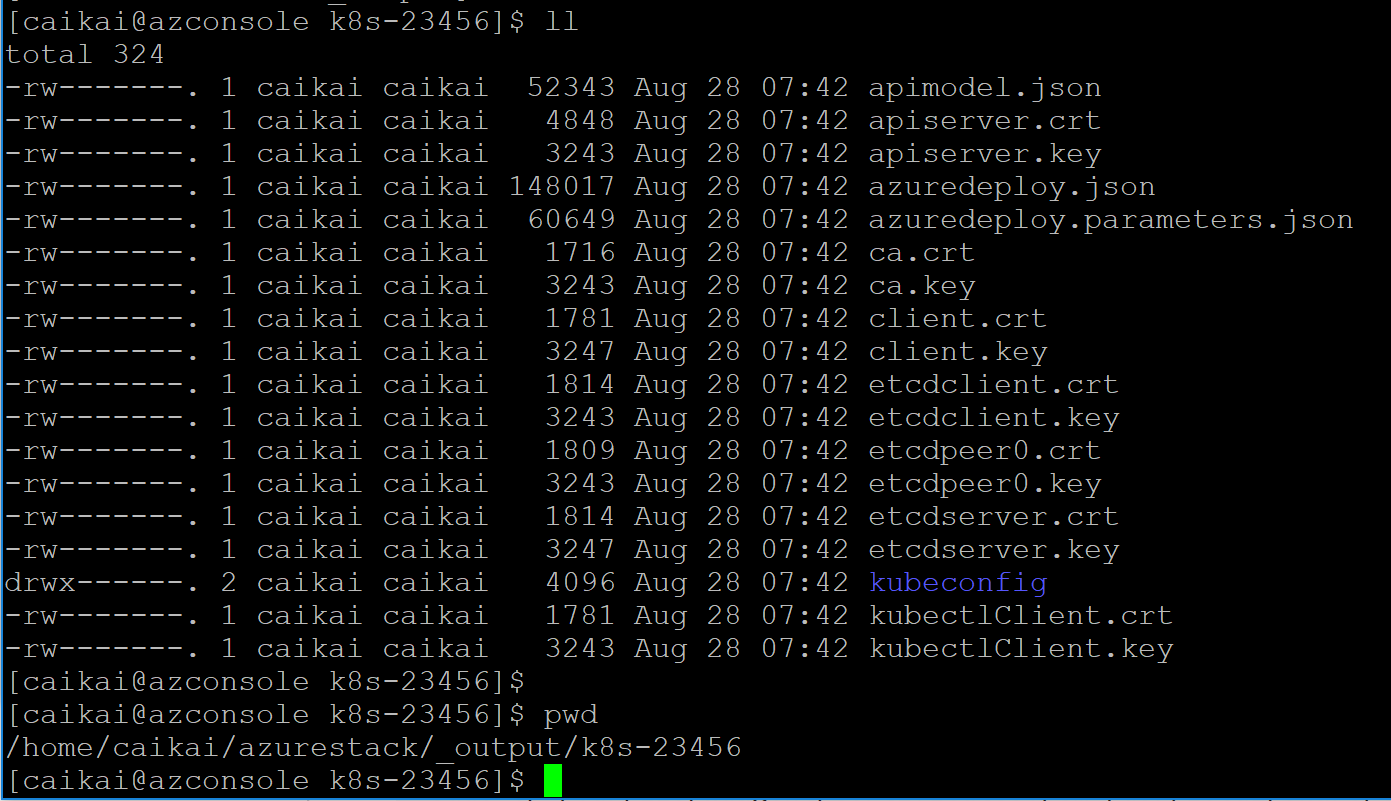
上传之前生成的apiModel文件到Linux VM

运行命令 “./acs-engine generate azurestack.json”，根据您当前所在的目录适当调整命令中的路径

在当前目录下，生成\_output目录。在目录下可以找到我们生成的：

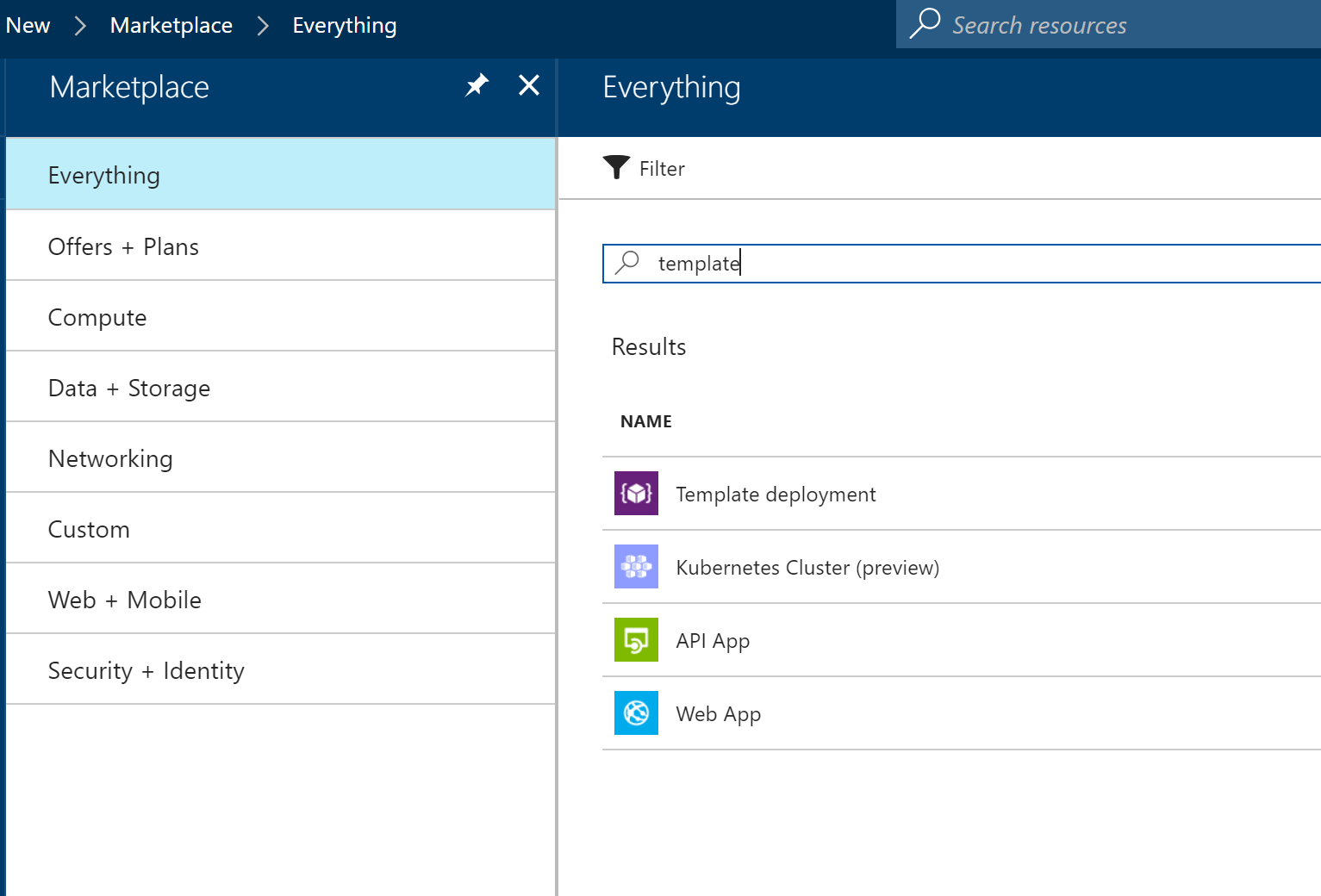
ARM template：azuredeploy.json

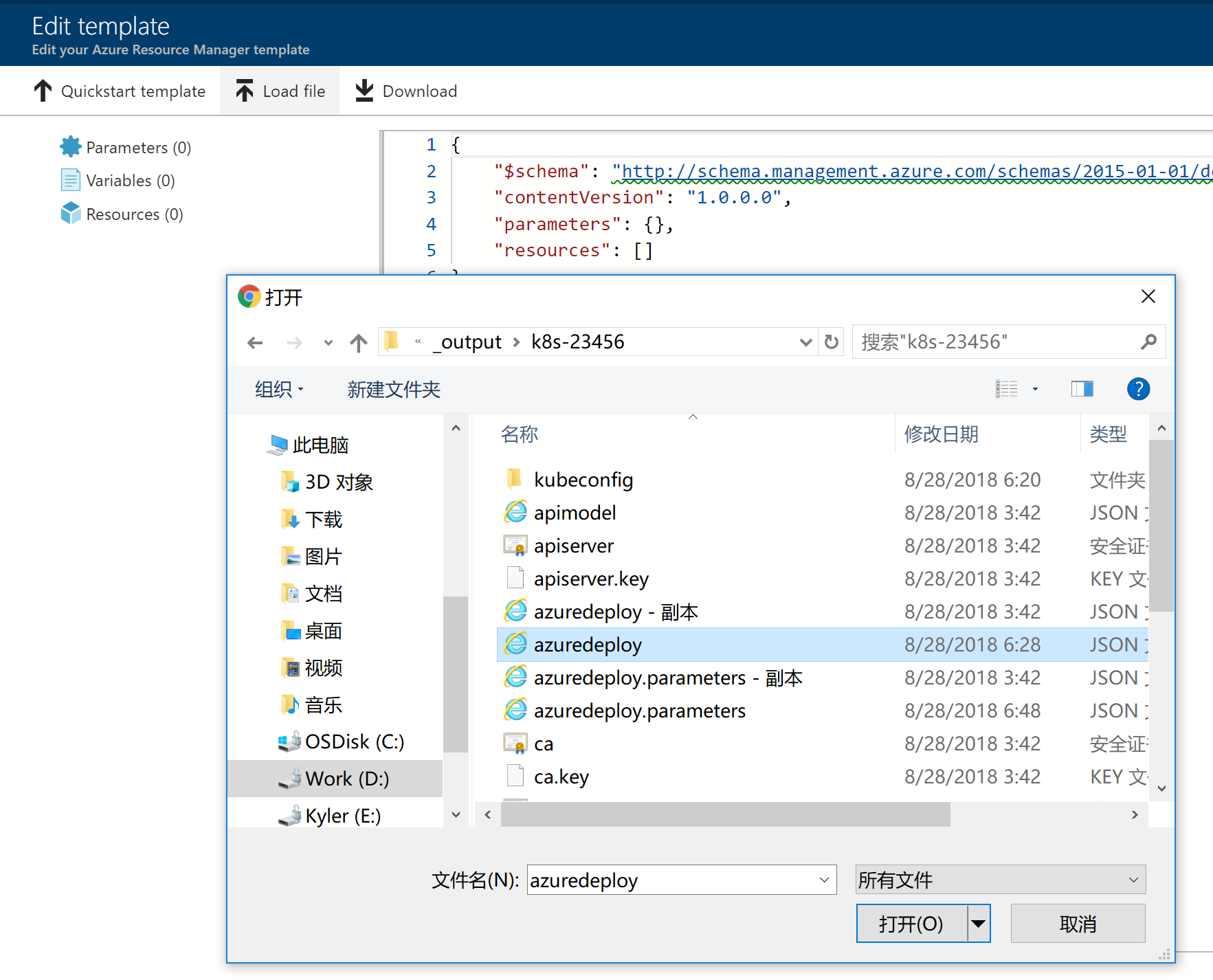
Parameter文件：azuredeploy.parameters.json

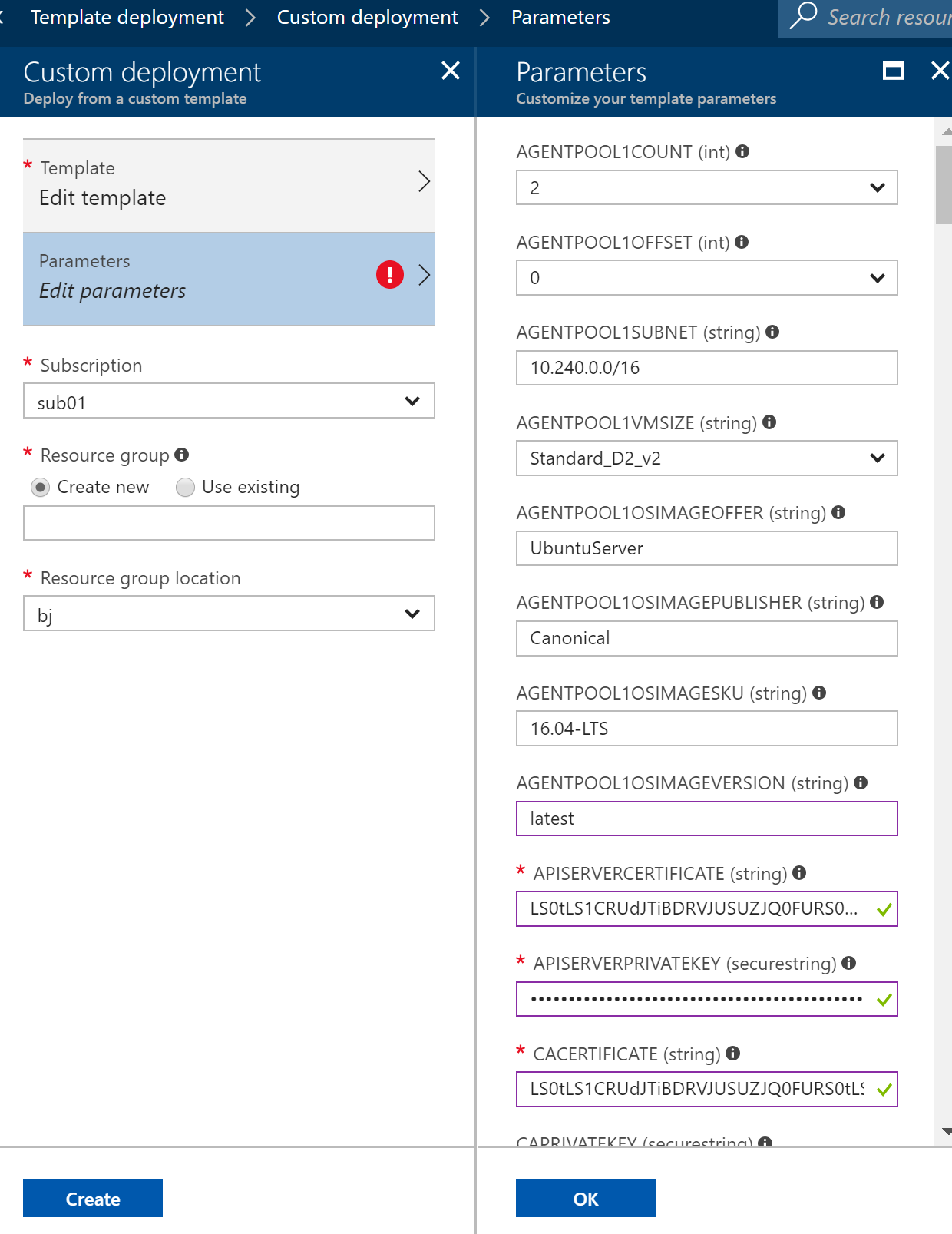


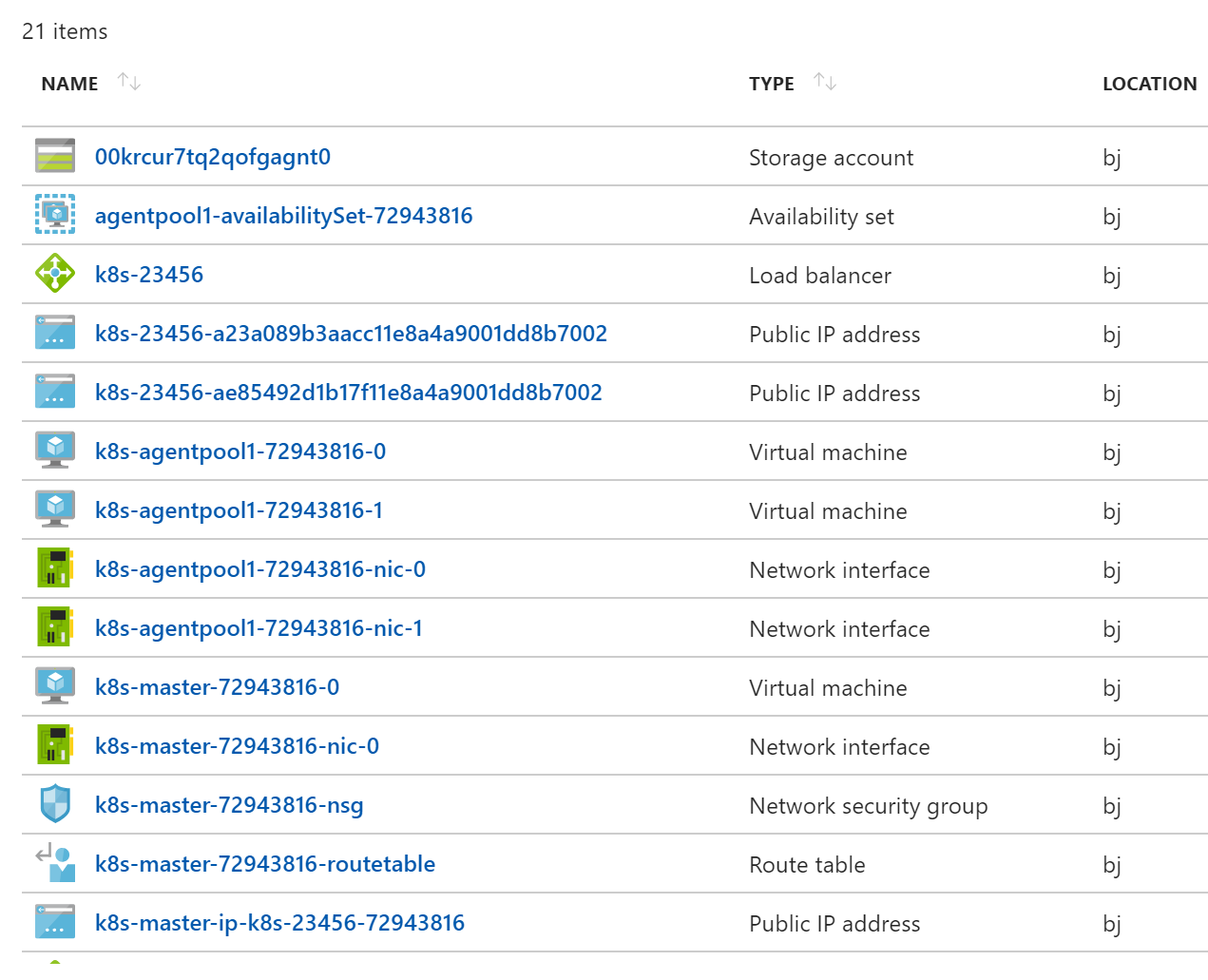
替换gcr

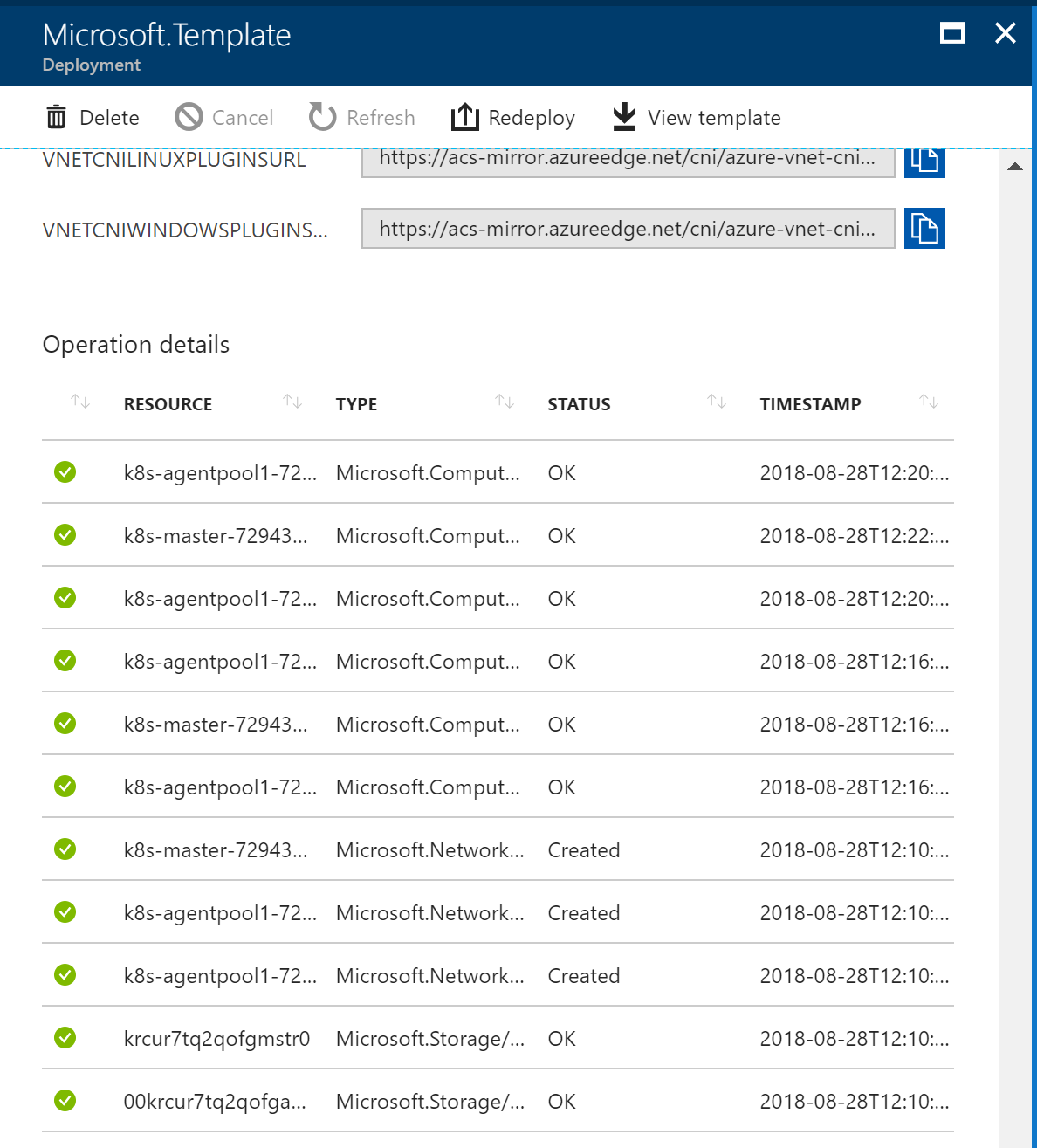
部署template

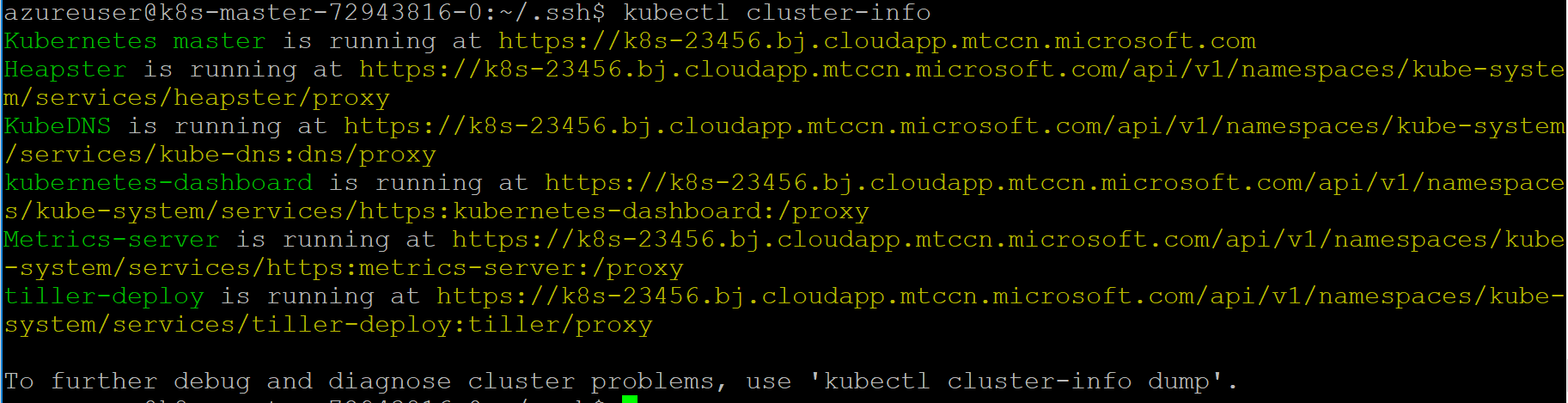


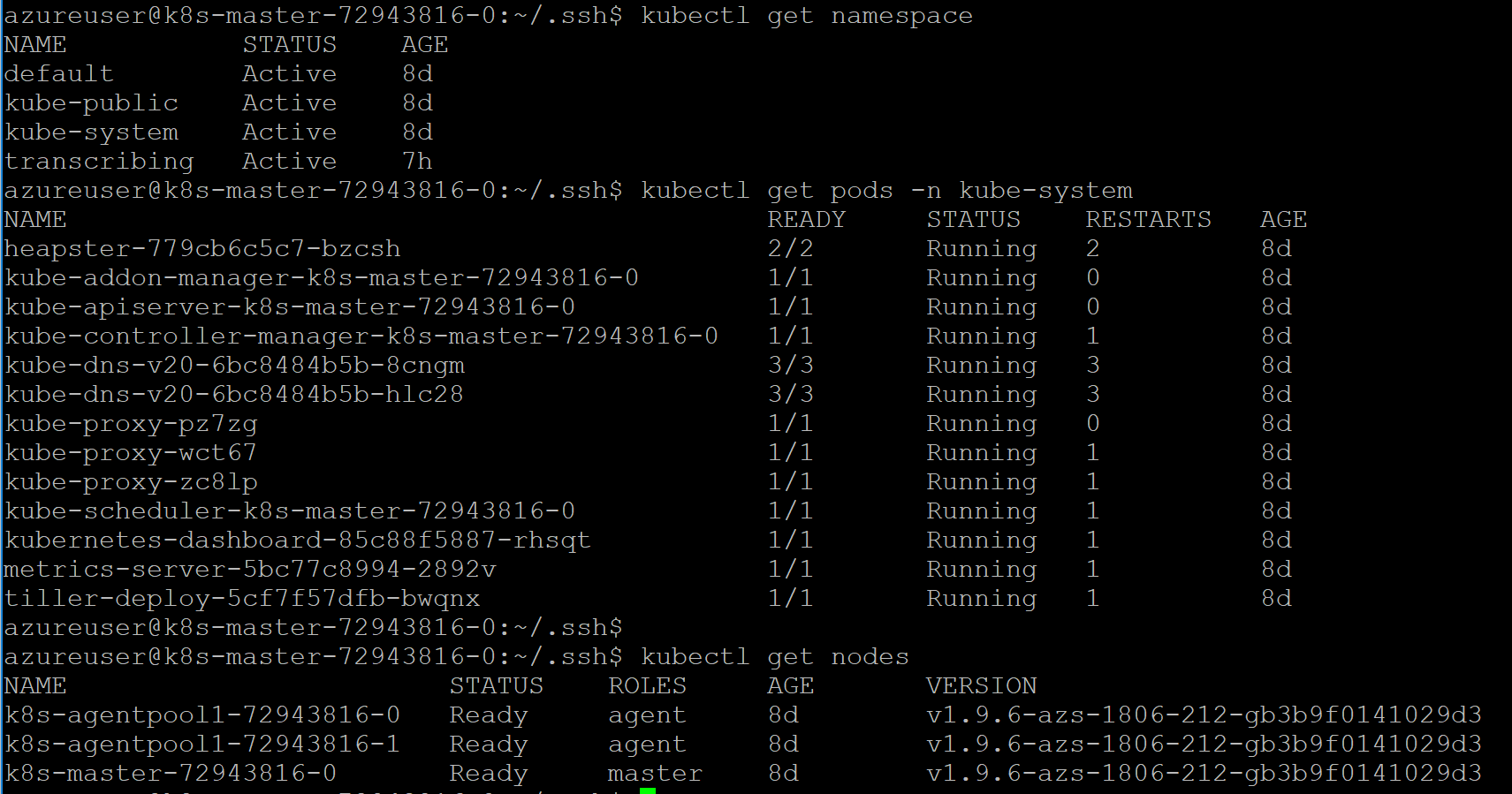


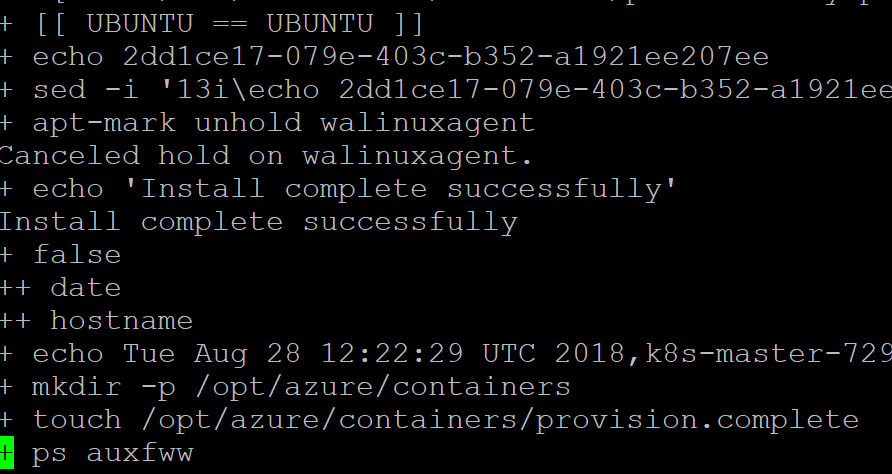






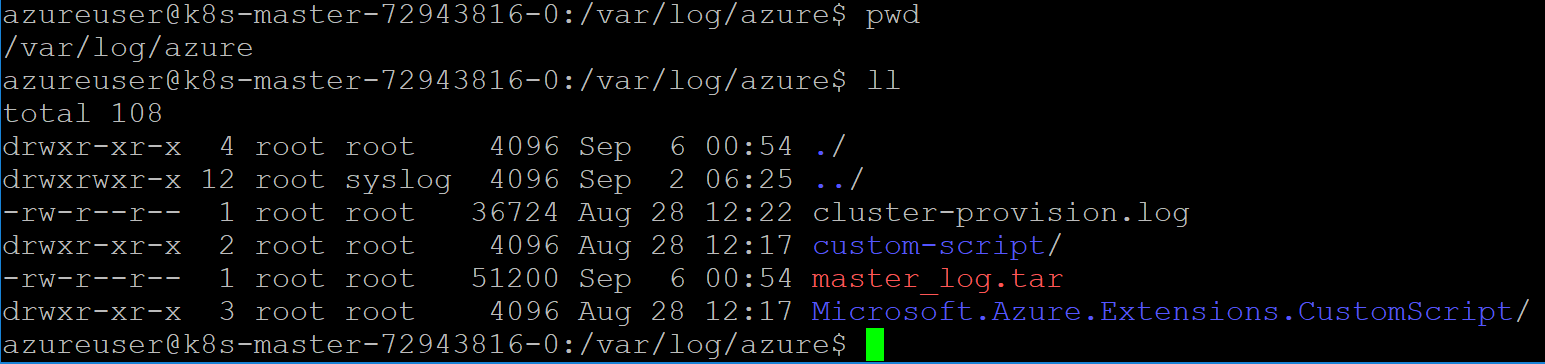




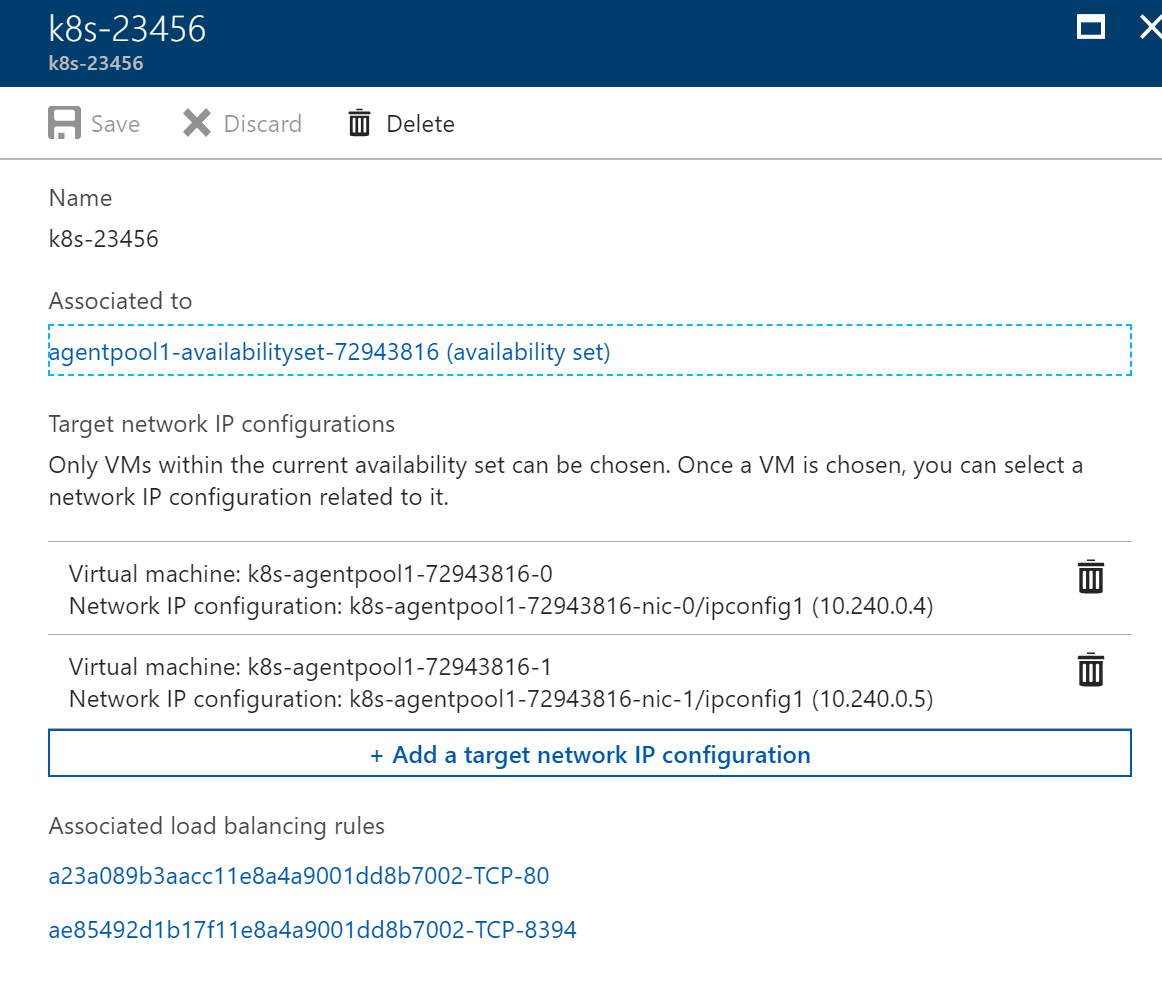


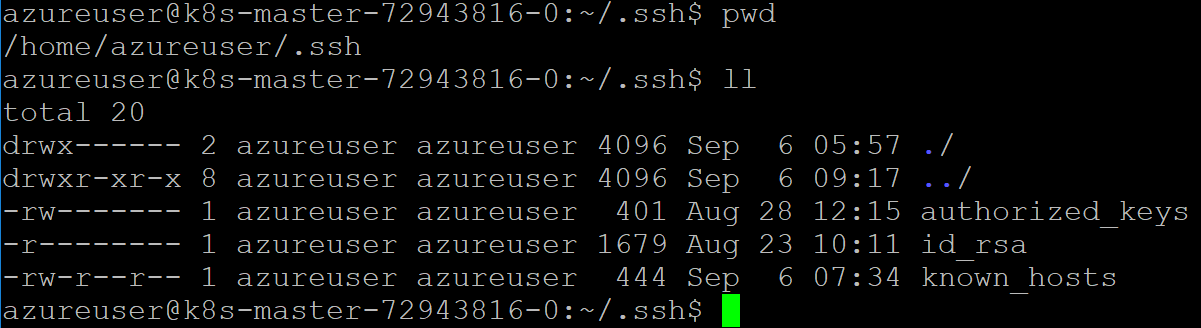
错误排查



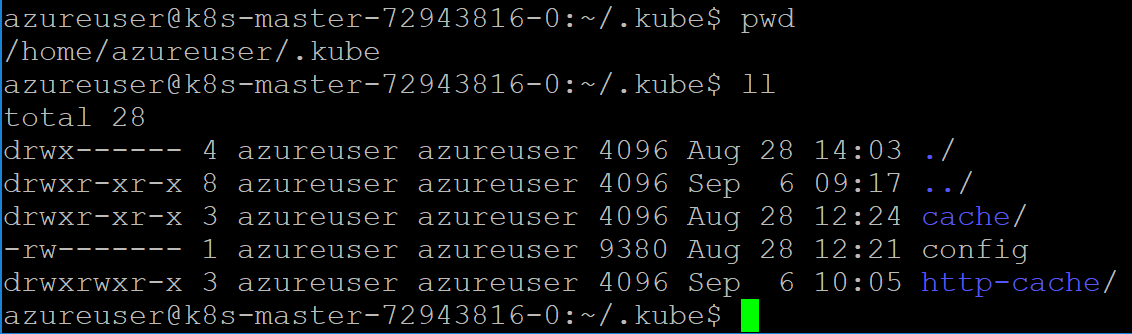


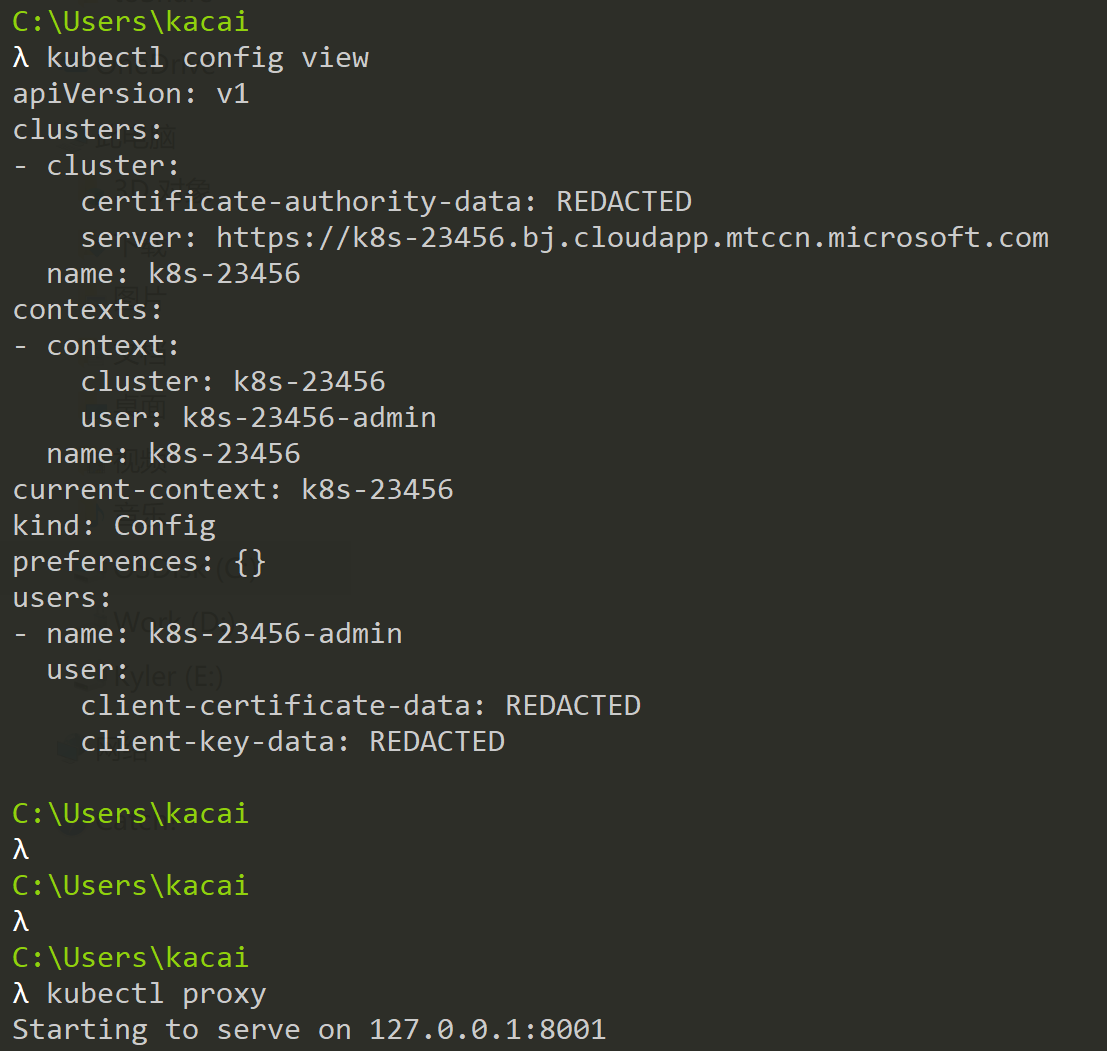
登录slave节点



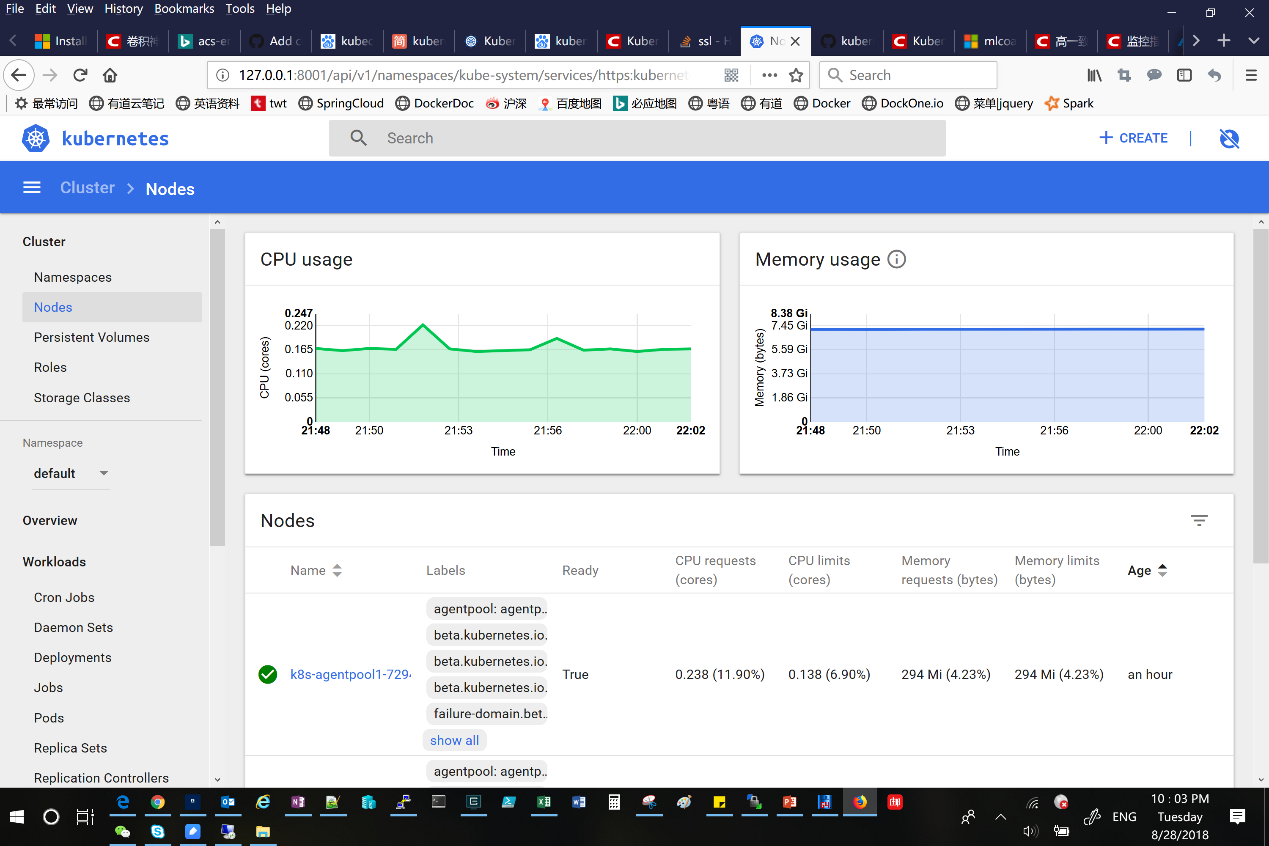


配置本地连接

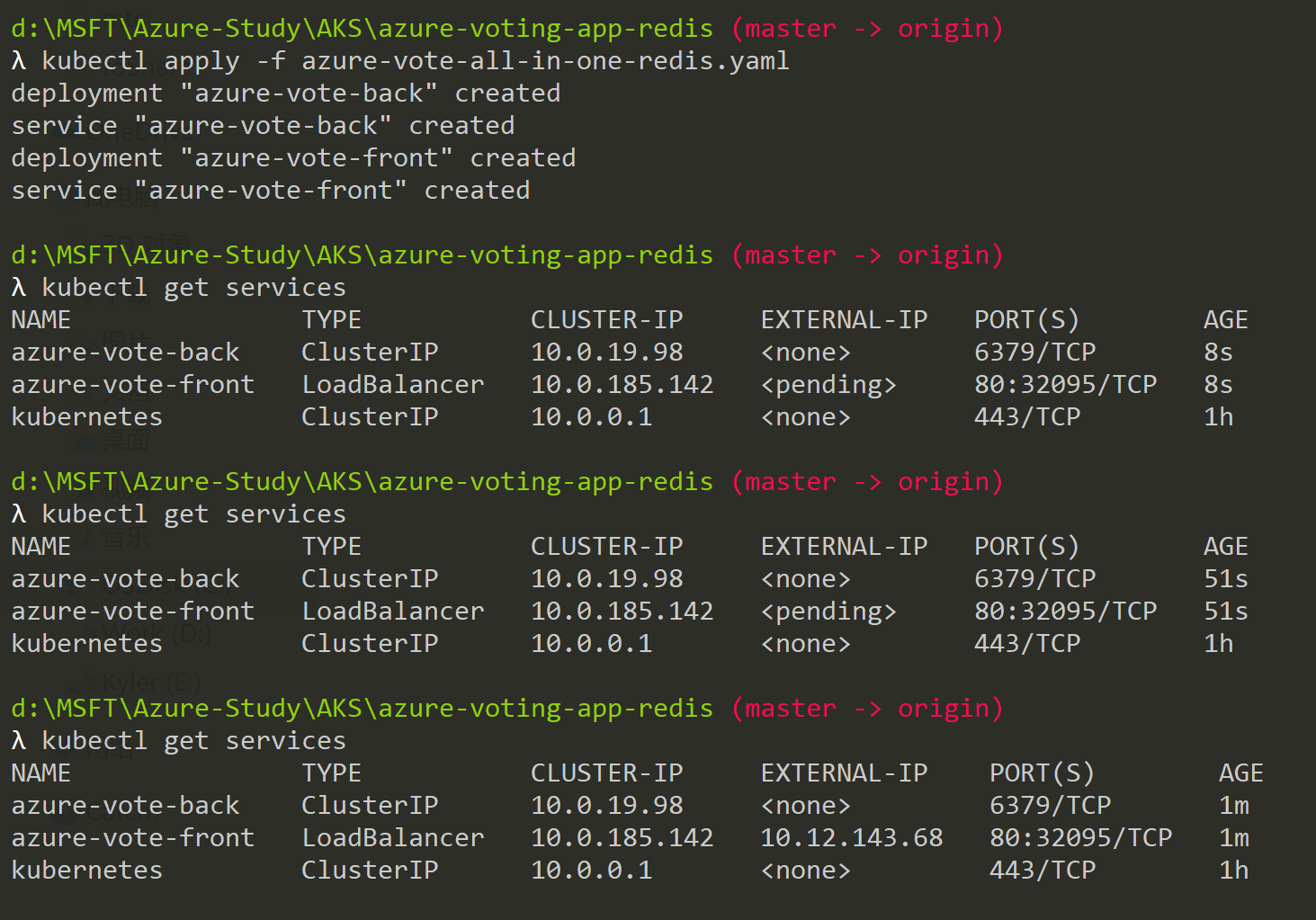


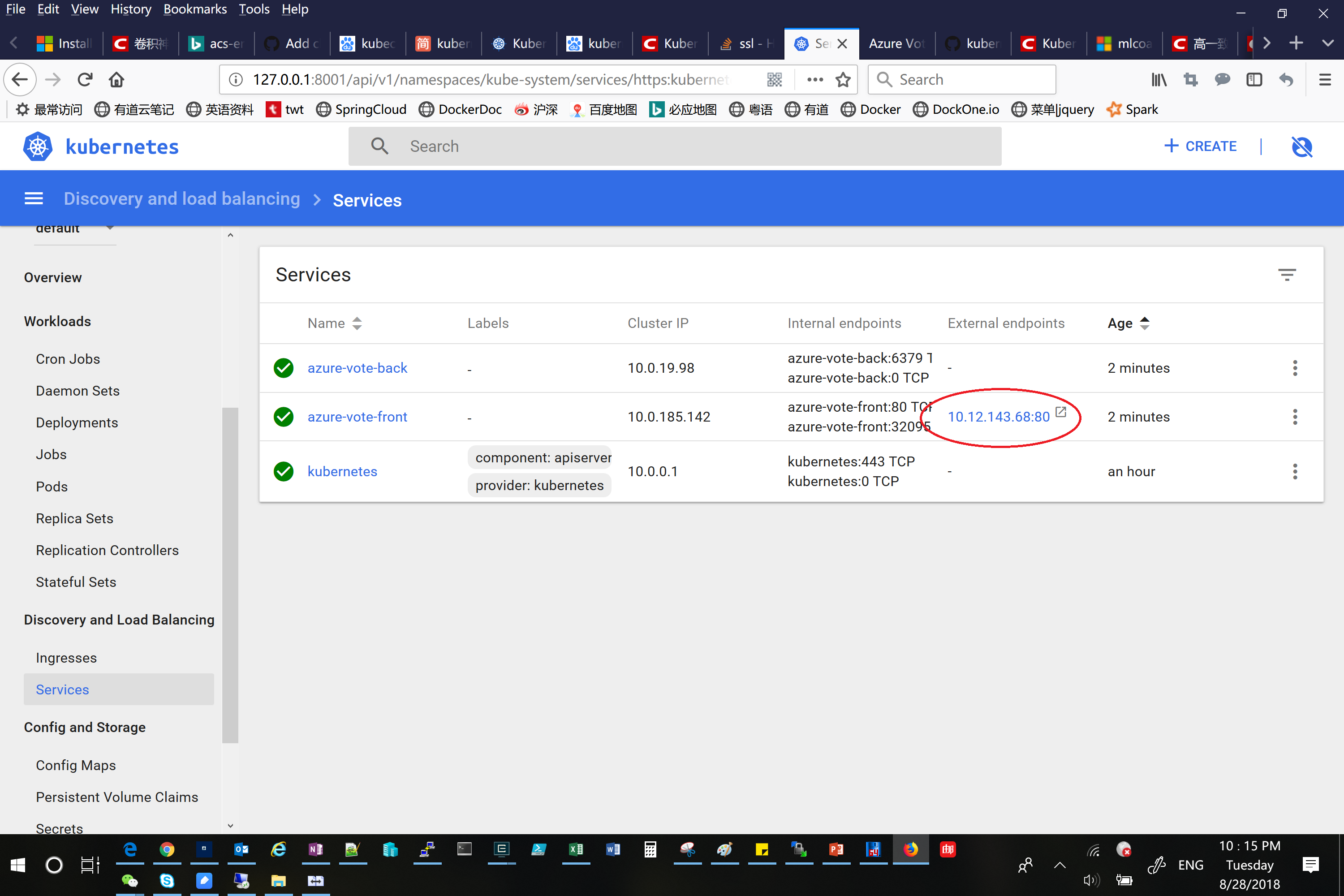


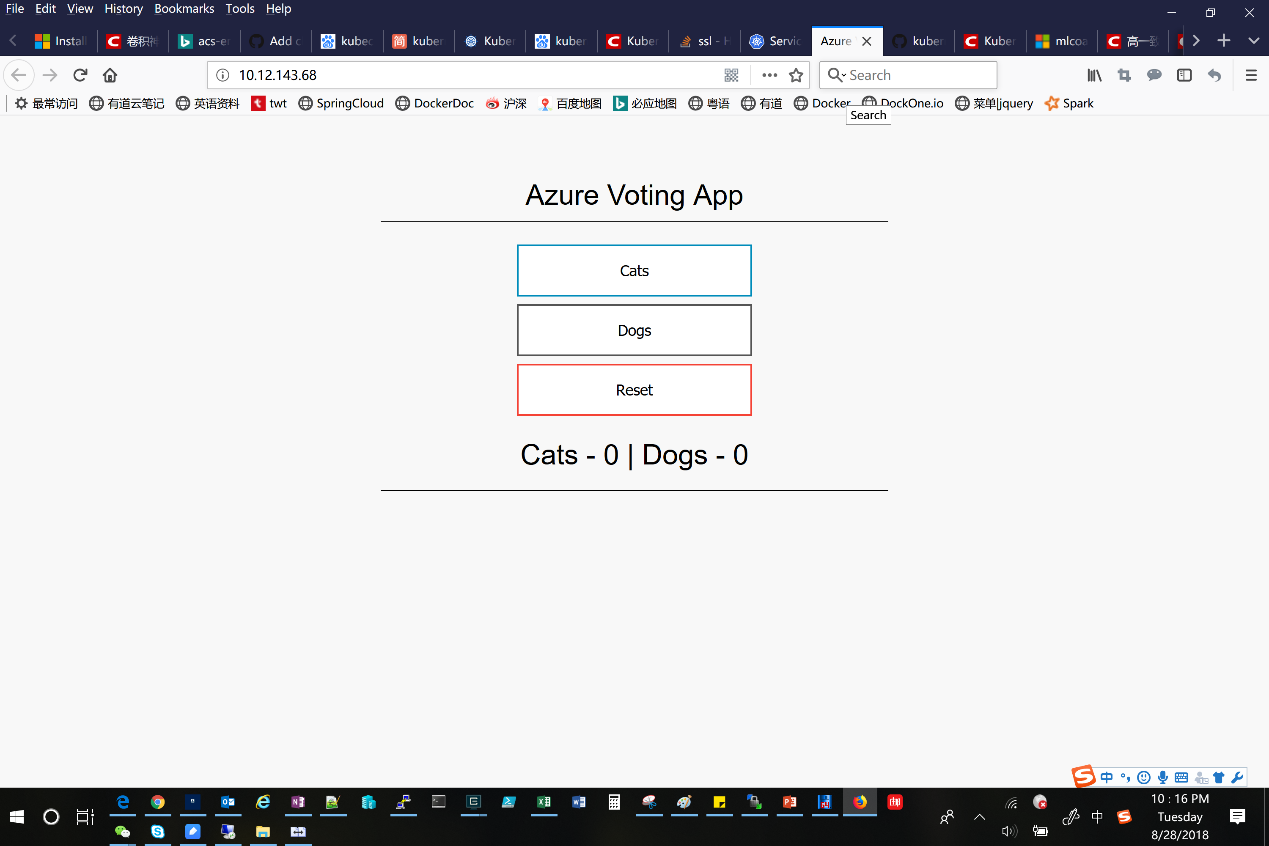
<http://localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/>



部署应用







参考信息

ACS on AzureStack

<https://docs.microsoft.com/en-us/azure/azure-stack/user/azure-stack-solution-template-kubernetes-deploy>

Add a Kubernetes Cluster to the Azure Stack Marketplace

<https://docs.microsoft.com/en-us/azure/azure-stack/azure-stack-solution-template-kubernetes-cluster-add>

<https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-create-service-principal-portal#check-azure-active-directory-permissions>

微信公众号文章：

在Azure Stack上部署Kubernetes

<https://mp.weixin.qq.com/s?__biz=MzA3NTM1MzE4Nw==&mid=2649617738&idx=1&sn=72e614800b05450390348b290a5b13e7&chksm=87688846b01f015006aeeaa32850fd7d2c8660599e026738b0845e4c908bb0ec98b135408ae9&mpshare=1&scene=1&srcid=0801r6UviPLPMgo9oEcOpUMD#rd>

有关中国Azure Stack部署K8S的疑难解答

<https://mp.weixin.qq.com/s?__biz=MzA3NTM1MzE4Nw==&mid=2649617746&idx=1&sn=8d110646120244bbab6b55e252fc744e&chksm=8768885eb01f014824a130cb4148cd9c8648e20d155549a1d09df7d88b3dceeec64821ee09dd#rd>

AzureStack K8S templates：

<https://github.com/radhikagupta5/AzureStack-QuickStart-Templates/tree/radhikgu-acs/101-acsengine-kubernetes-1803>

<https://github.com/radhikagupta5/AzureStack-QuickStart-Templates/tree/radhikgu-acs/101-acsengine-kubernetes-1804>

<https://github.com/ahpeng/K8SOnAzureStack>

步骤参考:

<https://github.com/Azure/acs-engine/tree/master/docs>

\* <https://github.com/radhikagupta5/AzureStack-QuickStart-Templates/tree/radhikgu-acs/101-acsengine-kubernetes-1804>

\* <https://mp.weixin.qq.com/s?__biz=MzA3NTM1MzE4Nw==&mid=2649617738&idx=1&sn=72e614800b05450390348b290a5b13e7&chksm=87688846b01f015006aeeaa32850fd7d2c8660599e026738b0845e4c908bb0ec98b135408ae9&mpshare=1&scene=1&srcid=0801r6UviPLPMgo9oEcOpUMD>

\* <https://github.com/radhikagupta5/AzureStack-QuickStart-Templates/tree/radhikgu-acs/101-acsengine-kubernetes-1803>

\* <https://docs.microsoft.com/en-us/azure/azure-stack/user/azure-stack-solution-template-kubernetes-deploy>

错误参考：

<https://github.com/Azure/acs-engine/issues/2591>

<https://github.com/Azure/acs-engine/issues/1806>

Use az cli with azure stack

<https://docs.microsoft.com/en-us/azure/azure-stack/user/azure-stack-version-profiles-azurecli2>