**Java启用ARM虚拟机诊断**

**问题描述：**

希望通过Java语言实现在创建ARM虚拟机时开启诊断，并配置相关指标。



**实现思路：**

通过调研最高版本的JAVA SDK（1.1.0）源码发现，SDK层面并未提供任启动诊断和配置诊断的相关接口。不过官方还是提供了相关的REST接口，参考：

* [Add or update a virtual machine extension](https://docs.microsoft.com/en-us/rest/api/compute/extensions/extensions-add-or-update)

这个接口为我们提供了向现有的ARM虚拟机添加或更新虚拟机扩展（vm extension）的功能，而我们在Portal上配置的“诊断功能”其实是一个名为“IaaSDiagnostics”的一个扩展。结合上述的REST接口，这个REST请求示例如下：

***https://management.chinacloudapi.cn/subscriptions/<subId>/resourceGroups/<groupName>/providers/Microsoft.Compute/virtualMachines/<vmName>/extensions/IaaSDiagnostics?api-version=2016-03-30***

这是一个Put方法请求，请求的参数主要包括Header和Request Body两部分的设置，官方连接中提供的解释略有不足，以下是我们关于相关参数的总结：

* Header：主要包括Authorization、Content-Type及Host的设置

1. Authorization：提供认证的凭据，如何获取这个值请参认证凭据小节
2. Content-Type：指定为application/json，以JSON数据返回响应
3. Host：指定为“management.chinacloudapi.cn”，终结点地址

* Request Body：关于扩展的配置信息，可以参考[VM 扩展配置示例](https://docs.microsoft.com/zh-cn/azure/virtual-machines/windows/extensions-configuration-samples)或以下模板

***{***

***'type':'Microsoft.Compute/virtualMachines/extensions',***

***'id':'/subscriptions/<subId>/resourceGroups/<groupName>/providers/Microsoft.Compute/virtualMachines/<vmName>/extensions/IaaSDiagnostics',***

***'location':'<location>',***

***'name':'IaaSDiagnostics',***

***'properties':{'***

***type':'IaaSDiagnostics',***

***'publisher':'Microsoft.Azure.Diagnostics',***

***'typeHandlerVersion':'1.5',***

***'settings':{***

***'StorageAccount':'<storageAccountName>',***

***'xmlCfg':'<诊断XML配置的Base64编码值>'***

***},***

***'protectedSettings':{***

***'storageAccountName':'<storageAccountName>',***

***'storageAccountKey':'<storageAccountKey>',***

***'storageAccountEndPoint':'https://core.chinacloudapi.cn'},***

***'autoUpgradeMinorVersion':true,***

***'instanceView':null***

***}***

***}***

1. subId：虚拟机所属的订阅ID
2. groupName:虚拟机所属的资源组名称
3. vmName：虚拟机名称
4. location：虚拟机地缘位置，示例“China North”或“China East”
5. storageAccountName：虚拟诊断数据存储账号名称
6. storageAccountKey：虚拟诊断数据存储账号秘钥
7. 诊断XML配置的Base64编码值：这个值是指是将诊断指标配置做Base64编码计算得到的值。关于诊断指标配置参考[诊断 1.2 配置架构](https://docs.microsoft.com/zh-cn/azure/monitoring-and-diagnostics/azure-diagnostics-schema-1dot2)、[诊断 1.3 及更高版本的配置架构](https://docs.microsoft.com/zh-cn/azure/monitoring-and-diagnostics/azure-diagnostics-schema-1dot3-and-later)。同时，github提供了一测试过的用例：<https://github.com/hello-azure/azure-vmop-demo/blob/master/src/main/java/geo/azure/test/WadCfg.xml>

接下来我们要做的就是，通过Java来模拟提交该请求，实现为虚拟机开启诊断功能。

**认证凭据：**

Authorization Header的值是基于AAD 方式验证返回的Token字符串，详细参考以下步骤：

1. 通过PowerShell获取认证凭据：

# 1.在PowerShell中，登录Azure 账户

Login-AzureRmAccount -EnvironmentName AzureChinaCloud

# 2.选择当前订阅ID

Set-AzureRmContext -SubscriptionId "订阅ID"

# 3.创建AD Application

$azureAdApplication = New-AzureRmADApplication -DisplayName "georgeapp" -HomePage "https://www.georgeapp.org" -IdentifierUris "https://www.georgeapp.org/example" -Password "1QAZxsw2"

$azureAdApplication

# 4.为你的AD应用创建服务凭证

New-AzureRmADServicePrincipal -ApplicationId $azureAdApplication.ApplicationId

# 5.为服务凭证授权。如果想了解更多内容，请参考：<https://azure.microsoft.com/en-us/documentation/articles/role-based-access-control-what-is/>

New-AzureRmRoleAssignment -RoleDefinitionName Contributor -ServicePrincipalName $azureAdApplication.ApplicationId

执行上述操作后，就可以获取可用的认证凭据信息：

* + tentantId：第二步执行完成后返回
  + subId：第二步指定的订阅ID
  + clientId：第三步执行完成后返回
  + clientSecret：第三步中设置的Password

1. 获取Token

**private** String getAccessToken() {

AuthenticationContext context = **null**;

AuthenticationResult result = **null**;

ExecutorService service = **null**;

service = Executors.*newFixedThreadPool*(1);

**try** {

context = **new** AuthenticationContext(

String.*format*("%s/%s",

"https://login.chinacloudapi.cn", tentantId),

**true**,

service);

ClientCredential cred = **new** ClientCredential(

clientId,

clientSecret);

Future<AuthenticationResult> future =

context.acquireToken(

"https://management.chinacloudapi.cn/", cred,

**null**);

result = future.get();

} **catch** (MalformedURLException e) {

e.printStackTrace();

} **catch** (InterruptedException e) {

e.printStackTrace();

} **catch** (ExecutionException e) {

e.printStackTrace();

} **finally** {

service.shutdown();

}

**return** result.getAccessToken();

}

1. Authorization: Bearer <Token>，以下是一个实际的示例

Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Im0yZmFQUUZkQzFEVGRmWU1pb09kaHdSblFUMCIsImtpZCI6Im0yZmFQUUZkQzFEVGRmWU1pb09kaHdSblFUMCJ9..C3n47Eoqvo65hYQa9UrZ6yRmedPA4HM96ZEoIQFkyeaql-ezNJVklkO0qUjA567REUJnFvQ\_zSDPufLvbrO-VAqCto3w3MN4Pjlkk-4qGTNEAzeoS6k4LOC0-YHHnP\_DJGv159IyUoukArNNnWrMs4ELTOw\_SbhqB5usTNs7N1GPoZPJCwj5wYryFVlwYA5PN\_o0ysIEj3XcdMvY18SEhsz663EJzxe-fwA0XRn7wejbbbTtXlTxvQfno5ZcB8t1IraL51CbZqyNyojc4Ya7n3ZsW6sX12Tt5e0DpMO4QpkIdyrSo1xZ3Mi7uJBAJIol-XIXnrO93\_8xTfjn5Mu8\_g

**完整示例：**

代码示例已经托管在github，请点击[azure-vmop-demo](https://github.com/hello-azure/azure-vmop-demo)查看或下载完整代码示例。

**运行测试：**

String resourceName ="geogroup";

String vmName ="geowin-test-005";

String vmLocation ="China North";

String storageAccountName = "";

String storageAccountKey= "";

VmDiagnosticOperation diagnosticOperation = **new** VmDiagnosticOperation(

"tenant id",

"client id",

"client secret",

"sub id");

String result = diagnosticOperation.EnableVMDiagnostic(

resourceName,

vmName,

vmLocation,

storageAccountName,

storageAccountKey);

System.***out***.println(result);

