**题目:**java调用Rest api 设置经典Linux 虚拟机的实例启停

**描述**：通过rest api 设置经典Linux 虚拟机实例的启停。在调用此api 时需要通过AAD获取token，但是因为mooncake 中通过 AAD的application 获取到的token 无法操作经典api，所以需要调用powershell 的 client id和管理员的用户名密码获取token 才可以。

**参考资料：https://msdn.microsoft.com/en-us/library/jj157189.aspx**

**前提：**创建一台linux 经典虚拟机

**空间引用：**

**import** java.io.DataOutputStream;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.net.URI;

**import** java.net.URISyntaxException;

**import** java.net.URL;

**import** java.security.KeyManagementException;

**import** java.security.\*;

**import** java.security.KeyStoreException;

**import** java.security.NoSuchAlgorithmException;

**import** java.security.UnrecoverableKeyException;

**import** java.security.cert.X509Certificate;

**import** java.util.HashMap;

**import** java.util.Map;

**import** java.util.Scanner;

**import** java.util.concurrent.ExecutionException;

**import** java.util.concurrent.ExecutorService;

**import** java.util.concurrent.Executors;

**import** java.util.concurrent.Future;

**import** javax.net.ssl.\*;

**import** javax.net.ssl.KeyManagerFactory;

**import** javax.net.ssl.SSLSocketFactory;

**import** javax.net.ssl.TrustManager;

**import** org.codehaus.jackson.map.ObjectMapper;

**java代码方法：**

public void GetToken() {

ExecutorService service = Executors.*newFixedThreadPool*(1);

AuthenticationContext ac = **new** AuthenticationContext("<https://login.chinacloudapi.cn/tenantID>", **true**, service);

Future<AuthenticationResult> future = ac.acquireToken("<https://management.core.chinacloudapi.cn/>", "1950a258-227b-4e31-a9cf-717495945fc2", "username", "password", **null**);

AuthenticationResult result = future.get();

String token = result.getAccessToken();

*rest*(token);

}

**public** **static** **void** rest(String accessToken) **throws** IOException{

URL url = **new** URL(String.*format*("https://management.core.chinacloudapi.cn/subID/services/hostedservices/{hostedservices}/deployments/{deployments}/roleinstances/{roleinstancesName}/Operations"));

HttpsURLConnection conn = (HttpsURLConnection)url.openConnection();

conn.setRequestProperty("x-ms-version", "2013-06-01");

conn.setRequestProperty("Authorization", "Bearer " + accessToken);

conn.setRequestProperty("Content-Type", "application/xml");

//StartRole

String roleInstance = **new** String("<StartRoleOperation xmlns=\"<http://schemas.microsoft.com/windowsazure\>" xmlns:i=\"http://www.w3.org/2001/XMLSchema-instance\">\n" + " <OperationType>StartRoleOperation</OperationType>\n" + "</StartRoleOperation>");

//ShutdownRole

String roleins = **new** String( "<ShutdownRoleOperation xmlns=\"<http://schemas.microsoft.com/windowsazure\>" xmlns:i=\"http://www.w3.org/2001/XMLSchema-instance\">" +"<OperationType>ShutdownRoleOperation</OperationType>" + "<PostShutdownAction>StoppedDeallocated</PostShutdownAction>" + "</ShutdownRoleOperation>");

**byte**[] data = roleInstance.getBytes();

conn.setDoOutput(**true**);

conn.setRequestMethod("POST");

**if** (data != **null**)

{

DataOutputStream requestStream = **new** DataOutputStream(conn.getOutputStream());

requestStream.write(data);

requestStream.flush();

requestStream.close();

}

String mess = conn.getResponseMessage();

**int** code = conn.getResponseCode();

InputStream input = conn.getErrorStream();

**if** (input == **null**)

input = conn.getInputStream();

String response = **null**;

**try** (Scanner scanner = **new** Scanner(input)) {

scanner.useDelimiter("[\\Z](file:///\\Z)");

response = scanner.next();

scanner.close();

input.close();

}

}