Azure_SDK_Java_Sample3_Create_Windows_VirtualMachine

0. Prerequisite

For creating a Java blank application in VSCode see this repos:

https://github.com/luiscoco/Java_Maven_Sample

https://github.com/luiscoco/Azure_SDK_Java_Sample1_CreateResourceGroup

1. Main.java

```
package com.example;

import com.azure.core.credential.TokenCredential;
import com.azure.core.http.policy.HttpLogDetailLevel;
import com.azure.core.management.AzureEnvironment;
import com.azure.identity.DefaultAzureCredentialBuilder;
import com.azure.resourcemanager.AzureResourceManager;
import com.azure.resourcemanager.compute.models.Disk;
import com.azure.resourcemanager.compute.models.KnownLinuxVirtualMachineImage;
import com.azure.resourcemanager.compute.models.KnownWindowsVirtualMachineImage;
import com.azure.resourcemanager.compute.models.VirtualMachine;
import com.azure.resourcemanager.compute.models.VirtualMachineSizeTypes;
import com.azure.resourcemanager.network.models.Network;
import com.azure.resourcemanager.network.models.NetworkInterface;
import com.azure.resourcemanager.network.models.NetworkSecurityGroup;
import com.azure.resourcemanager.network.models.SecurityRuleProtocol;
import com.azure.core.management.Region;
import com.azure.resourcemanager.resources.fluentcore.model.Creatable;
import com.azure.core.management.profile.AzureProfile;

import java.util.Date;

public class Main {

    public static String randomResourceName(AzureResourceManager azure, String prefix, int max
        return azure.resourceGroups().manager().internalContext().randomResourceName(prefix, m
    }

    public static void main(String[] args) {

        //=============================================================
        // 1. Azure SDK for Java: Azure Authentication
        //=============================================================
```

```
AzureProfile profile = new AzureProfile(AzureEnvironment.AZURE);
TokenCredential credential = new DefaultAzureCredentialBuilder()
    .authorityHost(profile.getEnvironment().getActiveDirectoryEndpoint())
    .build();
AzureResourceManager azureResourceManager = AzureResourceManager
    .authenticate(credential, profile)
    .withDefaultSubscription();

//=============================================================
// 2. Azure SDK for Java: Create a resource group
//=============================================================

final String rgName = "myFirstResourceGroup";
System.out.println("Creating a resource group with name: " + rgName);

azureResourceManager.resourceGroups().define(rgName)
        .withRegion(Region.US_WEST)
        .create();

//-------------------------------------------------------------------------------------
// 3. Prepare a creatable data disk for VM
//-------------------------------------------------------------------------------------

Creatable<Disk> dataDiskCreatable = azureResourceManager.disks().define(Main.randomRes
        .withRegion(Region.US_WEST)
        .withExistingResourceGroup(rgName)
        .withData()
        .withSizeInGB(100);

//-------------------------------------------------------------------------------------
// 4. Create a data disk to attach to VM
//-------------------------------------------------------------------------------------

Disk dataDisk = azureResourceManager.disks()
        .define(Main.randomResourceName(azureResourceManager, "dsk-", 15))
            .withRegion(Region.US_WEST)
            .withNewResourceGroup(rgName)
            .withData()
            .withSizeInGB(50)
            .create();

System.out.println("Creating a Windows VM");

Date t1 = new Date();

//=============================================================================
// 5. Create a network security group contains two rules
// - ALLOW-SSH- allows RDP traffic into the VM
// - ALLOW-WEB- allows HTTP traffic OutBound
// - ALLOW-WEB- allows HTTPS traffic OutBound
//=============================================================================
```

```
System.out.println("Creating a security group for the front end - allows SSH and HTTP"
NetworkSecurityGroup NSG = azureResourceManager.networkSecurityGroups().define("NSGNam
        .withRegion(Region.US_WEST)
        .withNewResourceGroup(rgName)
        .defineRule("ALLOW-RDP")
        .allowInbound()
        .fromAnyAddress()
        .fromAnyPort()
        .toAnyAddress()
        .toPort(3389)
        .withProtocol(SecurityRuleProtocol.TCP)
        .withPriority(100)
        .withDescription("Allow RDP")
        .attach()
        .defineRule("ALLOW-HTTP")
        .allowOutbound()
        .fromAnyAddress()
        .fromAnyPort()
        .toAnyAddress()
        .toPort(80)
        .withProtocol(SecurityRuleProtocol.TCP)
        .withPriority(101)
        .withDescription("Allow HTTP")
        .attach()
        .defineRule("ALLOW-HTTS")
        .allowOutbound()
        .fromAnyAddress()
        .fromAnyPort()
        .toAnyAddress()
        .toPort(443)
        .withProtocol(SecurityRuleProtocol.TCP)
        .withPriority(102)
        .withDescription("Allow HTTS")
        .attach()
        .create();

System.out.println("Created a security group for the front end: " + NSG.id());

//============================================================================
// 6. Create a Network
//============================================================================

Network network = azureResourceManager.networks().define("mynetwork")
.withRegion(Region.US_WEST)
.withNewResourceGroup()
.withAddressSpace("10.0.0.0/28")
.withSubnet("subnet1", "10.0.0.0/29")
.withDnsServer("8.8.8.8")  // Add your DNS servers here
.withDnsServer("8.8.4.4")
.withDnsServer("10.1.1.1")
.withDnsServer("10.1.2.4")
.create();
```

```
            System.out.println("Created a virtual network: " + network.id());

            //============================================================================
            // 7. Create a network interface and apply the network security group created in the s
            //============================================================================

            String publicIPAddressLeafDNS1 = "myPublicIPAddress";

            System.out.println("Creating a network interface for the back end");

            NetworkInterface networkInterface = azureResourceManager.networkInterfaces().define("m
                    .withRegion(Region.US_WEST)
                    .withExistingResourceGroup(rgName)
                    .withExistingPrimaryNetwork(network)
                    .withSubnet("subnet1")
                    .withPrimaryPrivateIPAddressDynamic()
                    .withNewPrimaryPublicIPAddress(publicIPAddressLeafDNS1)
                    .withExistingNetworkSecurityGroup(NSG)
                    .create();

            //----------------------------------------------------------------------------
            // 8. Create a VM
            //----------------------------------------------------------------------------

            String windowsVMName = "mynewvm19742000";
            String userName = "azureuser";
            String password = "Luiscoco123456";

            VirtualMachine windowsVM = azureResourceManager.virtualMachines()
                    .define(windowsVMName)
                        .withRegion(Region.US_WEST)
                        .withNewResourceGroup(rgName)
                        .withExistingPrimaryNetworkInterface(networkInterface)
                        .withPopularWindowsImage(KnownWindowsVirtualMachineImage.WINDOWS_SERVER_20
                        .withAdminUsername(userName)
                        .withAdminPassword(password)
                        .withNewDataDisk(10)
                        .withNewDataDisk(dataDiskCreatable)
                        .withExistingDataDisk(dataDisk)
                        .withSize(VirtualMachineSizeTypes.fromString("Standard_E2s_v3"))
                        .create();

            Date t2 = new Date();
            System.out.println("Created VM: (took " + ((t2.getTime() - t1.getTime()) / 1000) + " s
        }
    }
```

2. pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/mav
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.example</groupId>
    <artifactId>demo</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
    </properties>

     <dependencies>
        <!-- Azure SDK Dependencies -->
       <dependency>
            <groupId>com.azure</groupId>
            <artifactId>azure-identity</artifactId>
            <version>1.11.1</version>
        </dependency>
        <dependency>
            <groupId>com.azure.resourcemanager</groupId>
            <artifactId>azure-resourcemanager</artifactId>
            <version>2.33.0</version>
        </dependency>
        <dependency>
            <groupId>com.azure.resourcemanager</groupId>
            <artifactId>azure-resourcemanager-compute</artifactId>
            <version>2.33.0</version>
        </dependency>
        <dependency>
            <groupId>commons-net</groupId>
            <artifactId>commons-net</artifactId>
            <version>3.6</version>
        </dependency>
    </dependencies>

    <build>
            <plugins>
                <plugin>
                    <!-- Maven JAR Plugin Configuration -->
                    <groupId>org.apache.maven.plugins</groupId>
                    <artifactId>maven-jar-plugin</artifactId>
                    <version>3.1.0</version>
                    <configuration>
                        <archive>
                            <manifest>
                                <mainClass>com.example.Main</mainClass>
                            </manifest>
                        </archive>
```

```
                </configuration>
            </plugin>

            <plugin>
                <!-- Exec Maven Plugin Configuration -->
                <groupId>org.codehaus.mojo</groupId>
                <artifactId>exec-maven-plugin</artifactId>
                <version>3.1.0</version>
                <configuration>
                    <mainClass>com.example.Main</mainClass>
                </configuration>
                <executions>
                    <execution>
                        <goals>
                            <goal>java</goal>
                        </goals>
                    </execution>
                </executions>
            </plugin>
        </plugins>
    </build>

</project>
```