构建混合应用方式之 - 混合连接

前面介绍了通过WCF中继构建混合应用的方式，由于对WCF的依赖，使得其使用有一定的局限性，基本上只适用于本地服务是WCF的.NET应用。而混合连接则弥补了这一块的缺陷，除了支持原有WCF中继的功能之外，还提供了多平台多语言的支持，这只要是因为混合连接是基于开放标准协议web sockets实现的。

以下是与WCF中继的一个对比表：

|  |  |  |
| --- | --- | --- |
|  | WCF中继 | 混合连接 |
| 建立本地和云的安全连接 | Checkmark | Checkmark |
| 无需修改防火墙添加入站端口 | Checkmark | Checkmark |
| 无需对网络配置做重大修改 | Checkmark | Checkmark |
| 基于开放标准协议 | Close（只支持WCF） | Checkmark |
| 跨平台支持 | Close（只支持Windows） | Checkmark |
| 多语言支持 | Close（只支持.NET） | Checkmark |

下面主要分三种情况来介绍混合连接的使用：

1. 基于混合连接SDK
2. 基于Azure Web App混合连接工具
3. 基于PortBridge示例程序

# 创建混合连接

可以登陆[Azure中国门户网站](https://portal.azure.cn/)，按照界面提示一步步创建。也可以通过PowerShell命令，如下。

|  |
| --- |
| *# login to Azure China with your account*  Login-AzureRmAccount -Environment AzureChinaCloud  $rgName = "relaydemorg"  $namespaceName = "relaydemons"  $location = "China East"  $hcName = "hcdemo"  New-AzureRmResourceGroup -Name $rgName -Location $location  New-AzureRmRelayNamespace -ResourceGroupName $rgName -Name $namespaceName -Location $location  New-AzureRmRelayHybridConnection -Namespace $namespaceName -ResourceGroupName $rgName -Name $hcName -RequiresClientAuthorization $true |

创建出来混合连接字符串就是sb://relaydemons.servicebus.chinacloudapi.cn/hcdemo

# 基于混合连接SDK

目前微软官方提供了两个SDK，一个是.NET语言版本，另一个是Node.JS版本。两个SDK也都开源在Github上：[azure-relay-net](https://github.com/Azure/azure-relay-dotnet)，[azure-relay-node](https://github.com/Azure/azure-relay-node)。

### 本地服务（监听者）

首先本地服务端作为监听者，需要要使用具有监听权限的连接字符串，创建一个出站的WebSocket连接，也就是注册到Azure中继服务上。如果本地有多个监听者，那么进入的访问请求会随机地发送。一个混合连接最多支持25个监听者。

|  |
| --- |
| private const string ConnectionString = "connection string with listen permission";  // Listen  var listener = new HybridConnectionListener(ConnectionString);  await listener.OpenAsync();  // Accept  HybridConnectionStream relayConnection = await listener.AcceptConnectionAsync();  // Read  var reader = new StreamReader(relayConnection);  var line = await reader.ReadLineAsync();  // Output  Console.WriteLine(line);  // Close  await relayConnection.CloseAsync(CancellationToken.None); |

### 外部服务（发送者）

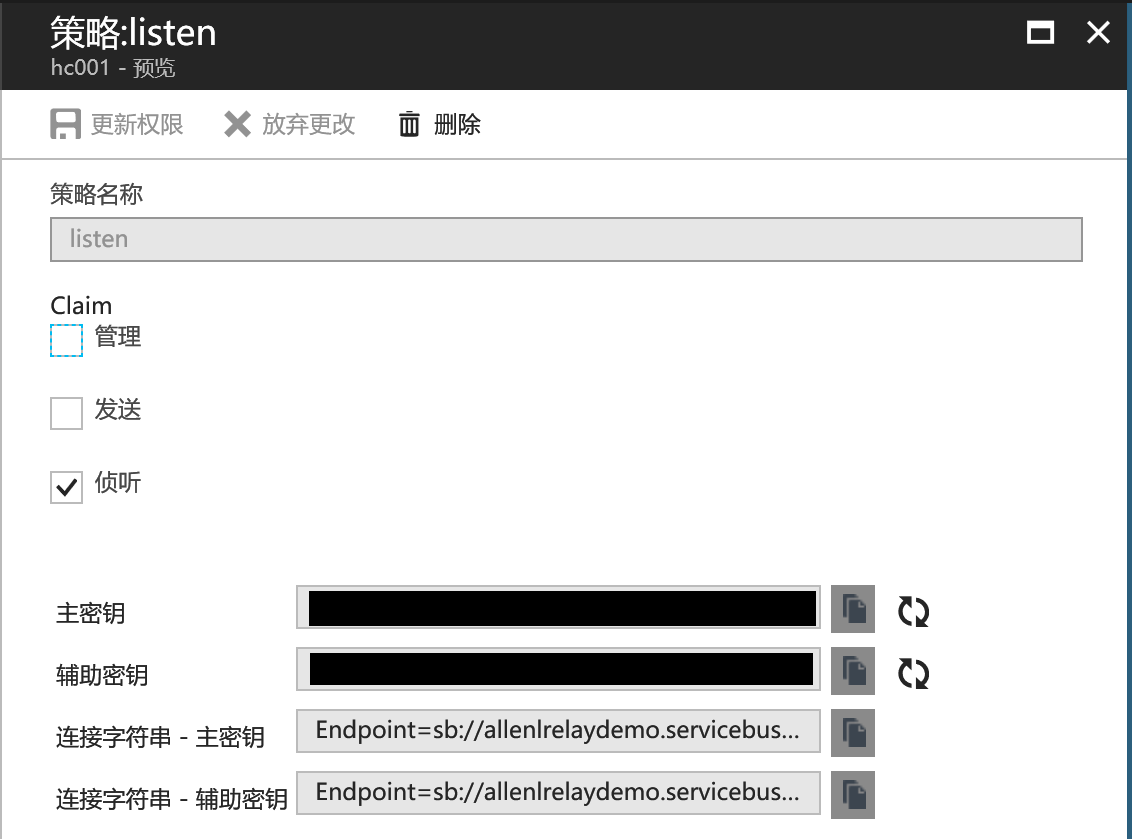
发送者提供具有发送权限的连接字符串（不是必须如果建立混合连接时没有要求客户端认证，不过从安全角度考虑，推荐要求认证），与混合连接服务建立WebSocket连接，最终会与一个监听者建立起连接，从而进行沟通。

|  |
| --- |
| private const string ConnectionString = "connection string with send permission";  // Connect  var client = new HybridConnectionClient(ConnectionString);  HybridConnectionStream relayConnection = await client.CreateConnectionAsync();  // Write  var writer = new StreamWriter(relayConnection) { AutoFlush = true };  await writer.WriteLineAsync("hello from outside");  // Close  await relayConnection.CloseAsync(CancellationToken.None); |

当然，因为搭建的WebSocket是双向通道，所以本地服务端也可以发送消息到外部服务端的。

### 连接字符串

前面提到需要用到不同权限的连接字符串，总共有三种权限管理，监听和发送，可以根据需要创建不同权限搭配的安全访问策略，然后在它的具体界面里会自动生成相应的连接字符串，直接复制使用即可，如下图。



如果本地服务不能或者不希望被修改成调用SDK，比如云端Web App调用本地的Web Api或者SQL数据库，那该如何使用混合连接呢？Azure Web App已经整合里混合连接，可以很方便的配置，而不需要对本地服务做任何修改。

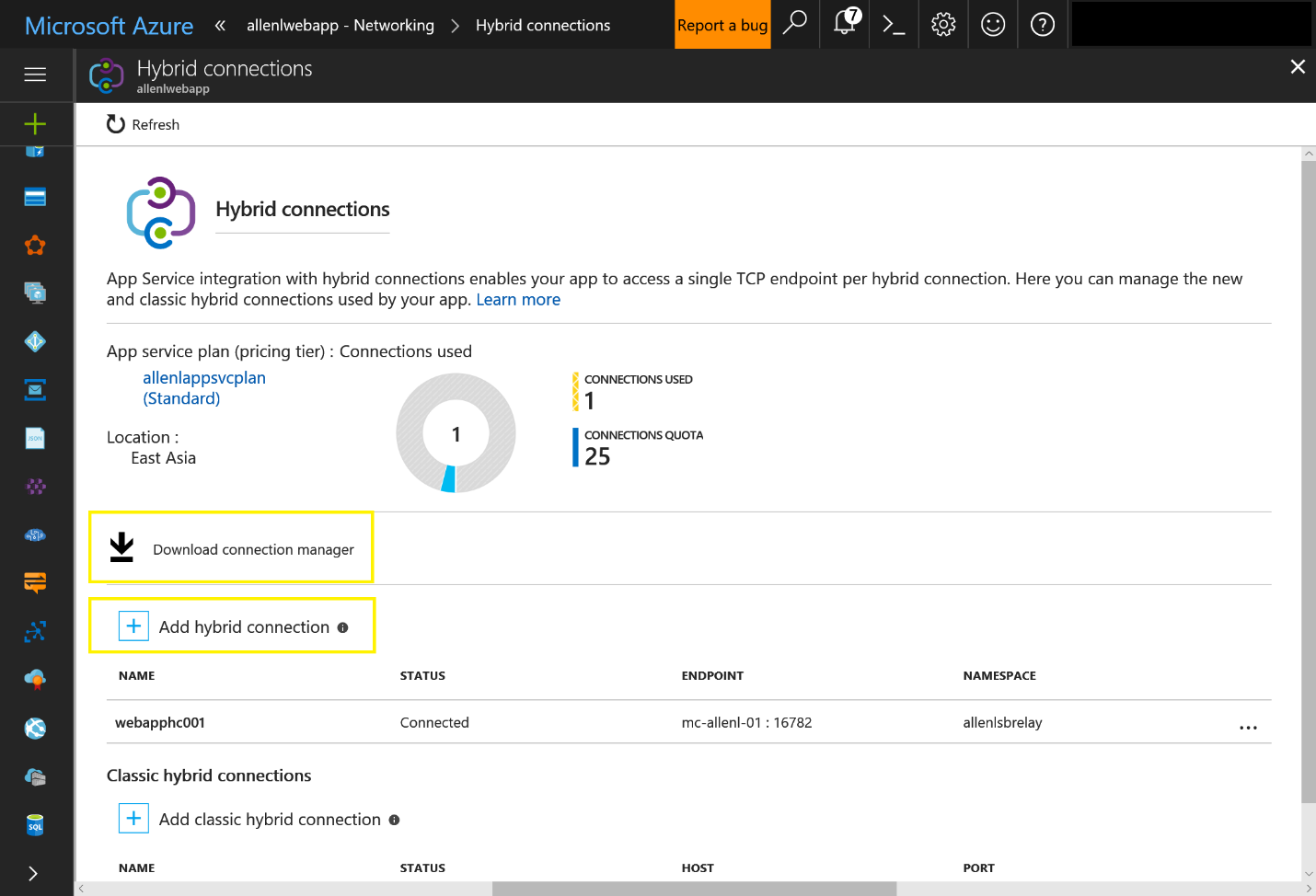
完整代码示例可查看[我的Github](https://github.com/allenhula/azure-china-get-started/tree/master/Relay/CSharp/Simple)。

# 基于Azure Web App混合连接工具

目前Azure中国还没有支持混合连接的整合，不过已经在路上了，应该会很快就可上线。以下用Azure全球来演示，最终Azure中国也会是同样的体验。

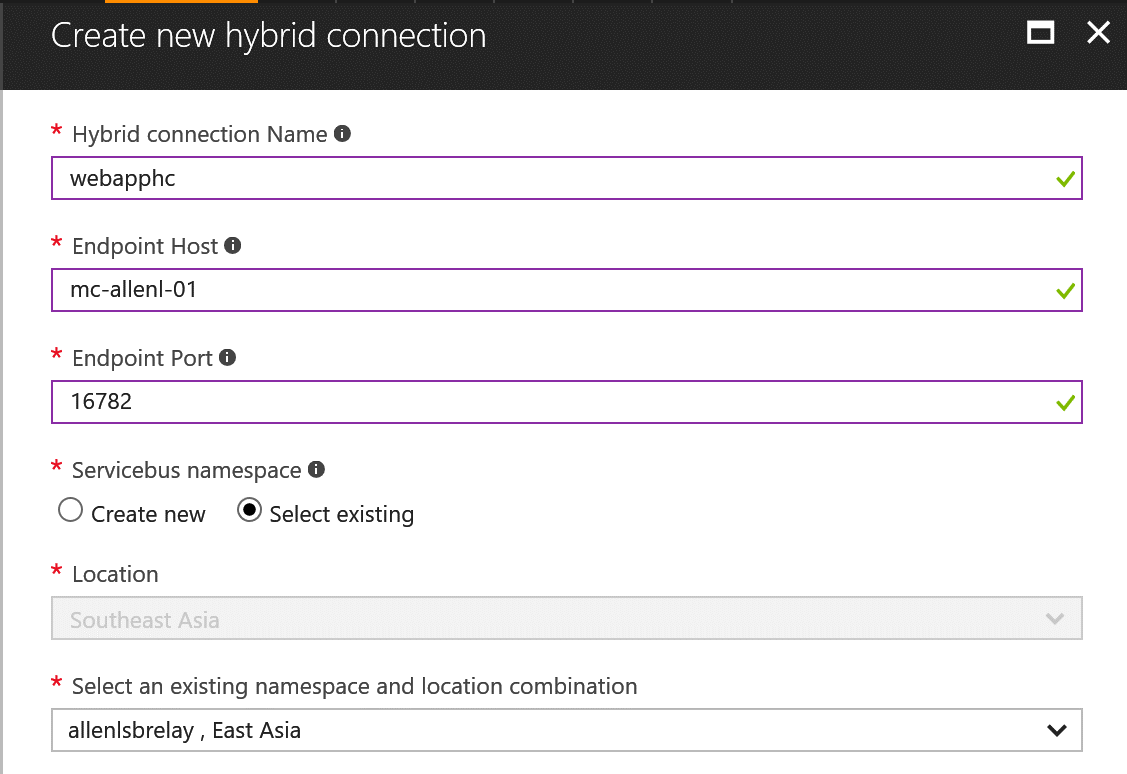
### 配置混合连接端节点

登陆azure门户网站，找到你的web app，在设置下面点击网络，在展开的页面中点击“配置你的混合连接端节点”，打开页面如下：

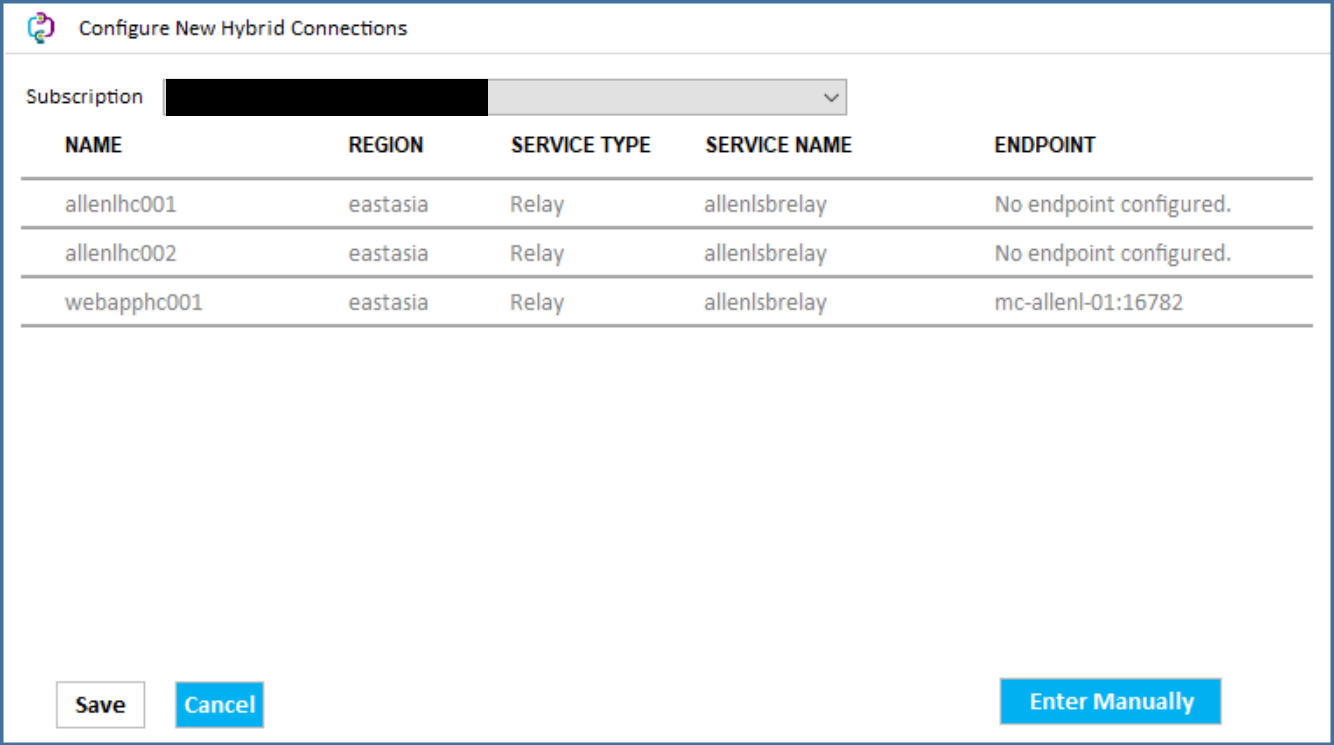


点击“下载连接管理工具”下载，后面会用到。

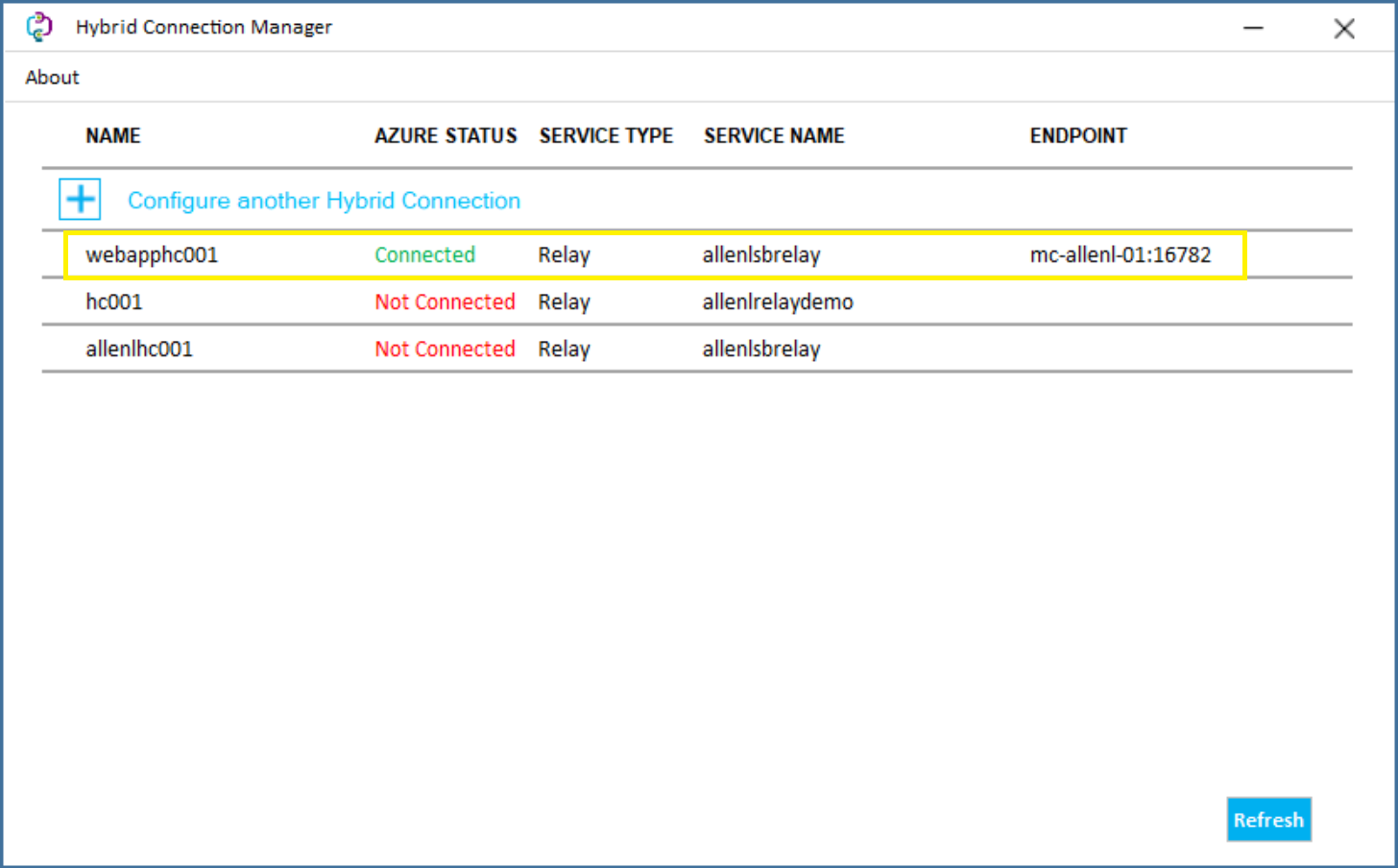
点击“添加混合连接”，在打开的页面里面点击添加，如下图。填入相应信息，其中名称可以自定义，而端节点主机必须是运行你本地服务的机器名称，端节点端口也是本地服务所在端口。命名空间选择前面已经创建好的，也可以这里创建新的。



创建完成后，把前面下载的连接管理工具安装在运行本地服务的机器上。然后打开混合连接管理工具，点击“Configure another Hybrid Connection”，这个时候会弹出登陆框，输入你的azure订阅账户登陆后，会显示当前订阅中已有的混合连接，如下图。



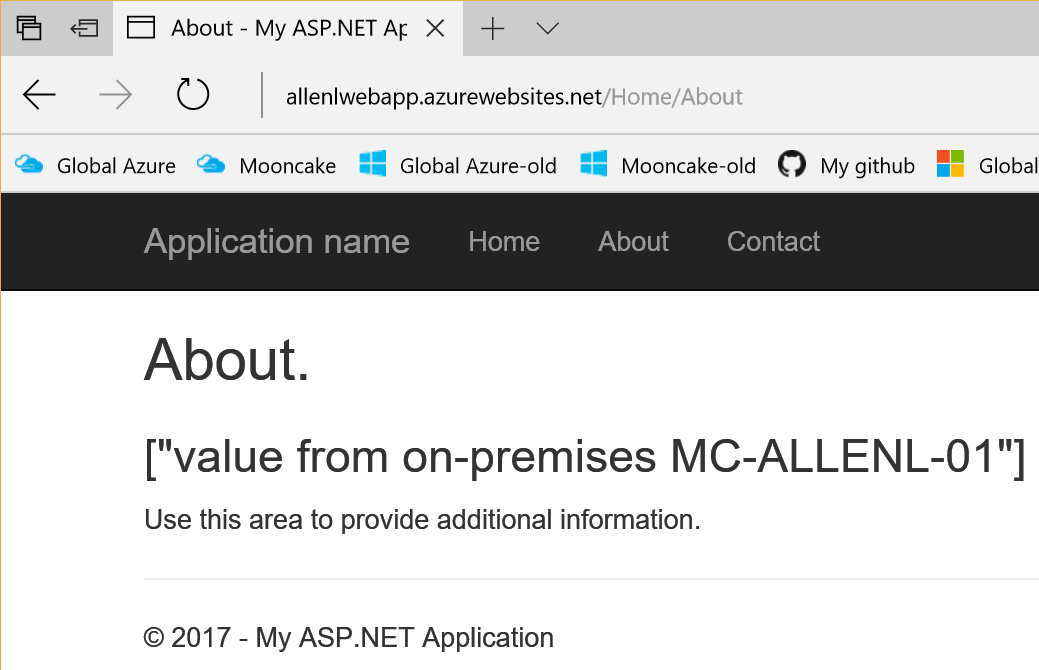
选择配置了正确端节点的混合连接，然后点击“Save”保存。这个混合连接就会将本地与云端web app建立起连接，状态显示为”Connected”。



这样，在web app里面就可以通过”端节点名称：端口“的方式来访问本地的服务了。比如我这里演示的，我在本地运行了一个web api服务（api/values）在16782端口，返回字符串”value from on-premises <本地机器名称>”，然后通过混合连接，云端web app就可以很简单的访问本地服务了，代码如下：

|  |
| --- |
| using (var httpClient = new HttpClient())  {  var onPremSvcUri = "http://mc-allenl-01:16782/api/values";  using (var response2 = httpClient.GetAsync(onPremSvcUri).Result)  {  if (response2.IsSuccessStatusCode)  {  ViewBag.Message = $"{response2.Content.ReadAsStringAsync().Result}";  }  }  } |

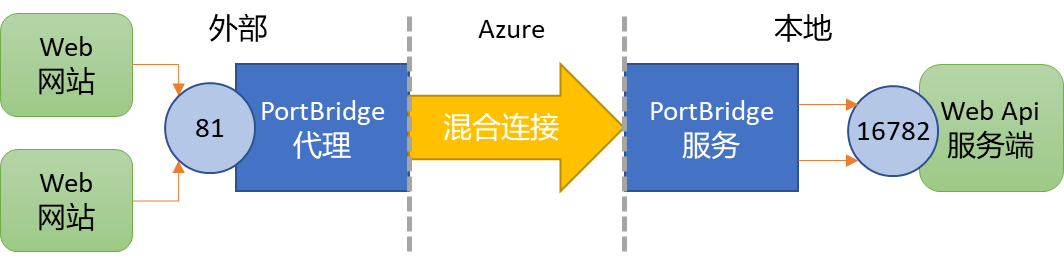
结果：



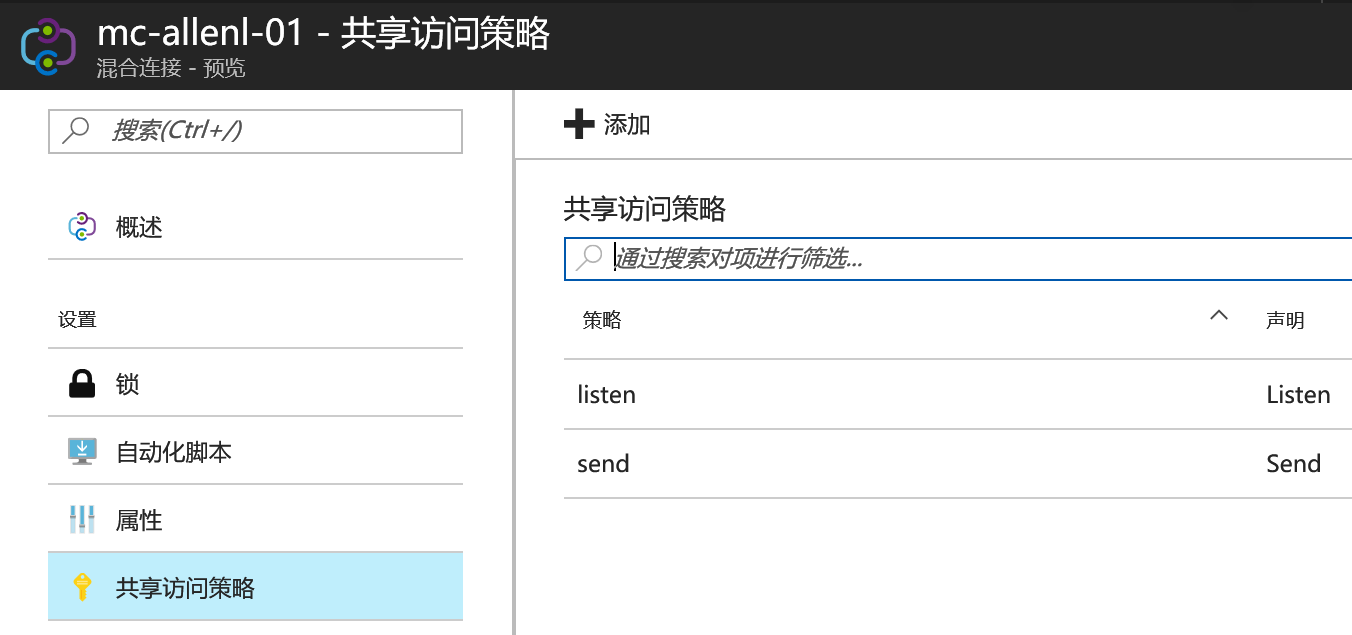
对于其他本地服务，比如SQL服务，也可以用同样的方式连接。

# 基于PortBridge示例程序

如果外部服务不是用Azure Web App，又该如何使用混合连接呢？可以通过混合连接来实现端口转发，从而建立连接通道。微软官方示例[PortBridge](https://github.com/Azure/azure-relay/tree/master/samples/Hybrid%20Connections/DotNet/portbridge)正是演示了这种功能。接下来，我们用它来演示在云端虚拟机中的web应用调用本地web api。



创建以本地服务所运行的机器为名称的混合连接（演示名称mc-allenl-01），如何创建请参照文章最开始的PowerShell命令。并分别创建发送和监听权限的共享访问策略。



将PortBridge下载并编译，修改PortBridgeServerAgent.exe.config文件中portBridge相关配置，其中targetHost配置成本地服务所运行的机器名称，端口则为本地服务所在端口，本示例配置如下。然后将整个文件夹复制到本地服务所运行的机器上，并双击PortBridgeServerAgent.exe运行。

|  |
| --- |
| <portBridge serviceBusNamespace="allenlrelaydemo.servicebus.chinacloudapi.cn" serviceBusAccessRuleName="listen" serviceBusAccessRuleKey="\*\*\*\*\*\*">  <hostMappings>  <add targetHost="mc-allenl-01" allowedPorts="16785"/>  </hostMappings>  </portBridge> |

修改PortBridgeClientAgent.exe.config文件中portBridgeAgent的相关配置，其中targetHost设置成本地服务所运行的机器名称，localTcpPort设为你期望使用访问的端口，remoteTcpPort则为本地服务所运行的端口，本示例配置如下。将整个文件夹复制到Azure虚拟机中，并双击PortBridgeClientAgent.exe运行。

|  |
| --- |
| <portBridgeAgent serviceBusNamespace="allenlrelaydemo.servicebus.chinacloudapi.cn" serviceBusAccessRuleName="send" serviceBusAccessRuleKey="\*\*\*\*\*\*">  <portMappings>  <port localTcpPort="81" targetHost="mc-allenl-01" remoteTcpPort="16782">  <firewallRules>  <rule source="127.0.0.1"/>  <rule sourceRangeBegin="10.0.0.0" sourceRangeEnd="10.255.255.255"/>  </firewallRules>  </port>  </portMappings>  </portBridgeAgent> |

在Azure虚拟机中访问http://localhost:81/api/values，没有运行PortBridge之前是会失败的，但开启PortBridge，也就是通过混合连接建立通道之后，就会访问成功，如下所示：

