**Tech Summit 2018**

**Hand­s-on lab**

用Azure SignalR Service开发实时应用

******Contents**

**OVERVIEW 3**

**Exercise 1: 创建一个AspNetCore SignalR的聊天室 5**

Task 1 – 下载源码 5

Task 2 – 使用code打开工程 5

Task 3 – 修改服务端代码 10

Task 4 – 修改客户端代码 13

**Exercise 2: 创建一个Azure SignalR Service 15**

Task 1 – 打开Azure Portal 15

Task 2 – 创建SignalR Service 16

**Exercise 3: 使用Azure SignalR Service 21**

Task 1 – 安装Azure SignalR Service SDK 21

Task 2 – 编辑源代码文件 21

Task 3 – 修改配置 21

Overview

Azure SignalR Service是一个完全托管的基于全新的AspNetCore SignalR的服务，提供了高可用和可伸缩的实时通讯方案。开发者只需要开发应用逻辑，SignalR框架就可以完成多协议的自动选择。

**Objectives**

* 1. 本实验将展示：
  + 创建一个AspNetCore SignalR的聊天室
  + 创建一个Azure SignalR Service
  + 修改聊天室，使用Azure SignalR Service

**System requirements**

* 1. 您需要准备下列软件来完成本次实验：
  + Windows 10(或其它操作系统)
  + DotNet Core 2.1 SDK
  + VSCode (latest update)
  + Git Bash

**Setup**

* 1. 安装步骤:
  2. 安装Microsoft Windows 10(或其它操作系统).
  3. 安装VS Code (<https://code.visualstudio.com/Download>).
  4. 安装DotNet Core 2.1 SDK (<https://www.microsoft.com/net/download/dotnet-core/2.1>)
  5. 安装Git Bash (<https://git-scm.com/downloads>)

**Exercises**

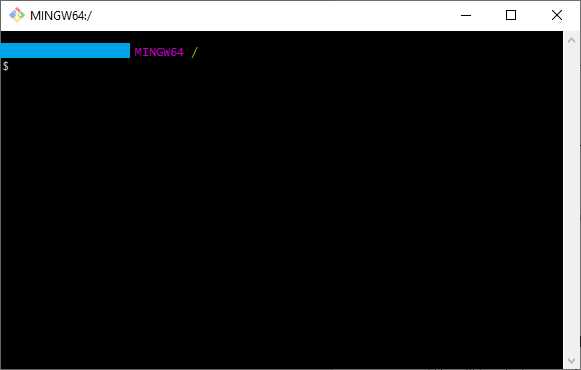
* 1. 本次实验包含一下内容:
  2. 创建一个AspNetCore SignalR的聊天室
  3. 创建一个Azure SignalR Service
  4. 修改聊天室，使用Azure SignalR Service
  5. 预计完成时间:  **30 to 45 minutes**.

Exercise 1: 创建一个AspNetCore SignalR的聊天室

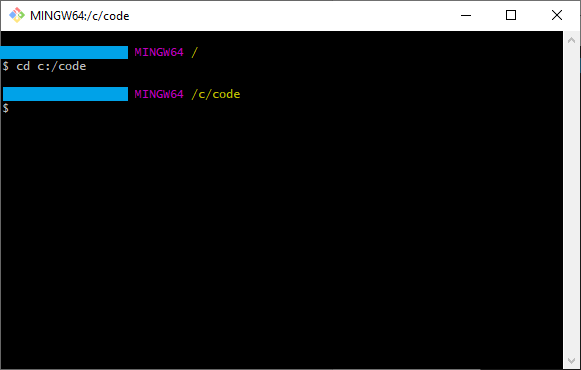
**Task 1 – 下载源码**

首先打开git bash，在一个适合的文件夹中clone本次使用的源代码

* 1. 打开git bash：



* 1. 找到一个合适的文件夹：



* 1. Clone源代码：

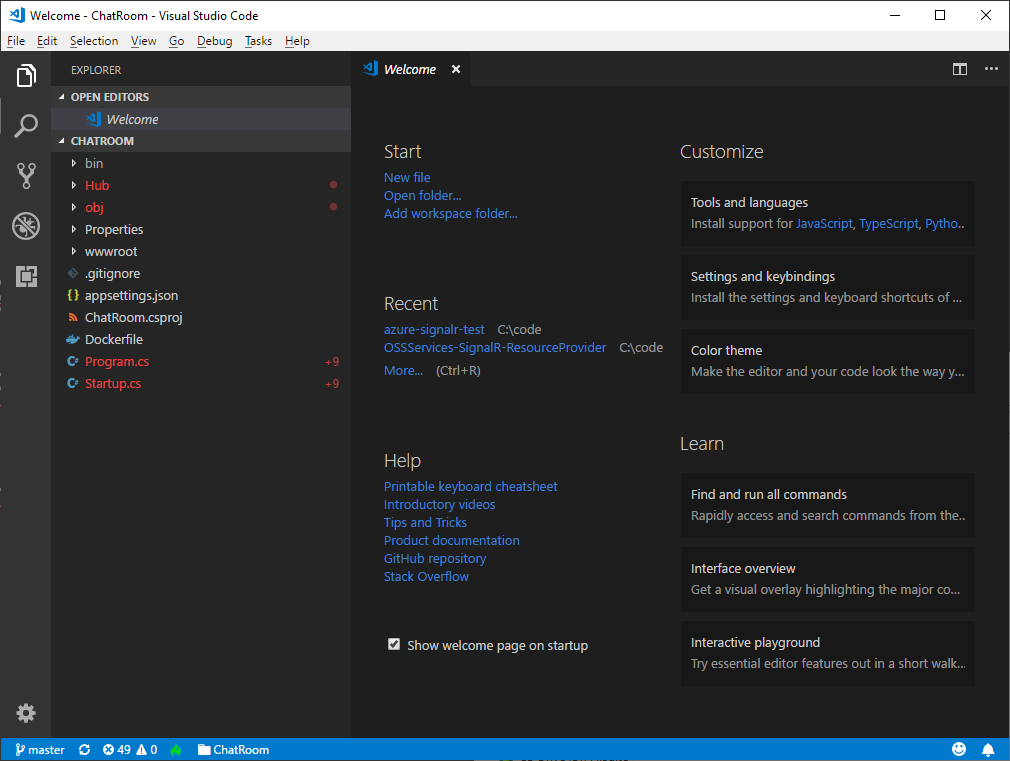
git clone https://github.com/vwxyzh/asrs-chatroom-demo

**Task 2 – 使用code打开工程**

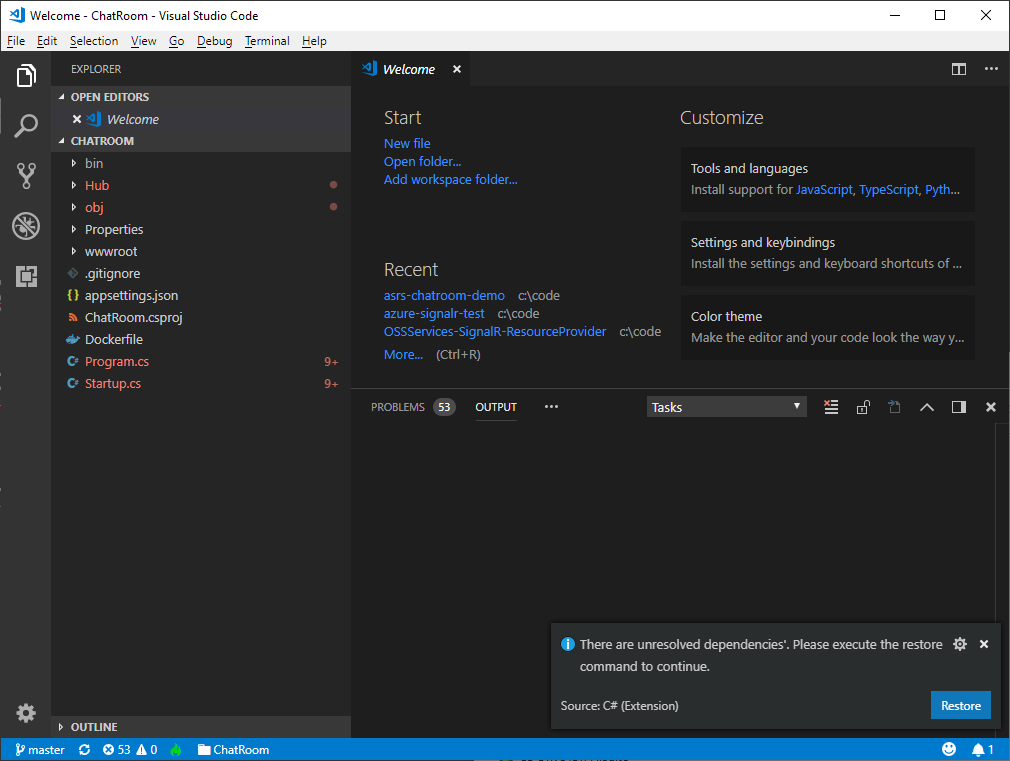
1. 用vs code打开ChatRoom目录：

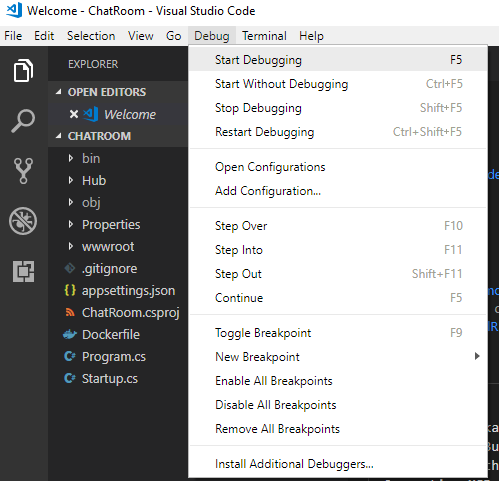
cd ChatRoom

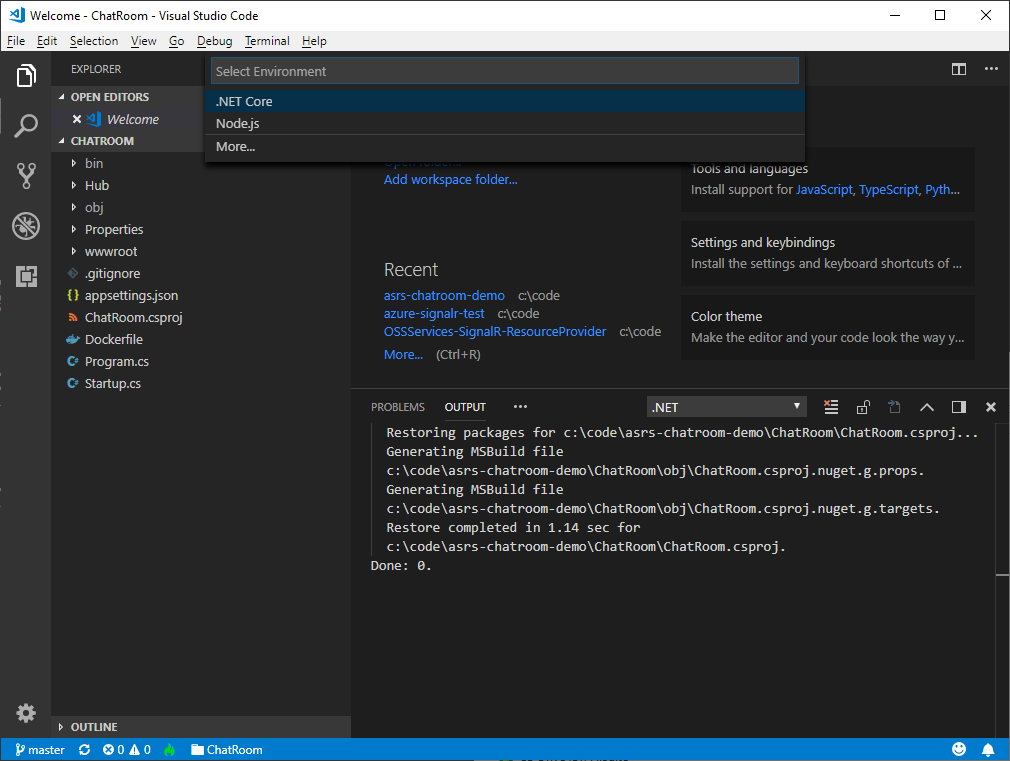
code .



1. 恢复依赖项和关联debugger.





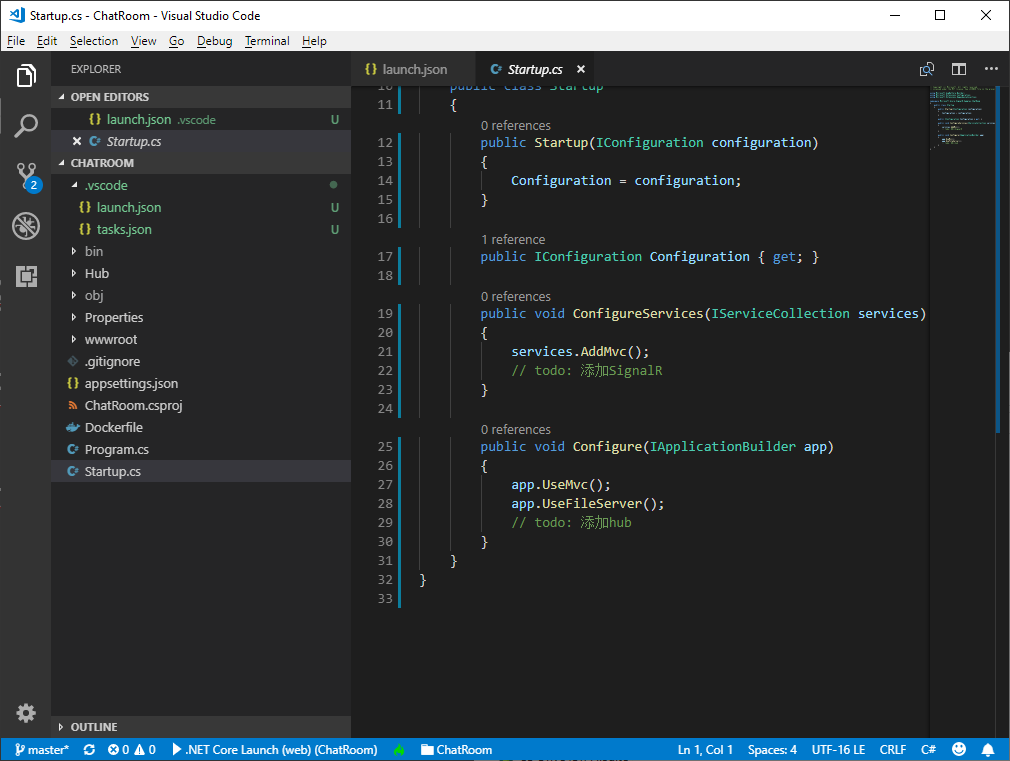


自动添加玩launch.json后，再次选择Start Debugging (F5)

Stop Debugging (Shift + F5)，返回Explore (Ctrl + Shift + E)

**Task 3 – 修改服务端代码**

1. 打开文件Startup.cs



添加SignalR:

services.AddSignalR();

添加Hub

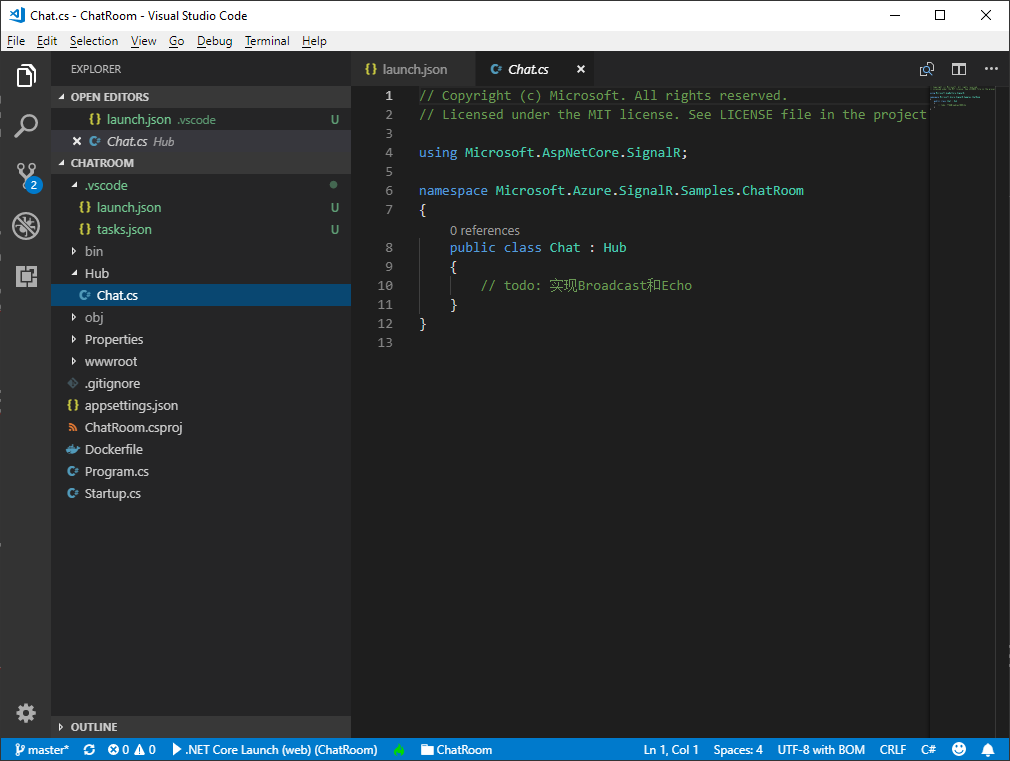
app.UseSignalR(routes =>

{

routes.MapHub<Chat>("/chat");

});

1. 打开文件Chat.cs



添加Broadcast方法：

public void BroadcastMessage(string name, string message)

{

Clients.All.SendAsync("broadcastMessage", name, message);

}

添加Echo方法：

public void Echo(string name, string message)

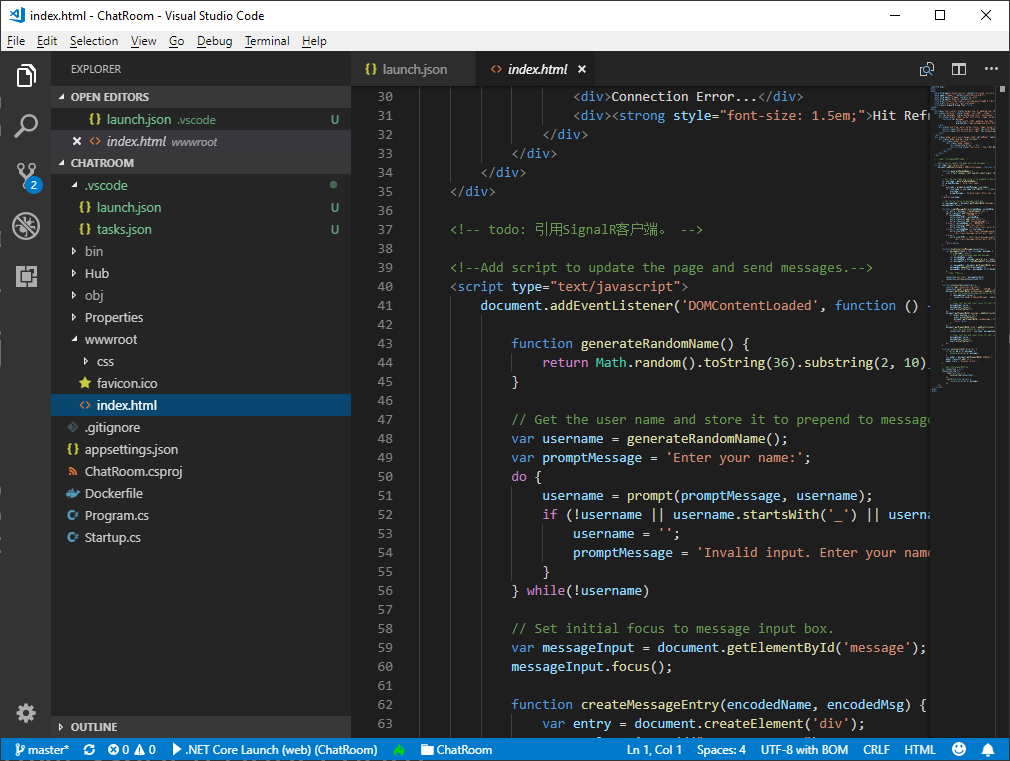
{

Clients.Client(Context.ConnectionId).SendAsync(“echo”, name, message + " (echo from server)");

}

**Task 4 – 修改客户端代码**

1. 打开文件index.html



1. 添加SignalR客户端：

<script type="text/javascript" src="https://cdn.jsdelivr.net/npm/@aspnet/signalr@1.0.0/dist/browser/signalr.min.js"></script>

1. 实现广播：

connection.on('broadcastMessage', messageCallback);

1. 配置signalR客户端：

var connection = new signalR.HubConnectionBuilder()

.withUrl('/chat')

.build();

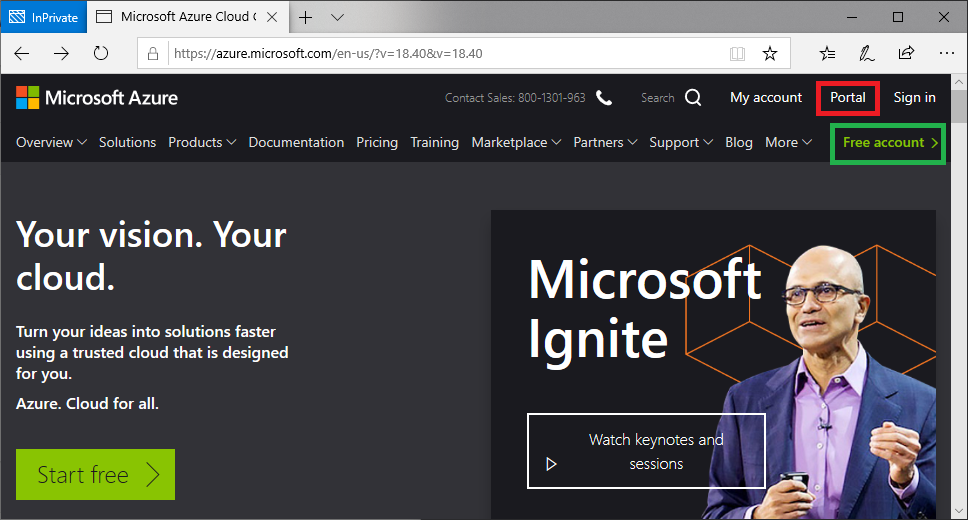
bindConnectionMessage(connection);

Exercise 2: 创建一个Azure SignalR Service

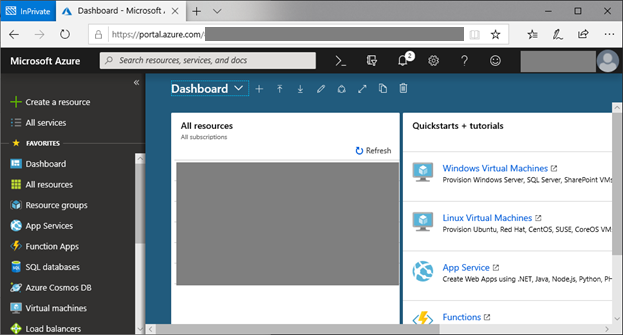
本练习将展示如何创建一个Azure SignalR Service。

**Task 1 – 打开Azure Portal**

1. 打开浏览器，打开Azure： <https://azure.microsoft.com/en-us/>

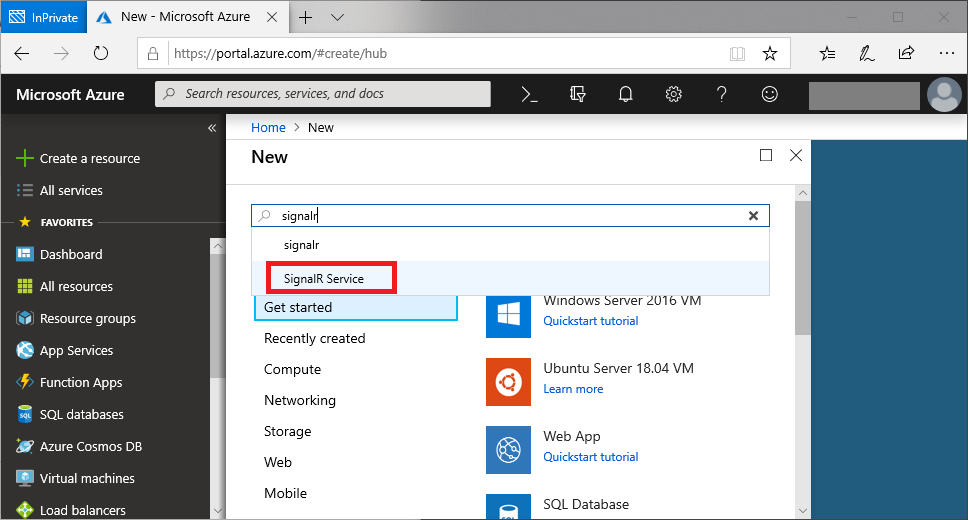


1. 登录或注册免费账户

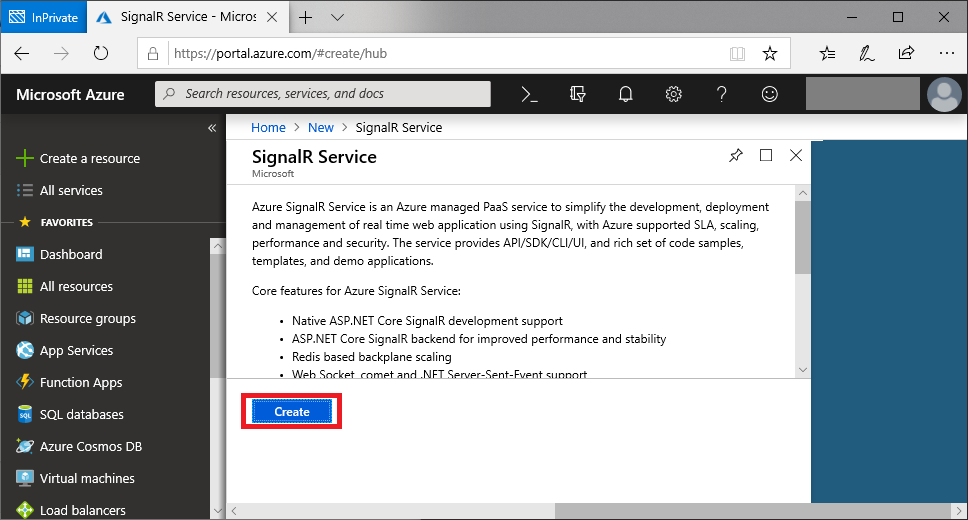


**Task 2 – 创建SignalR Service**

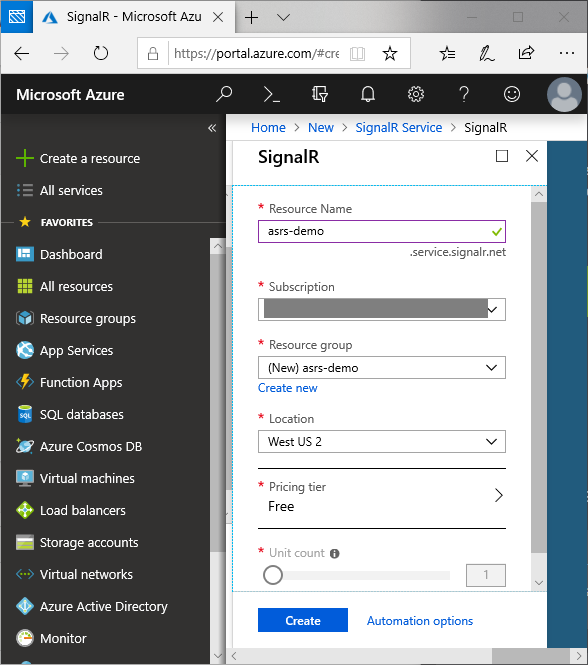
1. 点击创建资源，输入SignalR Service



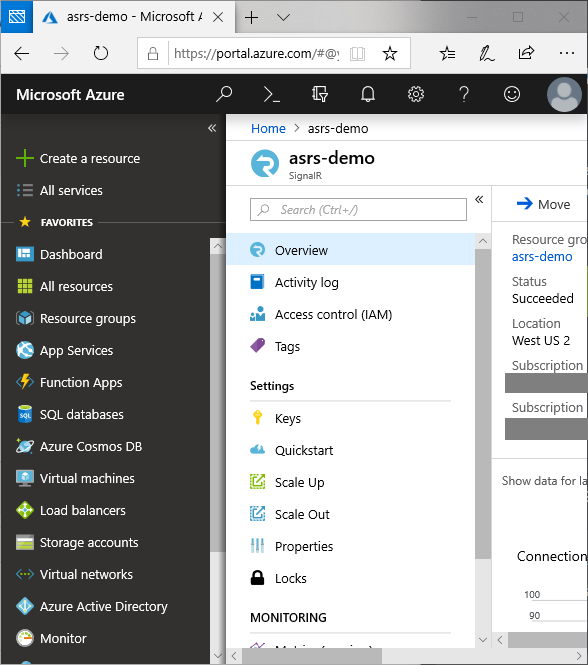
1. 点击创建。



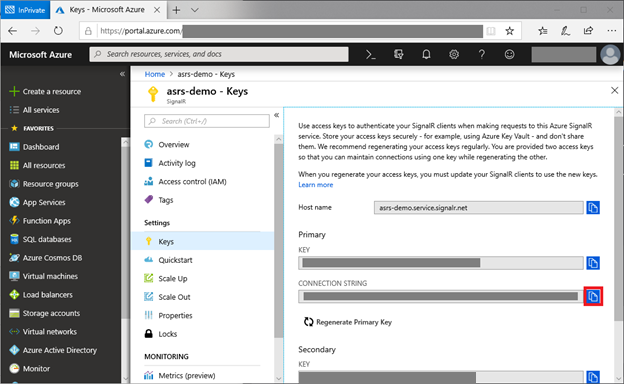
1. 填写服务名称和相关资料，Price Tier选择Free，点击Create



1. 等待创建完成。



1. 打开Key标签，复制连接字符串



Exercise 3: 使用Azure SignalR Service

* 1. 本练习将会把之前的聊天室修改成使用Azure SignalR Service。
  2. **Task 1 – 安装Azure SignalR Service SDK**
  3. 打开命令行窗口，并到达ChatRoom目录。
  4. 使用下列命令安装SDK：

dotnet add package Microsoft.Azure.SignalR --version 1.0.\*

* 1. **Task 2 –编辑源代码文件**
  2. 打开vs code
  3. 修改文件Startup.cs

services.AddSignalR()

.AddAzureSignalR();

app.UseAzureSignalR(routes =>

{

routes.MapHub<Chat>("/chat");

});

* 1. **Task 3 – 修改配置**
  2. 打开命令行
  3. 通过下列命令添加配置：

dotnet user-secrets set Azure:SignalR:ConnectionString "<your connection string>"