

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 ASP.NET Core 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 ASP.NET Core 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

本实验将展示：

- 创建一个 ASP.NET Core SignalR 的聊天室
 - 创建一个 Azure SignalR Service
 - 修改聊天室，使用 Azure SignalR Service
-

System requirements

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)
- .NET Core 2.1 SDK
- VSCode (latest update)
- Git Bash

Setup

安装步骤:

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

Exercises

本次实验包含一下内容:

1. 创建一个 ASP.NET Core SignalR 的聊天室
2. 创建一个 Azure SignalR Service
3. 修改聊天室，使用 Azure SignalR Service

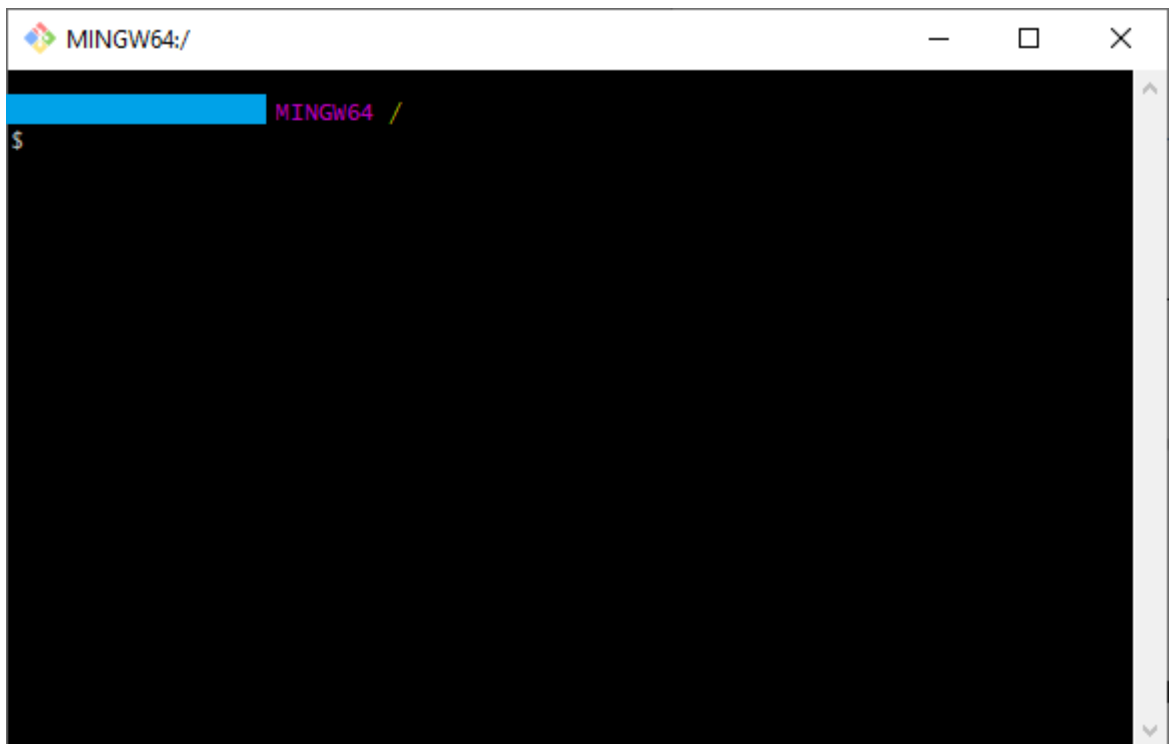
预计完成时间: **30 to 45 minutes.**

Exercise 1: 创建一个 ASP.NET Core SignalR 的聊天室

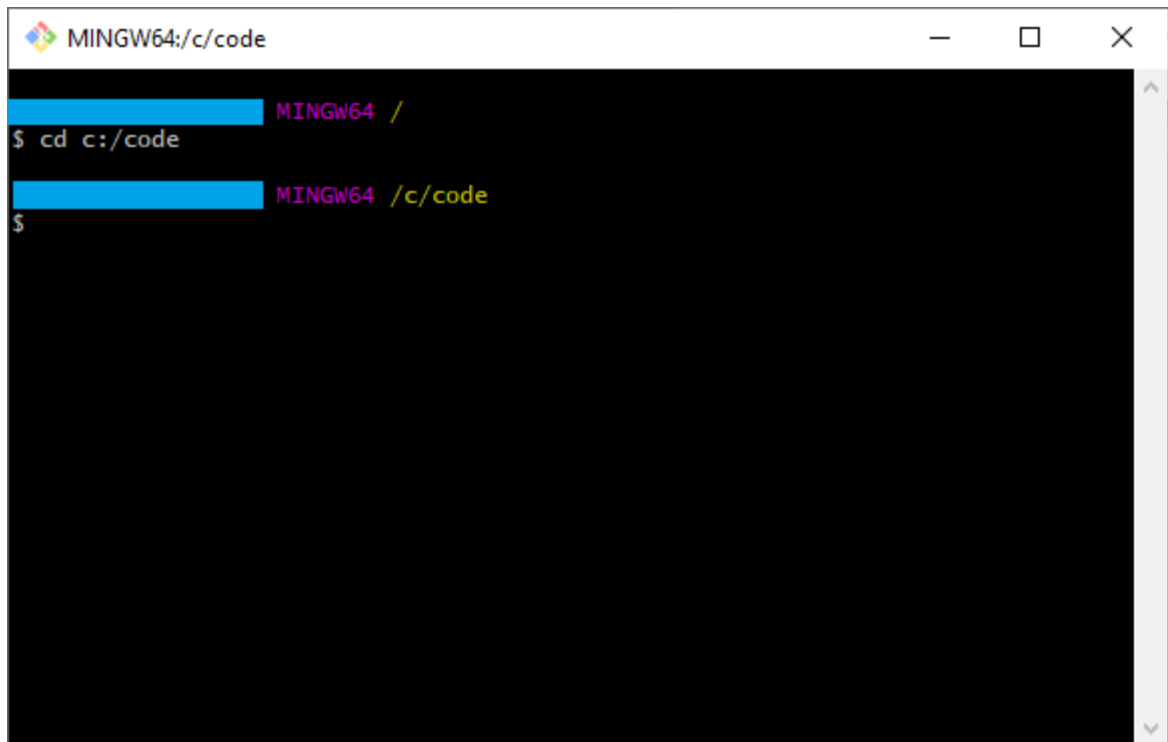
Task 1 – 下载源码

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：



2. 找到一个合适的文件夹：



A screenshot of a MINGW64 terminal window. The title bar shows 'MINGW64:/c/code'. The terminal has a black background with a blue cursor line. The first prompt is '\$ cd c:/code' followed by a new line. The second prompt is '\$' followed by a new line. The prompt text is colored: 'MINGW64' is pink and '/c/code' is yellow.

```
MINGW64 /  
$ cd c:/code  
MINGW64 /c/code  
$
```

3. Clone 源代码：

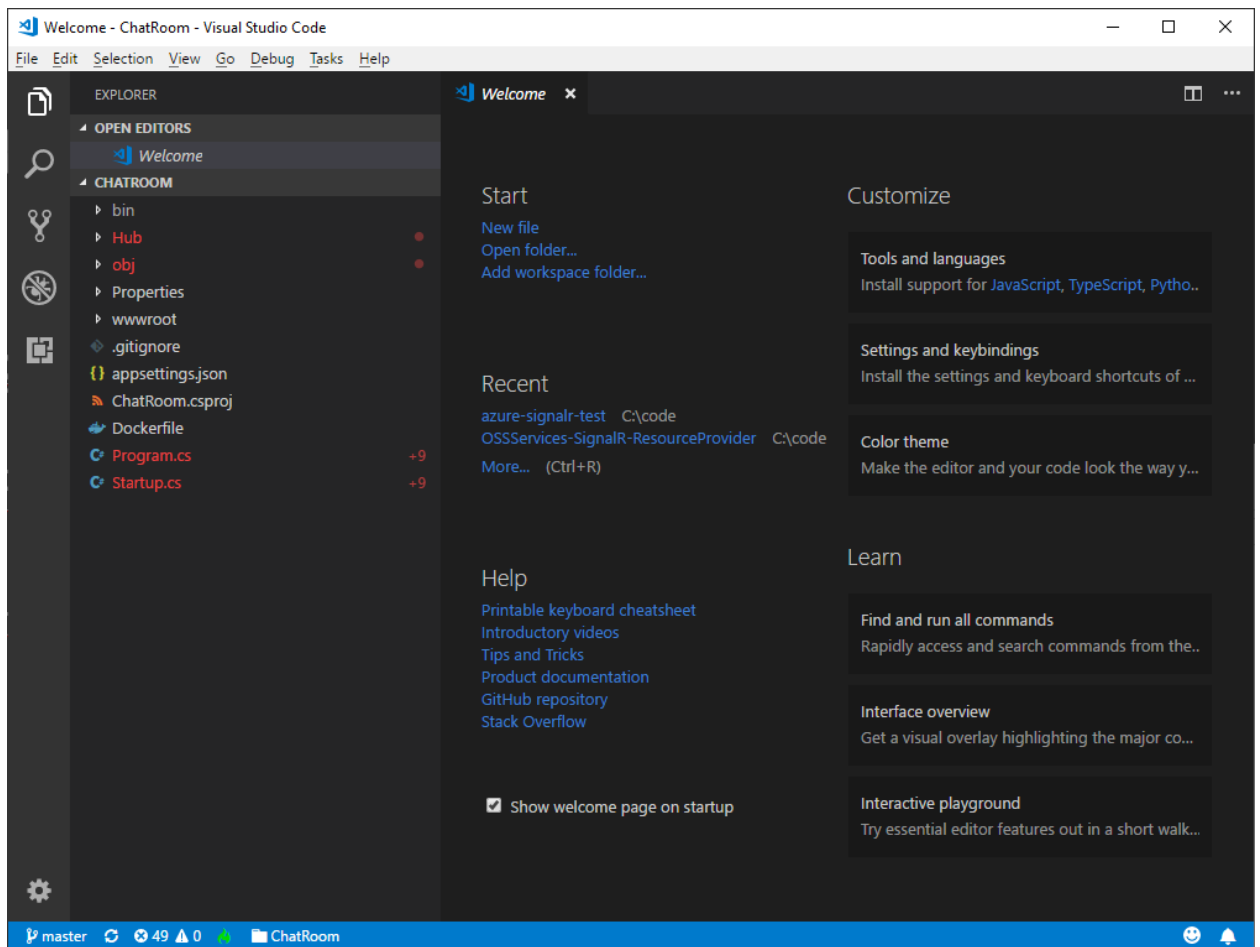
```
git clone https://github.com/vwxyzh/asrs-chatroom-demo
```

Task 2 – 使用 code 打开工程

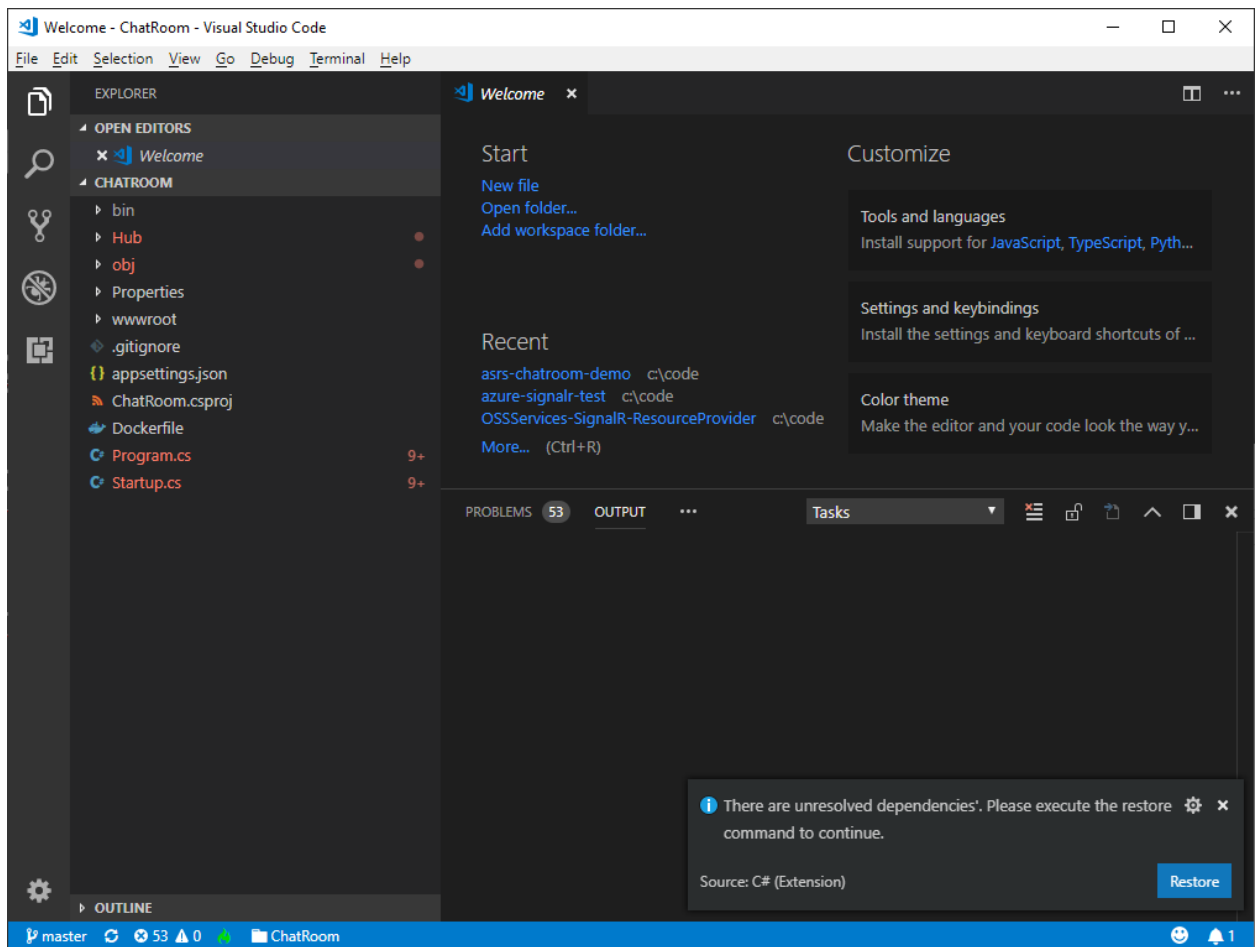
1. 用 vs code 打开 ChatRoom 目录：

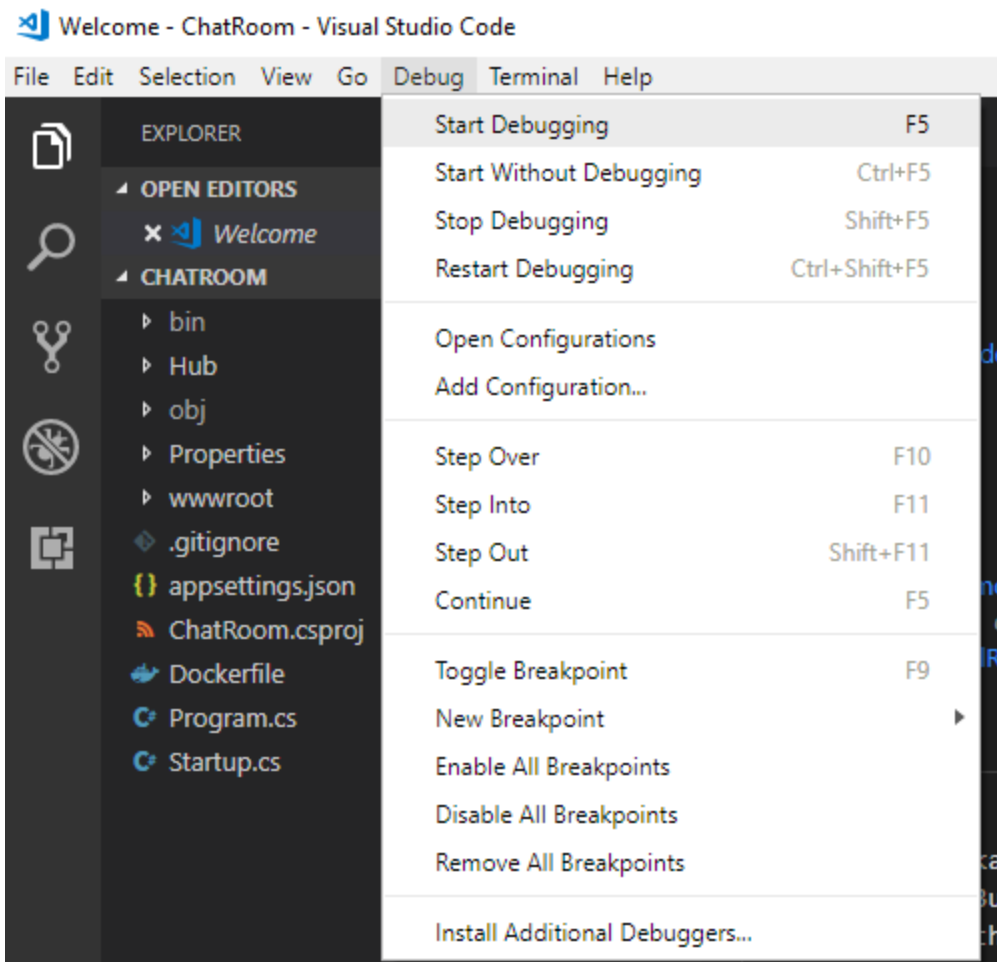
```
cd ChatRoom
```

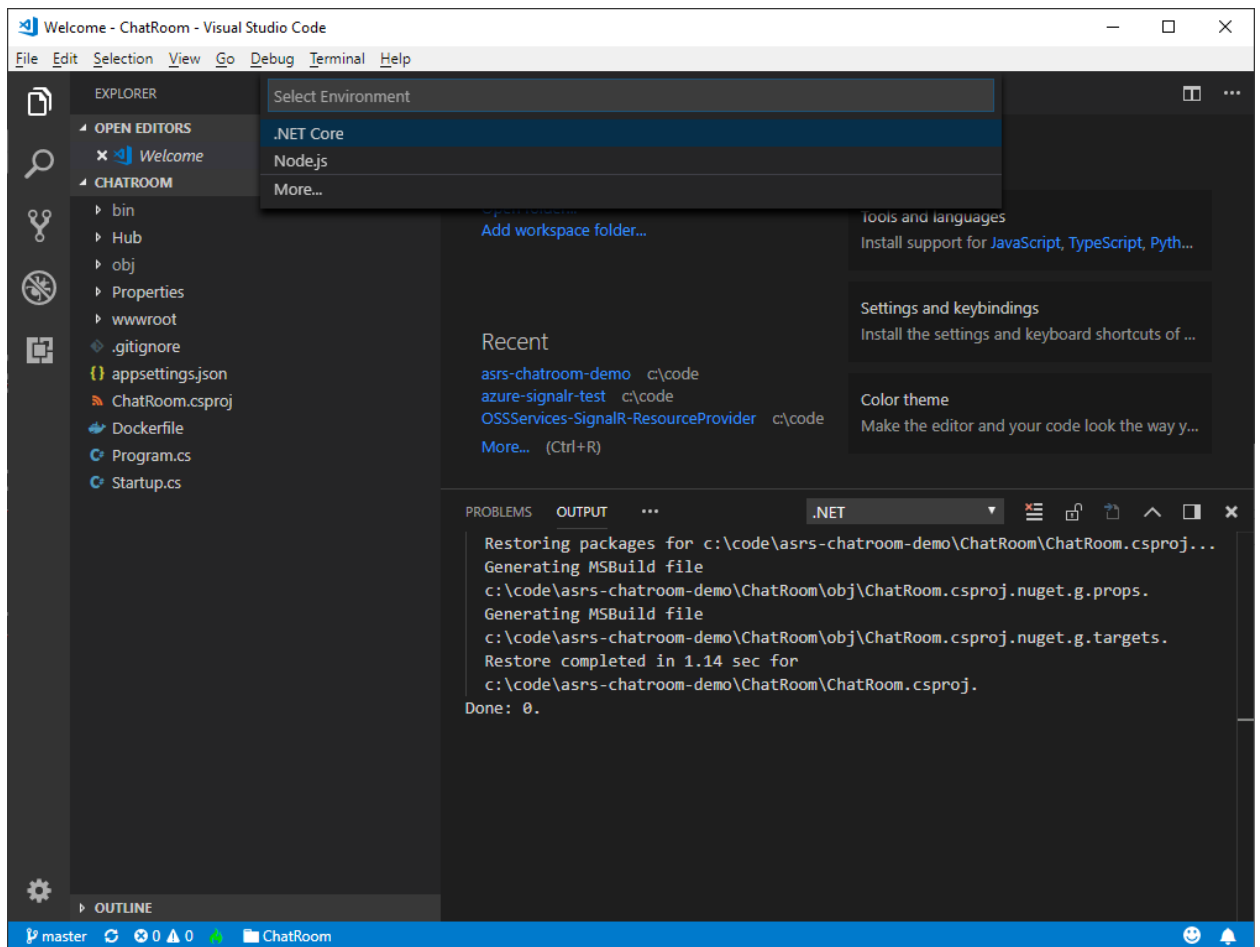
```
code .
```



2. 恢复依赖项和关联 debugger.





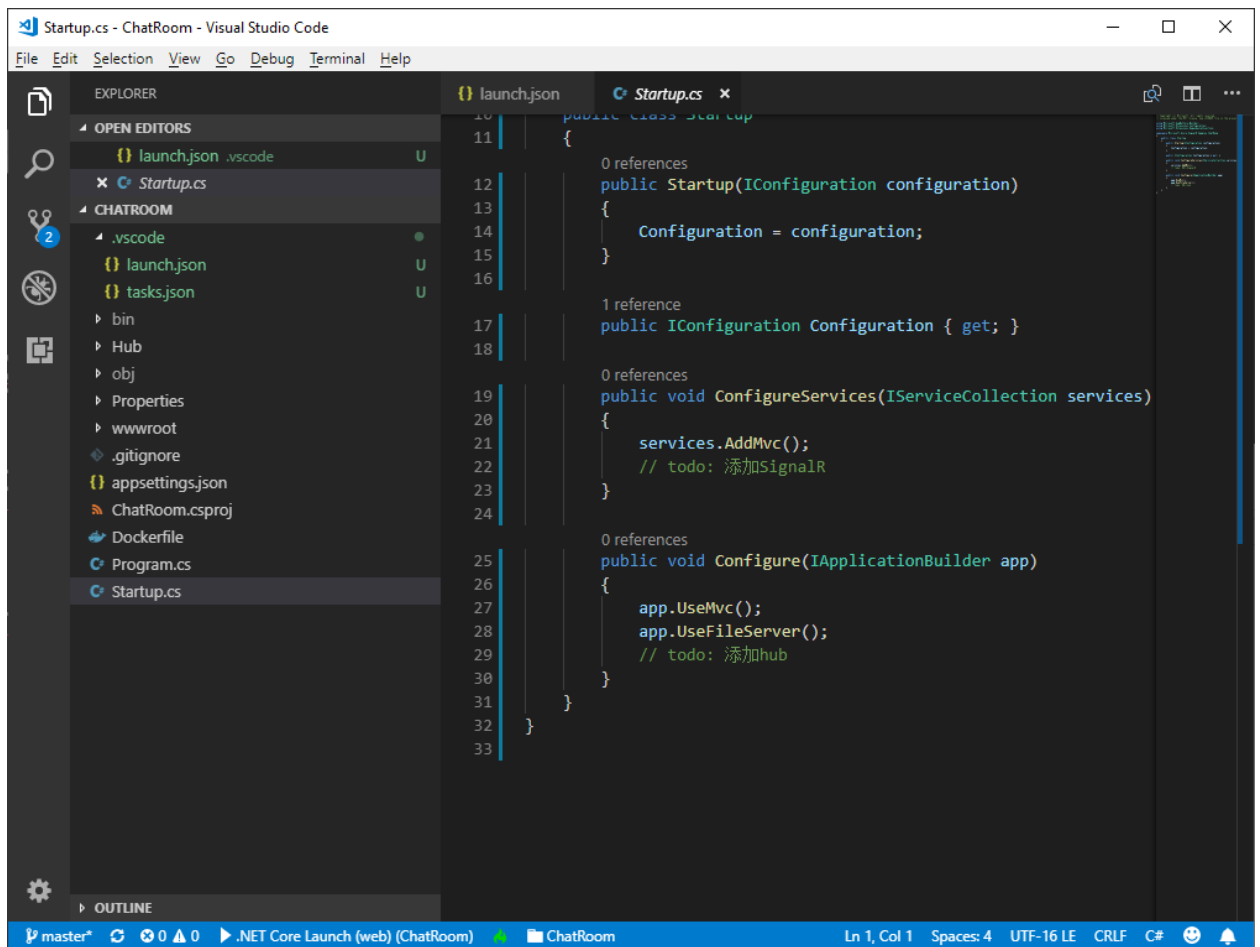


自动添加玩 launch.json 后，再次选择 Start Debugging (F5)

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

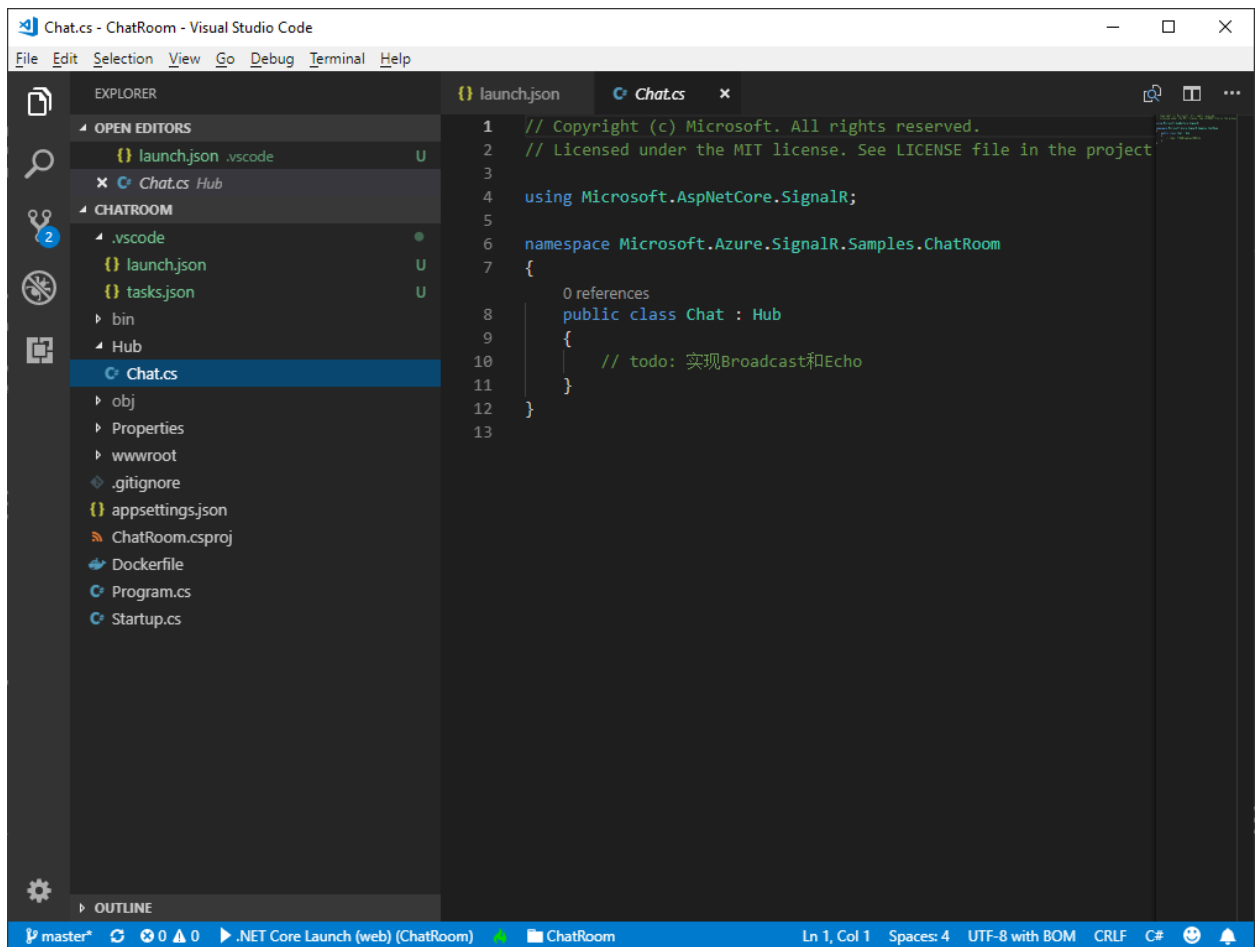
```
app.UseSignalR(routes =>
```

```
{
```

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

```
});
```

2. 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)
```

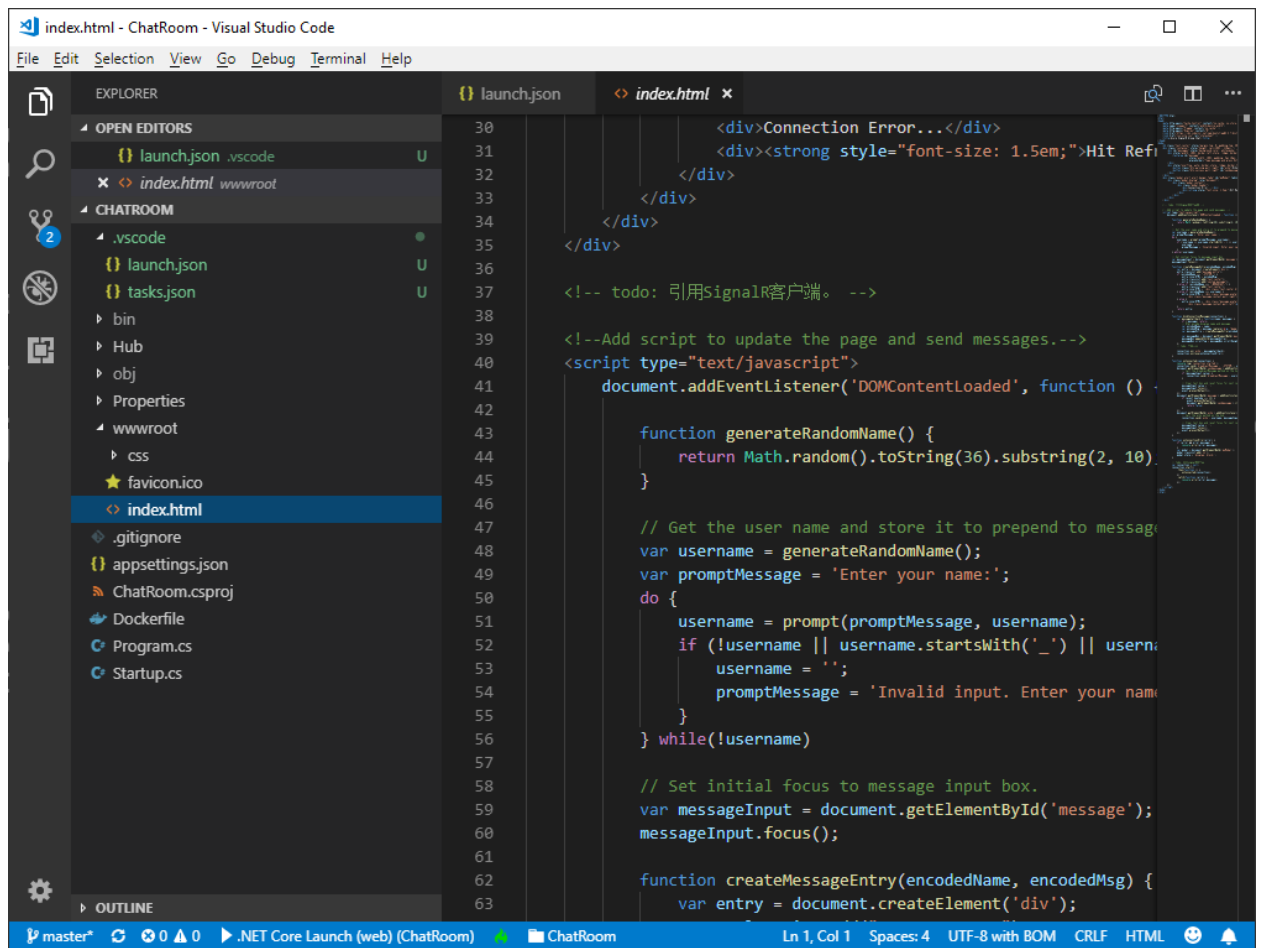
```
{
```

```
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



2. 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"></s
```

```
cript>
```

3. 实现广播：

```
connection.on('broadcastMessage', messageCallback);
```

4. 配置 signalR 客户端：

```
var connection = new signalR.HubConnectionBuilder()
```

```
.withUrl('/chat')
```

```
.build();
```

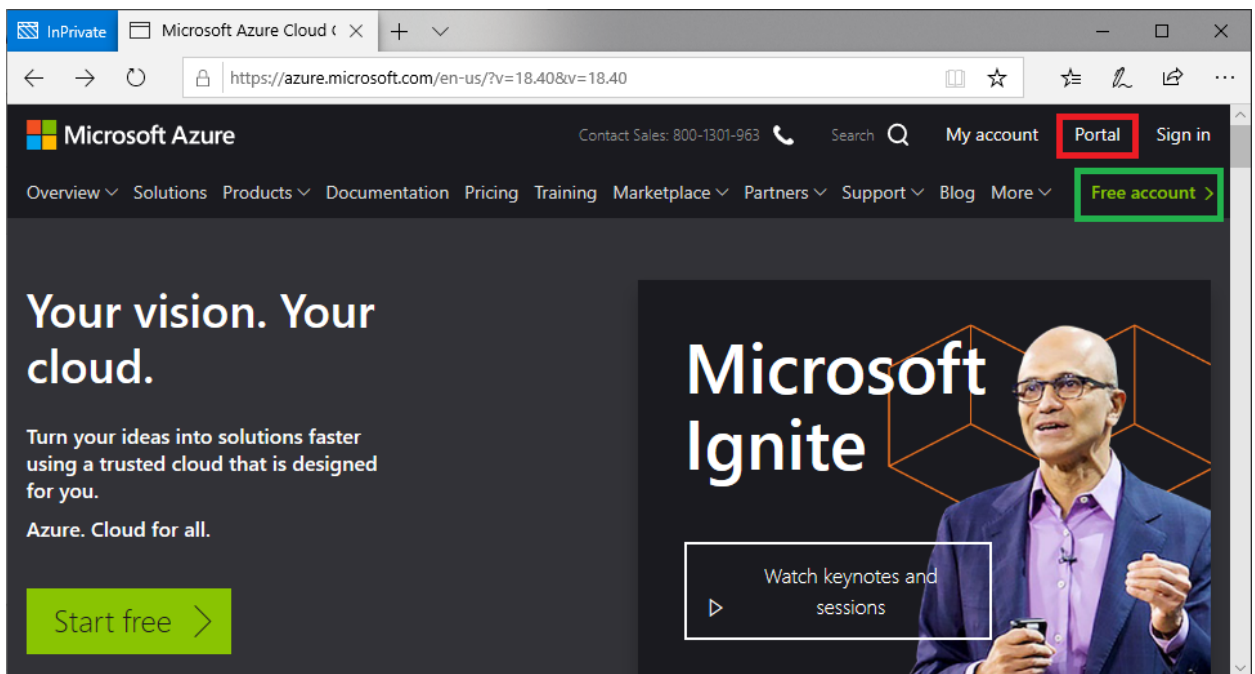
```
bindConnectionMessage(connection);
```

Exercise 2: 创建一个 Azure SignalR Service

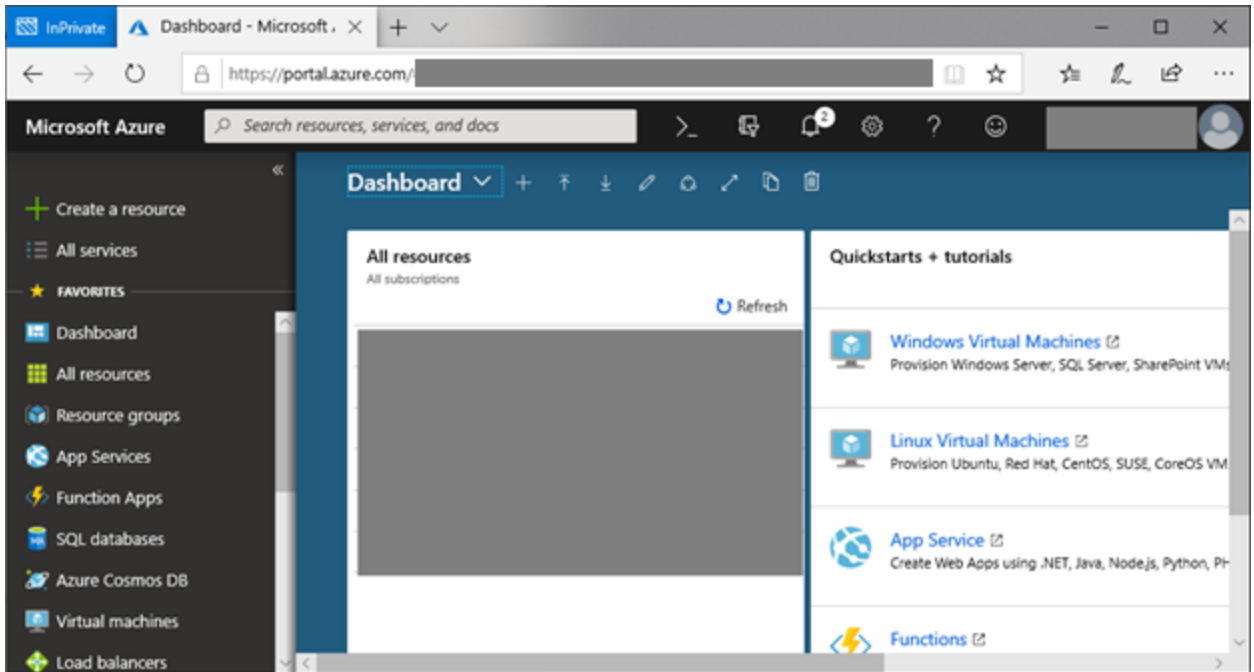
本练习将展示如何创建一个 Azure SignalR Service。

Task 1 – 打开 Azure Portal

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

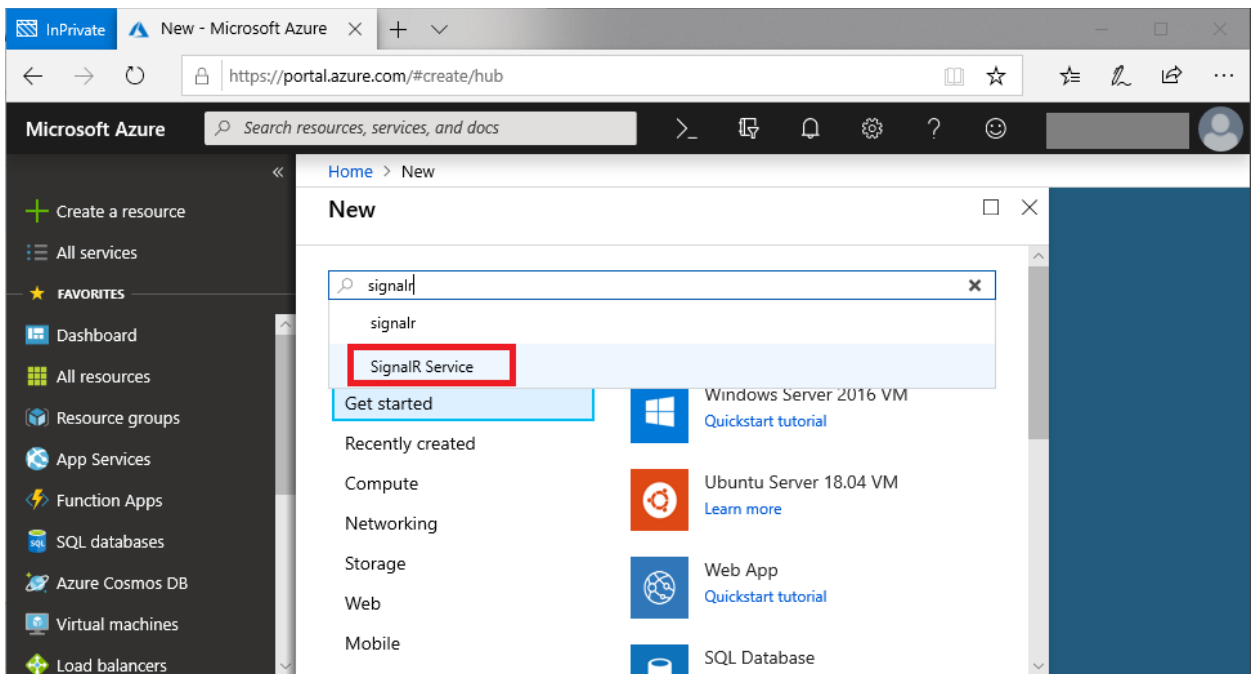


2. 登录或注册免费账户

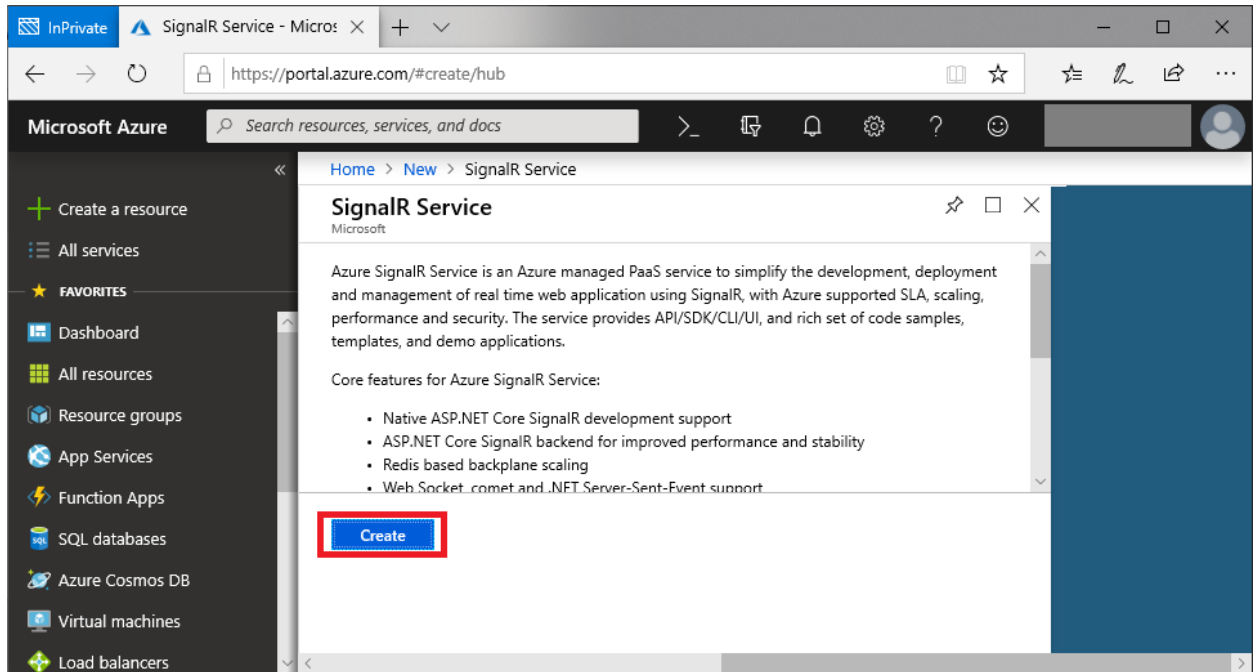


Task 2 – 创建 SignalR Service

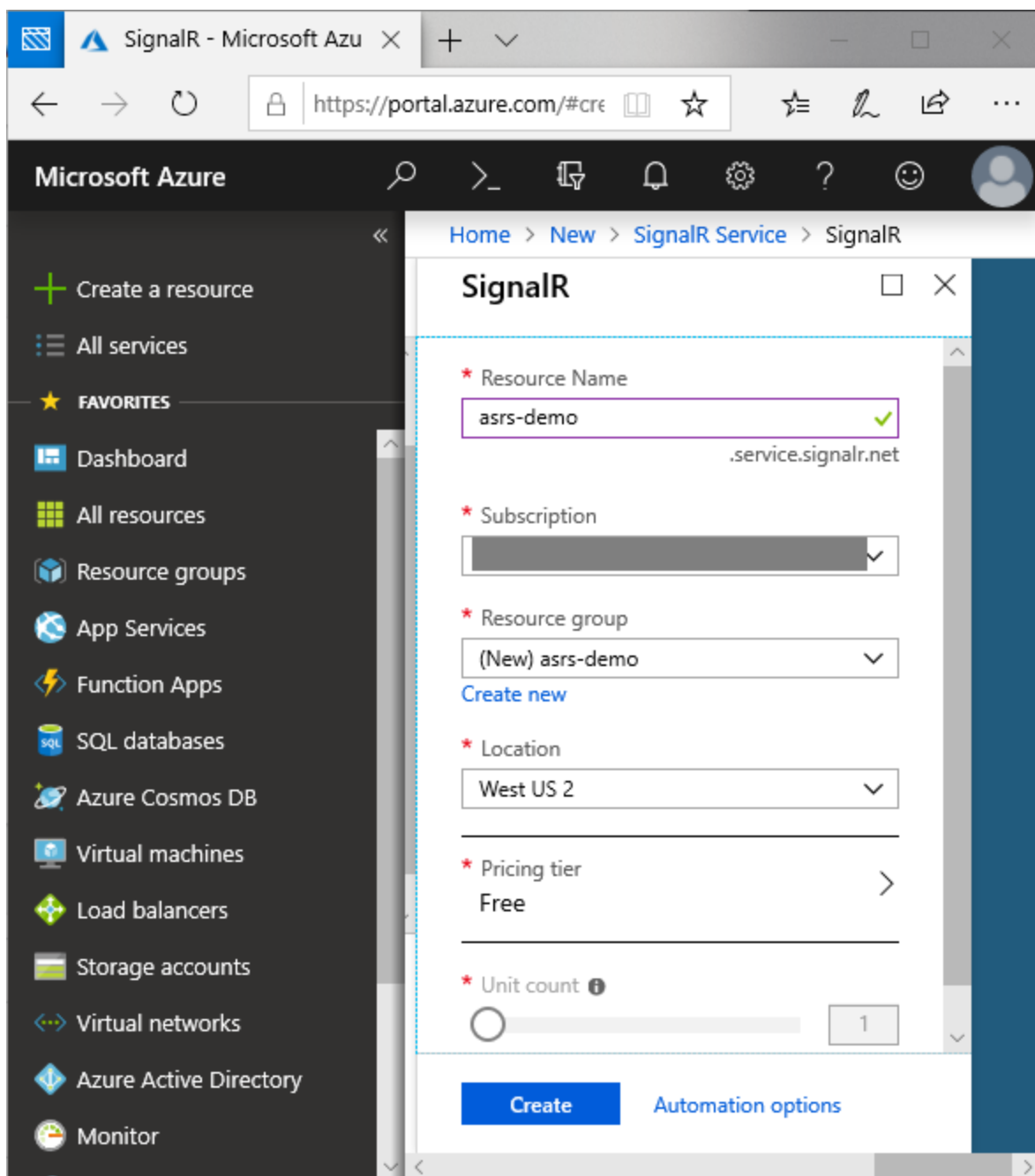
1. 点击创建资源，输入 SignalR Service



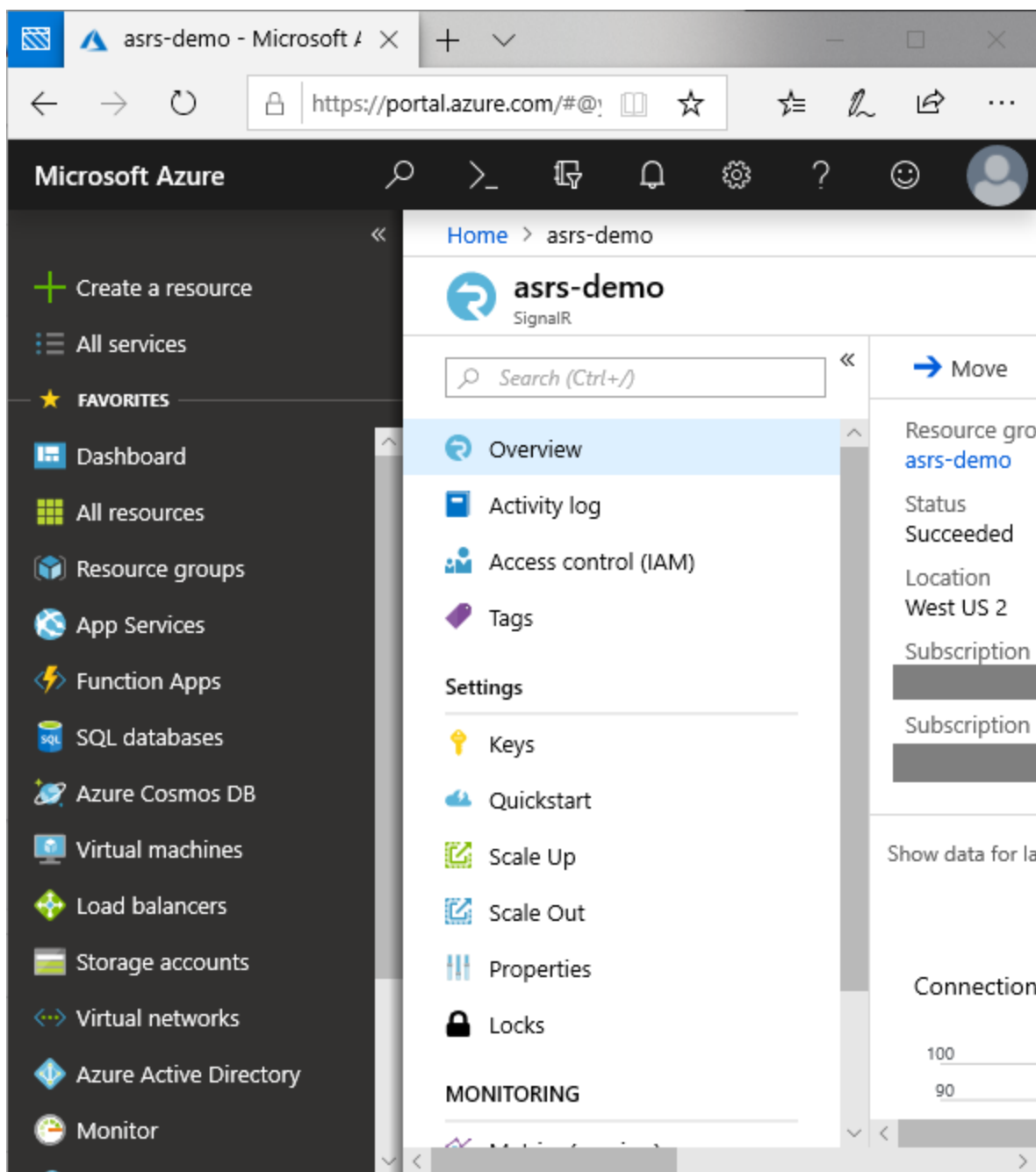
2. 点击创建。



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



4. 等待创建完成。



5. 打开 Key 标签，复制连接字符串

Microsoft Azure portal interface showing the 'Keys' page for a SignalR service named 'asrs-demo'.

The left sidebar contains the 'Microsoft Azure' navigation menu with options like 'Create a resource', 'All services', 'FAVORITES', 'Dashboard', 'All resources', 'Resource groups', 'App Services', 'Function Apps', 'SQL databases', 'Azure Cosmos DB', 'Virtual machines', 'Load balancers', 'Storage accounts', 'Virtual networks', 'Azure Active Directory', and 'Monitor'.

The main content area displays the 'asrs-demo - Keys' page. It includes a search bar, a list of settings (Overview, Activity log, Access control (IAM), Tags, Settings, Keys, Quickstart, Scale Up, Scale Out, Properties, Locks), and a 'MONITORING' section with 'Metrics (preview)'.

The 'Keys' section provides instructions on using access keys to authenticate SignalR clients. It shows the 'Host name' as 'asrs-demo.service.signalr.net'. Under the 'Primary' section, there is a 'KEY' field and a 'CONNECTION STRING' field, both with copy icons. A 'Regenerate Primary Key' button is also present. The 'Secondary' section shows another 'KEY' field with a copy icon.

Exercise 3: 使用 Azure SignalR Service

本练习将会把之前的聊天室修改成使用 Azure SignalR Service。

Task 1 – 安装 Azure SignalR Service SDK

1. 打开命令行窗口，并到达 ChatRoom 目录。
2. 使用下列命令安装 SDK：

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

Task 2 – 编辑源代码文件

1. 打开 vs code
2. 修改文件 Startup.cs

```
services.AddSignalR()
```

```
.AddAzureSignalR();
```

```
app.UseAzureSignalR(routes =>
```

```
{
```

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

```
});
```

Task 3 – 修改配置

1. 打开命令行
2. 通过下列命令添加配置：

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