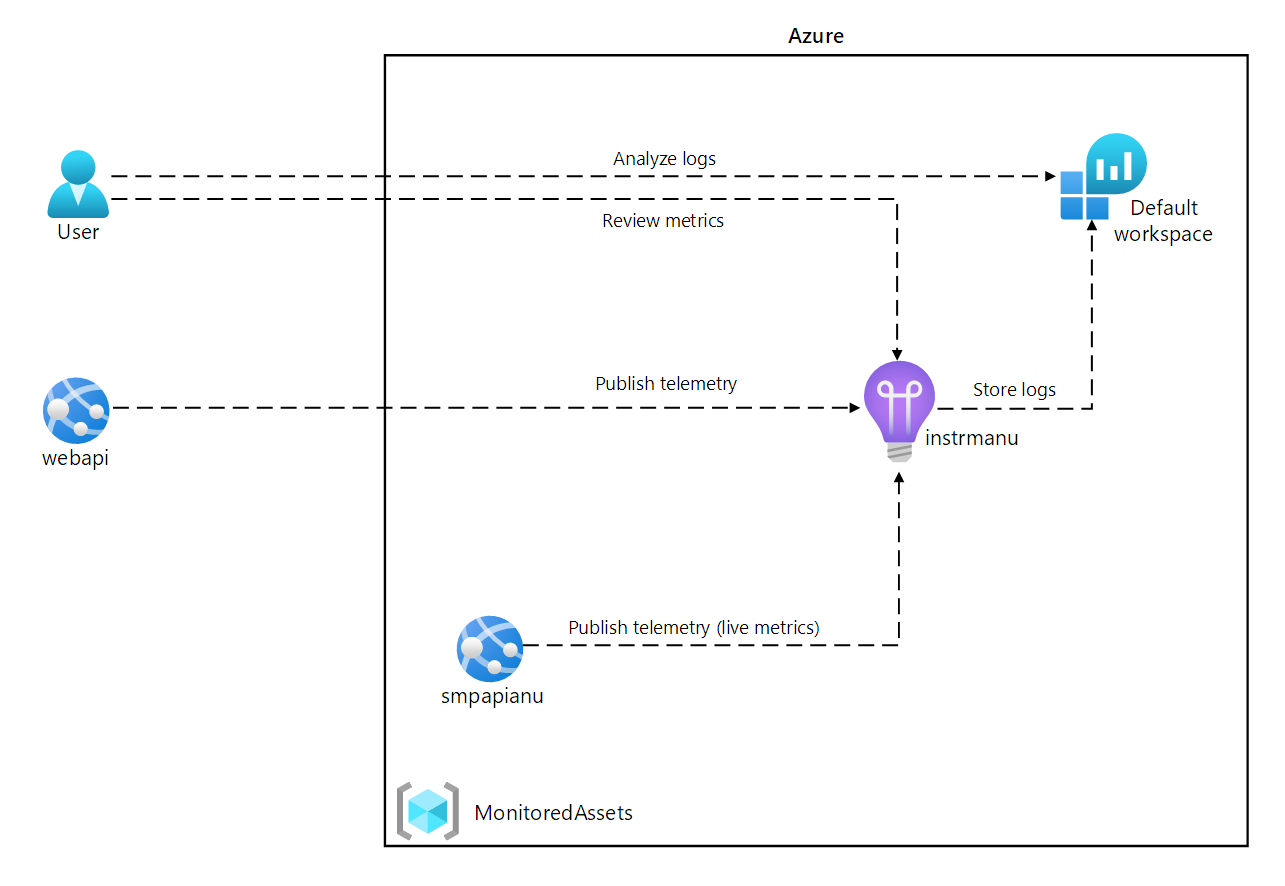
Lab 11: Monitor services that are deployed to Azure

Microsoft Azure user interface

Given the dynamic nature of Microsoft cloud tools, you might experience Azure UI changes that occur after the development of this training content. As a result, the lab instructions and lab steps might not align correctly.

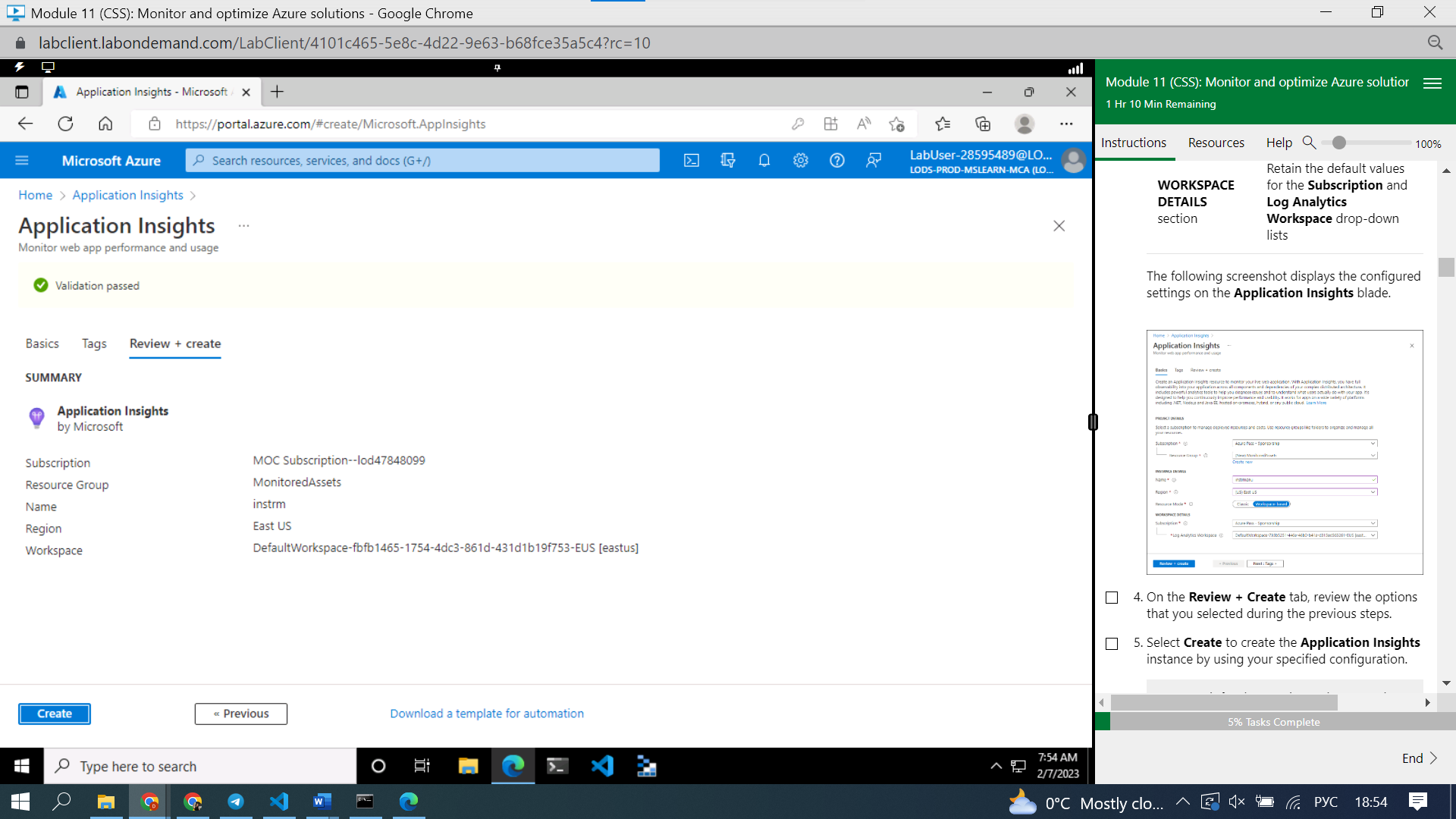
Microsoft updates this training course when the community alerts us to needed changes. However, cloud updates occur frequently, so you might encounter UI changes before this training content updates. **If this occurs, adapt to the changes, and then work through them in the labs as needed.**

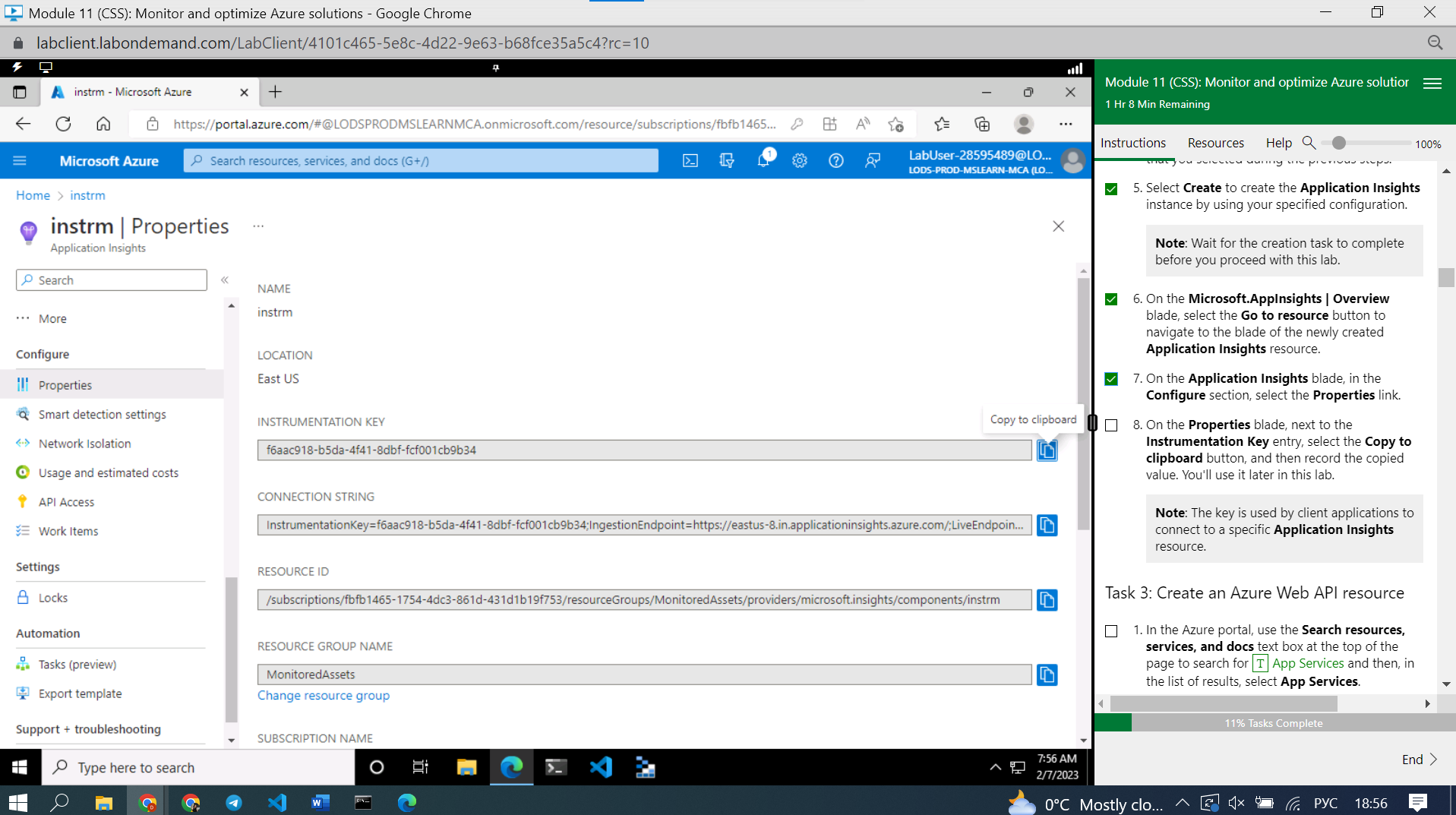


### **Exercise 1: Create and configure Azure resources**

#### **Task 1: Open the Azure portal**

#### **Task 2: Create an Application Insights resource**



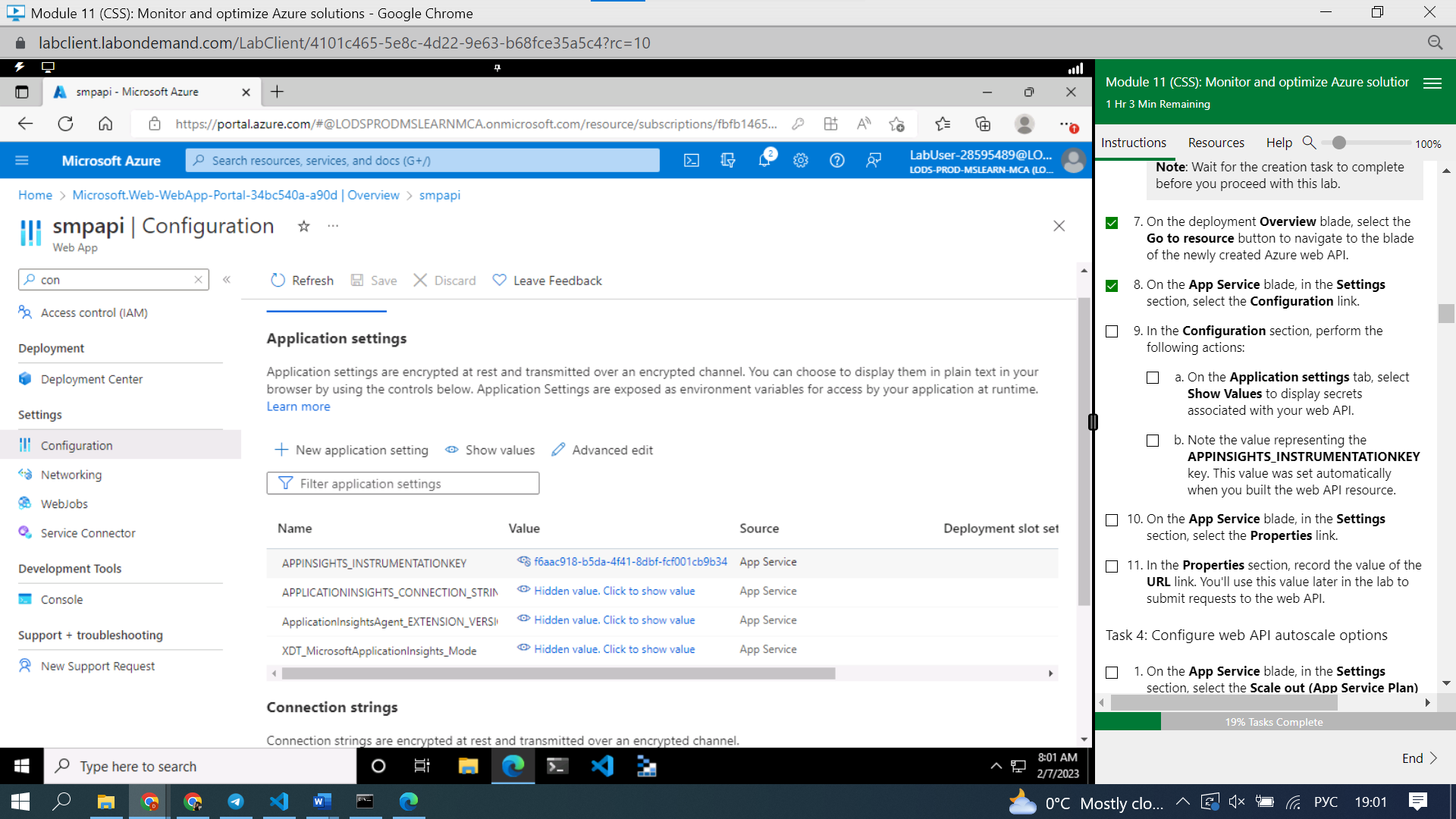


f6aac918-b5da-4f41-8dbf-fcf001cb9b34

**Note**: The key is used by client applications to connect to a specific **Application Insights** resource.

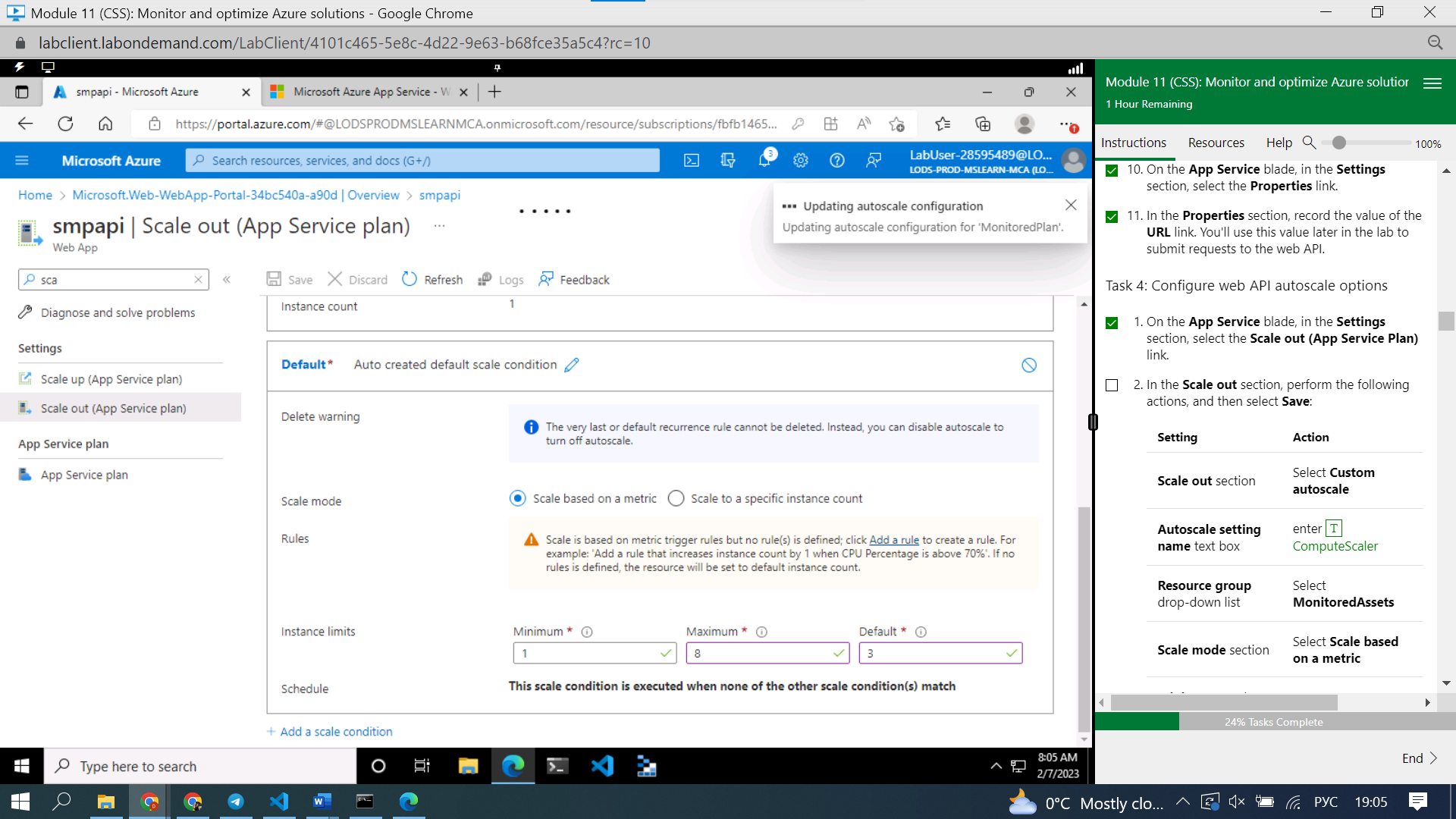
### **Task 3: Create an Azure Web API resource**



ote the value representing the **APPINSIGHTS\_INSTRUMENTATIONKEY** key. This value was set automatically when you built the web API resource.

<a href="https://smpapi.azurewebsites.net/">Microsoft Azure App Service - Welcome (smpapi.azurewebsites.net)</a>

#### **Task 4: Configure web API autoscale options**



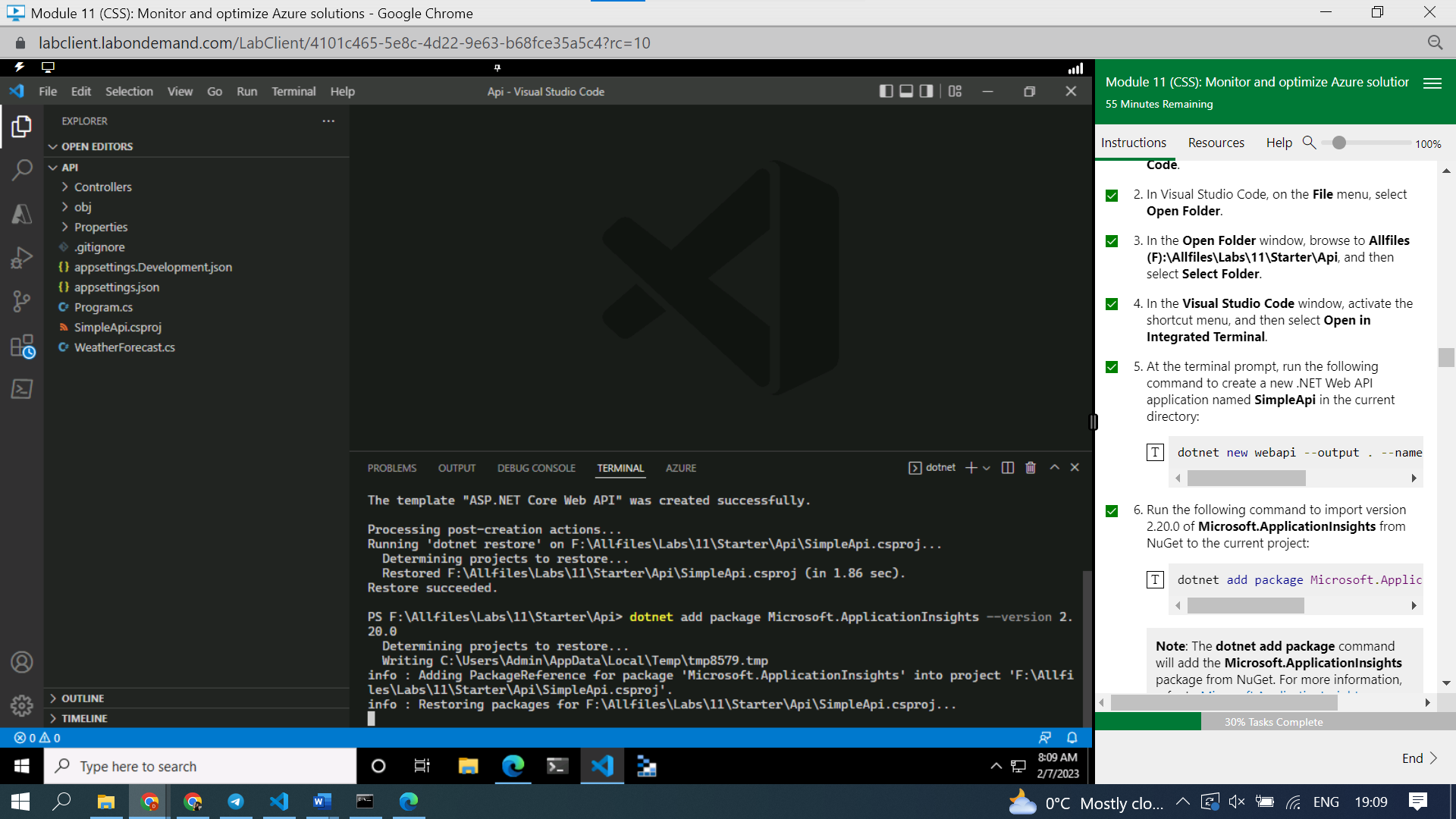
Add rule

#### **Review**

In this exercise, you created the Azure resources that you'll use for the remainder of the lab.

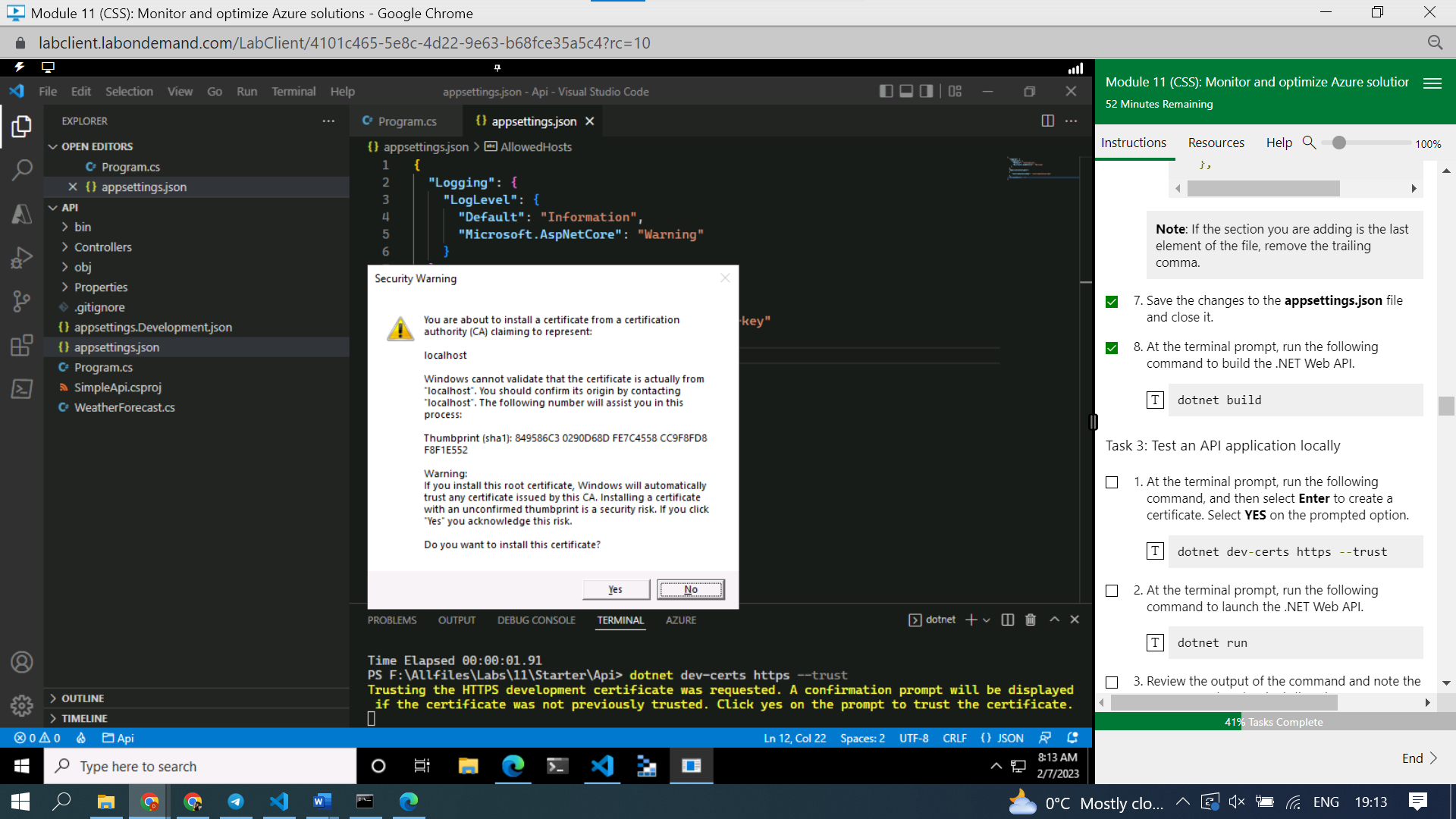
### **Exercise 2: Monitor a local web API by using Application Insights**

#### **Task 1: Build a .NET Web API project**



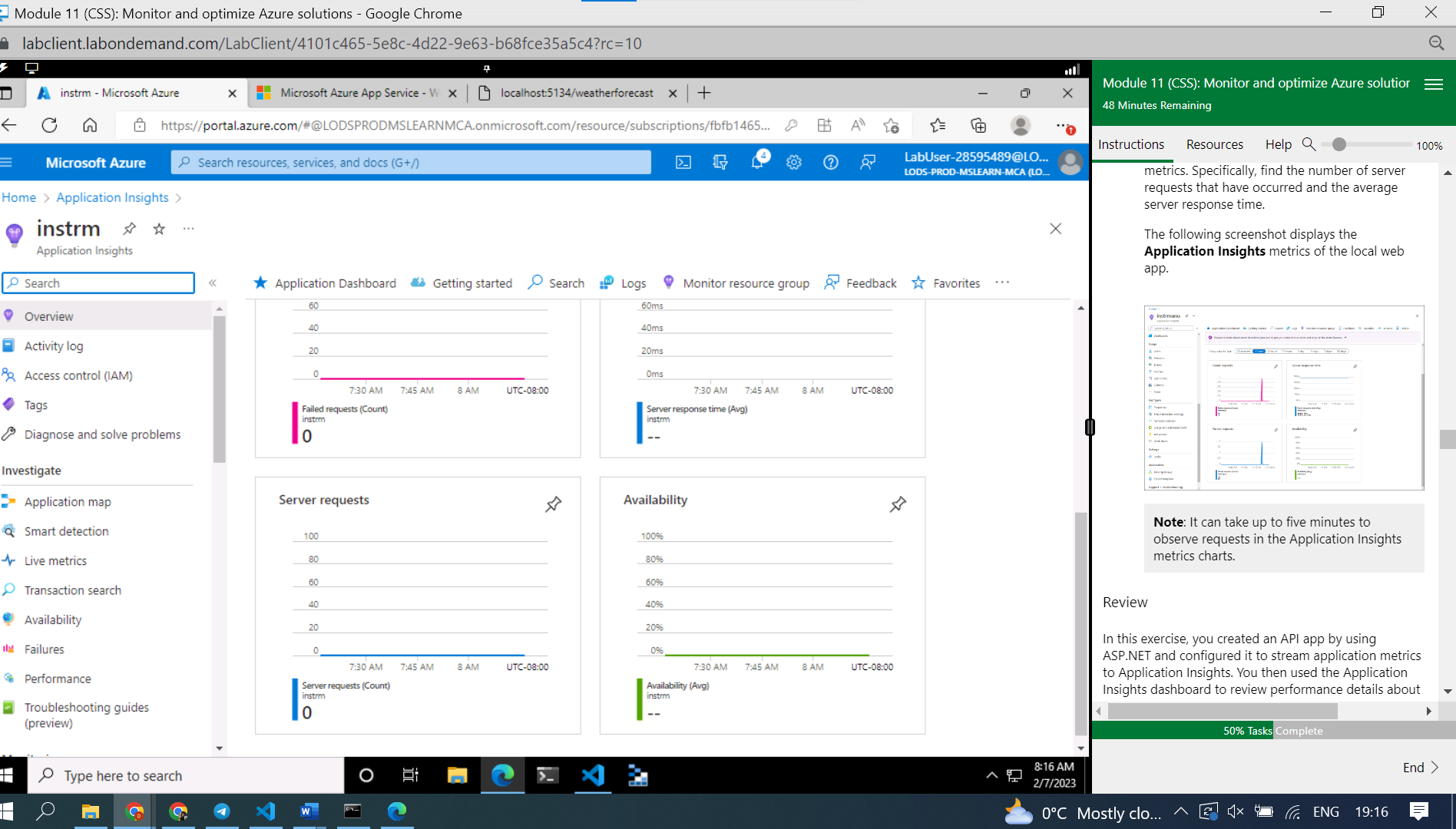
#### **Task 2: Update app code to disable HTTPS and use Application Insights**

#### **Task 3: Test an API application locally**



to create a certificat

#### **Task 4: Review metrics in Application Insights**

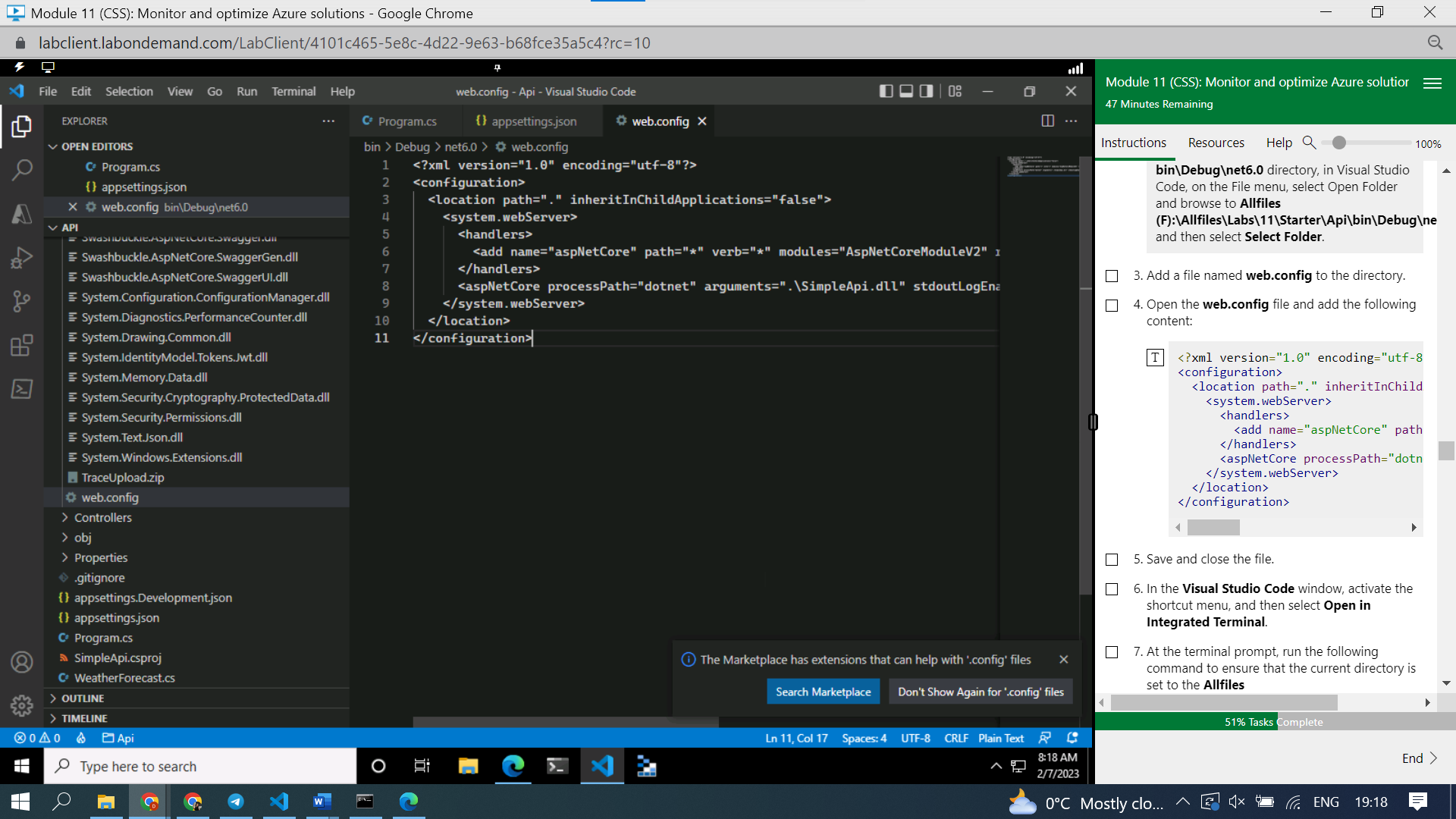


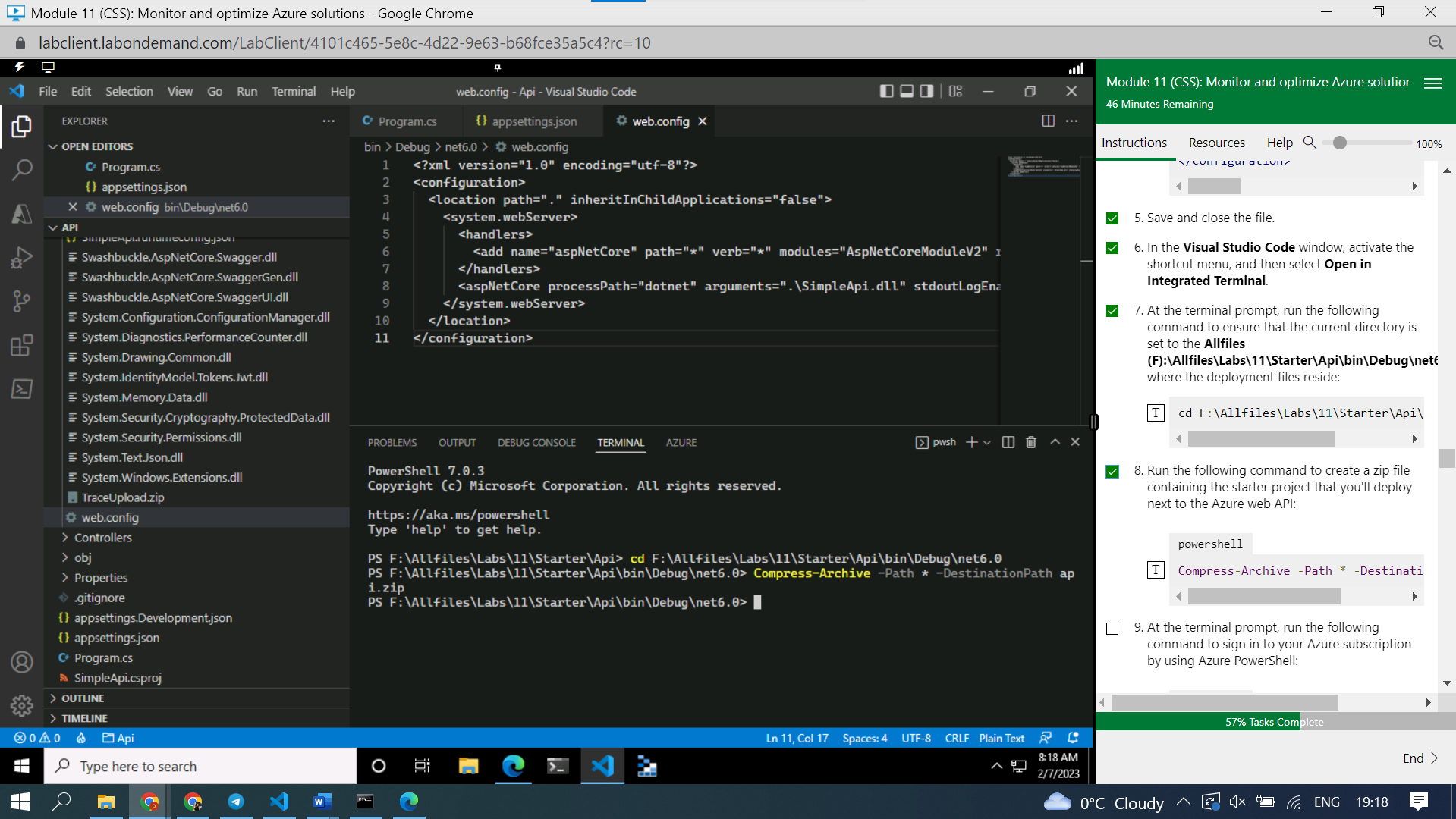
#### **Review**

In this exercise, you created an API app by using ASP.NET and configured it to stream application metrics to Application Insights. You then used the Application Insights dashboard to review performance details about your API.

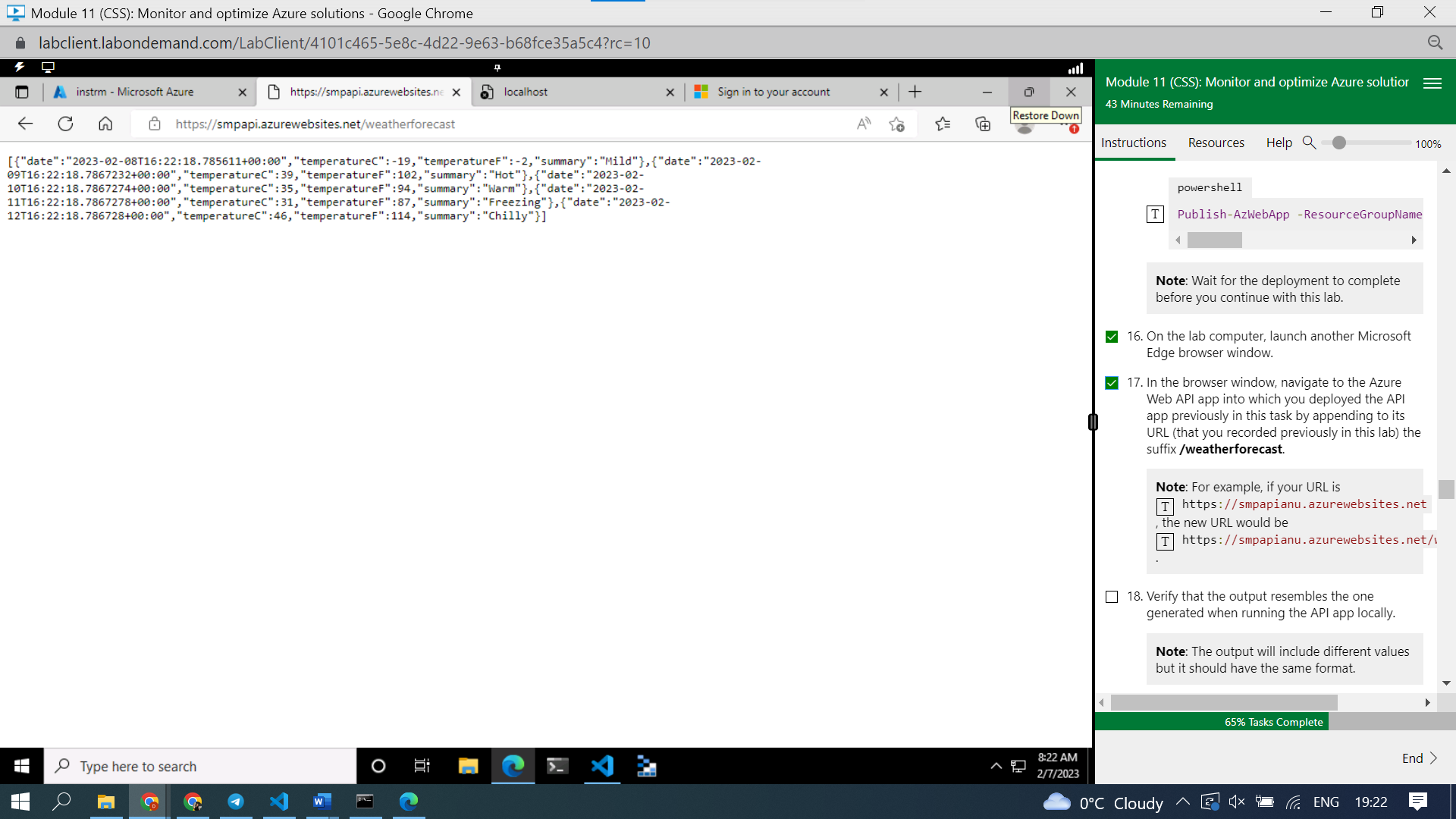
### **Exercise 3: Monitor a web API using Application Insights**

#### **Task 1: Deploy an application to the web API**



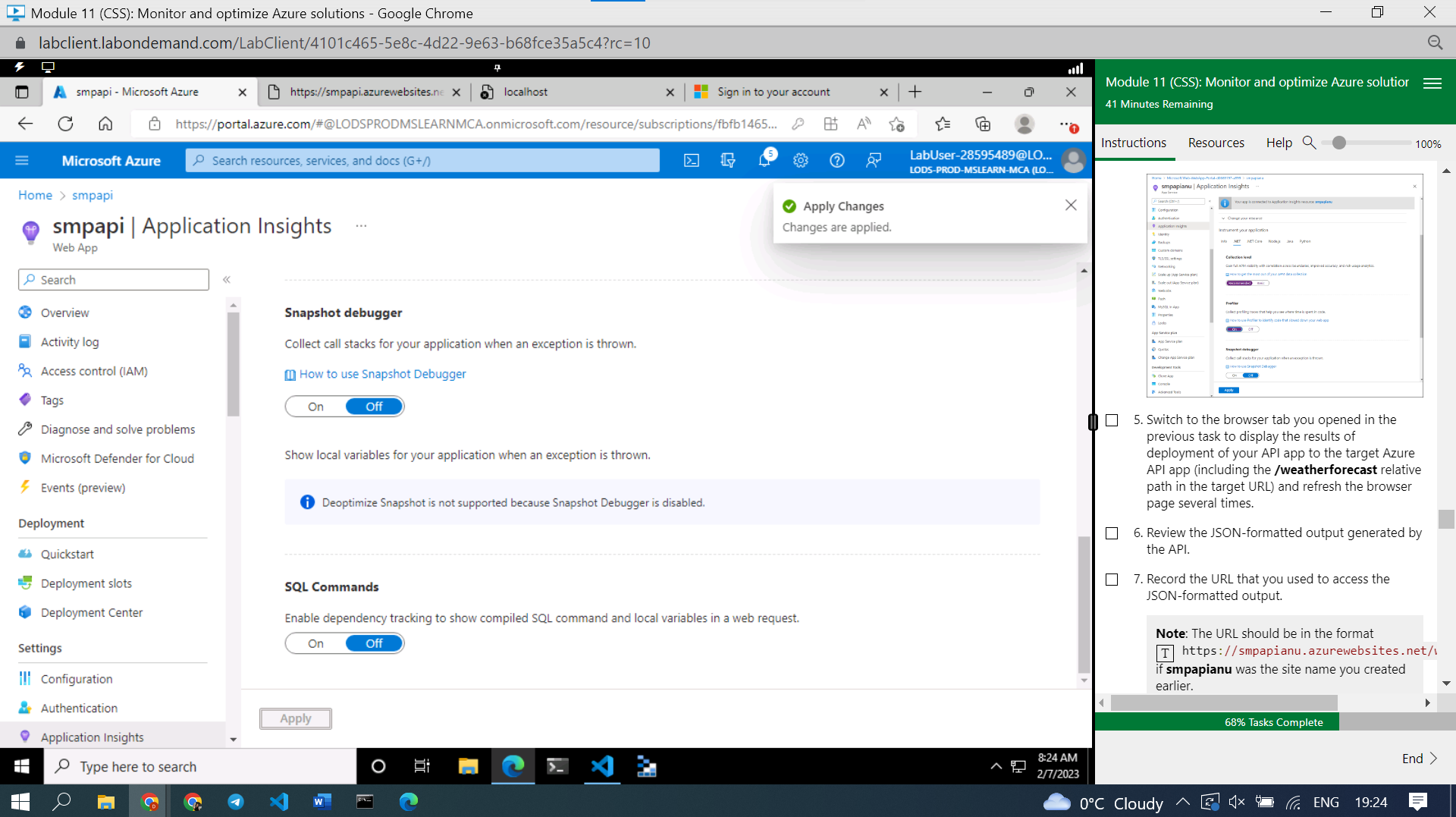


Зип сделали

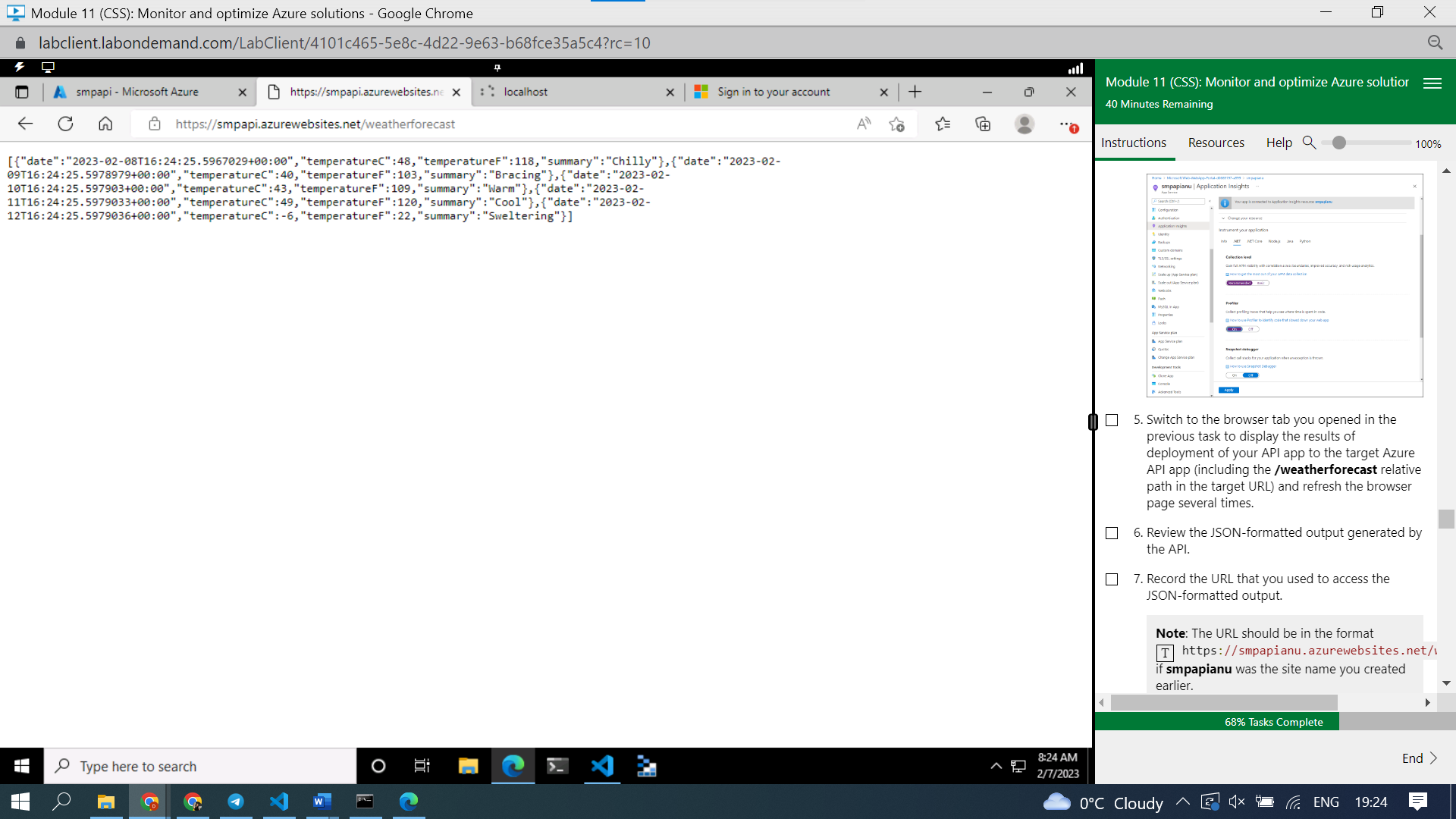


Published

#### **Task 2: Configure in-depth metric collection for Web Apps**

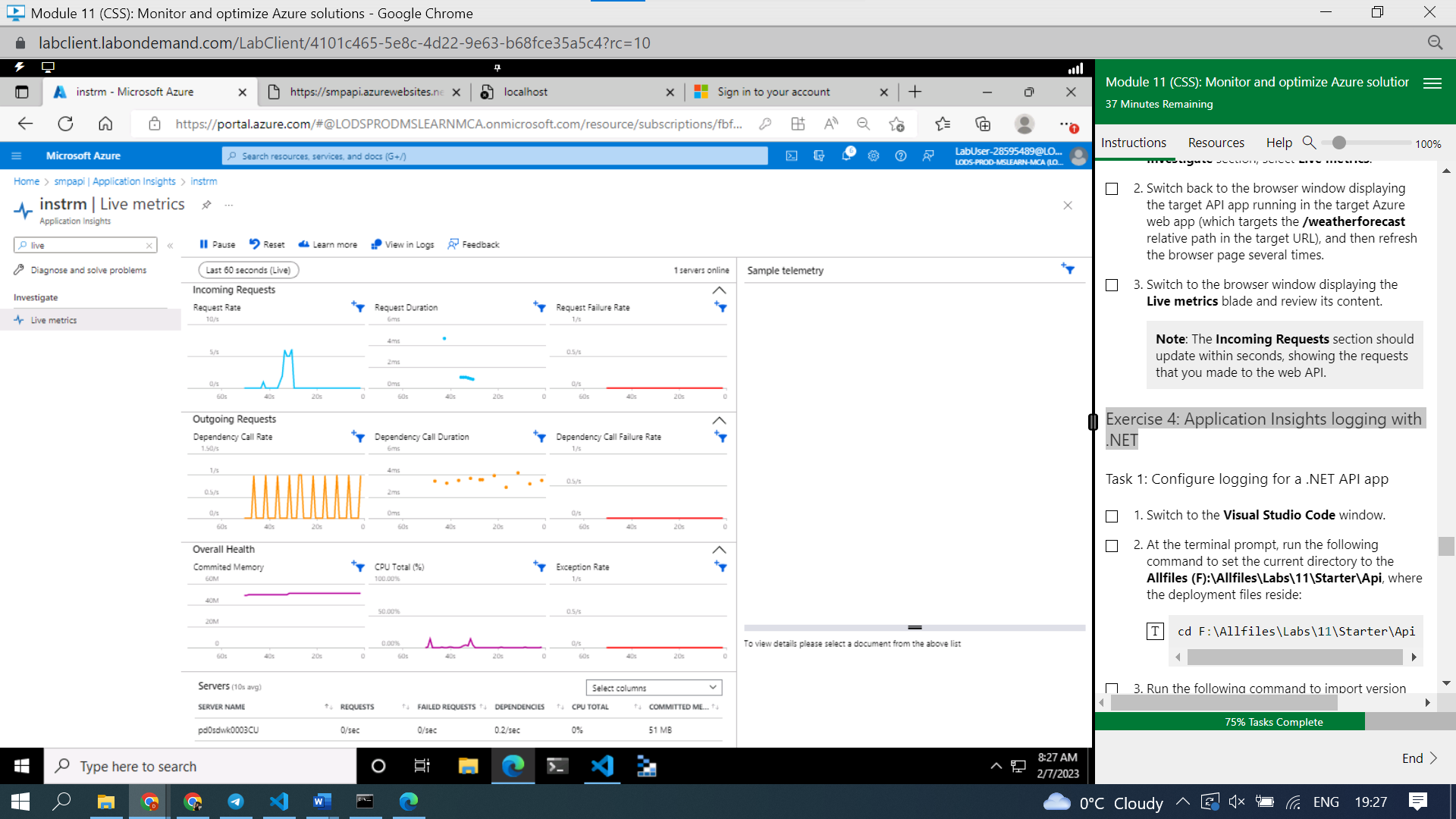


Review the JSON-formatted output generated by the API.

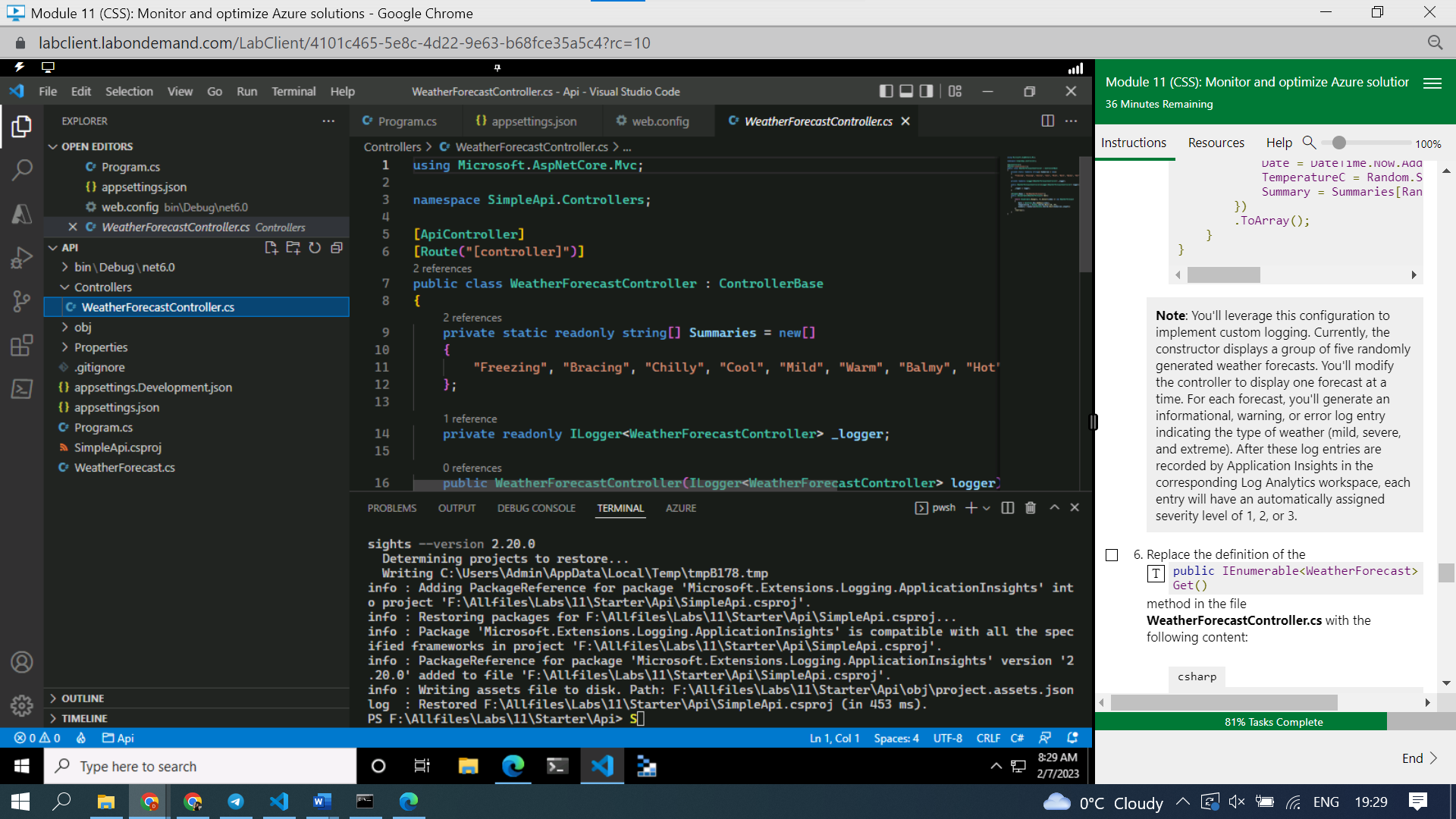


#### **Task 3: Get updated metrics in Application Insights**

#### **Task 4: View real-time metrics in Application Insights**



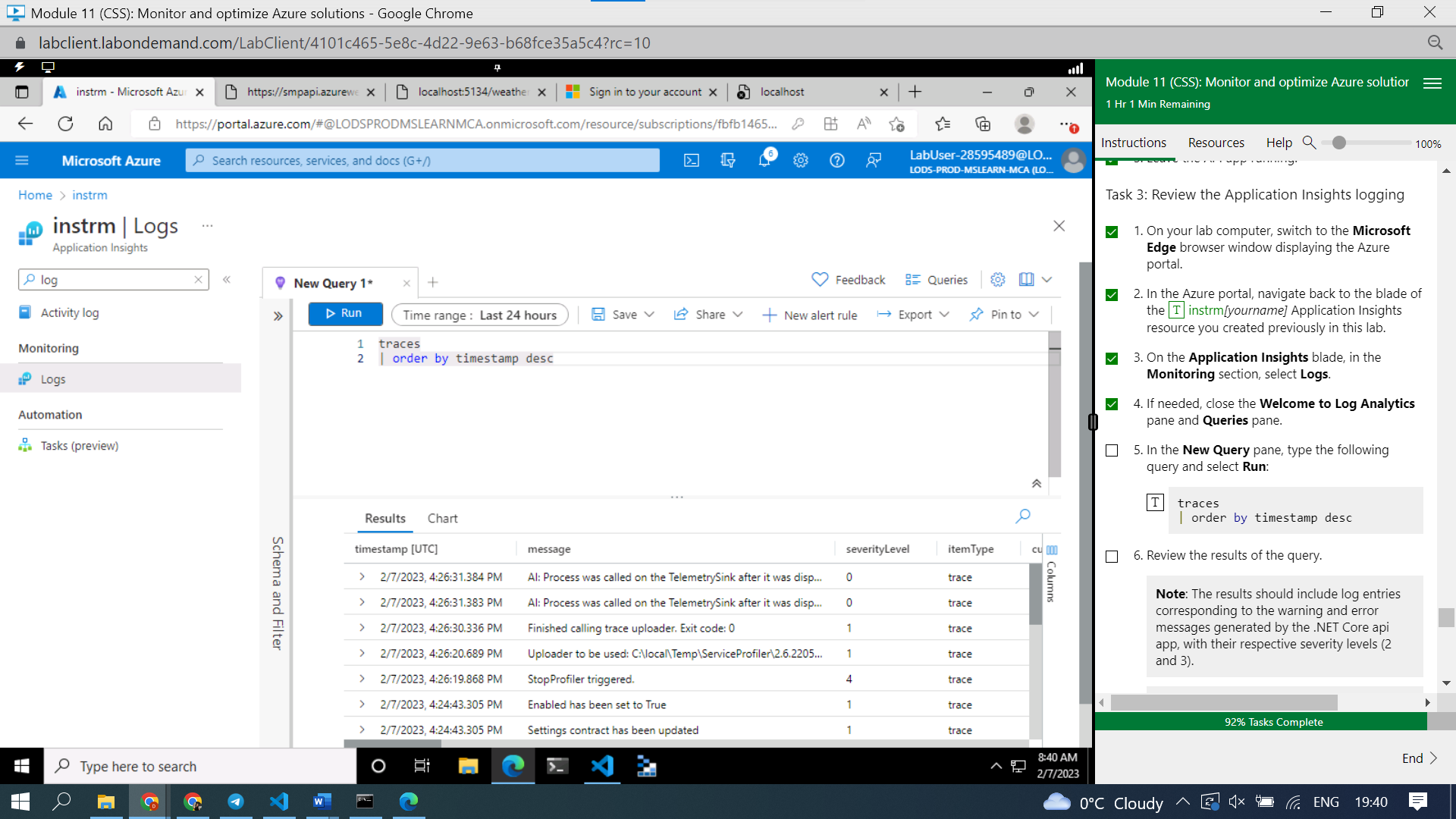
### **Exercise 4: Application Insights logging with .NET**



New

#### **Task 2: Test logging of a .NET Core API app**

#### **Note**: Each refresh of the page should result in an informational, warning, or error message display at the terminal prompt, in the following format: **Task 3: Review the Application Insights logging**



**Note**: You can extend the Application Insights logging level to include informational events by modifying the **appsettings.Development.json** (or **appsettings.json**) file as illustrated in the following code listing. However, you should keep in mind that this will considerably increase the log volume, which has potential network performance and pricing implications:

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft": "Warning",

"Microsoft.Hosting.Lifetime": "Information"

},

"ApplicationInsights": {

"LogLevel": {

"Microsoft": "Information"

}

}

}

}