
# Lab 1

| SUBJECT | TEACHER | DATE |
|---|---|---|
| IoT Hands-On Lab | Bob Familiar | November 19, 2015 |

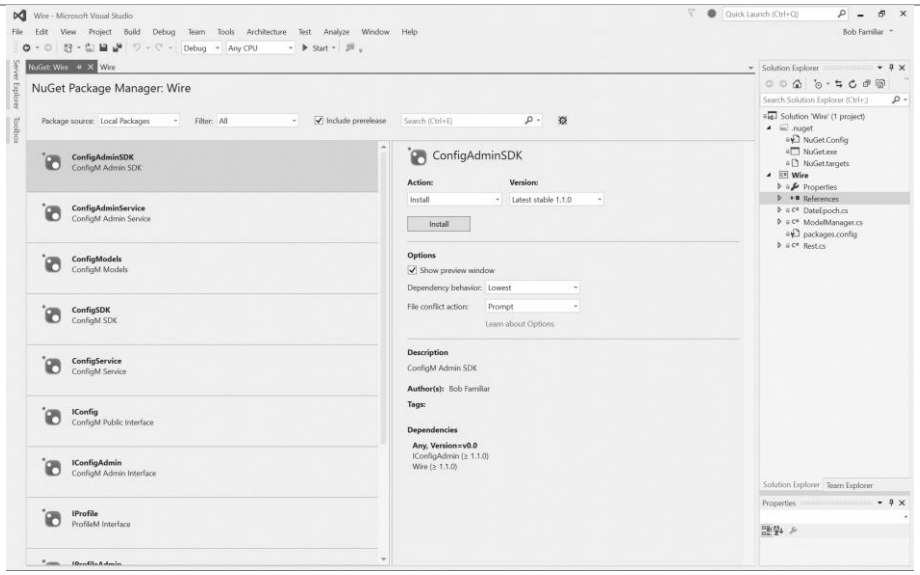
**OVERVIEW**

This lab will have you provisioning and deploying a set of data services, microservices and a real-time dashboard to Azure using a set of PowerShell scripts. The source code and automation scripts can be found on Git Hub at this location: http://bobfamiliar.github.io/microservices-iot-azure/

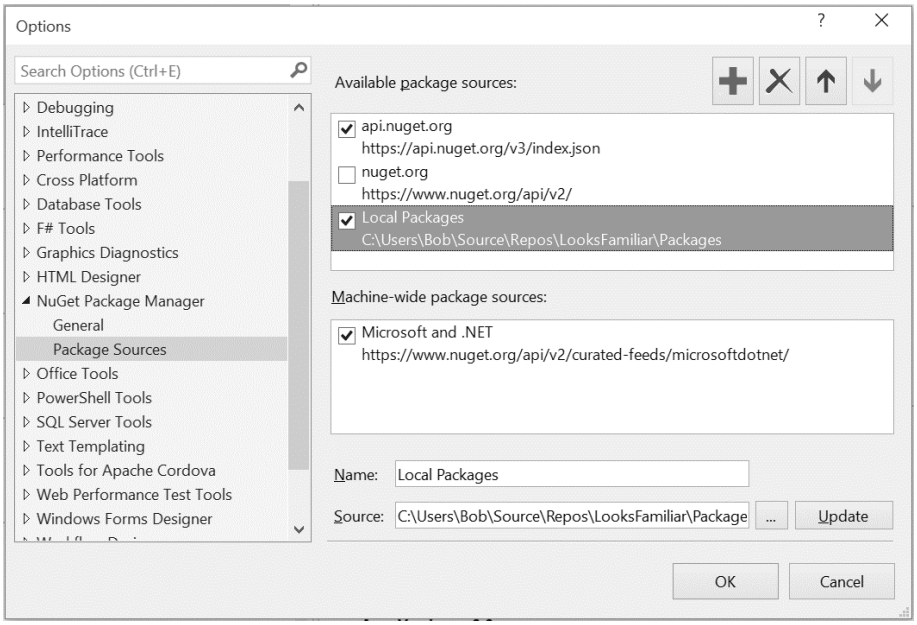
Download this repo as a Zip file and extract into the c:\users\[account]\source\repos folder.

You will be working in a shared Azure subscription. You will be provided azure credentials and a user tag that has the format 'userNNNN' where NNNN is a 4 digit number sequence; for example user0023. This unique tag will be used to create the names of your resources in Azure so that they are unique and easily located. All the resources you create will be organized into a Resource Group called 'HOL_RG_userNNNN' so you can easily list them.

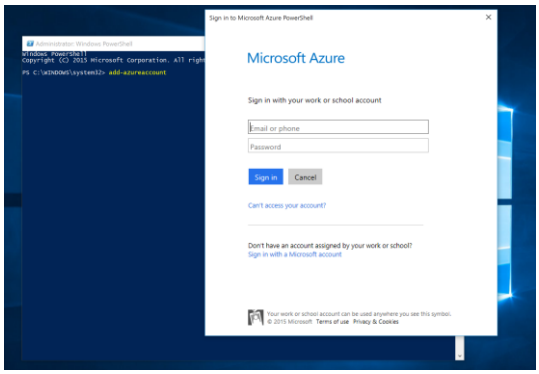
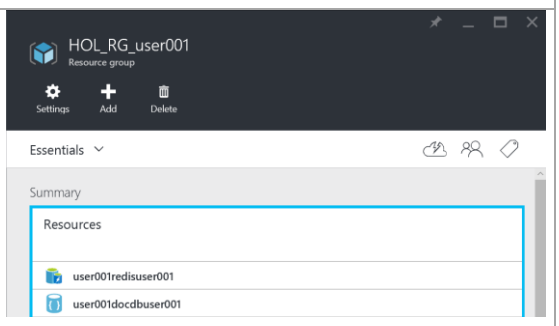
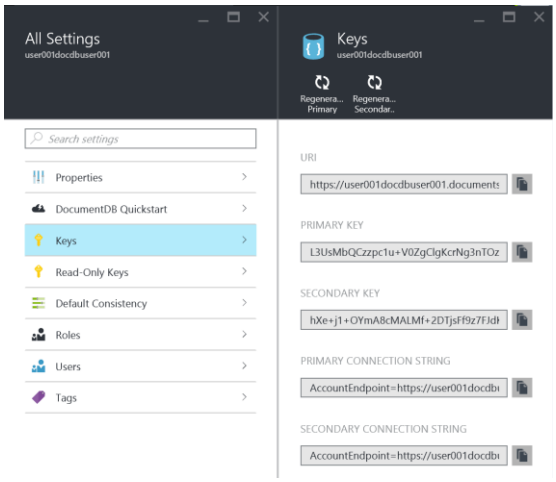
| LAB 1 | |
|---|---|
| STEP 1 | • Update the Visual Studio NuGet Manager settings by adding an additional package location<br>• Follow this menu path in Visual Studio: Tools Menu > NuGet Package Manager > Manage NuGet Packages for this solution. The NuGet Package Manager will display |



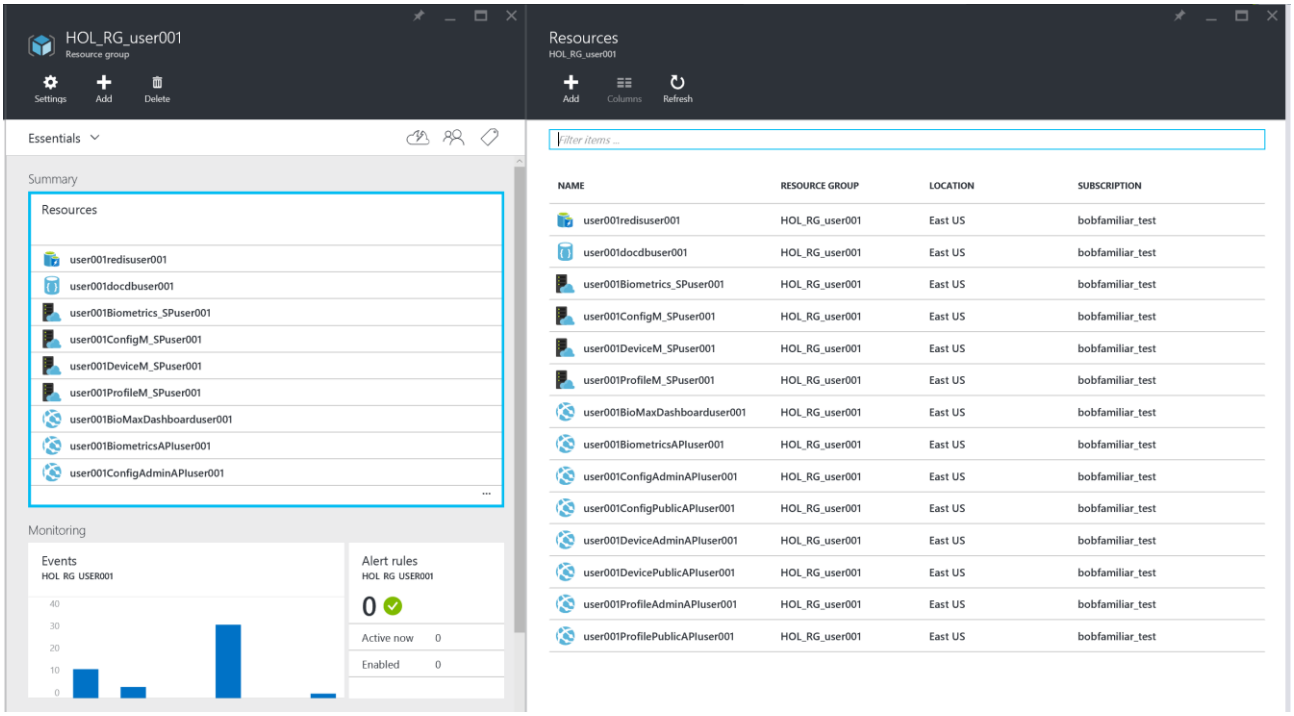

- Click the Options button  in the upper right corner of the manager window to bring up the options dialog
- To add a new NuGet Packages location, click the + icon.
- Change the name to something meaningful (Local Packages, for example). Use the … button to navigate to the `Packages` folder at the top level of the repo. Select the folder, click Update, and then OK.




- Now you can switch between the online NuGet catalogs and this local NuGet catalog when making NuGet package references. Referencing shared NuGet packages is now fully integrated into your development environment.

| | | |
|---|---|---|
| STEP 2 | • Run the PowerShell console as Administrator<br>• Log onto azure using the **add-azureaccount** command and the Azure credentials supplied by the facilitator |  |
| STEP 3 | • Navigate to the c:\users\[account]\source\repos\microservices-iot-azure\HandsOnLabs\Automation folder<br>• Type the provision command and provide the following values for the input parameters<br><br>> .\Provision<br><br>Repo: c:\users\[account]\source\repos\microservices-iot-azure<br><br>Subscription: [subscription]<br><br>Location: East US<br><br>UserTag: [usertag] | |
| STEP 4 | • Navigate to http://portal.azure.com<br>• View Resource Groups and select the resource group that has the name HOL_RG_usertag where usertag is your assigned user tag.<br>• Validate that you have provisioned an instance of Redis, DocumentDb and SQL Database |  |
| STEP 5 | • Open the Include-ConnectionStrings.Ps1 script file in Notepad.<br>• Click on the DocumentDb instance and the select the Keys option form the Settings blade.<br>• Copy the URI, Primary Key and Primary Connection String into the Include-ConnectionStrings.ps1 script located in the Automation folder<br>• Click on the Redis Instance and copy the Hostname and Primary Key into the connection string file and save. |  |

| | |
|---|---|
| STEP 6 | • Navigate to the HandsOnLabs\microservices\common\automation folder<br>• Run the 01-Build-Wire and the 02-Build-Store scripts. Each script takes two parameters, the repo path and a configuration, debug or release<br><br>> .\01-build-wire<br><br>> .\02-build-store |
| STEP 7 | • Navigate to the HandsOnLabs\microservices\config\automation folder<br>• Run the 01-Provision-ConfigM script using the same 4 input arguments that were used for the data services provisioning script<br>• Run the 02-Build-ConfigM script supplying the path to the repo and the build configuration of 'debug'<br>• Run the 03-Deploy-ConfigM command as follows:<br><br>> .\01-Provision-ConfigM<br><br>> .\02-Build-ConfigM<br><br>> .\03-Deploy-ConfigM –DeployData:$true<br><br>• Provide the same 4 parameters used in provisioning |
| STEP 8 | • Repeat Step 6 for the DeviceM, ProfileM and Biometrics microservices<br><br>NOTE: DeviceM and ProfileM will deploy data to DocumentDb, Biometrics does not have a data deployment option as the data for that service will come from the IoT services |
| STEP 9 | • Using Postman or a browser, test the microservice deployments using these end points<br><br>http://[usertag]configmadminapi[usertag].azurewebsites.net/config/manifests<br>- return a list of all manifests<br>http://[usertag]configpublicapi[usertag].azurewebsites.net/config/manifests/name/DeviceM<br>- return the DeviceM manifest<br>http://[usertag]deviceadminapi[usertag].azurewebsites.net/device/registrations<br>- return all the device registrations<br>http://[usertag]devicepublicapi[usertag].azurewebsites.net/device/registrations/model/BIOMAX-HOME<br>- return all device registrations for BioMax Home units<br>http://[usertag]profileadminapi[usertag].azurewebsites.net/profile/users<br>- return a list of all user profiles<br>http://[usertag]profilepublicapi[usertag].azurewebsites.net/profile/users/state/MA<br>- return a list of user profiles in Massachusetts |




*State of Azure after Lab 1*