**Ramp-Up Document**

# First Things (Administrative)

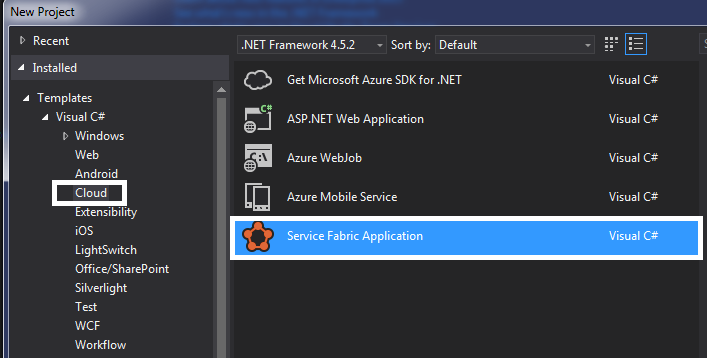
* Do you have a computer?
* Do you have a desk/location to call your own?
* Do you have a badge?
* Do you have a Global User Id?
  + Make sure you change your password to something custom using the Change Password option on the JCI home portal.
* Do you have access to Outlook for email?
* Do you have access to Skype for Business?
  + Sign-in Address is your JCI email
  + User Name is <global id>@jci.com
  + Password is global id password
  + Do you have MSDN access? [nitin.shakya@jci.com](mailto:nitin.shakya@jci.com) Zaxs1@123
  + Activate your account using your JCI email address as a Microsoft Id.

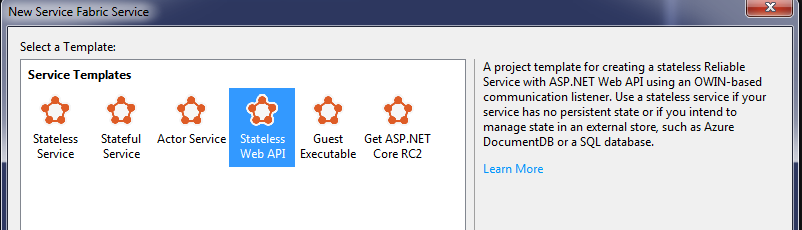
# Applications

* Visual Studio 2015 Enterprise
  + Update to VS Update 2
    - You may want to install the updates to Update 2 (has .NET core debugging fixes). https://msdn.microsoft.com/library/mt695655.aspx
  + Update any other packages (e.g. NuGet)
  + Install PowerShell Tools for Visual Studio 2015 (not necessary, but may make your life easier).
* Microsoft Azure
  + Microsoft Azure Service Fabric SDK and Tool 2.1 (VS 2015)
  + Microsoft Azure Power shell
  + Microsoft Azure SDK for .NET (VS 2015) - 2.9
* ReSharper
  + Contact
* Postman

# Service Fabric Application

* Open Visual Studio and Add New project and follow below screen shots.





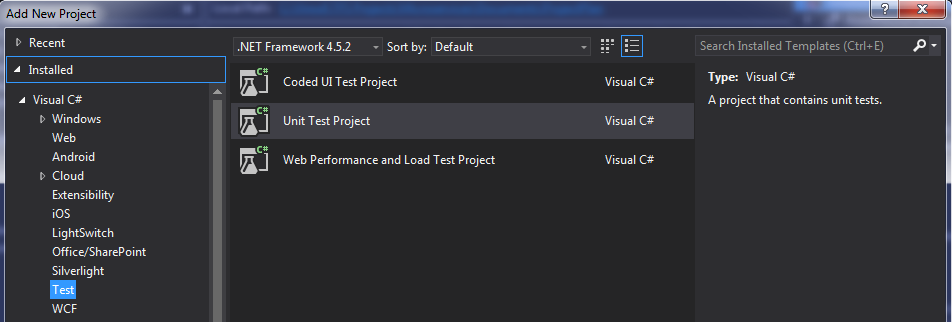
* Before running service fabric application, Open Power shell Command prompt (Run as Administrator)
* Then run following command (First time only)
  + ***Set-ExecutionPolicy <Parameters>***
    - <***Parameters***>
      * **Restricted** - Does not load configuration files or run scripts. Restricted is the default execution policy.
      * **AllSigned** - Only scripts signed by a trusted publisher can be run.
      * **RemoteSigned** - Requires that all scripts and configuration files downloaded from the Internet be signed by a trusted publisher.
      * **Unrestricted** - Loads all configuration files and runs all scripts. If you run an unsigned script that was downloaded from the Internet, you are prompted for permission before it runs.
* Right click on Application Project and set “Application URL” ex: <http://localhost:5001/api/values>
* Set Ap plication Project as a “Start Up Project”
* Then press F5 (Run Application)

**Create Cluster in Azure**

* Please follow below link to create cluster in Azure:
  + [**https://azure.microsoft.com/en-in/documentation/articles/service-fabric-cluster-creation-via-portal/**](https://azure.microsoft.com/en-in/documentation/articles/service-fabric-cluster-creation-via-portal/)

**Unit Test Cases**

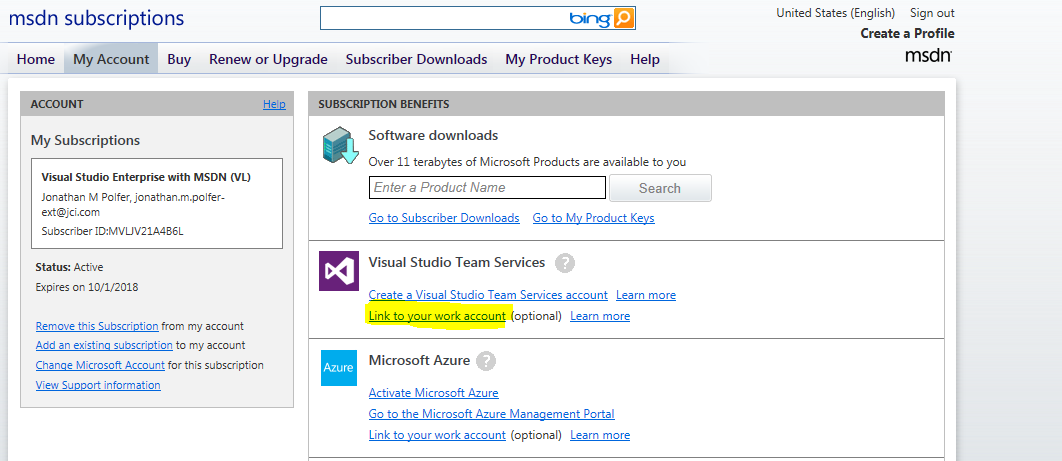
* Create Unit test project in your solution:



* Add service reference to unit test project
* Add following NuGet packages:
  + Rhino Mocks
  + Newtonsoft JSON – convert JSON object to actual class object
  + System.Net.Http.Formatting – Enable “ReadAsAsync” method in HttpResponseMessage
  + FizzWare.NBuilder – Creating dummy data

# Visual Studio Team Services (Visual Studio Online)

* Ensure an administrator has added access for you to VSO.
* Ensure your MSDN account has your JCI global id associated with it:
* Go to [www.microsoft.com/msdn](http://www.microsoft.com/msdn)
* Login with the MSDN account you activated with your JCI email address
* Click on MSDN Subscriptions in the top menu bar - should navigate you to the My Account page
* Click on Link to your work account under Visual Studio Team Services



* Put in <global id>@jci.com as your account
* Click Link
* Navigate to <https://jcidata.visualstudio.com/>
* Login as <global id>@jci.com

## Repositories

## Code Change Workflow

Below is a workflow to use when making code changes:

* Branch the master repository for your own work
  + Login to the TFS website
  + Go to the repository you want to work on and select the **Code** tab.
  + In the center there is a dropdown to select the current branch (will say master by default). Select it and pick the option to **“+ New Branch**”
  + Give your new branch a name (eg. your login id)
* Select the Branch in Visual Studio
  + Go to the **Branches** tab in VS
  + Under the “remotes/origin” section, select your branch (note: It might need to refresh a couple of times for the branch to show up)
* Check-in your changes to your local branch.
  + Select the **Changes** tab in VS.
  + Make sure the branch (at the top) is set to your branch (not master)
  + Check in the changes
* Sync the changes to the server
  + Go to the **Sync** tab in VS
  + Either select **Sync** which will do it all or just do the **Push**.
* Do a pull request to get your changes into Master
  + Go to the Pull tab in VS
  + Select “New Pull Request”
  + Select your local branch and then master
* Add reviewers to the pull request
  + Login to the TFS website
  + Open the Pull request
  + Add reviewers that at appropriate for the project

# Source Code Setup

# Development Environments

## Server/URL

# Appendix: DevOps Tools

|  |  |  |  |
| --- | --- | --- | --- |
| Team Collaboration | Document Storage for Team | Office 365 | [https://my.jci.com](https://my.jci.com/) |
| Develop | Integrated Developer Env (IDE) | Visual Studio 2015 Enterprise (C# code) |  |
| Builds & CI /Deploy | Builds | MSBuild Tools 2015 | <https://www.microsoft.com/en-us/download/details.aspx?id=48159> |
|  | CI | Visual Studio Team Services Build/Release | <https://www.visualstudio.com/en-us/features/continuous-integration-vs.aspx> |
| Testing | Test Case Management | Visual Studio Team Services - Test | <https://www.visualstudio.com/en-us/docs/test/overview> |
|  | Load Tests | Visual Studio Team Services - Test | <https://www.visualstudio.com/en-us/docs/test/overview> |
|  |  | jMeter - also using VSTS | <http://jmeter.apache.org/> |
|  | Web Test Automation | Coded UI Tests | <https://msdn.microsoft.com/en-us/library/dd286726.aspx> |
|  |  | Selenium | <http://www.seleniumhq.org/> |
| Package Management | Nuget Package Feed | myget | [http://jci.myget.org](http://jci.myget.org/) |
|  |  | Visual Studio Team Services - Package Management | <https://www.visualstudio.com/en-us/docs/package/what-is-packaging> |
|  | 3rd Party Package Installation | Chocolatey | <https://chocolatey.org/> |
| Azure Development |  |  |  |
|  |  |  |  |
|  |  |  |  |