



# OpenShift3.X Origin Tool for Visual Studio

User Guide

Version: 1.0

[Click2Cloud Inc.](#)



## **Legal Notice**

Copyright 2016 Click2Cloud Inc. The text of and illustrations in this document are licensed by Click2Cloud Inc.

## **Abstract**

This guide provides guideline for Click2Cloud OpenShift 3.X Origin Tool for Visual Studio 2015 and documents its application management functions on OpenShift 3.X Origin Cloud.



Introduction .....	3
Hardware Pre-requisite: .....	3
Pre-Requisites Software .....	3
<b>Installing GitHub Extension for Visual Studio 2015</b> .....	5
<b>Installing Git for Windows</b> .....	6
<b>Installing OpenShift 3 CLI</b> .....	7
Installing OpenShift 3.X Origin Tool for Visual Studio .....	8
Create Application on OpenShift .....	10
Import / Open an existing application from OpenShift.....	18
Manage Projects.....	25
OpenShift Explorer - The embedded OpenShift Console for Visual Studio.....	26
<b>Deploy Application Changes</b> .....	26
<b>Deploy Application Build (Manually)</b> .....	26
<b>Deploy Application Build (Automatically Using Git Webhooks)</b> .....	27
<b>Log Streaming</b> .....	28
<b>Stream Application Logs</b> .....	28
<b>Stream Build Logs</b> .....	29
<b>Manage an OpenShift Project</b> .....	30
<b>Open Terminal</b> .....	31
<b>Port-forwarding</b> .....	32



## Introduction

**OpenShift** is Red Hat's Platform as a Service (**PaaS**) offering which provides an application platform in the cloud where application developers and teams can build, test, deploy, and run their applications with automatic scaling. OpenShift provides developers with a wide selection of programming languages and frameworks including Java, .Net, Ruby, PHP, Perl, Python, and Node.js.

From developing to deploying on OpenShift 3, **Click2Cloud OpenShift 3.X Origin Tool for Visual Studio** provides you with a fully-fledged environment for your project and aligns with the standard workflows within Visual Studio. While designing the OpenShift Tool for Visual Studio, we've focused on giving integrated developer as well as user experience for **.Net Developers** to support the application lifecycle and seamless deployment in **Visual Studio Integrated Development Environment (IDE)**.

The following step-by-step guide shows you how to install tool, connect, create and deploy applications in OpenShift, from your Visual Studio IDE.

## Hardware Pre-requisite:

1. Processor: **Intel Core i3** (or above for better performance)
2. RAM: **Minimum 8 GB** (or above for better performance)
3. Hard disk: **10 GB Minimum**

## Pre-Requisites Software

Before you can install **Click2Cloud OpenShift 3.X Origin Tool for Visual Studio**, you must download and install the following pre-requisites software:

1. **Microsoft Visual Studio 2015 Update 1**

Install [Microsoft Visual Studio 2015 Update 1](#), the following SKUs are supported:

- a. Enterprise 2015
- b. Professional 2015
- c. Community 2015



## 2. Microsoft ASP.NET and Web Tools 2015 (RC)

Install [Microsoft ASP.NET and Web Tools 2015 \(RC\)](#).

Note: Please install the corresponding language pack to make sure Web Tools works properly.

## 3. GitHub Extension for Visual Studio 2015

To download and install GitHub Extension for Visual Studio, follow instruction specified in [Installing GitHub Extension for Visual Studio 2015](#).

## 4. Git for Windows

To download and install Git for Windows, follow instructions specified in [Installing Git for Windows](#) or watch video [Git on Windows](#)

## 5. OpenShift Command Line Interface (CLI) – Latest Version

To download and install OpenShift 3 CLI, follow instructions specified in [Installing OpenShift 3 CLI](#).

## 6. Vagrant V 1.7.4 .msi , Virtual Box and Vagrant File

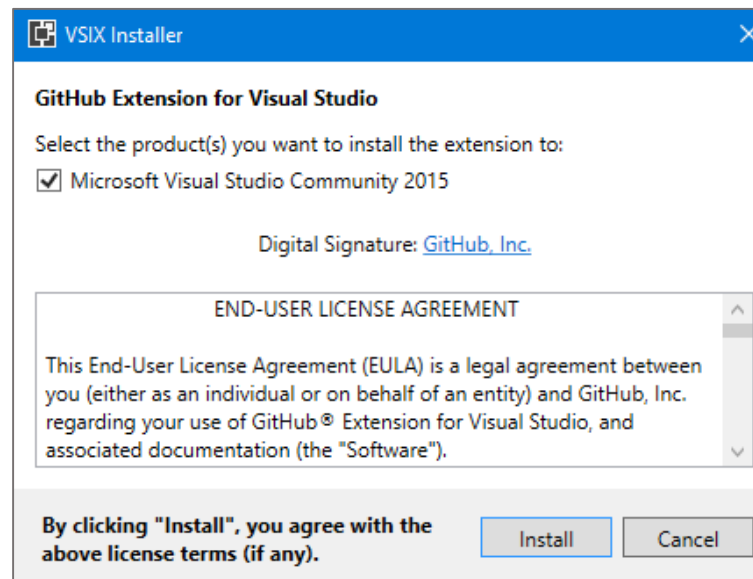
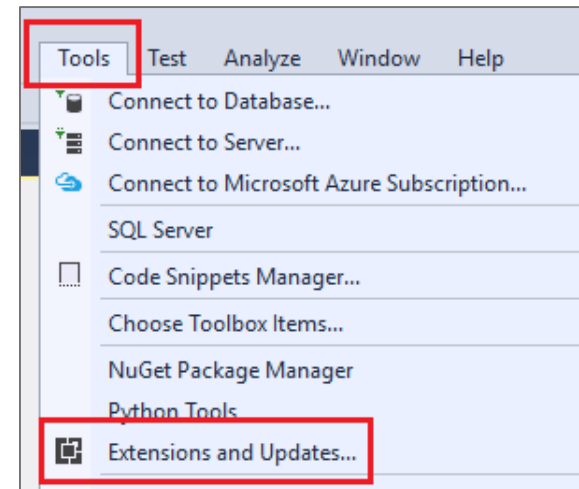
To download and install Vagrant, Virtual Box and Vagrant File, follow instructions specified in <https://www.openshift.org/vm/>

**Please Note:** You will need the 'Vagrant, Virtual Box and Vagrant File' for creating OpenShift 3.X Origin environment on local machine

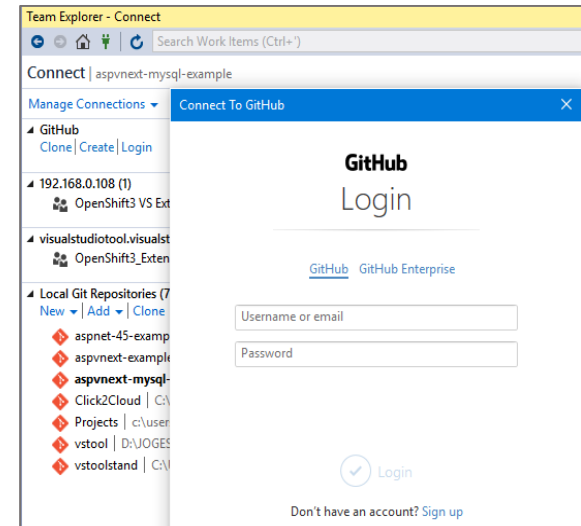
**Note:** You must have sufficient privileges to install software on your computer. Depending on specific user permissions, disabling the User Account Control (UAC) on Windows 7, 8 or 10 operating system may be necessary.

## Installing GitHub Extension for Visual Studio 2015

1. You can download **GitHub Extension for Visual Studio 2015** from <https://visualstudio.github.com> or directly from Visual Studio Gallery **Tools -> Extensions and Updates -> Online**)
2. Double-click the **GitHub.VisualStudio.vsix** file you downloaded to launch the installation wizard. Select the Visual Studio product(s) and click on **Install**.



3. Once installation is done, close all instances of Visual Studio 2015.
4. Relaunch Visual Studio 2015 instance and then select **View -> Team Explorer -> Manage Connections -> Connect To GitHub**.
5. If you have already signed up for a GitHub account then you can input your **Username** and **Password** and click **Enter** key to login. If you do not have a GitHub account, you can create one through **Sign up** link at the bottom of the **Connect To GitHub** screen.



## Installing Git for Windows

You can download latest version of **Git for Windows** from <https://git-scm.com/download/win> and install it to your desired location.

**Note:** Please make sure that you have set the System Environment Variable **PATH** with Git installation location. Refer [this](#) link to set environment variables on Windows. Once setup is done, check your **PATH**, open the **Command Prompt** and run:

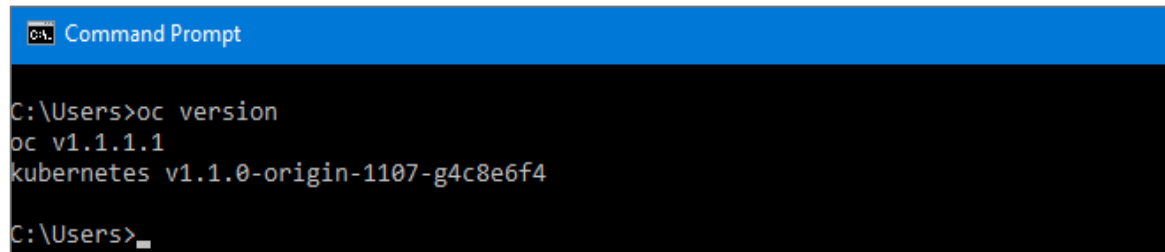
path

## Installing OpenShift 3 CLI

1. Download latest version of **OpenShift 3 CLI for Windows** from <https://github.com/openshift/origin/releases>.
2. Unzip the archive and move the OC client tool to your desired location.
3. Now set the System Environment Variable **PATH** with location of OC client tool location.
4. Refer [this](#) link to set environment variables on Windows.
5. Now verify OpenShift 3 CLI installation is configured correctly. To do so, open **Command Prompt** and run:

### oc version

6. If OC client tool is configured correctly, you will get version of OC binary as shown in below image



```
C:\Users>oc version
oc v1.1.1.1
kubernetes v1.1.0-origin-1107-g4c8e6f4
C:\Users>
```

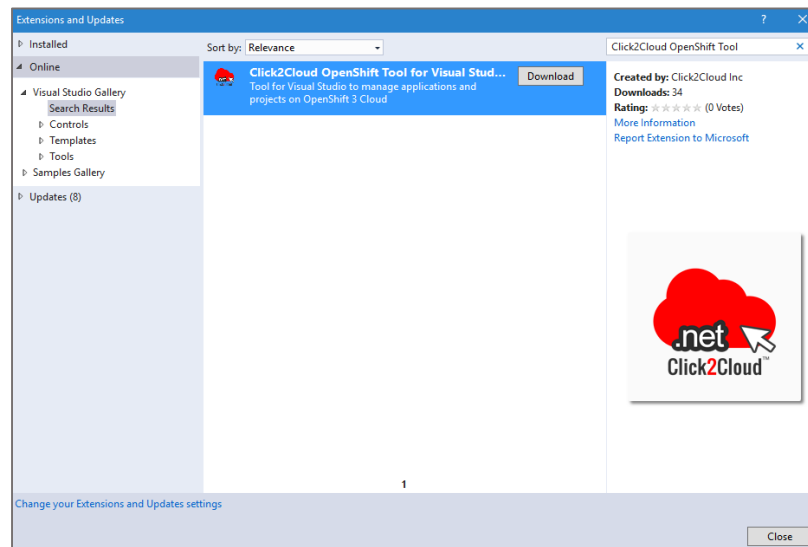


## Installing OpenShift 3.X Origin Tool for Visual Studio

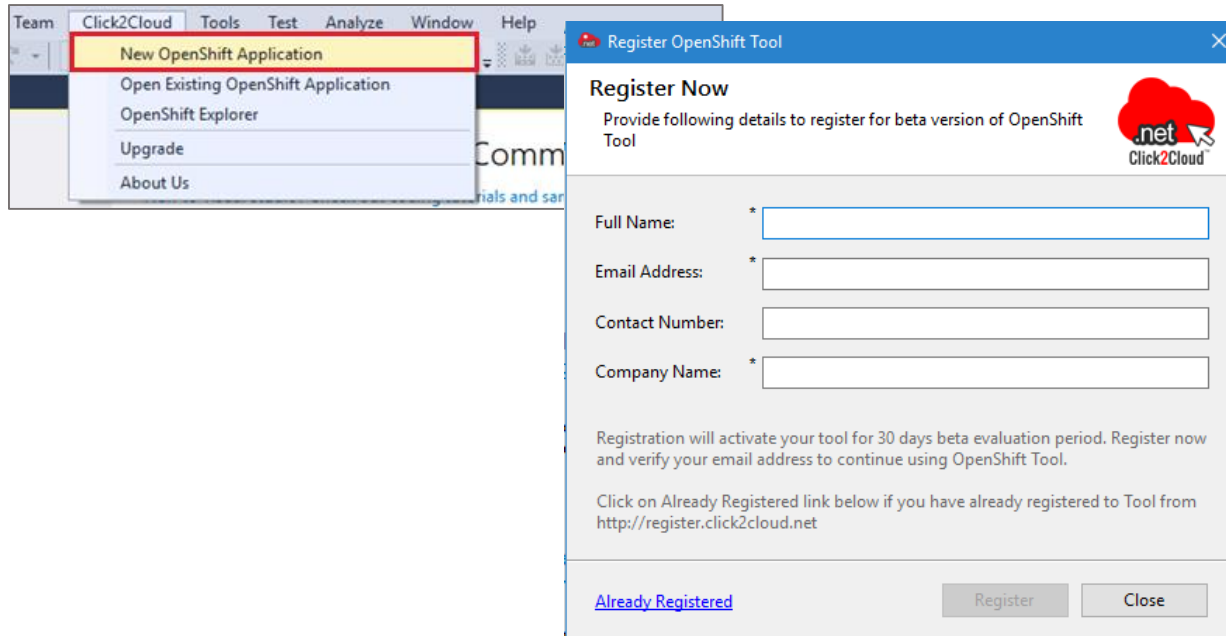
You can download the **Click2Cloud OpenShift 3.X Origin Tool for Visual Studio (Beta 1)** from [Visual Studio Gallery](#) after successfully registering for beta evaluation period.

For getting OpenShift Tool for Visual Studio, follow below steps or watch video [Click2Cloud OpenShift Extension for Visual Studio #1: Install & Register](#) to download extension from <http://register.click2cloud.net>.

1. Launch **Visual Studio 2015**
2. Go to **Tools -> Extensions and Updates -> Online** and search for **Click2Cloud OpenShift 3.X Tool for Visual Studio**.
3. Click on **Download** button. This will download and install **Click2Cloud OpenShift 3.X Tool for Visual Studio**.



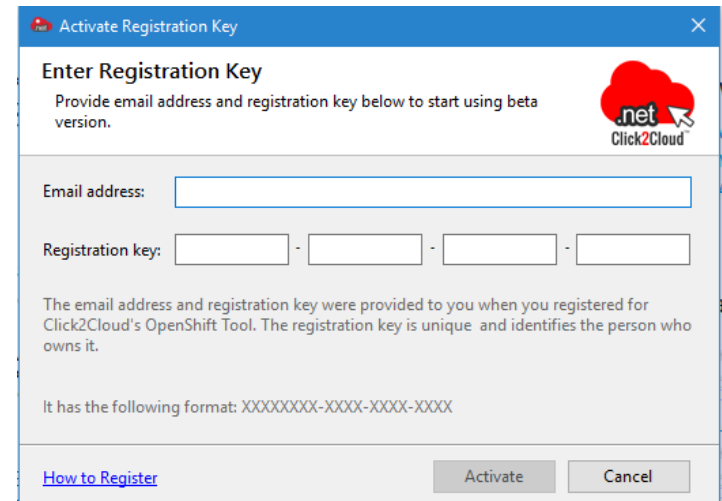
4. Launch OpenShift 3 Wizard forms to register to beta evaluation period from **Click2Cloud** menu -> **New OpenShift Application** and fill in correct information and click on **Register** button.



**OR**

If you have registered from <http://register.click2cloud.net> and have registration key, click on **Already Registered** link from **Register OpenShift Tool** box and provide your **Email address** and **Registration key** received after successful registration in **Activate Registration Key** window.

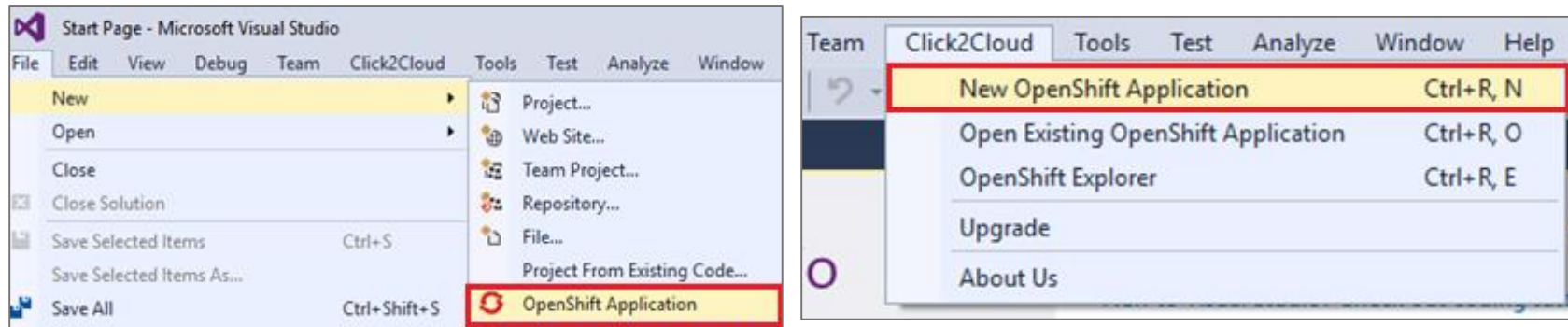
5. Once you receive verification mail from Click2Cloud, **verify** your email address to activate your registration.



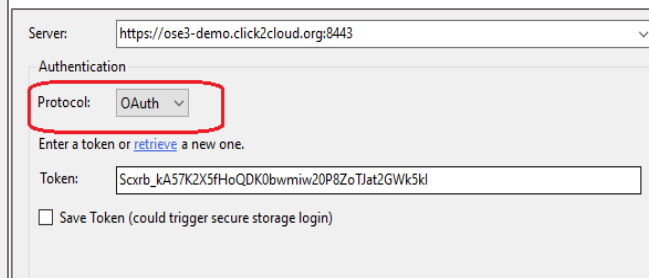
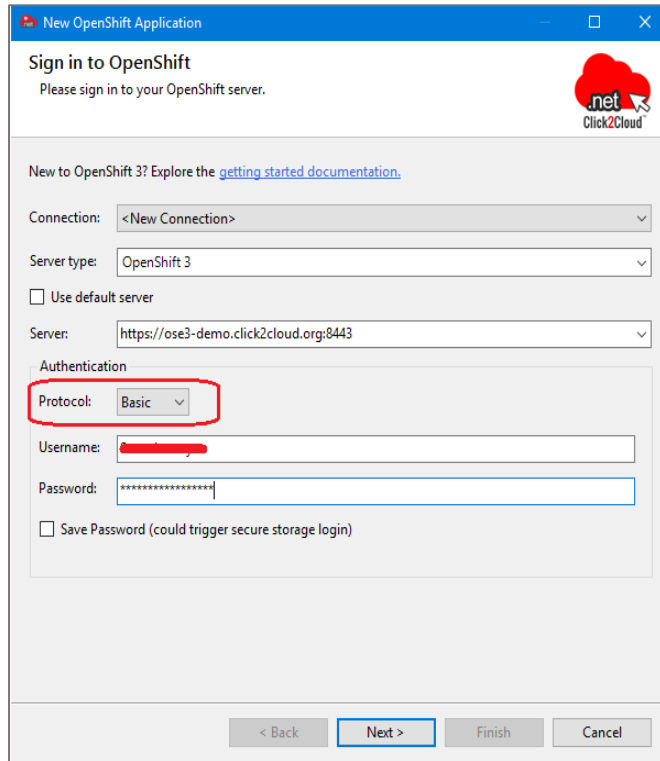
## Create Application on OpenShift

Now you have the OpenShift Tool installed, it's time to create OpenShift project and begin development. You can follow below instructions to create new application on OpenShift or watch video [Click2Cloud OpenShift Extension for Visual Studio #2: Create New Application](#)

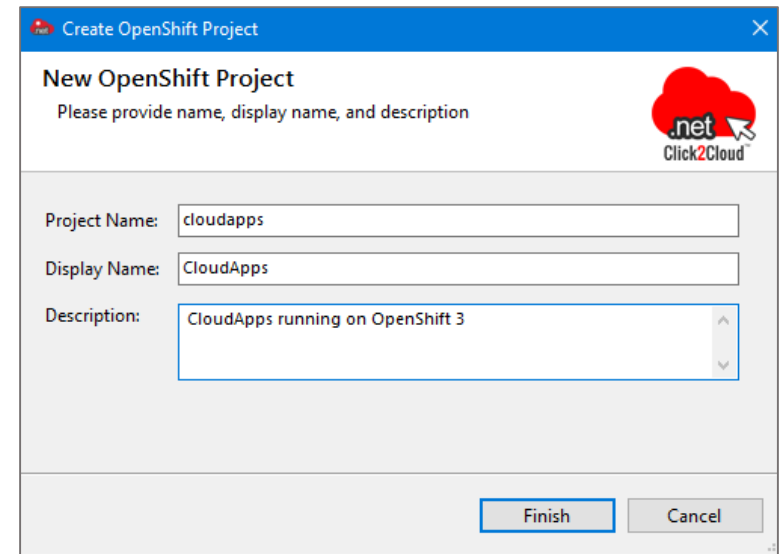
1. Select **Click2Cloud** menu from Visual Studio menu bar and select **New OpenShift Application** or select **File -> New -> OpenShift Application**



2. Before you can start creating and deploying your projects, you will need to authenticate with the OpenShift 3 Sever.
  - i. In **New OpenShift Application** wizard select the **New Connection** to create OpenShift Connection and enter the **OpenShift Server URL**
  - ii. OpenShift 3 supports two authentication protocols i.e.
    - Basic Authentication:** type in your login/password
    - OAuth Authentication:** you retrieve a token by clicking the retrieve link, following the prompts in the browser, and pasting it back into the connection wizard.
  - iii. Check on **Save Password** checkbox, if you want to be remembered login details for next time.



3. Once you have successfully authenticated to the OpenShift server, you will be prompted with **New OpenShift Project** dialog box, if no project created in your account yet.
4. Provide your new **Project Name** and other details and click on **Finish** button.
5. After creating your project, you will be presented with **Select Template** wizard. The wizard provides the following functionality:

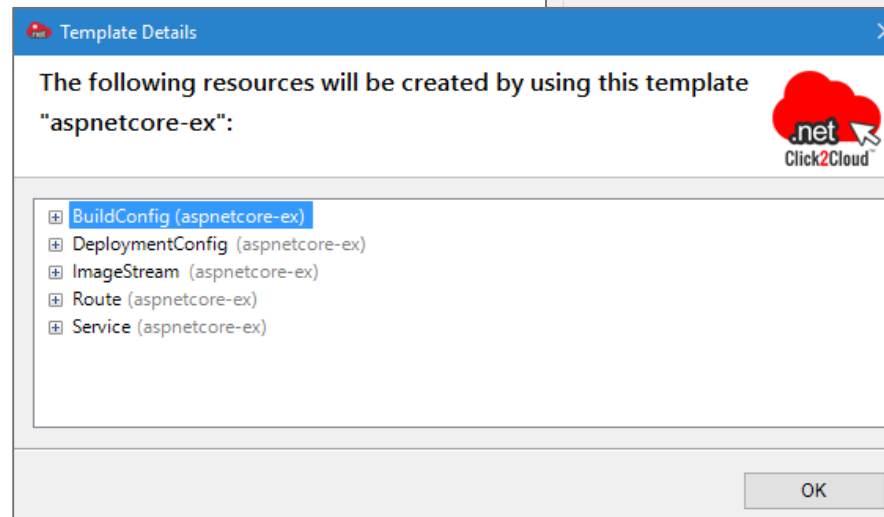
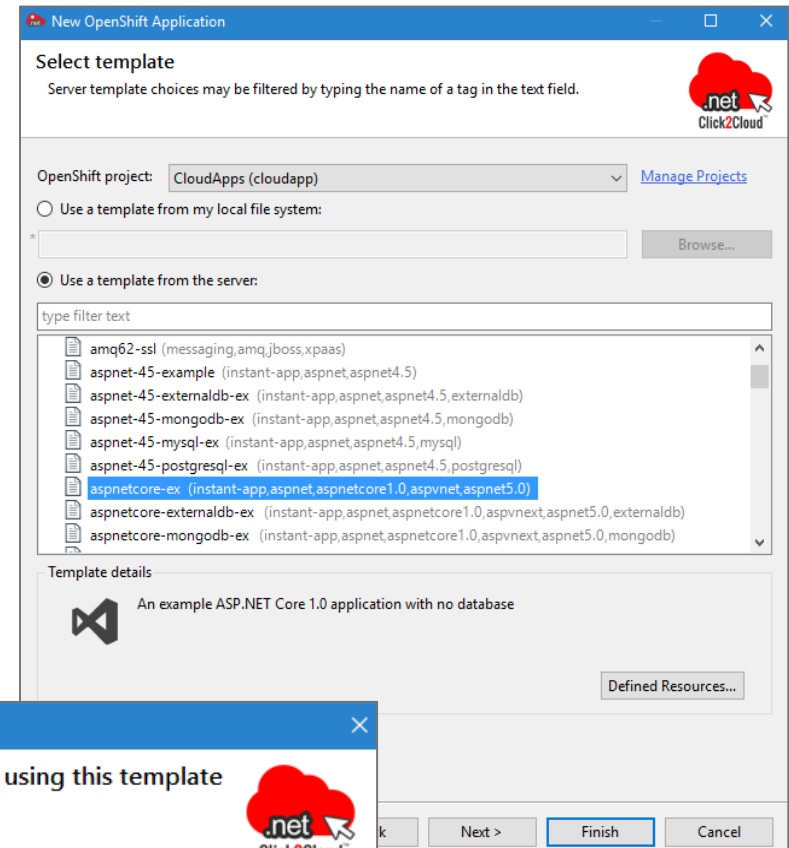


**Manage Projects:** To manage projects on OpenShift 3 server

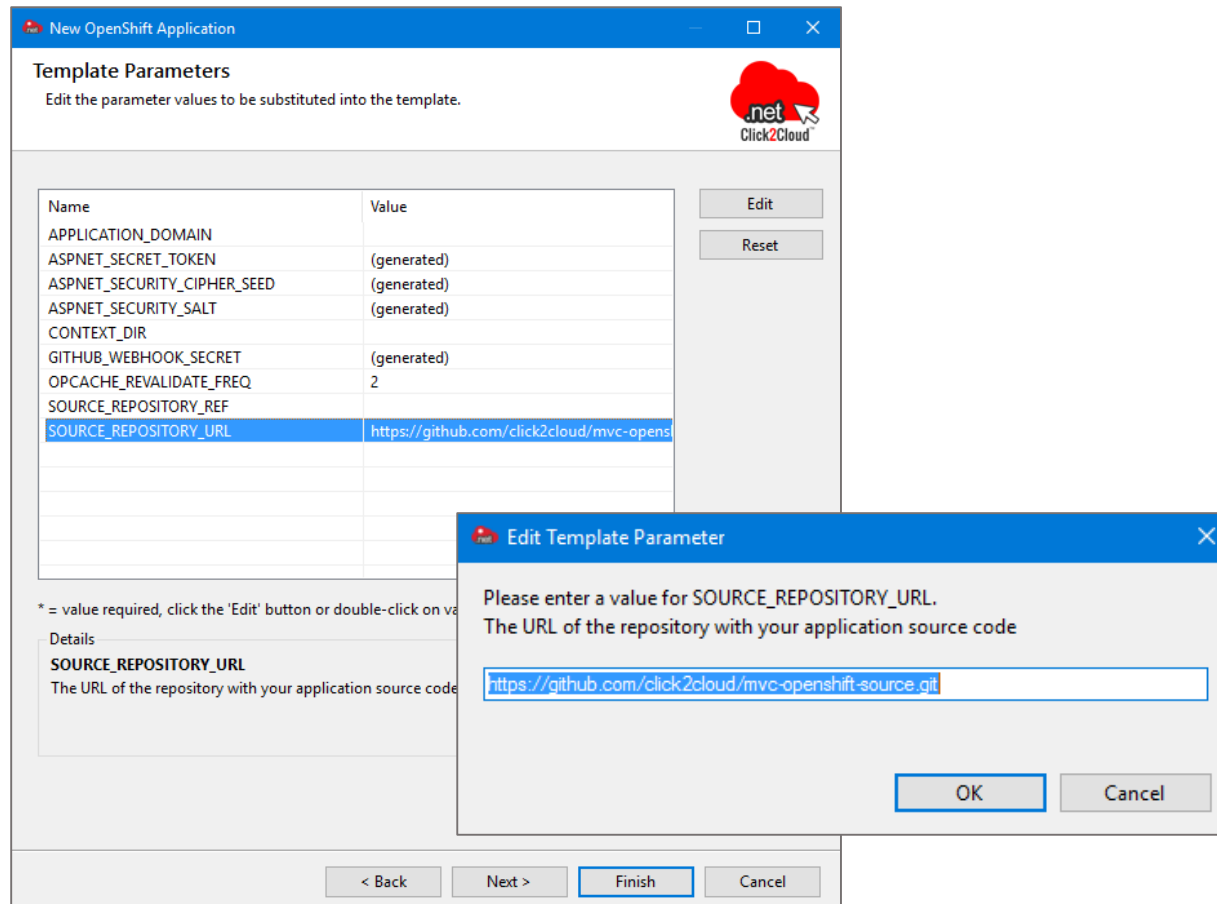
**Use a template either from local file system or from the server:** Using this you can create an application based on template offered by the OpenShift Server, or provide a template that you have ready on your local file system.

**Defined Resources:** To view resource going to be created based on selected template.

6. You can use list of templates available on server to create OpenShift 3 resources that make up an application or provide your custom template. In this example we will use **aspnetcore-ex** template to create ASP.NET Core 1.0 Application.
7. Then click on **Defined Resources...** button. Using **Template Details** dialog box, you can find the OpenShift 3 Resources which are going to be created with selected template.
8. Click **Next** or click **Finish** to create application with default template parameters and labels.



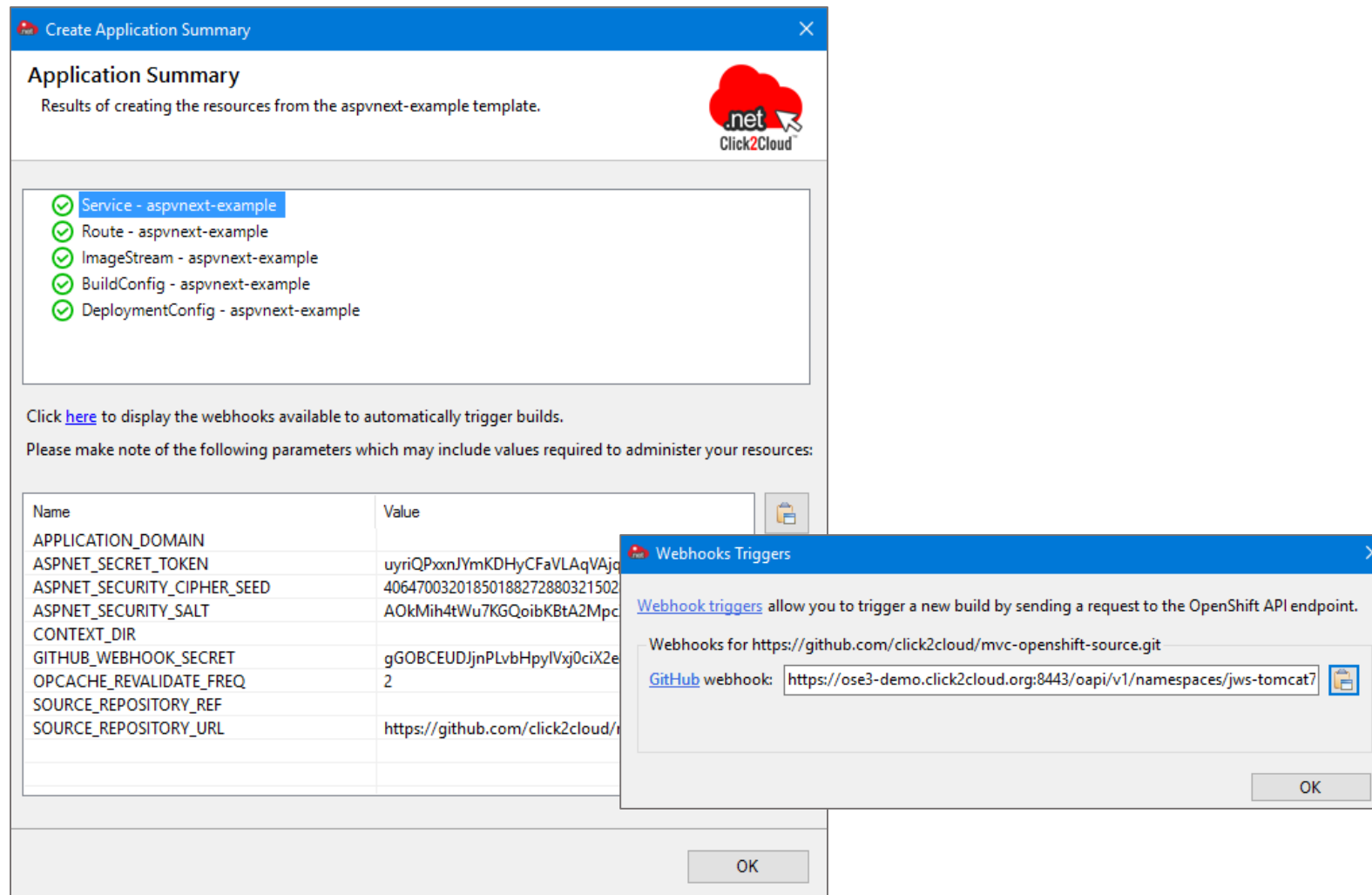
- On selecting **Next** from **Select Template** screen, you will be prompted with **Template Parameters**, this wizard allows you to edit the template parameters like GitHub repository URL, to customize them for your needs.
- Click **Next** or click **Finish** to create application with default labels.





14. Once application is created successfully on OpenShift 3 cloud, you will be prompted with **Application Summary** page.

A resource name collision detection prevents the creation of incomplete applications. A summary of all generated application parameters is shown, with a **copy-to-clipboard** button and a link to **Webhook build triggers**, once the application has been created.



The screenshot shows two overlapping dialog boxes from the Click2Cloud application. The background dialog is titled 'Create Application Summary' and contains an 'Application Summary' section with a list of resources created: Service - aspvnext-example, Route - aspvnext-example, ImageStream - aspvnext-example, BuildConfig - aspvnext-example, and DeploymentConfig - aspvnext-example. Below this list is a table of parameters and their values, and a 'Webhooks Triggers' dialog box is overlaid on top.

**Application Summary**  
Results of creating the resources from the aspvnext-example template.

- ✓ Service - aspvnext-example
- ✓ Route - aspvnext-example
- ✓ ImageStream - aspvnext-example
- ✓ BuildConfig - aspvnext-example
- ✓ DeploymentConfig - aspvnext-example

Click [here](#) to display the webhooks available to automatically trigger builds.

Please make note of the following parameters which may include values required to administer your resources:

Name	Value
APPLICATION_DOMAIN	
ASPNET_SECRET_TOKEN	uyriQPxxnJYmKDHycFaVLAqVAj...
ASPNET_SECURITY_CIPHER_SEED	40647003201850188272880321502...
ASPNET_SECURITY_SALT	AOkMih4tWu7KGQoibKBtA2Mpc...
CONTEXT_DIR	
GITHUB_WEBHOOK_SECRET	gGOBCEUDJjnPlvbHpylVxj0ciX2e...
OPCACHE_REVALIDATE_FREQ	2
SOURCE_REPOSITORY_REF	
SOURCE_REPOSITORY_URL	https://github.com/click2cloud/...

**Webhooks Triggers**

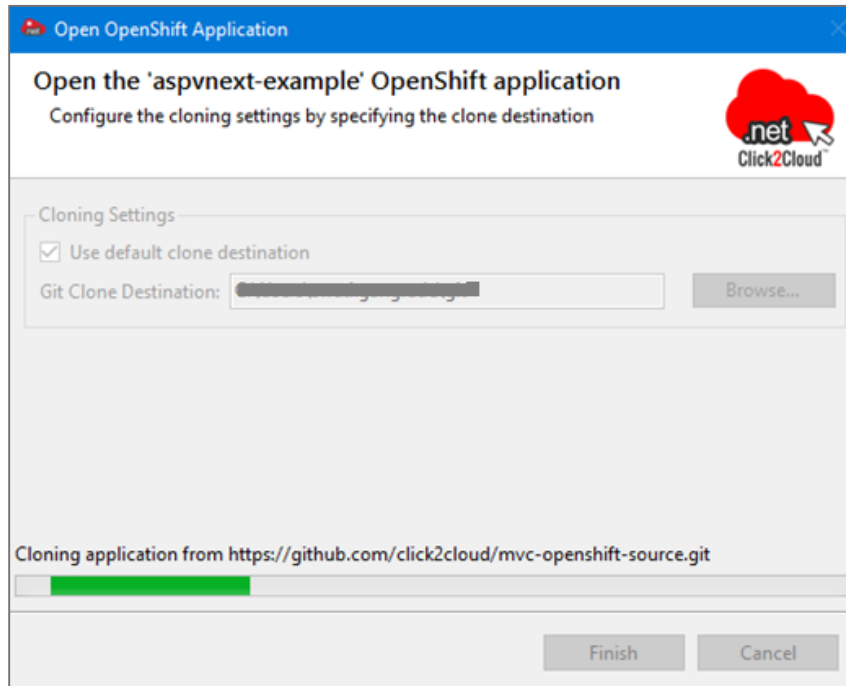
[Webhook triggers](#) allow you to trigger a new build by sending a request to the OpenShift API endpoint.

Webhooks for <https://github.com/click2cloud/mvc-openshift-source.git>

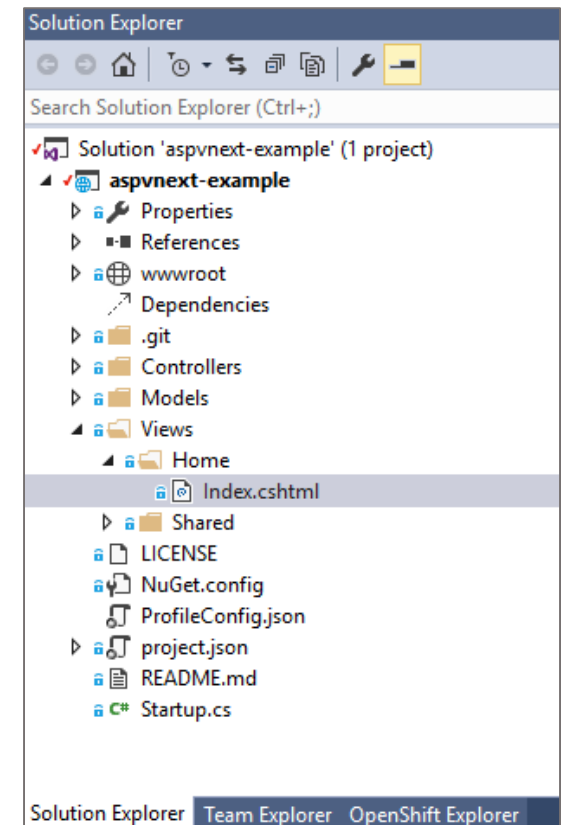
GitHub webhook: <https://ose3-demo.click2cloud.org:8443/oapi/v1/namespaces/jws-tomcat7>



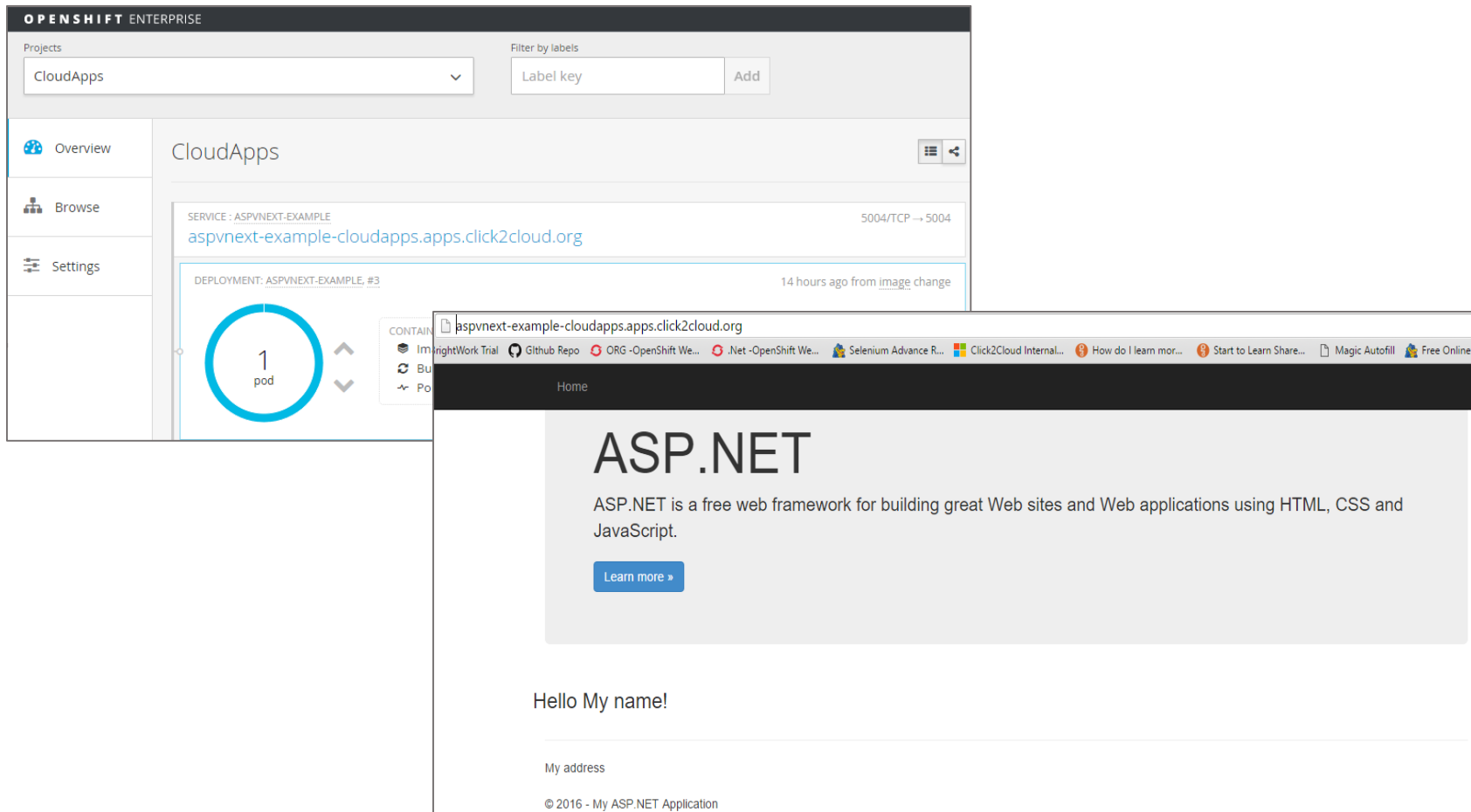
- Click on **OK** button. The **Open OpenShift Application** Dialog box will be open.  
This dialog box allows you to clone the application to specific location on your local file system.
- Uncheck **Use default clone destination** checkbox; if you wish to clone application other than default location; and provide required destination path from local file system and click **Finish**.



- As application cloning process is complete, the application will get launched automatically in **Solution Explorer**.



18. On OpenShift server, once application build process is completed for newly created application and Pod is running up, you can see application output as below.

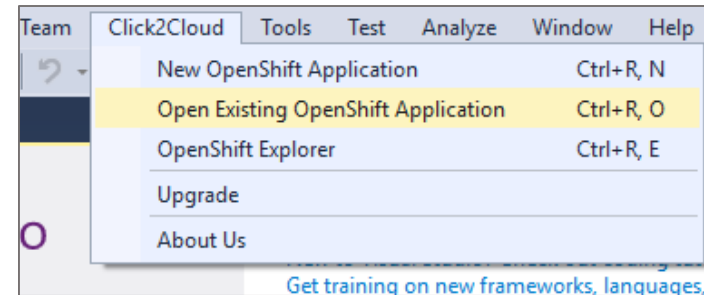
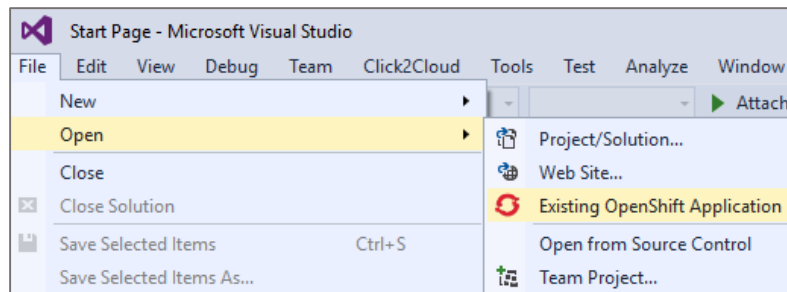


The screenshot displays the OpenShift Enterprise console interface. The top navigation bar shows 'OPENSHT ENTERPRISE'. Below it, the 'Projects' dropdown is set to 'CloudApps'. The main content area shows the 'CloudApps' project details, including the service 'ASP.VNEXT-EXAMPLE' and the deployment 'ASP.VNEXT-EXAMPLE, #3'. A circular progress indicator shows '1 pod' is running. A browser window is overlaid on the console, displaying the application output: 'ASP.NET' with a description and a 'Learn more' button, followed by 'Hello My name!' and 'My address'.

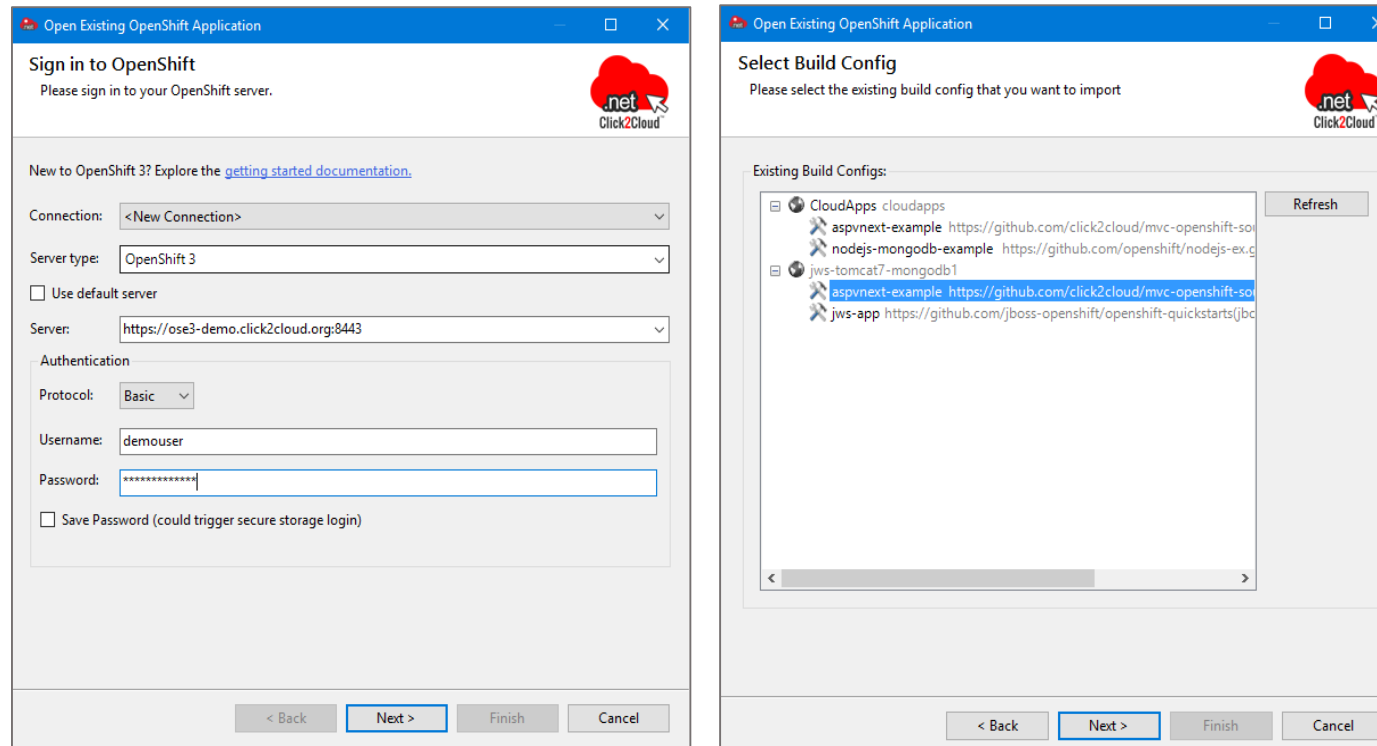
## Import / Open an existing application from OpenShift

An existing OpenShift 3 application can be imported/opened into Visual Studio using the **Open Existing OpenShift Application** menu available in the **Click2Cloud** menu or by selecting **File -> Open -> Existing OpenShift Application**. You can follow below instructions to open existing application from OpenShift or watch video [Click2Cloud OpenShift Extension for Visual Studio #3: Modify Existing Application](#)

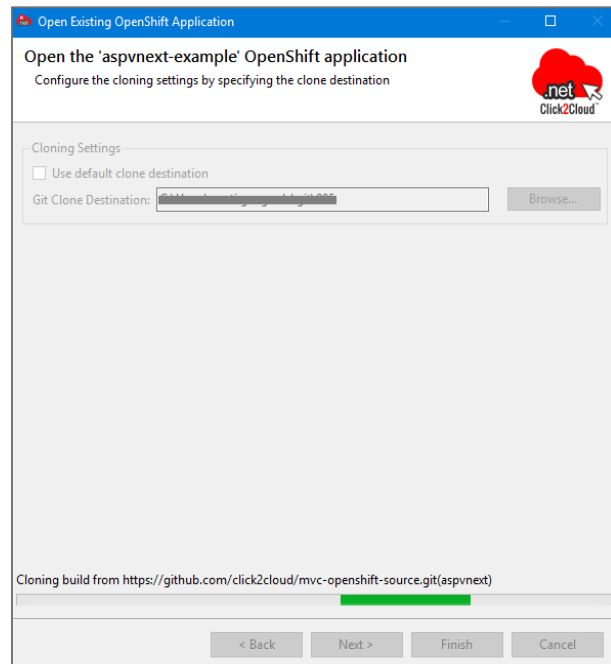
1. Select **Click2Cloud -> Open Existing OpenShift Application** or **File -> Open -> Existing OpenShift Application**
2. Provide server details and login credentials or select existing connection from **Connection** dropdown in **Sign in to OpenShift** screen and click **Next**



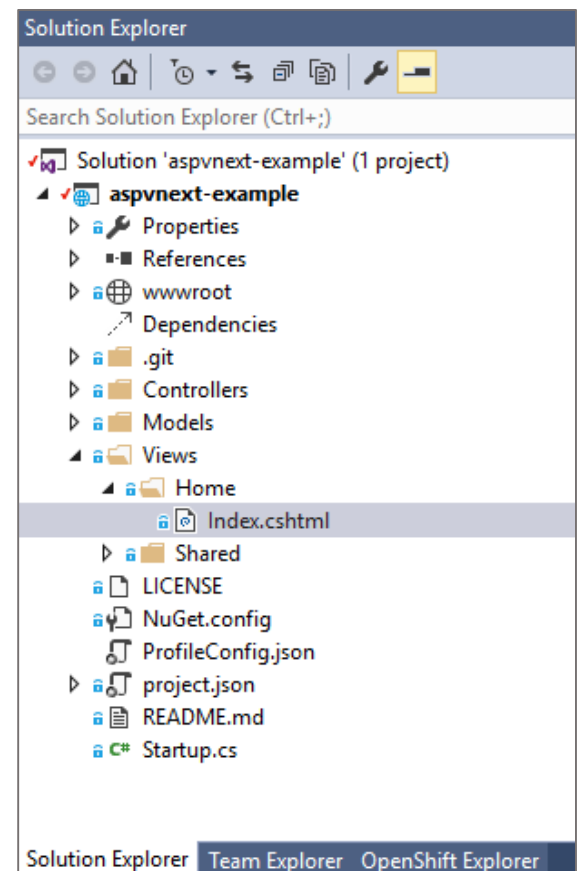
3. Select application from **Existing Build Configs** section in **Select Build Config** screen and Click **Next**



4. The **Open Existing OpenShift Application** screen will be open.  
This screen allows you to clone the application to specific location on your local file system.
5. Uncheck **Use default clone destination** checkbox; if you wish to clone application other than default location; and provide required destination path from local file system and click **Finish**.

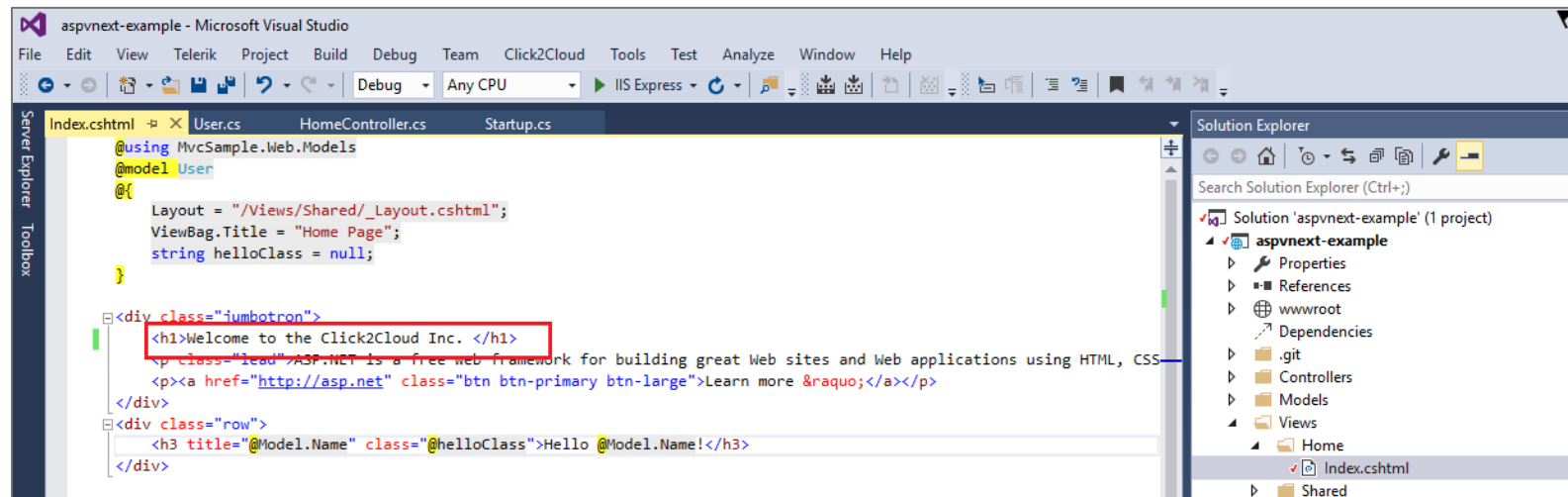


6. As application cloning process is complete, the application will get launched automatically in **Solution Explorer**.
7. Now, you can modify changes in application and deploy it into OpenShift.



**Note:** Before deploying changes make sure you have connected to GitHub and having permission on GitHub Repository to push the code changes.

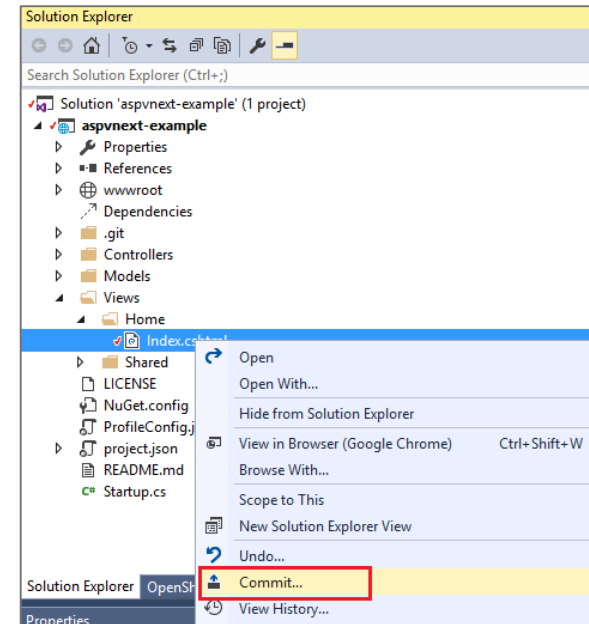
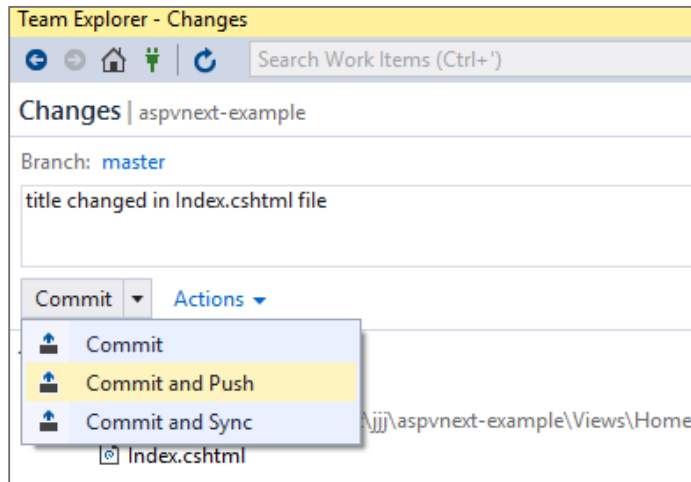
8. Click on **Solution Explorer** and do changes in source. Let's change the title in **Index.cshtml** file, i.e. **Welcome to the Click2Cloud Inc.** and save file.



- Now to deploy changes in OpenShift, first you need to commit your changes on GitHub repository. For that, right click on **Index.cshtml** in **Solution Explorer** file and click **Commit**

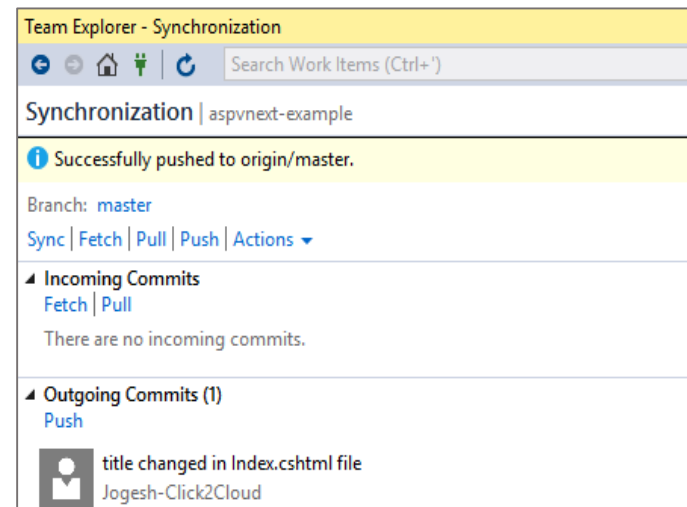
You will be prompted with GitHub window available on **Team Explorer**.

- Provide **comment** and click **Commit and Push**.

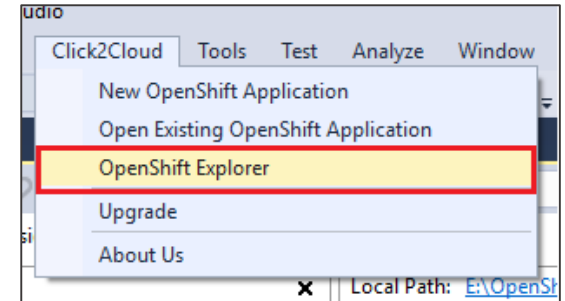
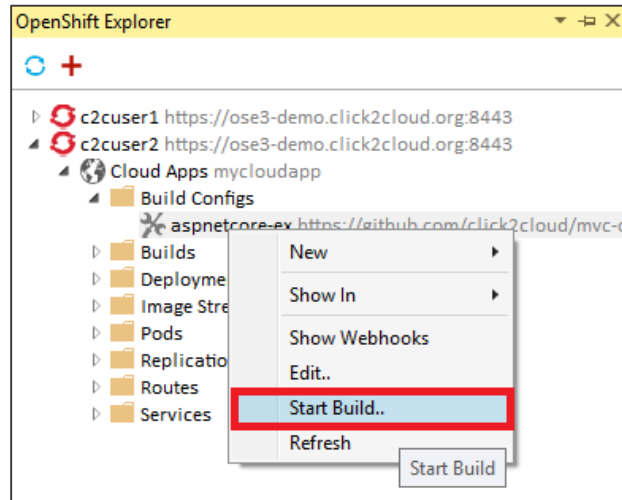


- If you have sufficient privileges in GitHub repository and you will see below screen after once changes are successfully pushed.
- Now to commit changes in OpenShift application, launch to **OpenShift Explorer** tool window from **Click2Cloud menu** -> **OpenShift Explorer**.

- Expand **Connection** and then **Project**.



14. Now expand **Build Configs** resource and right click on build config name for which you wish to deploy latest code change and click on **Start Build**.

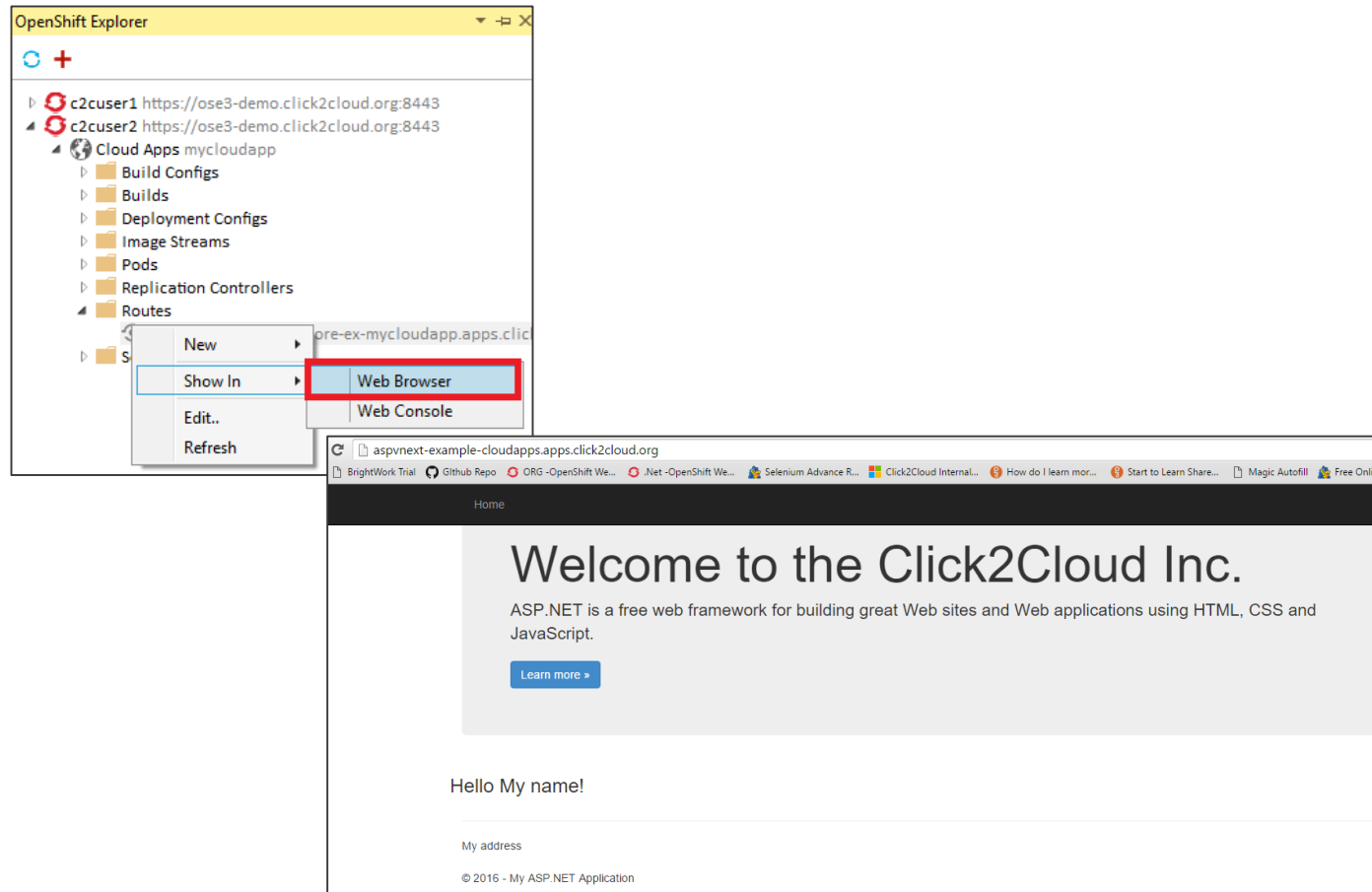


**Note:** If you want to configure automatic build deployment after committing changes in GitHub repository, install Webhook build triggers into your GitHub repository.

15. Within few seconds or minute you will find your build is completed and pod is running with updated source code changes.  
16. Let's launch the application and see the source code changes.



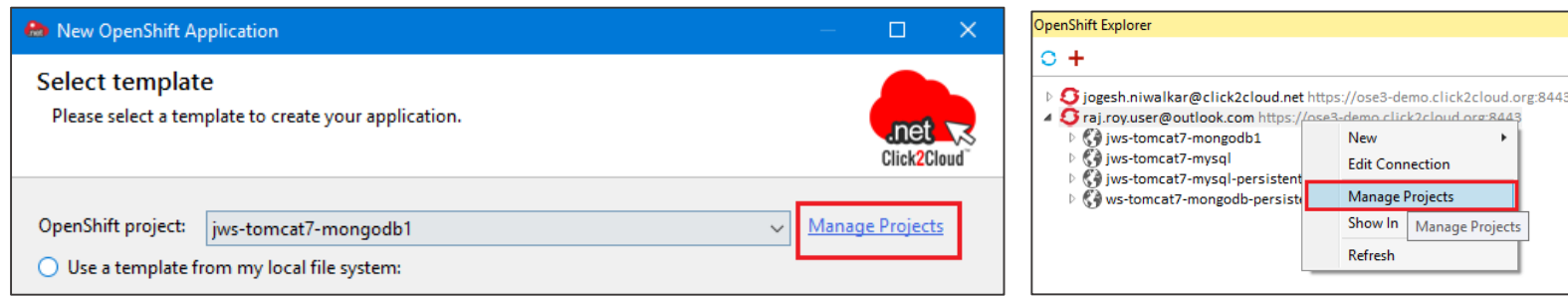
17. Expand the **Routes** folder -> right click on application route name -> **Show In -> Web Browser**. you will see the updated application running on your OpenShift cloud.



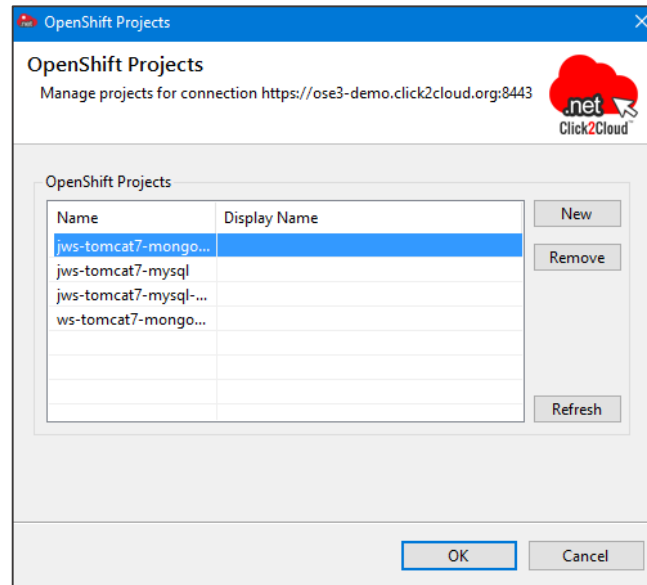
## Manage Projects

Using Manage Projects form, you can manage projects in OpenShift 3 cloud. You can create new Project or delete project just form Visual Studio IDE. You can follow below instructions to manage projects on OpenShift.

1. From the **New OpenShift Application** wizard, select **Manage Projects** link or from **OpenShift Explorer** tool window, select **Manage Projects** option for an existing **OpenShift Connection**.



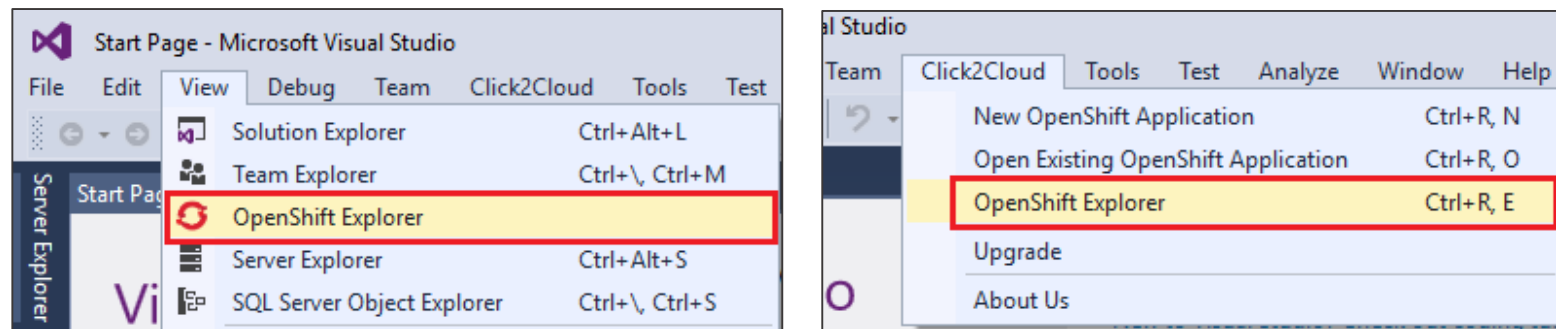
2. The **OpenShift Projects** forms is open from where you can create new project or delete existing projects.



## OpenShift Explorer - The embedded OpenShift Console for Visual Studio

The OpenShift Explorer window lets you connect to OpenShift 3 cloud, create new applications, manage projects, and execute action such as Port-Forwarding and Tail Files (Log Streaming). Specific actions are available, depending on the version of the OpenShift instance you're connected to.

1. In order to launch OpenShift Explorer window, select **Click2Cloud** menu from Visual Studio Menu bar or select **View -> OpenShift Explorer**.



## Deploy Application Changes

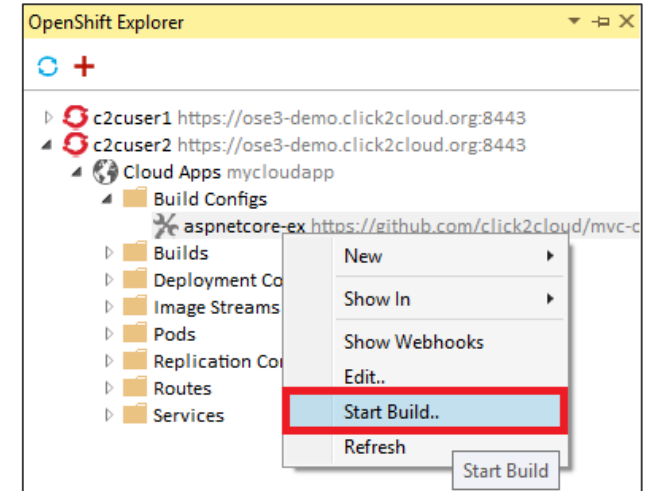
OpenShift 3 supports deploying application changes either manually or using the GitHub Webhooks triggers to build it automatically. Building applications automatically makes the build process quicker. Also, if set to build automatically, the application build process is triggered every time a change is made to the application source code without you having to remember to build it manually.

### Deploy Application Build (Manually)

To build an application manually:

1. In the **OpenShift Explorer** window, expand the **Connection**, then expand **Project**
2. Under **Project**, expand **Build Configs** folder.

3. Right click the **Application Build Config** for which you wish to start new build and click **Start Build**.
4. This will start new build for selected application. To view status of build process, expand **Builds** folder which shows the status of the build.

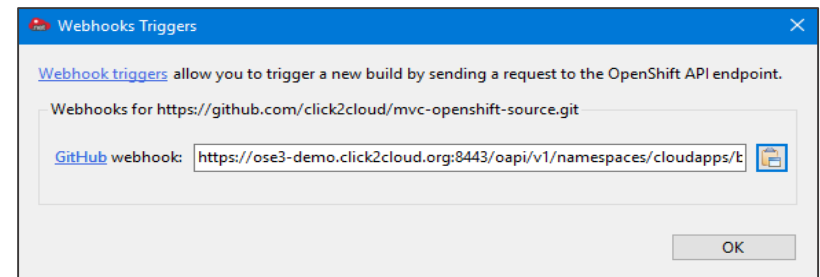


## Deploy Application Build (Automatically Using Git Webhooks)

Webhook trigger facilitates you to configure automatic build deployment. Using this feature you can configure your source code repository to trigger build deployment automatic in OpenShift when you commit any code change in source code.

To get a Webhook:

1. In the **OpenShift Explorer** window, expand the **Connection** and then expand **Project**.
2. Under **Project**, expand **Build Configs** folder.
3. Right click the **Build Configs** resource and then click **Show Webhooks**.
4. Copy the webhook in the **GitHub webhook** field.



**Note:** To configure Webhook on GitHub Repository, follow instructions specified [here](#).

## Log Streaming

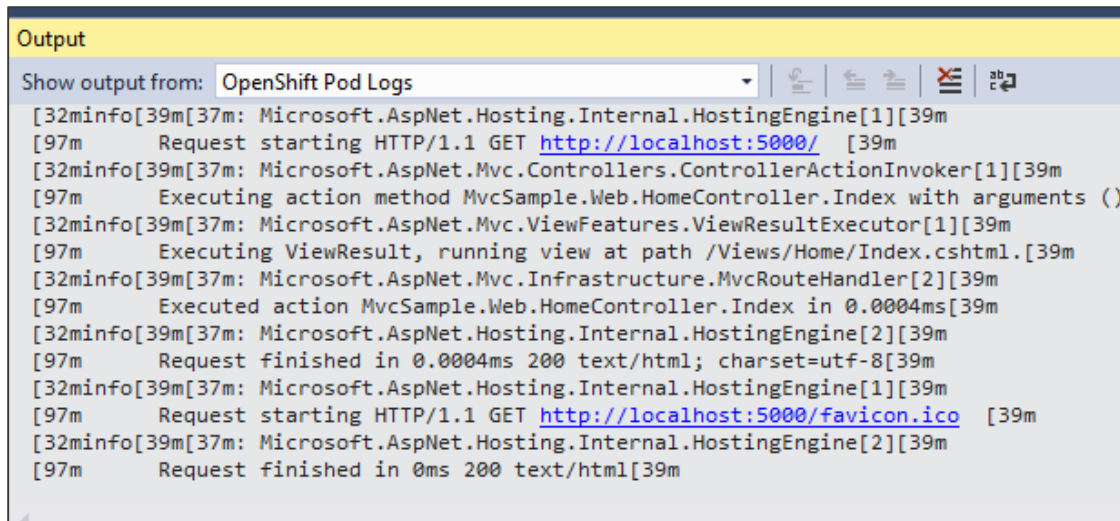
**Prerequisites:** The OpenShift CLI must be installed in order to Log Streaming Feature. Follow instructions specified in [Installing OpenShift 3 CLI](#).

### Stream Application Logs

Application logs are general logs for an application running on OpenShift 3 instance. The streaming application logs feature in OpenShift tool is used to monitor applications and use the historical information to troubleshoot if the application fails or returns errors.

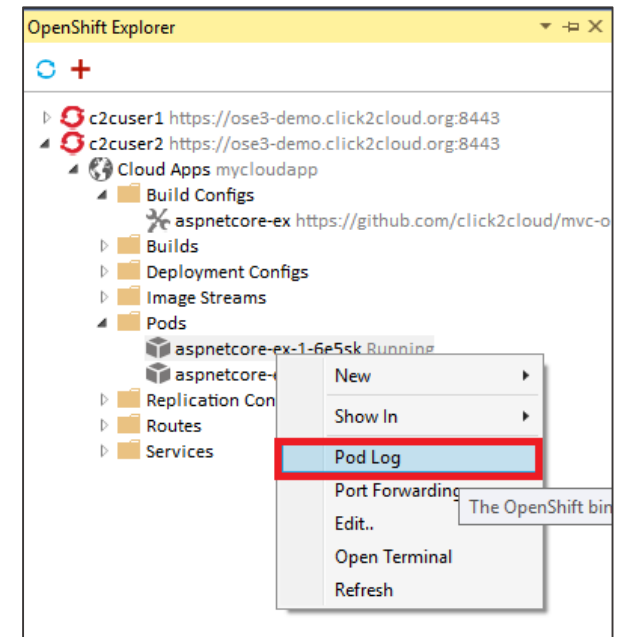
To stream the application logs:

1. In the **OpenShift Explorer** window, expand the **Connection** and then expand **Project**.
2. Under **Project**, expand **Pods** resource folder.
3. Right click the relevant pod and then click **Pod Log**.
4. Output Window displays the pod log.



```

Output
Show output from: OpenShift Pod Logs
[32minfo[39m[37m: Microsoft.AspNetCore.Hosting.Internal.HostingEngine[1][39m
[97m   Request starting HTTP/1.1 GET http://localhost:5000/ [39m
[32minfo[39m[37m: Microsoft.AspNetCore.Mvc.Controllers.ControllerActionInvoker[1][39m
[97m   Executing action method MvcSample.Web.HomeController.Index with arguments ()
[32minfo[39m[37m: Microsoft.AspNetCore.Mvc.ViewFeatures.ViewResultExecutor[1][39m
[97m   Executing ViewResult, running view at path /Views/Home/Index.cshtml.[39m
[32minfo[39m[37m: Microsoft.AspNetCore.Mvc.Infrastructure.MvcRouteHandler[2][39m
[97m   Executed action MvcSample.Web.HomeController.Index in 0.0004ms[39m
[32minfo[39m[37m: Microsoft.AspNetCore.Hosting.Internal.HostingEngine[2][39m
[97m   Request finished in 0.0004ms 200 text/html; charset=utf-8[39m
[32minfo[39m[37m: Microsoft.AspNetCore.Hosting.Internal.HostingEngine[1][39m
[97m   Request starting HTTP/1.1 GET http://localhost:5000/favicon.ico [39m
[32minfo[39m[37m: Microsoft.AspNetCore.Hosting.Internal.HostingEngine[2][39m
[97m   Request finished in 0ms 200 text/html[39m
  
```

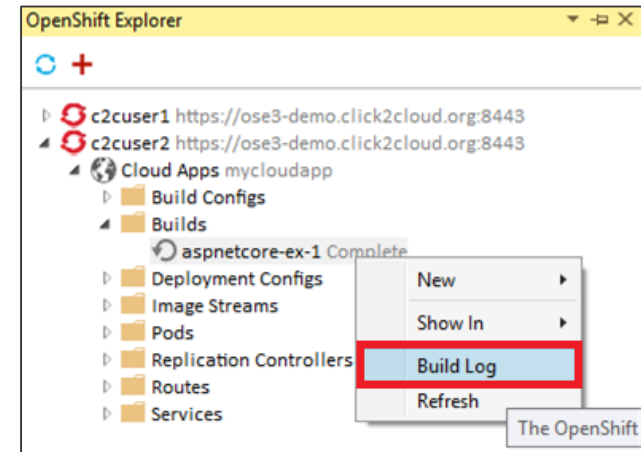


## Stream Build Logs

Build logs are logs documenting changes to applications running on a remote OpenShift 3 instance. The streaming build logs feature in OpenShift tool is used to view the progress of the application build process.

To stream build logs:

1. In the **OpenShift Explorer** window, expand the **Connection** and then expand **Project**.
2. Under **Project**, expand **Builds** resource folder.
3. Right click the relevant build instance and click **Build Log**.
4. Output window displays the build log. Logs display in an Output window, allows users to monitor builds or application server logs in real time.



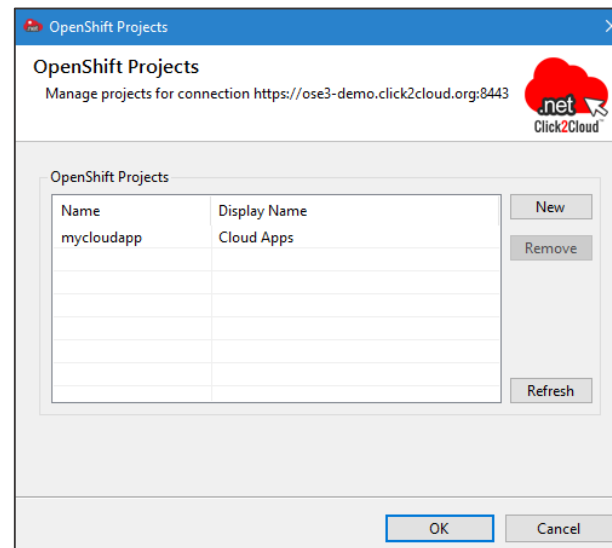
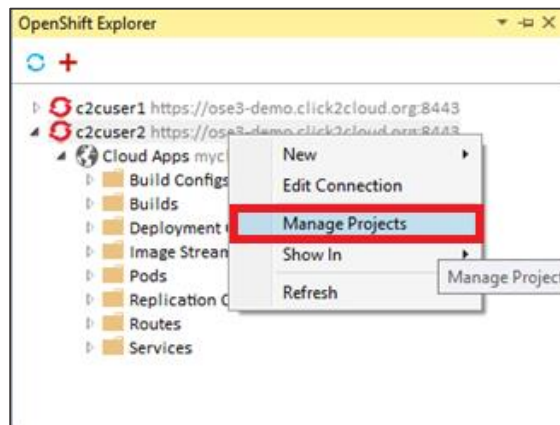
```

Output
Show output from: OpenShift Build Logs
Installed:
74 package(s) to /opt/app-root/src/.dnx/packages
[1m[INFO] Packages restored successfully...[0m
I0325 03:24:41.289271      1 docker.go:481] Container wait returns with 0 and <nil>
I0325 03:24:41.289328      1 docker.go:488] Container exited
I0325 03:24:41.289342      1 docker.go:571] Invoking postExecution function
I0325 03:24:41.289385      1 sti.go:270] No .sti/environment provided (no environment file found in application sources)
I0325 03:24:41.624892      1 docker.go:606] Committing container with config: {Hostname: Domainname: User:1001 Memory:0 MemorySwap:0 CPUS
I0325 03:25:56.907591      1 sti.go:315] Successfully built 172.30.43.26:5000/mycloudapp/aspnetcore-ex:latest
I0325 03:25:57.308739      1 cleanup.go:23] Removing temporary directory /tmp/s2i-build469912296
I0325 03:25:57.308814      1 fs.go:117] Removing directory '/tmp/s2i-build469912296'
I0325 03:25:57.321882      1 sti.go:214] Using provided push secret for pushing 172.30.43.26:5000/mycloudapp/aspnetcore-ex:latest image
I0325 03:25:57.321930      1 sti.go:218] Pushing 172.30.43.26:5000/mycloudapp/aspnetcore-ex:latest image ...
I0325 03:29:40.103957      1 sti.go:234] Successfully pushed 172.30.43.26:5000/mycloudapp/aspnetcore-ex:latest
  
```

## Manage an OpenShift Project

Using OpenShift Explorer, you can also manage your projects on OpenShift cloud. To launch Manage Project form, follow below steps.

1. In the **OpenShift Explorer** tool window, right click on **Connection**.
2. Select the **Manage Projects**.
3. This will launch **OpenShift Projects** form as shown below.



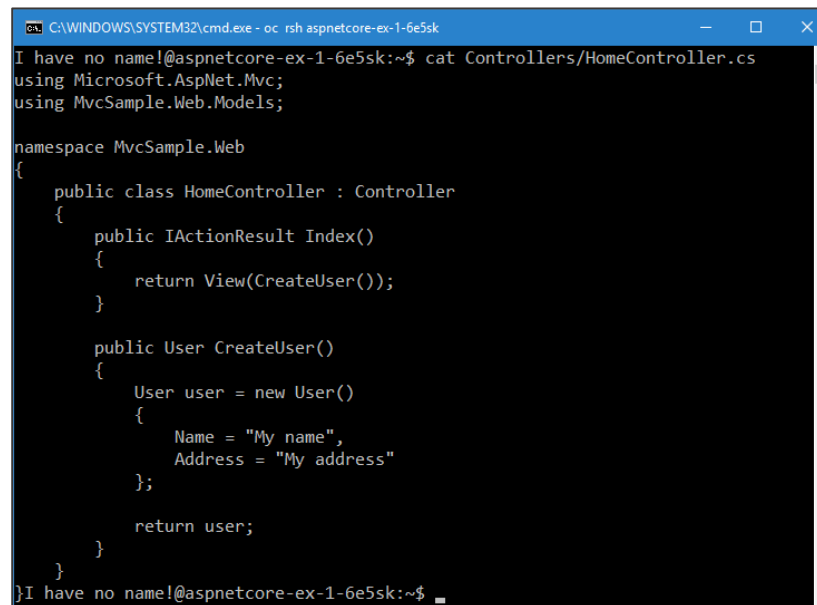
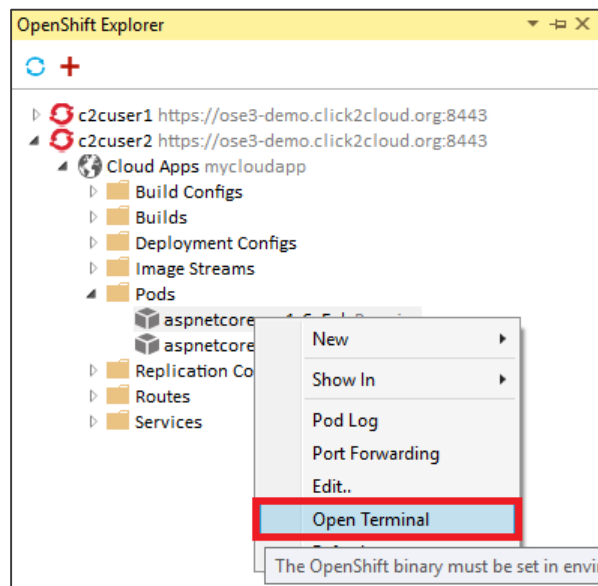
## Open Terminal

Using OpenShift Explorer, you can also connect to terminal running an application.

**Prerequisites:** The OpenShift CLI must be installed in order to Launch Terminal Feature. Follow instructions specified in [Installing OpenShift 3 CLI](#).

To Open Terminal:

1. In the **OpenShift Explorer** window, expand the **Connection** and then expand **Project**.
2. Expand **Pods** and right click on pod which is in **running** state.
3. This will launch container in terminal.
4. You can now run commands in to you container.



```
C:\WINDOWS\SYSTEM32\cmd.exe - oc rsh aspnetcore-ex-1-6e5sk
I have no name!@aspnetcore-ex-1-6e5sk:~$ cat Controllers/HomeController.cs
using Microsoft.AspNet.Mvc;
using MvcSample.Web.Models;

namespace MvcSample.Web
{
    public class HomeController : Controller
    {
        public IActionResult Index()
        {
            return View(CreateUser());
        }

        public User CreateUser()
        {
            User user = new User()
            {
                Name = "My name",
                Address = "My address"
            };

            return user;
        }
    }
}
I have no name!@aspnetcore-ex-1-6e5sk:~$
```



## Port-forwarding

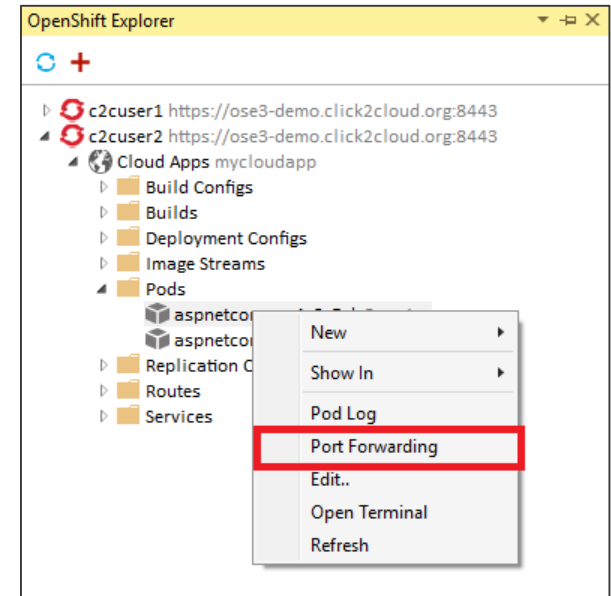
Port-forwarding lets you bind ports on your local machine to ports on remote OpenShift services, using secure communications. You can then connect to local port for data access or application debugging.

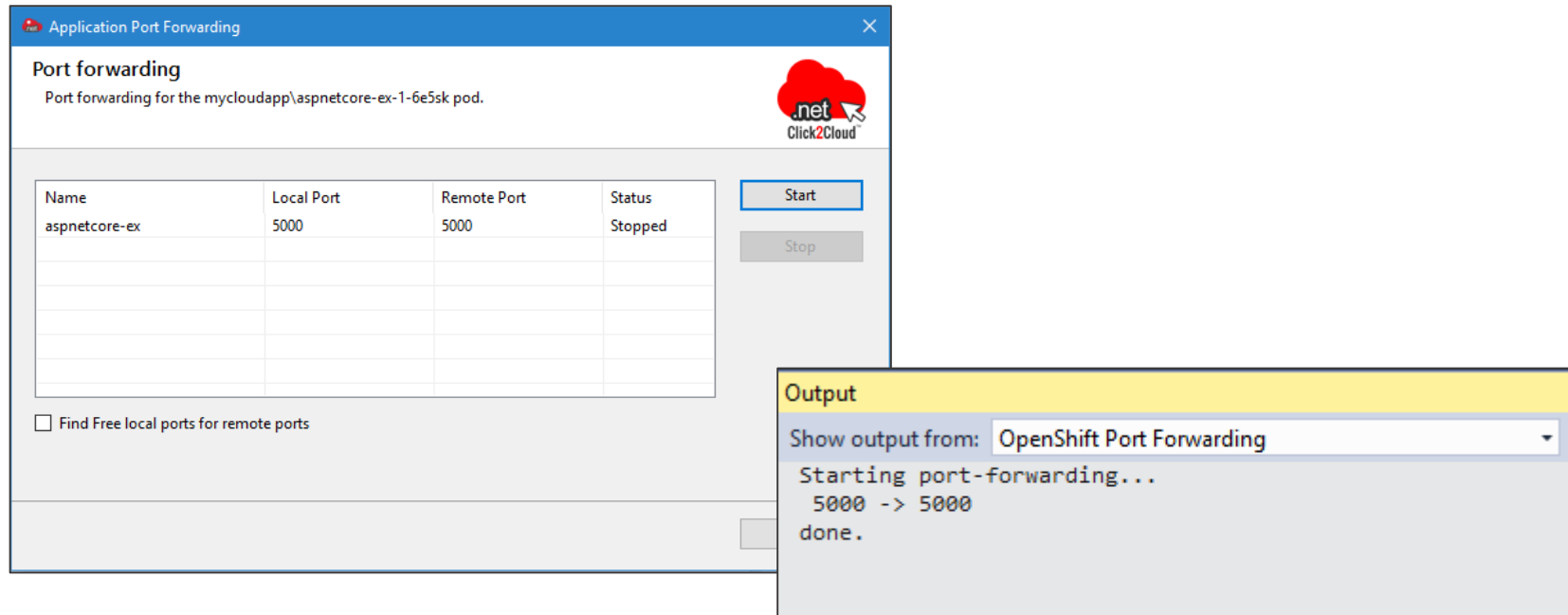
Port forwarding automatically stops when the OpenShift connection terminates, or when the IDE is closed.

**Prerequisites:** The OpenShift CLI must be installed in order to use Port forwarding Feature. Follow instructions specified in [Installing OpenShift 3 CLI](#).

To forward port:

1. In the **OpenShift Explorer** window, expand the **Connection** and then expand **Project**.
2. Expand **Pods** and right click on pod which is in **running** state.
3. Select the **Port Forwarding**.
4. In the **Application Port Forwarding** window, click the **Find free local ports for remote ports** check box to port application on available port or got with default value
5. Now click on **Start** button.





The screenshot shows the 'Application Port Forwarding' dialog box. The title bar reads 'Application Port Forwarding'. The main heading is 'Port forwarding' with a subtitle 'Port forwarding for the mycloudapp\aspnetcore-ex-1-6e5sk pod.' and the Click2Cloud logo. A table lists the port forwarding configuration:

Name	Local Port	Remote Port	Status
aspnetcore-ex	5000	5000	Stopped

Below the table is a checkbox labeled 'Find Free local ports for remote ports'. To the right of the table are 'Start' and 'Stop' buttons. An 'Output' window is overlaid on the bottom right, showing the following text:

```
Show output from: OpenShift Port Forwarding
Starting port-forwarding...
5000 -> 5000
done.
```

6. Once port forwarding is done, you can now access your application from Localhost.

## Version History

Version	Date	Author Name	Description of Change
V1.0	14-03-2016	Jogesh	
V1.0	26-03-2016	Shrutika M	

## Reviewers

Reviewer Name	Version Approved	Role	Date

Click2Cloud



# About Click2Cloud

Use our solutions and services to

Click2Cloud Inc. is a Red Hat ISV and Advanced Partner Company, and also a Microsoft Gold Partner for Content & Collaboration, based in Bellevue, WA

To learn more via our FAQ's, instructional videos, and more go to [www.Click2Cloud.net](http://www.Click2Cloud.net)

At Click2Cloud, we believe in empowering the developer and giving you choices. We help you get to the cloud to be more agile, deliver faster, unleash innovation and become a modern digital enterprise. This is why we have developed a true DevOps experience allowing you to deploy containers, images and applications to OpenShift with one-click. It is why we have developed a way to run Windows applications within OpenShift, giving you new choices and opening the door to new possibilities. These solutions accelerate your journey to the cloud and allow you to embrace the transformative power of new technologies.

## Click2Cloud Headquarters

Suite 235 G, Impact Hub Bellevue, 555, 116th Avenue, NE,  
Bellevue 98004 , WA, USA

Phone 1: +1-425-748-9666

Email: [contact@click2cloud.net](mailto:contact@click2cloud.net)

Phone 2: +1-425-749-7495

Website: <http://click2cloud.net>