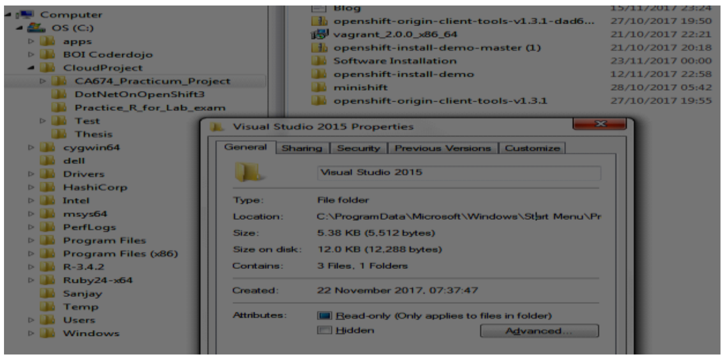
OpenShift Blog for CA674 Cloud Architecture Project

**22/11/2017 – Sanjay**

After analysing Open Shift add-on which was downloaded for Visual Studio 2017, discovered that Click2Cloud add-on does not support and no template was available for asp.net 4.5. Hence downloaded Visual Studio 2015.



```
Computer
  OS (C:)
    apps
    BOI Coderdojo
    CloudProject
      CA674_Practicum_Project
      DotNetOnOpenShift3
      Practice_R_for_Lab_exam
      Test
      Thesis
    cygwin64
    dell
    Drivers
    HashiCorp
    Intel
    msys64
    PerfLogs
    Program Files
    Program Files (x86)
    R-3.4.2
    Ruby24-x64
    Sanjay
    Temp
    Users
    Windows

Blog                                     15/11/2017 23:24
openshift-origin-client-tools-v1.3.1-dad6...  27/10/2017 19:50
vagrant_2.0.0_x86_64                     21/10/2017 22:21
openshift-install-demo-master (1)        21/10/2017 20:18
Software Installation                    23/11/2017 00:00
openshift-install-demo                   12/11/2017 22:58
minishift                                28/10/2017 05:42
openshift-origin-client-tools-v1.3.1     27/10/2017 19:55
```

Visual Studio 2015 Properties

General | Sharing | Security | Previous Versions | Customize

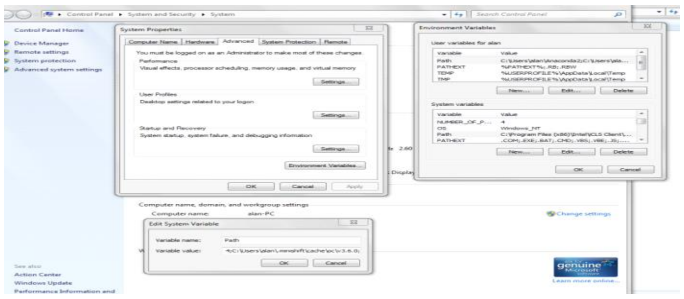
Visual Studio 2015

- Type: File folder
- Location: C:\ProgramData\Microsoft\Windows\Start Menu\Pr
- Size: 5.38 KB (5,512 bytes)
- Size on disk: 12.0 KB (12,288 bytes)
- Contains: 3 Files, 1 Folders
- Created: 22 November 2017, 07:37:47
- Attributes: Read-only (Only applies to files in folder); Hidden

Using Visual Studio Community edition 2015, first tried to launch FacApi website in local laptop and follow step by step process to download Click2Cloud add-on and additional software.

Pre-Requisite Software's

- Install Microsoftware Visual Studio 2015 – Already downloaded and launched FacApi
- Git for Windows
- Open Shift Command Line Interface (CLI)
- MicroSoft .Net Core SDK Preview 2
- Setup environment path to access Open Shift from Command line & Open Shift Command Line Interface (CLI)



Control Panel ▸ System and Security ▸ System

System Properties — Advanced

You must be logged on as an Administrator to make most of these changes.

- Performance: Visual effects, processor scheduling, memory usage, and virtual memory
- User Profiles: Desktop settings related to your logon
- Startup and Recovery: System startup, system failure, and debugging information

Environment Variables

User variables for alan

| Variable | Value |
|---|---|
| Path | C:\Users\alan\Anaconda2;C:\Users\ala... |
| PATHEXT | %PATHEXT%;.RB;.RBW |
| TEMP | %USERPROFILE%\AppData\Local\Temp |
| TMP | %USERPROFILE%\AppData\Local\Temp |

System variables

| Variable | Value |
|---|---|
| NUMBER_OF_P... | 4 |
| OS | Windows_NT |
| Path | C:\Program Files (x86)\Intel\iCLS Client\... |
| PATHEXT | .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;... |

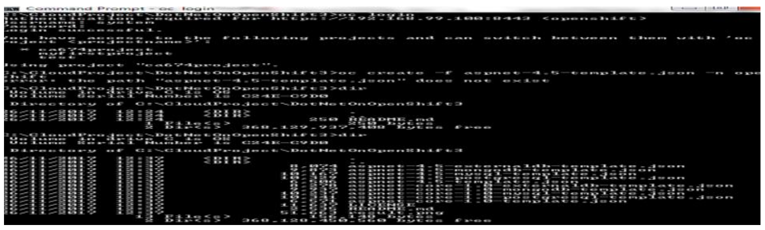
Computer name, domain, and workgroup settings

Computer name: alan-PC

Edit System Variable

- Variable name: Path
- Variable value: ;C:\Users\alan\.minishift\cache\oc\v3.6.0;

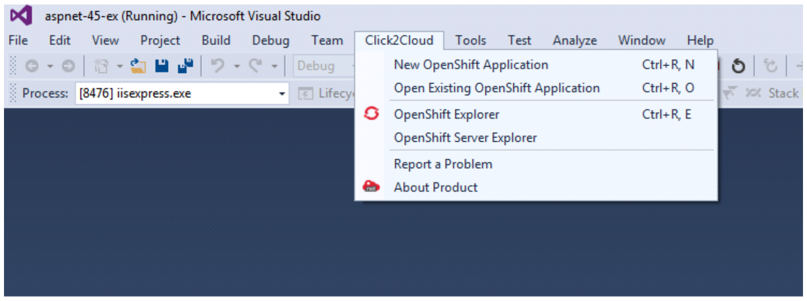
Open Shit Command line login screenshot



```
C:\CloudProject\DotNetOnOpenShift3>oc login
Authentication required for https://192.168.99.100:8443 (openshift)
Username: system
Password:
Login successful.

You have access to the following projects and can switch between them with 'oc
project <projectname>':

  * ca674project
    myfirstproject
    test

Using project "ca674project".

C:\CloudProject\DotNetOnOpenShift3>oc create -f aspnet-4.5-template.json -n ope
Error: the path "aspnet-4.5-template.json" does not exist

C:\CloudProject\DotNetOnOpenShift3>dir
 Volume in drive C is OS
 Volume Serial Number is C24E-C9D0

 Directory of C:\CloudProject\DotNetOnOpenShift3

26/11/2017  18:24    <DIR>          .
26/11/2017  18:24    <DIR>          ..
               1 File(s)            250 README.md
               2 Dir(s)  368,129,937,408 bytes free

C:\CloudProject\DotNetOnOpenShift3>dir
 Volume in drive C is OS
 Volume Serial Number is C24E-C9D0

 Directory of C:\CloudProject\DotNetOnOpenShift3

[illegible]
               2 Dir(s)  368,120,[illegible] bytes free
```

**26/11/2017 – Sanjay**

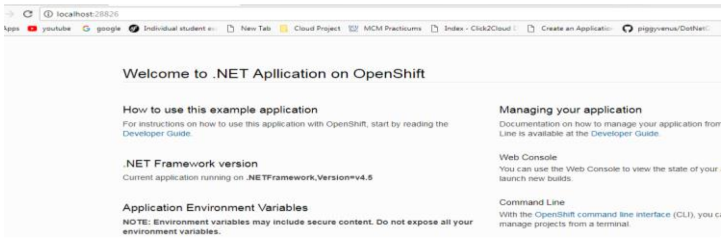
After downloading all Pre-requisite software, created first asp application using Custom template and followed below steps


aspnet-45-ex (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Team Click2Cloud Tools Test Analyze Window Help
Process: [8476] iisexpress.exe
New OpenShift Application Ctrl+R, N
Open Existing OpenShift Application Ctrl+R, O
OpenShift Explorer Ctrl+R, E
OpenShift Server Explorer
Report a Problem
About Product

Created project in connected server by selecting "New OpenShift Application" and provided new project and other details and click Finish button.

Select template using template wizard.

Open Shift Command Line Interface (CLI) and downloaded Click2Cloud add-on for asp.net 4.5 templates and select Use a template from my local file system. Then click on Define button using Template details dialog box from Open Shift resources and click next or finish to create with default template parameters and labels. Below are the screenshots



localhost:28826

Welcome to .NET Apllication on OpenShift

How to use this example application
For instructions on how to use this application with OpenShift, start by reading the Developer Guide.

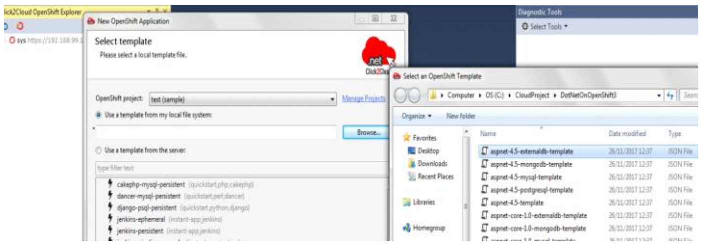
.NET Framework version
Current application running on **.NETFramework,Version=v4.5**

Application Environment Variables
**NOTE: Environment variables may include secure content. Do not expose all your environment variables.**

Managing your application
Documentation on how to manage your application from
Line is available at the Developer Guide.

Web Console
You can use the Web Console to view the state of your
launch new builds.

Command Line
With the OpenShift command line interface (CLI), you c
manage projects from a terminal.



New OpenShift Application

Select template
Please select a local template file.

OpenShift project: test (sample)
● Use a template from my local file system:
○ Use a template from the server:

cakephp-mysql-persistent (quickstart,php,cakephp)
dancer-mysql-persistent (quickstart,perl,dancer)
django-psql-persistent (quickstart,python,django)
jenkins-ephemeral (instant-app,jenkins)
jenkins-persistent (instant-app,jenkins)

Select an OpenShift Template
Computer ▸ OS (C:) ▸ CloudProject ▸ DotNetOnOpenShift3

| Name | Date modified | Type |
|---|---|---|
| aspnet-4.5-externaldb-template | 26/11/2017 12:37 | JSON File |
| aspnet-4.5-mongodb-template | 26/11/2017 12:37 | JSON File |
| aspnet-4.5-mysql-template | 26/11/2017 12:37 | JSON File |
| aspnet-4.5-postgresql-template | 26/11/2017 12:37 | JSON File |
| aspnet-4.5-template | 26/11/2017 12:37 | JSON File |
| aspnet-core-1.0-externaldb-template | 26/11/2017 12:37 | JSON File |
| aspnet-core-1.0-mongodb-template | 26/11/2017 12:37 | JSON File |

**Sanjay 30/11**

Working with Open Shift and .NET Core combo, i face many challenges which i want to document as part of Cloud Project learning exercise.

Below are high level steps to create a sample web page using Visual Studio 2017 using .Net Core2.0 framework and clone to GitHub repository to deploy code in Open Shift.

Prerequisites - Before creating Sample webpage using .NETCORE, download Visual Studio 2017 community edition free from - https://www.visualstudio.com/downloads/

Create account in GitHub, Open Shift

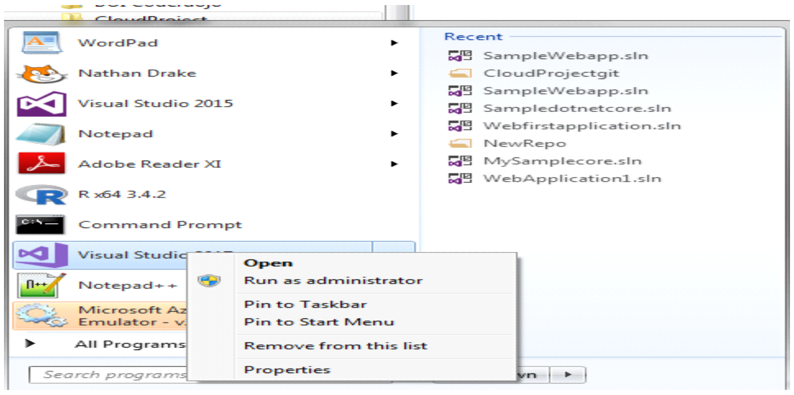
Step1: Getting Stated in Visual Studio 2017 Community edition

Create a folder structure in local C:drive of a file system to store Web site project.

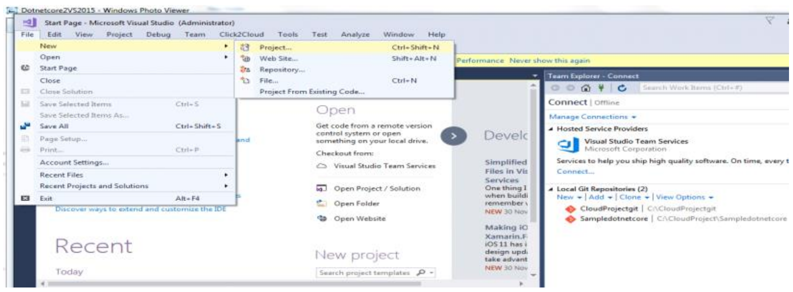
C:\SampleWebapp

To create a file system Website using GUI

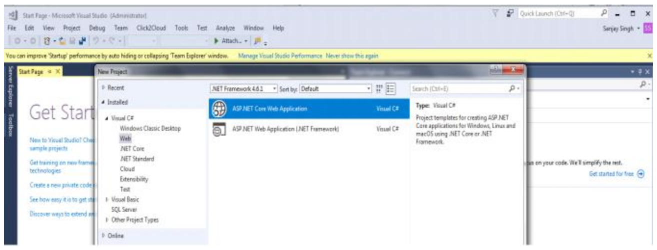
a) Open Visual Studio 2017 from Start menu as an administrator



b) On the File menu, Click New Web Site

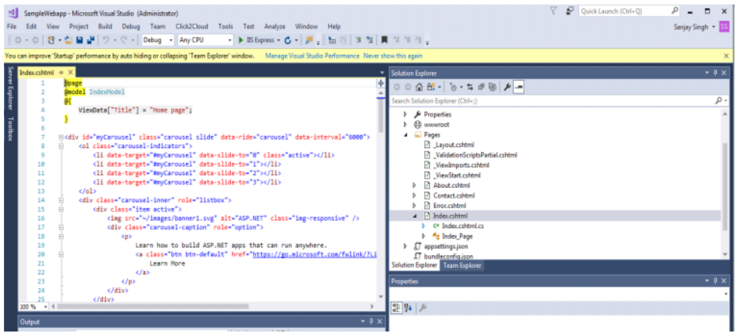


The New Web Site dialog box appears and under Installed, Click Web and ASP.NET Core Web Application Visual C#



Step2: Editing Data in ASP.NET

In Solution Explorer, edit Inex.cshtml under Pages and change code



```
@page
@model IndexModel
@{
    ViewData["Title"] = "Home page";
}

<div id="myCarousel" class="carousel slide" data-ride="carousel" data-interval="6000">
    <ol class="carousel-indicators">
        <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
        <li data-target="#myCarousel" data-slide-to="1"></li>
        <li data-target="#myCarousel" data-slide-to="2"></li>
        <li data-target="#myCarousel" data-slide-to="3"></li>
    </ol>
    <div class="carousel-inner" role="listbox">
        <div class="item active">
            <img src="~/images/banner1.svg" alt="ASP.NET" class="img-responsive" />
            <div class="carousel-caption" role="option">
                <p>
                    Learn how to build ASP.NET apps that can run anywhere.
                    <a class="btn btn-default" href="https://go.microsoft.com/fwlink/?Li
                        Learn More
                    </a>
                </p>
            </div>
        </div>
```

Also Team Explorer setup folder location and clone and commit change.

**Challenges: -** While trying to launch the sample website, getting environmental error - problem in loading page asp.net http://localhost:61878/ and IIS server.

After analysis and google search got some understanding about the error and how to fix the problem. The problem was related to environment so after deleting file from the .vs folder and restarted Visual Studio
https://developercommunity.visualstudio.com/content/problem/24939/iis-express-not-working.html

Before updating setup wizard, create new repo in GitHub https://github.com/Singhsk/MySamplecore.git

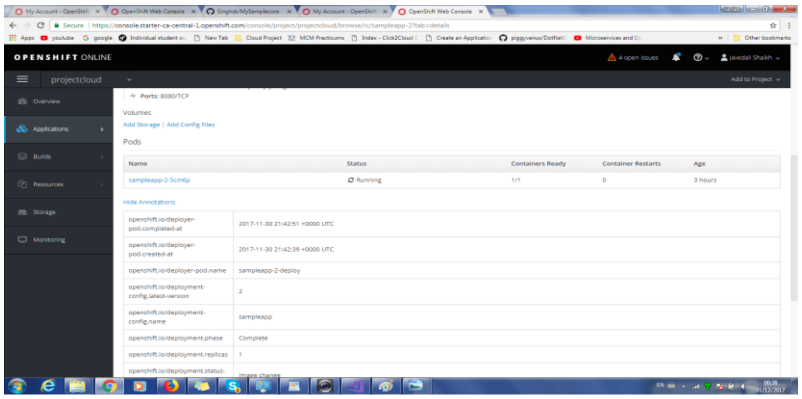
Step3: Running Sample ASP.NET Core on Open Shift

Login to Open Shift and create project by selecting options from dropdown menu
https://manage.openshift.com/

Under Project there are following Options for Build and Configuration:

1) Overview, 2) Application 3) Builds 4) Resources 5) Storage 6) Monitoring

Pods are the rough equivalent of a machine instance (physical or virtual) to a container. Each pod is allocated its own internal IP address, therefore owning its entire port space, and containers within pods can share their local storage and networking.



Ports: 8080/TCP

Volumes

Add Storage | Add Config Files

Pods

| Name | Status | Containers Ready | Container Restarts | Age |
|---|---|---|---|---|
| sampleapp-2-5cm6p | Running | 1/1 | 0 | 3 hours |

Hide Annotations

| | |
|---|---|
| openshift.io/deployer-pod.completed-at | 2017-11-30 21:42:51 +0000 UTC |
| openshift.io/deployer-pod.created-at | 2017-11-30 21:42:39 +0000 UTC |
| openshift.io/deployer-pod.name | sampleapp-2-deploy |
| openshift.io/deployment-config.latest-version | 2 |
| openshift.io/deployment-config.name | sampleapp |
| openshift.io/deployment.phase | Complete |
| openshift.io/deployment.replicas | 1 |
| openshift.io/deployment.status- | image change |

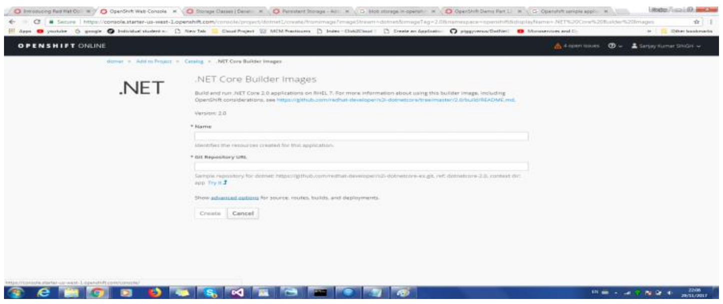
Ports: 8080/TCP

Challenges: - Creating account and setting project was very easy and sample process.

Step4: Deploy Code in Open Shift using GitHub repo

Once the project is build successfully, syn code using GibHub repo - https://github.com/Singhsk/MySamplecore.git

.Net Core Build image


.NET Core Builder Images

Build and run .NET Core 2.0 applications on RHEL 7. For more information about using this builder image, including OpenShift considerations, see https://github.com/redhat-developer/s2i-dotnetcore/tree/master/2.0/build/README.md.

Version: 2.0

* Name

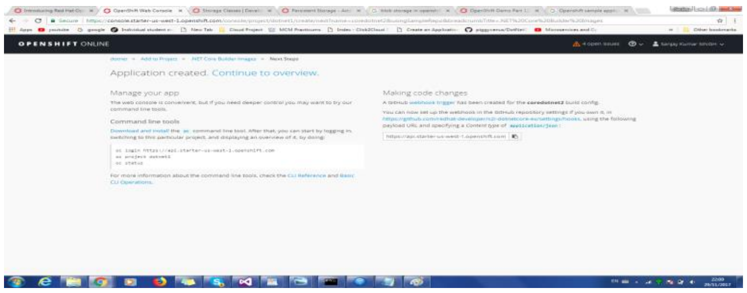
Identifies the resources created for this application.

* Git Repository URL

Sample repository for dotnet: https://github.com/redhat-developer/s2i-dotnetcore-ex.git, ref: dotnetcore-2.0, context dir: app Try it

Show advanced options for source, routes, builds, and deployments.

Enter image name and GitHub path to copy image and  hit Create, next screen will be presented - Application Created, continue to Overview , Command Line tools, path, login user details etc


Application created. Continue to overview.

Manage your app

The web console is convenient, but if you need deeper control you may want to try our command line tools.

Command line tools

Download and install the oc command line tool. After that, you can start by logging in, switching to this particular project, and displaying an overview of it, by doing:

```
oc login https://api.starter-us-west-1.openshift.com
oc project dotnet1
oc status
```

For more information about the command line tools, check the CLI Reference and Basic CLI Operations.

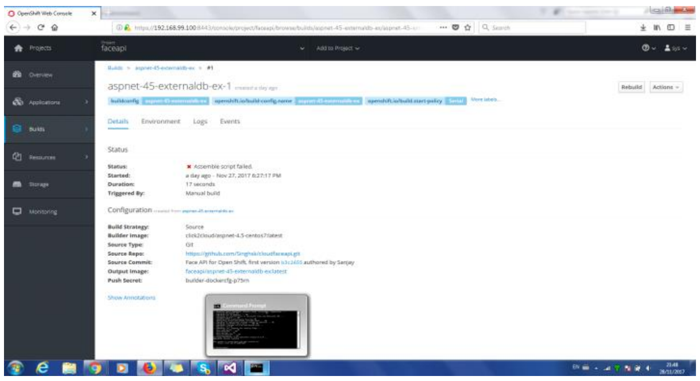
Making code changes

A GitHub webhook trigger has been created for the coredotnet2 build config.

You can now set up the webhook in the GitHub repository settings if you own it, in https://github.com/redhat-developer/s2i-dotnetcore-ex/settings/hooks, using the following payload URL and specifying a Content type of application/json:

https://api.starter-us-west-1.openshift.com

From next screen on top right corner select build. Initially there were some issues in open shift online so it was taking time to complete the build process and finally failed with build error.


aspnet-45-externaldb-ex-1

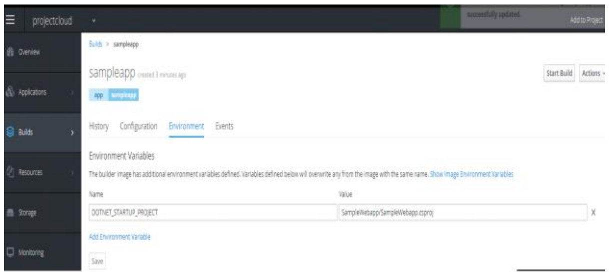
Status

Status: Assemble script failed.
Started: a day ago - Nov 27, 2017 6:27:17 PM
Duration: 17 seconds
Triggered By: Manual build

Configuration

Build Strategy: Source
Builder Image: click2cloud/aspnet-4.5-centos7:latest
Source Type: Git
Source Repo: https://github.com/Singhsk/cloudfaceapi.git
Source Commit: Face API for Open Shift, first version b3c1655 authored by Sanjay
Output Image: faceapi/aspnet-45-externaldb-ex:latest
Push Secret: builder-dockercfg-p75m

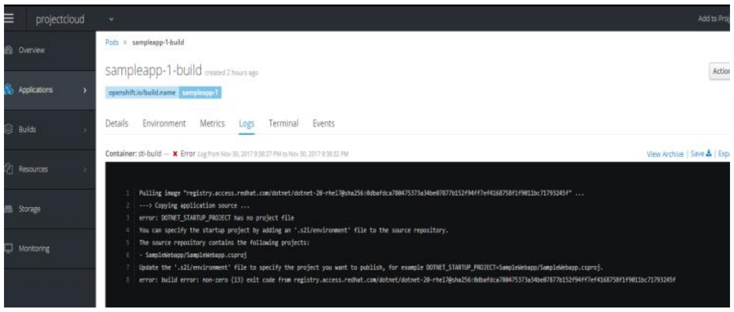
OpenShift has user friendly option to view the log to more details about the error. After going the log, noticed that there were some initial configuration were missing.
From the log file copy the missing path and added missing variable and the value in Build - Environment



The builder image has additional environment variables defined. Variables defined below will overwrite any from the image with the same name. Show Image Environment Variables

| Name | Value |
|---|---|
| DOTNET_STARTUP_PROJECT | SampleWebapp/SampleWebapp.csproj |

**Challenges :** Deploying code in OpenShift is very error, using user friendly menu option, user can able to build application but prior to build there are some variable which needs to be setup in advance to avoid error. E.g



Container: sti-build — ✕ Error Log from Nov 30, 2017 9:38:27 PM to Nov 30, 2017 9:38:32 PM

```
Pulling image "registry.access.redhat.com/dotnet/dotnet-20-rhel7@sha256:0dbafdca780475373a34be07877b152f94ff7ef4168758f1f9811bc71793245f" ...
---> Copying application source ...
error: DOTNET_STARTUP_PROJECT has no project file
You can specify the startup project by adding an '.s2i/environment' file to the source repository.
The source repository contains the following projects:
- SampleWebapp/SampleWebapp.csproj
Update the '.s2i/environment' file to specify the project you want to publish, for example DOTNET_STARTUP_PROJECT=SampleWebapp/SampleWebapp.csproj.
error: build error: non-zero (13) exit code from registry.access.redhat.com/dotnet/dotnet-20-rhel7@sha256:0dbafdca780475373a34be07877b152f94ff7ef4168758f1f9811bc71793245f
```

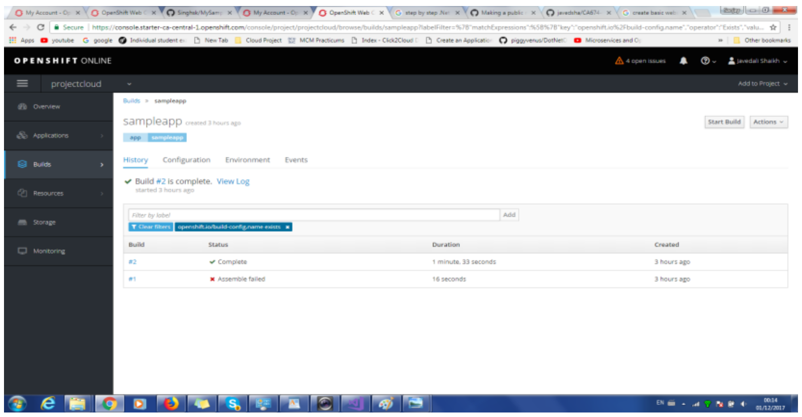
While going thru the build issue, under Application menu there were other very important details which can't be ignored as part of standard container free spec build e.g     Memory , CPU,
Network usage for each Pod.

In Details Option, Node and ip address etc and on the right side Container and image details, CPU and Memory. Using Action button at the top right, user can delete the build.
Environment tab - Environment form will let you to add all key value pair for the Config map or secret as environment variable.

There is another very important option to calculate and compute resources this will give the amount of resources to build application after each Pod

After adding environment variable from the error log in Environment tab, Rebuild the imager and it completed successfully.

Environment Variables added
Name = DOTNET_STARTUP_PROJECT
Value = SampleWebapp/SampleWebapp.csproj



Builds > sampleapp

sampleapp created 3 hours ago

✓ Build #2 is complete. View Log
started 3 hours ago

Filter: openshift.io/build-config.name exists

| Build | Status | Duration | Created |
|---|---|---|---|
| #2 | ✓ Complete | 1 minute, 33 seconds | 3 hours ago |
| #1 | ✗ Assemble failed | 16 seconds | 3 hours ago |

Under Build there are following OPTIONS

History, Configuration, Environment Events, Triggering Builds

Build Triggers - When defining a Build Config, user can define triggers to control the circumstances in which the Build Config should run. The following build triggers are available :

- Webhook Triggers Webhook triggers allows user to trigger a new builds by sending the OpenShift online API endpoint, User can define their triggers using GitHub, GitLab, Bitbucket or Generic webhooks.

- Oc new-app and oc new-build will create GitHub and Generic webhook triggers automatically, but any other needed webhook triggers must be added manually (see Setting Triggers).

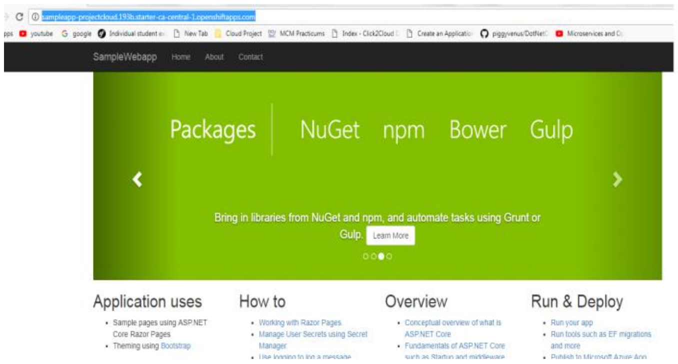
- Image Change Image change triggers allow your build to be automatically invoked when a new version of an upstream image is available. For example, if a build is based on top of a RHEL image, then you can trigger that build to run any time the RHEL image changes. As a result, the application image is always running on the latest RHEL base image.

- Configuration Change A configuration change trigger allows a build to be automatically invoked as soon as a new BuildConfig is created.

Launch Web application using OpenShift Router - An OpenShift Online route exposes a service at a host name, like www.example.com, so that external clients can reach it by name.

http://sampleapp-projectcloud.193b.starter-ca-central-1.openshiftapps.com/

Screenshot


sampleapp-projectcloud.193b.starter-ca-central-1.openshiftapps.com
SampleWebapp
Home
About
Contact
Packages
NuGet
npm
Bower
Gulp
Bring in libraries from NuGet and npm, and automate tasks using Grunt or Gulp.
Learn More
Application uses
Sample pages using ASP.NET Core Razor Pages
Theming using Bootstrap
How to
Working with Razor Pages
Manage User Secrets using Secret Manager
Overview
Conceptual overview of what is ASP.NET Core
Fundamentals of ASP.NET Core
Run & Deploy
Run your app
Run tools such as EF migrations and more

Finally I could able to successfully launch sample ASP.NET Core2.0 website in OpenShift.

**Next Step:** Explore OpenShift auto scale option and try to add code to experiment how auto scaling setup works.

Explore Storage and collect server matrix, Monitoring option in details

**Sanjay 1/12**

Container sampleapp does not have health checks to ensure your application is running correctly. Add Health Checks to project sampleapp

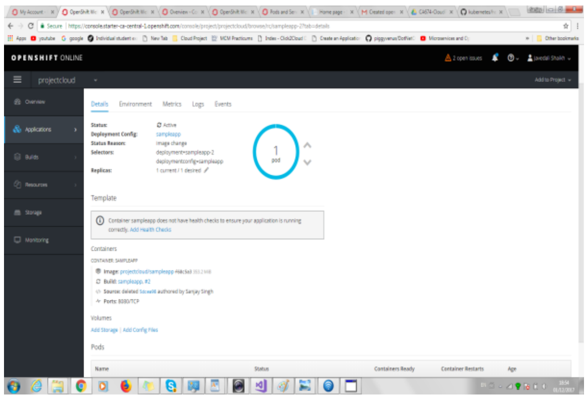
What is Health Checks for sampleapp - Container health is periodically checked using readiness and liveness probes.

Readiness Probe - A readiness probe checks if the container is ready to handle requests. A failed readiness probe means that a container should not receive any traffic from a proxy, even if it's running.

What is Liveness Probe: - A liveness probe checks if the container is still running. If the liveness probe fails, the container is killed.

Pause rollouts for this deployment config

Pausing lets you make changes without triggering a rollout. You can resume rollouts at any time. If unchecked, a new rollout will start on save.



Status: Active
Deployment Config: sampleapp
Status Reason: image change
Selectors: deployment=sampleapp-2
deploymentconfig=sampleapp
Replicas: 1 current / 1 desired

Template

Container sampleapp does not have health checks to ensure your application is running correctly. Add Health Checks

Containers

CONTAINER: SAMPLEAPP
Image: projectcloud/sampleapp 353.2 MiB
Build: sampleapp, #2
Source: deleted authored by Sanjay Singh
Ports: 8080/TCP

Volumes

Add Storage | Add Config Files

Pods

| Name | Status | Containers Ready | Container Restarts | Age |
|---|---|---|---|---|

**Sanjay 1/12**

**OpenShift Auto Scaling Features**

What is Pod

OpenShift Online leverages the Kubernetes concept of a pod, which is one or more containers deployed together on one host, and the smallest compute unit that can be defined, deployed, and managed.

Each Pod is the rough equivalent of a machine instances ( Physical or virtual) to a container

Pods have a lifecycle, first they are defined, then assigned to a run on a node and then they run until their contain(s) exist or they are removed.

Each Pod is allocated its own internal IP address, therefore owing its entire port space, and containers within pods can share their local storage and networking.

Horizontal pod autoscaler, defined by a **HorizontalPodAutoScaler** object, specifies how the system should automatically increase or decrease the scale of a replicatior controller or deployment config.

Autoscale parameter can be defined using project main menu from Applicatiom -> Deployment -> PROJECT NAME (sampleapp) -> Autoscale

Autoscaler Name = cpuhigh

Min Pods = 1     Max Prods = 2

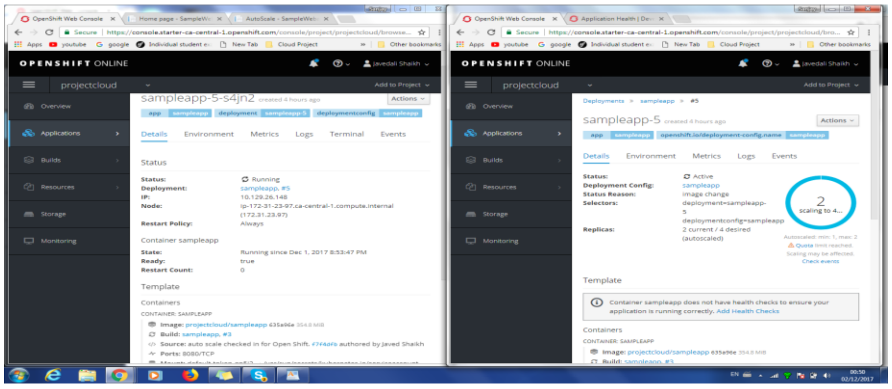
CPU Request Target 70%

The CPU percentage request that each pod should ideally be using. Pods will be added or removed periodically when CPU usage exceeds or drops below target value - 70% - Defaults is 80%

Openshift has a limited option supported by horizontal pod autoscalers

- CPU Utilization     - Memory Utilzation

Autoscaling Screenshot



sampleapp-5-s4jn2 created 4 hours ago

app sampleapp deployment sampleapp-5 deploymentconfig sampleapp

Status

Status: Running
Deployment: sampleapp, #5
IP: 10.129.26.148
Node: ip-172-31-23-97.ca-central-1.compute.internal (172.31.23.97)
Restart Policy: Always

Container sampleapp

State: Running since Dec 1, 2017 8:53:47 PM
Ready: true
Restart Count: 0

Template

Containers

CONTAINER: SAMPLEAPP

Image: projectcloud/sampleapp 635a96e 354.8 MiB
Build: sampleapp, #3
Source: auto scale checked in for Open Shift. f7f4dfb authored by Javed Shaikh
Ports: 8080/TCP

Deployments » sampleapp » #5

sampleapp-5 created 4 hours ago

app sampleapp openshift.io/deployment-config.name sampleapp

Status: Active
Deployment Config: sampleapp
Status Reason: image change
Selectors: deployment=sampleapp-5 deploymentconfig=sampleapp
Replicas: 2 current / 4 desired (autoscaled)

2 scaling to 4...

Autoscaled: min: 1, max: 2
Quota limit reached. Scaling may be affected.
Check events

Template

Container sampleapp does not have health checks to ensure your application is running correctly. Add Health Checks

Containers

CONTAINER: SAMPLEAPP

Image: projectcloud/sampleapp 635a96e 354.8 MiB
Build: sampleapp, #3

00:50 02/12/2017

**Sanjay 2/12**

**Adding Storage steps in Open Shift**

Storage : A PersistentVolume object is a storage resource in an OpenShift Online cluster. Storage is provisioned by your cluster administrator by creating PersistentVolume objects from sources such as GCE Persistent Disk, AWS Elastic Block Store (EBS), and NFS mounts.

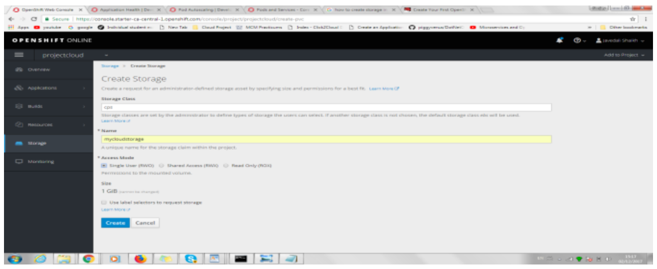
Reference : - https://docs.openshift.com/online/dev_guide/persistent_volumes.html

Creating Storage is not easy using Open Shift onlne GUI. From the project window console (left side) select Storage option and enter Storage Class, Name, Access Mode and Size.

Storage Class - Storage Classes are setup by the admin to define types of storage the users can select. If another storage class is not chosen, the default storage class ebs will be used.
Name - A unique name for the storage class claim within the project.
Access Mode - Set permission to the mounted volume - Single User (RWO), Shared Access (RWX) & Read only (rox).
Size - Limited to 1GB, cannot be changed for free subscription.
Labels : Labels are used to organize, group or select or select objects and resources such as prods.

First time created with Storage class as mycloudstorage failed with PersistentVolumeClaim which means the storage class ebs invalid access mode Read Write Many. For free user subscription the only valid option is Read Write Once (RWO). See below screenshot

After deleting, recreated Storage with ROW option



Storage > Create Storage

Create Storage

Create a request for an administrator-defined storage asset by specifying size and permissions for a best fit. Learn More

Storage Class
cps

Storage classes are set by the administrator to define types of storage the users can select. If another storage class is not chosen, the default storage class ebs will be used.
Learn More

* Name
mycloudstorage

A unique name for the storage claim within the project.

* Access Mode
Single User (RWO) Shared Access (RWX) Read Only (ROX)

Permissions to the mounted volume.

Size
1 GiB (cannot be changed)

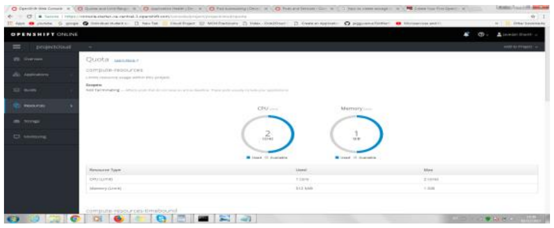
Use label selectors to request storage
Learn More

OpenShift Quotas and Limited Ranges

Using quotas and limit ranges, cluster administrators can set constraints to limit the number of objects or amount of compute resources that are used in your project. This helps cluster administrators better manage and allocate resources across all projects, and ensure that no projects are using more than is appropriate for the cluster size.

As a developer, you can also set requests and limits on compute resources at the pod and container level.

Quotas



| Resource Type | Used | Max |
|---|---|---|
| CPU (Limit) | 1 core | 2 cores |
| Memory (Limit) | 512 MiB | 1 GiB |