An introduction to .NET Core development on OpenShift
Introductions

Andrew Slice
Senior Linux .NET Core Developer
andrew.slice@redhat.com

Bob Davis
Senior Product Manager
.NET Core PM, among other things
bodavis@redhat.com
@bobdavis
A Hasty Overview
What is OpenShift?

A container orchestration platform built on Kubernetes.

Allows for the collective management of containers.

Enables automatic container build and deployment pipelines.

Enables container health checks, resource quotas, and load balancing.

Openshift clusters can exist locally, in the cloud, and extend across both.
What is Dotnet?

A cross-platform family of languages (C#, F#, VB) that share a common API.

 Has automatic memory management with garbage collection.

 Also allows for manual memory management (including things like function pointers)

 Can interface with “unmanaged code” (C, C++, assembly, etc.)

 Programs compile to a ‘common intermediate language’

 This bytecode can then be run on a VM (like Java) or recompiled into a native application.
Dotnet on Linux?

Specifically, .NET Core and ASP.NET Core

**NOT** .NET Framework

Available as RPMs on RHEL, Centos, and Fedora.

Containers available from the RedHat and Centos registries.

Source Code is on Github.

Mostly open source, growing more open with every release.

Can build for other platforms (i.e. for Windows from Linux and vice versa)

lldb and gdb can be used to debug dump files

https://github.com/dotnet/coreclr/blob/master/Documentation/building/debugging-instructions.md
Is it Mono?
Is it Mono?

No!
Is it Mono?

No!

It’s a native build from the same source code used for the MS releases.
Why .NET Core from Red Hat?

- We build .NET Core from source with our build system
- We test .NET Core on our supported platforms
- Our container images are always updated whenever there’s a CVE in any part of the container contents, as well as when there’s a CVE in .NET Core
- Fully supported on any Red Hat platform hosted on prem or on cloud, physical, virtual or container
- We follow fast behind the upstream:
  - New versions generally within 5 days (rpms and containers)
  - .NET Core CVEs usually available in rpms on same day, with updated containers available in under a week.
- Full support available throughout the lifecycle of the version. When we can’t identify the problem, we escalate to Microsoft for you.
Where to get this all?

To install .NET Core, you can either start by installing the content in RHEL or by downloading the appropriate container image from the Red Hat Container Catalog.

RHEL 7:
1. Enable the dotnet repository
2. Install:
   ```bash
   $ su -
   # yum install rh-dotnet22
   ```

RHEL 8:
1. Install from Application Streams
2. 
   ```bash
   sudo yum install dotnet-sdk-2.1
   ```

Containers:

<table>
<thead>
<tr>
<th>Repository Name</th>
<th>Latest Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>dotnet/dotnet-2.2-jenkins-slave-rhel7</td>
<td>6 days ago</td>
</tr>
<tr>
<td>.NET Core 2.2 Jenkins Slave on RHEL by Red Hat, Inc.</td>
<td>2.2-5</td>
</tr>
<tr>
<td>Provides the latest release of Red Hat Enterprise Linux 7 in a fully featured and supported base image.</td>
<td>Health: A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Repository Name</th>
<th>Latest Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>dotnet/dotnet-2.1-jenkins-slave-rhel7</td>
<td>6 days ago</td>
</tr>
<tr>
<td>.NET Core 2.1 Jenkins Slave on RHEL by Red Hat, Inc.</td>
<td>2.1-20</td>
</tr>
<tr>
<td>Provides the latest release of Red Hat Enterprise Linux 7 in a fully featured and supported base image.</td>
<td>Health: B</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Repository Name</th>
<th>Latest Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>codeready-workspaces/stacks-dotnet</td>
<td>22 days ago</td>
</tr>
<tr>
<td>Red Hat CodeReady Workspaces - .Net Stack by Red Hat, Inc.</td>
<td>11-5</td>
</tr>
<tr>
<td>Red Hat CodeReady Workspaces - .Net Stack container</td>
<td>Health: A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Repository Name</th>
<th>Latest Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>dotnet/dotnet-2.1-runtime-rhel7</td>
<td>6 days ago</td>
</tr>
<tr>
<td>.NET Core 2.1 Runtime Only on RHEL by Red Hat, Inc.</td>
<td>2.1-8</td>
</tr>
<tr>
<td>.NET Core 2.1 runtime only on RHEL 7</td>
<td>Health: A</td>
</tr>
</tbody>
</table>
.NET Core Lifecycle

Red Hat supports .NET Core on Red Hat platforms on the same schedule as Microsoft -

- **LTS Releases:** 3 years, or 1 year after the following LTS release
- **Current Releases:** 3 months after the general availability of a subsequent (Current or LTS) release

### Currently Supported Versions

<table>
<thead>
<tr>
<th>Version</th>
<th>Support Level</th>
<th>General Availability</th>
<th>End of Life</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Maintenance</td>
<td>June 2016</td>
<td>June 27, 2019</td>
</tr>
<tr>
<td>1.1</td>
<td>Maintenance</td>
<td>November 2016</td>
<td>June 27, 2019</td>
</tr>
<tr>
<td>2.1</td>
<td>LTS</td>
<td>June 2018</td>
<td>At least three years from LTS declaration (August 21, 2018)</td>
</tr>
<tr>
<td>2.2</td>
<td>Current</td>
<td>December 2018</td>
<td>3 months after a subsequent Current or LTS release</td>
</tr>
</tbody>
</table>
.NET Core Development on Linux
VS Code w/C# Plugin
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Http;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;

namespace app
{
    public class Startup
    {
        public Startup(IConfiguration configuration)
        {
            Configuration = configuration;
        }

        // This method gets called by the runtime. Use this method to
        // add services to the container.

        public void ConfigureServices(IServiceCollection services)
        {
            services.AddMvc();
        }

        // This method gets called by the runtime. Use this method to
        // configure the HTTP request pipeline.
    }
}
Eclipse w/ aCute Plugin
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore;
using Microsoft.AspNetCore.Hosting;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.Hosting;

namespace app2
{
    public class Program
    {
        public static void Main(string[] args)
        {
            CreateWebHostBuilder(args).Build().Run();
        }
        public static IWebHostBuilder CreateWebHostBuilder(string[] args) =>
            WebHost.CreateDefaultBuilder(args)
                .UseStartup<Startup>();
    }
}
Eclipse Che w/ .NET Stack
VIM!!!
Things to Consider

- Not all libraries are supported (for example, WinForms). This is improving.
- While new releases tend to have little to no delay, beta and preview releases are not synced.
- Some low level functions (usually related to file or shell access) may behave slightly differently between Windows and Linux.
- Microsoft’s .NET Core Debugger (vsdbg) is not open source.
  - There are some community alternatives out there though.
Things to Consider

- Not all libraries are supported (for example, WinForms). This is improving.
- While new releases tend to have little to no delay, beta and preview releases are not synced.
- Some low level functions (usually related to file or shell access) may behave slightly differently between Windows and Linux.
- Microsoft’s .NET Core Debugger (vsdbg) is not open source.
  - There are some community alternatives out there though.

Some are more helpful than others.
.NET Core on Containers
Why Containers?

- Isolation and Repeatability
- Scalability
- High Availability
- Layerability (not just for clothes anymore)
- Multiple version and platform support - ability
Developing on Containers?

Not all that different than normal development. Builds can either happen on a dev system or a build server. Builds can be triggered manually or automatically.
Development Tip!

Mount your local source drive into the build container.
Keeps build artifacts and nuget packages on your local machine.
Develop in a local IDE, with your own packages, tools, and configs.
Repeatedly build/run in the same container.
Build/Run environment is not affected by your toolset.

[aslice@aslice example]$ podman run --rm -v=`pwd`:/opt/app-root/src -it dotnet/dotnet-21-rhel7 bash
bash-4.2$ dotnet watch run
  watch : Polling file watcher is enabled
  watch : Started
OpenShift Deploy Plugins

Red Hat has created plugins to simplify development and deployment to OpenShift from popular IDEs and DevOps Toolchains:

- Azure DevOps
- VS Code
- JetBrains IDEs (e.g. IntelliJ)
- CodeReady Workspaces
- Eclipse IDE

Use It To: Enable simple deployment to OpenShift from IDEs and Toolchains
What about OpenShift deployment?

Container Development Kit (CDK)

Download and run to setup a quick openshift cluster that runs in a VM.

Ports automatically mapped for access to the browser interface, or just use the command line.

OpenShift cluster is completely self contained.
What about OpenShift deployment?

**Quick Deploy Templates**

- Select a template, point it to a git repo, and the containers are created for you.
- Hooks can be used to automatically rebuild and redeploy on every git commit.
- Hooks can also be used to rebuild when the base Dotnet container updates (like for security fixes)
- Custom templates can be created to do things like deploy database containers alongside the build and runtime containers.
- Build on a full SDK container, and deploy on a minimalistic runtime container.
What about OpenShift deployment?

Quick Deploy Templates

.NET Core + PostgreSQL (Persistent)

Information

Configuration

Binding

Results

A secret string used to configure the GitHub webhook.

* Database Service Name

Database Username

Database Password

* Database Name

musicstore

Maximum Database Connections

100

Shared Buffer Amount

12MB
What about OpenShift deployment?

Quick Deploy Templates
What about OpenShift deployment?

CodeReady Workspaces
What about OpenShift deployment?

OpenShift Online
What about OpenShift deployment?

OpenShift on AWS and Azure

Roll out openshift nodes in the cloud. Nodes can form a cluster with local, on-premises nodes as well.

Use the previously mentioned templates, or CodeReady Workspaces, with these cloud nodes as well.

Or work with local containers and push them into the Openshift registry for deployment.
Migrating from Windows
Checking for Portability

- Microsoft has a portability checker plugin for Visual Studio.
- It will attempt to offer recommendations for more platform generic libraries.
First Steps

Cross platform build targets let you use Linux for development, even if you deploy to Windows servers.

It also lets you develop on Windows and deploy on Linux servers/containers.

This allows for a slow transition.

It also lets you keep a consistent development environment, even when developing for multiple platforms.
First Steps
First Steps

Like MS SQL? It runs on Linux, and we have containers for it.

We even have an article about it with a demo.

An ASP.NET app container paired with an MS SQL container in a single deployment.

Manage it with SQL Operations Studio

Using containers on the CDK means no server installation.

Database gets in a bad state during development? Throw the container away and deploy a new one.

Try the demo
Documentation for .NET using Microsoft SQL Server Example

Information

- **Name**: dotnet-mssql-example
  - The name assigned to all of the frontend objects defined in this template.

- **Microsoft SQL Server hostname**: mssql
  - The hostname of the database server.

- **Name of Secret containing administrator password**: mssql-secret
  - Secret containing MSSQL_SA_PASSWORD.

- **Git Repository URL**: https://github.com/rmdev/dotnet-mssql-ex.git
  - The URL of the repository with your application source code.

- **Git Reference**: master
  - Set this to a branch name, tag or other ref of your repository if you are not using the default branch.

Configuration

Results

[Cancel] [Back] [Create]
```sql
SELECT TOP (1000) [Id], [Name]
FROM [myContacts].[dbo].[Customers]
```
The Inevitable Question
The Inevitable Question
The Inevitable Question

Yes you can.
The Inevitable Question

Yes you can.

No, I won’t help you.
THANK YOU

linkedin.com/company/Red-Hat
youtube.com/user/RedHatVideos
facebook.com/RedHatInc
twitter.com/RedHat