Speaker: Chris McKenzie

.NET Developers Association – Westside

Parametric: dev group in C#

* Financial services firm

Software

March 2nd international district

March 14th

Speaker: Chris McKenzie

Software dev for 18 years

Failing terribly with Chef.io

Octopus Deploy was a better move

Has continuous delivery

Who runs?

Program then test?

Test before execute?

Automate?

Have you ever done prod release??

A person did a deployment on 7 servers (financial services)

Something was wrong said Bloomberg.

Person check logs, it was okay.

Still was wrong, Bloomberg cut off trading

Server did not deploy properly

Server 8 was deployed with an old text code.

Financial services are responsible for data error to clients

Company lost 400 million dollars

**Pipeline: prod release with Octopus (with no manual intervention)**

1 Developer commits code to source control

Version control – build server

2 Code is built by a build server, tested, and packaged

3 Package is pushed to Octopus

4 Configuration

Dev – testing (tester) -staging (release manager) – prod (release manager)

5 Packages are deployed

6 Whole-team visibility

Best practice

Don’t put your secrets in source control

Octopus has function for this

Octopus Deploy has Tentacles to hand-shake with servers

Tentacles

A windows service that runs on

Can communicate with other tentacles

Can deploy simultaneously

Added ssh targets on Windows

Linux has ssh already

**OctopusDemo (code)**

Manage -> NuGet Package -> “OctoPack” (install in each deployable project)

At OctopusDemo.csproj…

OctoPack is just a fancy wrapper around NuGet

Reads your project file and makes a new spec, new spec to create a NuGet package

Octopus

has a built-in NuGet

it is file-based

read-write logs

used artifactory

OctoPack is not required but a convenience

When it pulls in, it uses the version of the MS Visual Studio’s version

Does it work with ASP.NET Core?

Json project file? Not sure.

In Explorer

OctopusDemo 1.0.0…package

In Command Line,

import -module vscx

use VisualStudioTools

msbuild /p: runOctoPack-true

if at root, it will generate a file on its own.

Open the “NuGEet Package Explorer” – “OctopusDemo 1.0.0.0.nupkg”

Running the server and the tentacle agent at the same time.

Octopus in architecture standpoint

Everything you see in the UI you can see in the API

React / Angular

Run own extension then use Octopus

**Create project “Octopus Demo”**

“use the version number from the included project”

**-> Process**

**Choose Step Template**

**-> deploy to IIS**

Execution plan

Deployment target can be a machine, service or web.

Easier to move from one server to another

**Package**

-> Octopus Server (built-in)

**Application name/Website**

OctopusDemo

**Bindings**

Protocol: http

Localhost: (any port)

**IIS Authentication**

Enable anonymous authentication

**Configuration Transforms**

**Configuration Variables**

Automatic moving of variables to configs

**Environments**

Run only for specific environment – dev – local

Backing of the screen is just a powershell

Can have the powershell and just execute with Octopus (or in the Octopus Server)

**Settings**

Use the version number from an included package

Deploy to IIS

Release

Usually at the end of engineering cycle

Octopus “release” comes at the beginning of the deployment

Reason: changes the process step for deployment and variables in Octopus

When I create a release, Octopus takes a snapshot with the package version.

Does not change even if the ecosystem changes

**Create Release**

**Save**

When? Default “now”

You don’t have to be on site

(In this demo -> deploys in the local IIS)

**Deploy – dev local**

Back to Internet Information Service (IIS) Manager (to check)

After **create release**

Produces Raw Log

Back to Internet Information Service (IIS) Manager (to check)

When you install Tentacle, it keeps history of deployments

C:\Octopus\Applications\dev-local\OctopusDemo\1.x.x.x.x

C:\Octopus\Applications\dev-local\OctopusDemo\1.2.x.x.x

Roll-back strategy, implement older version.

Retention policy

Tell me how long we want to store the file

Tell me how many versions we want to store

In “index.ccshtml”,

Viewing.Title = “Home Page”;

<div class = “jumbotron”;

<h1 #{title} </h1?>

</div>

It looks for the “title”

Coney moon? Best powershell.

**Library**

Upload package

**Variables** tab

Name – “title”

Value – “Awesome Demo”

**Add another value**

Value – “Awesome #{target} Demo”

Scope – “test-azure”

**Add list**

Name – “target”

Value – “Azure”

**Create Release**

It takes the snapshot of the new variables

**Configure Features**

**Substitute Variables**

**Substitute Variables**

Target files - \*\*/\*.cshtml

Use all cshtml

Can be useful for cluster of servers

**Create Release**

**Save**

**Deploy – dev local**

If change the variable to “sensitive”. The value of the variable becomes encrypted. It would never show up in query or log files.

After deployment Reload “Home Page”

Octopus is focused with deployment, not the infrastructure.

Can use a custom powershell or chef.io for infrastructure

To Azure

**Add Step**

**Azure Web App**

Step Name: **Deploy to Azure**

**Execution Plan**

Octopus Sever

**Package**

Octopus Server (built-in)

OctopusDemo

**Account and Web App**

“AzurePersonal” click “bind” to check

“Isg-octopus-demo west US”

**Run on a specific**

**Substitute Variables in Files**

\*\*/\*.cshtml

**Create Release**

Save

Deploy to Dev-local. **Why?**

**The warm-fuzzy**

Deploy to Dev-local

You must deploy it success in test environment

You must deploy it success in dev environment

Will give the ***option to promote***

Don’t need to do the manual testing for a release

i.e. http binding, certificate, password.

**Create Release**

**Deploy to Azure**

**Go to website i.e. http://isg-octupus-demo.azurewebsites.net to check**

Logs the **release process** even if not changing the code.

How about missing a server to deploy?

**Infrastructure** (section)

Can tag all that is map in the infrastructure in real-time

Will mark in red if it fails

Can put a fake node

The phase is called “dev”

***The lifecycle requirement is you must deploy at least 1 successfully.***

Developers can practice without going live.

Smoke-test pass

30-minutes deployment from just watching a video tutorial

Less headaches

Less time

What if secret key changes in Google API?

The service account password

Write-only in Octopus

Write-only in Windows

Octopus’ idea is to automate when the secret key is okay.

For the certificate, it usually requires password. Are you prompted for password to enter for certificate?

It prompt you for password if it has, stores it as a variable.

It has a manual approval step. Sends email to the user. User can click at the approve button

How about auto-scale group, server is added. Is it automatic for new nodes?

Can set automatic deploy if new node if found.

Very responsive support

Good community

In 1 github repo

Experience in team services equivalent? Can have DSTS? management.

Still don’t know that philosophy is.