

## Moonshot Innovations – DevOps KC Presentation

September 17, 2019 – Transcript from Edited Video

As a company grows so do its needs and so does the complexity of its software. Real world deployment automation can often be much more complex than what you read about in an octopus deploy tutorial. At moon shot innovations our largest client needed faster, more reliable deployments for its platform, which is grown to support their business for well over a decade. We were tasked with making that happen and lived to tell the tale. Can deploy absolutely here's how we did it?

The question why automate you want a repeatable reliable deployment. And when you've got an automated deployment process that's inherently testable and that kind of gives you a little bit better assurance that your deployment is going to be a success. With an automated deployment pipeline, you've got scalability. In that you can use that pipeline to deploy to as many services. You need too, and there's no. There's no real increase in the amount of effort to deploy to one server versus in servers. You have the potential for achieving continuous deployment. If you want so that is always pushing the latest and greatest out, whichever environments work for your team that doesn't always work. For every team, but it works for your team. Then that's great with an automated deployment process you can do that and also. An automated deployment is no longer a highly technical task, so what's beneficial. There is that it's not always best for the developer to do a deployment and sometimes it might sound like a controversial thing to say, but And the DevOps world we're thinking about bringing the experts to the table, so that the expert needs to be the expert in their area of expertise. QA needs control of their environment in order to properly execute their strategy. So when they want to put code out when they want to put the latest code out to test it. They shouldn't have to rely on the developers to do that that shouldn't be a. You go back the developers to put the move code out to your environment and we get to it when we get to it. With an automated deployment pipeline, you can open deployment up to QA so that they can do things like AB and code. You have 2 different releases 1 release look fine. They were ready to call that or release candidate. Another release came available and it broke something so they were able to go back to the previous release and in fact. This was broken in the latest release but being earlier was fine. So those are some reasons why you might automate your deployment process. Here at moon shot we've been working toward automating our deployment process since like the middle of last year and some of the goals that we had for achieving automated deployment. Our deployment packages that we build one time and then once that package is built we go ahead and deploy it to death. Once we, as developers are satisfied with the work and we get to the end of spring or whatever or not to the end of the Sprint, but to the point where it needs to be touching the code we go ahead and promote that same deployment package to QA rather than going back to the source code. So, in that way your deployment package kind of acts is a little mini code for use. You know that the code that's going to QA is the same code that you were just fine with development an once QA is satisfied that the deployment that they got is stable has all the features in it. Everything looks good then we can promote that same deployment package to production. So that we know that what's on production is the same code that QA tested in which language? Uh. Obviously, to do something like this to promote the same deployment package from QA to production got to have some kind of environment configuration going on, some kind of environment variable substitution. Another goal was the flexibility to deploy one part of the platform or the whole platform. So we needed that modularity. I mean, you may have a service that needs to go out. But we don't need to deploy the entire platform in order to get that. One service out so we wanted that modularity and we also wanted to be able to handle multitenant deployments. Because we have several different businesses that use. The same platform of ours. So we white label our platform. We want to be able to get that code to all of them customized to their needs. Octopus deploy turned out to be a really great vehicle for getting that done. And I stole this from the octopus site actually still a lot of these slides from the optical site. It says optimistically makes it easy to Automate. The deployment of real worldasp.net applications. It's not just a sp.net applications. You can deploy

things that are not netcentric as well. But we are a Microsoft shop. So it worked well for us. So here I will cover a couple of concepts in octopus deploy that allow us to kind of talk about how we use to octopus deploy in order to meet our needs the first of those concepts. Projects of project is just actually kind of got to kind of jump the slide real quick so projects in octopus. Are deployable unit like a website would be a project or an individual individually deployable service might be a project or scheduled task that you're going to deploy might be a project or if you use octopus deploy a database? Which we do that would be a project and this is just a little blurb from octopus to let you kind of know more about what projects are. I won't read the whole thing out, but Can give you an idea? What projects are in octopus? A release is the deployment process scripts environment variables. For a particular project. So I've got a project in octopus or website or what have you. I create a release of that project and that releases a specific version of the code variables that are associated and all that. Uh. And uh. Deployment target. Is going to be well in this demo this? Is the deployment target a server that you deploy to? Pretty much it. In octopus you have the ability to configure environments. So I can set up several deployment targets grouping together in an environment and that environment might be something like development test or production and. Once I have these environment setup. I can organize them into life cycles. So. Projects. Are. You're individually deployable units. Releases. Our particular version of the code for a project so. Your environment variables in all that for a particular project. The employment targets are individual servers that you're going to deploy code 2 and then with your deployment targets you can create environments. An you can organize those environments into life cycles. So that's kind of a lot, but I've got some examples to try to help that makes sense. Before I do that, though, in order to deploy code in octopus. You need to build that code Ann. Drawing a blank need to build the code and package it up into deployment packages so. A deployment packages contains everything that's necessary to deploy your software be that binaries or static content configuration. And then in our cases were deploying databases as well. Our deployment packages contain backpacks and I don't know how many of you have worked with database projects. But database deployment. Project will contain a DAC pack as well as a public profile so Dylan I didn't know if you wanted to talk about. In that case, I've got. I'll just show you guys real quick. With the contents of a deployment. Package might look like. So this is the database project for the little demo that we're going to run through an inside of the deployment package for this project. You've got a backpack? How many of you guys have worked with database project. OK so you know the backpack represents the schema. That sequel package is going to. Diff against the target come up with migration script. It also contains your pre and post deployment scripts and all that type of stuff and so that's what's here. The schema of what we hope our database ends up as? We also have a public profile here. And this XML is what we used to parameterise this database deployment. I will show you what it looks like right now. I'm not going to go into like all of the nitty gritty for every one of these but because. No, I cannot Oh there goes there goes. So the reason I'm going to show you like all the nitty gritty of this one is because unlike web config or an app config file or an environment dot JSON or or whatever. Most people probably haven't seen in public. XML file each of these little hash curly name. These are tokens that octopus deploy recognizes as places where it can insert an environment variable so using this published profile I can. Publish that backpack. And. Customize the the deployment of the backpack by swapping out the tokens for actual variables relevant to the environment. And that's what's in a database? A deployment package. Other than that. Your deployment package like I say it just looks like what you're going to stage onto the you know the server that you're deploying to it's just that you've got some kind of a hook. That allows you to. To swap out tokens for actual values that are relevant to the environment, you going to. The PDB and the older DLL is related the PD before this package is actually not supposed to be in there, I was just going to rush good, I but. Actually brings up irrelevant point. I don't know if you meant to or not, but maybe you did. When you build these deployment packages. The question does come up? How do I debug a deployment package and without going into too much detail 'cause you could do a whole Top topic. Just symbol servers. Solution is to have a simple server and you can build and then publish your PD BS to your symbol server. Then, after that, you can just connect your debugger to the simple server, and pull down the symbols images. So. But I didn't prepare that so. I don't know anything else about it. I'm kind of rolling through this right now, so. The scenario that I put together for you guys. It's a big company

called real fake events and real fake events is a simple little event scheduling and tracking system. It's real Real Simple and real real fake. It's comprised of a database database. Stats Aggregator Cron job or scheduled task. This Web API and angular client. UI Anna Legacy Webforms Admin UI. The reason that I put it together this way is you know at octopus our largest software platform is kind of a bear. I mean, it's it's grown to support a business over several decades. So you've got quite a mix of technologies in it, and we've been able to get it automated with the exception of reports but I swear it bill that's coming. Octopus does support deploying SSRS reports as well, but we haven't gotten to it. Yet there's a lot of work to do so in order not to put company code out there, I didn't know. What kind of areas, I might need to stay away from or whatever I just made this little fake project instead of digging into our code. Hopefully this will keep things a little bit more streamlined. Um. With real fake news. We're deploying the database and the angular client. UI and you may notice like both of those kind of places with Web API. You've got an app config that you can parameterise with a webforms app. You've got a web config using parameterized. With a scheduled task I mean, you got a nap configuring parameterized there as well. But when it comes to an. Database well how do you parameterise that I tried to call that out earlier on with the public profile that XML file that you can parameterized there. With the angular deployment because angular builds themselves are environment specific so you have to type in G build in the specify your environment? What we did to get around that and give us a deployment package that we can promote from environment to environment without building there again is we created an environment called just DevOps inside of that DevOps environment, the environment variables are octopus tokens. So once you do your build you do your DevOps building. Those octopus tokens are disseminated all over the built code and then you can use octopus to swap out the values of those tokens or swap out the tokens for real values, so that's how we did that I wanted to call that out before we get too into it so. Here's our deployment. I know that text is tiny did not realize until I got it up here. But that is really tiny so 1st. We deploy our service and the steps. There are we acquired the package. We unpacked the package and deploy it and at that time, we do variable substitution on it in order to customize the deployment for an environment. And we do that variable substitution on all config files and then. After that there's a configuration of the IS bindings and I'll show you guys that in a second. After that, we have the web forms. UI it's the same thing acquired the package deployed with variable substitution and the substitution happens on dot config files configure IIS bindings. The angular UI acquired the package deployed the package of variable substitution on JS and Jason files and then configure IIS bindings because we are still serving it up with IS even though it's an angular site. Schedule task we acquired the package stop and remove the existing scheduled task and then we deployed the package with variable substitution on config files and then we create and configure new scheduled task octopus makes that really easy. They have some built-in steps. They do that for you and then we deployed the database and you know in that acquired the package unpacked the package. Variable substitution on the public publish XML and then we apply the backpack to the target database and does its thing. SQL package doesn't thing to build on migration script and apply it with databases. Even though you have the ability to automated deployment. It's a little harder to automated rollback. Unlike with code if the deployment goes wrong in your database. You may drop columns or whole tables or whatever and lose data so whereas with code rolling back can. Can be just you know deploying the previous version with the database you really need to restore from a backup of the deployment goes wrong sorry to call that out. OK, so we are at the point of showing tell time. Here is octopus deploy. Yes. That is a Windows Task Egeler That's what we get for Cron jobs and I will not hang my head in shame. No unless you want to make them in your database and then I really hope my head and check so this is our projects. Remember I said projects are individually deployable units. So we've got a Web API Project, a public UI project. That's the angular project. We've got a database project. This aggregator Here is the scheduled task. And then we've got an admin UI that's the web forms. UI and then got an event platform here that deploys all 5:00 AM so you wanna click deploy the whole platform or you can deploy one of these guys so that's the modularity that we wanted. Uh. Show you guys. I will show you. I will show you one process just want cigarettes. All of the Steps necessary to deploy this admin UI or taken care of in one step template, but it does a lot of things so? Your deployment targets can be tagged with A roll. An once a deployment target is tagged with a row when you deploy every deployment target with

the role. In that step of the life cycle. It's a lot gets the deployment so. In the deployment of the admin UI. Every deployment target in the current life cycle with the admin UI role will receive Department. The deployment will be from the admin UI package. The deployment will be targeting this directory. Where I've parameterized the directory with a tenant tag and will get to tenants in a second? Directory will be purged before installation. I'll be building a website. That's the name of which is parameterized with the tenant tag. The application pool. Parameterized with the tenant tag the name of it and then it will run as a service account that is. That is parameterized as well. We set up our bindings. IS authentication and then here is where the magic actually happens where I replace the octopus variables and configuration files? And that's actually. Ignore this one this is where the Magic really happens where I substitute variables inside the web config file. So it's going to be a similar process for pretty much all of these. I will show you the database project separately because it's the only one that's really very different. With the database project, the first thing you gotta do is get your package pulled down. We're going to target all deployment targets with databases the role. In the process of pulling down the deployment. Package. I will do my variable substitution on publish XML files. So. Once the data packet stage dandy publish XML file has been. Has been substituted once the environment variables been substituted into that file? Now I can deploy the backpack. This is called deployed backpack better because I want to be able to create a database if it doesn't already exist so the existing step template is octopus has a built-in step template. But that step template will not create the database and doesn't exist already so. Just a couple of quick modifications to the PowerShell that backs their default deployed that pack and I'm able to deploy database. If one doesn't exist so I'll demonstrate that in a little bit. It needs to know the name of the step that stage the backpack so that way. You can find it. I tell the name of the backpack and the public profile name of the option of creating a report to just say you know this is what the deployment going to do I have the opportunity of separately generating a script that gets? Attached to the release is an artifact, so I can check that script out later on, and see what happened in my case. I don't want to do any of that. I just wanted to deploy. So when you deploy it still creates the script. It just doesn't attach it as a release artifact. You want to audit what happened later on, you can get that migration script here. Target server name database name. Those are parameterized and then I'm deploying with integrated security because I ran out of time to make a bunch of service accounts. So that is what? That's kind of a quick and dirty look at at the process an appointment process. Oh my goodness. My God, Can I get this to work? What I'm trying to show you guys? OK, here, we go. I've already done the development and QA deployments of the whole event platform. Uh. I didn't think you guys wanted to sit there and stare. At the screen. While I did that and here's the result of those deployments. This is the development environment and this is a QA environment. Now they're on the same box and the reason for that is because I only created one deployment target and tagged it with the correct rolls and with the correct environments for everything I'm going to use this same deployment target for every part of this. Here you can see that the. Deployment process replace the token. Um. With our actual connection string. I like to use integrated security in these kinds of deployments, so that way. You know you don't have to worry about there being a password plaintext inside of your configuration file. All of the app settings. These values were specified an octopus and replaced. And I guess that's about it. In the database. I've got real fake devs and I've got real fake UA. And those were created at deploy time. So. Real fake events have been going great. Ended up selling the platform to somebody they wanted a copy of it. So. I didn't want to. I didn't have the time to rewrite the whole deal. Just a white label it. But the least I was able to do. Is uh? Is parameterized all of the magic strings and everything like that so that way I can I can swap those values out with configuration? In addition to. In addition to real fake events which is my company. I've added another tenant. Called event source. An event source, they want to do the same thing that I want to do, which is manage events. So I make events and then people say, yes, I'm going to Denny's. So I've sold them by platform. And it's parameterized so all that I have to do is enter there. Inter values for their their platform instead of mine. And during the deployment these values will be substituted for the octopus tokens, instead of my stuff. Real quick. I'm just going to show you guys, the actual production deployment of this event platform. So one of the production environment. I'm deploying to both events or 10 and the real fake events tonight. The database Web API, the aggregate are the admin UI in the public UI.

During this deployment process we have a little bit of time I know I'm kind of breezing through 'cause I'm nervous so there so take some questions. I have one. No. It would create. Order. Order. Copy it. The PowerShell script. Going offenders and custom that's really deployed. Yeah, I did gloss over that, but you can you can script your own step templates and you can use bash and Python as well? I think you can you see 2 In addition to PowerShell so they've got something for pretty much any flavor of scripting that you like. Please. Sure. That is a good question to this installation is in the cloud. So this is a cloud managed version of octopus. You can also download the octopus server and install it in house. If you like if you not want to host your stuff in the cloud. Taking its sweet time. Device. Let's take a look. Let's find out together. Oh see. The other thing that I would point out is that if all else fails. Well come on. I know that's there. If all else fails, you have a raw scripting step. You can set up your parameters. You can script inline Oh they supported start II. I didn't know that you could script in line so you can pretty much do anything that you want to do. So for notifications. Of course, there's the scripting you can do your thing they also. Do support emails. So. You going to your host import. And all that information and you can send emails through I built 10 step template that will do that for you so you can send emails to notify people when you're getting ready to deploy you can also insert manual approval steps, so somebody has to come into octopus and respond to that email in order to move the release law. Yep. Yes. Do you have? Yeah, so it's very customizable, I mean, really to your needs. I'm not your problem. Oh yeah, there's a subscription 2nd as well so. Is that what you're looking for the books? Yep. Less. Again, it's just what works for you. So. All very customizable configurable. So. The production deployment is finished. And because I'm deploying to the same deployment target for everything just for the 2nd this demo. Here are the applications that. We're just deployed. Real fake events and event source. Anne what you'll notice is that? The connection string is been customized to event source DB. Event sources is a tenant an? My company real fake events has been converted into a tentative well so when I do is tentative deployments. I just have to fill out the variable template for the tenant an those values are? What are used to parameter my Department? Uh. See. Alright Here we go. I think we covered this. Well, I just burn through this interesting so as you can see I mean octopus deploy is very customizable pretty much supports whatever usage scenarios, you may need it to do. They have a huge Library of custom step step templates and you can make custom step from templates. If you like using different methods of scripting. Hum. I guess that's about all that I had. I thought my presentation was a little longer than that so. Sure. Octopus support is actually really good, I have had to email them couple of different Times Now I've had weird things go on where something randomly broke they push code and next thing you know something's not working quite right. But I'm not been able to reach out to them and get an answer. Now they are based in Australia. So you have to give him a little bit of time just because you're probably asleep when you send that email. But yeah, I've I've had pretty good support from them. Did you have a bad situation like to share about it or? Skype. I like you. Start recruiting globally. Yeah. Waiting for Right it's a lot more responsive it's actually. Pretty amazingly responsive through. They do. And their ability to get answers from other areas of their company in order to answer your ticket is pretty impressive as well. Yep, Yep. Oh. Yeah. So, your package acquisition that's the deployment packages. I was talking about where that package is going to contain whatever needs to be staged on the server to run your application. So it will have binaries in it, it will have static files if it needs to have static files in it, your configuration is going to be in there pretty much whatever you would expect to see. You know inside of your folder and applications being served out of that, so it's going to be in the deployment package format. They have they have a knock to pack format. But you can also use new get you can also use zip you have different archiving format so you can use. Yep. Also. Yes. Order. Either. In octopus also has an internal feed that you can use as well, you can publish. Cool. But we use now. Doesn't matter if it's? Survey. And you can actually. I did that, after the elections next to your server on our basic cloud service. Yes. He asks So he building. Absolutely. Yeah, I mean, there's a whole Library of. Of. As far as I know there all vetted but I'd tend not to use the community step templates myself. So I'll tend to write my own or else use the built-in ones. OK. OK, that's good to know I would hate to tell you guys. Yes, they're vetted and then you get in there and. Absolutely. Of course me if I've got time to vet somebody elses code. I would just write it. The Destiny is a code not see I can't help it. Uh. What else one of the questions in the term? In a way. In a way you know they still need. The

concern there is you give somebody that view and then they see that something is changed, but they don't see the work items that are associated. Or maybe they do see the work items that are associated but they don't understand the details of those work items so. It's a great it's a great way to communicate with a technical project manager. It's a really great way to get yourself into hot water with a non technical project manager with a quick mouse paper you know what I mean? But that that is true that dashboard is there and it's very helpful. So to take a look at the dashboard. We see that the event platform has been deployed. This little badge right here tells you that there was one tenant that could receive this deployment which is true because I only have the one development tenant. Uh. One QA tenant and then I have my 2 production tenants real fake events and event source. So this, let's you know that everything across the whole system is up to date with the latest and greatest which is version 1, so good point. Because I prefer to use the same process across the entire life cycle so that way. I know that I'm testing what's going to go to production. You said I mean, so. In my death tenant here. I filled out the variable template. And. It's the same variable template that I filled out for a production tonight. And I know. That. No matter. How weird and situational their code could potentially get behind the scenes and I don't know anything about it. I know that I am testing. The exact same process. I'm getting a tenants variables and I am customizing the deployment package with attendance variables. Element. Yes. Yep, and well and potentially the infrastructure is remember you have these deployment targets. And I've only got one deployment target. It's my laptop an on that deployment target. I specified that my laptop is the deployment target for dev QA event source and real fake events tenants. I could just as easily have had 4 separate. Uh. Deployment targets and send each tenant to a separate one, I just. That appointment targets may not all, be the same. What we did see your everything? Oh. What I was talking about wasn't necessarily the environments person it's more about OK so? Um. Outside of tenants outside of outside of a tenant variable library say I just have project variables an I use different. You know they've got different. Scoping the options that you can use for a project variables, a lot of the scoping options aren't available if you're using a? If you're using a? Eh. Environment Variable Library of using an environment variable template I'm sorry. A lot of those options are available to you so I want to make sure that I'm using the. I'm using the method that I'm going to use in production of an example of a place where. Scoping options are different between project variables in tenant variables. You don't have the option of inside of a variable template. And this is one thing that bothered me about octopus. Which I'm a fanboy up but? Here inside of this variable template. I don't have the option of saying This is the tenant tag for this scope where scope might be an environment or a deployment target or what have you. I only have the one value? No scoping so in order to handle like load balanced environments for multitenancy you end up having to create a load balanced like. Octopus environment where I have. Production environment and production load balance environment. Anyway, those kinds of. Those kinds of differences in scoping options that are available. Most of the reasons why I prefer to do things one way. So. Very disciplined. Yes, you can do that. Will work? Or. Variable. All right. Well. Yes, but That's a That's another area where I mean, I'm glad you pointed that out because there's some there's some issues with variable scoping that we didn't end up touching on because I'm using variable templates here for the tenant variables, but inside of a project inside a project variables. I mean, you can scope things 1,000,000 different ways. You can really rap yourself up in knots with what substituted into what and when its scope. This way or that way. What's the value. You have to be very careful and disciplined in how you scope those variables. No. Um. There there are you can export Jason. That contains all that there's an API as well that you can use to get access to that information so you can build their own UI. There's a lot of ways to get the information that you want. Do I like. Very good. Yeah. It's pretty nice. Whole whole UI zigzag. Anything. Very nice. And there's a client, for that as well. I can't remember the name of the talk to. Well, there's actually EXE but that's more limited than what I was thinking of. With some open source stuff going on where they make parts of the API available to you in different ways. Hush. Yes. It is. So guys that's what I got. Are there any other questions? Finance. Define. Over yelling seated before checking this sub stuff. For Pasadena air level back out through collect. Articles like this stuff player. They like you probably probably. Delta fair. Those are massive output. Well, so my biggest recommendation is typically you know you can solve a lot of these kind of problems by breaking things down into smaller pieces. Like if I were if I were to do the

whole. Event platform is just one giant chunk one, not only what I lose the ability to deploy just one little module. If I needed to. But if something were to go wrong if something were to break the build or break the deployment that just really adds a lot of difficulty so you trying to troubleshoot it sounds kind of like. Something similar to what you guys are doing I. I'm not really familiar with what you would be doing where you're deploying something they accepted the PDF if you could give me some more details on that. Hello. Yeah. When I hear swarm I automatically want to run away. But. In general. Thank you. Carefully. How you doing? Other everything that might be. Oh, I see what you're saying and yes. Whatever strictly Absolutely. Yeah. Speed up database calls. Really slow. So we Unknown. Come here. Uh so we tried to then go ahead and do you need the women? The moment happen. My God, You're going to hear me, laughing this quick and decoration where we're not all that. 'Cause it ain't the last exit code. Yeah. One thing I like. You know. The gathering on the job. So manage are exceptions. Yeah, what's the deal now? I'm glad we're able to have like a conversation about it and I think it is a great tool. It's just it's one of those tools, where you can really shoot yourself in the foot. If you're not really diligent and careful and you make sure that you do things one way. I guess it's working clean is a common theme in there. So. Right that's all that I have, I don't know how to wrap this up other than thank you for coming.