Foize Customer evidence

# Customer profile

Foize is one of emerging startups, coming from Netherlands, that operates globally. They are focusing on creating software solutions for Digital Social Communication, Geo-fencing, Phone as a Sensor, Location Based Innovations and etc. both for consumer and enterprise customers. Their unique solution provides new possibilities, in modern communications, by adding Geo-tracking dimension to convention communication mechanisms, information exchange and alerting. This solution is called **24Coms**, and it is growing number of customers and companies.

In Foize's own words " "Relevant and Meaningful software solutions for People" is what we pursue for all our research, development and innovative solution thinking processes.*"*

For this DevOps Hackfest we have partner up with Foize and hacked their GPS component of overall solution called 24Coms.

Hack Team:

* Leo Winder
  + Owner / Ops
* Michael Vlaar
  + Developer C# and Ops
* Leon Keijzer
  + iOS Dev
* Ronald Koster
  + Developer Android and DBA
* Valery Jacobs
  + MS TE- Azure; Dev
* Srdjan Bozovic
  + MS TE – Mobile/Azure; Dev
* Aleksandar Dordevic
  + MS TE - DevOPs

More at [www.foize.com](http://www.foize.com)

*Preparation steps, for DevOps Hackfest delivery, was assessing current state, areas of improvement and focus areas of the actual hack. Motion consisted of 3 phases/steps – 1. Forming; 2. Storming; 3. Performing. During these phases we have managed to narrow down what area to improve, what to deliver and to go and DO it!*

# Problem statement

Let’s start with brief description of challenges, how we got to it and what was the road. I’ll provide you with context, but prior jumping to it, allow me to share challenges (or areas of improvement) that Foize team has been facing.

It can be summed up in to following: **Scaling** to support rapid growth of net new customers; **automating** as much as possible operations, development and recovery; and last but not least **crafting/accepting DevOps skills** and “spirit” needed to provide them accelerated migration to full Cloud solution. Can we agree it sounds interesting? …Well let’s see how we tackled mentioned, had fun along the way and solved parts of the puzzle.

After teaming up with Foize team, we have had our initial “forming” session where we had discussion around overall application architecture, used technologies, current production, operations and business cases that solution 24Coms addressed. Initial improvement areas were captured, such as level of automation, deployment techniques and vNext for the solution, taking in consideration fast improvements and capabilities that modern Cloud is offering. This meeting has set up the focus on GPS component, of the solution 24Coms, that is currently running on Azure, as few Worker roles, has been developed with C# and is using Service Bus as main building block of this service.

First drill or “dive in”, we did by performing Value Stream Mapping (VSM), where we have captured overall process, from development, via staging and testing, all up to production usage. By using VSM we identified DevOps practices, and areas of improvement – e.g. there were some manual deployment processes, more optimal usage of Build process and delivery to production/staging/development environments.

Take a look at the captured, and see what we have crafted, captured and created during whole day VSM session.

Figure Value Stream Map of 24Coms solution



As mentioned before currently, Foize has Hybrid solution, based on hosted and Azure services and combining these worlds (hosted world with Cloud world) when providing service to end customer. It is clearly visible, at picture above, that we can see following major building block of their solution:

* CI/CD/CM – Atlassian Bamboo
  + Build Servers at AWS
* Hosted Environment at local ISP
  + Production and Staging Enviroments
* Microsoft Azure and Visual Studio Online
  + As a their choice of vNext and interm mode of the 24Coms solution
* Users and Devices of the solution

As expected it has solution as it’s good, bad and ugly sides (as any other solution has), and by growing numbers of customers Foize made strategic decision to depart from this current, working model, towards full Cloud solution. They started this journey, by them self, did an awesome job and then partnered up with us to improve overall model.

Output of VSM provided us with focus areas and a challenge. We had, in front of us, to 1) improve current DevOps processes, 2) create a “blueprint” or template for future services migration towards Azure and to 3) keep the production running while we experiment – it is fair to say it is demanding, right?

In DevOps “terminology” we decided to focus and hack following:

* **Infrastructure as Code**
* **CI/CD**
* **Staging and Production environments**

Let see how we crafted plan when we meet for the DevOps hack…but prior jumping there, take a look that happy faces of my friends from Foize

Figure Ronald (left), Leo (middle) and Michael (right)



# Solution, steps and delivery

With Value Stream Map process done, we have captured focus/hacking areas. Next time when we have met, week after, we crafted a plan and steps toward solution. Prior sharing experiences, journey, fail and success attempts let me share few pictures so you get atmosphere we have had.

|  |  |
| --- | --- |
| C:\Users\adjordj\AppData\Local\Microsoft\Windows\INetCache\Content.Word\WP_20151111_10_48_13_Pro.jpg | C:\Users\adjordj\AppData\Local\Microsoft\Windows\INetCache\Content.Word\WP_20151111_11_02_50_Pro.jpg |

As we have identified manual deployment to production GPS service, manual recovery procedures, improvements at continuous integration and delivery. We have focused how to address these problems and as we previously agreed first we are going to do IaC, prep the Visual Studio Online environment, focus on CI/CD and lastly deal with staging and production environments.

If you wish to see our “master” plan – well it is simple, but effective…here you go.

Figure Overall Architecture and Action items for the hack

|  |  |
| --- | --- |
|  | C:\Users\adjordj\OneDrive - Microsoft\Work\!! IT PRO !!\DevOps Hackfest\Foize - 24Coms\Pictures\WP_20151111_12_41_07_Pro.jpg |

Let me drill down in to the details of Action Items what we were aiming to achieve:

* Utilize VSO as source rep and CI/CD/RM component
* Create IaC that will able to recreate and “enforce” identical Infrastructure environment setup and configuration.

First part, with VSO was relatively easy since it included all of us present and our experience with the platform. So making changes to source code, enablement CI was smooth. But in parallel we had some challenges to create IaC for creating Service Bus, Queue and Topic. As you probably might know currently with Azure you can use old management module Service Manager (SM) or new Azure Resource Manager (ARM). Microsoft Azure is currently in interim mode, and as we are departing from old management model to new, in these case new model didn’t had support for resources we were trying to create. So first task for “IaC team” was a) make it run from your machine; b) make it work with VSO. I believe you are guessing we have had problems with it, so I’ll share nuggets and how to overcome this.

*Prior going there, I would like to thanks to creator of the initial script we have “hacked” - Paolo Salvatori (@babosbird) - Thanks for the awesome job, it helped us a lot!*

So let’s see what we had as a problem and how did we solved it. But first let’s see failure:

|  |
| --- |
|  |
| C:\Users\adjordj\OneDrive - Microsoft\Work\!! IT PRO !!\DevOps Hackfest\Foize - 24Coms\Pictures\WP_20151111_14_51_41_Pro.jpg |

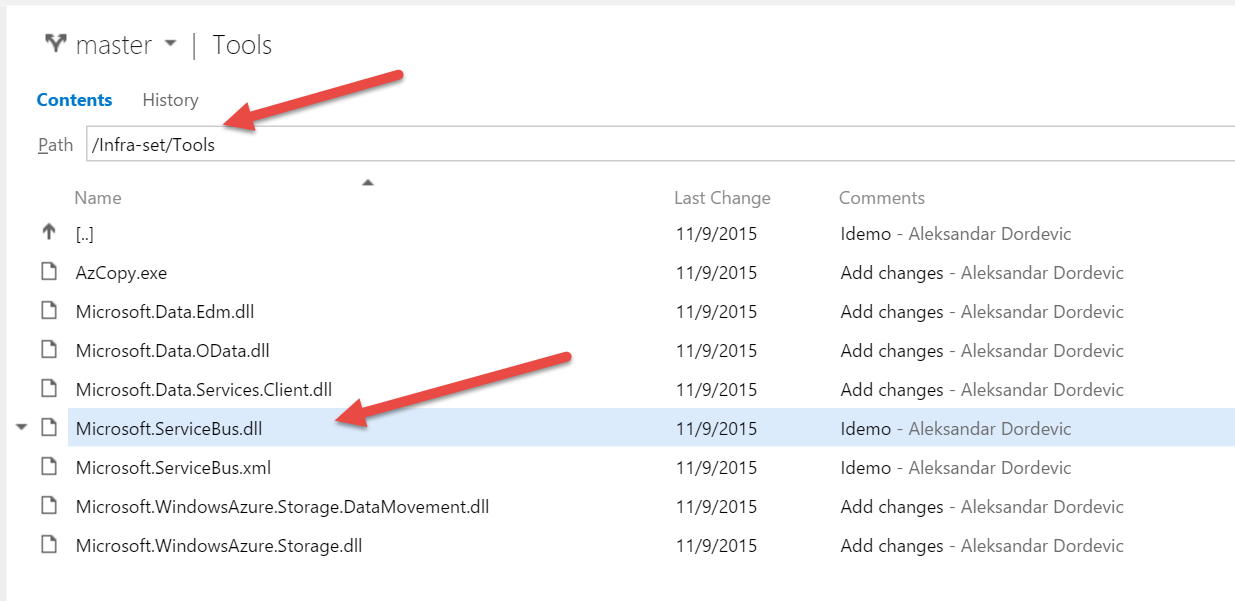
…after few attempts, and failing, we have learned following…

In order to create Service Bus, and other related components, you need to use Power Shell and reference code to use Microsoft.ServiceBus.dll – must say it is not pretty, but it does the job. It is easy to do this when you manually are running script form your laptop, but if you want to automate and use Agents, it is another story…

Challenge was how to point VSO Build agent to a) finds Microsoft.ServiceBus.dll; and b) use it while running task/PS script in VSO.

Here is how we solved it:

1. Add Microsoft.ServiceBus.dll to source repo:



1. Update your code, especially variable that will point out to location of DLL ($srcDir)

Param**(**

**[**String**]$srcDir** **=** **$env**:BUILD\_SOURCESDIRECTORY

**)**

# WARNING: Make sure to reference the latest version of the \Microsoft.ServiceBus.dll

Write**-**Output "Adding the [Microsoft.ServiceBus.dll] assembly to the script..."

Add**-**Type **-**Path "$srcDir\AzureResourceGroup1\AzureResourceGroup1\Tools\Microsoft.ServiceBus.dll"

Write**-**Output "The [Microsoft.ServiceBus.dll] assembly has been successfully added to the script."

1. Run again Build at VSO and enjoy (finally step forward)

|  |
| --- |
|  |
|  |

Once we have solved IaC, we had rapid acceleration of our hack since all team members could focus their efforts on CI/CD processes.

Next step in our journey was to deliver CI/CD to Staging, and we were able to do so by using Azure Cloud Service Deployment. We have crated final task, and we were ready to Rock ‘n ‘Roll…



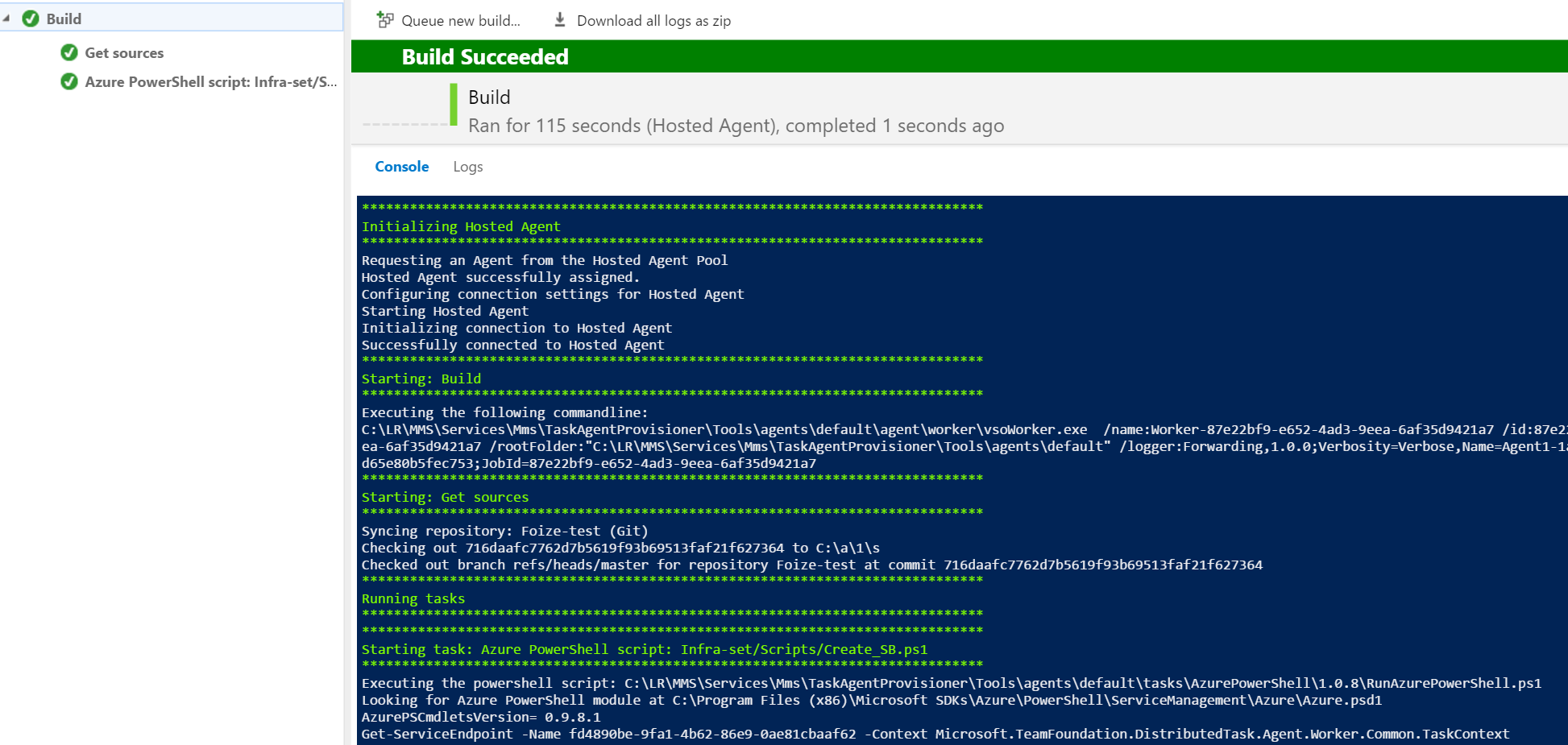
…but there was another obstacle – again our Build and Deploy was failing. For some reason task Azure Cloud Deployment was not getting the right context regarding ***CurrentStorageAccountName***. So, we went to GitHub, found PS script for the task, examine it little bit and found a workaround. I must admit it was not the smoothest one, but we wanted to push forward and to get to our goal. If you ask what we did, simple, we added a reference to PS script.

Here are steps to fix this step:

1. Go to GitHub repo vso-agent-tasks/Tasks/AzureCloudPowerShellDeployment
2. Download Publish-AzureCloudDeployment.ps1 and add to your source repo
3. Add following line at the row 222

Set-AzureSubscription -SubscriptionId <Your SubscriptionId> -CurrentStorageAccountName <Your Storage Account Name>

1. Save and commit
2. Add it as Build step instead of Azure Cloud Service Deployment
3. Run Build again!
4. And… SUCCESS – FINALY! (we were in need of fruits of success to keep us motivated, so this came at the right moment)



So far we have achieved following - we were able to Collaborate, check-in our code, deliver CI and use IaC to ensure Service Bus, Topic and Queue.

Next step was to create Worker roles and environment to “drop” the code. Finally, I can share things were getting smoother, we were able to create Worker Roles with few lines of PS code. Let me give you a sneak peek in to it.

**$opstat** **=** New**-**AzureDeployment **-**Slot **$slot** **-**Package **$package** **-**Configuration **$configuration** **-**label **$deploymentLabel** **-**ServiceName **$service**

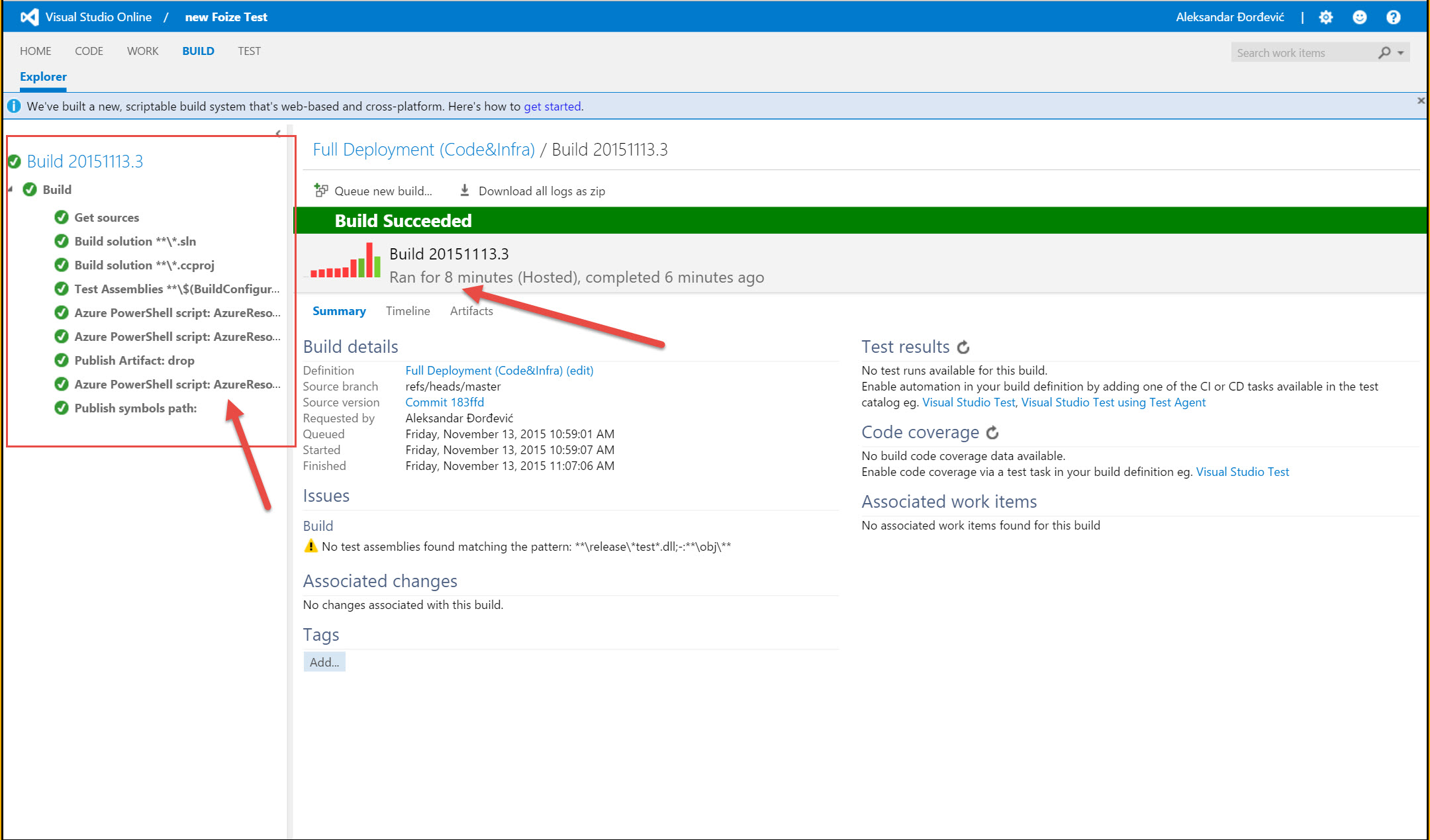
**$completeDeployment** **=** Get**-**AzureDeployment **-**ServiceName **$service** **-**Slot **$slot**

**$completeDeploymentID** **=** **$completeDeployment.**deploymentid

write**-**progress **-**id 3 **-**activity "Creating New Deployment" **-**completed **-**Status "Complete"

Write**-**Output "$(Get-Date -f $timeStampFormat) - Creating New Deployment: Complete, Deployment ID: $completeDeploymentID"

After completing this step, we were ready, excited and nervous to see what will happen…and take a look, after ~ 8 minutes, on the day 3 of our hack we have successfully assembled solution. In our eyes success was looking like this:



# Conclusion and final words

As you could read/see, we had very interesting journey over 3 days of DevOps Hackfest we have conducted. It was very intense, interesting, demanding and fruitful process. Ultimately it helped everyone involved to change, evolve teams spirit and most importantly have better understanding of failure – Failure is essential component of improvement process, hence it shall happen as we are constantly trying to excel in our work/life...everything.

We did achieve what was agreed, and set as target. Now Foize team has its “DevOps spirit” enrichen, they have experienced new practices and polished some very well known to them, and ultimately created automated recovery. For, if I should choose one achievement that is easy measurable, I would go to MTTR. Initially Foize had manual process of recovery, and by their own words it could take from 2 to 4 hours, prior coming to hack - Now it is done in ~15 minuts. IMO that is improvement – do you agree?

Of course, we had our fun with CI, and then with CD – but it was known already to them, they were using it, so we polished it a bit…anyway, question to you (all), who is not using at least CI today? Right? ☺

Let us little bit turn now to non-functional side of the hack. I think it is always good to spend time to understand, talk, exchange, learn, share and have fun with your peers from other business. It provides us all better understanding what is out there, how our IT world is evolving and what our others ideas. Believe we all got or hand full of nuggets, both technical and non-technical. Can’t wait to do repeat this again…

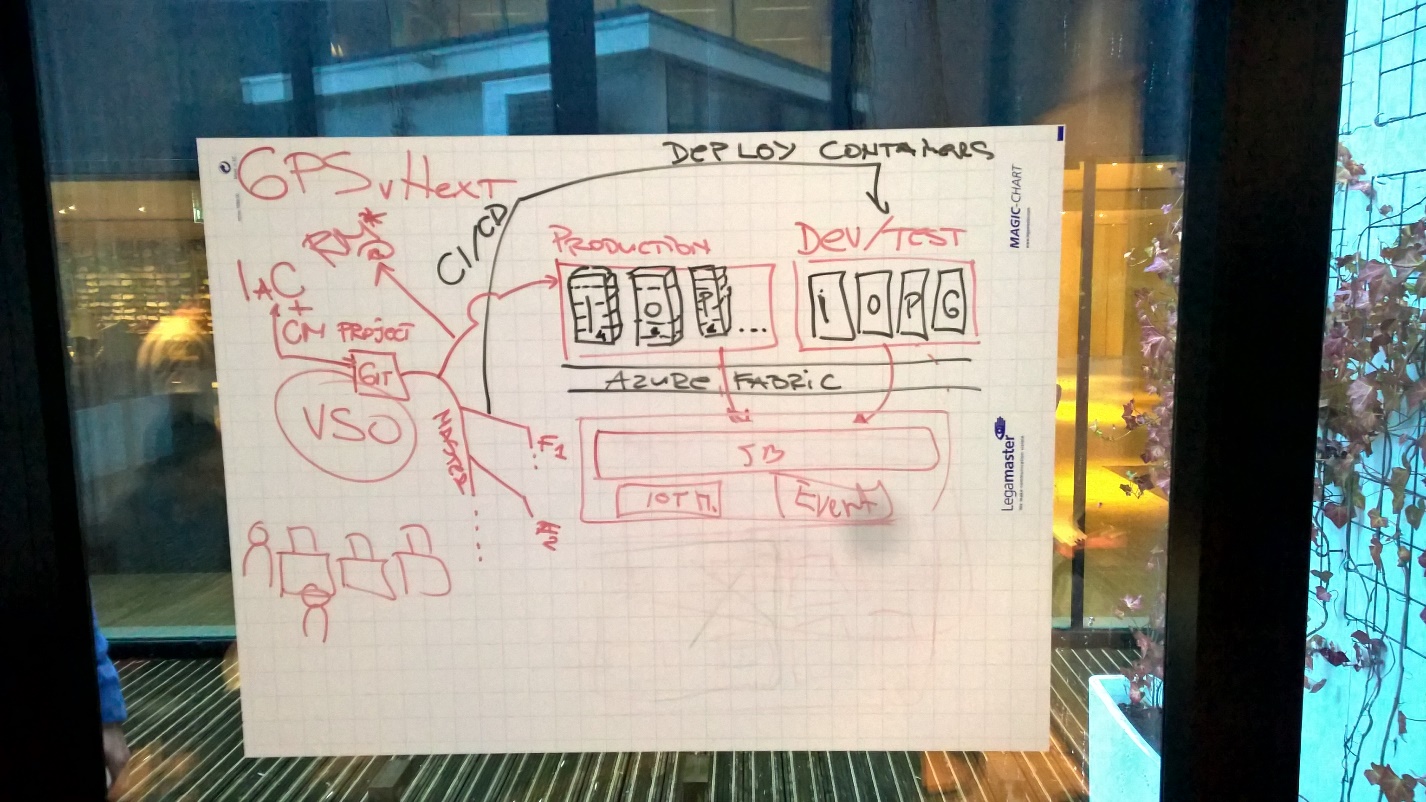
I’m grateful to Foize team for multiple reasons:

1. They had a great idea and they were persistent and after while it became a product
2. As their solution was getting more attention, they recognized potential “bottleneck” and were open-minded to solve it
3. For having us, as their extended team, for these 3 days
4. Awesome spirit they have

I believe that picture is telling more than 1000 words, and video even more – so take a look what they have to say about hack

(see “Foize quick1.mp4 in the same folder as this document)

As any work, it is far from done, this is just one step, out of many that Foize team will make in the future. As we have bringing our hack to an end, we took little time to imagine how current solution could look like in the near future. Immediately there were few ideas in the air, Azure Fabric Service, Azure Container service and etc. We decided to allow us to “go wild” for a bit, and we have created brief/draft future Architecture of it.



It is not detailed, but it represents beacon and way forward. We shall reach out to Foize team, check how they are doing and how far they have got. Who knows, we could/should/will do this again if things and “runes” connect. ☺

For the end, I must emphasize – DevOps is a team game, hence it can’t be played/conducted alone. Grateful, proud and happy to have had opportunity to deliver this DevOps hackfest – Thank you Leo, Michael, Leon, Ronald, Valery and Srdjan! Without you this would not be possible!

Alex