**[Optional] Presentation: e2e tests and on demand test environments with Gitlab and Kubernetes for W (2023-02-21 12:03 GMT-5) - Transcript**

# **Attendees**

Cory Massaro, Genoveva Galarza Heredero, James Forrester, Julia Kieserman, Peter Wangai, Rummana Yasmeen, Stef Dunlap, Stef Dunlap's Presentation, Vaughn Walters, zdw

# **Transcript**

*This editable transcript was computer generated and might contain errors. People can also change the text after it was created.*

Stef Dunlap: Okay. Sorry for sharing my screen.

Stef Dunlap: Really.

Stef Dunlap: Let's see. Some pictures of yourself for a moment. and,

Stef Dunlap: Oh no. Okay.

Stef Dunlap: All right. Hello, everybody. Welcome, welcome to my Ted talk. Just kidding. Welcome to my regular talk, we're gonna be talking about that which stands, which is an acronym stands for doing unique CI things. I've got this a pattern for creating her patch test environments, for any and tests and reporting back to Gitlab or Garrett currently Garrett. And for folks who are joining from the community, you probably know, but you may know. But Garrett is our legacy version control system. Gitlab is a new version control system that we're in the process of moving towards.

Stef Dunlap: Just a little bit by way of introduction. So I think just about everybody here knows me, so I'm Stef my pronouns are here steps offer, engineer and tests at Wikimedia, Foundation. I'm currently invented in the abstract Wikipedia team, and these are all of the ways that you might choose to contact.

Stef Dunlap: Okay, so here's the problem statement. It's currently hard to automate creating her patch test environments, for wikimedia extensions or services that integrate with other Wikimedia services. So for instance, Wikifunctions which is the product of the team I'm invented with has Media Wiki, which is the open source software that power of sites, like Wikipedia and embedded with as embedded in an extension weekly lambda. And that talks to micro service that we call the function orchestrator, and that talks to another micro service. We call the function evaluator and pretty soon. There's going to be even more. There's gonna be different function. Executor, so, there are tools that we can media to

Stef Dunlap: Both, can you hear me? Sorry, my headset when my ears had such just went out. I want to make sure you can still hear me.

Stef Dunlap: oh, let me just Double, okay, great. Thank you very much. Okay.

Stef Dunlap: So, the tools that we have it, we can media right now, for testing primarily is what is often referred to as the pipeline, which is a mixture of some tools like Jenkins, Zool Jenkins Job builder. So, this is what most teams are using to test their product. The problem is that these tests aren't designed to spin up more than one container per job. So, if you're testing against a microservice, you have the difficulty of I need to spin up my media wiki and my microservices. It's very hard to spin up a container inside a container, you know, it is doable, but it didn't seem like the right tool for the job.

Stef Dunlap: There is patch demo, which if you haven't heard of, it is a tool that specifically about making a test environments, but it can't deploy customer, specified, wikimedia, services, alongside of it. Lastly, of course, there's beta cluster, but this is a shared tenant environments. We so we don't control all the aspects of it and can be changed out from underneath us. And we don't control the deploy schedule. So, It can be hard if we want to test per patch on beta cluster with different. With different micro with different matrices of microservices.

Stef Dunlap: Okay, so now I'm going to give you a demo of the duct pattern, it comes in a few steps. The first is pushing a patch. Next is waiting for the deployment and tests to run in less than seeing the results. The results come in three. There are three parts to the result that we'll look at. So this will be on Garrett. We'll see that it's verified plus or minus 1 depending on if the end plus path. There'll be a link to the pipeline tests. And will be a link to the custom environments. So we're going to run through just a dummy patch. So just so you know, this is what the page looks like, beforehand. Take a few moments to commit this to memory because you will be comparing it with What shows up at the end of the test? All right. End of the test run. This is just the page that is talking about the function echo.

00:05:00

Stef Dunlap: very briefly, wikifunctions is a sort of a function or function executor like think of A Think of the wikimedia movement, made aws lambda so that everybody could edit it. That is kind of the elevator pitch wikifunctions into. This is just the function echo, not super important. That's what we're about to see in this pipeline. So first, I'm gonna public a pasture review. So, here I am at my terminal. I, this is just a styling change. Just added some information to one of the container div, given it a nice background that we'll see in a moment. next part of this is I can go and confirm that the pattern is ready to review, explaining waiting to be reviewed and verified,

Stef Dunlap: So next is the important step of waiting. You can get up and go get a coffee or a tea or some other delicious beverage at your cat.

Stef Dunlap: And finally, the results are in. So we're backing Garrett, the Garrett UI and we see that the aw gitlab thought so abstract, Wikipedia, I gitlab, but our end tests the past. And it's given us a link to the results and a link to the environments that will still be around until after the patches merged in. So first, let's go look at our test results. So if you click on this link right here in this middle link, it'll take you to this page. this is a, this is our gitlab This is our gitlab CIA instance and it's showing the pipeline. So you see the steps of it and you can figure configure the deploy set up.

Stef Dunlap: Because the deploy and it ran the tests. And then this last bit is how it links back to Garrett. Let's report success at the end if it failed, it would have ported failure. So, you may want to actually see what was run in the tests. So if I click on this this test, Section right here in the UI, it'll take you to something that you've probably seen for energy, something similar to Neci system, the output of the test, and you can see that this is done and tests or making a function and evaluating that function, so it's really gone through and driven the browser. As if the user is the user, drove it.

Stef Dunlap: All right. Now I'm just going to step back. If I, you'll notice that there's also a link to the environments. So, this is the environments used to run the end to end test in its kept around until the patches merged in. So this is useful for designers and PMS and even engineers, or people, maybe reviewing open or open source patches or just patches of things that maybe they're not super familiar with and want to test it out manually, or do the exploratory testing. I'm gonna. So if I click that, I would see. This is the, This is our new Echo function page. You notice that I have updated the background, it has now we could pay ten in a sealer, uniform, giving a salute. This is a great improvement to the Web page that I think we can. All agree is what Wikifunctions is missing.

Stef Dunlap: Jokes. Okay, so that's the demo. There will be time for questions at the end, but I'm gonna step next towards the architecture ducts.

Stef Dunlap: so in a very high level, we have Garrett where the code is pushed, Gitlab ci where the tests are run. Or at least the tests are set up pipeline is set up in the pipeline is run and Kubernetes. Where

Stef Dunlap: where we deploy the services in the code and run the tests against and then we have

Stef Dunlap: Gut and stuff, or sorry. We have duct tape, which is a microservice. Then I'll talk about a little bit more in a second but it's, it's chief role is to pull Garrett for new connects and then trigger the pipeline. That's, that's really that's really all at this. The reporting and the actual running is handled by MCI.

Stef Dunlap: so, duct tape is the only custom code induct In the stock pattern that we're talking about, it's a rough step that runs on Tool. Forge Tool Forge is a platform as a service that we run for the community at wikimedia.

00:10:00

Stef Dunlap: And it's designed to be obsolete when the Garrett to Gitlab migration completes. So, you know, hence the name duct tape right now, it's dictating the two services together but what you'll notice is it gets a lot simpler. Once we migrate to gitlab ci inside of Gitlab there or insight and gitlab, there's the Persian control system and the CI system and they are a couple together to integrate seamlessly. So as you push new things to, as you make a merge request and gitlab, it will make a corresponding CI run, CI, run, will still talk to Kubernetes. I get the results back and then put the results in the merge request, and that's all pretty much platform.

Stef Dunlap: The last component that I want to talk about in, this could really have a whole talk in and of itself is helm. So Helm is a Kubernetes configuration and deployment tool. It's what we're using to actually put together. The matrix of microservices that we're testing and get all of that stood up inside of Kubernetes. The good news is if you're testing another app from the Wikimedia Foundation, there's a good chance you already have a production home chart that can be adopted into CI. So if this is something you're interested in setting up later, that's something that we could work with the SRE team. In the running or sorry the religion team.

Stef Dunlap: To get started for it. Sorry to get started for you. You can also define tests to run against the deployment what helm tests. So those tests that we saw from Gitlab CI are being run by home tests and they're actually being run inside the cluster on a container that's configured to talk to the components inside inside your deployment. So and handles a lot of handles, a lot of that networking and domain name pointing for you which is nice. And then there's a link to our home chart. So we have a home chart right now just for CI, mostly because we are using like we are testing this and didn't want the test interfere with our production home chart, I can see a time in the future will, this will merge together with the production home chart or maybe it won't. Maybe we'll always have a home chart for CIA and I'm sure production

Stef Dunlap: So lastly, these are some links to the relevant repos. I will be sharing out a link to this presentation so you'll be able to go check out these repos. There's that which is this presentation and also where some of the work and progress, documentation will be going. Are Awci chart. This is the helm chart for wikifunctions and the CI environments are a wikifunctions VI environments. And lastly, there's this other repo, AW, ete and this is an sort of an implementation detail of gitlab CI. Like, you need to have a repo that configures the pipeline and is responsible for where the pipeline will be triggered. So if you go to awe, you'll see our gitlab kind of thing for the spike plane. Oh, and I forgot to put a link to

Stef Dunlap: Conductate the Rust out. I'll be sure to include that. I think it's earlier in the project, but I'll be sure to include it. afterwards before I distribute this out, so, All right. So that's the end of the formal presentation. We have a few moments believe in the time. Now we have. Oh yeah, we have roughly a 10 minutes or so, for questions. There's certainly a lot that I didn't get a chance to dive in deep here, but I wanted to give an overview and then see I had questions so we're gonna stop sharing. So I can see all your lovely faces again. Okay. Um,

Stef Dunlap: So, any questions?

Stef Dunlap: Go ahead, James.

James Forrester: Hey Steph. So you set the like the end end Kind of repo was kind of an implementation detail. Do you think if we were, if when we move all of our deployment, developments over to Gitlab from Garrett, then that would just be the regular repo, which has a helm chart inside the code, we were writing anyway,

00:15:00

Stef Dunlap: Yeah, that would be a good chance. That that's what we would do. There is a chance that we might have a separate repo if it's specifically in integration tests for a lot of different repos. So because like, it abstract on abstract, Wikipedia team, we have the wikilean extension, the function orchestrator, and the function evaluator and seeing the function executor like, if there was a place that was doing the end-to-end tests, integrating all those together, it might make more sense for it to have its own. Its own rico though could be put into one of the repos for the tests but for one of the repos, one of the into one of the code repos though, what I imagine is the code repos test. So like you know functional orchestrators test will probably be like the unit tests for those For that project. And then the end-to-end integration tests, might still be its own repo?

Stef Dunlap: I'm going Cory.

Cory Massaro: so then this is really cool, but then following that question then does that mean that the workflow will be End-to-end tests. Do not run when you.

Stef Dunlap: Here.

Cory Massaro: On patches for the microservices. Like would we basically be like you know, the intent test fails at a certain Commit. And then you go in and figure out why and fix it. Would that be the workflow?

Stef Dunlap: I should you could do either but it should be possible and would be my recommendation that we do it on a perm. Merge request. So the you can have That you can have different repos, trigger different tests in gitlab, so and they can be configured to report back to our merge request from another repo, it's pretty flexible.

Cory Massaro: That last bit is really cool. I didn't know that.

Stef Dunlap: Like, seeing any other hand? So we may we make it out of here early, which would be funded. I'm trying to think if there's anything else that would be interesting to show.

Stef Dunlap: I guess, one thing that I didn't touch on that would be interesting for future doctors to know would be. I'm gonna share my screen again. 16.

Stef Dunlap: Backing up. I'm very recent question that you might ask yourself. Is Where is this Kubernetes coming from? So we already mentioned, like Garrett is run Garrett is run by a foundation. Gitlab CI, It's run by the Foundation duct tape. Just run on Pool Forge, which is a platform as a service but there isn't currently a foundation Kubernetes cluster. For the specific purpose of CI like we have one for production. We have some for specific projects but at this moment there's not one for like A random projects CI.

Stef Dunlap: If this became a pattern that more teams adopted that might change, we might make a managed Kubernetes cluster that different teams could share for this purpose but we don't have to let that block us today. It's very easy to set up a Kubernetes cluster in just a few commands. Using a tool are using a distribution called K3 F.

Stef Dunlap: Okay3s is just a distribution of Kubernetes that's meant to run on a single VM. And so currently what we're doing in In abstract, Wikipedia is we're using cloud VPs. So cloud. Wait like that, right? We're using a VM that we can get from wiki media. I think that service is called cloud PPS. And then, we are setting up a VM and putting K3 us on it. To make a Kubernetes cluster and then that is being pointed to by gitlab CI. So all the home commands and the Kubernetes configuration commands are going

00:20:00

Stef Dunlap: Into cloud DPs into our k3s cluster, which is just a single VM and it's easy to tear that down and bring it back up if it ever becomes unhealthy. So, I think if more teams started doing this, it would probably make sense. There was a foundation, run cluster, so that teams didn't have to worry about keeping this VM healthy.

Stef Dunlap: Yeah, go ahead.

Genoveva Galarza Heredero: About this comments of whether other teams would do this, have you come across other teams? That would be benefited from this kind of testing strategy? Or does anyone here know? I mean, I am aware that there's a lot of people that might have a lot of ideas.

Stef Dunlap: I'd be interested in if Vaughn or Peter or Rummana from Qce have any insight. I can say from like a very high level. There has been interest in from folks in Rowland and and like it, we are not It's pretty common for there to be an extension that relies on another service. Like We're not the only team that's feeling this pain point. And there's there's expectation in the like from

Stef Dunlap: There's expectation that some other teams might be interested in this. On Peter or Ramona.

Stef Dunlap: Yeah, go ahead.

Vaughn Walters: Um, I don't I'm not sure right now actually Stef. That's not,…

Stef Dunlap: Okay.

Vaughn Walters: that's a known answer, but

Vaughn Walters: I don't think so, but I want to talk to you more about it after this meeting though.

Stef Dunlap: Yeah, I know that you shall goes on The Qpe team. Had a team team or two in mind for this. He was pretty excited about it.

Vaughn Walters: okay, I can talk to him about it too,…

Stef Dunlap: Great.

Vaughn Walters: I've had I've been working with him on indent tests, so, but I'm not sure that for the project that I'm working, most with that, it would be That I'll use it.

Stef Dunlap: Yeah.

Stef Dunlap: Alright time. Now just give the floor a minute for any other lingering question.

Stef Dunlap: okay, thank you so much for attending and the this recording and slideshow will be shared out with With the newsletter for the community and internally for folks that are on our teams. Thank you very much for coming.

Meeting ended after 00:23:24 👋