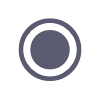
**David EV2 WhatIf Research-20251002\_151111-Meeting Recording**

October 2, 2025, 7:11PM

53m 17s

 **Saebom Lee** started transcription

 **D.M.**0:03  
It's like a ring one service, so we have to a Ring 0 service, sorry. And we have to deploy Event Grid at like quick paces and stuff like that for regions like new regions coming up and new clouds and stuff like that. So a lot and obviously EV2 is our main deployment strategy, right? So I believe and I joined Microsoft and.  
2020 and that was like when we had like like right after like the Event Grid team has switched off of like deployment engine onto EV2 as to use as our like deployment strategy. And when we originally used EV2 we did it like in these like I guess what they call now like classic releases. So you know we have like rollout specs.  
And stuff and I believe like the way that it was templatized was that like we have like essentially this like monolithic like very huge PowerShell script that then reads in these setting files and then creates like raw specs and roll out.  
Parameter like artifacts for our service group group to use and then deployed through EB2. So like a lot of times like the flow would be like OK let's we have like a new release coming up, we package everything into like whatever build and then we then do a release and.  
Used to be like OK, we'll submit for our own definition of like a role aspect of like ring one, ring two, rings ring four and five and then I believe like recently or like and and especially like a lot of the work that I had done was that in 2020.  
That was like back when we had this Jedi project and there was a huge push to integrate EV2 into like anything that would needed to be automated, right? So like ARM, like our manifest uploads were becoming a thing, like any ACES action that you would want to like run at like a steady pace would be becoming a thing like just even just writing.

 **Saebom Lee** 1:44  
Thank you.

 **D.M.**1:54  
So I saw like a lot of EV2 usage there. And then interestingly the thing I think like this is like kind of like the biggest pain point that I saw was that like we then everyone started switching to using agnostic rollouts, right? Which was fine, I thought like I definitely love the concept now that it was.

 **Saebom Lee** 2:09  
But.

 **D.M.**2:13  
All fleshed out, but it was just a lot of time to like build up the like essentially rebuild how we do that. And also like we had to maintain our classic releases because that is how we were doing a lot of our ad hoc deployments or you know, not even just ad hoc deployments, but just like you know, operationally.  
Maintenance stuff, right? So like our manifest and stuff like that. So like really like when that was developed, I felt as if that like the like a lot of the main use cases that we were using EV2 for was like kind of missed in that.

 **Saebom Lee** 2:46  
Mhm.

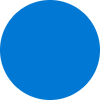
 **D.M.**2:46  
So, so now we do cloud agnostic for most of our like regular releases I believe and like new pipeline builds and stuff like that. And then we also do that for specifically region build outs, which were great because it helped us, you know, there's like a huge push at the company I believe to define our each resource in the EV2 way.  
And then you know like I believe like the region build out teams really pushing that out to kind of like use as like our like main driver to like drive parity across regions and build out these things and and and and like I think it helped us like like when I had first in my first region build out here it probably took like a couple weeks or so.  
And now I think it gets down to like you know on a like a very like good day like it it like in the EV2 way it can be done in like 8 hours if everything works it's like configured correctly which is like excellent. So yeah and that's just like you know like a a long story of like how how I kind of like see EV2 now and.  
And definitely like I think the stuff that like I'm kind of looking forward to with EV2 in general because I've I've been definitely monitoring the projects and stuff going on is kind of like seeing like how these you know there's like definitely a push like use I think these like build pipelines to then trigger rollouts and like.

 **Saebom Lee** 3:58  
OK.

 **D.M.**4:00  
Reach more of a CICD manner and I think like any integration to like CICD or like specifically like testing and rollbacks, uh, being better integrated would be cool. But yeah, that's like you know.

 **Saebom Lee** 4:14  
That is awesome to hear. So actually I had a list of like I I organized this list that I wanted to ask to you and you pretty much touched down on some of them. Angel, please chime in if you have any question like follow up question to David and Azima, please feel free to do so. I think you have written.

 **D.M.**4:16  
Mhm.  
Oh yeah, mhm.

4:31  
Mhm.

 **Saebom Lee** 4:34  
Mentioned on it, so you guys are pretty heavily on um region agnostic.

 **D.M.**4:39  
Yes. And I mean notably though we still maintain the classic release pipeline because so my team and you know like some the people who are in charge of release trains on Event Grid, they use EV, they still use like we push for region agnostic deployment.  
But the problem is right, like nobody is really trained on them except for us. So then like if there is somebody that needs to do a one off thing or something like that or like there's like an on call, a lot of them still, you know, even though you know like region agnostic was probably implemented on our team like in 2022.  
Or something like that or 2023 and there will be new people who have since joined like in 20242025 and they are using the EV2 classic still just because obviously it's just hard to teach. I think still after like a couple years and also you know, I think like I would say like.  
You know, releases aren't like a super, um, like interesting thing for people to wanna like, relearn. So uh, yeah.

 **Saebom Lee** 5:43  
Yeah, we wanna stick to whatever we are used to it.

 **D.M.**5:46  
Yeah, yeah, yeah. So, but yeah, that's that's that's like kind of like how our setup is right now.

 **Saebom Lee** 5:49  
OK, OK.  
Mm-hmm. And then you are using ARM template mostly.

 **D.M.**5:56  
Yes, yes, almost exclusively. Uh ARM templates and then uh Azure DevOps I guess.

 **Saebom Lee** 6:01  
Mhm.  
OK, OK, that's where that works. All right. I I remember that you mentioned that you are interested in like RBAC and like testing features also. Like do you currently use what if outputs to validate changes before deploying deployment?

 **D.M.**6:22  
No, no, I don't. I I I was not actually aware of like like I think like cloud formation and other things have like dry runs, right. So I mean that would be great. I did not know honestly because I and and that has been a requested feature that I would like kind of have mentioned to like EV2 support.

 **Saebom Lee** 6:26  
Mhm.

 **D.M.**6:42  
In the past I believe so.

 **Saebom Lee** 6:43  
OK. And then you're currently like for your testing then how are you approaching?

 **D.M.**6:50  
So when we like if we wanna test like a like for example one, right? I think for us it is kind of almost easy to test like new EV2 things in prod in a way in terms of like when we have a new region.

 **Saebom Lee** 7:07  
Mhm.

 **D.M.**7:07  
It is very easy to register news like region artifacts with the clip at which new regions are coming up. So generally right if we have like a new EV2 action or extend and and I would say like almost all of our stuff for the ring 0 obligation we have to deploy event grid.

 **Saebom Lee** 7:15  
Mm-hmm.

 **D.M.**7:27  
Is EV2ified and automated already, so we don't really have a lot of manual action yet now, but when we do that like there's like a new feature or like something new in the ARM template or something like that, we just test it in these new regions or.  
And in the past and that and that's what we have been doing in the past probably year, but like now we kinda just mostly do this in like Canary or we do it in like a test cluster or something like that.  
I I believe, um, uh, Angel, you have a question? Yeah, mm-hmm.

 **Saebom Lee** 8:01  
Yep.

 **Angel Perez** 8:03  
Yeah, just for the sake of time to make sure we cover what what we have, I'll ask you a couple questions just to see kinda how where do we take this. So just to level set your team's using region agnostic model.  
It's been fully migrated from classic to RA, but folks still maintain the classic pipeline just in case. Is that your point on that one? OK, reason I ask is because all the stuff that we're gonna be talking about new feature wise in EV2 is tied to this new region agnostic model, so it's good.

 **D.M.**8:25  
Yeah, mm-hmm, mm-hmm.

 **Angel Perez** 8:37  
In terms of ARM template versus Bicep, you mentioned ARM templates. Have you heard of Bicep before?

 **D.M.**8:38  
Mhm.  
No, I don't know what Bicep is actually. Yes, yes.

 **Angel Perez** 8:47  
At all. OK, cool. Then this is also like. I'll just give you a quick description of what it is, but happy to jump on a separate call to dive in deeper for that one. So Bicep is basically a new infrastructure as code language in Azure.  
Is built on top of ARM, but as you see ARM templates, it's basically like Jason with a bunch of embedded logic in it. You like you have these concat functions and a bunch of random stuff that was added to a language that.

 **D.M.**9:05  
Mhm.  
Mhm.

 **Angel Perez** 9:18  
Not necessarily was meant to be used that way. The JSON language, I mean. So in Bicep we basically took a stab of creating a totally new language that brings the best features and things that we are familiar with.

 **D.M.**9:23  
Mhm.

 **Angel Perez** 9:35  
As developers from more popular programming language like Python etc to a new native language that we support in Azure. So Bicep ends up being a much simpler template, not as many lines as those ARM templates.  
Get to kind of get out of hand with and then also it brings stuff like for loops, user defined functions and a bunch of other stuff that lets you really optimize for all you can do in your infrastructure as code.

 **D.M.**10:00  
Mm-hmm.

 **Angel Perez** 10:09  
So that's one thing and we can dive into Bicep later, but basically in the future there will be native Bicep support any V2. So what you see now in the service resource group definitions where you define you know each template.

 **D.M.**10:17  
M.

 **Angel Perez** 10:25  
You'll just be able to do dot bicep rather than dot JSON in the future. That's one thing. The relevance that there is that since we did a lot of work to kinda catch errors as far left as possible in authoring.

 **D.M.**10:31  
Mhm.

 **Angel Perez** 10:41  
Folks that use Bicep and by folks I mean our external customers that has have adopted this a lot see basically a 2% failure rate whereas ARM template is still like at above 10% failure rate.  
So yeah, that's one stat there. And then I think for everything else we we can probably move on. Have you heard of what if before?

 **D.M.**10:58  
Mhm.  
No, but I mean, I was, I mean, yeah, I I'm aware of like the if it is it like a dry run where you kind of like, you know, I just.

 **Angel Perez** 11:09  
OK.  
It's a preview API essentially. So we take the template, we basically get the state of resources in Azure and then compare them to the potential new state that would result if we were to deploy the template that you're running a what if on.

 **D.M.**11:19  
Yeah, yeah, mhm.

 **Angel Perez** 11:36  
And give you some kind of property diff before and after of what's gonna happen. The way it's designed or I guess the way Azure has been architected, what if has some issues of itself, but.

 **D.M.**11:41  
OK.

 **Angel Perez** 11:53  
That's a separate discussion as it relates to this one. The integration of what if into EV2 is something we're working on and the purpose of this call is just to talk a little bit about, hey, this is what what if looks like. Does this make sense to you?  
Since. Uh, so that's one thing. If we move on, I guess you haven't heard of stacks either. Have you heard of deployment stacks before by any chance?

 **D.M.**12:09  
Mhm.  
Uh, no, no, I have not.

 **Angel Perez** 12:20  
OK, cool. So lots of new stuff being thrown at you today, which is good. You can see the teams have been investing in better tooling for for folks. Saebom, maybe we can just move through the slides and I can say when we can stop.

 **Saebom Lee** 12:28  
But.  
Yeah.

 **Angel Perez** 12:35  
Just to kind of see or actually you can walk through this one. This is good information for us to kind of talk through.

 **Saebom Lee** 12:40  
Yeah, OK. All right, David, here are some other topics that I want you to like hear what you think around and what your team is devoted to. So it's this is to understand your deployment like.  
Workflow, especially regarding the deletion scenario. Let's start from the first thing. How often do you need to delete resources in your team? Is it monthly?

 **D.M.**13:10  
I mean, I guess like in what context? I mean like there's obviously like SFI items I feel like where resources were created wrong and they need to be deleted or fixed, right? But I would say like we don't have a lot of zombie resources or there's not a lot of orphan resources that are just like.

 **Saebom Lee** 13:19  
Yes.  
Mhm.

 **D.M.**13:30  
Like not, um, there or like not supposed to be there, right? Is that what you guys are talking about?

 **Saebom Lee** 13:35  
Yeah, just to like understand how how much you engage with like deletion or like do you even delete or if you are deleting is like manual or automated?

 **D.M.**13:50  
I would say like a lot of our deletion would be, yeah, it would be manual because like it's more like on call responsibilities or a lot of times like what we deal with is like.

 **Saebom Lee** 13:50  
Some wine.

 **D.M.**14:05  
I mean, I think a known thing in Azure is that like resources are flaky, right? So the common practice, a common practice is to delete them and then just recreate them and try to see if the error replicates for. Yeah, and and that happens very often of like.

 **Saebom Lee** 14:19  
Yeah.

 **D.M.**14:22  
Specifically VMSS resources like they are very flaky in my opinion, so they have a constant need to be deleted and then recreated because nodes will like hardware will just come up bad. So mm-hmm.

 **Saebom Lee** 14:32  
Yeah, got it. So like for the SFI items or like any deletion that you are manually doing, like are you using JIT today?

 **D.M.**14:43  
Yes, yes. And it's always attached to like an ICM of some sort, right?

 **Saebom Lee** 14:49  
OK, OK, sounds good. Um, is it different from like testing um and prod to delete resources?

 **D.M.**15:00  
So like we have a subscription checkout system, right? So people can check out stuff and I will say like there were probably like orphan resources there where like people are just messing with their subscriptions and then they kind of forget about them or.

 **Saebom Lee** 15:06  
Isn't it?

 **D.M.**15:16  
You know, they flag or something like that. So then they go delete them, but they don't. They only delete the offending resource, not necessarily like the orphaned ones. So yeah, that's that's kind of like our and I believe like we recently have a new team that we've spun up in India and a lot of their goal is to.  
Create an entire test environment set up for reach agnostic deployments. So I mean I I like that they would maybe be also good to talk to if you guys are looking for people. But yeah they have they they are specifically doing test environment life cycles and that's we we have performance testing. We also have stress testing.

 **Saebom Lee** 15:43  
Oh, yeah.

 **D.M.**15:53  
That's done nightly for our stuff, so.

 **Saebom Lee** 15:56  
Mhm. OK, OK, um.  
All right, so you guys are taking like extra care for the lady, I guess. So have you ever encountered any incidents or like instances like where your team or the other team that you just mentioned or some other teams like made changes and then that caused some unwanted changes in return?  
And.

 **D.M.**16:21  
Yeah, yeah. I think anything manual obviously has like problems, right. And it's not checked in. I'd say like there's been a new push recently in the company to do like R2D ready to deploy. So a lot of these changes are tracked very closely or and I I would say there's like.

 **Saebom Lee** 16:36  
Yeah.

 **D.M.**16:41  
A lot of pressure to not if it is a code change like you know like we have to be very vigilant in like recording these changes in configurations and stuff like that. So I think the mitigation for that has been more so like.  
Rcas and stuff like that for sub twos that have been had follow up at our like VP level.

 **Saebom Lee** 17:05  
OK, then how do you prevent those changes? Like do you use like Azure policy or also like what RBAC?

 **D.M.**17:14  
Um, I guess like you mean like how do we stop the manual changes from happening? Uh.

 **Saebom Lee** 17:18  
Yeah.  
Yeah, we're trying to do like anything to prevent it.

 **D.M.**17:23  
We have no, I think there's probably a lot of pushback to like put like guardrails on it as more so like there's more push to just be like, OK, like if you are on call right and you have to do a manual change for an incident then.

 **Saebom Lee** 17:31  
Mhm.

 **D.M.**17:42  
Make you know you you like there is a more of a check up like did they check in that change right. But there is no we don't have any more guardrails in like \*\*\*\* or something like that or like I think.

 **Saebom Lee** 17:50  
Yeah.

 **D.M.**17:58  
Like I know for a fact like the VMSS resources, you cannot just delete them now anymore. You have to submit like ASUS actions or like requests at their production and live and stuff like that. So like when we do other things like.  
Like the resource creation or resource clean up for like migrating SKUs, which is like like there's a huge push to change things. So like zonal SKUs then that's when we have had to like do a lot of tickets and tracking and like I think you have to get RTD requests for those and stuff.

 **Saebom Lee** 18:29  
Mhm.  
That makes sense. OK, Angel, Azima, do we have further question to David before moving on? OK, sounds good. So yeah, thank you, David. Now we think we understand your workflow much better.

 **Angel Perez** 18:41  
Nope.

 **Speaker 1** 18:42  
I think I might end here.

 **Saebom Lee** 18:52  
So from now on we have a design stimuli that we wanted to use to talk to you. It's a it's a design mock up. But before showing you that, I would like to set up some context with the help of Angel here.

 **D.M.**18:59  
Mhm.  
No.

 **Saebom Lee** 19:08  
Zhang Yao, if you're ready, I think you can take it over.

 **Angel Perez** 19:10  
Yeah.  
Cool. So I guess First things first. What we're seeing here David, is the output of a what if, but more specifically a stacks what if. So the pre context of this is.

 **D.M.**19:21  
Mhm.

 **Angel Perez** 19:26  
A new resource type called Deployment Stacks, but since you haven't heard about it yet, um, I think maybe I could quickly walk through this and then we can jump back to this view and get your thoughts.

 **D.M.**19:39  
Yeah, that that sounds good.

 **Angel Perez** 19:42  
Let me share my screen real quick.  
And we can dive right in.  
Um, one second.  
Right, so.

 **D.M.**20:12  
Mm-hmm.  
Me.

 **Angel Perez** 20:14  
So bicep, as you can see, it's basically a little bit cleaner. So this is what a resource definition would be like. You see it's a lot less lines and a little bit more readable than the Jason thing we were dealing with before.

 **D.M.**20:26  
Um.  
Mhm.

 **Angel Perez** 20:28  
I won't dive too much into the functions and all the things possible, but you can just see at a glance like it's a much cleaner modern language. So here's my bicep file, kind of describing a couple things that I have.

 **D.M.**20:37  
Mm-hmm.  
OK.

 **Angel Perez** 20:44  
Um.  
Here I had a couple of storage accounts. I'm gonna comment this back out. So ARM templates. We would either use the CLI API or PowerShell. Folks can also just click through the portal as well. No problem with that.  
But basically before it would be a new AZ subscription group subscription deployment and now it's just deployment stack.  
So deployment stack, you can think of it as the evolution of deployments, but it has two new features it brings into the mix and also the concept of the stack managing the resources it deploys. So for now I'm just going to quickly walk through this. Here we have this file.  
File that creates a bunch of different resources across different scopes, uh across different subscription scopes, and the target scope of this deployment is subscription.  
So all of the resources defined in this file are managed. They're considered to be managed by the deployment stack resource. So different than the regular deployment, this deployment stack resource is an actual new resource type that has its own resource ID. So you could use RBAC to like block folks from you.  
Even see in this deployment stack, I'm calling this deployment stack here AP Test 02, but let's jump to the portal to see all this a bit more visually as that deployment runs.

 **D.M.**22:19  
Mhm.

 **Angel Perez** 22:20  
And feel free to ask any questions if you have them as we walk through this.

 **D.M.**22:25  
Is there like do you guys have a tool to convert ARM templates into bicep like or like bicep files?

 **Angel Perez** 22:35  
Um, there.  
I think we have a compiler of sorts.  
I can point you towards that. The bicep at the end of the day compiles to to ARM templates as well.

 **D.M.**22:49  
Hmm.

 **Angel Perez** 22:50  
I think an interesting thing with bicep, it's a good opportunity for your team also to decide what kinds of components you want to modularize. There's a concept in bicep of modules kind of aimed to replace and improve upon what link templates used to be.  
So that this is also in that reauthoring phase that y'all may want to do or not do. That's one consideration I can see.

 **D.M.**23:22  
OK.  
Yeah, because I think like or or just my overall thought is like like I will probably try to use Bicep for new ARM templates, right? And maybe there are like you know, oppressive ARM templates that are kind of like you know, like that would it would be well suited for to replace.

 **Angel Perez** 23:23  
But for.  
Mhm.

 **D.M.**23:42  
But then I think it it would be very hard I think across you know at least some of the teams I know to change them to use Bicep if they are not like either forced to right in a way or you know they there's no tool to like like I think the hard part is like you have.

 **Angel Perez** 23:43  
Mhm.

 **D.M.**24:02  
You have to understand the new tool or the new file, but then you also have to like convert it over and like a lot of the people who have made and then are charged with it. Like I own a bunch of ARM templates I guess, right? But I don't necessarily know how they were made because this was done like with multiple people across, you know, multiple files of a long time ago.

 **Angel Perez** 24:05  
Mhm.  
Mm-hmm.  
Yeah, yeah, yeah. In terms of migration compilers, we have that, but really it's it's meant to be an either or like if folks want to keep using ARM templates, we're not gonna block them there. We support both, but it's a significant quality of life improvement for both the maintenance and net new authoring as you mentioned.

 **D.M.**24:22  
Yeah.  
Mhm, mhm.

 **Angel Perez** 24:39  
But yeah, we can dive into Bicep more specifically in a separate call as it relates to deployment stacks. You can pass in a Bicep file or an ARM template file dot Jason. It doesn't really matter.

 **D.M.**24:39  
Mhm.  
Interesting. OK.

 **Angel Perez** 24:53  
So like today you'd be able to run that command that I showed the same way. You you could just basically you use the same exact template you have today. You don't have to tweak anything in it, you just the way you create it has to be with stack amendments rather than the regular deployment ones.

 **D.M.**24:58  
Yes, mm-hmm.  
Mhm.

 **Angel Perez** 25:11  
And what that ends up doing is it creates this new resource. So this is an actual deployment stack resource. As you can see here, it has its own ID settings specific to it, but that's pretty much it.

 **D.M.**25:12  
Mhm.

 **Angel Perez** 25:27  
This resource has a concept of managing other resources, so all the resources defined in that template are managed resources. The first property that stacks brings into the mix is action on unmanaged. This basically says if a resource were to become unmanaged or.  
The association between it and the deployment stack resource is broken. The stack has permission to perform this action at this scope, in this case delete. For example, here I have all of these resources, all those storage accounts.

 **D.M.**25:57  
Mm.

 **Angel Perez** 26:05  
And this is across multiple resource groups. If an SFI incident comes in and it calls out a specific resource that you can just control find in your in your source, you could go back to where we were. Let me just show this code real quick.  
And I say, you know what, I don't need this.  
First instance anymore because of SFI or because we're testing something or because what have you and I just rerun this. So I call out here that this setting is set to delete all. That means when the stack sees in this next deployment oh.  
OK, this resource that was previously defined is no longer there. That means it's no longer managed. What's my permission for action on unmanaged?  
So I'm gonna run that and basically you could see the value in this as two twofold. For huge testing environments you could just clean up the whole thing in one go. So rather than having to be one to one per scope, it's one to many.

 **D.M.**26:59  
Um.

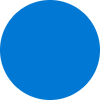
 **Angel Perez** 27:15  
You just delete the stack and the stack takes care of fanning out that operation across all the resources it manages. Or you can be a little bit more surgical and see you know this is the exact thing I'm looking for that I need to get rid of.  
And that's what I'm going to remove rather than doing it through the portal with something like JIT, which might be a little bit more error prone. This integration into EV2 is kind of born between the fact that stacks is already GA and available to third party customers.  
And growing in adoption and also the fact that this past year there were like 100 some odd outages related to folks deleting stuff that they weren't supposed to. So here we're providing a new mechanism to perform that delete. It's a bit safer.

 **D.M.**28:03  
Mm-hmm.

 **Angel Perez** 28:08  
But also can scale and and help with that test environment tear down very quickly. So I'm gonna go back to the portal just to show this final result and then the very last feature and then we can jump back to.  
The slides.  
So this is running. It's deleting resources at the moment. So basically the stack is gonna go ahead and remove those other resources as that runs. I'm gonna show you very quickly this other capability stacks has. So besides action and unmanage, we have deny settings mode.  
Deny settings mode basically lets the stack create a deny assignment on each of the resources it manages. So there's two types. You can either deny, delete or deny, write and delete. This is where you can add an extra level of protection.  
To these environments you're provisioning to teams to just make sure they don't make any unwanted changes that may cause problems.  
And I'll just show you this in action very quickly. I and my user has access to all of the scopes, all of these resources, so I can go ahead and attempt to delete this Venet for example.

 **D.M.**29:19  
E.

 **Angel Perez** 29:37  
And I do have the permissions over it, so I should be able to go through with this. However, since the deny assignment is added to this by the deployment stack, I'm gonna get failed instantly. It's gonna say hey, there's.  
You do have access to this resource, but this other resource called deployment stack. What is preventing you from doing so?  
So those two features are what stacks brings to the table and all this context just leads to well, one stacks is being integrated into EV2 and we can preview that for you next month. If that's something your team would be interested, we can add you to the preview list.  
And so the reason for wanting to move to stacks is because only in stacks is that we're going to have stacks what if which is getting integrated into the the portal EV2 portal. So just to round this out, you can see here these resources indeed ended up getting deleted and these.  
Are still managed, so I think we can jump back to the the slides, Saebom.

30:52  
All right, uh, let's see.

 **D.M.**31:05  
Yeah, yeah. And I I think our team would be interested in potentially trying out a preview. I think like especially if if this is something we could explore maybe when they have all these like you know, CCLAS and stuff like that and there's, you know, obviously not as much deployment to be done, right. The do some trial testing would be cool.

 **Saebom Lee** 31:18  
Mhm.

 **Angel Perez** 31:21  
Yeah.  
So the the biggest thing as it relates to this discussion is let's assume your team made that move of just creating deploying those ARM templates you already have of stacks as stacks rather than creating them as regular.  
Deployment objects. These are created at stacks. Now you can run a stacks what if on them and basically what that does is just previews. Like I said, what is going to happen as a result of the changes you just made to the template.  
So here for example, if you see at the very bottom we have a couple of updates to a resource group, update to a virtual network, and within that you'll see the property diff of what's actually happening to that resource in the before and after sense.  
At the very top we see net new like something's getting created, in this case a deployment stack. So here you get the stack specific property information. What we're trying to figure out here is 2 things so one.  
The property diff everything shows under the resource. That's the existing what if works that way. But now we have these new stack related statuses that are technically not properties of the resource, but they are.  
Information of of that resource's relationship to the stack resource. So that's where you can see that at sign stack dot managed status for example, or stack dot deny status.  
And just information on that resource. Um.  
Does that make sense to you like calling those out separately than like actual properties that belong to the resource?

 **D.M.**33:22  
Yeah, I mean, it's just like, um.  
It's it's like it's like Terraform bit right or something. Is that a fair comparison?

 **Angel Perez** 33:30  
Yeah. So Terraform plan, are you familiar with that?

 **D.M.**33:33  
Yeah, yeah, yeah. I like very loosely, right?

 **Angel Perez** 33:35  
Pretty much. That's where a lot of that inspiration came from. In Terraform, you kind of have a state file. We don't have that in Azure, but you can somewhat think of the deployment stack as the state of that deployment.

 **D.M.**33:43  
Mm-hmm.  
I see.

 **Angel Perez** 33:52  
Because it does track the, you know what resources are quote UN quote managed. But yeah, so so relevant to the stack, it's just the status of is this resource managed or not? Is it going to become unmanaged? Is it going to become managed? And is there any changes to the deny assignments related to that?  
Yeah.

 **D.M.**34:13  
Mm-hmm.

 **Angel Perez** 34:13  
And then the ones at the very top are just specific to the stack resource itself. So the deny setting mode mode, that's a property of the stack resource as well as all the other stuff listed out there. Action on unmanaged would be one is actually missing from this mock up now that I notice it.

 **D.M.**34:33  
Is there a way to like, I mean, I think I'm like more interested in like the manage unmanaged. Yeah, because like I don't, we don't really have problems with like deleting resources in terms of, I mean like any SFI item or anything like that. We generally do roll that out via EV2.

 **Angel Perez** 34:40  
Mm-hmm.  
OK.

 **D.M.**34:52  
You know, because and and that's not necessarily because like, you know, we're we're trying to be safe in a way. It's more sort of like we have to do this for like, you know, 50 something regions or something like that, right. So it's just like, yeah.

 **Angel Perez** 34:53  
Mhm.  
Yeah, and this would be the the same thing, right? Because it's being, I guess with this text integration that can be done across all those regions with that change you're doing to to the template.

 **D.M.**35:07  
Mhm.  
Uh, interesting. And and it was how? How does the stack scope like regions or resource groups or like can you define those?

 **Angel Perez** 35:25  
The the stack has the same scope of the deployment, so it's like again an evolution of the deployment object. So a stack can go to the resource group, it can go to the subscription scope as well as it can go to the management group scope.

 **D.M.**35:29  
I see. I see.  
Yeah.

 **Angel Perez** 35:41  
So basically all the templates you have authored today can just be recreated as stacks right from the get go, but then the the determination of what do you want to be stricter on with the deny settings or you know what kind of delete you want to perform.  
That's where your team would have to reflect and decide OK, what all resources share a similar lifecycle such that we might want to consider putting those in the same stack. But for example here I am in the Azure portal again.

 **D.M.**36:13  
I see.

 **Angel Perez** 36:19  
And I can go to the subscription.  
And you're familiar with deployments. So here in deployments you can see these and these are all just records. And if you use the same name, this ends up overriding with that latest template. But now you have deployment stacks in settings for each scope as well. And this has all of the list of deployment stacks, which is that new resource type.

 **D.M.**36:28  
Yes.

 **Angel Perez** 36:42  
That.  
Does a deployment on your behalf. So at the end of the day, if you click into a deployment stack, I can still find the deployment that was created by it. We get the same deployments object, same everything, but that other layer.  
Uh, within the stacks UX.  
Has all the managed resource information and all the other stuff.

 **D.M.**37:10  
Mhm. Mhm.

 **Angel Perez** 37:13  
Um.  
So yeah, I think this would be worthwhile, but for this topic specifically was more. All right, what if it's for previewing stuff? So any V2?  
The main thing we want to catch very seriously is will this change I made to the template cause a delete? Yes or no? And for that preview that you get be something that you can quickly act on and say yeah we can move on or.  
No, we shouldn't be deleting this at all. This will cause problems. So if you go back to the image, Saebom, you can quickly ask about how well.  
Deletes are surfaced in this output, if at all or not. Um.

 **Saebom Lee** 38:00  
Yeah.

 **Angel Perez** 38:01  
Or sorry, not in the stop, but in the EV2 portal mockups if you wanna jump to to that.

 **Saebom Lee** 38:01  
No.

 **Angel Perez** 38:09  
Next slide, probably.

 **Saebom Lee** 38:09  
So.  
OK.

 **Angel Perez** 38:15  
Yeah, so you want you wanna talk through this one?

 **Saebom Lee** 38:19  
Oh yeah, I can, I can, I can go and then you can chime in. So like David, so from now on we're going to show you like 3 to 4 pages and on those we would like to hear just just like what Annie just mentioned that we would like to see like where would be the best place.  
for you to see the what if results on the EV2 portal and then what are your expectation on the EV2 what if results or any relevant information? What would be a good scope of that information to be listed on EV2 portal for you? Um  
um What we're thinking is that we wanted to add this what if result tab to have the information from this screen. If you were to dig into the deeper level level of what if results information, then what where or how would you like?  
To do on the screen, what would be the best way to approach navigate to that information?

 **D.M.**39:25  
I guess if it were like expanded upon like the deploy thing and like it was just like a different view where I could see it like line item by line item because.

 **Saebom Lee** 39:33  
Hmm.

 **D.M.**39:41  
If I is this is this a picture or is it this is like a actual clickable thing? OK, OK, OK, cool. Yeah, so so if I were to like click on I I generally have it so like I look at the list.

 **Saebom Lee** 39:47  
It's a it's a screen.  
Mhm.

 **D.M.**39:57  
Right. So sometimes I look at the graphical stuff just because I want to see like at a higher level, like visually like how far right have we moved from left, right. But then on like I generally use a list and I generally expand the drop downs and that's what I mostly look at. So it's kind of nice to see.

 **Saebom Lee** 40:06  
Yeah.

 **D.M.**40:17  
Like, I mean yeah, this would be nice too, right? I think if like it just shows the drop down lists and everything and but if if it was also somewhat like how would it look when it's in like motion I guess like does it just go away?

 **Saebom Lee** 40:36  
Got it, got it. All right, so this is which we have prepared a way to move forward from this screen and I want to know like which one do you prefer from that?  
Navigation way versus what you just described. I'm not the alternative way that we prepared is to click the deploy and then you get the new pane open.

 **D.M.**41:04  
Hmm, I see. No, I think I would like the the first one would be better.

 **Saebom Lee** 41:11  
OK, in the in the list view.

 **D.M.**41:14  
Yeah, I think, I think it's and specifically I kinda like that just because you know when people troubleshoot this, they don't they, you know, they're not aware of the, I think like the.

 **Saebom Lee** 41:17  
Mhm.  
Yeah.

 **D.M.**41:30  
Overall how deployments work in general, so like they they would not necessarily take the click if that makes sense. So yeah.

 **Saebom Lee** 41:35  
Got it, got it. Yeah, that sounds a lot of sense. I do have a follow-up question then. So like if you open up a new panel like here you can see there are further information. I have like questions regarding the type of information that's listed in this pane, but do you also expect to see the similar?  
Uh, information in the list format like this tabular format too.

 **D.M.**42:00  
Yeah, I like my interpretation of like these things is just that like when I click on it right like it should it should show me like like what I clicked on and everything on that page and also just like it's just a larger screen, right? So then like in the larger screen it should just I should be able to like.

 **Saebom Lee** 42:05  
Mhm.  
Yes.

 **D.M.**42:20  
Have more details on like exact. Does that make sense I guess?

 **Saebom Lee** 42:25  
Yeah, then what are the types of information that you're looking? So like for now we only have these new four column items. Um.

 **D.M.**42:34  
I I'm I'm looking on like, well, one like I think a big thing I check a lot of times is is because we're running the same rollouts. So a lot of the things that I'm flagging here is like, oh, like how long is it taking specifically?

 **Saebom Lee** 42:40  
Mhm.  
Yeah.

 **D.M.**42:52  
So I I liked like I think like the original EV2 portal shows you like how long a rollout is taking and I and some I think it does do it on reaching as well. I just haven't looked at it like or don't have a top of mind but the so like like you know for example like if a VMSS is taking like 8 hours.

 **Saebom Lee** 42:52  
Mhm.

 **D.M.**43:10  
Then I just know, oh, it's probably gonna time out. And that's like the issue I have to look at for the day, you know, or like, yeah, so and then I I also just generally look at like what what, like what? What are the next steps that are gonna be run, right. So.

 **Saebom Lee** 43:16  
God.

 **D.M.**43:28  
I kind of like that idea with like, um, you know, the graphical thing is nice just because you can't kind of see that, right? Um.

 **Saebom Lee** 43:36  
That's true. Do you mean like any kind of like notification for the next steps?

 **D.M.**43:41  
Um, like like notification of the next steps? Would I like that? Is that what you're saying?

 **Saebom Lee** 43:48  
Yeah.

 **D.M.**43:50  
I think, I think that would be kind of nice to like have a view or a filter for that, you know, 'cause.  
The a lot of a lot of the issues of like I think how these rollouts go is that they the big push was to put everything in regionnostic and do like SDP and everything. But like if you it just very much depends on the team and the rollout and like some teams have done it so that each each thing is one step right?  
So they did reach an agnostic, but really they have really micro serviced out their deployments. But then other teams have done it like and our team notably has like done like a kind of mix of this, but like we've done it where we put everything all in one template.

 **Saebom Lee** 44:29  
Yes.

 **D.M.**44:34  
Right. So it becomes super unwieldy and it just it's just yeah, yeah. So I need any filter I and I I have used the filters before but I just haven't like you know I think I don't think there's like any presets really right. So.

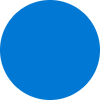
 **Saebom Lee** 44:38  
Yes.  
Got it, got it. OK, before we move on to the next screen, I want you to focus on this column here, like create, delete, group, delete, and modify. Is it? Are they useful information for you?  
Or or is there any other information that you would like to see up front here?

 **D.M.**45:10  
No, it it looks useful I think. What is what is the group delete kind of mean compared to delete?

 **Saebom Lee** 45:21  
Azima, please feel free to chime in. It's a My understanding is Angel, it's a book, right?

 **Angel Perez** 45:27  
I think that's a typo. It would be under the same delete column. I think resource groups getting deleted will be listed out as resources just like other resources.

 **Saebom Lee** 45:31  
Yeah, OK.

45:34  
M.

 **D.M.**45:40  
OK. Yeah, then that then that looks fine then, right. I think definitely right. Like any like like CRUD like update type thing, you know like it's kind of like what I look at for deployments, right. So like.  
You know, like create, update, delete, you know type stuff, so.

 **Saebom Lee** 46:01  
M.  
Um, then what is your understanding on this part here this for?

 **D.M.**46:08  
That that if I were to click one of those it would show it would filter out those resources.

 **Saebom Lee** 46:09  
Stop.  
Mm-hmm. Is that the filter that you were mentioning or?

 **D.M.**46:19  
Yeah, I guess, but more so like I I kind of look for stages of my deployments. So that those are the filters that I'm like more interested in, right? I think like, oh, like how like you know a lot of the times I want to look at like when does ring 2 finish?

 **Saebom Lee** 46:25  
Yeah, yeah.  
OK.

 **D.M.**46:39  
Finish and then like what is like failing for ring three or you know, like once it gets blasted out to a bigger radius then you can kind of see what the issue is. Um, because then I prepared those, yeah.

 **Angel Perez** 46:49  
I think for what if that you're might be confusing here 'cause again and what if nothing ever gets deployed?

 **D.M.**46:57  
But would.

 **Angel Perez** 46:57  
You're seeing a preview.

 **D.M.**46:59  
Would would it not show as the deployment is going? Would it not still show it or no like so like if I'm like in ring 2?

 **Angel Perez** 47:05  
You would see it, but it's a. It's still a record of that instance when you ran it in the particular point in time. It's not like a running thing that that would be more like.

 **D.M.**47:08  
Um.  
I see. I see.

 **Angel Perez** 47:20  
As your rollout is progression, you you get that information for the rollout. What if can be used to determine if we want to start the rollout or not? And that's kind of what we're getting at here with, you know, is this summary of change type and resource name.

 **D.M.**47:35  
I don't actually. I need to be with this.

 **Angel Perez** 47:39  
A good way for folks to determine if things should proceed or not, and what does that UX look like in any way, shape or form? But there's no like a what if won't be constantly evaluating as the rollout's progressing.

 **D.M.**47:55  
Mm.  
OK, well then.

 **Angel Perez** 47:58  
It's a thing that happens before you say, yeah, I wanna deploy.

 **D.M.**48:02  
Then yeah, I mean, yeah, this, this kind of, I guess in the lens of that these are the only things I guess. I mean I I would like, I guess would you be able to see like specific properties being changed here?

 **Angel Perez** 48:18  
That we have in the Azure portal, so if you open up the next slide, Saebom.  
Basically similar to no. You can go to the previous one. Here you have that summary, but if it's supposed to be color blue, but you can see here there's a Uri to the what if result.  
That is a link to the full result. So you would then jump to the Azure portal that has the full context of property diff across all of the resources present in that what if.  
The reason we kind of separated these is cause there's no RBAC in EV2 and although what if does ignore stuff that's marked as secure string in manifest, some RP's might not do that correctly and we don't wanna leak any secrets accidentally from.  
You know, teamed across teams because EV2 doesn't have RBAC. So we have this kind of broad summary of change type and resource name in the EV2 portal and then a link to get the full context of property diff in the Azure portal.

 **D.M.**49:18  
I see.

 **Angel Perez** 49:31  
So yeah, there's a couple slides still missing for what that experience looks like in the Azure portal that Azima's working on. But if you go to the next slide, say, that link will basically land you here where you just get to see all the different resources.

 **D.M.**49:32  
Yeah.

 **Angel Perez** 49:47  
And clicking on a resource, you'll get a side pane from right to left that shows the JSON object diff for that particular resource. So you could just go very specifically to the resource you want to see.  
And see you know if it's a modify, like what properties are actually getting modified and so on.

 **D.M.**50:11  
OK. I mean, yeah, I think, I mean definitely like, um.  
The most useful thing about what if to me or like just off the top of my head is like to see what properties are being modified or what new ones are being created. So you know any any way to like obviously just.

 **Angel Perez** 50:27  
Mhm.

 **Saebom Lee** 50:28  
Mhm.

 **Angel Perez** 50:29  
Yep.

 **D.M.**50:34  
Get to that, get that information is is what I kinda think would be like best.

 **Angel Perez** 50:39  
Yep, well, I I think it'll be varying from team to team. Like my guess is most developers are just gonna want to use the CLI and parse that whole response however they want.

 **Saebom Lee** 50:42  
Go ahead.  
Mm-hmm.

 **Angel Perez** 50:55  
But if you're, you know, investigating things through a rollout, you can click through it and get to it very easily as well from within the AV2 portal.  
But in terms of like automating any kinds of pipelines around this result, I do suspect folks are still gonna make use more of the CLI commandlets for UV2.

 **D.M.**51:05  
Mhm.  
Yeah, no, definitely. I think that sounds very interesting.

 **Angel Perez** 51:22  
Cool. Well, this has been great, David. We appreciate you taking the time to chat with us and answer all these questions. If you are interested in learning more about Bicep, we can definitely set up some other time with my team and walk through that.

 **Saebom Lee** 51:26  
Yes.

 **Angel Perez** 51:39  
And as well for for deployment stacks, if you're interested in the preview, you can let me know and we can have your team on the list for when that becomes available next month.

 **D.M.**51:50  
No, yeah, I think if you guys have any docs or stuff like that to you could forward me, that'd be cool. So I could just review it more and then yeah, yeah, it it all sounds neat and interesting. I think like like once again, like my.  
The adoption of it for my team, like I think the main issue right is the migration portion, right? So if there's the tooling on that would be interesting or you know, I think like that.

 **Angel Perez** 52:20  
I think for for bicep really you won't need to like to use stacks. You don't need to migrate anything again, it's just a small change to your service resource definition, but all ARM templates and everything stay the same.

 **D.M.**52:23  
Yeah.

 **Angel Perez** 52:36  
But yeah, I'll send you some docs just so you can kind of review the concepts a little bit more and think about, OK, is this just a we leave the service resource definition structure as is and all of these become stacks or?

 **D.M.**52:36  
Uh.  
Hello.  
Yeah, I have. I have like, uh, I have like.

 **Angel Perez** 52:53  
Are there any resources that need to be moved around from one template to another so they can be part of the same stack?

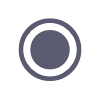
 **D.M.**53:01  
Cool. Well, thanks. Thanks. Thanks guys. I appreciate it.

 **Saebom Lee** 53:03  
Alright.

 **Angel Perez** 53:05  
Awesome.

 **Saebom Lee** 53:05  
Thank you so much, Steven, for your time, and thank you, team, for the discussion. All right, then, talk to you later.

 **Angel Perez** 53:15  
Yeah.

 **David Moy** stopped transcription