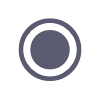
**JR EV2 WhatIf Research-20250930\_130656-Meeting Recording**

September 30, 2025, 5:06PM

53m 29s

 **Saebom Lee** started transcription

 **JR M.**0:03  
Might get deleted. What will get deployed et cetera.

 **Saebom Lee** 0:05  
Yeah. OK. Sounds good. Sounds good. So I'll record this call because this is kind of considered to be like our user testing. And then for me to just like listen afterwards, OK.

 **JR M.**0:13  
Mm-hmm.

 **Saebom Lee** 0:20  
So I think we can just like kick ahead before folks joining. So what, what team are you in and then how are you using EV2 recently?

 **JR M.**0:34  
So right now I am on the Azure messaging team, specifically within the USEC usnat clouds. So I kind of branch between like service bus and event hubs currently, but at the time of all the EV2.

 **Saebom Lee** 0:45  
OK.

 **JR M.**0:52  
Stuff I was doing it mainly on under event grid and that's yeah, that's primarily where a lot of the EV2 deployments. I I I did came from.

 **Saebom Lee** 0:58  
OK.  
OK, OK, alright. I would like to share my screen just to help us move forward. 'cause, we only have 45 minutes and then I would like to cover as much as I can. Let's see. Oh, that's nice. Our photos, except mine here.

 **JR M.**1:08  
Mm-hmm.

 **Saebom Lee** 1:22  
I have these four topics that I would like to ask about your current usages, so like EB2 classic versus region agnostic like, can you describe whether your team currently uses either way for deployments.

 **JR M.**1:36  
Yeah. So, EV2 region agnostic primarily is used for like our build outs at least on the event grid side has been for like cloud and regional build outs for classic though that's primarily for like updating our apps.  
And then things are moving towards the region agnostic side that just haven't been completed yet or like fully migrated.

 **Saebom Lee** 1:59  
OK. And what are the reasons it's still like migrating?

 **JR M.**2:06  
Pretty much just like cycles time testing. So like I came up with like a migration plan and then kinda got pulled off that task for a little bit but it was it was the migration plan was typically.

 **Saebom Lee** 2:09  
Yeah.

 **JR M.**2:19  
Kind of like a what? If So what I was doing was I our repo had a couple of configs where you would make changes to and it would generate the rigid agnostic artefacts for it.

 **Saebom Lee** 2:21  
Mm-hmm.

 **JR M.**2:34  
And you know how, like a lot of those come from, like, roll out configurations and like find and replace scope bindings that made our that made the team a little bit nervous as to like, all right, this is a whole lot of changes all at once. So like what I was doing was.  
I wrote a PowerShell script that would go and get like their service fabric cluster. So pretty much app parameters, dump them and then I would do a rigid agnostic deployment for app deployments, right? So app updates.

 **Saebom Lee** 2:57  
Mm-hmm.

 **JR M.**3:06  
And I would just tweak 1 parameter and then I would do get on all the app parameters again for service fabric and then the only thing I should see change is that one parameter right? So and I was successful in that, but then kind of got pulled off that test for a while, so it hasn't been picked up since.

 **Saebom Lee** 3:15  
Mm-hmm.  
OK, OK, got it. Hi, Azima and hi, Anil. Please chime in if you have any questions or JR, if you have like questions for us.

 **Speaker 1** 3:28  
Aye.

 **JR M.**3:30  
Mm-hmm.

 **Angel Perez** 3:32  
Awesome. Yeah, sorry for joining a bit late. I was having some network issues. They are a quick question on that. So when you're saying you get all the gets, are you just comparing the like a DIF on the?

 **JR M.**3:43  
Mm-hmm.  
Just a.

 **Angel Perez** 3:48  
On what you're getting from the API itself.

 **JR M.**3:50  
Yeah, it's just from the service fabric app parameters essentially. So mainly application parameters. So like a lot of our lot of our so like we would build out the whole region out of region agnostic. But then when it came to like supporting it, so like.

 **Angel Perez** 3:56  
Got it.  
Mm-hmm.

 **JR M.**4:06  
Devs are making app changes or like adding new features. Maybe new parameters get introduced, those would go out as a classic EV2 deployment and one of the main reasons was because like we would, we'd had to do some like weird thing with stamps. So for example we.

 **Angel Perez** 4:10  
Mm-hmm.  
Got it.  
Yeah.

 **JR M.**4:25  
Our clusters were named like EG so for event greater than dash, environment name DASH 251, they always start with 251 for some reason and then the next cluster I would be added would be 252253254 sometimes that.

 **Angel Perez** 4:30  
Mm-hmm.

 **JR M.**4:43  
That like that naming convention would like change with different types of clusters like another one would be 501502503 and so like we'd had to come up with like a rollout configuration where we would map like how many active clusters we have. Like let's say there's five and then.

 **Angel Perez** 4:52  
Yep.

 **JR M.**5:02  
We would use our stamps that way it would be we would target stamp one which would be 251 essentially our first active cluster, right? So so that that makes it a little bit weird with like updating 'cause. Then if someone wanted to specifically update like 253 that would technically be our third stamp.

 **Saebom Lee** 5:10  
Mm-hmm.

 **JR M.**5:21  
Right 251252253 so they would have to know that the third active cluster is 253.

 **Angel Perez** 5:22  
Mm-hmm.  
Got it. So this is all more of AI guess broken off of the naming convention that your team decided on.

 **JR M.**5:38  
Mm-hmm. Mm-hmm. Yeah. So. Oh, that's why that's kind of why.

 **Angel Perez** 5:41  
Is there like a plan for migrating in the future or?

 **JR M.**5:45  
So like a lot of those configs I kind of done already, but just getting I mean getting everybody on the team to like change is one thing and then I kind of got pulled off the service a little bit. So I'm I'm on a different service now.

 **Angel Perez** 5:58  
Mm-hmm.  
I see.

 **JR M.**6:03  
Yeah. Mm-hmm.

 **Angel Perez** 6:05  
Got it. Cool. We'll we'll definitely have more things towards the end of this call that we can talk about that can, I guess, motivate folks to plan the migration in the future. There's a lot of support that we're building into EV2 of.  
Already highly adopted third party features that we're having in for our external customers, but yeah, good. Good to know the kind of what the breakdown was and and that naming.

 **Saebom Lee** 6:35  
Hmm.

 **Angel Perez** 6:39  
Design decision that kind of made that split.

 **JR M.**6:43  
Mm-hmm. When you say external customers, you mean like the public cause? Like, EV2 is more of an internal internal tool, is it not?

 **Angel Perez** 6:47  
Yeah, yeah, yeah.  
Yeah, I mean, external customers as in ARM. So to put it more plainly, there is a lot of ARM concepts and features that are delivered as part of what the public cloud offers to our external customers.

 **JR M.**6:54  
Oh, OK.  
Mm-hmm.  
Hmm.

 **Angel Perez** 7:08  
But within the EB2 mechanism that is used internally, there's still a little bit of a disconnect and we're doing the work and and part of this research study is for one of those features that we're integrating to bring all that goodness into the native platform that internal folks have to use.

 **JR M.**7:15  
Yeah.  
Yep, the what if?

 **Angel Perez** 7:27  
Use but yeah.

 **JR M.**7:28  
OK. Yeah, that makes sense.

 **Angel Perez** 7:31  
But feel free to continue save on.

 **Saebom Lee** 7:31  
Yeah.  
OK. Yeah. Just to pick it up from there, so like I'm template versus spice up like which one do you currently use and any anything you wanted to share with us like regarding this?

 **JR M.**7:47  
Arm templates 100% we've never we don't, we don't really do any bicep deployments.

 **Saebom Lee** 7:49  
And.  
Any how do I say it? I'm not leading you to favour something, but like any reason why you're not currently adopting bicep?

 **JR M.**8:07  
It was just the right the repo had already been stood up years ago, years before I joined the team, so I kinda just adapted. But yeah, I not too many of us know much about bicep other than like every once in a while I'd see a different.

 **Saebom Lee** 8:12  
OK.  
Mm-hmm.

 **JR M.**8:22  
Some some docs on it I I think bicep does introduce more functionality if I remember correctly.

 **Angel Perez** 8:29  
Yeah. So, so bicep is essentially a more modern.  
Language that provides infrastructure as code in Azure A compiles to arm templates, so there's a compiler that converts it to an ARM template and the deployment gets submitted, but the bicep language itself has.

 **JR M.**8:41  
Mm-hmm.

 **Angel Perez** 8:51  
It it is more like a programming language rather than regular Jason, which is what ARM templates is in ARM templates. We embedded a bunch of functions into the Jason itself and in bicep what we did is we.

 **JR M.**9:04  
Mm-hmm.

 **Angel Perez** 9:07  
We basically changed a lot of things to be able to make it feel more like a programming language. Adding four loops, user defined functions, and a bunch of other stuff. We can dive more into bicep and in in another call for sure.

 **JR M.**9:20  
Yep, Yep.

 **Angel Perez** 9:25  
But as it relates to this topic here of of EB2 so bicep, you can use today no problem for your any any deployments that you're doing more so one off but within the within the service model dot Jason that you deploy in EB2.  
There is no native bicep support yet they will be available very soon. But today where you have those service resource group definitions, those are all ARM template DOT Jason today.

 **JR M.**9:46  
Oh OK. Gotcha.

 **Saebom Lee** 9:49  
Hmm.

 **JR M.**9:54  
Mm-hmm.  
Yep, that makes sense.

 **Angel Perez** 9:59  
In the future, you'll be able to just reference a bicep file, but yeah, that's all the context on on that bullet for now.

 **JR M.**10:07  
OK.

 **Saebom Lee** 10:10  
OK. And then what is your current cicd tooling way? Is it Azure DevOps or get up actions?

 **JR M.**10:18  
Azure DevOps.

 **Saebom Lee** 10:19  
OK. Do you perform a delete by?  
Azure DevOps like delete stuff.

 **JR M.**10:26  
Yeah.  
No, just yeah, just build deployments and then releases as well.

 **Saebom Lee** 10:36  
OK.

 **Angel Perez** 10:40  
Cool.

 **Saebom Lee** 10:41  
All right. So it sounds like you guys are working on like on template for now. So like you're so like what testing like do you, how do you do like testing or validation before deployment?

 **JR M.**11:00  
That I'm a little bit disconnected on the current methods that were used in the past depending on like how how much change there was in APR.

 **Saebom Lee** 11:04  
Mm-hmm.

 **JR M.**11:15  
Upper manager might suggest doing an arm. What if so? Like that's where I became familiar with arms. What if feature? And then we've also done like.

 **Saebom Lee** 11:23  
OK.

 **JR M.**11:32  
Some of the EV2 test I think there's like a test deployment, one where essentially tests are artefacts. If I remember correctly, but that's that's about it really. Like we don't. We should have more testing involved as as far as like EV2 and whether where whether or not things are like formatted.

 **Saebom Lee** 11:41  
Mm-hmm.

 **JR M.**11:50  
Correctly and all that stuff, but that's probably one of our biggest things, is it? It breaks often.

 **Saebom Lee** 11:52  
Yeah.  
All right, all right.

 **JR M.**11:56  
Especially not being completely rigid agnostic, there's a.

 **Saebom Lee** 12:01  
Mm-hmm.

 **JR M.**12:03  
Kind of like a need for us at the moment because we aren't fully hundred 100% re diagnostic where we could like persist those into our into our repo. We're generating things so someone breaks things in the generation script it it just stops generating artefacts right so.

 **Saebom Lee** 12:19  
OK. OK. Would you say, would you say that you, your team is like dependent on testing or do you the how how much do you engage with testing, do you do a lot of testing?  
'Cause like it varies by chain back.

 **JR M.**12:35  
What with the with deployments not so much. I mean the testing that's in there right now is like R2DR2D testing, which is like testing through Canary and stuff like that and PPE, but it's about it.  
I mean, there's there's also app testing and stuff like that end to end test, stress test proof tests that they also do through PRS and stuff, if that's what you're asking.

 **Saebom Lee** 12:49  
OK.  
OK, OK.

 **Angel Perez** 13:00  
Yeah, I think for this is more so just specific to the template itself. So in in our we have the preflight API which is something we ask all RPS to implement and if your service hasn't implemented it then we can.

 **Saebom Lee** 13:05  
Mm-hmm.

 **JR M.**13:05  
Oh, OK. Yeah. So then no.

 **Angel Perez** 13:18  
Talk about that later, but for this discussion, basically what we have is two things. So when a template gets deployed, that API gets called and it fans out the subset of resources that belong to each different resource provider namespace.  
They get to do some pre emptive validation check on these more robust than what the ARM team can do so each RP is the subject matter expert on their own thing and then return to arm a hey this you know these resources look good. Proceed.

 **JR M.**13:49  
Mm-hmm.

 **Angel Perez** 13:55  
Or no, you should fail this validation. That way we can give customers a more immediate response. It usually prefly runs in under a minute, so customers will quickly know like you know if you put the name potato for this property value, this is not going to work. So don't wait 40 minutes.  
It's or 20 minutes, which is I guess the latest average on those big deployments for the top 100 customers.

 **JR M.**14:20  
Yeah.  
Mm-hmm.

 **Angel Perez** 14:23  
But then what if it takes it a bit farther left, so you'll start hearing this term shift left more and more, where the ARM team is trying to do their best to try and catch errors.  
As they're being written, pretty much so if you use VS code, there's an extension that makes use of a bunch of stuff that we have to give you like real time Intellisense and validation on on certain properties that we can detect. But on others that are.  
Can't be determined until it gets sent to the RP. There is where the preflight API exists. What if is a bit different? What if aims to basically give you a? What's going to happen?  
In comparison to your last deployment.  
And it makes use of preflight. But what? What if does? Is it just gives you a? You know if you made these changes to the template and you run a, what if it's gonna tell you? Hey, here, you're gonna modify two vnets. Delete one storage account and create 5 new things or what have you.

 **JR M.**15:17  
OK.

 **Angel Perez** 15:37  
So that's what if in in a nutshell.

 **JR M.**15:40  
OK. Yep.

 **Angel Perez** 15:43  
Cool. I think we're gonna move on. Say, mom, and and just walk through what we have.  
But yeah, go ahead.

 **Saebom Lee** 15:50  
OK, OK, sounds good. So like for the next 15 minutes, I would like to just spend our time to discuss around these topics. Let's start with the first one. How often do you need to delete resources in your service? Is it weekly or monthly? Oh, I I remember that you mentioned that you don't usually delete.  
It.

 **JR M.**16:09  
Correct. Yeah. We don't delete. This is. Yeah. Yeah. We just don't have a need for it unless we're, like, decommissioning some parts of our infrastructure for some reason.

 **Saebom Lee** 16:11  
OK.  
All right, let's.  
Thank you.  
OK. OK. Do you ever build test environments?

 **JR M.**16:28  
Yes, we have our PPE environment that's just getting stood up I think, but for the most part a lot of people test and like internal subscriptions that are given to us.

 **Saebom Lee** 16:37  
OK.  
Alright. And then after it's no longer in use, you're not deleting them though, right?

 **JR M.**16:48  
Those ones we used to have a subscription library where you could like check out the subscription and after like 10 days. I think it is it would auto delete. I think that's since like been kind of like fallen off. I think there was an SFI thing that kind of broke that whole thing and nobody fixed it.  
So now those subscriptions kind of just get a little wild and out of hand sometimes until like someone decides to clean everything up.

 **Saebom Lee** 17:14  
OK. OK. I think that's pretty relevant to the next topic though, like any resource cleanup that you are interested.

 **JR M.**17:25  
Yeah. Nothing, nothing. Automatic. And this actually reminds me. I need to. I need to delete a few few cosmos databases that are being flagged right now, so.

 **Saebom Lee** 17:36  
OK, OK, any, any, any, anything that you mined up when you were cleaning your resources?

 **JR M.**17:49  
And I'm sorry, what was that?

 **Saebom Lee** 17:50  
Like any anything, you care when you are deleting your resources.

 **JR M.**17:54  
Oh.  
I mean, I would say there are like with this when we had the subscription library, it was nice to like.  
Elect keep certain things like some people in the past used to keep just the database and everything else would get torn down that way, like a few important documents that maybe they imported manually were still there, be it for like testing and then like.

 **Saebom Lee** 18:14  
None.

 **JR M.**18:22  
Maybe weren't incurring any or very little cost and then they could spin up things a lot easier.  
But yeah, we don't. I don't think we really have a need for specific way of deleting.

 **Saebom Lee** 18:37  
OK.  
Any safeguard methods involved?

 **Angel Perez** 18:42  
Sorry real quick. Are you using JIT today to delete whatever gets called out?

 **JR M.**18:44  
Mm-hmm.  
OK, in in int, yeah, in int it's free for all, right? But yeah, in prod, yes, we we definitely used JIT and those. Yeah, those are the infrequent deletions where like we're migrating a cluster or we've added a cluster and we're decommissioning a cluster.

 **Saebom Lee** 18:55  
Mm-hmm.

 **Angel Perez** 19:01  
Me.

 **JR M.**19:05  
Possibly like.

 **Angel Perez** 19:05  
So would you say your team usually deletes like once a month when it's, you know in those unusual cases? And then whatever's in testing every six months when it, you know, it gets out of hand.

 **JR M.**19:12  
Maybe.  
In, in testing I mean in testing. Yeah, in testing I would probably even say that would be once a month and like a bulk cleanup, maybe even once every three months in prod. It's hard to really say maybe like once every three months as well, but.  
Not as not as much. It would be. It would just be like one cluster here or there.

 **Angel Perez** 19:39  
Yeah, yeah, yeah. More individual on the prod side and then full environments on the testing side. But you're saying that it's still like every one to two months on the testing cleanup.

 **JR M.**19:47  
Yeah, and.  
Yeah. And I would say, yeah, and in prod actually there is a specific test, I mean a specific way of deleting resources. So like it would be like disabling our traffic managers that way, no traffic goes on there, maybe verifying logs to make sure that nothing's hitting that cluster.

 **Angel Perez** 19:58  
Mm-hmm.

 **JR M.**20:08  
Umm.  
Draining our lease collections.

 **Angel Perez** 20:15  
And how are those deletes performed?  
Like that, the ones for the the full environment. Is it with Jade as well?

 **JR M.**20:19  
Manual.  
Yeah, for jit. Uh-huh. It would be JIT going to portal. I think they, I think we had just created an ARM template for or not even an ARM template. I think. I think we created an EV2 ext app that handles it now.  
Essentially doing a deployment to like disable our arm, our traffic manager, and then doing like some service Fabric update command to like resetting their input listener, those types of things.

 **Angel Perez** 20:41  
Interesting.  
Got it. Cool.

 **JR M.**20:56  
Mm-hmm.

 **Saebom Lee** 20:56  
Interesting. Is it only your team that's using?  
Thanks.

 **JR M.**21:01  
Umm.  
I'm not too sure how other teams are doing it, cause like I've I've been between two teams. The current one I'm on.  
I'd say their deletion process is more so cleaning up database stuff first. It's it's a lot, so it's a lot more crucial. It's a.

 **Angel Perez** 21:18  
Mm-hmm.  
Sorry, what team is this, by the way?

 **JR M.**21:23  
Yeah. So the one I'm on right now, service bus event hubs, the one I was on and did a lot more EV2 deployments on was event grid.

 **Angel Perez** 21:31  
Got it. OK.

 **JR M.**21:32  
Mm-hmm.

 **Angel Perez** 21:34  
Cool.  
Go ahead, say well.

 **Saebom Lee** 21:36  
All right. OK. OK. Could you just share like any incidents or like any unwanted changes that happened as you made change during the process like you when you were removing the resources in your testing environment like?  
Is there any case that you would like to share with us when you had, like incidents happening?

 **JR M.**22:03  
Unwanted changes I can't think of any off the top of my head.

 **Angel Perez** 22:11  
Like, does your team make use of? I'm assuming RBAC just to keep folks from accessing stuff they don't have to.

 **JR M.**22:11  
Umm.  
Mm-hmm.  
Yeah. Uh-huh.

 **Angel Perez** 22:22  
Aside from that, do you make use of Azure policy by any chance?

 **JR M.**22:28  
I believe there are some Azure policies, but we I have never touched Azure policy all that much.

 **Angel Perez** 22:35  
Has there ever been an instance where a team member that had access that they they needed to work on their project or what have you that made an accidental change?

 **JR M.**22:53  
Umm.  
Yeah, probably more so like misunderstanding a flow of of things, though. Mm-hmm.

 **Angel Perez** 23:01  
Yeah. Yeah. So this I guess where this is coming from is kind of related to the SFI topic. So this past year, I think the company had like 100 plus incidents that were related to.

 **JR M.**23:09  
Mm-hmm.

 **Angel Perez** 23:18  
You know, folks trying to do the right thing and deleting what's called out, but then not noticing that this thing is being used by something else and causing other issues.

 **Saebom Lee** 23:28  
Yeah.

 **JR M.**23:29  
Mm-hmm. That makes sense.

 **Angel Perez** 23:31  
So yeah, this topic gets more around that, but we'll we'll talk more about that. And for access governance, I guess you already mentioned our back is is the main thing and and a couple policies here and there. So I think we're we're good on this slide say Bob.

 **Saebom Lee** 23:46  
OK, sounds good. Let's move on to the fun part. Then they are. We have prepared a signed mock up. It's not the finalised version, but we would like to go through the screens that Azima created for us today with you and then hear how you think.

 **JR M.**24:03  
Mm-hmm.

 **Saebom Lee** 24:05  
Around the way that we are reaching out to see whatever you we we wanted to show you. So what we want to do here is that we want to.  
We want you to see how the what if output here on the CLI screen could be reflected on the EV2 Azure Portal, EV2 Portal and Azure Portal. So here we will. We would like to take this to be the prompt that we would take today.

 **JR M.**24:37  
Mm-hmm.

 **Saebom Lee** 24:37  
Please take a minute to have a look at it and then we can move on to the web.  
Screen whenever you're ready.  
Umm.

 **JR M.**24:50  
OK, I'm ready, Angel.

 **Saebom Lee** 24:52  
OK, you know, yeah.

 **Angel Perez** 24:53  
Sorry real quick. I think Saebom and now I'm noticing in the order of what we have here, there might be some extra context missing and I think that's my bad for not flagging it sooner. So one thing I forgot to ask even at the beginning of all.

 **Saebom Lee** 25:01  
OK.

 **Angel Perez** 25:10  
They are is. Have you ever heard of deployment stacks?

 **JR M.**25:16  
No, I have not.

 **Angel Perez** 25:18  
Gotcha. That's very important for us to be able to interpret all this. So let me give a quick summary of what that is. So I already explained a little bit of what, what if is and as you can see here.

 **JR M.**25:32  
Mm-hmm.

 **Angel Perez** 25:35  
What I mentioned of a DIF on what's the before on after on property values of a template. That's what's getting shown in the kind of bottom half of this mock up here.

 **JR M.**25:43  
Mm-hmm.  
Yeah.

 **Angel Perez** 25:49  
Where you can see a resource group being called out or a different kind of resource type like a virtual network and what kind of properties are changing as a result of whatever changes you made to the ARM template dot Jason.  
So when you run this, what if you're not submitting a deployment to arm, so it's just a preview API, in other words.

 **JR M.**26:15  
OK. I mean it's getting your, it's probably getting your your current state though right kind of like terraform or something?

 **Angel Perez** 26:21  
Yeah. Yeah, it compares the current state of things to what's in the template.

 **JR M.**26:26  
Mm-hmm. Yep, that makes sense.

 **Angel Perez** 26:28  
And what we do is we kind of categorise it by different change types. So we have create, modify, delete and a couple of others, but it gives you an output that breaks it down per resource and all the changes that we can predict are going to happen.

 **JR M.**26:46  
And this.

 **Angel Perez** 26:46  
I think the most important thing as it relates to EV2 for this conversation is predicting deletes. That way you have like an extra safeguard step that says like this is the list of resources that are going to be deleted and then you.  
You know what? Whether it's through scripting or more of as a meeting amongst your team, you can decide like, oh, OK, we can proceed or no, that actually shouldn't be deleted. We should double check that template update.

 **JR M.**27:15  
Yeah, and this, I mean, I'd imagine this is probably gonna honour like the scope bindings and stuff, right? So you'll do the find, replace with the scope bindings and kinda show us what the naming convention I guess would be for some of those things.

 **Angel Perez** 27:30  
That I have to double check with our EB2 engineers. I'd assume so. What if right now that's being built is very specific to what's in the template, so the ARM template dot Jason it's not a. What if for the service model dot Jason?

 **JR M.**27:37  
OK.  
OK.

 **Angel Perez** 27:51  
So this is still within the scope of what gets deployed to Azure Resource manager's front door. So EV 2:00, what if for the other components of EV2?

 **JR M.**27:57  
Mm-hmm.

 **Angel Perez** 28:04  
That would also be awesome, so I'm definitely gonna quote your ask for this in in some future, but for now, the the first piece of this that we're kind of integrating into the EV2 portal and experience is.  
The ARM deployments, what if or in this case more specifically the deployment stacks? What if so that brings me to the other topic of stacks deployment stacks is essentially the next version of.

 **JR M.**28:25  
Mm-hmm.  
Mm-hmm.

 **Saebom Lee** 28:28  
Yeah.

 **Angel Perez** 28:38  
Arm template deployments, so ARM template deployments as you know them today it is more like a record you submit a deployment and within the scope of your Azure portal you can see that deployment section and settings and it has like every deployment you've made and the name of that deployment.  
You're with me so far on that context.

 **JR M.**29:00  
Yep.

 **Angel Perez** 29:01  
OK, so now deployment stacks is a new resource type within Microsoft DOT resources as well. But this new resource type brings two capabilities into the fold. One of them is.  
We call it action on unmanaged. The deployment stack essentially is a new resource that tracks the state of the ARM template and all of the resources that get deployed are considered to be managed resources.  
So that stack is.

 **JR M.**29:34  
So everything within that ARM template.

 **Angel Perez** 29:37  
I'm sorry, what?

 **JR M.**29:37  
Is a stack. Everything within an ARM template is a stack, is that am I understanding correctly?

 **Angel Perez** 29:42  
No, On the contrary, the STACK deploys an ARM template. The stack is a separate resource that gets created and maybe I can share my screen real quick. Say well if that's OK with you to just kind of give that context.

 **JR M.**29:46  
Oh, OK.  
OK.

 **Saebom Lee** 29:51  
Yeah.

 **Angel Perez** 29:56  
You just open up the portal real quick.  
But yeah, so basically stack is a new resource type that brings two capabilities into the fold. One of them is for allowing you to scale this cleanup operation that we're kind of alluding to at the beginning of this call.  
And the second is.

 **JR M.**30:19  
Are you are you sharing? Oh, OK, sorry. OK, gotcha.

 **Angel Perez** 30:21  
No, not yet. So I'm getting everything set up, but still sharing a little bit of the context. No, you're good.

 **JR M.**30:26  
Uh-huh.

 **Angel Perez** 30:29  
Uh.  
Here we go.  
Can all of you see my screen?

 **JR M.**30:36  
Yep.

 **Saebom Lee** 30:36  
Yes.

 **Angel Perez** 30:39  
Cool. So I'll just dive right into a particular scope. Stacks can be created at any scope, just like deployments, resource group scope, subscription, scope management, group SCOPE. For this example, I'll just take a look at the subscription scope.  
And we can see in settings before you had deployments, right and these are just those records. So now we'll have also deployment stacks but different than deployments. Deployment stacks are full on.

 **JR M.**31:00  
Mm-hmm.  
Yep.

 **Angel Perez** 31:13  
Resource types. So what do I mean by that? They have a resource ID, so if I go to the properties here.  
This resource has its own ID and you can use RBAC to I guess limit folks that to see this stack resource.

 **JR M.**31:31  
Mm-hmm.

 **Angel Perez** 31:33  
OK. So that's one one thing in terms of capabilities tied to this new resource type, there are two action on unmanage. This relates to the cleanup. So what this says is all of the resources.  
That the stack manages.  
So if I see this deck right now and manage 6 resources, it has two RGS and a couple of V Nets.  
All of those resources, if I were to.  
Unmanage any of these by removing them from the ARM template or maybe even deleting the stack. The stack has permission to perform the action or unmanaged setting that you set. So in this case they're all set to delete.  
So what this means is, if I were to go to the ARM template that defines those six resources and just comment out a couple of those V Nets 'cause they showed up in some SFI item, or maybe fifty of them or what have you. And I update the I redeploy the stack.

 **JR M.**32:22  
OK.

 **Angel Perez** 32:41  
The stack is gonna take care of finding out that delete operation to all the scopes that those resources are in. So before we're kind of doing this kind of 1 to one going RG per RG and finding what needs to be deleted or.

 **JR M.**32:56  
Mm-hmm.

 **Angel Perez** 32:57  
I don't know how that bites that EV2 ext was authored to to kind of handle that traffic manager cleanup, but now with this native capability you have an easy way to juice do A1 to many delete.  
So for clean up like if you have a stack that we call, you know event hubs, test environment 6 or what have you and we're all done with that. We can just go ahead and and say all right.  
The work team and we just delete the stack and here we get a couple of options. We can say I want to delete everything and we're all. We're all done or I wanna just delete resources but let the resource group stay there. Maybe you still need that.  
Infrastructure for, you know, whatever other work folks need to do, or maybe even you just want to leave everything there, and that's what detached does. It just breaks that association from the manage resource to the stack resource.

 **JR M.**33:56  
Hmm.  
For the stack.

 **Angel Perez** 34:01  
So that's one thing in the topic of preventing unwanted changes. Here we have what we call deny settings. Are you at all familiar with deny assignments?

 **JR M.**34:15  
Yeah, and like role assignments, you're saying?

 **Angel Perez** 34:16  
S.  
No deny assignments more specifically.

 **JR M.**34:23  
Just in like possibly like network security groups and stuff, but.

 **Angel Perez** 34:28  
No. Yeah. So deny assignments more broadly in Azure, I think. I'm not sure that's something I have to actually work on, like figuring out what other ways are still possible to create out there. But policy is one of them.

 **JR M.**34:29  
In this sense.

 **Angel Perez** 34:43  
But basically, what deny assignments does in this context of stacks is it creates kind of a lock on all of those managed resources that blocks other users that have access to them from performing.  
Well, there's two settings. You can either say, deny write and delete, or just deny delete, and in that case you know folks that do have access. Let's say for example myself. This is said to deny, write and delete.

 **JR M.**35:06  
Aha.

 **Angel Perez** 35:15  
And now we're basically gonna go and try and delete this V net.  
And I will end up getting blocked and we'll say like, hey, you, you actually don't have access to. Oh, wait, I didn't get blocked.  
What?  
Well, it's still there. This might be a broader issue with oh, it's not there anymore.  
We've been having so many issues with Venus, so maybe that's not the best example, but essentially what this does, it blocks users from from performing that action. Let me try with the resource group because I'm actually curious now.  
And throughout all this, Jr, if you have any questions, feel free to just let me know.

 **JR M.**36:32  
Mm-hmm.

 **Angel Perez** 36:41  
Umm.  
Yeah, so stacks is already a Georgia service. This is available to our third party customers and that's what.  
Kind of. I was alluding to at the beginning of we're trying to bring a lot of the goodness that we're provided already externally to our internal platform.

 **JR M.**37:02  
OK.

 **Angel Perez** 37:11  
I will circle back to this, but in a nutshell, those are the two capabilities that stacks brings into the table and the way it was built is essentially you don't have to make any changes to your templates, you just pass in your templates to a stack create command. So rather than doing like an A.  
The group deployment.

 **JR M.**37:33  
Mm-hmm.

 **Angel Perez** 37:34  
Create you would do an AZ stack create and that's it and and just pass in the configurations for the those different properties that we talked about. So now with all of that context, we can jump back, say bomb and essentially what we were showing is.

 **JR M.**37:37  
OK.

 **Saebom Lee** 37:51  
And.

 **Angel Perez** 37:53  
What if for stacks so so today we have. What if for regular deployments?  
But regular deployment, what if it's not being integrated into EV2? We're doing the work for this future version of deployments called deployment stacks that we think will be a huge benefit to to all of Azure services as it relates to.  
Just having safer guardrails as well as being able to clean up stuff faster and more safely.  
Cool.

 **Saebom Lee** 38:28  
Please feel free to chime in Hong Kao and Azima in case I don't ask. I missed some of the questions we have prepared, so like what we're trying to learn today is we want to see like as you get through the CLI screen and you.  
You have your goal with the what if you end deployment stacks? You wanna see how you can actually reach out or navigate those information on EV2 portal. So here it might be familiar to you. Where do you expect to see the what if output on this EV2 screen?

 **JR M.**38:52  
Mm-hmm.  
Mm-hmm.  
Maybe the all details.  
Are you are you asking me? You're like, are you 'cause? Like it says? What if results like? I thought it would just be.  
Pretty easy to see, maybe graphical.

 **Saebom Lee** 39:26  
OK, no worries at all. This is the, this is the testing. So like we had to build some screens that we wanted to show you first, but we it does not mean that we are stuck to that way.  
Let's see if I can pull something out. So what we're thinking here is that we had this, what if results tap and then something. Oh, oh, OK, OK.

 **JR M.**39:50  
Oh sorry, I thought that was already selected. OK, OK, gotcha.

 **Saebom Lee** 39:56  
All right. Does that, does that change your understanding then then how's your understanding about this column? So this is very basic for you now. So like, please feel free to use your imagination or like anything you would like to have instead of this.

 **JR M.**39:59  
Yes, yeah, yeah.  
Yeah, that makes sense.  
Create group delete. Yeah, I mean this makes this makes sense. If you click on deploy though I'm imagining like it would show you what three resources it's creating.

 **Angel Perez** 40:33  
Sorry Jr, have you used the EV2 portal before?

 **JR M.**40:37  
Mm-hmm.

 **Angel Perez** 40:39  
OK. Yeah. So this is the same view you're used to today.

 **JR M.**40:43  
Yeah, but like the what if I mean the what if results aren't in there, right? So like how do I view what three resources it's creating? Am I clicking deploy and then that like side panel comes out?

 **Saebom Lee** 40:47  
Hmm.

 **Angel Perez** 40:47  
Mm-hmm.  
Yep.

 **Saebom Lee** 40:55  
Actually yes.

 **JR M.**40:58  
OK.  
OK. That makes sense then.

 **Saebom Lee** 41:04  
OK, though, what do you think about this screen? Is it easy for you to read or do you have any thoughts?

 **JR M.**41:15  
Yeah, I think it. I think it makes sense. I mean it makes sense that like those two resources are being created, they're resource one resource two, if you click the drop down for delete three, it'll show those. And so on. Same thing with the modifications.

 **Angel Perez** 41:29  
So in the EV2 portal, what you're gonna get is a combination of change type and resource name. But for the full context of the property diff that we saw in the CLI view at the beginning.

 **JR M.**41:44  
Mm-hmm.

 **Angel Perez** 41:44  
That's going to live within the Azure portal, so here we provide a link to the water result resource type that jumps to the portal and and shows us.

 **JR M.**41:58  
Hmm.  
OK. That way we get more context on things, huh?

 **Saebom Lee** 42:01  
So that's.  
Mm-hmm.

 **Angel Perez** 42:04  
Yeah. So and and the reason we did this is cause in. So in EV2, there is no concept of RBAC.

 **JR M.**42:14  
Mm-hmm.

 **Angel Perez** 42:15  
And although what if does remove everything that is defined as secure string in in the manifest from? From what gets shown that it?

 **JR M.**42:30  
Yeah.

 **Angel Perez** 42:31  
It masks it. But you know there might be a case where some team did not author their manifest properly and didn't define that we don't want to expose secrets accidentally. So this is a way we found that we're able to kind of.  
Achieve both things like give the the biggest call out in the EV2 portal of what's going to happen, and then a way to jump to the other portal that does have our back end can assure that no one that is supposed to be seeing that.

 **JR M.**42:54  
Mm-hmm.

 **Angel Perez** 43:03  
Got to see it.

 **JR M.**43:04  
Yeah, that Uri. Do you just like click it and it goes to it or do you have to copy it and then put it, OK?

 **Angel Perez** 43:08  
Yeah.  
No. We can probably highlight that in blue, in the mockups, asima in the future.

 **Saebom Lee** 43:15  
You can just click and go would would you say Jr the way to move on to the next portal like by clicking here? Would you say it's easy to catch on your first glance or?

 **JR M.**43:17  
Mm-hmm.  
If yeah, I I mean if it looks like a link then yeah. But like right there, yeah, it just looks like like text. Right. But.

 **Saebom Lee** 43:34  
Then, knowing that you can move on to the next portal, did you find any like missing information other than these here or like is there any information that you would like to have?  
On top of this.

 **JR M.**43:54  
Not necessarily.

 **Saebom Lee** 43:58  
OK.

 **JR M.**43:59  
Like those those creates right there it says resource one that'll be. Will that be like Microsoft DOT event hubs or will it be like the name of the event hub?

 **Saebom Lee** 44:03  
Mm-hmm.  
Which way would you like to have? Which way do you prefer?

 **JR M.**44:15  
Possibly the.  
I mean, they're both very useful, maybe like having the the provider and then the actual name of the resource might be better.

 **Saebom Lee** 44:28  
OK, OK. Sounds great here on the on this little list here, what is your thought regarding this? Do you have any suggestions or do you have any question?

 **JR M.**44:43  
Not necessarily. All that looks very familiar, so that that still looks the same to me.

 **Saebom Lee** 44:48  
OK, OK. Sounds good. There's one thing that I wanted to double check with you in the previous screen, so here in the column here it's the same information with the the panel. Do you have any suggestions or do you think it's meeting your expectation? Because the reason why I'm asking?

 **JR M.**45:00  
Mm-hmm.

 **Saebom Lee** 45:08  
Is that's like when you first and came, like landed on to EV2 portal design. Here you were mentioning like something about like visual information is this I don't know.

 **JR M.**45:21  
Yeah. So like the graphical thing.

 **Saebom Lee** 45:24  
OK, the graphical thing in, if you choose the the graphical view like this one.

 **JR M.**45:28  
Yeah. So I don't know if it if it would provide a different like graphical thing of like I don't know 3 green blocks with like resources being added. You know how like it has like a graph essentially.

 **Saebom Lee** 45:39  
OK, by meaning graphical view like is there any specific graphic that you are looking for in terms of what if results like do you want to see like where the changes happened or?

 **JR M.**45:40  
Umm.  
Uh-huh.  
So like if I think Angel had mentioned before of like dependencies.

 **Saebom Lee** 46:00  
Play song.

 **JR M.**46:02  
And like deleting things, but like in the graphical thing. If like if you're doing a, what if deployment?

 **Saebom Lee** 46:09  
Mm-hmm.

 **JR M.**46:10  
And you happen to be deleting something that was previously dependent by some by another resource, like seeing that that break between those that association might be important. So like, let's say I typically deploy.

 **Saebom Lee** 46:18  
OK.

 **JR M.**46:28  
My traffic manager and then my next step of like the sequence of deployments is like my traffic manager has to come up first and then my cluster needs to have that traffic manager as a prerequisite step, right?

 **Saebom Lee** 46:35  
Mm-hmm.  
Yeah.  
Mm-hmm.

 **JR M.**46:43  
So those two exist and they would have a link to it in that graphical user interface where I'd have like all the left things, and then that would deploy first and then on the next like right side of it would have all the things that come after right that are dependent.  
If you're somehow deleting something on the left, then that like is no longer that. Would that kind of breaks that relationship of like my next step essentially right? So maybe having that graphical.

 **Saebom Lee** 47:08  
Mm-hmm.  
Mm-hmm.  
OK, got it. Got it. That kind of makes me think like we already have the graphical view here. So like what if we just add some like a what if results were like doing stacks information on top of the existing graphical view?

 **JR M.**47:14  
Picture would be nice.

 **Saebom Lee** 47:28  
Would it be helpful for you, or would you then let's take it this way, would if you had, let's assume you had that on the graphical view, you can find the what if related results there, would you prefer to go directly to graphical view?

 **JR M.**47:30  
Mm-hmm.

 **Saebom Lee** 47:46  
Or do you still find the need to have this? What if results tap?

 **JR M.**47:50  
No, I think the what if results tab is a lot nicer. I'm just saying like it would be nice. So like if if I'm deleting something so I know if like networking resources there's times where you're trying to delete a public IP, but it's being in use by.

 **Saebom Lee** 47:53  
OK.  
Aye.  
OK.

 **JR M.**48:06  
Like a vnet or something, right? And so like whenever you try deleting it from the portal, it doesn't allow you because it's like this or maybe the subnet is in use, right? And then you have to, like, disassociate it. So like if you try deleting that resource it it wouldn't let you. So I don't know if this could display it a little differently maybe.

 **Saebom Lee** 48:15  
None.

 **JR M.**48:25  
Yeah.

 **Saebom Lee** 48:25  
OK, OK. And I want you to focus on the information here in detail with columns here. Do you what is your thoughts like? Do you like the way it's presented as it is or do you have any suggestions?

 **JR M.**48:30  
Mm-hmm.

 **Saebom Lee** 48:40  
Or if you want to add more information on top of like create, delete, group, delete and modify, please feel free to share with us.

 **JR M.**48:47  
No, that's that's good. I also need to hop off in like 3 minutes.

 **Saebom Lee** 48:53  
OK, I'm sorry. All right, all right. I think those are pretty much my questions. Aziman, Hal, do we have further question to Jr?

 **JR M.**48:54  
But yeah, that that looks fine to me.

 **Angel Perez** 49:04  
I think the the last thing and thanks again Jr for your time. If we can go back to the CLI view.

 **JR M.**49:08  
Mm-hmm.

 **Saebom Lee** 49:08  
Mm-hmm.

 **Angel Perez** 49:13  
Say Bob.

 **Saebom Lee** 49:14  
Yep.

 **Angel Perez** 49:17  
So I guess this is since you haven't seen the regular. What if before? This is still just as new to you, but the.

 **JR M.**49:25  
Oh, I've seen the arm. The arm. What if if that's what you're asking?

 **Angel Perez** 49:28  
Yeah, you have seen anything. Oh. OK. Then I guess this is the stacks. What if? And I don't know if I had a glance. You notice anything specifically different from the regular arm? What if?

 **JR M.**49:29  
Yeah. Yeah. Mm-hmm.  
It's been like probably a year and a half, two years since I've used it.

 **Angel Perez** 49:43  
OK, yeah, no worries. So what I'm trying to get at here is.

 **Speaker 1** 49:45  
Yeah.

 **Angel Perez** 49:49  
So we have this stacks resource type itself before in the regular deployments. What if deployment is just a record, so you just get the property DIF of the resources themselves and that's it. But now the stack itself is also a resource.

 **JR M.**49:56  
Uh-huh.  
Mm-hmm.

 **Angel Perez** 50:07  
We gotta give you a property diff of that too.

 **JR M.**50:07  
Mm-hmm. Yep.  
Yep.

 **Angel Perez** 50:11  
So at the very, very top, we're kind of reserving that space for calling out the stack specific property changes.

 **JR M.**50:21  
Oh, OK. Mm-hmm.

 **Angel Perez** 50:23  
And then if you notice in this resource group that's getting deleted.  
So property changes are get called regularly as as you see in very far below in. In the modifies to the VNETS, but for that one that's getting deleted. You can see there's like an at symbol with a stack dot.

 **JR M.**50:32  
Mm-hmm.  
Mm-hmm.

 **Angel Perez** 50:49  
Some property.

 **JR M.**50:50  
Mm-hmm.

 **Angel Perez** 50:52  
We're trying to figure out if you know, is that notation good in terms of US kind of embedding within the resource, what's happening with that resources relationship to the stack rather than just calling it out as as we would a property of that specific?  
Resource because technically managed status and denies that this are not properties of that resource. They're properties. They're things that the stack tracks, yeah.

 **JR M.**51:21  
Of the stack, right? Yeah, I think it would. I think it would make more sense at the stack level like up up above, right.

 **Angel Perez** 51:31  
The status of what's happening to the resource itself.  
So 'cause up above for example what we have is. So take a look at deny settings mode, that one it's going from. It didn't have a deny setting to. Now we're using deny write and delete.

 **JR M.**51:36  
Umm.  
Deny settings.  
Uh-huh.

 **Angel Perez** 51:51  
And below basically what it's saying for that resource group is what is the latest deny status of this resource group?  
And as of that, what if that you just ran the status is set to that?  
So you can see see it in one of two ways like.

 **JR M.**52:13  
Umm.

 **Angel Perez** 52:16  
If it's at the very top and only at the top, then folks need to know you know this applies to all the resources here.

 **JR M.**52:25  
Oh, I see. Mm-hmm.

 **Angel Perez** 52:27  
Whereas if we do per resource, I think with deny status we can probably get away with it. But with managed status is a bit trickier cause some resources are gonna become unmanaged. So actually in this example we we need to fix that too. That managed status should be not managed.

 **JR M.**52:40  
Mm-hmm.

 **Angel Perez** 52:48  
'Cause we're deleting it. So that's what's gonna be the results. Yep, Yep. Yep. I was also looking at this. Like, wait, why is this not clicking?

 **JR M.**52:50  
Yeah, that makes sense.

 **Saebom Lee** 52:50  
Thank you.

 **JR M.**52:56  
Yeah, 'cause, I guess the default is managed, right?

 **Angel Perez** 52:59  
As you can see, Jr, you're our first call and you're helping us figure out all the things that we have to fix for, for telling a better story here. But but we really appreciate, you know, helping us walk through this.

 **Saebom Lee** 53:11  
Yeah.

 **JR M.**53:14  
Yep.

 **Saebom Lee** 53:14  
Thank you so much Jr. If you have any further question, please feel free to let me know. Thank you so much.

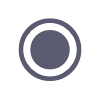
 **JR M.**53:22  
Yep. Thank you. Yes.

 **Speaker 1** 53:23  
Thank you so much GR.

 **JR M.**53:24  
Uh-huh. Bye, bye.

 **Speaker 1** 53:26  
Bye.

 **Angel Perez** 53:26  
See you.

 **Saebom Lee** stopped transcription