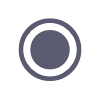
**Imran EV2 WhatIf Research-20251007\_171339-Meeting Recording**

October 7, 2025, 9:13PM

40m 11s

 **Saebom Lee** started transcription

 **Saebom Lee** 0:03  
Recording is on.  
And.  
Yeah, it's good.

 **I.S.** 0:10  
OK. Are we good?

 **Saebom Lee** 0:11  
Yeah.

 **I.S.** 0:12  
OK, so we use, so we were not even on EV2 personally. So this year we made a huge transition to make sure all of our pipelines and we have like 100 plus pipelines actually to move from classic to EV2 regional agnostic. So we go use.  
The latest version of EV2 and things are fine. I think we are happy with all the moves. So last one year we did a bunch of changes. We moved to we use EV2, now we use secured safe deployment practices.  
And we have all the other ABH and safely integrations too.

 **Saebom Lee** 0:52  
OK, OK, I forgot your team. Like what service is your team working?

 **I.S.** 0:58  
So I represent two teams, one is learn organization and the other is localization. So they are called ecosystems.

 **Saebom Lee** 1:06  
OK.  
OK, sounds good.

 **I.S.** 1:10  
ecosystems part of Azure experiences, ecosystems organization under any.

 **Saebom Lee** 1:16  
OK. So is it 100% original mistake right now?

 **I.S.** 1:20  
Yeah.

 **Saebom Lee** 1:21  
OK, OK. Did you have any challenges? Like what? OK.

 **I.S.** 1:24  
We do have classic ones too. Actually we do have classic ones too, but I think it's all because like we have a lot of pipelines we do have we. The thing which we did in our side is we create a lot of shared templates.  
So we have a lot of shared templates, so our EV to.

 **Saebom Lee** 1:41  
Mhm.

 **I.S.** 1:47  
Our EV2 usage has been pretty good personally because we have a lot of shared modules and stuff written which we use.

 **Saebom Lee** 1:56  
OK, so for those few EV2 classic ones, um, you guys are keeping it classic because it's uh, what are the reasons?

 **I.S.** 2:06  
No, I think it's just like there is no reason to just be on classic for our things. I think the major reason is like the prioritization. So I think we have moved most only the ones where we thought maybe either we're going to just delete them eventually or we don't use them. I think they we didn't do that exercise of moving.

 **Saebom Lee** 2:11  
Mhm.

 **I.S.** 2:25  
From classic to the original agnostic.

 **Saebom Lee** 2:28  
OK, that makes sense. Um, then are you currently using ARM template or have you heard of bicep?

 **I.S.** 2:35  
Yeah, we do. We use myself.

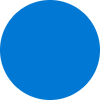
 **Saebom Lee** 2:37  
Oh, like bicep only or?

 **I.S.** 2:38  
We use my.  
I need to check which pipelines use arm templates, which one use bicep, but we use mix of both. So let me check. Just give me one second.

 **Saebom Lee** 2:50  
Yeah.  
I think Angel might have like follow up question regarding template and bicep like in detail.

 **I.S.** 2:58  
Hmm.

 **Saebom Lee** 3:04  
Do you have any differences in the use cases?

3:05  
No.

 **Angel Perez** 3:09  
I think right now the biggest thing distinction is just whether the the files defined in your service resource definitions are ARM templates or or bicep and I believe.  
It should be ARM templates because Bicep isn't supported natively today. That's actually work that we are doing, but it will soon be available that. So that's more of an aside. The distinction in region agnostic and classic, it's mostly because everything we're gonna talk about today is.

 **Saebom Lee** 3:34  
Good.

 **I.S.** 3:38  
OK.  
Mhm.

 **Angel Perez** 3:45  
Investments that are being done in integrating into a region agnostic model. But yeah, that's pretty much it. And this last bullet here is on whether you're familiar with Azure deployments. What if?

 **I.S.** 3:49  
That makes sense.

 **Angel Perez** 4:01  
And use it in any any capacity for your pipelines.

 **I.S.** 4:04  
Yeah, so we use Azure DevOps for sure for the CSC tooling and I haven't heard about what if, but I think maybe that would be a good one to actually go deeper today.

 **Angel Perez** 4:14  
OK, cool. All right, then I can cover all these as we go through this. This call is more specifically to touch on EV2 stacks, what if and what that's gonna look like in the Azure portal.

 **I.S.** 4:18  
Hm.  
OK.  
OK, mm-hmm.

 **Angel Perez** 4:29  
But I'm first gonna cover stacks, because I'm assuming you haven't heard about stacks before yet.

 **I.S.** 4:37  
Yeah.

 **Angel Perez** 4:41  
OK, then let's quickly do that. I can give you a quick demo.

 **Saebom Lee** 4:44  
OK.

 **Angel Perez** 4:48  
Uh, can you see my screen?

 **I.S.** 4:54  
I can see the screen, so we're just trying to pull up the information about ARM versus Bicep infrastructure. So we do use both and I'll just share like I think we have planned to move to 1.

 **Angel Perez** 5:01  
Mhm.  
Cool.

 **I.S.** 5:09  
I we recently did this exercise, so we might be more doing this soon. I'll just share what where we are, where we stand.

 **Angel Perez** 5:17  
That's awesome. Um.  
I think it it we see a.

 **I.S.** 5:21  
I think it.

 **Angel Perez** 5:25  
The statistic is 11.3% failure rate for ARM templates and about 2.6% for bicep. So there's just a lot more that you end up catching here in the VS code experience from as much as we're able to push into.

 **I.S.** 5:32  
Yeah, that's one of the reasons.  
Hmm.

 **Angel Perez** 5:42  
To the validation to section the language has, but yeah, having that in EV2 natively will be a plus as well. So First off deployment stacks. So you're familiar with Bicep and ARM templates, so I'm assuming you're familiar with the deployment.

 **I.S.** 5:47  
Got it.  
Makes sense.  
Mm-hmm. Yeah. Yeah.

 **Angel Perez** 6:02  
Object that exists today. So the deployment stacks is meant to be seen kind of like the evolution of that deployment object. So today in Azure and let me just open a portal window quickly.

 **I.S.** 6:11  
OK.

 **Angel Perez** 6:20  
We just share the full screen first.  
Today in the portal, like we said, those deployments are records. Basically just track what template got submitted and the name of that template and that's it. The deployment stack is.  
Meant to be a resource itself, so it's a resource type that does a template deployment on your behalf. So you end up getting a couple more things included as part of this. Sorry, let me know when you can see my screen.

 **I.S.** 6:49  
OK.  
Mhm.

 **Saebom Lee** 7:01  
That's good.

 **I.S.** 7:02  
I can see it now.

 **Angel Perez** 7:03  
All right, sweet. Let's just go to a quick subscription scope so we can see this example. So at subscription scope or resource group scope or whichever we have those deployments.  
As we talked about, now we're also going to see deployment stacks, but different than deployments, a deployment stacks is a full resource. So what I mean by that is that it has its own resource ID.

 **I.S.** 7:33  
OK, that makes sense.

 **Angel Perez** 7:34  
So the stack resource is a deployment.  
It it is a resource that keeps track of a deployment object at a given scope and you can use RBAC to block folks from having access to that method of deployment. So the important thing to quickly get with stacks is stacks has this concept of manage resources.  
So everything that's here in this template is like the state of that stack. So if I were to create a deployment before I was just using a commandlet for subscription deployment, now it's deployment stack and alongside with the template.  
While that we already pass in this qtest dot bicep, I have some stacks specific properties. So the first one here is action on unmanage. The first capability stacks brings to the table is an ability to do things automatically to manage resources.  
So as I showed, this template pretty much represents that state of the stack. Action on a manage is set to delete all. There's three possible states. It can either be delete all, delete resources, or detach all.  
Detach all basically just says break the association of that resource and the stacks resource and delete all is effectively gives us stacks of permission to delete that resource. So let's say for example an SFI item came in or something that flagged a specific.  
Resource that we can command F in our bicep file. You could find that and just remove that resource altogether. I'm just gonna comment it out rather than deleting it.  
And then submit an update to that.  
Stack with the same template as is. What's going to end up happening is.  
These managed resources, some of them are gonna become unmanaged because they no longer show up in the template.  
So what that means is the stacks is going to say what permissions do I have, what can I do to these newly unmanaged resources? And that's where the delete comes in. So one capability we're bringing into deployments is this ability to do a clean up very efficiently.

 **I.S.** 9:48  
This.  
Read.

 **Angel Perez** 10:02  
Up very efficiently for test environments. We see this as the biggest value. You know, a lot of times we spin up these subscriptions and have a bunch of stuff that gets created and can sometimes remain idle unnecessarily. Here you have an easy mechanism to.  
Uh, delete that all in one go. So the all in one go would be we delete the stack.

 **I.S.** 10:22  
But.

 **Angel Perez** 10:26  
And delete everything in it. The more surgical delete is as we saw, we find what we want to remove in VS code and and we run that.  
So that's one capability of the stack. Another that I'll show quickly as this runs is how do we protect these managed resources from unwanted changes? That's where deny settings come in.  
Deny settings is in a way that stacks has to create deny assignments on the resources that it manages and you're able to create a deny, write and delete or deny deletes uh lock if you will.  
What this basically does is all the resources at the stack manages, even though I have access to them. So let's say I take this Vnet uh resource for example or the resource group.  
My principal ID has access to these scopes and I have uh read and write access to these scopes, but if I attempt to do anything.  
I'll quickly get failed and it says even though you have access to this, there's this resource ID deployment stack that has a deny assignment on this resource blocking it. So those are the two things that stacks brings to the table to a traditional deployment.

 **I.S.** 11:45  
OK, got it. That makes sense. That makes sense. That is good.  
Mhm.

 **Angel Perez** 11:53  
There's more advanced scenarios, but we can dive into those later. For the sake of covering stacks very holistically, this is we can see this was done.  
Um.  
So what ends up happening here is you can see those storage accounts are no longer here. The stack took care of deleting those.

 **I.S.** 12:19  
I think that's cool, that's cool. As far as you have a resource, you can always add more features on it too. So I think maybe you are saying you start with these two, but you can actually support more production safe operations.

 **Angel Perez** 12:26  
Yeah.  
Well, those are the two capabilities the stack has. I think for deny settings the only other scenarios is you can exclude certain people or certain actions from a deny setting. So let's say for example you want to lock everything down as much as possible.

 **I.S.** 12:37  
Hmm.

 **Angel Perez** 12:53  
Possible, but maybe your principal ID and the one of your leads on your team should have an exclusion such that you can go and manually delete something if you need to. But these are more just levers that you have but don't necessarily need to use.

 **I.S.** 13:04  
Yeah, that makes sense.  
OK.  
OK.

 **Angel Perez** 13:11  
Um, so beyond this, what brings us back to AV2 if you want to take over Saebom Lee?

 **Saebom Lee** 13:22  
OK.

 **Angel Perez** 13:29  
Um, so stacks. Go ahead.

 **Saebom Lee** 13:30  
OK.

 **I.S.** 13:30  
So.

 **Angel Perez** 13:37  
Yeah. So stacks integrated into EV2 is something that we're close to being done with as something. If it's something you're interested in, we can have your team on the preview list for it later this month when it becomes available.  
As it relates to what we came to talk about in this meeting, it's EV2 stacks. What if? So stacks is a prerequisite to this. That's why I kinda went over it very briefly. But what if?  
Is basically our validation, our preview validation API in Azure deployments. It exists today and you can do a what if any ARM template, bicep template you have and it's gonna give you a diff of the before state of what resources look like in Azure today.  
And the potential new state after those changes you just made to a template and it gives you a kind of resource diff of that before and after stacks. What if is gonna be a new?  
API. So we have the regular what if that's tied to the deployment object. But since we have this new deployment stacks object, we need to create a new what if API and this image you're seeing here is an output of that.

 **I.S.** 14:55  
Mhm.

 **Angel Perez** 14:59  
And more or less what what it looks like, I think. Mm-hmm.

 **I.S.** 15:02  
One second. One second.  
So I'm back. I'm back.

 **Angel Perez** 15:22  
You're good. Um.

 **I.S.** 15:23  
Yeah.

 **Angel Perez** 15:26  
So yeah, this output here. I think the the main feedback we're looking for is gonna be in the next couple of slides. But just to give you the context, what what if is gonna show you is First off a diff.

 **I.S.** 15:31  
Yes.  
OK.

 **Angel Perez** 15:40  
If of what's happening before and after, if any properties are changing, and if the relationship between that resource and the stack is changing, we can see this a little bit more visually in the next couple of slides.

 **I.S.** 15:57  
Let's do that makes sense, makes sense. This all looks good to me. The only question I have is if you have to onboard to deployment stacks, how much effort is is it going to be?

 **Angel Perez** 16:09  
So it'll be as simple as in the service resource definition, you're gonna add a Jason block that's called stack properties and the configurations inside of that and just adding that is gonna turn that deployment into a deployment stack.

 **I.S.** 16:18  
Hm.  
Uh, OK, then I think.

 **Angel Perez** 16:28  
So there it's not gonna be a a big like in terms of using it day one, it's not gonna be a huge investment. What might change is now that you have this concept of managing resources, you're gonna wanna group together resources that share the same lifecycle.  
But that should be, you know, deleted on the same cadence or what have you, and that might require moving some resource definitions around to the same templates you want them in so that the one template can be managed by the one stack you're you're deploying.

 **I.S.** 17:07  
Right. That makes sense. That makes sense.

 **Angel Perez** 17:09  
Saebom Lee, can we move to the the next views? I think we're missing one from Asima, but not getting a response from her.

 **Saebom Lee** 17:13  
Sure.  
OK. Um, do you think we can move forward with the screens?

 **Angel Perez** 17:22  
Yeah.

 **Saebom Lee** 17:23  
OK, sounds good. So here I think it will be familiar to you like usable EV2 portal. What we're trying to do is that as you just seen the what if output, I want to see where would be the best place for you to to.

 **I.S.** 17:37  
OK.

 **Saebom Lee** 17:43  
Reflect those what if results on the EB2 portal as well so that you you can also check up with the what if results on the portal. So here what we're thinking is that we are very open to whatever your idea.  
I'm just like having we just have this like one path, one design for you for you to work as a conversation starter. So like our one idea here is that so list view graphically view like which are very familiar to you we can have another tab.  
Call it what if results?  
So you can see the what if results here in this format. What do you think about? So this tab here? So this part will change accordingly 'cause you clicked the what if results tab. What is your understanding about this?  
Screen.

 **I.S.** 18:44  
And it is interesting. So um and this you are going to show when before the deployment after the deployment because this.  
I think can you also go back a little bit like I think in the overall flow, when does it actually show up?

 **Saebom Lee** 18:56  
What would be here? That's short.  
So this is the usual list view that you might be very familiar to. One way is that we can have another tab for what if results and here if you can find the information that you saw in the CLI screen.

 **I.S.** 19:08  
Hmm.

 **Saebom Lee** 19:19  
Then where would you like to click and go to find that information?

 **I.S.** 19:25  
I think the most interesting one for me would be I will jump. I think when I when I see the screen I would be jumping into the delete ones and I would be very much interested to see what's happening on that because it creates are OK, modify is fine, I know deletes are the but I would be interested in.  
Oh, and maybe.

 **Saebom Lee** 19:42  
Mhm.

 **I.S.** 19:46  
I'm not really sure. It has a lot more information. I think it has all the columns. See only one will be true. Or is it like are you saying all three could be true too? So what is this showing to me right now, especially the rows? Can you explain me the rows here?

 **Saebom Lee** 20:04  
So for here we have create, delete and group, delete and modify. I'm trying to refrain myself to explain what these are. Instead I would like to see it's easier for you to imagine or understand like there are these and the these should be this.

 **I.S.** 20:06  
Hmm.

 **Angel Perez** 20:19  
I think same on for this that that part is fine. So basically these are the what if change types. So in that previous view that we saw if you go to that one same bum, we can see those at the top very quickly.

 **Saebom Lee** 20:35  
Oh, OK.

 **Angel Perez** 20:39  
You see at the very top you're gonna have the different symbols calling out the different changes that were found in this. What if it can be a create, it can be a modify of a resource, it can be a deletion of a resource. So if you move on to that slide we were on.

 **I.S.** 20:43  
Hmm.  
Yeah.  
And can be deleted.

 **Angel Perez** 20:57  
That's what that's getting at. In EV2 portal we're gonna be able to show a change type and resource name summary, but that's the that's as far as we we bring to the EV2 portal.

 **I.S.** 21:00  
8.  
And we have limitations on the AV2 portal.  
Right now.

 **Angel Perez** 21:17  
Today the fact that there's no RBAC, so sending the full what if result is tricky because even though what if ignores everything marked as secure string, there might be RPS that didn't define stuff properly in their manifest and secrets might be included in that.

 **I.S.** 21:20  
Mhm.

 **Angel Perez** 21:37  
That result. So to avoid folks in EV2 which has no RBAC, see in each other's rollouts that may contain secrets on accident. What we're doing is in the EV2 portal, we're including this change type and resource name summary.  
I think the if you move to the next slide, Saebom Lee, we see the list, the full list, yeah, but then in.

 **I.S.** 22:02  
OK, once I click on one of them, this is what I would see.

 **Angel Perez** 22:04  
Mm-hmm. Yep.

 **I.S.** 22:07  
OK.

 **Angel Perez** 22:08  
And then in the so if we if we see here in this slide, there's a link to the what if result resource in the Azure portal itself. So if you go to that next slide, say Bum.

 **I.S.** 22:09  
Uh.

 **Angel Perez** 22:25  
We might actually have that one.  
Yeah, that's gonna jump here. So you see how on the side we have deployments, deployment stacks, and now there's this new what if results that's gonna be capturing those different what if evaluations you've done for a given deployment stack.  
So here in in the Azure portal we actually have RBAC. So folks that click into that if they don't have access are not gonna be able to see the full what if results. So the secrets problem isn't it's not a security concern in this case the this.

 **I.S.** 23:01  
And what if result is also a resource type today or no? What is it OK?

 **Angel Perez** 23:05  
Not today. Today it's just an ephemeral thing output that gets returned to a client, but now we're gonna create it as its own resource type for that case of being able to.  
Link from EV2 to this object here we're tracking in the portal.

 **I.S.** 23:25  
Hmm, I see. OK, OK, so one question. So when I click on this, what if results right now?  
How? How do I know which one is this associated with in this screen?

 **Angel Perez** 23:44  
So in this screen you're just gonna see all the different resources. In the previous one we were in the context of a specific deployment stack. So within a specific deploy action you have your deployment stack and your deployment stack.

 **I.S.** 23:53  
Mhm.  
Yeah.

 **Angel Perez** 24:00  
What if result? So clicking here is gonna take us to that exact same result in the portal. So if we go to the next slide, what we see is a list of all those resources that match that create and delete.  
Um, change type.  
Um, so if we click into one of these resources and let's see if that next slide is there, say bomb.

 **Saebom Lee** 24:28  
OK.

 **Angel Perez** 24:30  
Oh, it's not. That's the one I need from Asima. Basically, if we click into one of these, you're gonna see the specific JSON diff for each resource as you click through them. So you can go like, oh.

 **Saebom Lee** 24:34  
OK.

 **Angel Perez** 24:45  
Uh, a container app's getting deleted. Let me see what what the definition of that in the template is, what's getting removed. Um.  
But yeah, I think this. I need that final view to land that point home. Let me see if I see my consent at my way.

 **Saebom Lee** 25:11  
OK.

 **I.S.** 25:15  
I think one of the questions my team had was.  
Do we have any plans to have EV2 as an extension in the portal itself so we are not switching between the other two?  
Back and forth between the portal and the UV2 portal.

 **Angel Perez** 25:33  
That's a good question. I don't know.

 **I.S.** 25:34  
That's a good question.

 **Angel Perez** 25:37  
If there's any specific restraints to that outside of the one around our back that I mentioned earlier, but yeah, I could definitely bring it up to to my team.  
2.

 **I.S.** 25:53  
Yeah, if we can do that because I was talking to any also any people who is our CVP in the X in New York and I think one of the and I was actually talking to some of the people in general, my team and some of the trends also I'm seeing right. I think there are three things here and which is.

 **Angel Perez** 25:58  
Mhm.  
Yep.  
Hm.

 **I.S.** 26:13  
Actually hops and maybe for us might be OK because we have a lot of other tools we go through safely, automation, R2D and stuff. But then I think the developers outside don't don't really have to go through all the steps. So you go to.

 **Angel Perez** 26:23  
Hmm.  
Yeah.

 **I.S.** 26:30  
One of the ID which is VS code or you know Visual Studio or whatever Azure DevOps or maybe GitHub. You do those GitHub, sorry the bicep changes and you actually go to the deployment. Then EV2 portal comes in. If you're using EV2 then you go to Azure portal and try.  
To see what really happened. And there's a lot of hops, per se, lots of hops.

 **Angel Perez** 26:50  
Yep, yeah, there is indeed. And then if you count extensions as well having to be authored separately, it's a whole mess. So I think from the deployments team side, we hear you and we're doing our best to try and bring all.

 **I.S.** 26:57  
Mm.  
Yeah.

 **Angel Perez** 27:10  
All that goodness that exists for our third party customers into the first party realm. I think the integration of Bicep into EV2 is a big part of that. So now all of these extensions that different teams author.

 **I.S.** 27:15  
Mhm.  
Hmm.  
Right.

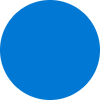
 **Angel Perez** 27:26  
As well as AV2 team authors, you're gonna be able to author them all using Bicep, so you could just have that same workflow for all of those extensions and also just reuse ones that some like the AV2 team.  
Authored and and saw as a common pattern across a lot of different services and be able to leverage that just kind of reducing the amount of of hopping being done. In terms of the UI itself, I think the main thing that's limiting folks today is the fact that RBAC is.

 **I.S.** 27:54  
That makes sense. Mm-hmm.

 **Angel Perez** 28:03  
Isn't in EV2, I think why? That's why you see a lot of hopping from there to to the Azure portal.  
Do we implement RBAC into EV2? Maybe that's what solves all that, but there's already too many experience living in both that I think what you said is a a really good point of discussion.

 **I.S.** 28:27  
Yeah, that makes sense. But I think let's go back to our user research we are doing today for this one. So maybe Saebom Lee, if you can go back to the slide you were showing me where maybe the two slides back two, we have one more back.

 **Angel Perez** 28:32  
Mhm.

28:37  
Yeah.

 **I.S.** 28:42  
I think starting with this one right where this makes sense. I think see the reason I would say it's still OK is.  
You just get it like for example once you are here you get all the options and then what if is is actually also in front of you. So it should be OK because I don't think they think there is any other way where we can show up. It will be another click here you are like here and then you're just getting more information the more you go into the tab.  
Which might be OK. I think it's good enough. The can you click on the like next slide?

 **Saebom Lee** 29:13  
No.  
Yes, sure.

 **I.S.** 29:21  
Yeah, so when you go when you click on one of the deploy and then you see the screen and then here you get deployment stack you're associated with the deployment stack. What if result? What is the difference between the two deployment stack? What if result and deployment stack?

 **Angel Perez** 29:43  
So stacks is that stack resource I demoed at the beginning and the what if result is this new what if result resource?

 **I.S.** 29:45  
Hmm.  
Yeah.  
Oh.

 **Angel Perez** 29:53  
It the naming is tricky cause basically you won't. The only what if that's gonna be present in EB2 is gonna be stacks. What if it's not to be confused with the regular deployment? What if stacks? What if is built on top of the regular deployment? What if? But you also have the nuance of getting the.  
The property diff of the stacks resource itself. Like was there any changes to the action on unmanaged setting that I had at the beginning or or any of those things of of that nature?

 **I.S.** 30:19  
Uh, OK, OK, OK.  
OK, and then it will have a unique ID and then you.

 **Angel Perez** 30:32  
Yep.

 **I.S.** 30:34  
And can you have a multiple word if is associated with the stack or is it always gonna be one-on-one? Like how is the mapping?

 **Angel Perez** 30:39  
You can have multiple what if results. You're always gonna just see the the latest one that you ran. But yeah, you can have multiple.

 **I.S.** 30:52  
OK, this makes sense. And what is the switch to graphical view?

 **Angel Perez** 31:01  
Uh, that's a good question. I'm not sure if that's meant to be there. That might be from a past mock up that that's the money.

 **Saebom Lee** 31:07  
Um.  
Yeah, but yeah, yeah. Another thing that like the other user mentioned that they also would like to see some clues regarding where they could see the what if results in term like when when they're exploring through graphical view.  
So it's not a specific question that I have, but do you also find the need to enter work results from graphical view at this moment?

 **I.S.** 31:42  
I am not really sure because some of our deployments are gonna be really huge and bulky and I'm not sure what it will be able to show. Like if you're showing just the divs, maybe it will not be a complete picture and if you're showing everything, it's gonna be like too much.

 **Saebom Lee** 31:48  
Yeah.  
Mhm.

 **I.S.** 31:59  
Like how are you gonna zoom in, zoom out?

 **Saebom Lee** 32:00  
Yeah, that's good. Are you currently using list view the more or the graphical view?

 **I.S.** 32:07  
I'm sure we use a list too.

 **Saebom Lee** 32:09  
OK, OK.  
Got it. Um.  
So if you look at the part here, do you find any like a missing information that would be great to have here on top of this?

 **I.S.** 32:33  
I think one of the thing which just an ask I maybe but I'm I'll tell you a use case where what users like especially the engineers in the team want to do is also find out the history.

 **Saebom Lee** 32:45  
Mhm.

 **I.S.** 32:51  
Like are you going to also have some kind of like what if history or something or no it just like the latest what if we have?  
Maybe question to you?

 **Saebom Lee** 33:07  
Um, we don't have it any fixated yet. Um, do you find the need to have a list of history of what if results?

 **I.S.** 33:11  
Hmm.  
Mhm.

 **Angel Perez** 33:18  
Yeah, that'll be in the Azure portal. Um, so we actually have those views, but they're not uh for some reason in these slides. But yeah, in at each scope, just like you would see the list of deployments.

 **Saebom Lee** 33:18  
OK.

 **Angel Perez** 33:35  
Or the list of deployment stacks should be able to see the list of deployment of what if results run for a given scope.

 **I.S.** 33:44  
OK, that would be good. That would be good.

 **Angel Perez** 33:44  
Um, so this this image is a bit, uh, misleading here.

 **I.S.** 33:50  
Yeah, that's what I was thinking because this even this Azure portal is not giving me all the information right now because it says whatever it is, but I don't even know what deployment or stack it is associated with. Anyways, if you go back to the previous slide, yeah, let's stick to this one.

 **Saebom Lee** 34:03  
Yes.

 **I.S.** 34:07  
This is good. I think the left side you are actually telling the operations, operations you're performing, you're showing the stack information. Obviously you have a drop down where you're saying action history, so maybe that is where you are saying we can pick.  
And we would be able to show something. Then you have an entire requirements type to go and then you are showing the create, delete, group, delete. What is a group delete?

 **Angel Perez** 34:41  
That is a resource group delete. A lot of the things you're calling out actually were already updated, so I'm trying to get the designer in so though that latest view gets shown.

 **I.S.** 34:51  
So.  
OK, OK, OK, OK.

 **Angel Perez** 34:57  
But yeah, uh, this.  
Good catch.  
Then and the other views that you were asking about. So I'm like, I thought they were already already here.

 **Saebom Lee** 35:15  
OK.

 **I.S.** 35:18  
OK, that makes sense. This is good. This is good. This is reasonable. Yeah, it's reasonable.

 **Saebom Lee** 35:24  
Mhm.

 **I.S.** 35:31  
From here can I actually see I'm just trying to see do I really have to go double click and can I get some information here itself so I don't really have to go one more step or get that get to that window.

 **Angel Perez** 35:31  
Awesome.

 **Saebom Lee** 35:43  
Mhm.

 **I.S.** 35:46  
And close that window.

 **Angel Perez** 35:48  
Yep.

 **I.S.** 35:49  
So here you are going to show all and then all the time. Then you're also going to tell me and here you're not going to tell the resource names, right? You'll just say 3 resource, one resource.  
When I hover over those resources, will you be able to tell me the names or no?

 **Saebom Lee** 36:07  
Is that what you're looking for? So like hover over it?

 **I.S.** 36:09  
No, I was just thinking maybe I can quickly like for example if I'm looking for a particular resource which got deleted and like I had a lot of operations which ran, maybe can I just quickly go through this window and figure out where it really happened. So maybe all the deletes I just hold.

 **Saebom Lee** 36:14  
Mm.  
M.

 **I.S.** 36:24  
I get the names.

 **Saebom Lee** 36:26  
OK, gotcha. Anything else that you would like to have on this screen up front before digging into the next play? So for example, you could have like not just these four columns, you can add 1 column and then there's one more information that you would like to have.  
Of like any types of information that you would like to add on to this screen.

 **I.S.** 36:53  
I think the only thing which would be very see create delete, maybe group delete was confusing but I think now you mentioned it's a resource group delete. Modify will be very interesting because it's a lot of different kinds of modifications which could happen like it's a property update or like.  
What exactly the the authentication updates? I think there's a lot of different kind of updates which can happen, but I think anyways you we cannot go to details on this one, so.

 **Saebom Lee** 37:24  
8.

 **I.S.** 37:25  
No, it failed. Succeed. Are you telling anything on those sides or no? Because only when I click I come to know when you click on this one, go to the next slide and you're telling me it failed or succeeded. But here in this screen you don't tell me anything.

 **Saebom Lee** 37:41  
Got it. Um, so like having a column reflecting like fail succeed will be good.

 **I.S.** 37:47  
Maybe or just an icon maybe or something which I should fence me because I am getting. So what I'm trying to see is like I most of the time people don't really want to click and go back, click and go back, click and go back. So a lot of time they just get information obviously maybe and then the key thing you're looking after then you go and then use it.

 **Saebom Lee** 37:49  
OK.  
It makes sense.

 **I.S.** 38:07  
And uh, what is missing here? I think from that perspective.

 **Saebom Lee** 38:11  
Mhm.

 **I.S.** 38:12  
Mhm.  
But this is good otherwise.

 **Saebom Lee** 38:16  
OK, um, we passed the delete scenario question. Do we still wanna revisit it or is it good?

 **I.S.** 38:26  
Who?

 **Angel Perez** 38:26  
I think it it's fine for now.

 **Saebom Lee** 38:28  
OK. OK. Sounds good.  
I think that's all I have on my side of questions. Imran, do you have any further questions that you would like to ask us?

 **I.S.** 38:42  
No, I think this looks cool. I think the deployment stack looks cool. I think I can introduce folks from my side to you who can actually be an early adopter so we can adopt maybe maybe onboard our test PP environments.

 **Saebom Lee** 38:50  
Good.

 **I.S.** 38:58  
And uh, so this way we can also give early feedback.

 **Saebom Lee** 38:58  
Good.  
OK. OK. That'll be great. Just share me their contacts or any way that you prefer.

 **I.S.** 39:10  
I can do the interaction. I can actually, yeah.

 **Saebom Lee** 39:12  
Sounds good. Awesome. Azima Angel, do we have more question?

 **I.S.** 39:15  
Other teams.

 **Angel Perez** 39:23  
No.

 **Saebom Lee** 39:25  
OK. OK. Then thank you, Imran. We're gonna, we're gonna gather your feedback and then try to reflect those into the next design. Thank you so much for your time.

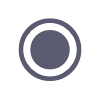
 **I.S.** 39:35  
OK.  
Absolutely. I'm really sorry for 10 minutes late. The guy just lost the time in person lens here.

 **Saebom Lee** 39:41  
OK.  
No worries. I'm sorry too. We had to see you early. Yeah. Thank you, team. Then talk to you later. Bye.

 **Angel Perez** 39:44  
All good, no worries at all.

 **I.S.** 39:47  
Awesome.  
Bye.

 **Angel Perez** 39:52  
Thanks. See ya.

 **Saebom Lee** stopped transcription