**Part 7 [VHD-2, AWE]**

August 21, 2023, 3:35PM

1h 1m 19s

 **Dinesh Vijayakumar** started transcription

 **Leopoldo Estrada Vargas** 0:10  
You can see my screen right?

 **Dinesh Vijayakumar** 0:13  
Yes.

 **Leopoldo Estrada Vargas** 0:16  
Thanks.  
Yeah.  
So a couple of weeks back, we started with this agenda right about the BSD.  
So we started with some background.  
I think we started explain right how publishing happens for PC tax, right?  
So I just showed you again.  
There's less for the PSD's, yeah.  
So basically these are the update types, right?  
We police on on media and we have several types for disease because we published two different audiences for different products, right?

 **Tavishi Gupta** joined the meeting

 **Leopoldo Estrada Vargas** 0:50  
Umm, so we have several really tickets because for each of these updates we may support different products as in the following table, right?  
So in my in for me, for me that busy we police around 40 tickets, I think it's 41 or so or 40.  
So the audiences will publish this with this tool are before you write.  
So we publish to the regular Azure marketplace.  
This is one of the audiences.  
Umm, there's a special marketplace for Azure Stack which we also publish right?  
So usually this means is are a subset of the original marketplace images.  
Yeah, you have the marketplaces for for the polyglot saying, right?  
And also we have the internal realities.  
There is a like a regular.  
I'm gonna call you regular.  
The OR original PME gallery that mostly test base team uses right?  
But recently, like this year, we started policing images to a special, also internal gallery.  
There's gonna be used for for Azure, right?  
For the pretty standard initiative.  
So every BD that we police can go to when any of these, uh, how this is right.

 **Kruti Kallola Mohanta (WSD)** joined the meeting

 **Leopoldo Estrada Vargas** 2:05  
So I think I think we more or less explain the the marketplace policing flow, the first one.

 **Tavishi Gupta** joined the meeting

 **Leopoldo Estrada Vargas** 2:12  
Uh and the logicapps I I can.  
I can do a a recap later, but let let's go to the umm to the gallery.  
So and and you know small refresh, right?  
So this is very simple.  
So we we publish images to the Azure compatibility umm.  
Remember, when you use partner center right?  
As in marketplace in in marketplace case we have partner center that exposes the UI and rest API for person.  
Usually we don't use it the UI.  
Of course, everything is in automation.  
VRS API, but we still have it.  
Wait, we still have this nice portal where we can see, and I don't know, maybe check some computation or debug some.

 **Achal Shah** joined the meeting

 **Leopoldo Estrada Vargas** 2:52  
You said, right?  
So for the gallery we only have the Azure Rest API, but it's also also very simple.

 **Tavishi Gupta** joined the meeting

 **Leopoldo Estrada Vargas** 3:00  
Just write you basically just just to give the presence for deployment and you you can create emails if you need some image versions, right.  
Similar to the skills and images that we publish marketplace, you have to define the imagination and the version.  
Yeah, basically this is how how you define an image in the gallery.  
So how this works?  
Maybe I mentioned this last the two weeks back.  
So will use an asynchronous adapter implementation.  
So basically it when when we receive an an an activity from AWE we target after the workflows but they get a mediator service which called the six policy manager.  
So this service just creates a document in Cosmos after some validation and stopped.  
Yeah.  
So not, not unless he's done by this by this Singh policy manager.  
Now the adapter for the cigarette after in this case is pulling for work from Cosmos.  
I think everything 20 minutes or so.  
So when there's another document that is not complete yet or not not notified back to Italy, so the doctor now publicity images to the gallery.  
Yeah, this is the the adapter pattern that we implemented.  
Uh, if you there's a back umm, once the position is complete, the adapter will notify back to Italy.  
Yeah, media mediator service and the activity will be complete right?  
Ohh like I should write the code is for example for the Raptors you could see that at the right so that so the adapter because the three main operations right one operation is invoked.  
So invoke request means last year the operation on the back end, right?  
In this case it is Azure for some if.  
If we're gonna create an image right, the quality with with version one point 2.3 for example, we'll start with this operation and then let's say go to sleep for a few minutes, right?

 **Leslie Trowbridge** joined the meeting

 **Leopoldo Estrada Vargas** 5:01  
So I I think every 15 minutes or so because it's operations take a few hours every 15 minutes or so, we the adapter polls again for work and he discovered that the operation is already started.  
So next step is to monitor right and so if we already started this operation we just monitor that auto completion and every 15 minutes.  
So these these monitor request operation is being called.  
If the if the presence is complete on the back end, think yeah, laptop called the operation which is notified right?  
So what would he means?  
It may mean notify clients right?  
Four, notify Italy, so all the notification notify operations happen in this give me the right.  
This is how the adapter works with these three operations, right?  
Uh.  
Invoke Monitor notified the the only only two adapters that we have is for for Sharma, Salini or Azure Compute Gallery which is called now right?  
And and the Azure stack marketplace, they started to adopt us.  
We have right and they and they are basically the same flow you know.  
So we trigger these service to prison document in Cosmos and the data just polls right for work to do and then complete, right.  
So I'll I'll switch to the marketplace.  
Do you?  
Because he's basically the same implementation, right?  
Except for the input.  
So in this case, we don't go directly to the Azure resource, so the actual stack team provided an endpoint.  
The probably they stuck in marketplace images too right?  
But but the flow is same.  
So we're stuck with an AWE impression.  
That request, one of these workflows right.  
In this case they publish ohh, but you kept so like we'll call the mediator service and then the adaptive implementation comes in, right?  
So we're going to call an internal service, yeah.  
So start service which puts a dozen validation and then puts a a document in Cosmos DB.  
In this case this different collection the adapter pulse every.  
I think every 20 minutes fortunately missed to the endpoint and wait for completion, right is it's basically the same this inflow now the last audience for SSD, they won't be actually completely, it's slightly different.  
We wouldn't use an adapter because they they won't be already is special.  
Yeah.  
So first of all is is he lives in another tenant.  
You need some on Amy tenant and also I because it's gonna be consumed by Azure in multiple clouds and regions.  
Well, we have to use something different, right to publish right in, in.  
In this case, we're kind of of locking.  
Let's say that Azure, but they provided some some implementation for the back end, right?  
So similar to similar to partner center will reduce this pipelines publish misses there right?  
So so this is the flow to police one.  
When you missed that special one piggledy, so first as as any other media type, we have to ingest the images.  
So in ingestion is done by the ingestion workflow on ingestion service.  
So this is the one of the first logic apps that are trigger right for release ticket.  
So ingest means let's create a set of of release documents.  
Out of this proposal metadata documents, they're gonna be used later, right?  
For the publishing, yeah.  
So Jason is basically ohh position creation of this document and also for for most of the data types we send images to ESRP for scanning him.  
So you said piece is happening in parallel.  
It doesn't block publishing.  
So where one thing isn't it should operation is done.  
Ohh will notify back to Italy to continue with the play but these are piece scanning is happening right on on the back end in parallel.  
Yeah, after ingestion is done, it will.  
It will call the public activity, so let me go.  
Sorry, I'm going back and forth, right.  
So we're gonna try to get the police.  
Ohh workflow to do this.  
The the next activity, right?  
So publish operation means that we're gonna do basically three operations, right for this, right?  
So the first thing is we're gonna have to copy the images over to AME tenant because the images are stored on on in PME.  
Yeah.  
So this is where HCC for SMF pushes the images to only only to PME.  
So we have to copy to to AME and this is where the the contact starts right to have special contact with Azure.  
Well, we'll have to put the images in a special place.  
With a special name for the continued name on the storage that we use.  
Uh has the image version right for some this case is 2 zero 348-1487-2303 ten.  
This is the version that comes from SMF and we're going to use this exact version.  
Uh for the in the Ali.  
Remember the the mission.  
The needs to have a name or imagination and a version right like one point 2.3 and the BLOB name.  
Uh, it has to be changed to exactly match the immediate finition name, right?  
But somebody we're gonna police to 22 to dress enter.  
So the BLOB name has to have the has his name.  
Dot BSD, yeah.  
This is the contract for copying the message to the storage after the messages are there in automation, we publish a build artifact pipeline.  
So these these these basically what I'm gonna just generate some artifacts for Azure pipelines to to to interpret or use and also with you is gonna prepare some metadata for the EV2 pull out speech right.  
So what?  
Not relax person EV2, but he'll be Tuesday 2.  
That Ashar teams uses to push images to all regions and clouds, right.  
Similar to marketplace that also uses EV2 to do the replication, Rd clouds.  
In this case, we we reuse right?  
That implementation from Azure, yeah.  
So we propose prepare some with the data for for the images in this.  
In this step, let's see.  
And then we're we're ready to to three year the real policy, right, so so the last part, but they want to get exactly to get release pipeline.  
So this list pipeline will push images 1 by 1 and using EV2 to all regions and clouds right the the first thing also actually waiting for an approval today.  
This approval is manual and this created that ESRP only when you have been is complete.  
Somebody like like 5 or sample data has access to this pipeline.  
Can approve the list right?  
As I mentioned, this is manual today, but it will be automated soon.  
I hope right so that you will still have the same gates.  
We will be totally automated, right?  
And and the gate succeeds because we will have to make sure that images are good, right, that they're they're, I know.  
Like they approved LCU that, uh, you, Sophie?  
Succeed. Right?  
So what?  
Wouldn't want want to send boggy or prerelease and approve images to the guy, right?  
Because once the images are there, they can be consumed by anybody in Azure, right?  
You, even when they have also deployment.  
Ohhh, alright.  
Process very messages will be already arrived for anything to consume because images are widely shared to Amy and PME tenants.  
One once in an image, is there anybody on on Microsoft can see the image right as long as they have subscription in those tenants.  
So then the restriction of happens that they use to to pick it to all clouds.  
Uh, starting for the from public and the other including air gap, right.  
So once the polling completes, they miss is already available in the gallery.  
So this is just a UI view or how can you discover the calories you can go to the portal, for example using PME or AME tenant.  
Hmm, you can in this case I'm using the AME example right number one.  
So you can log also as PM and it will work.  
So you select for example creating a resource and create a virtual machine, but he said looking in the marketplace for a specific image in on the public marketplace, you have to go to directory images.  
Yeah.  
And then change the scope to tenant and then you will see the list of images.  
These images and or or the available images are not only for Windows.  
There are also Mariner images and .net images.  
Yeah, like windows and .net.  
Yeah.  
In this list there are several images like eight.  
I think today we only support 4 because with the packaging one which is yeah, maybe I can show you one quick example.  
For what we have saved, right?  
I think I can look.  
You mean?  
Also there is.  
There is also a way to query the images or even using right BR API and the only thing you need is a subscription on the allowed tenant right?  
For some in this case you're using PME.  
You only need a subscription on PME and you will be able to query the message.  
Would you happen to have explicit access right?  
Because it misses are already shared.  
When you go to your virtual machine.  
Just one second.  
What's about here?  
I I can see your old images right?  
Like I click on this button you can say for the marketplace for some you can search for Windows Server images.  
Yeah, tell the public images image right when in this case we are looking for the internal images, right?  
The images are not shared yet with the world.  
I I think they they will actually be never survive with work because this these are hardened for us specifically.  
So we have to go to the share images, change the scope to tenant.  
Nick outside of this box and then that's it.  
So you can see even the Canary images on the Microsoft tenant and they're real.  
He missed right on on AME that that are also shared 2:00 PM me right.  
Ohh so this is it.  
Any questions so far?  
I know two weeks back I explained the marketplace, but I hope you still remember little bit about about this workflow, so feel free to jump to any question and I can jump to the next on final section.  
Just I don't.

 **Achal Shah** 15:51  
He take a take.

 **Leopoldo Estrada Vargas** 15:52  
I don't see the hands, so feel free to speak this.

 **Achal Shah** 15:55  
Follow quick question, if you when you're describing the adapter right.

 **Leopoldo Estrada Vargas** 15:57  
Sure.

 **Achal Shah** 15:59  
So we have the Cosmos DB which, yeah, that's a good actor.  
I think that's the one second after collection which acts as in as a way for the adapter to know that there's some new work, right?

 **Leopoldo Estrada Vargas** 16:12  
Right.

 **Achal Shah** 16:12  
Is this just runtime stuff or is this?  
Does this need to be kept or is it?  
Is this data remote later on?

 **Leopoldo Estrada Vargas** 16:20  
No, there is kept, let's say forever.  
So with with usually don't delete documents right from the laptop.

 **Achal Shah** 16:23  
It is OK.

 **Leopoldo Estrada Vargas** 16:26  
So I I think this the story goes more or less like this I some few years back.  
Uh, no.  
You guys were limited on a new right for a lot of people.  
So for example, for the early case, so you got some some that didn't work because they only lasted for a month or so.  
It was just back for for today, the last for two years or so. Right?  
So so in in the legacy, let's say 1 Pop without system low, you guys were still not ready, that's for 20.  
All the workflows right?  
So so we had to develop some a different solution, very similar right to the Logicapps scope, but that can last longer.  
So I think this is when we created adapter implementation.  
So for media, we don't really need the adapter to live for more than a month or so.  
It's just maybe for a few days, right?  
Or maybe a few hours, but still what?  
I use the idea to have to have some requests from Cosmos and then Paul right for it's like a like an asynchronous kind of pattern, right?  
This is what why we have the adapters today because people already have expertise like when.  
There was a guy named Amanda that was very familiar with the doctor, so he helped us developing these two adapters for Azure Stack and for C and I don't know which was that because it was easier, Amit because he was proven right that it worked.  
Yeah.  
But but yeah, so you mentioned, so it relies actually on Cosmos DB.

 **Dina Helal** joined the meeting

 **Leopoldo Estrada Vargas** 17:52  
So Cosmos, we have to.  
Keep these documents.  
They can also be deleted.  
Yeah, but actually actually we we just, I I'm sorry, I don't have access to almost but you.  
Just mark them as complete.  
That's you.  
Right.  
So so they are still there, but not artist is not polling for them when they are complete, right?  
This is the pattern.  
Yeah.  
Any other question?  
Thanks achal.

 **Dinesh Vijayakumar** 18:23  
Yeah, well, one thing you said that they'll be polling every 15 minutes, right?

 **Leopoldo Estrada Vargas** 18:25  
Sure.

 **Dinesh Vijayakumar** 18:27  
So right now we will be aware of what would be the time taken in general, right?

 **Leopoldo Estrada Vargas** 18:28  
I think so.

 **Dinesh Vijayakumar** 18:33  
So for publishing this.

 **Leopoldo Estrada Vargas** 18:36  
Yeah.  
Yeah.  
Exactly because images take, I think for Azure Stack marketplace take like one hour because basically they just download the BLOB and the metadata is not a lot of work for the gallery you have to create the image definition and then the image versions and when when there are a few images sometimes takes more than an hour or so.  
I can.  
I can show you right the iterations, so I think polling every 15 minutes doesn't add too much delay.  
So in in the worst case, that's the the biased right, the 15 minutes, the polling time.  
So you mind that you, Paul and it's not complete, but in next minute it actually completes, right?  
So you still have 414 minutes more, right?

 **Dinesh Vijayakumar** 19:18  
And yeah, that's what I was asking.  
Like, is there any uh notification or even that gets triggered when it is complete from the external system?

 **Leopoldo Estrada Vargas** 19:27  
Yeah.  
Yeah, there's something from Italy.  
So Italy for sure, yeah.  
So yeah, so you you mean some ridiculous time?  
Alright, you mind that I pull every every day, right?  
So for a person that takes only one hour, pulling one day is not good, right?  
Because you have, you can waste 23 hours, right?  
So in this case, we pull 50 minutes to not not overload.  
Umm little yeah, after service.

 **Alexey Loginov** joined the meeting

 **Leopoldo Estrada Vargas** 19:54  
And and whipping.  
That's generally OK for these two cases, right?  
For marketplace for stack marketplace and for vitality, maybe we can we can go.  
Even faster?  
Yeah, it's up to us, right.  
Also I want to mention something also that there is an issue with adapting to the sensor.  
Yeah.  
So when you have more than one instance running, you're you're gonna take.  
Yeah.  
Collide you.  
You can have two adapters.  
Uh.  
Potentially right.  
Doing the same work.  
So to prevent that you can use a locking mechanism that we don't have yet or or we have we we don't use or just deploy these adapted to 1 instance right?  
Only one host in the service fabric cluster, right?  
So we decided that for simplicity, when we deployed the update to 1 instance, maybe we could implement a very locking mechanism and then deploy that after to multiple hours.  
So I think we have five hours, 5 hours right in on PME.  
Yeah.  
Or even better, maybe we can now finally migratory caps in this case, or some other implementation.  
Right.  
But we have to rely on the on the other adapters, so the the reason we have them because we the developed them a few years back and they work and generally they work fine they they don't cause a lot of only on the on call.  
Anyway, but I think later we should pick them.

 **Dinesh Vijayakumar** 21:16  
Yeah, makes sense. Yeah.

 **Leopoldo Estrada Vargas** 21:18  
Yeah.  
So for example, if you look for this is the last Explorer page.  
I I created a dashboard on the PME for for yeah on the PME it didn't.  
So you could do a D for example.  
You can filter right.  
This is why you like these tables.  
You can look for the busy types.  
So I can sell it so we can accept these others.  
For sample Publishment published internal audience.  
So, so actually some activities like for physical and CPC with the policy service to the reality because they are not consumed by test base right in that in those cases the activity just completes very fast like in seconds, right?  
Umm, but other first of all I in this case you took one hour this error code, the end of the day, the day it succeed.  
Actually the locks record.  
Look at the other stuff.  
Right, this busy server.  
This is more just one hour, right?  
When I work 51 minutes right.  
Do you wanna say more or less?  
The 15 minute is is OK you can maybe make it more frequent, but today we pull that frequency right once every 15 minutes.  
Yeah, if there there's an issue, you'll see a very high duration of the operation.  
Yeah.  
On until we substitutively right.  
So in this case, I think you said it was a blocker.  
I think it was this or the other one or the other one.  
Anyway, when there's an issue of for the duration was very high.  
You can go even.  
Ohh or for more hours or days right?  
So in some cases, Yep.  
Yeah.  
Any any question I I can show you that also this for a little bit later but let let me go back to the presentation to the final session.  
So there are other things related to visit that don't happen for other other types, right, for example.  
So with valid images right on, on preview on live, yeah, partner work.  
So for for the marketplace case, for example.  
So what's love here?  
To validating the images before the level operation. Yeah.  
So I'll how this works.  
So once we police images to partner, center famous and in preview right, so those those images only only available to the police right to test on them. Yeah.  
So in this case, we consider what's now as publishers, so we add them.  
So specialist in Paris Center, that is the publisher audience so that they can create the info from the preview offer right?  
Even before the offer slide so, So what is what we do is to we policy measures and then send and we should be from one person the person complete right.  
So now we also send those from Italy to the panel service bus.  
Ohh sorry actually this is it.  
Basically bus, but panel has access to it, so panels she didn't notification they they could the message and they're ready to validate, right?  
So the the most important field for these notifications are product release, an update type.  
Once you have these three parameters, basically they they they interfere somehow their list ticket or they're the lizard, they want to validate and they have to call our API.  
I really proposal API to get the data.  
Basically, this data will tell them for for marketplace.  
What is it?  
Publisher offers you an version that they that they want to test, right?  
And what they do is to get they the image right from from Azure, they create a VM and they they check the internal features from from the VM to to to ensure that it has the latest LCU RSS expected right?  
So on the the vacation, we also tell them what is the version so they can match also built under VR. Yeah.  
So if all the images are good.  
Umm, there will show as green in target.  
Do I have a?  
Yeah, a trackit base.  
So for the waaslab page is the the pre live validation.  
So if all the images are OK, then you will see the screen ohh bar right?  
If something is wrong with one of the images, or maybe all of them, you'll see inside.  
I think it it shows as red and you'll have a nice him.  
So they create an ICM for us right to to check you.  
You still the images are OK, some sometimes I don't know there is an issue with the data right or or with the workflow.  
Sometimes the workflow for some, I don't know, bulk finishes before the mission already published, so they they find an error, right?  
But after the retries, usually they're OK.  
Hmm, OK, this is the second validation is done by eliciting uses more of the same model, except that they don't check against their real PM, right?

 **Venkatnagaraju Goursetti** joined the meeting

 **Leopoldo Estrada Vargas** 26:21  
So for SBT is is the same process after the live operation with an interpolation today?  
Will it will service bus and either we represent notification too to service bus that punish use they also the kindnesses get these three parameters, all recent update type.  
They query the API but the validation for for Lt is actually not creating VM's, they just quoted it marketplace recipe I to confirm that the images are available with that specific version.  
Yeah, if the API says that the version is 1.2 point 3 for some skew, you have to query this API for Azure to check that the version is live, right?

 **Eduardo Toraya** joined the meeting

 **Leopoldo Estrada Vargas** 27:01  
If the version is not live, then they they'll sweetie column will not be green will be like red and will have have also nice team right.  
If an image is not like when expected, we we'll have an, uh, an ICM or also they they have a like an hourly workflow.  
So they they put it in the API that we post not only based on the definition, but also efficiently every every hour or so.  
So they they query multiple release tickets, right?  
That that they captured in, it's in the last couple of months.  
They do a lot of calls for for the vision of the types and if if something is live on marketplace, there is not captured in this data, they also create an ICM for us, right?  
Let me do an example in mind that for a parent request I have to publish an an email manually for an offer right?  
Like Windows Server, Windows 10 or Windows 11.  
So I list 50 will catch that they're they're gonna say this.  
This image with this version is not expected right for from the metadata of the of the API.  
So we create an ICM.  
Yeah.  
So at least has this double validation in an image that is unexpected.  
Yeah, is is signal right?  
They race and I seem.  
Or the other way.  
Otherwise, if the animus is expected but not live, we also could not.  
An ICM?  
Yeah, this one, this is this is actually part of the release.  
So for easier release for for the updates that they can apply like marketplace, will you see validation flow right there is there's other special case that is very similar, but I think create a special picture for you, right?  
So the other aspect states raised this space is used to be very similar to to the Waaslab case.  
So for all for all prerelease images that we publish to the to the internal team gallery, not the one P gallery, but the regular Timmy calorie for this phase.  
So they quality misses improvise with some customer scripts I think.  
So they also have to be notified when there's there are new images.  
Initially they started using the same flow.  
Uh.  
Like when we publish something to the daily, we notify it will system, then it will notifies the punish via service bus and test base used to also create the API right on on this in this fashion.  
Yeah, after some improvements, right with a predicted API part for this space.  
So.  
So in the initial notification to the service bus that the thing listens to, we also include image information.  
Yeah.  
So the the notification to test basis.  
Uh.  
More Rich did say maybe you have access to this latest Yep.  
So they don't have to create the API and then just from the passion from the 1st and right the there was there was some uh interaction.  
So they they specify what content they have they want.  
Sorry and what metadata?  
So we umm is not on the immediate. Ohh.  
So anyway, so the they need.  
I I have a copy of the notification that they listen to right?  
Anyway, maybe I need to also deploy the reader?  
Umm.  
What can I see?  
Let's see.  
Let's see.  
So this was the most recent runs for test base, so you can see the update types and the products also the release.  
So this column says the option is done and this other column says that that we notify right then of completion.  
So the notification is is not not sent to the same service boss as initially we actually created a new topic for them with with the more more rich or richer notification right?  
For this base, I don't think I have an example for this.  
Anyway, I'll.  
I'll, I'll try to look at for the Jason later and maybe paste in chat.  
So the notification for test base is has basically the same metadata like product, listen, update type but it also has an array umm with all with every emails right on the other list and the material like it has a location on the gallery, the version, the pilot version, etcetera.  
And they build the media ID, so everything that they can use to validate also backtrack, right?  
If there is an issue with the image, they cannot exactly what but image is.  
It is right, they don't have to ask.  
That's what that anymore.  
The other aspect for reason is that so there there is a contract with HCC team or or the one that owns SMF, right?  
So to publish any any abuse image to Azure for example right like your marketplace or Azure stack marketplace or the one pager leading it has to have some some because a condition right it has to have.  
You have to be a fixed size block because you you cannot be unblocked Lori.  
Initially it has to be a a phase lock fix size visist yeah.  
And he has to be able to Azure right?  
So what I'm saying is that they they missed are for, for, for media.  
Are you still?  
Please native services.  
Yeah, which is very is basically a storage solution, right?  
Yeah.  
So they they can deduplicate images and be more efficient.  
Yeah, but we we cannot grab images from outward services and just publish to the marketplace, right?  
Because they have to be azurified.  
Remember I should file means they have to be uploaded to Azure as fixed size.  
Uh, please flow, right?  
So let me take one example.  
So so they they BLOB that they store network services is for some like 10 G's.  
This one is not right.  
It's 10 GB.  
It's it's dynamic size, so you can grab the image and and create images in Hyper-V yeah, you can do that for sure, but you cannot you grab that BLOB and upload washer and then make it work right?  
It has to be fixed size and and also be applied as space block.  
Yeah.  
So there was a contract with HE like, you know, they say, OK, we we we can support umm making this process for you.  
We can upload to Azure for you.  
Make it fixed size and also upload a space block.  
Yeah, but you have to pay for the Taurus.  
Yeah.  
So if you just like to, we said, OK, we're gonna pay for the storage as as, as long as you keep pushing the the business there.  
But then we have to clean up right as we're paying for the storage, not only for for the reasons, but also for the quality.  
So we try to clean up resources as as soon as they are not needed.  
Yeah, but and also based on some silly, right?  
So basically there's a list like this in emissaries.  
Live has to be get for six months.  
An enemy that is not live like any release churn right for this later or so, or any candidate that is not the final churn can only be kept for one month and then delete it, yeah.  
So we permitted this resource in upload you cap and also I can service to actually perform the pression.  
So what?  
What it does is very very simple.  
It Lisa checks because any inventory right of the social in nature and then checks against their legal comments, right?  
Well, like what images were live or not.  
So to determine they pull their attention policy right?  
So so for busy right part, we clean out the storage account and also there's a compatibility on only the one we control, the PME one.  
We don't clean up the 1P LSD.  
So these ones weekly, I think every Monday around noon or so.  
So we we run the workflow.  
Oops, there is no slight because I have the lens explorer. Yeah.  
And you'll see.  
So what we need to be to be and me, I think the.  
Yeah, the there were some key key lookups.  
So they they work for us running into places.  
So we we'll try once the to make sure it works fine be me, but other than that you see the frequencies weekly, right?  
This will be a Monday 7/3.  
Seven 17724.  
Yeah, so every Monday and this is the how many GB's right with it.  
So I I think uh.  
67,000 degrees is around around more or less 6067 or 60 something terabytes, right?  
So in mind this is the volume that HCC is pushing the service account for us to publish, right?  
And then clean up when when it's not used to it or it's not needed anymore and the number of lobs we delete every week is close to 1000, right?  
It's like, you know, it's 751 that's that's Monday.  
Yeah.  
So today, in a few hours, we'll see a new run for this. Yeah.  
So this is the storage I I think they see clean up is not logging to PME.  
What you know, maybe we need to the bucket.  
Pull back to this.  
But we also did not the images from the gallery right?  
Because they also cost some money.  
So they looks can show you all the like the aggregated data, but also specific blog that you can see exactly what blobs were deleted and and I think we can show the stories name, yeah.  
You can also see because there are these today.  
Basically one starts for this product including active branch or Hot patch products.  
So we have to go through all of them and and and see what images can be deleted, right every week.  
This is the the other aspect I of is you have to clean up.  
We also clean up, I think, for containers.  
Yeah, but those are in a different lock.  
Different dose.  
Also, these only four beauties, OK, that makes sense.  
Yeah.  
So this is it any, any any questions regarding anything for this is that I can help wait.  
Yeah, feel, feel, feel free to jump, yeah.  
Yeah.  
Also, uh, you just let me mention.  
So when you want to lens.  
But I I think you can find the the workspace.  
So the worst place will say like uh WSD, sorry, UPS Media publishing, you click on the Workspace II or you can just type in his for UPS and you'll see the dashboard.  
So when you click on the on the, that's what it says.  
You'll you'll have to.  
The place I'm showing you.  
Right.  
So you can see all these sections like request marketplace stack.  
So we try to put the logs right separated by section like.  
This is the all the media requests they would receive, and whether they succeeded in automation or or something occur.  
Right.  
Uh.  
Also, there's this kind of operation doesn't say compete yet.  
I'm I I need to improve the locks on this.  
I think we improve the locks we need to complete the query.  
Yeah.  
So but in in any way, most of the operations are here from end to end, right?  
You can see the completion time for all the like creating just police all the work for the support should be in here, right?  
And there are a few others like there's like a cancel workflow for for ISO, for example, then now the marketplace please shows you the the presenter locks.  
So for how long he took for a for an offer right to to be live, he took around one hour plus one hour and a half for this offer, cause you can you can see some graph will see also in this space how how long it took for an offer to start publishing.  
Yeah, because for easier remember when we have a a churn there is publishing and then we have another term for the same offer.  
We have to wait right for the first one to complete.  
Today we don't cancel, so we wait because we don't know what, what will, what the performance will be.  
So this is how long I was sit there.  
Umm, waiting to be police, right?  
He he maybe a short time or maybe a few hour.  
The prevention is still polishing.  
It may actually sit there for I don't know if you hours.  
Yeah.  
So the advantage of this is that we we can publish anything, right with the have to wait for very long.  
As soon as the free cycle will start publishing, we had to wait for the last minute when all the chunks arrive, right?  
This this is our sort of the OR the early staging solution.  
Hmm, The thing is, of course, that marketplace is still has a limitation.  
Once an operation starts, we can we don't, we cannot start another one in parallel, right?  
Either we cancel or we wait for the.  
We just want to complete that, but anyway, I at least we're more efficient now.  
I think she's since last year or so.  
Yeah.  
Umm you can see also the stack marketplace logs that all the misses we publish that should be listed here.  
I I think this is restricted to seven days or so.  
This is why we don't have I love looks like.  
But you can you can go here.  
Alter the query and see more you can see OK days run.  
Like I would choose.  
Yeah.  
And then apply if you want to give it, you can say, right?  
You can save it and and just move forward right?  
So look, look at these.  
These these are all the images that we have published for stack marketplace recently.  
UH-1.  
You can.  
Yeah, maybe for one big lady, we trigger a pipeline, we go back to the pipeline.  
So these these will artifacts pipeline has an an idea right?  
So you can see.  
Oh no, no data.  
Anyway, you can see the pipelines that will start and then they releases that we also three right on the right on the last step because italis information right in this logs for isotopically.  
This is the test base salary.  
Not not the one.  
This is a test base one, so you can see the see similar to the request.  
You can see the request and also the notification part or the image that we have publish and the versions for somebody.  
This is a A put to this.  
I mean image definition.  
I know.  
Yeah, my boy.  
Yeah.  
Thanks.  
Sorry, what else?  
Dependencies, yeah.  
Uh, and the clean up, right?  
Yeah, yeah, this is.  
This is the presentation more or less.  
Yeah, yeah.  
Feel free to look into the logs.  
Yeah.  
Or look into the code and then you can ask more questions.  
Umm Yep.  
OK.  
Any any other thing?  
Any other questions, just feel free to jump in.  
But you know, we can we can ask offline with that also, OK.  
What it look?

 **Dinesh Vijayakumar** 42:34  
You know, I think I'm good at this point.  
Yeah, you can come back on questions.

 **Leopoldo Estrada Vargas** 42:36  
Umm.

 **Dinesh Vijayakumar** 42:38  
Thanks.  
Polo professional that pointers, yeah.

 **Leopoldo Estrada Vargas** 42:39  
Yeah, things initial look.

 **Dinesh Vijayakumar** 42:45  
Uh, yeah.  
I mean, we can go over for the AWE session on uh tavishi uh.  
Do you want to start?

 **Tavishi Gupta** 43:00  
Start.  
Hey, let me share my screen.  
So to just start with, sorry.  
To just start with a A, a a brief overview of what we do from end to end pipeline, we have packaging that creates the the pack after the build is completed on LCU SSU.  
And safe OS critical. Do you?  
Uh, and the the feature updates they they give they they create the packages, send us a Pvt complete on that Pvt complete.  
We then send AWE.  
Sends the to HCC.  
Ohh trigger basically to start creating the media and then in that and then you it's you see uses the payload from AWE and the images that they need to create do then to then create the media they sign off on this what we called an intent one pub then Vista and then we start orchestrating publishing to one pub and one pub then publishes it to different audiences like you can see which I think polo Jacob.  
Here and then Kruti also for ISO VLSC ISO explain already.  
So this is basically ISO where she's ESD and APPF and pass native.  
So from the AWE side, this is an high level architecture diagram in which what we do is like 8 of like media orchestration listens to two signals, it listens to BVD complete notification and it also listened to a cat complete.  
Why do we also listen to cat complete?  
Because it could be that cat could be rejected or like a failure could be rejected, or cat could be cat.  
Cat could go through a failure, in which case we have to send a media intent again to SMF so that they can with a new with the new payload so they can they can get the corrected information.  
It's important that we have to signals one is BVT.  
One is CAG.  
Most of the time, CAG is 90% of the time the cat doesn't like.  
It doesn't go, it's not cancelled or doesn't go through a failure.  
So that means we OBT and CAC has the same payload a lot of time, yes, but on SMF side they have a duplication logic.  
That means if the payload is the same then the then the then they don't waste their resources to create new media for it.  
So they're just like disregarded?  
Umm, so on on these signals we we we created intent the way we created intent is we read we from our tables we read we read the packages but from we also need to send SMF and image list or media list of what they need to be created.  
What needs to be created so we actually there is a get API for the list of media images that we that we ask one pub to send us so we can we we get that image list ohm here and then we after after the after we have created a JSON basically the the package information and the media list.  
Please send media creation kickoff to SMF.  
Then SMF does its creation.  
It umm it it after the image creation is done it uploads it to Azure BLOB storage and artifact service and the and then we and then SMF signs off on the media creation that which means that they do they have done some validation on the creation and then this think that the media of their creator is good.  
After that they sign sign off Ohm which is the signal for AWE to start place which is like basically orchestrates publishing with one pub and one pub reads and then this part.  
I think we're already explained one that reads it from the artifact, clouds the images, and then they publishes to different, uh, different endpoints here.  
So we already like I think most of us are like already aware of this.  
Some like these, a lot of these media types, there is ohm, there is a ISO.  
Uh, which is only which only goes through VLSC or like TV.  
Umm, we are CVS, CVS right now through UPS uh to OEM channels we were we were actually what we were doing was we were creating the media but we were not publishing it through umm to one pub ESRP base with one with publishing to ISO.  
Basically means ESRP submissions.  
It doesn't go to an endpoint.  
Umm, but we have a deliverable this time which Eduardo is going to help with in which she's going to enable OEM channels as well as VLSC and VSS channels for ESD.  
We, uh, we publish it to the to decad for VHD.  
We publish it to.  
To Azure marketplace, the one pub one big gallery like Toll already explained.  
There's also, like VHDX is also MTP.  
There are we MTP media which are VHDX but they are not published through UPS.  
We just media, we use media creation for them.  
We create media and then and then Etsy actually uploads this images and media seeker and then MTP folks can like ohm download the images from them.  
Containers are goes to Microsoft Container Compository and this is and then VHD goes through a marketplace stack APPF.  
Umm.  
And then this is the data like the way the data flows through our system.  
So on.  
Then PRT UM the but PRD.  
Basically, uh.  
When they have created the package, they signal AWE through BVT complete notification, which basically gives us the which the release ticket ID that created by AWE.  
But like it it it has the package version and package job ID that is the most important and for me piece of information in the BVT we actually store this information in a play runtime.  
Umm for non media please we create like uh a place for it.  
We put it in play runtime but and then also this triggers an additional like this.  
It triggers the media orchestration as well, which passes on the release product and trigger type is basically whether it was automated or manual.  
This is an automatic trigger type, so we we pass that release and product information here.  
Then for that release and product in the, this is the media orchestrator, right?  
This all of this is this is media orchestrator umm for that release and product we we we query a play runtime we get the package and the package job ID and the update and then for for the for the update type or for the payload that media needs to or media team needs to create the media part.  
Then we calculate the caller, calculate the payload based on this information that we have stored in our plate on time.  
There are two ways to calculate this information.  
One is through, one is in release, which means which means that we for for example for A9B we have, for example, LLC's are mostly like always like we always do release else use which means like they are almost like we would mostly we we will never look at like LKG from LCU or else use.  
So we look at like which basically we query a play runtime, see like what a A what payload and package version we need.  
We we are currently publish it and we we we added to our media intent list and then Ltd is basic it's so not so As for example SSU and some feature updates they don't churn each release.  
So we we we look at the last known good which is basically what went live ohm it in a previous release. Umm.  
And then we we basically use the, we use the package version and job ID uh for that ohm and then through through one pub we we get them the media list which is basically we get the media name and media architecture we before sending it out we store we store this information in media orchestrator runtime.  
So we have a log or like auditing of all of the intents that we send.  
We store the intent.  
We stored the intent details which is like basically package.  
So all of the packages that are associated with that intent, the media list, which is basically what media needs to be created, that's we get this from one pub and statuses, which is basically if we have just sent them them, just send them the intent then we we say request sent when when it's SMF receives it we change that to a request acknowledged and then the sign off we changed the status to request completed so we keep on we keep track of where or how the what status the media creation or like media creation and completion processes process is in umm.  
And then ohh, and then after this after we have logged it in the Media orchestrator runtime we we basically umm we we also send it then we send it to SMF, we send the what we sent to SMF is the request ID, the release product, the package package list with package.  
All of the package information and then the media then need to create umm this is?  
Yeah.  
Then this is a little bit like this is then on the on the ingestion publishing approval.

 **Aravind Siddoju** left the meeting

 **Tavishi Gupta** 55:22  
Go live ohm.  
The data flow on on the.  
Umm.  
On the on on the right hand side, we can call it the right hand side because this is after the media creation has done validated and then now we have to publish it.  
So what we do is the it's STC like SMF team, they sign off on an intent.

 **Alexey Loginov** left the meeting

 **Venkatnagaraju Goursetti** left the meeting

 **Tavishi Gupta** 55:50  
So what?  
That what I mean by that is there is a there is a request ID, there's an intent ID associated with each intent they sign off on and on that intent there is a uh and uh.  
And this is how we identify.  
This is a unique ID and this is how we identify ohm.  
The intent are uniquely so.  
When they sign off, this is again and this.  
This is all media orchestrator.  
Then when this sign off, media orchestration media Orchestrator listens to that sign off and then what?  
The media orchestration does, or media media Orchestrator does is it looks so in the previous slide.  
Like I mentioned it be we we store all of these internal information in the media orchestrator runtime.  
So when when SMF has signed off, we read that intent using the request ID release and product.  
If if we find the the find the intent on it which changed the status for it to complete it because now we have gotten a completed notification and we then start the publishing.  
But we also one one thing we also check is the orchestrator config.  
Sometimes, for example, we don't have to start.  
Uh.  
Ohh it now it is not the case because we have we have.  
Play we have release ticket for everything, but it used to be that MTP we do not have a release ticket or I.

 **Achal Shah** left the meeting

 **Tavishi Gupta** 57:40  
I see Dinesh is Dinesh.  
They should have a higher hand up.

 **Dinesh Vijayakumar** 57:46  
Yeah, I mean, I had a question and another thing was the third time check, so just wanted to see if others have any other meetings that we can continue in the next week.

 **Tavishi Gupta** 57:57  
Sounds good.  
Yeah, we can definitely continue this for the next next week, but yeah.

 **Dinesh Vijayakumar** 57:58  
Yeah.

 **Alexey Loginov** joined the meeting

 **Dinesh Vijayakumar** 58:01  
Yeah, I think, yeah, that'll be good.  
The other question is, I mean in the previous slide and the other slide, so you mentioned that, uh from AWE we calculate payload etcetera, etcetera, right?  
So how how much of this only is media specific?  
Are this is common function right that is used by other our publishing also right?  
Like uh, you'll be publishing on other teams as well.  
So I mean, how tight and how, uh, how tight these things are and how much is a media independent workload in this right data being.

 **Tavishi Gupta** 58:24  
I think this is all.  
So this is umm this this part is all like this is.  
I mean the way you're connected is that we read this from non media, right like this.  
This, like the play play right?  
Like this play runtime is shared across, so we share resources, right?  
Like all of this like this, this this is a very important component.  
Like release Ticket is a very important component here.  
BVT completes is like what like goes and feeds into the media orchestrator.  
But this logic over here we this calculate payload we do use like uh, I mean the LG API is shared right?  
Like there is.  
That's that's but umm, I mean I I think other publishing also uses that API but this this this this logic off like ohm, uh.  
What do you say combining the media list and like basically it's looking at in release and then if in release is not present then looking at LKG that's that's this this this is very media specific and then just storing it in the media orchestrator, runtime media orchestrator runtime and media orchestrator config.  
If you are here, I actually kinda forgot to say that this is how we say which intents are umm, which which intent interns are enabled and which product for what product.  
This is this is also an important concept, but this these two tables we orchestrated runtime.  
This is a runtime table and this is a static table media orchestrator config.  
Our our solely media specific, not other publishing doesn't or non media doesn't use that.  
I'm not sure if I was able to answer your question, Dinesh.

 **Dinesh Vijayakumar** 1:00:25  
Yeah, yeah.  
I mean, I got it right.  
I mean, so few of these are common ones and if there are any errors or issues with that, that would affect our publishing versus the same UPS that used by media workflows also and only the the config is the static table for media or flows only.  
And the runtime runtime is for media, right?

 **Tavishi Gupta** 1:00:46  
That's correct, yeah.

 **Dinesh Vijayakumar** 1:00:52  
Thanks tavishi.  
Sari, I thought it would be good to continue next week because I think folks have already spent an hour and it would be a good to have uh, the session continued next week with the with the focus on the workflows and then people can come back with questions there.

 **Tavishi Gupta** 1:01:11  
Thank you so much folks.  
Thank you.  
Thank and thanks for love for sharing VHD's lights.  
Thanks a lot.  
Have a good day.

 **Dinesh Vijayakumar** stopped transcription