Harold Byun:

Hi, good morning. Good afternoon. Good evening. Welcome to today's webinar on Controlling Data Leakage from Cloud Storage. My name is Harold Byun. We're going to start in just a moment here. Let's give another minute or two for folks to join in. So please be patient. If there's any questions that you may have just as a order of logistics, you can use the chat window as part of your viewing screen, and feel free to ask questions along the way. We will do a Q&A. There are also some attachments and some links. All of you will get a copy of the presentation as well after the presentation. Thanks. (silence).

All right. Well, good day, I guess or good evening. Again, welcome. We're going to kick things off here, so I apologize for the slight delay. And today's webinar is on Controlling Data Leakage from Cloud Storage. Really what we're going to be covering today, as indicated in kind of, I think the reason why you all registered, is an overview of existing controls that are available within cloud storage environments, what we would consider some key gaps in terms of some of that model, some of the key differences between, I guess, object encryption, or file encryption or blob encryption, dependent on your terminology, versus things like the identification and tokenization, some of the architectural models that we're seeing supporting a de-identified data pipeline for folks that are looking to basically import more data into a cloud-based analytics environment, and then we'll show you kind of a live demo of de-identification process, how we believe that process can be streamlined, as well as offer enhanced data security.

And Q&A throughout, I mean, we'll obviously have a dedicated Q&A portion at the end of the session, but feel free to use the chat window throughout. I do check that periodically. And you can always email us and post at info@baffle.io or Harold is my email, @baffle.io. I'm not the fastest on email, but feel free to reach out and contact us if you have any more in-depth questions. So without further ado, well, a little background on me, I've been doing data security or security in general for over 25 years at this point. I started out as a kind of hands on keyboard, network and sysadmin and kind of graduated into security architecture, and that evolved into a number of roles and data leak prevention or data loss prevention and overall data containment. So it's been the focus of the bulk of my career in Data Security and Information Security Management.

Today, what we're going to cover first is an overview of existing controls and some key gaps with Cloud Storage. And so before we jump into some of the details, why is this probably of interest to many of you? I mean, one, I think that there's obviously a significant migration that we are seeing across a number of enterprise customers to leverage cloud-based data lakes for either data warehousing or analytics, and obviously, everybody apparently is doing AI. AI apparently is going to rule the world whether you believe that or not. The reality is that data is moving in bulk. That may contain sensitive data, and that's being used for a number of different AI and ML scenarios. And even within the analytics context of what's going on in today's world with COVID, and contact tracing, and everything else, this obviously becomes a sensitive area as it relates to data privacy.

Part of this challenge is also that multiple sources of data or we'll call multiple producers of data, are basically pulling this information from a largely distributed data set and piping it into this massive data lake that people are effectively establishing and which cloud infrastructure by and large, lets people leverage very easily. And then thirdly, kind of this continued, I guess, data exposure within some of these cloud environments, and there's a multitude of reasons for that. A lot of people will point to misconfiguration, or lack security policies or poor automated checks. There's a number of reasons for it, but the net net is that data continues to be exposed in the cloud storage environments inadvertently in most cases. But it's obviously presenting a problem, and you can just look at any of the recent headlines around this. There are multiple major companies that have been impacted by this as well as over a billion records exposed, and some of the cloud storage data leaks that have occurred.

So when we look at kind of the migration, there's a cover juxtapose that's with AWS and Azure, the same principles apply for GCP, and I could have just spent more time putting together slides. But effectively, on the left-hand side of the screen, you have your modern enterprise, more or less with a set of applications and data stores, and it could include other distributed data formats, whether that be from IoT, or other service devices, or other devices that are collecting information, as well as third party suppliers of information, and that's the weird three pronged icon in the lower right-hand corner.

And effectively, what's occurring in these types of environments is we're seeing a pipeline of data going into the cloud data lakes. And in the case of AWS, they have an S3 established data lake, which can effectively be the back end data source for a multitude of analytics sources as indicated in the slide, whether that be Redshift or something access through Athena, or a third party data source, whether that be Snowflake or another data warehousing solution, or it could be something like EMR. It could also be Kinesis data streaming piping data into that S3 bucket. But the net net is that you end up with a ton of information in this semi-structured object format within S3 that multiple downstream analytics solutions are accessing.

And similarly, within the context of Azure, Azure has Azure Data Lake services, and they also have Azure Blob Storage, which sits underneath. And in effect, the same process is going on where people are piping in data to Azure Blob Storage, or what they're calling the Data Lake Gen2, which is a slight variant of the Azure Data Lake services, and a combination writing on top of Azure Blob Storage, rather. And so the net net is that you have data that is hitting storage in the cloud infrastructure that is protected in a variety of different ways. Some of the methods of controlling that are what we're going to review very quickly, but when we look at kind of the challenges within good exposure, obviously, data breaches are occurring every other day. I think you're all familiar with that.

Some interesting notions, as you start gathering more information and sharing it in a distributed format, obviously, data exposure tends to increase. And based on some recent surveys, roughly 60% of organizations have had a data leak via a third party. And as you look to aggregate data across parties and multiple producers, this continues to be a data management problem as more broadly affecting organizations. And then thirdly, really, that these cloud storage data leaks continue, again, over a billion records leaked at this point, and there's an industry estimate of roughly 11% of cloud storage left open to the public. So there's a variety of different scenarios where that's occurred, obviously, and there's a lot of defaults that can help control that. But we're going to cover some of those controls and go a little bit beyond that into de-identification in the coming presentation.

This is a bit of an eye chart, I hope that it's legible for most of you. The slides will be available for you, so I'm not going to cover everything in detail. But what I tried to do was basically juxtapose some of the controls between two of the major infrastructure providers, AWS and Azure here. So aside from blocking public access, and removing anonymous access, which would seem to be obvious steps for anybody looking to implement security, there are a number of different types of controls, whether that be using Azure AD integration, there's this notion of one time Secure Access Signatures. Well, they're not really one time, but they're a unique URL that is generated that has resource protections that can be used to control access to a given storage container and Azure. Similarly, I guess a similar construct, it's not really one time would be an IAM Role within AWS that provides a role-based overlay of access from instances to S3 storage.

So there are a number of controls, obviously, TLS or HTTPS access as it's fairly standard in industry today. And then there's a range of different monitoring and logging capabilities that are available, whether that be some type of policy-based discovery. There's a content-based discovery called Macie that's available on AWS to scan your environments. Microsoft has advanced threat protection, which has kind of an intelligence offering looking at access to Azure Blob Storage. And these are always to obviously detect potential inadvertent or unauthorized accesses to your storage. And then there's a range of encryption at-rest capabilities that are available, and we basically break down into provider Managed Keys or what Amazon dubs SSE, or server-side encryption. Microsoft calls them Microsoft Managed Keys. Then there's kind of customer controlled keys within a cloud provider key instance. And so those are variants of some of the server-side encryption. And then last but not least, there's kind of Client-Side Encryption or customer provided keys, which are encrypting data as it goes into a cloud infrastructure environment.

So those are kind of ranges of different methods that can be used, and they're all viable controls. And so there's a ton of, I guess, tools within your arsenal to help reduce risk and control security gaps, or accidental data leaks, or accidental data accesses to data that's being stored within a cloud environment. I think that the challenge that we continue to find, and something that I think a lot of customers are starting to realize more and more is that inevitably, things are going to be misconfigured. That's why solutions exist to automatically check config controls to ensure that those are in place appropriately. But people are going to continue to struggle in terms of configuring the data, or configuring and appropriately protecting the data, rather.

And so when we look at the Object, Container, Bucket, Blob Model here, this is obviously simplified. But what I've tried to represent here is a notion of where your data is protected and how its protected. And so if we look at an S3 bucket on the left, or Azure Blob storage container on the right, which is the blue box, inside that, we are depositing data in the form of objects or blobs in the case of Azure. And so effectively, it's the same thing. It's a piece of unstructured or semi-structured data or a file or effectively an object, and that object is encrypted. It can be enabled to be encrypted by default. In Azure, it is encrypted by default, and you can enable this. And as soon as the object is basically uploaded to the environment, it is encrypted using some of the encryption mechanisms that we've just covered in the previous slide.

Now, the id of the key question here is what happens when you access an object or blob that's been encrypted from storage? And so, if you think about this for a second, a lot of people immediately assume because I'm encrypting data inside a container, or because I'm encrypting data at-rest, the data is effectively encrypted and protected. It's a bit of a misnomer in the security industry. And over the past several years, we've talked to hundreds and probably close to thousands of security practitioners, and there's a fair amount of confusion on this, which is why we spent some time on it. For some of you, it may be second nature. But what happens when you access an object or blob from storage is effectively you get clear text because the object is basically decrypted on the fly for anything accessing that piece of data.

The object encryption that's in place to protect it is really designed for theft of a disk. It's really designed for protecting against somebody breaking into a data center at Microsoft, or AWS, and stealing disks out of the data center. It doesn't really do anything to protect anybody who gets access to that container, or that bucket, or that given object, or blob. And so the challenge that you have here is that people have implemented methods to encrypt things at the object or blob level, but it does nothing to actually protect the data that's living inside that object or blob. And so that's a fundamental gap in the de-identification or data protection process for piping data into the cloud.

And so if we look at this, from a object level encryption, to simplify this, we just call the thing on the left a container with objects, and we encrypt the objects, and we compare that to what we would call data centric encryption, or de-identification, which is encrypting the data inside of an object. The fundamental difference is that when you pull that data out of the container, you're getting clear text if you're using object level encryption. Whereas if you're using a data centric protection mechanism, you're going to get encrypted data. And so that is a fundamental difference. And so if you subscribe to the notion of that, misconfigurations are going to occur, and if you subscribe to the notion that your network will be penetrated at some form, or you operate from an assumed breach posture, or the over-marketed zero trust model, then you have to logically agree that a network or an attacker is going to get inside your network, whether that be cloud based or on premise, and will ultimately get access to an object store.

And so if you're using object level encryption, when that occurs, and there's a lateral move inside the network, then they're going to get clear text data versus encrypted data that's protected at the data level, per se. So that's kind of the fundamental difference. This is kind of a view juxtaposing that. So in the top half, if we look at the screen, this is representative of what we call data centric encryption or encrypted data. And that would be an example of what an attacker might get if they were to get a de-identified data set that's encrypted at the data level. And then below image is an extract of data that is effectively clear text data that would be piped out of any kind of semi-structured object living in one of those cloud containers. Again, it does provide you some mitigation and perhaps some compliance capabilities, but it really does nothing to protect data at the level that most people, I think, in the security industry would want to ensure.

When we look at some of the key benefits of data centric protection, whether that be de-identification or tokenization, you're really protecting the data inside objects and files versus the actual container or the object itself. It grants you safe harbor from accidental data leaks, from key privacy and compliance regulations, which is obviously important in today's world. And ultimately, it can help accelerate any type of cloud-based analytics program that your business may be running, while overcoming a lot of the security and privacy objections that typically occur in terms of adopting cloud infrastructure. So hopefully some of those benefits are evident to you as well.

So when we look at some of the architectural models for de-identified pipeline, this is kind of an overly simplistic view of somebody migrating data from an on-premise data store on the left, moving it to the cloud and hitting some type of container, whether that be an S3 bucket or Azure Blob Storage. And typically, this then evolves into what we call the Modern Cloud Data Lake, and that data lake is then being used as a data source for feeding downstream analytics solutions, whether that be a big data environment, HDFS or something through AWS, EMR, or Redshift, or Snowflake computing. So a lot of people are moving to solutions like this. And in Azure, it's the same thing. It's just being consumed by Azure Data Lake services or Azure Synapse, from an analytics perspective.

There's a more detailed view of what that might look like in terms of some of that pipeline. And so we have, again, an on-premise data store or producer of data that's being migrated, in this particular example, it's using AWS Database Migration Services. This Baffle Shield is an encrypt, decrypt proxy that we actually supply as a product offering. And so that's in here to actually encrypt the data on the fly as it hits the S3 bucket where it can then be exposed via catalog, in this case, via AWS Glue, and consumed by multiple downstream analytics services, again, whether that be Redshift, an EMR, or just exposed via Athena. And so this is kind of a common pattern of data pipelining that we're seeing across a multitude of different organizations and a variety of different industries. Everybody wants to get a better handle of their data on their data to gather business insights and intelligence to ultimately monetize that information or identify new services and opportunities for the business.

And so when we look at how this might play out, what we're going to do, switching gears into this live demo, is basically a portion of this, there's basically going to run through a live Database Migration Services, implementation type data into an empty S3 bucket, and then pull it into the Athena console, as well as running some kind of de-identified quick analytics. And this is going to be very rudimentary demo. I don't want anybody to think that this is going to be anything glamorous. And then there's a subset of this where we'll pipe data into a separate S3 bucket and render it using an application to kind of show you the live nature of how data can be manipulated. So that's what we're going to cover. And ideally, I don't know, five minutes or less or so and then we'll obviously open up for any questions as we go along.

Here is where we're going to start. So in this particular case, here, I'm going to start with this empty S3 bucket which doesn't have any data in it. What I'm also going to... It might help if I start sharing my screen, so I'm going to go ahead and do that. Apologies. Okay. So I think you can all see the screen right here. You can see that this S3 bucket is empty. And what I'm going to do here, in this particular instance, I have this data store called HB2 Store, which is my initials, and so it has some potentially sensitive data in this data store. So customer names are potentially sensitive PII. And we want to run this into the cloud for some analytics. And so what I'm going to do here is I have this, again, S3 bucket which is empty, and if I go to Amazon DMS, I've been doing this particular scenario, it's actually restart this task, which is basically sourcing from this database server, so the SQL server which I was just on, and it's going through to this S3 target that I've established as an endpoint. So let me kind of...

See here, you'll see it's starting. This isn't the fastest process in the world just because of the way the process kicks off, but it's going to source from that database that I just showed you where the data was living in HB2 store, and that will be piping over to this S3 bucket that I was showing you that's empty. So just give that another moment or so. Hopefully, it won't take too long. And you'll see that it's running now. And it should be completed in just a couple seconds. Again, if you have questions, please don't hesitate to chat them in while we're waiting to watch the grass grow here, but it should just be a couple more minutes, a couple more moments, rather. Okay, it's completed.

So if I go back here and I hit refresh, you'll see that we've now established this folder, which is the hierarchy of the database and tables from the source. And within this, we actually have this file that's been actually brought over. And if I download this file, just to quickly view it, you can see that opening that file shows you that we have effectively de-identified some key fields in this particular environment. So customer name, customer ID, and I think we chose city, and effectively, this is done via data privacy policy. And so this is the file as it's living within S3 at the object level, but the data inside from the source has been basically de-identified on the fly as we move it to S3.

And so if I go over to something like Athena, I already set up these glue catalogs just because I didn't want to take time to do that. But effectively, what we've done here is we've set up a database with a target of store, and the store is referencing the that CSV repository. And so if I run a query on the store, you can see that, again, we get city, and customer ID, and customer name, and a de-identified format, but it does expose it to the sub-Athena interface, which can then be used for things like analytics. And so if I wanted to do something where I wanted to search on a given customer name, and I've established a search on the de-identified token... Let's see if that runs. What we're able to do is basically pull up a deterministic token for the customer identifier, and we can get catalog information or other relevant information around this particular individual or entity. And so this is one way that you can quickly expose source data to a broader pool of analytics solutions that may be available in using this cloud-based footprint, but doing so well, again, de-identifying the data so that it's protected, and you're not violating any privacy or compliance regulations.

This is a quick view of one of the capabilities. The other capability that I wanted to show you is a live rendering of some of that data. And so you'll see here, again, I have this empty bucket in this particular example. And so what I'm going to do here is I'm actually going to go over here. Sorry, And you'll see that I have some API calls that I created here just to basically do S3 API calls. And so what I do here is I'm also going to remove this file, flights-local.csv. So you'll see here that I've got this file, flights.csv, and what I'm going to do is I'm going to upload that to S3 using an API call. And it's going to hit that Baffle Shield encrypted on the fly, and then we'll be able to render that in a web application. And so you'll see here that I have this flight information, which is a bunch of dates and destinations and delay information in this file. And to this point, within this web application that we have, you'll see that there is no data available.

So if I click on mornings and nights, basically, this is an empty set going through this Baffle Shield. And similarly, this application or this web interface is going to a different destination where we're going to show the encrypted version of this data. But again, right now, there's no data here, and once again, there's no data in this S3 bucket, right? So it's all empty. And so let's kind of encrypt that data on the fly. And so the way I'm going to do that is get some API put, it's what I did. So if I do this, we're going to actually use the S3 APIs and put the flights CSV object into S3. And so that's completed. And so you'll see that if I go back to S3, we now have flight CSV. And if I open flights CSV directly, again, accessing that object from within cloud storage, bringing this up just a little bit slow in this number's application, should reveal that that is an encrypted data set.

Apologies for taking so long here. This is, I think, why people use Excel. But at any rate, this will open up and we should see that, at least in this case, destination has been encrypted. And so this is one set of data that's been encrypted. And then if I go back to our web applications, so I go back to our web application here and hit refresh, you can see that we're now rendering some of the data. And if I go into January, we see that this is encrypted because this particular view is not going through the Baffle Shield. Again, I can toggle through that data, and look at different parameters here where we're rendering that data. But one of the fields in this case was deemed sensitive, and so it's not being decrypted. Conversely, if I refresh this, we'll also render the data, but we can see the full data set in terms of the destination field. And then if I wanted to, I could, again, toggle on some of that data to get a different view.

So this just gives you a sense of the transparency that is offered as part of the data pipeline process where, again, without any code modifications, we're simply moving data from point A to point B enabling the de-identification process to ensure data privacy, and then ensuring that that data is protected as it sits in the cloud. And if somebody were to get access to a given bucket or to a container, unless they're going through an authorized channel, they're not going to be able to see your data in the clear. And so it significantly reduces your attack surface and mitigate some of the risk around data privacy.

Now what I'm going to do as well is issue kind of a retrieval and this is going to retrieve that file into a flight local file. And so just to reiterate here, I don't have a flight's local file here, but if I issue this get function, it's actually a text error there, but we're retrieving this file. And then if I do an LA listing, we have this flight local file, and we basically have decrypted that file on the fly by going through that Baffle Shield. So that's one method of transparency and, again, to prove it that it is operating on a live data source. If I go back into this bucket, and I choose to delete this file, I can figure out how to do that, I will delete that file, and then go back to my web application, and if I refreshed, you can see I no longer have the data set available. So it is a seamless mechanism to, again, expose protected data to a variety of authorized subscribing sources, whether that be a data warehousing application, a data lake analytics tool, or some other method of presenting that information to your users or distributed data subscribers. So it's kind of a demo in a nutshell. So what I'm going to do here is then shift gears and switch back to C, enclosing and cover a couple topics here.

So in summary, implementing data-centric production can help reduce the risk and the attack surface, reduce the risk of data leakage and theft. There are simplified de-identification capabilities that can accelerate some of the business initiatives that you may have, while still ensuring compliance with data privacy regulations. And you should definitely consider some operational models that simplify or reduce friction for your Devops groups your business users and stakeholders. There's a lot of ways to kind of skin a cat. We've covered some of the existing configuration controls that are available to you today. But there's also a huge operational concern in terms of how do you deploy or operationalize a solution that protects data in a consistent manner without adversely impacting the business.

To some resources, and then we'll jump into some of the Q&A here. So some links here, baffle.io/dfp, which is data-centric file protection, and then slash privacy, there's a wealth of resources on the site. There's a ton of articles on how we can actually do this. We also have a blog on the AWS DMS site in terms of integration with Database Migration Services, and what we can do for RDS. This was, obviously, an example with an S3 bucket. But feel free to leverage those resource, and you can always reach us, again, via email. And if you're interested in a one-on-one, technical deep dive, we're happy to do that as well with some of our security architects.

So let me open it up for a couple of questions here. Again, feel free to add these. "In terms of what types of de-identification and tokenization methods do you support today?" So we support a range of different data protection techniques. So we have a data tokenization capability, which is what is known as a faultless capability. So with deterministic mode, that supports joins and foreign keys and consistent tokenization, which is what we demonstrated in that Athena console. We have a format preserving encryption capability, which is a length preserving capability, which can preserve a given format of a data type. So that could be something for credit cards, or emails, or birthday, or just preserving overall length. There is a completely randomized encryption capability set that we have. We also have a record level encryption capability, which can tie records to given entities to divide the records in a multi-tenant or shared data storage solution, and effectively segment the data in a shared data store. And then we also have a Advanced Encryption mode, which allows for operations on encrypted data or in-use processing encrypted data without ever decrypting the data values in the store in memory.

That is akin to a homomorphic capability, but we use a different cryptographic technique known as multi-party compute. And there's a ton of information on our website on multi-party compute and the differences between homomorphic encryption and some of the key use cases around privacy preserving analytics and secure data sharing. So hopefully that answers your question on the different encryption modes.

So another question, "Could you share more details about the crypto behind de-identification implemented?" Those were the different encryption modes. These are all basically various encryption modes that we can support via the Baffle Shield. And the way we do that, I guess, I probably should have included a slide as we basically create a key mapping to data. So actually, I can pull up some of this for you, and maybe this will help while we go through this. Let me see if I can share this really quickly just to show you this one more time. I apologize for shifting back here, but I think this will help. So what we let people do is basically specify an encryption policy. And so you can see here that this is our management console, which I did not show you earlier.

If I go in here, what we're basically doing is we create a mapping between an encryption key store or an HSM and effectively allow a user to create a mapping of keys to data. And so in this particular example, I've specified a given application, and if I wanted to encrypt the data or protect it, what we do is we basically expose a schema selector and so the schema selector... It's operating a little bit sluggishly today. But let's go in and basically pick a variety of different fields, and specify key mapping, as well as an encryption mode, as well as deterministic and non-deterministic. And so this is basically random or tokenized, in terms of the variants. I apologize for the clipping, it's because I've a tiny laptop. They won't give me a bigger one at work, and so we've had some scaling issues on this. But this effectively lets you define a Data Protection Policy at the field level, and then that's what's actually applied to data that's being migrated to the cloud.

So kind of a point and shoot operation in terms of being able to select what fields are important to you. And then we simply apply a cryptographic algorithm going back to your original question of what encryption algorithms do we actually support. And again, deterministic or randomized? We can do either. We could do format-preserving encryption as well with validation. So all of those things are included. We have a number of different pattern libraries. So again, if you look back in the slide deck earlier, there were variants of how we formatted email addresses, what we preserved. We also have a dynamic data masking capability as well that can also choose to mask the data as it's presented to a user. We have some other sessions on that as well.

"Is there any way to perform analytics on the encrypted data that is stored in the cloud? Or can analytics only be performed on clear text data?" As I mentioned earlier, we do have the Advanced Encryption mode, which uses the multi-party compute. That is a method that allows us to absolutely perform analytics on data that is encrypted. So it's a unique capability to our solution in addition to the format preserving encryption and the deterministic tokenization that we have for de-identification. There's a couple other webinars here on that, but the short answer, just in the interest of time, is yes, we can do it. We're a partner with Tableau. We've proven out with Microsoft Power BI in terms of off-the-shelf analytics applications that are running operations via dashboard on underlying encrypted data without ever decrypting the underlying data stores.

And there's an entire set of white papers and articles on what we call secure data sharing and privacy-reserving analytics, which is this notion of enabling multiple parties to submit data and share data in a common data store that is encrypted and remains encrypted, but still allow for frequency analytics or aggregate analytics to be performed on that data set without ever revealing the underlying private values. So hopefully, that answers your question on clear text. And we can do things like any mathematical operation, any ad hoc algorithm in terms of the operations on the encrypted data, as well as wildcard text searches on the underlying encrypted data.

"What databases are supported?" The databases that are supported are basically every RDS flavor on-premise or in-cloud. So SQL Server, Microsoft SQL Server, Azure SQL, [inaudible 00:43:24] Variant, Azure Synapse, Data Lake Services, Postgres, MySQL, MariaDB, RDS, Aurora, and then via this method of de-identification, the downstream services of synapse Redshift, and Snowflake, and any other type of data lake analytics can be supported via that model. And then in terms of the source databases from on-prem, a lot of that we're relying on the Database Migration Services to support. So we've had customers migrate from a DB2, as well as an Oracle from an on-prem to an in-cloud footprint. So hopefully that answers your question. We do also have Mongo and Cassandra on the roadmap. We've integrated with Spark and Databricks to a number of different alternatives in terms of database platform support.

One last question, "How are you integrating with different key management providers?" So we support industry standard protocols for key managers, and so that includes KMIP, which is common for key management solutions, and PKCS 11, which is the standard protocol for hardware security modules or HSMs. Azure Key Vault has an HSM mode, AWS has a cloud HSM mode, there's On-premise HSM, and the flavor of nCipher, SafeNet Luna, there's Vormetric DSM 1. There's a number of different HSM solutions out there, and so they all pretty much uses PKCS 11 protocol library. And so that's what we use interface with those. And then with cloud key managers or secret stores, we also have an integration with HashiCorp, and those tend to be rest-based in terms of integration with KMS, or an Azure Key Vault, or a HashiCorp, or Secrets Manager. So hopefully that answers that question.

So I hope that some of this information was useful for you and of interest... That was the Q&A, I apologize. And so you can reach us, again, at info@baffle.io. There's the resources that we posted. All of you will get a copy of the slide deck as well for your review. And if you need any additional information, we're here to help any way we can. So hopefully this was useful for you and enjoy the rest of your day. Thank you.