Celine:

Welcome and thank you for joining us today. There are several registrants still signing in. So we'll pause for the next minute or so to allow everyone to get settled. Thank you for your patience. (silence) Good afternoon, everyone and thank you for joining us for today's broadcast, Securing Data "IN" the Cloud. A couple of housekeeping items before we get started. Today's broadcast is being recorded and all attendees are in listen only mode. During this broadcast, we'll be taking questions which you can post using the question and answer module in the communication panel. Your questions will be answered during the Q&A session at the end. And if we do not get to your question, we will follow up with you by email. Following today's broadcast, an email with a link to the recording will be sent to you for future viewing and to share with your colleagues. You can also download our advanced data protection data sheet, AWS DMS Integration with Baffle Solution Brief, as well as requested demo under the attachments in the links section. And with that, I'd like to hand the mic over to our presenter, Harold Byun, VP of products at Baffle.

Harold Byun:

Thanks, [Celine 00:02:26]. Thank you all for taking the time to join us today. Obviously today, we're going to be talking about securing data in the cloud. A little bit opposite from the common view of security of the cloud. And so there's a lot of controversy, I guess, over whether or not people think the cloud is secure enough. I think that the prevailing wisdom is that many of the cloud operators and infrastructure providers can deliver superior security in many ways to on-premise deployments, but it still leaves an open question of how people are actually securing data in the cloud, and what tools and solutions are available to you.

Harold Byun:

So this is the agenda for today's session. We're going to cover common gaps in this shared responsibility model, some common conceptions about encryption, and then also cover some services and solutions that are available to you from the cloud providers, such as database migration services, cloud key management solutions, and also some capabilities that are unique around serverless PaaS implementations. So we'll be mixing this in with some live demonstrations as well so you can see how the tools are actually implemented.

Harold Byun:

Many of you are probably familiar with this. This is actually from Amazon Web Services. It's their view of the shared responsibility model where everything in orange on this slide is effectively the cloud provider, or the infrastructure provider's responsibility for security of the cloud. And the top half of this screen, or top half of this view, is actually the customer responsibility of security in the cloud. And there's obviously a difference there. And you can see the smaller text where AWS is responsible for protecting the infrastructure that runs all of the services. The customer's responsible for configuration of the cloud infrastructure and the services, as well as the data that resides in it.

Harold Byun:

And if you kind of go into, not a technical controls breakdown, but at least a controls breakdown, this is kind of the second level of detail of the shared responsibility model in terms of the types of controls. And you'll see that there are inherited controls that are listed at the top. And then this slide, the

inherited controls are controls which a customer fully inherits from AWS. And these are typically the physical and environmental controls.

Harold Byun:

There are shared controls which are partial or joint responsibility between the cloud provider, infrastructure provider, and the customer. And some examples that they give for this are things like patch management. Everybody has kind of heard the stories ad nauseum around configuration settings not being appropriately applied, or app holes that have been left open in some degree. And so these are things that are the responsibility of the customer, whereas the cloud provider or the infrastructure provider is merely making them available. And then the bottom tier is customer specific controls, that are obviously the responsibility of the customer as they utilize the service and deploy applications and leverage the infrastructure that's being provided to them.

Harold Byun:

But there's an interesting notion here. And there's one kind of explicit omission in this type of presentation of the types of controls. And it may not readily be apparent. But well, perhaps it is. And the thing that is really missing is ultimately a reference to data. And so there isn't really an explicit call out of how you're securing the data that lives in the cloud. There's a ton of information around configuration settings, configuration management, configuration drift. But there isn't a lot in terms of securing the actual data that you're putting into the cloud. And so that's kind of what we want to cover today.

Harold Byun:

Obviously, many of you are familiar with the ongoing ramp of privacy regulations that are being passed. I'm sure many of you have been GDPR-ed to death in terms of some of the noise in the market. But ultimately there are continuing, or an ongoing concern, around data privacy. We expect roughly 30 to 40 states in the US will pass their own legislation as well as individual countries continuing to pass their own legislation around privacy. And you always have the overarching GDPR umbrella. I think that the other, more probably significant, trend around this is that ultimately data privacy matters to the consumer and it matters to your customers. And when you look at the level of focus that people have around the vendor focus on privacy and using that as a differentiator, as well as some of the financial implications of this as well.

Harold Byun:

So in this next slide, you can see here that obviously this is a scenario where, actually, this is a story from yesterday, Facebook is being basically being held liable for billions of dollars for some of the privacy gaps that they've held. And so it's ultimately just something that's more prevalent in society today. And the expectations have largely shifted around how people are protecting data.

Harold Byun:

So when we look at kind of current encryption schemes and what's available, I guess the prevailing wisdom, or what people gravitate to in the market, are two prevailing solutions for protection of data. One is encryption at-rest, which many people are very familiar with. It generally is in a storage level protection for physical storage. And then there's transparent data encryption, which is a database container model for database level encryption. And the challenge with these is that they're very good for physical types of breaches, but that's not really how data is being exfiltrated or breached today in

the ongoing onslaught of breaches that we continue to read about. These really just protect the physical storage, or if the end point is lost, or if the server is compromised in some physical manner. We jokingly say here that that is protecting at the Tom Cruise threat model, where Tom Cruise breaks into the data center and drops in from the ceiling and steals your data. And that's just not how people are really stealing data today. And that's not how things are being breached.

Harold Byun:

And so when we look at common misconceptions, the bulk of the market still thinks that encryption at-rest is good enough, and the data is protected. But if you really look at it, any user or attacker that gets on the database or the machine gets full access to the data in the clear. That data is in the clear. It's clear in the memory, it's clear when it's in use. And it's a little bit of a misnomer for people to say... It still addresses a compliance checkbox in many ways, but ultimately it's not really protecting your data that well.

Harold Byun:

And same thing for transparent data encryption, or database level encryption. TDE is a container based encryption model. And the data is in the clear for any user or attacker on the database. Marriott is kind of the most recent poster child of this. They actually were utilizing transparent database encryption, and still had half a billion records breached. So that's just a common misconception. We still feel like 70% of the market, 80% of the market still thinks that this is adequate. Although we are seeing a lot of auditors even make the realization that field level encryption is definitely a requirement going forward.

Harold Byun:

So when we look at alternate approaches to this, there are a number of solutions that you can choose to leverage. But as it relates to database migration services, and we're seeing a huge trend to migrate to database platform as a service across the different cloud infrastructure providers, the data actually in the current standalone migration process lands in the cloud in the clear, and then is actually poured it into an RDS structure. And many people view that as a compliance or security gap potentially. And so what Baffle actually has is an integration with DMS that I'm about to show you, it also works for other cloud infrastructure providers, where we can integrate directly with database migration services on the fly and encrypt the data using field level encryption and using customer owned keys. And so it's a very streamlined approach to migrating your data into DBPaaS, and ultimately lets you move more workload into the cloud environment.

Harold Byun:

This is a blog that's available on the AWS DMS site that details our field level capabilities. So you can just do a quick search on that. It's also linked from our website. And we obviously can do the same thing with Azure and the data migration flow. It's a slightly different process, but very similar in principle in terms of how that data is actually landing and being migrated.

Harold Byun:

So what I'm going to do here is I'm actually going to switch into presentation mode just to give you a kind of a live demo here. I'm just going to share my desktop. You might get a little bit of a glimpse of things here. Kind of go over to this website. So you should be able to see this database migration or DMS dashboard. And here I have database migration tasks. And so what I'm going to do is I'm going to set this

up on the fly just so you can kind of see this at work. Just going to name this. We choose a replication instance, which we already set up. That takes a little bit of time so I'm not going to spend time doing that. And then I select my source database, which is listed here. And I'm going to choose a target database.

Harold Byun:

And so we're basically walking through what was on that prior slide. And we're going to migrate the data. And there's some other options here that you guys can all configure on your own. I'm going to accept the defaults. And then I'm going to specify what I actually want to copy over. In this case, I'm going to select a database called Baffle test, and this small table called pets. Giant 10 record table, but it's just for demo purposes here. And basically I'm going to select the rest of the defaults and create those tasks. And so you can see the create is successful.

Harold Byun:

And while this is actually being created and starting up, let me walk you through the dataset that we're actually going to be working with. So I go into this work bench environment, we can see that if I go into this Aurora database, which is the Aurora RDS target, and I show databases, there is no Baffle test present here right now. And so it's actually going to be migrated. And it's being migrated off of this database where we have, again, this database of pets of 10 records, and it's living in this database called Baffle test.

Harold Byun:

And so that's, what's going to be migrated. If we go into DMS and look at the setup, what we have is a Baffle demo target, which is Aurora MySQL. It's actually pointing to this IP address, which is a little different than the default pointing to an RDS data structure. And then our source was this source MySQL database, which is MariaDB, which is just a variant of MySQL, where we're sourcing that data. And the reason that we're pointing to this IP address in the demo target is because what we're actually running through is what we call the Baffle Shield, which is our encrypter and decrypter engine. And so this is integrated with DMS. And so as DMS is actually running through this task, if I go into this task, we see the current status of it. Looks like it's still spinning up.

Harold Byun:

And it'll migrate, and the data will hit the Baffle Shield, and encrypt on the fly for certain columns or sensitive fields. Let's see. [inaudible 00:15:49] goes. It looks like it's completed now. And so if I go into the data structure for Aurora and show databases, again, you can see that Baffle test has been created. I can refresh, and we'll see that here we have the Baffle test database. And if I go into the pets database, we'll do a select. And we can see that when we try and look at this data in the cloud, it's actually encrypted at the field level. And so this represents really how data would be protected in a data-centric fashion using the Baffle solution, where we can specify given columns that are actually protected.

Harold Byun:

And if I actually want to view the data or presented to an application, this is basically the Baffle Shield. I think if I go into the Baffle Shield and show databases, you can see that we have Baffle test. And I can run that same select. And we see that we're decrypting this information. And so this is how we're now interfacing with this RDS environment. And if I chose to insert a bunch of new records, so these other

potential pets, this tiger and a cat named Chipper, when I re-select, we can see that this data set's been updated, and we've added some additional data to the environment. So the tiger and the new cat. And if I go back to Aurora, we can see that we've added the tiger, but again, the more sensitive information is still encrypted at the field level.

Harold Byun:

And so this is kind of more or less, just a quick review of our integration with DMS. And the way we're doing this is obviously hooking in through some of the rest of the cloud infrastructure as well. So there's a number of different capabilities that we have associated with DMS. This is kind of the architectural flow that we just walked through, where there is a source database flowing through the Baffle Shield, and ultimately being encrypted in the RDS data store. There are a number of customers that are leveraging this today. Particularly in this case, there's an international bank that wanted to leverage the cloud infrastructure for developers, but didn't want any of their data to leave customer prem. Interestingly enough, there were also some limitations around Snowball availability, internationally, which is kind of the appliance for massive data migrations. And so we were able to support that migration into AWS, using DMS and encrypting certain columns on the fly.

Harold Byun:

Another aspect that we wanted to cover today is our integration with key management services. So obviously customer owned keys are a big deal as it relates to cloud. And so we've got a unique solution around how we integrate with key managers. And so I'm going to actually just jump into the demo really quick here to show you that. And then we'll kind of come back to the architecture. And so within the Baffle solution, we have the ability to basically enumerate any of the data schema and select the given fields that you want to encrypt.

Harold Byun:

And so here I have a data schema that has potentially sensitive information, some customer information. And I'm going to select this information or select these fields, and we can specify certain keys that we would want to leverage out of AWS KMS. If I go to the data structure here, just to show you what this looks like and get into this, so we run this query. You can see this data set is in the clear. I just refreshed it. These records are actually sitting in the clear.

Harold Byun:

And what I'm going to do using our Baffle Manager solution, which is our admin console, is drive an encryption migration. And what's happening here is we are actually sourcing the keys from AWS KMS. And with AWS KMS, we're using a customer managed key, or a CMK, and we're leveraging these database encryption keys identified by three and four that are available in the key store. And we are migrating them, or we are migrating the data using those keys through Baffle Shield. And so we can see that this is an ongoing process.

Harold Byun:

And the architecture of what's going on underneath the covers is effectively this. So the upper right-hand corner, that green icon is representative of Baffle Manager. It is pointing to AWS KMS. And it's basically saying I want to encrypt these fields, or these columns, with key three and four. And then it's telling our Baffle Shield, go get keys three and four. We never persist the keys, but they are held in

memory and apply those to the columns that have been selected. And we're actually driving that migration process. So if I go to this data structure now, and I refresh this query, we can see that those columns have actually been encrypted in terms of selecting those keys that we specified, and driving a data migration in place for this particular data structure.

Harold Byun:

This is how we actually integrate with AWS KMS. So you can see that the key view in the lower right-hand corner is what I was actually showing you. The upper-left is more or less kind of our setup in terms of what's required to actually drive that integration when you go into configuring a key store. We support a CMK and a deck hierarchy so you can roll the CMK or you can roll the deck. I go back to the key store, just so you guys all get a flavor for, we basically would select a key store, and you could fill out the appropriate secret access keys, and what region this is operating in. And the IAM user that actually has the right that we leverage to provision and request keys.

Harold Byun:

And when I go into this, what we'll actually do is also have the ability to also rotate keys if we wanted to. So from a compliance event perspective, if you needed to rotate the deck, which is often a requirement, at least annually, if not bi-annually, we would rotate the keys. And we also have the ability to support multiple key versions. And so by doing that, when I go back into the application, we'll see that key three, and any references to key three, have been rolled to key five. And key three is no longer available to us. And so you see that key five is referenced here on city. And if I wanted to encrypt any other data, or select a different field for encryption, key three is no longer available to us.

Harold Byun:

But I can also still choose to decrypt the data. And if I decrypt this, we'll still be able to decrypt all the data and back it out. So you aren't locked into Baffle, but it also demonstrates our ability to support multiple versions of keys. So hopefully, this gives you a sense of how we are integrating with key management. Orchestrating that client key exchange can often be a pretty complex effort in terms of how people are actually encrypting data. And so we're more or less simplifying that through what we would call a no-code approach. Meaning that you don't need to embed SDKs. You don't need to figure out some type of key exchange. And your developers can work on probably higher value application development versus figuring out how to do security in the application. This is something that we're more or less simplifying. So if I go through this, and check back on the data set and refresh. I mean, this job is completed. You can see the data is now in the clear, and we've effectively backed out any of the data and restored it to its original form in the data structure.

Harold Byun:

Moving on beyond key management, just additional feature sets, I guess, around key management. We support KMIP, PKCS #11 with the cloud key management solutions. We support REST API access because they're not necessarily always using a KMIP or PKCS #11 library, which are more standards for traditional key managers and HSMs. One other aspect of cloud and cloud native technologies that we also wanted to cover today was our support for serverless PaaS, in particular AWS Lambda. So this was an announcement that we made earlier in the year, particularly for clients that are looking to adopt more cloud native technologies. As many of you are probably aware, serverless code functions are obviously not in a persistent state on a physical server. Makes it very hard for them to intermediate with. They're

also passing through an API layer. So again, something that, from an architectural standpoint, may be difficult to intermediate with. Our Baffle Shield component is incredibly flexible from an architectural standpoint. Again, it's a no-code model, and we hook in below the API layer.

Harold Byun:

And so for a microservices or container environment, leveraging service mesh, or a serverless PaaS environment that's writing through a web services layer, we're a very efficient control point to encrypt data as it's being accessed in a Lambda environment. And effectively the architecture here is more or less events or events source trigger in Lambda function, which executes your code logic and access services and data. And you can see the Baffle Shield here sitting behind the Lambda function as data is being accessed, which can secure sensitive data, or as well invoked events based triggers on encrypted values. And so there's a way for us to facilitate a greater level of automation and integration with Lambda while still leaving the data in a very protected state.

Harold Byun:

So let me walk through what this actually looks like. If I go into the Lambda console, we have a bunch of different Lambda functions that I'm going to execute here just to kind of demonstrate. This is probably the easiest way to make it clear that we are executing a Lambda functions since it is just a piece of code. And what I've set up here is a environment where we've got two setups, one without Baffle, which is in red, and one with Baffle. And so if I do something like show databases with Baffle, we can see that there's a set of databases here. And if I go into the red structure, you can see that the same set of databases live here as well.

Harold Byun:

And what we're going to do is I'm going to execute a set of functions. One of these is creating a new database and inserting some data in it. And the other is going to execute a customer transaction insert. And so with this particular transaction database, there's 125 records in this table. And I'm going to execute the same command without Baffle, just so you all know we're not pulling any smoking mirrors. And you see the same record set exists. And what I'm going to do is execute this transactional insert from Lambda.

Harold Byun:

And so while that's executing, I'm going to go back to my databases, and you'll see that we've now created a new database called notes. And if I use notes and select some information from it, I'm going to select it in a HEX format because it's actually been encrypted. So it's been encrypted on the fly as we created that database, and inserted using a Lambda function from this table called meeting mins. And so you can see that the note name is encrypted, and there's also some note text here that is encrypted. Or did I... Sorry. Nope, TXT. Sorry. You can see that this is also encrypted data. And I'm just rendering it in HEX because if I didn't do that, it's going to break my terminal.

Harold Byun:

But if I also use notes here, which is now available, you can see that we can see that data in the clear. In particular, if I look at a specific record, we can see that there's a record here with a note ID of 11. And I'm going to do some updates on that note. So I will update the note. And once I finish updating the note, I'll also actually change status. So let's see, we've updated the note. You can see that we've now

updated the text. So that is something that is happening via that Lambda function. And again, this is actually going to stay in an encrypted form. And what I'll do while we're going to check that out on the other side is I'm going to update the note. And if I go to the encrypted version, or the back end Aurora DB... Oops. There. You can see that obviously it's encrypted, but we're still rendering that note update and clear. And if I go back to the note one more time, the note status has now been deleted because we changed that.

Harold Byun:

So our ability to exercise through that web services layer, the AWS web services layer, as things are being read and written via Lambda functions in the RDS is very efficient. And then for the last item I wanted to, I think, select count star. We'll see we've gone to 126 records. Some of you may recall that that was 125 before we started. And when I look at this, we can see the data in the clear. And it's name that I want. And then when I do a select, we can see that that's also encrypted data.

Harold Byun:

So hopefully that gives you a good sense of some of the flexibility in terms of the architecture that we have by being able to support API based calls, or service mesh related calls. Whether you're using serverless or microservices, that's generally a highly fluid access control point. And we're able to intermediate and support that type of field level encryption for RDS.

Harold Byun:

This is kind of where we see the evolution that a lot of customers are going through today. Obviously many of you still have on-premise infrastructure, and will continue to have on-premise infrastructure for many years. But as you look to migrate or lift and shift to cloud we can obviously integrate with DMS as we've shown you today in this presentation in demo. We obviously can support on-premise infrastructure as well. And then as you adopt cloud native technologies and architectures, we're able to apply the same data abstraction layer for cloud native architectures going forward.

Harold Byun:

This is just high-level architecture on serverless, and some of the capabilities that get opened up in terms of automating different functions and triggers to process data where no human actually would necessarily need to intervene or visually validate or look at data. It's a different type of processing mechanism that a lot of companies are starting to adopt.

Harold Byun:

So in summary, we're going to open it up for some questions, but how can we help? Well, one, we are going to help customers fulfill their portion of the shared responsibility model and actually protect the data that is in the cloud. We integrate with several cloud management services, data migration, database migration services, cloud key management providers, and serverless PaaS. And we're ultimately working to simplify a lot of the complexity of implementing data security encryption in cloud. So we provide a fair amount of orchestration, which hopefully you were able to gather with that. And ultimately this is going to help you move more workload to cloud, which will make you more nimble over time.

Harold Byun:

Just a little bit more about us. Upcoming, we're going to be at the Financial Services Information Sharing and Analytics Conference, the annual summit next week. And we'll also be sponsoring AWS re:Inforce, which is the security focused AWS conference in Boston in June. So if any of you are attending that and would like to meet, please don't hesitate to reach out or swing by. You can always reach us at info@baffle.io. And so that's pretty much the presentation that we have for today. I'd like to open it up for any questions that folks have.

Celine:

Awesome. So we have a few questions. The first one being what's your performance overhead?

Harold Byun:

Yeah. So it's a common question that we get. Well, I'll give you a very generic answer here in terms of general benchmarks that we have, which is roughly the bulk of the traffic that passes through us is in the clear. So any overhead is negligible. That's typically sub 5%. And then for encrypted data, we're generally in the range of transparent data encryption, which is five to 15% overhead. I want to qualify that, again, being that it really depends on your application workload, the nature of the query, the number of concurrent connections, the network latency, and every other variable that you could probably think of. But in general, we're in that range for a number of different environments.

Celine:

Do you support all RDS databases?

Harold Byun:

So it's a great question. Do we support all RDS databases? So we support all flavors of RDS on-premise and in cloud. For RDS specifically, we are MySQL and Aurora MySQL supporting. And then we have Aurora Postgres on the near term roadmap. We also support RDS SQL server and Oracle. So it's a pretty wide variety of different database platforms that we can support.

Celine:

And one last question, how is the solution deployed from a high availability perspective?

Harold Byun:

Yeah. So the solution is software. We deploy a software. And so we're typically on an EC2 instance when we're deployed in a cloud environment or in an infrastructure instance. And where we can work with any type of elastic load balancing or load balancing solution to basically load concurrency and fail over as well as integrating with database clusters. So that's typically how customers are using us in an HA mode. And so the availability is just subject to however you architect your load balancer and the availability numbers that you can achieve in that load balance environment.

Celine:

Thanks Harold. And on behalf of the entire Baffle team, thank you all for joining us today. And just a reminder that an email with a link to the recording will be sent to you for future viewing, and also to share with your colleagues. We look forward for you to join us at our next webinar. Thank you, guys.

Harold Byun:

Thank you.