Jillian Forde (00:00):

Hey, everyone, re:Invent is coming right around the corner, so we're going to give you a recap coverage of the keynotes that are happening every single day, starting Monday, November 27th with Peter DeSantis's keynote and all the way through Thursday, November 30th with Werner Vogels's keynote. So starting in December, we're going to be doing deep dives into all of these releases from the re:Invent Keynote, so you'll definitely want to make sure you've hit subscribe on the AWS Podcast so you can keep up with all of the exciting announcements that happen at re:Invent.

(00:41):

This is the AWS Podcast. I'm your host for today, Jillian Forde, and this one's for all the data lovers. If you're using Amazon Aurora MySQL, maybe some Amazon Redshift, we're going to be talking about Amazon Aurora MySQL zero-ETL integration with Amazon Redshift. I know that is a lot to say, but we're going to unpack what that exactly means, and I've got two experts on this very topic with me, Jyoti and Adam. So let's do some intros. Jyoti, please say hi and tell everyone what you do at AWS.

Jyoti Aggarwal (01:15):

Hey everyone, my name is Jyoti Aggarwal. I'm a senior product manager with Amazon Redshift and working on zero-ETL right now and really excited to be here and talking to you all.

Adam Levin (01:27):

Hey everyone, my name is Adam Levin and I'm Jyoti's counterpart on the product management team for Amazon Aurora. So super excited to be here and speak with you Jillian, thanks for having us.

Jillian Forde (01:38):

Super pumped to talk to both of you. There's a lot that's really going on underneath the covers between the two services. So let's really unpack it. Let's start with ETL. So maybe for those who are not familiar with what that exactly means, let's unpack what does ETL actually mean, what does it stand for, and how is it related to a data architecture?

Adam Levin (02:04):

ETL, three letters, E-T-L, it's an abbreviation that is part of the world of data architectures, specifically around data integration. The E stands for extract, the T stands for transform, and the L stands for load. And so it's process of getting data out of a source data store, extract, doing something with that data to get it ready for analysis or for use, the T, transform, and then loading it into a destination, typically like a data warehouse or some other data analytics platform, so that's the L, the load. It doesn't always happen in that order. Sometimes it's ETL, sometimes it's ELT. It sort of depends on what tools you're using, what your use case is. At a high level, you're getting data out of one data store and putting it in another that's optimized for something else. So for example, Aurora is an operational database, Redshift is a analytics optimized database. We're extracting data from one, loading it into the other. And this is all a really important part of customer's data architectures.

Jillian Forde (03:19):

Love that definition, but what's interesting about this specific feature is that you've got the words zero-ETL. So let's unpack that. So what is Amazon Aurora MySQL zero-ETL integration with Redshift?

Adam Levin (03:37):

With zero-ETL, we're really trying to make it easy for customers to get data from Aurora to Amazon Redshift. And so the zero is we're signaling that customers don't need to go through all of the heavy lifting, the operational headaches, the operational overhead of building a data pipeline to handle ETL. With zero-ETL, customers click a couple buttons and we get the data from Aurora to Redshift, we keep it in sync, and we maintain and manage this integration on behalf of customers. And there's a lot of reasons why data pipelines are tricky.

Jillian Forde (04:29):

Yeah, super cool. I'd love to unpack more of the use cases here. So what are you seeing are the use cases for Aurora MySQL zero-ETL integration with Redshift?

Jyoti Aggarwal (04:42):

Yeah, so once the data lands in Redshift, really, customers can leverage everything and all the powerful tools and features that Redshift already has, such as they could end up sharing this data with other consumers, they could end up using the SQL support that we have in Redshift, or using materialized views or training ML models on their warehouse itself. Generally speaking though, once this data is available in Redshift, we are seeing customers join it with other data and which probably is already existing in their warehouse from other data sources. And they're running transformations to their liking using Redshift materialized views, and probably they're also building dashboards on top of this data using Amazon QuickSight because Redshift is already integrated with QuickSight. And since with zero-ETL, the data really lands near real time in Redshift, customers can leverage these dashboards to get business critical information and take better informed decisions across their organization.

(05:55):

And another popular use case that we are seeing with customers is, like I was talking about, Redshift data sharing. They end up sharing the zero-ETL data across multiple other consumers or across different business units, be it in the same AWS account or a different AWS account. And this is helping them leverage the same application data across different business units. And the third use case that we are seeing gain traction or is a very good use case to explore is creating ML models using application data that is landing into Redshift using zero-ETL. Redshift ML allows customers to seamlessly build models on this data right from their warehouse using SQL itself. So they can run inference queries to take guided prescriptive actions across multiple business units.

Jillian Forde (06:55):

Wow, I mean super handy by just being able to take the data in Aurora, no ETL, you dump it in Redshift, and now you can be able to do your queries or other capabilities like you were saying like with Redshift ML with that data that was in Aurora. I'm really excited to see what people start to do with this. But I know there's some people who are listening and they're probably not really sure if it's a good fit for them. So for someone who's listening and they're not sure, can you describe what are maybe the, let's call them symptoms, that maybe they are experiencing right now that can help them decide, oh, I wonder if Aurora zero-ETL with Redshift is actually a good fit?

Adam Levin (07:40):

There are a lot of reasons why zero-ETL integration may help customers out when all they want to do is get data from Aurora to Redshift. And this really comes down to challenges operating data pipelines. So there's a few categories here. There are things like data integrity or data quality where if a data pipeline breaks down at some point, you may have incomplete data in Redshift and you may not have sort of a

mirror image of data that's in your operational database and what's in Aurora. And so incomplete data is one source of challenge. Another challenge is really around data lag and so the time it takes for data to get from Aurora to Redshift. And in a lot of customers that we've spoken to, they have expectations around getting data, like if data gets from Aurora to Redshift within a day, they're happy. But with zero-ETL integration, data can get there in seconds. And so that really opens up a whole set of new use cases that customers can start to think about when data lag is really reduced.

(08:51):

Another set of challenges is around schema handling and data model changes on the source. And so when changes happen on the source, with zero-ETL, we handle that automatically and so we update Redshift to reflect those changes. With other solutions, there may be a whole bunch of operational work that's needed to get those things back in sync. And then the last category is just sort of general monitoring and logging and resiliency. So if something goes wrong, how much effort is it to fix and get your pipeline back up and running? And with zero-ETL, all of this is automated and so we recover regardless of where the issue is, we can resync the data as necessary and we've built a lot of optimizations into the sort of ongoing streaming and replication capabilities to make sure that your data is there when you need it.

Jillian Forde (09:54):

It's really fascinating like just when you look at all the different just areas that zero-ETL addresses like the time savings that these data engineers who are having to do this right now, and now with zero-ETL, they don't have to spend as much time debugging and error handling this portion of their ETL system. Yeah, it's really exciting. So what about some of the other challenges that maybe customers are facing with their ETL? Is there anything else that you wanted to add?

Adam Levin (10:25):

I think one other interesting area is we've heard from customers, obviously, like they spend a lot of significant time and resources building and managing pipelines. When you remove a lot of the undifferentiated heavy lifting around building these pipelines, you can build additional pipelines or build additional integrations. And so one key use case here is around building integrations from multiple source databases into a single Amazon Redshift data warehouse. And so you can run analytics across a whole bunch of petabytes of data from multiple source Aurora clusters. An example there might be like a global manufacturing company with factories in many locations and each one of those factories represents a different Aurora cluster, but the headquarters wants to see aggregate views across all of that data. And so this helps simplify creating that aggregate set of analytics because instead of having to build and maintain pipelines for each one of those Aurora clusters, it's a few clicks and integration is there. And so that company can focus on decision-making and not necessarily on the operational load of keeping those data connections up and running.

Jillian Forde (11:55):

Super interesting. Jyoti, I want to go back to you and really expand on a question we were talking about earlier about the use cases. So can you talk more about what are you seeing that customers can now do with this Aurora MySQL zero-ETL integration with Redshift?

Jyoti Aggarwal (12:15):

We touched upon multiple Redshift functionalities that are available at their disposal once this data lands into Redshift. And really, Redshift being the central part of the system, they can explore Redshift

data sharing or build materialized views on top of it to run transformations to their liking or have a central governance on this data within Redshift and still end up connecting multiple other business units to this data. I'd also like to take listeners on a journey that we've been on with zero-ETL because we announced it at re:Invent last year and customers were actually in disbelief.

(12:57):

We announced something that was actually replacing the decades old ETL pipelines that probably were existing. Once they tried us in public preview, they were actually amazed at how the zero-ETL integration allowed them to build their analysis environment at minimal cost and maximum performance. It eased their lives so much, but they ended up getting a better performing system. And some customers who went through the challenges of building these ETL pipelines gave us feedback that what took them probably months to years or weeks can now literally happen within minutes or a day. It's really from customers being in disbelief of this idea to adopting this and loving this so much, it's really a journey that we've been on with zero-ETL.

Jillian Forde (13:54):

So cool. And what are some of the things that you've seen them say to you that they really love about this integration?

Jyoti Aggarwal (14:00):

They love the performance, to be honest, because it's a near real time replication of data from Aurora to Redshift and the data lands near real time within Redshift, so customers love that once they make some changes or they run some DDL or DML for testing purposes on Aurora, the data magically appears in Redshift without them doing literally anything in the middle. The system is smart enough and intelligent enough that it kind of monitors and does a lot of calculations in the background to make sure the end-to-end systems are up and running and customers get a coherent view of their data in Redshift.

(14:43):

And customers also love the ease of use because we've built a very intelligent system and a lot of work has gone into building the system so that it knows when the integration is failing and it recovers on its own and it knows up till the table or a column level what's happening on this data in the background. It's really a hands-off kind of an environment that they get and they love how easy it is for them to use. End up utilizing all their time in running analytics rather than maintaining these ETL pipelines.

Jillian Forde (15:17):

Well, yeah, this is really exciting. You've got time left for really just a few more questions, but I'm sure people just really want to go and get started, so we'll get to that at the end. But so I'd love to really just unpack how does it actually work, getting the data from Aurora to Redshift?

Jyoti Aggarwal (15:36):

So we are a CDC based replication model where we write transactions on Aurora storage using enhanced binlog and then replay them on Redshift, and a lot of work has gone into making the CDC performant for zero-ETL. Customers just tell us the source Aurora MySQL DB cluster that they want to run analytics on and we then go and do a bunch of work on their behalf. Traditionally speaking with binlog in MySQL, first the transaction is written and then the binlog files are written to storage. But with enhanced binlog, we push as much processing of these binlog files as possible to the Aurora storage. We also write the binlog in transaction files on a per transaction basis in parallel. We've seen massive

improvements in the amount of transaction a same sized Aurora cluster and process with enhanced binlog.

(16:36):

We then go and export this data from Aurora Storage and load it into Redshift. And all the subsequent changes that customers make to Aurora are copied over as continuous CDC to Redshift. And system handles most of the DDLs and DMLs on its own and changes just automatically appear on Redshift. And while all this is going on, we are also monitoring the integration and we detect whether tables need to be receded or integration needs some fixing or we take care of that on customer's behalf. Just to give customers more confidence in the end-to-end system, we've also built in observability for customers around lag and integration metrics down to table and column level so that they get more visibility into this end-to-end system and are more confident in adopting it.

Jillian Forde (17:34):

I'm really excited. I just get so pumped when there's anything that just saves people time until they can just really focus on the parts of their business that they are super pumped about. This is definitely one of them, so please tell people how is it that they can actually go and get started.

Jyoti Aggarwal (17:52):

They can learn more using Amazon Aurora and Amazon Redshift documentation. We also have AWS News blog that we can share. We'll be having a getting started blog for them as well.

Adam Levin (18:05):

The zero-ETL integration between Amazon Aurora MySQL and Amazon Redshift is generally available in nine regions and so customers can just log right into the AWS console and start creating a zero-ETL integration from the RDS console. You need to create a Aurora MySQL 3.05 or hire a cluster and an Amazon Redshift data warehouse, and then you just go and create the integration from the RDS console.

Jillian Forde (18:38):

Super cool. I'm really excited for people who are listening today. Jyoti, Adam, thank you so much for being here on the AWS Podcast.

Adam Levin (18:48):

Thanks for having us, Jillian.

Jyoti Aggarwal (18:49):

Yeah, it was a very fun conversation.