Transcript for Introduction to Migration Strategies video

You have just learned about using a structured framework called AWS Cloud Adoption Framework, or AWS CAF. This framework guides customers in progressing their current business and IT landscape to a more cloud-ready architecture. AWS CAF helps to uncover existing gaps an organization might have and gets it to a stage of readiness.

To make decisions about the workloads you are migrating, you must fully understand the migrating environment, the individual workload, and how each component relates to other components. You will typically begin your project by discovering the existing workloads in the customer's IT landscape.

The data collected from your customers IT landscape allows you to make better decisions when choosing migration strategies. There are a number of tools you can use to complete the discovery process. It's also important to understand the business goals, stakeholders, and impact to make an informed migration strategy decision.

Some questions to ask customers and application business owners when gathering data to inform the right migration strategy for their environment include the following:

- Who owns or supports the application?
- Which business units does this application support?
- What is the relative importance (or criticality) of the application to the business?

After you have aligned the collected data for the organization's business and technical drivers, next is choosing the actual migration strategy.

When customers migrate to Amazon Web Services, or AWS, they choose strategies that best fit the application being migrated. AWS provides seven common approaches, which include: refactor, replatform, repurchase, rehost, relocate, retain, and retire. Because each application is unique, enterprises often use multiple strategies for separate applications.

For example, you might recommend that your customer use a rehost strategy if they must quickly migrate and scale to satisfy a business need. Or, you might recommend a refactor strategy if your customer must add features, performance, or scale that would be a challenge or difficult to achieve with their current application design.

Let's walk through these seven strategies in more detail. After the explanation, I will show you how these strategies can be applied with an example.

1. Starting with the relocation strategy. Relocate means to move applications or workloads to the cloud with minimal downtime and without any change to your existing operations or technical stack.

- Why would you want to relocate in the first place? Let's say you're running applications on VMware, and your data center lease is about to expire without being renewed. A common use case for this migration scenario is to quickly relocated to AWS using the infrastructure that's most familiar to your customer. A relocate strategy would be the fastest method to use in this type of scenario.

2. A second commonly employed migration strategy is rehost, also known as "lift and shift. In this approach, applications or workloads are migrated to AWS without any changes to the underlying operating system or application stack. Customers can use lift and shift strategy to quickly migrate, and then focus on application modernization. After the application is in the cloud, it will be more convenient to get to a cloud-optimized state.
3. A third migration strategy is to replatform, also called the lift, tinker, and shift. It's like changing an automobile engine out for one with higher performance, or adding newer functionality. In practice, this could look like modernizing the database layer of your customer's application to a managed service. Which means spending less time on day-to-day operations and more time focused on your business logic.
4. The next migration strategy is to repurchase applications. This means replacing your existing application with a different version or product. This approach makes sense when you are looking for a strategy that provides more business value than the existing, on-premises application. Such benefits include accessibility from anywhere, no infrastructure maintenance, and pay-as-you-go pricing models. Repurchasing the application typically reduces costs associated with maintenance, infrastructure, and licensing.
5. A fifth migration strategy is to refactor. This means re-designing application architecture or rewriting an application before the migration to make it a cloud-native application. Overall, it's taking monolithic applications and breaking them down to modular architecture, which is beneficial for adding new features, improving performance, and scaling to meet your customers' current resources.
6. The sixth migration strategy is to retire. This is useful if your customers discover that an application is no longer necessary and can be decommissioned. After doing an audit of on-premises applications, you may be surprised with how many applications are up and running but simply are no longer needed or not being used!
7. The last migration strategy, is retain. Your customers can retain the application for now, or revisit at a later date. This strategy is useful if some applications can't be immediately migrated due to licensing or other reasons. Retain could also be the appropriate strategy if your customer recently invested in upgrading their current system and you want to postpone migrating the application until the next refresh. Examples include mainframe applications and databases, which need dedicated time and effort investments.

AWS Partner: Accreditation (Technical)

There are certain migration styles that fit different situations and needs. Just to re-emphasize, there is no one-size-fits-all approach to migration. Here's an example scenario where a school district engaged an AWS Partner to improve performance and scalability of their digital operations. This school district used a mix of migration strategies that included: retire, repurchase, relocate, rehost, and refactor.

- In this scenario, the partner organized meetings with the application owners and identified apps that were no longer needed or didn't have clear ownership. These applications ended up being retired.
- The partner also worked with the school district to transition one of their applications to a software as a service, or SaaS, model to reduce operational overheads. This effort used a repurchase strategy.
- Another thing the partner did was move the school district's containerized internal applications to Amazon Elastic Container Service, or Amazon ECS, to scale their digital systems more efficiently. This effort used a relocate strategy.
- In speaking with application owners, the partner uncovered that the school district had already invested substantially in existing vendor licenses and support. They took a rehost strategy for these applications, moving them as-is to Amazon Elastic Compute Cloud, or Amazon EC2, instances.
- Lastly, the partner suggested rewriting their multi-tiered applications as a modular architecture to use EC2 scaling capabilities. Using a refactor strategy helped to improve the performance and cost to operate these applications.

AWS Partners can help to accelerate the customer cloud journey by providing business expertise, infrastructure and application migration expertise, migration tools, education, and ongoing support to customers. For more information about migration strategies, check out the resources provided after this video.

After you decide on the migration strategy, the next thing you should think about is how your applications adhere to AWS best practices. I will introduce some helpful tips in the next training topic.

See you in the next video!