

Native Third-party tools boost interoperability in cloud

AWS exposes APIs and provides container services for portability, but does it go far enough? Some IT pros still find themselves leaning on third-party tools to fill critical gaps.

Public cloud providers tend to turn a cold shoulder toward the notion of interoperability. While the customer might find it advantageous to smooth the paths between multiple cloud environments, a cloud provider looks at it as a rebuke of its service and infrastructure offerings.

But, in doing so, providers miss the key point. For many enterprises, interoperability in cloud is not simply a feature; it's a necessity. Clouds are not islands of systems. They might interact with on-premises resources or with other public clouds, and enterprises increasingly need this type of **multi-cloud or hybrid cloud support**.

AWS takes an API-first approach to interoperability, offering a common mechanism that applications -- inside or outside AWS -- can easily use. These APIs could point to microservices, standard web services, database call-level interfaces or even IoT devices, such as a thermostat or robot on a factory floor. Enterprises use these APIs to connect remote systems, local resources and AWS-hosted workloads.

AWS exposes most of its infrastructure or service offerings as a set of APIs that an enterprise can use to tie in non-Amazon services to meet the exact needs of its application.

However, there is a downside to achieving interoperability in cloud through these APIs alone. First, you will need to bind APIs together that are native to different cloud providers, such as AWS to Google. This means you must code the links between these APIs to translate the way each API treats data, security, structure and exception handling. Ultimately, it's your responsibility to make the connection work, not each respective cloud provider.

How cloud providers approach interoperability

Here's a look at three categories of services from cloud providers that aid interoperability:



CLOUD PROVIDER	API MANAGEMENT	CONTAINER SERVICES	DATABASE API
AWS	Amazon API Gateway	Amazon Elastic Container Service	Varies by native database
Azure	Azure API Management	Azure Container Service	Varies by native database
Google Cloud Platform	Google Cloud Endpoints	Kubernetes Engine	Varies by native database

As mentioned above, APIs and proper management of them are critical to interoperability in cloud. Cloud applications use the APIs of their host cloud, but other public and private clouds need to discover, embed and interact with APIs via remotely hosted applications.

Container services also fit into interoperability. An IT pro creates a container-based application or containerizes an existing app, which enables them to run within a layer of abstraction. This approach buys you portability, as containers move from cloud to cloud with little to no modifications.

Database APIs are also important, but they vary by cloud database brand. For example, DynamoDB has its own set of APIs, as does Amazon Redshift and other

native AWS databases. The same goes for Azure SQL Server and Google's native databases.

Data integration decisions

When it comes to data integration, there are third-party offerings that enable more interoperability in cloud, as they abstract away the complexity and differences of each database and database API.

AWS, Google and Azure offer primitive data integration around native messaging systems, but there are more competent third-party data integration technology products. While many consider this a weakness on the part of cloud providers, if you turn to a third-party data option, they will treat each integration with the source or target cloud data stores equally and independently.

These tools are especially helpful for enterprises that have a lot of legacy data that sits on premises. That data might need to sync, for example, with an AWS-hosted data store to which applications can connect to perform analysis. An ops team typically synchronizes data with cloud-based apps or stores via events, such as the input of customer information, or on a schedule, such as an overnight batch update.

Some tools can manage the semantic differences from the source to target systems, including structure and content transformation. For example, a third-party tool might address the issue of customer numbers being assigned to a text field in one database, but a numeric field in another. Dell Boomi or SnapLogic are two examples of tools that can help with these data integration tasks.

AWS, like other public cloud providers, can't be a silo. It must exist within an ecosystem that includes all other types of systems, residing in all other types of places.