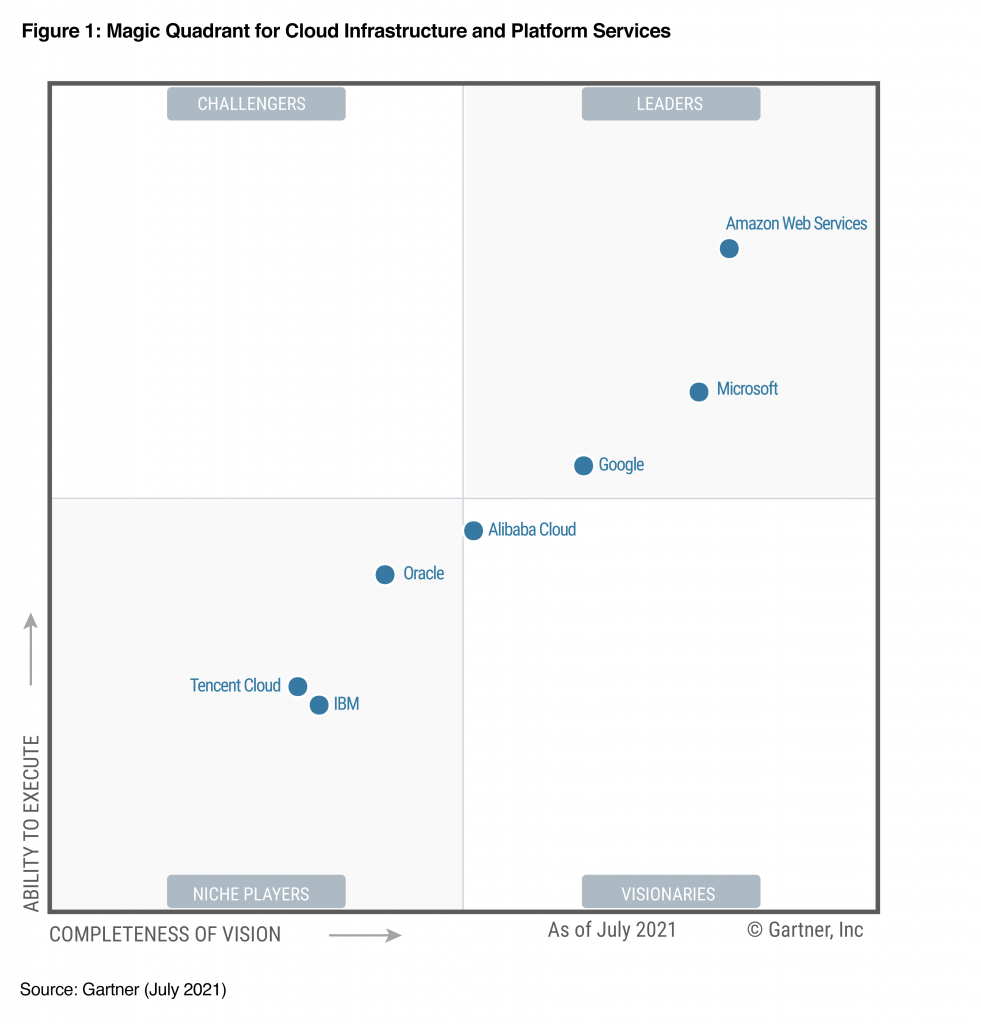
Big Data na AWS

"With AWS a new server can be up and running in three minutes (it used to take Eli Lilly seven and a half weeks to deploy a server internally) and a 64-node Linux cluster can be online in five minutes (compared with three months internally)... The deployment time is really what impressed us." Dave Powers, Associate Information Consultant at Eli Lilly

## Amazon Web Services

A Amazon Web Services (AWS) é uma plataforma de serviços de computação em nuvem líder de mercado que oferece diferentes soluções para Big Data, por vezes inspiradas nas que vimos ao longo de todo este curso.



Fonte: [AWS](https://aws.amazon.com/blogs/aws/aws-named-as-a-leader-for-the-11th-consecutive-year-in-2021-gartner-magic-quadrant-for-cloud-infrastructure-platform-services-cips/)

## Propósito

O propósito deste material é te introduzir ao ambiente AWS, apresentando algumas de suas principais ferramentas e como armazenamento e processamento de dados envolvendo grande volumetria, variedade e velocidade podem ser implementados.

## Armazenamento

Existem inúmeras soluções de armazenamento dentro do ambiente AWS, dentre eles se destacam bancos de dados relacionais gerenciados e escaláveis como o [Amazon Aurora](https://aws.amazon.com/rds/aurora/) e bancos de dados não relacionais como o [Amazon DynamoDB](https://aws.amazon.com/dynamodb/) ou o [Amazon Neptune](https://aws.amazon.com/neptune/).

Entretanto, como serviço para armazenamento de objetos, similar ao HDFS do Apache Hadoop ou o DBFS do Databricks explorado em materiais anteriores, existe o [Amazon S3](https://aws.amazon.com/s3/) que apresenta altíssima resiliência (99,999999999%) e diferentes classes de armazenamento. Dados persistidos no S3 podem ser acessados não apenas por outros serviços da AWS, mas também pelo Databricks, por exemplo.

## Processamento

Há muitas ferramentas para processamento de dados nesta plataforma, dentre elas, em cenário de Big Data, destacam-se:

### AWS Glue

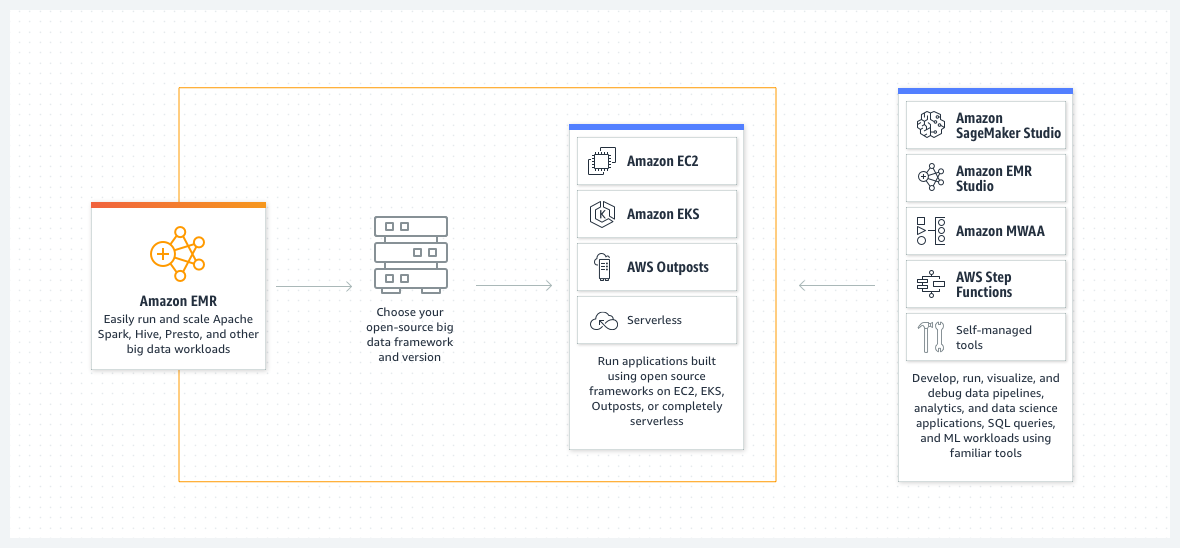
O [AWS Glue](https://aws.amazon.com/pt/glue/) é um "serviço de integração de dados com tecnologia sem servidor que facilita a descoberta, preparação, movimentação e integração de dados de várias fontes para análise, machine learning (ML) e desenvolvimento de aplicações." Com ele podemos utilizar o Apache Spark em uma experiência totalmente gerenciada, selecionando a quantidade de nós que processarão o código criado. A cobrança é feita através do tempo em que o código esteve em execução, considerando seu total de nós. Desta forma, não é necessário se preocupar com sistema operacional, instalar ou framework ou mesmo atualizá-lo.

#### Glue Data Catalog

Além da solução de processamento de dados o AWS Glue também tem uma ferramenta para catalogação, que de forma similar ao Apache Hive organiza dados em bancos e tabelas (inclusive de diretórios do S3) para permitir governança (especialmente quando combinado ao [AWS Lake Formation](https://aws.amazon.com/pt/lake-formation/)) e mais fácil acesso por outros produtos, como o próprio Glue com Spark ou [AWS Athena](https://aws.amazon.com/pt/athena/) para consultas.

### Amazon EMR

"[Amazon EMR](https://aws.amazon.com/pt/emr/) é a solução de big data em nuvem líder do setor para processamento de dados, análise interativa e machine learning que usa estruturas de código aberto, como Apache Spark, Apache Hive e Presto." Com ele é possível construir aplicações serverless ou provisionando clusters utilizando Amazon EC2, Amazon EKS etc.



Fonte: [AWS](https://aws.amazon.com/pt/emr/)

### AWS SDK for pandas (AWS Wrangler)

Se por ventura precisar trabalhar com arquivos parquet ou tabelas do catálogo de dados do Glue com Python comum, como em AWS Lambdas, por exemplo, indico fortemente o uso da biblioteca [AWS SDK for pandas (AWS Wrangler)](https://pypi.org/project/awswrangler/) que pode lhe trazer mais praticidade e agilidade dentro desta plataforma se comparada a bibliotecas tradicionais como fastparquet ou PyArrow nas quais ela se baseia.

Com ela é possível ter uma experiência muito próxima daquela oferecida pelo Spark e Databricks visto anteriormente, onde facilmente é possível recuperar dados de tabelas em dataframes, ler arquivos de diferentes formatos e salvar dataframes atualizando o catálogo.

## Consulta

### AWS Athena

Para consultas a dados organizados em arquivos podemos utilizar o já mencionado AWS Athena combinado ao catálogo de dados do Glue. Sua cobrança é baseada na quantidade de dados escaneados em cada consulta e se beneficia de formatos colunares como parquet e compressões. Vale ressaltar que o Athena, assim como o Apache Hive, permite o particionamento das tabelas e pode ter seu funcionamento otimizado quando bem utilizado, consulte na sessão de indicações deste material um link com dicas para melhoria de performance.

### Amaozn Redshift

Uma outra solução para consulta de dados em grande volumetria é o [Amazon Redshift](https://aws.amazon.com/pt/redshift/), banco de dados colunar escalável horizontalmente que apresenta também soluções de [Machine Learning](https://aws.amazon.com/redshift/features/redshift-ml/) e possibilidade de conexão com muitas ferramentas de visualização.

Além da persistência de dados dentro de um cluster RedShift (que também pode ser utilizado de forma [serverless](https://aws.amazon.com/pt/redshift/redshift-serverless/) e sob demanda) o [Redshift Spectrum](https://docs.aws.amazon.com/redshift/latest/dg/c-getting-started-using-spectrum.html) permite a criação de um Data Lakehouse, com dados sendo persistidos em disco (S3) e consultados pela ferramenta.

## Indicações e Bibliografia

[Amazon Web Services](https://aws.amazon.com/)

[Amazon Aurora](https://aws.amazon.com/rds/aurora/)

[Amazon S3](https://aws.amazon.com/s3/)

[AWS Glue](https://aws.amazon.com/pt/glue/)

[AWS Glue Data Catalog](https://docs.aws.amazon.com/glue/latest/dg/catalog-and-crawler.html)

[AWS Lake Formation](https://aws.amazon.com/pt/lake-formation/)

[AWS Athena](https://aws.amazon.com/pt/athena/)

[Amazon EMR](https://aws.amazon.com/pt/emr/)

[AWS SDK for pandas (AWS Wrangler)](https://pypi.org/project/awswrangler/)

[Performance Tips Athena](https://aws.amazon.com/pt/blogs/big-data/top-10-performance-tuning-tips-for-amazon-athena/)

[Amazon Redshift](https://aws.amazon.com/pt/redshift/)

[Amazon Redshift Serverless](https://aws.amazon.com/pt/redshift/redshift-serverless/)

[Redshift ML](https://aws.amazon.com/redshift/features/redshift-ml/)

[Redshift Spectrum - Getting started](https://docs.aws.amazon.com/redshift/latest/dg/c-getting-started-using-spectrum.html)