

Semana 6

Objetivo: Nuvem com produtos/serviços para analytics na AWS que serão usados no programa.

Conteúdo: Computação em Nuvem:

- Soluções para Analytics
 - AWS EMR
 - AWS Glue
 - AWS Athena
 - Amazon QuickSight
 - AWS Lake Formation

Desafio: Realizar o curso completo, realizar os 12 exercícios assistindo as vídeo aulas e os labs propostos neste material.

AWS EMR

O Amazon EMR é a plataforma de big data nativa da nuvem líder do setor, que permite que as equipes processem grandes quantidades de dados com rapidez, de forma econômica e em grande escala. Usando ferramentas de código aberto, como o Apache Spark, o Apache Hive, o Apache HBase, o Apache Flink e o Presto, combinados à escalabilidade dinâmica do Amazon EC2 e ao armazenamento escalável do Amazon S3, o EMR oferece às equipes analíticas os mecanismos e a elasticidade para executar análises na escala de petabytes por uma fração do custo dos clusters locais tradicionais. Desenvolvedores e analistas podem usar Notebooks EMR baseados em Jupyter para permitir o desenvolvimento iterativo, a colaboração e o acesso a dados armazenados nos produtos de dados da AWS, como o Amazon S3, o Amazon DynamoDB e o Amazon Redshift, para reduzir o tempo para obtenção de informações e para operacionalizar rapidamente as análises.

Clientes de diversos setores usam o EMR para proteger e manipular de forma confiável grandes conjuntos de casos de uso de big data, o que inclui machine learning, transformações de dados (ETL), simulações financeiras e científicas, bioinformática, análises de registros e aprendizagem profunda. O EMR dá às equipes flexibilidade para executar casos de uso em clusters de curta duração específicos que são dimensionados automaticamente para atender à demanda ou em clusters de longa duração com alta disponibilidade que usam o novo modo de implantação multi-master.

Fácil de usar

O Amazon EMR simplifica a criação e operação de ambientes e aplicativos de big data. Os recursos EMR relacionados incluem fácil provisionamento, escalabilidade e reconfiguração de clusters e notebooks para desenvolvimento colaborativo.

Provisionamento de Clusters em minutos: é possível iniciar um cluster EMR em minutos. Você não precisa se preocupar com provisionamento da infraestrutura, a configuração de clusters, a configuração ou os ajustes. O EMR cuida dessas tarefas para que você concentre suas equipes no desenvolvimento de aplicativos de big data diferenciados.

Aumente ou diminua facilmente os recursos para atender às necessidades empresariais: é possível aumentar e diminuir a escala facilmente em políticas e deixar o cluster EMR gerenciar automaticamente os recursos computacionais para atender suas necessidades de uso e performance. Isso melhora a utilização do cluster e economiza custos.

Provisione rapidamente notebooks para vários usuários: os Notebooks EMR fornecem uma experiência de notebook poderosa e fácil de usar que permite que seus usuários comecem imediatamente a experimentar com o Apache Spark. Os Notebooks EMR são baseados no Notebook Jupyter e ajudam cientistas de dados, analistas e desenvolvedores a preparar e visualizar dados, desenvolver aplicativos e executar análises interativas usando clusters do EMR. Os Notebooks EMR são hospedados fora do cluster do EMR. Portanto, não existe servidor de notebook ou software de notebook para ser submetido a manutenção, implantação ou upgrade.

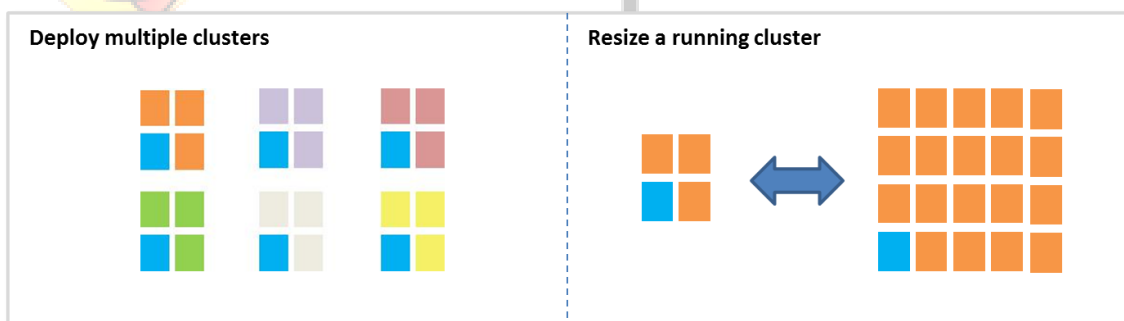
Alta disponibilidade em um clique: a configuração multi-master para aplicativos como o YARN,

o HDFS, o Apache Spark, o Apache HBase e o Apache Hive é feita com uma única caixa de seleção. Ao habilitar o suporte multi-master no EMR, o EMR configurará esses aplicativos para a alta disponibilidade e, no caso de falhas, fará fail-over automaticamente para um master em espera para que o cluster não seja interrompido. Os hosts são monitorados para detectar falhas e quando problemas são detectados, novos hosts são provisionados e adicionados ao cluster automaticamente.

Reconfigure clusters em execução facilmente: agora você pode modificar a configuração de aplicativos executados em clusters do EMR, incluindo Apache Hadoop, Apache Spark, Apache Hive e Hue sem reiniciar o cluster. A reconfiguração de aplicativos do EMR permite modificar aplicativos em execução sem necessidade de desativar ou recriar o cluster. O Amazon EMR aplica as novas configurações e reinicia o aplicativo reconfigurado de forma controlada. As configurações podem ser aplicadas usando o console, o SDK ou a ILC.

Elástico

O Amazon EMR permite que você provisione de modo rápido e fácil a quantidade de capacidade necessária, além de adicionar e remover capacidade de forma automática ou manual. Isso é muito útil se você tiver requisitos de processamento variáveis e imprevisíveis. Por exemplo, se o maior volume de processamento ocorrer à noite, talvez você precise de 100 instâncias durante o dia e 500 instâncias à noite. Mas, por outro lado, você pode precisar de uma quantidade significativa de capacidade por um período curto de tempo. Com o Amazon EMR, você pode provisionar rapidamente centenas ou milhares de instâncias, escalar automaticamente para atender a requisitos de computação e encerrar o cluster quando o trabalho for concluído (para evitar pagar por capacidade ociosa).



Há duas opções principais para adicionar e remover capacidade:

Implantar vários clusters: se você precisa de mais capacidade, pode executar rapidamente um cluster e encerrá-lo quando não precisar mais. Não há limite para quantos clusters você pode ter. Você pode querer usar vários clusters se tiver vários usuários e aplicativos. Por exemplo, você pode armazenar seus dados de entrada no Amazon S3 e executar um cluster para cada aplicativo que precisa para processar dados. Um cluster pode ser otimizado para CPU, um segundo cluster pode ser otimizado para armazenamento, etc.

Redimensionar um cluster em execução: com o Amazon EMR, é fácil escalar automaticamente ou redimensionar manualmente um cluster em execução. Aumente a escala horizontal de um cluster para aumentar temporariamente seu poder de processamento ou reduza a escala horizontal do cluster para economizar custos quando a capacidade estiver ociosa. Por exemplo, alguns clientes adicionam centenas de instâncias aos seus clusters quando ocorre seu

processamento em lote, e removem as instâncias extras quando o processamento termina. Agora, ao adicionar instâncias ao cluster, o EMR poderá começar a utilizar capacidade provisionada assim que estiver disponível. Ao reduzir a escala, o EMR escolherá proativamente nós ociosos para reduzir o impacto sobre os trabalhos em execução.

Baixo custo

O Amazon EMR foi desenvolvido para reduzir o custo do processamento de grandes quantidades de dados. Alguns recursos que mantêm o baixo custo incluem baixa definição de preço por segundo, integração com instâncias spot do Amazon EC2, integração com instâncias reservadas do Amazon EC2, elasticidade e integração com o Amazon S3.

Preço baixo por segundo: a definição de preço do Amazon EMR é por segundo de instância, com um mínimo de um minuto, a partir de 0,015 USD por hora de instância para uma instância pequena (131,40 USD por ano).

A definição de preço do Amazon EMR é simples e previsível: você paga uma taxa por segundo para cada segundo usado, com um mínimo de um minuto. Por exemplo, um cluster de 10 nós executado por 10 horas custa o mesmo que um cluster de 100 nós executado por 1 hora. A taxa por hora depende do tipo de instância usado (por exemplo, padrão, CPU de alto desempenho, armazenamento de alta capacidade, etc.) e o faturamento é calculado por segundo e exibe o tempo no formato decimal. Os preços por hora variam de 0,011 USD/hora a 0,27 USD/hora (94 USD/ano a 2.367 USD/ano).

O preço do Amazon EMR é adicional ao preço do Amazon EC2 (o preço dos servidores subjacentes) e ao preço do Amazon EBS (se há volumes do Amazon EBS anexados). Esses preços também são cobrados por segundo, com um mínimo de um minuto. Você pode escolher entre diversas opções de definição de preço do Amazon EC2, incluindo instâncias sob demanda, instâncias reservadas de 1 e 3 anos e instâncias spot. As instâncias spot são capacidade ociosa do Amazon EC2, disponível com um desconto de até 90% em relação às instâncias sob demanda.

Integração de spot do Amazon EC2: as instâncias spot do Amazon EC2 permitem que você indique seu próprio preço pela capacidade do Amazon EC2. Basta você especificar o preço por hora máximo que está disposto a pagar para executar um determinado tipo de instância. Desde que o preço sugerido por você exceda o preço de mercado do spot, você manterá as instâncias e pagará, normalmente, uma fração do preço por demanda. O preço spot oscila com base na oferta e na demanda das instâncias, mas você nunca pagará mais que o preço máximo que especificou. O Amazon EMR facilita o uso de instâncias spot para que você economize tempo e dinheiro. Os clusters Amazon EMR incluem “nós de núcleo” que executam HDFS e “nós de tarefa” que não executam. Os nós de tarefa são ideais para spot, pois se o preço spot aumentar e você perder aquelas instâncias, você não perderá os dados armazenados no HDFS.

Integração de instâncias reservadas do Amazon EC2: as instâncias reservadas do EC2 permitem que você preserve os benefícios da computação elástica ao mesmo tempo em que diminui os custos e reserva a capacidade. Com as instâncias reservadas, você paga uma pequena taxa única e recebe em troca um desconto significativo sobre a cobrança por segundo dessa instância. O Amazon EMR facilita o uso de instâncias reservadas para que você possa economizar até 65% do preço sob demanda.

Elasticidade: como o Amazon EMR facilita escalar automaticamente seu cluster, não será necessário provisionar excesso de capacidade. Por exemplo, você pode não saber quantos clusters de dados estará usando em 6 meses, ou pode ter picos de necessidade de processamento. Com o Amazon EMR, você não precisa adivinhar seus requisitos ou provisões futuras para pico de demanda, pois você pode adicionar/remover capacidade facilmente a qualquer momento.

Integração com Amazon S3: o EMR File System (EMRFS) permite que os clusters do EMR usem o Amazon S3 com eficiência e segurança como um depósito de objetos para o Hadoop. Você pode armazenar seus dados no Amazon S3 e usar vários clusters do Amazon EMR para processar o mesmo conjunto de dados. Cada cluster pode ser otimizado para uma determinada carga de trabalho, que pode ser mais eficiente que um único cluster atendendo a várias cargas de trabalho com requisitos diferentes. Por exemplo, você pode ter um cluster que é otimizado para E/S e outro, otimizado para CPU, cada um processando o mesmo conjunto de dados no Amazon S3. Além disso, armazenando seus dados de entrada e saída no Amazon S3, você pode encerrar clusters quando não forem mais necessários.

O EMRFS oferece uma ótima performance na leitura/gravação de/para o Amazon S3, é compatível com a criptografia do lado do servidor e com a criptografia do lado do cliente do S3 usando o AWS Key Management Service (KMS) ou chaves gerenciadas pelo cliente. Além disso, oferece uma visualização consistente opcional que verifica a consistência da listagem e da leitura após gravação para os objetos rastreados em seus metadados. Além disso, os clusters do Amazon EMR podem usar EMRFS e HDFS para que você não tenha que escolher entre o armazenamento no cluster e o Amazon S3.

Integração ao AWS Glue Data Catalog: é possível usar o catálogo de dados do AWS Glue como um repositório gerenciado de metadados para armazenar metadados de tabela externa do Apache Spark e do Apache Hive. Além disso, ele disponibiliza uma descoberta automática de esquemas, bem como um histórico de versões de esquema. Isso permite persistir metadados de modo fácil para tabelas externas no Amazon S3, fora do cluster.

Datastores flexíveis

Com o Amazon EMR, você pode aproveitar vários datastores, incluindo o Amazon S3, o Hadoop Distributed File System (HDFS) e o Amazon DynamoDB.



Amazon S3: o Amazon S3 é um serviço de armazenamento resiliente, escalável, seguro, rápido e econômico. Com o EMR File System (EMRFS), o Amazon EMR pode usar o Amazon S3 com eficiência e segurança como um depósito de objetos para o Hadoop. O Amazon EMR fez

inúmeras melhorias no Hadoop para que você possa processar grandes quantidades de dados armazenados no Amazon S3. Além disso, o EMRFS pode habilitar a visualização consistente para verificar a consistência de lista e de leitura após gravação para objetos no Amazon S3. O EMRFS oferece suporte à criptografia do lado do servidor e a criptografia do lado do cliente do S3 para processar objetos criptografados do Amazon S3. Você também pode usar o AWS Key Management Service (KMS) ou um fornecedor de chaves personalizadas.

Quando você executa seu cluster, o Amazon EMR transmite os dados do Amazon S3 para cada instâncias do seu cluster e começa a processá-lo imediatamente. Uma vantagem de armazenar seus dados no Amazon S3 e processá-lo com o Amazon EMR é que você pode usar vários clusters para processar os mesmos dados. Por exemplo, você pode ter um cluster de desenvolvimento Hive otimizado para memória e um cluster de desenvolvimento Pig otimizado para CPU, ambos usando o mesmo conjunto de dados de entrada.

Hadoop Distributed File System (HDFS): o HDFS é o sistema de arquivos do Hadoop. A topologia atual do Amazon EMR agrupa suas instâncias em 3 grupos lógicos de instâncias: grupo mestre, que executa o YARN Resource Manager e o serviço HDFS Name Node; grupo principal, que executa o HDFS DataNode Daemon e o serviço YARN Node Manager, e grupo de tarefas, que executa o serviço YARN Node Manager. O Amazon EMR instala o HDFS no armazenamento associado às instâncias do grupo principal.

Cada instância do EC2 é fornecida com uma quantidade fixa de armazenamento, denominada “armazenamento de instâncias”, anexada à instância. Também é possível personalizar o armazenamento em uma instância adicionando volumes do Amazon EBS à instância. O Amazon EMR permite adicionar os tipos de volume de uso geral (SSD), provisionado (SSD) e magnético. Os volumes do EBS adicionados a um cluster do EMR não persistem dados após o encerramento do cluster. O EMR limpa automaticamente os volumes após o encerramento do cluster.

Também é possível habilitar a criptografia completa do HDFS usando uma configuração de segurança do Amazon EMR ou criar manualmente zonas de criptografia do HDFS com o Hadoop Key Management Server.

Amazon DynamoDB: o Amazon DynamoDB é um serviço de banco de dados NoSQL rápido e gerenciado. O Amazon EMR tem uma integração direta com o Amazon DynamoDB, assim você pode, com rapidez e eficiência, processar dados armazenados no Amazon DynamoDB e transferir dados entre o Amazon DynamoDB, Amazon S3 e HDFS no Amazon EMR.

Outros datastores da AWS: os clientes do Amazon EMR também usam o Amazon Relational Database Service (um web service que facilita a configuração, a operação e o dimensionamento de um banco de dados relacional na nuvem), o Amazon Glacier (um serviço de armazenamento de custo extremamente baixo que oferece armazenamento seguro e resiliente para arquivamento e backup de dados) e o Amazon Redshift (um serviço de data warehouse rápido e gerenciado em escala de petabytes). O AWS Data Pipeline é um web service que ajuda os clientes a processar e movimentar dados de forma confiável entre diferentes serviços de armazenamento e computação da AWS (incluindo o Amazon EMR), além de fontes de dados locais, em intervalos especificados.

Use seus aplicativos de código aberto preferidos

Com os lançamentos com controle de versão no Amazon EMR, você pode facilmente selecionar e usar os projetos de código aberto mais recentes no cluster do EMR, como aplicativos nos ecossistemas do Apache Spark e Hadoop. O software é instalado e configurado pelo Amazon EMR para que você possa passar mais tempo agregando valor aos seus dados sem se preocupar com tarefas administrativas e relacionadas à infraestrutura.

Ferramentas do Hadoop

O Amazon EMR é compatível com ferramentas poderosas e comprovadas do Hadoop, como o Apache Spark, o Apache Hive e o Apache HBase. Além disso, ele pode executar ferramentas de deep learning e de machine learning, como o TensorFlow, o Apache MXNet, e usando ações de bootstrap você pode adicionar suas próprias ferramentas e bibliotecas específicas de casos de uso. Para o desenvolvimento interativo, o Hue e os Notebooks EMR podem ser usados para criar tarefas do Apache Spark e enviar consultas SQL para o Apache Hive e o Presto.

Popular Hadoop Applications



Processamento de dados e Machine Learning

Apache Spark é um mecanismo no ecossistema do Hadoop que processa rapidamente grandes conjuntos de dados. Ele usa conjuntos de dados distribuídos resilientes (RDDs) na memória e tolerante a falhas e gráficos direcionados acíclicos (DAGs) para definir transformações de dados. O Spark também inclui Spark SQL, Spark Streaming, MLlib e GraphX.

Apache Flink é um mecanismo de fluxo de dados de streaming que facilita a execução do processamento de streaming em tempo real em fontes de dados com alto throughput. Ele é compatível com semântica de tempo otimizada para eventos fora de ordem, semântica do tipo exactly-once (exatamente uma vez), controle de pressão de retorno e APIs otimizados para escrever aplicativos de streaming e em lote.

TensorFlow é uma biblioteca de matemática simbólica de código aberto para aplicativos de inteligência de máquina e aprendizagem profunda. O TensorFlow reúne vários modelos e algoritmos de machine learning e deep learning, além de treinar e executar redes neurais profundas para muitos casos de uso diferentes.

SQL

Apache Hive é um data warehouse e um pacote analítico de código aberto que é executado com base no Hadoop. O Hive é operado pela Hive QL, uma linguagem baseada em SQL, que permite aos usuários estruturar, resumir e consultar dados. O Hive QL vai além do SQL padrão, adicionando suporte de primeira classe às funções mapear/reduzir e a tipos de dados complexos e extensíveis definidos pelo usuário, como JSON e Thrift. Esse recurso permite o processamento de fontes de dados complexas e até não estruturadas, como documentos de texto e arquivos de log. O Hive permite extensões de usuário via funções definidas pelo usuário escritas em Java. O Amazon EMR tem feito inúmeras melhorias ao Hive, incluindo a integração direta com o Amazon

DynamoDB e o Amazon S3. Por exemplo, com o Amazon EMR, você pode carregar partições de tabela automaticamente do Amazon S3, gravar dados em tabelas no Amazon S3 sem usar arquivos temporários e pode acessar recursos no Amazon S3, como roteiros para operações personalizadas de mapeamento/redução e bibliotecas adicionais.

Presto é um mecanismo de consulta SQL distribuído de código aberto otimizado para baixa latência e análise de dados ad-hoc. Ele aceita o padrão ANSI SQL, que inclui consultas complexas, agregações, junções e funções de janela. O Presto pode processar dados de várias fontes, como o Hadoop Distributed File System (HDFS) e o Amazon S3.

Apache Phoenix permite o uso de SQL de baixa latência com recursos de transação ACID em dados armazenados no Apache HBase. É possível criar facilmente índices secundários para obter performance adicional, como também visualizações diferentes sobre a mesma tabela subjacente do HBase.

NoSQL

Apache HBase é um banco de dados de código aberto, não relacional e distribuído, modelado de acordo com o BigTable da Google. Foi desenvolvido como parte do projeto Hadoop da Apache Software Foundation e é executado com base no Hadoop Distributed File System (HDFS) para fornecer recursos similares aos da BigTable para Hadoop. O HBase disponibiliza uma maneira eficiente e tolerante a falhas de armazenar grandes quantidades de dados esparsos usando compactação e armazenamento baseado em colunas. Além disso, o HBase disponibiliza pesquisa rápida de dados porque armazena em cache dados de memória. O HBase é otimizado para operações de gravação sequencial, e é altamente eficiente para inserções, atualizações e exclusões em lote. O HBase funciona sem dificuldade com o Hadoop, compartilhando seu sistema de arquivos e servindo como entrada e saída direta para os trabalhos do Hadoop. O HBase também se integra ao Apache Hive, possibilitando consultas tipo SQL em tabelas HBase, junções com tabelas baseadas no Hive e suporte para Java Database Connectivity (JDBC). Com o EMR, você pode usar o S3 como um datastore do HBase, o que permite diminuir custos e reduzir complexidade operacional. Se você usa o HDFS como um datastore, é possível fazer backup do HBase no S3, além de restaurar usando um backup criado anteriormente.

Análise interativa

Hue é uma interface de usuário de código aberto para o Hadoop que facilita a execução e o desenvolvimento de consultas do Hive, o gerenciamento de arquivos no HDFS, a execução e o desenvolvimento de scripts Pig e o gerenciamento de tabelas. O Hue no EMR também se integra ao Amazon S3, permitindo executar consultas diretas no S3 e facilitando a transferência de arquivos entre o HDFS e o Amazon S3.

Notebooks EMR são baseados no projeto Jupyter de código aberto e são pré-configurados para o Spark. Eles oferecem suporte aos kernels mágicos do Spark, o que permite executar interativamente tarefas do Spark em clusters do EMR escritas em linguagens como PySpark, Spark SQL, Spark R e Scala. Os blocos de anotações são fornecidos com bibliotecas de código aberto encontradas no Conda, permitindo que você importe essas bibliotecas e as use para manipular dados e visualizar resultados computacionais gráficos sofisticados. Além disso, cada bloco de anotações incorpora recursos de monitoramento do Spark que permitem monitorar o progresso das tarefas e depurar código diretamente do bloco de anotações.

Notebook Jupyter é um aplicativo web de código aberto que você pode usar para criar e compartilhar documentos que contenham código ativo, equações, visualizações e texto narrativo. O JupyterHub permite hospedar várias instâncias de um servidor de Notebook Jupyter de usuário único. Ao criar um cluster EMR com o JupyterHub, o EMR cria um contêiner do Docker em um nó principal do cluster. O JupyterHub, todos os componentes necessários do Jupyter e o Sparkmagic são executados no contêiner.

Apache Zeppelin é uma GUI de código aberto que cria blocos de anotações interativos e de colaboração para a exploração de dados usando o Spark. Você pode usar Scala, Python, SQL (usando Spark SQL) ou HiveQL para manipular dados e visualizar rapidamente resultados. Os blocos de anotações do Zeppelin podem ser compartilhados entre vários usuários, e as visualizações podem ser publicadas em painéis externos.

Programação e fluxo de trabalho

Apache Oozie é um programador de fluxo de trabalho do Hadoop, no qual você pode criar Directed Acyclic Graphs (DAGs) das atividades. Você também pode acionar facilmente os fluxos de trabalho do Hadoop por atividades ou horário.

Outros projetos e ferramentas

O EMR também oferece suporte a vários outros aplicativos e ferramentas populares, como R, Apache Pig (processamento de dados e ETL), Apache Tez (execução DAG complexa), Apache MXNet (aprendizagem profunda), Apache Mahout (Machine Learning), Ganglia (monitoramento), Apache Accumulo (banco de dados NoSQL seguro), Apache Sqoop (conector de banco de dados relacional), HCatalog (gerenciamento de tabelas e armazenamento), entre outros. A equipe do Amazon EMR mantém um repositório de ações de bootstrap de código aberto que pode ser usado para instalar software adicional, configurar clusters ou servir como exemplos de codificação de suas próprias ações de bootstrap.

Recursos adicionais

Ajuste seu cluster: você escolhe quais tipos de instâncias do EC2 provisionar no seu cluster (padrão, com mais memória, com CPU de alta performance, com E/S de alta performance, etc.) de acordo com os requisitos dos aplicativos. Você tem acesso à raiz de cada instância e pode personalizar totalmente seu cluster para que se ajuste aos seus requisitos.

Depure seus aplicativos: quando você habilita a depuração em um cluster, o Amazon EMR guarda os arquivos de log no Amazon S3 e indexa esses arquivos. Em seguida, você pode usar a interface gráfica no console para percorrer os logs e visualizar o histórico de trabalhos de forma intuitiva.

Monitore seu cluster: você pode usar o Amazon CloudWatch para monitorar 23 métricas personalizadas do Amazon EMR, como o número médio de tarefas de mapeamento e redução em execução. Você também pode definir os alarmes nessas métricas.

Responder a eventos: você pode usar os tipos de evento do Amazon EMR no Amazon CloudWatch Events para responder às alterações de estado em seus clusters do Amazon EMR. Usando regras simples que você pode configurar rapidamente, combine eventos e encaminhe-os para tópicos do Amazon SNS, funções do AWS Lambda, filas do Amazon SQS, entre outros.

Programe fluxos de trabalho recorrentes: você pode usar o AWS Data Pipeline para programar fluxos de trabalho recorrentes envolvendo o Amazon EMR. O AWS Data Pipeline é um web service que ajuda você a processar e movimentar dados de forma confiável entre diferentes serviços de armazenamento e computação da AWS, inclusive fontes de dados locais, em intervalos especificados.

Cascading: o Cascading é uma biblioteca Java de código aberto que oferece uma API de consultas, um planejador de consultas e um programador de trabalhos para criar e executar aplicativos Hadoop MapReduce. Os aplicativos desenvolvidos com o Cascading são compilados e empacotados em arquivo JAR compatíveis com o Hadoop, similares a outros aplicativos Hadoop nativos.

Aprendizado profundo: use estruturas conhecidas de aprendizado profundo como Apache MXNet para definir, treinar e implantar redes neurais profundas. Você pode usar essas estruturas em clusters do Amazon EMR com instâncias de GPU.

Controle o acesso de rede ao seu cluster: você pode executar seu cluster em uma Amazon Virtual Private Cloud (VPC), uma seção logicamente isolada da Nuvem AWS. Você tem controle total sobre seu ambiente de redes virtuais, incluindo a seleção do seu próprio intervalo de endereços IP, a criação de sub-redes e a configuração de tabelas de rotas e gateways de rede.

Gerencie usuários, permissões e criptografia: você pode usar ferramentas do AWS Identity and Access Management (IAM), como os usuários e as funções do IAM, para controlar acesso e permissões. Por exemplo, você pode conceder a determinados usuários acesso de leitura, mas não de gravação, aos seus clusters. Além disso, você pode usar as configurações de segurança do Amazon EMR para definir várias opções de criptografia de dados ociosos e em trânsito, incluindo o suporte à criptografia do Amazon S3 e à autenticação do Kerberos.

Instale software adicional: você pode usar ações de bootstrap ou uma Amazon Machine Image (AMI) personalizada executando Amazon Linux para instalar software adicional no cluster. As ações de bootstrap são roteiros que são executados nos nós do cluster quando o Amazon EMR executa o cluster. Elas são executadas antes que o Hadoop inicie e antes que o nó comece a processar dados. Também é possível pré-carregar e usar software em uma AMI personalizada do Amazon Linux.

Copie dados com eficiência: você pode mover rapidamente grandes quantidades de dados do Amazon S3 para HDFS, do HDFS para Amazon S3 e entre buckets do Amazon S3 usando o S3DistCp do Amazon EMR, uma extensão da ferramenta de código aberto Distcp, que usa o MapReduce para mover com eficiência grandes quantidades de dados.

Hadoop Streaming: o Hadoop Streaming é um utilitário que acompanha o Hadoop e permite que você desenvolva executáveis do MapReduce em linguagens diferentes do Java. O Streaming é implementado na forma de um arquivo JAR.

JAR personalizado: escreva um programa Java, faça a compilação com a versão do Hadoop que pretende usar e faça upload no Amazon S3. Você pode, então, enviar trabalhos do Hadoop para o cluster usando a interface JobClient do Hadoop.

Exercícios:

- 1) Assistir ao curso de Data Analytics Fundamentals para consolidar os conhecimentos:
<https://www.aws.training/Details/eLearning?id=35364>
- 2) Ver o vídeo: Introduction to Amazon Elastic MapReduce (EMR)
<https://www.aws.training/Details/Video?id=16023>
- 3) Getting Started with Amazon EMR -
<https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-gs.html>

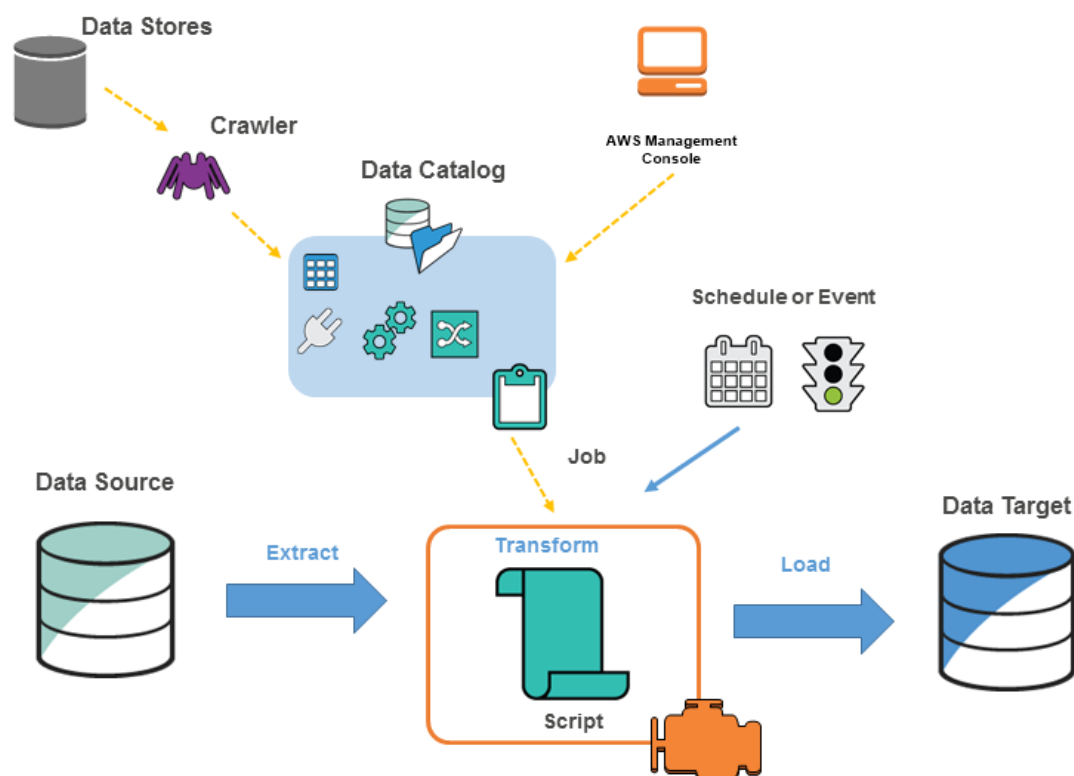
Para executar esse exercício, usaremos dados de entrada oferecidos pela AWS que são uma versão modificada de um conjunto de dados de inspeção de estabelecimentos de alimentos disponíveis ao público com resultados de inspeção do Departamento de Saúde em King County, Washington, de 2006 a 2020.

AWS Glue

AWS Glue é um serviço de integração de dados serverless que facilita descobrir, preparar e combinar dados para análise, machine learning e desenvolvimento da aplicação. Ele oferece todos os recursos necessários para a integração dos dados, portanto é possível começar a analisar seus dados e usá-los em minutos, ao invés de meses.

O AWS Glue proporciona interfaces visuais e baseadas em código para facilitar a preparação dos dados. Os usuários podem encontrar e acessar facilmente os dados usando o catálogo de dados do AWS Glue. Engenheiros de dados e desenvolvedores de ETL (extrair, transformar e carregar) podem criar, executar e monitorar visualmente fluxos de trabalho ETL com apenas alguns cliques no AWS Glue Studio. Analistas e cientistas de dados podem usar o AWS Glue DataBrew para enriquecer, limpar e normalizar visualmente os dados sem escrever código. Com o AWS Glue Elastic Views, os desenvolvedores de aplicação podem usar um SQL (Structured Query Language) familiar para combinar e replicar os dados em diferentes armazenamentos de dados.

O objetivo principal do Glue é ser uma solução gerenciada para ETL. Mas além de ETL o AWS Glue possui um repositório central de metadados conhecido como AWS Glue Data Catalog. Vamos começar entendendo o diagrama de arquitetura do Glue



Você define JOBS no AWS Glue para realizar o trabalho necessário para extrair, transformar e carregar (ETL) dados de uma fonte de dados (Data Source) para um destino de dados (Data Target). Normalmente, você executa as seguintes ações:

- Para fontes de armazenamento de dados (data store sources), você define um *crawler* para preencher seu AWS Glue Data Catalog com definições de metadados de tabela (nome da tabela, nome dos campos, tipo dos campos, entre outros). Você aponta seu *crawler* para um armazenamento de dados (data store), e o *crawler* cria definições de tabela no Catálogo de Dados (Data Catalog). Para fontes de streaming (streaming sources), você define manualmente as tabelas do Catálogo de Dados e especifica as propriedades do fluxo de dados. Além das definições de tabela, o AWS Glue Data Catalog contém outros metadados que são necessários para definir JOBS ETL. Você usa esses metadados quando define um JOB para transformar (Transform) seus dados.
- O AWS Glue pode gerar um script para transformar seus dados. Ou você pode providenciar o script no console AWS Glue ou através de APIs.
- Você pode executar seu JOB sob demanda ou pode configurá-lo para iniciar quando um determinado gatilho (*trigger*) ocorrer. O gatilho pode ser uma programação baseada em tempo ou um evento (time-based schedule or an event).

Quando seu JOB é executado, um script extrai dados de sua fonte de dados (data source), transforma os dados e os carrega em seu destino de dados (data target). O script é executado em um ambiente Apache Spark no AWS Glue.

Terminologia do AWS Glue

O AWS Glue depende da interação de vários componentes para criar e gerenciar seu workflow de extração, transferência e carregamento (ETL).

AWS Glue Data Catalog: O armazenamento de metadados persistente no AWS Glue. Ele contém definições de tabela, definições de trabalho e outras informações de controle para gerenciar seu ambiente AWS Glue. Cada conta da AWS tem um catálogo de dados do AWS Glue por região. Mais tarde estudaremos mais sobre este catálogo de metadados

Classificador (Classifier): Determina o esquema de seus dados. O AWS Glue fornece classificadores para tipos de arquivo comuns, como CSV, JSON, AVRO, XML e outros. Ele também fornece classificadores para sistemas de gerenciamento de banco de dados relacional comuns usando uma conexão JDBC. Você pode escrever seu próprio classificador usando um pattern grok ou especificando uma tag de linha em um documento XML.

Conexão (Connection): Um objeto Catálogo de Dados que contém as propriedades necessárias para se conectar a um armazenamento de dados específico.

Crawler: Um programa que se conecta a um armazenamento de dados (origem ou destino), avança por meio de uma lista priorizada de classificadores para determinar o esquema para seus dados e, em seguida, cria tabelas de metadados no Catálogo de Dados AWS Glue.

Banco de dados(Database): Um conjunto de definições de tabela do Catálogo de Dados associadas, organizado em um grupo lógico.

Armazenamento de dados, origem de dados, destino de dados (Data store, data source, data target): Um armazenamento de dados é um repositório para armazenar persistentemente seus dados. Os exemplos incluem buckets do Amazon S3 e bancos de dados relacionais. Uma origem de dados é um armazenamento de dados usado como entrada para um processo ou

transformação. Um destino de dados é um armazenamento de dados no qual um processo ou transformação grava.

Development endpoint: Um ambiente que você pode usar para desenvolver e testar seus scripts AWS Glue ETL.

Dynamic Frame: Uma tabela distribuída que oferece suporte a dados aninhados, como structures e arrays. Cada registro é auto descritivo, projetado para flexibilidade de esquema com dados semiestruturados. Cada registro contém dados e o esquema que descreve esses dados. Você pode usar Dynamic Frame e Apache Spark DataFrames em seus scripts ETL e converter entre eles. Os Dynamic Frames fornecem um conjunto de transformações avançadas para limpeza de dados e ETL.

Tarefa (Job): A lógica de negócios necessária para realizar o trabalho ETL. É composto de um script de transformação, origens de dados e destinos de dados. As execuções de tarefas são iniciadas por gatilhos que podem ser programados ou disparados por eventos.

Notebook server: Um ambiente baseado na web que você pode usar para executar suas instruções PySpark. PySpark é um dialeto Python para programação ETL. Para obter mais informações, consulte Apache Zeppelin (<http://zeppelin.apache.org/>). Você pode configurar um notebook server em um endpoint de desenvolvimento para executar instruções PySpark com extensões AWS Glue.

Script: Código que extrai dados de origens, os transforma e os carrega em destinos. O AWS Glue gera scripts PySpark ou Scala.

Tabela (Table): A definição de metadados que representa seus dados. Esteja seus dados em um arquivo Amazon Simple Storage Service (Amazon S3), uma tabela Amazon Relational Database Service (Amazon RDS) ou outro conjunto de dados, uma tabela define o esquema ou estrutura de seus dados. Uma tabela no AWS Glue Data Catalog consiste em nomes de colunas, definições de tipo de dados, informações de partição e outros metadados sobre um conjunto de dados base. O esquema dos seus dados é representado na definição da tabela AWS Glue. Os dados reais permanecem em seu armazenamento de dados original, seja em um arquivo ou em uma tabela de banco de dados relacional. O AWS Glue cataloga seus arquivos e tabelas de banco de dados relacional no AWS Glue Data Catalog. Eles são usados como origens e destinos quando você cria um trabalho ETL.

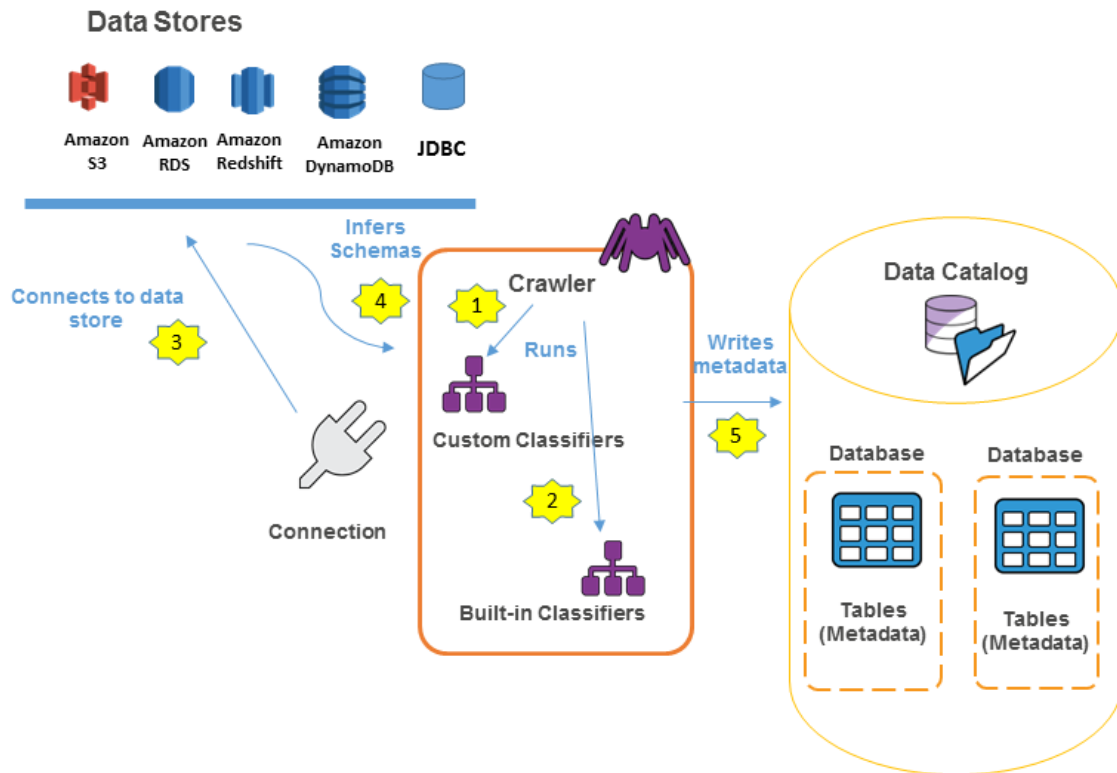
Transform: A lógica do código usada para manipular seus dados em um formato diferente.

Gatilho (Trigger): Inicia um trabalho ETL. Os gatilhos podem ser definidos com base em um horário programado ou evento.

Carregando o Glue Data Catalog

O AWS Glue Data Catalog contém as referências a dados que são usados como origens e destinos de seus trabalhos de extração, transformação e carregamento (ETL) no AWS Glue. Para criar seu data warehouse ou data lake, você deve catalogar esses dados. O AWS Glue Data Catalog é um índice para a localização, esquema e métricas de tempo de execução de seus dados. Você usa as informações no Catálogo de Dados para criar e monitorar seus Jobs ETL. As informações no Catálogo de Dados são armazenadas como tabelas de metadados, onde cada tabela especifica um único armazenamento de dados. Normalmente, você executa

um crawler para fazer o inventário dos dados em seus armazenamentos de dados, mas existem outras maneiras de adicionar tabelas de metadados em seu Catálogo de Dados. O diagrama de fluxo de trabalho a seguir mostra como os crawlers do AWS Glue interagem com armazenamentos de dados e outros elementos para preencher o Catálogo de Dados



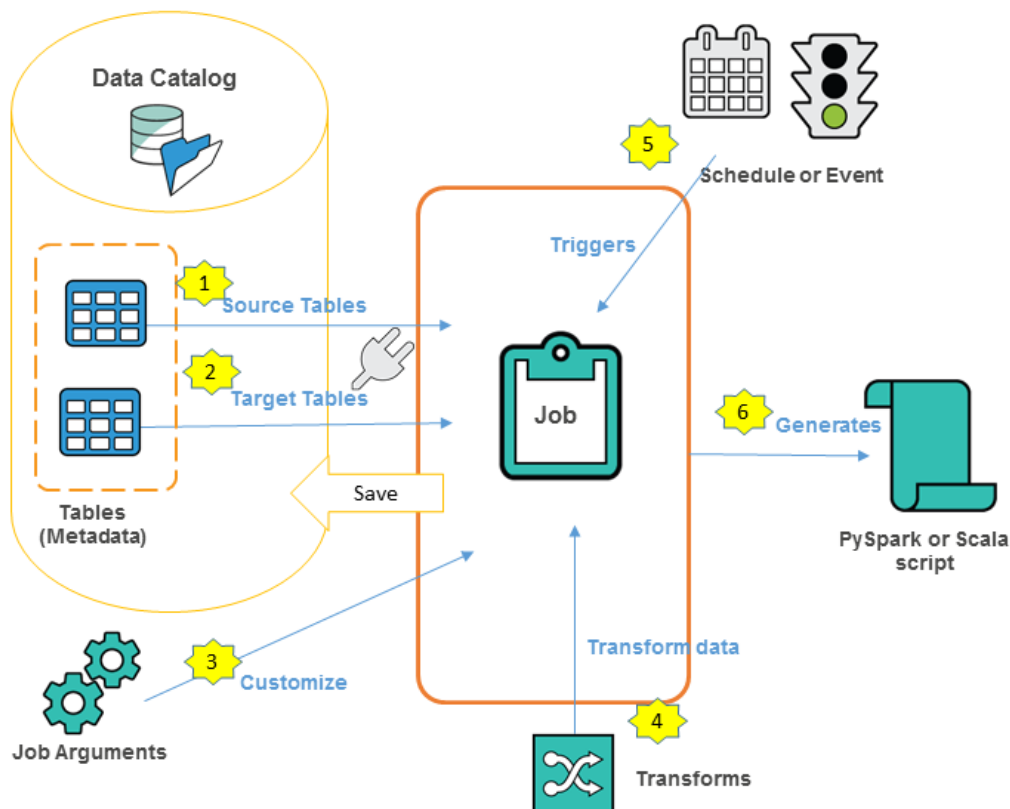
- 1) Um crawler executa quaisquer classificadores (classifiers) personalizados que você escolher para inferir o formato e o esquema de seus dados. Você fornece o código para classificadores personalizados e eles são executados na ordem que você especificar. O primeiro classificador personalizado a reconhecer com sucesso a estrutura de seus dados é usado para criar um esquema.
- 2) Se nenhum classificador personalizado corresponder ao esquema de seus dados, os classificadores integrados tentam reconhecer o esquema de seus dados. Um exemplo de classificador integrado é aquele que reconhece JSON.
- 3) O crawler conecta-se ao armazenamento de dados. Alguns armazenamentos de dados requerem propriedades de conexão para acesso do crawler.
- 4) O esquema dos seus dados é criado
- 5) O crawler grava metadados no Catálogo de Dados

Os bancos de dados são usados para organizar tabelas de metadados no AWS Glue. Ao definir uma tabela no Catálogo de dados do AWS Glue, você a adiciona a um banco de dados. Uma tabela pode estar em apenas um banco de dados. Seu banco de dados pode conter tabelas que definem dados de muitos armazenamentos de dados diferentes. Esses dados podem incluir objetos no Amazon Simple Storage Service (Amazon S3) e tabelas relacionais no Amazon Relational Database Service.

Agora que já sabemos o que é e como carregamos o Catálogo do Glue, estamos prontos para extrair, transformar e carregar os dados com Glue ETL.

Glue ETL

Um job (trabalho) é a lógica de negócios que executa o trabalho de extração, transformação e carregamento (ETL) no AWS Glue. Quando você inicia um job, o AWS Glue executa um script que extrai dados de fontes de dados de origem, transforma os dados e os carrega em fontes de dados de destinos. Você pode criar trabalhos na seção de ETL do console do AWS Glue. O diagrama a seguir resume o fluxo de trabalho básico e as etapas envolvidas na criação de um trabalho no AWS Glue:



Visão geral do fluxo de trabalho

Ao criar um job, você fornece detalhes sobre origens e destino dos dados e outras informações. O resultado é a geração de um script da Apache Spark API (PySpark). Em seguida, você poderá armazenar sua definição de job no Catálogo de dados do AWS Glue. Abaixo uma descrição do processo geral de criação de jobs no console do AWS Glue:

- 1) Você escolhe uma fonte de dados de origem para o seu job. As tabelas que representam sua fonte de dados já devem estar definidas no Glue Data Catalog. Se a fonte exigir uma conexão, ela também será referenciada no seu job. Se o job precisar de várias fontes de dados, você poderá adicioná-las posteriormente editando o script.
- 2) Você escolhe um destino de dados para o seu job. As tabelas que representam o destino de dados podem ser definidas no Glue Data Catalog ou seu job poderá criar as tabelas de destino quando for executado. Você escolhe um local de destino ao criar o

job. Se o destino exigir uma conexão, ela também será referenciada no seu job. Se o job precisar de vários destinos de dados, você poderá adicioná-los posteriormente editando o script.

- 3) Você personaliza o ambiente de processamento de job informando argumentos para seu job e o script gerado.
- 4) Inicialmente, o AWS Glue gera um script. No entanto, você pode editar esse script para adicionar fontes de origem e destinos e transformações.
- 5) Você especifica como seu job é invocado/disparado: sob demanda, programação baseada em tempo ou evento.
- 6) Com base nos dados informados, o AWS Glue gera um script PySpark ou Scala. Você pode personalizar o script com base nas suas necessidades

Você pode usar scripts gerados pelo AWS Glue ou pode criar seus próprios scripts “na mão” e adicioná-los a um job. Dados um esquema de origem e um local ou esquema de destino, o gerador de código do AWS Glue pode criar automaticamente um script Apache Spark API (PySpark). Você pode usar esse script como ponto de partida e editá-lo para atingir seus objetivos. O AWS Glue pode escrever arquivos de saída em vários formatos de dados, incluindo JSON, CSV, ORC (Optimized Row Columnar), Apache Parquet e Apache Avro. Para alguns formatos de dados, é possível gravar formatos de compressão comuns.

Existem três tipos de jobs no AWS Glue: Spark, Streaming ETL e shell Python.

- Um trabalho do Spark é executado em um ambiente Apache Spark gerenciado pelo AWS Glue. Ele processa os dados em lotes/batch.
- Um job de ETL de streaming é semelhante a um job do Spark, exceto que ele executa ETL em streams de dados. Ele usa a estrutura de job Apache Spark Structured Streaming. Alguns recursos de job do Spark não estão disponíveis para jobs ETL de streaming.
- Um job de shell do Python executa scripts do Python como um shell e oferece suporte a uma versão do Python que depende da versão do AWS Glue que você está usando. É possível usar esses jobs para programar e executar tarefas que não exigem um ambiente do Apache Spark.

AWS Glue versão	Versões compatíveis do Spark e do Python
0.9	Spark 2.2.1 Python 2.7
1.0	Spark 2.4.3 Python 2.7 Python 3.6
2.0	Spark 2.4.3 Python 3.7

Mais informações sobre parâmetros de Jobs:

https://docs.aws.amazon.com/pt_br/glue/latest/dg/add-job.html

Na lista Jobs, você pode fazer o seguinte:

- Para iniciar um job existente, escolha Action e, em seguida, Run job.
- Para interromper um job Running ou Starting, escolha Action e, em seguida, Stop job run.

- Para adicionar gatilhos que iniciam um job, escolha Action, Choose job triggers.
- Para modificar um job existente, escolha Action e, em seguida, Edit job ou Delete.
- Para alterar um script associado a um job, escolha Action, Edit script.
- Para redefinir as informações de estado que o AWS Glue armazena sobre o seu job, escolha Action (Ação), Reset job bookmark (Redefinir marcador de job).
- Para criar um endpoint de desenvolvimento com as propriedades deste job, escolha Action, Create development endpoint.

Gatilhos do AWS Glue

No AWS Glue, você pode criar objetos do Data Catalog chamados de gatilhos, que podem ser usados para iniciar manual ou automaticamente um ou mais crawlers ou jobs de extrair, transformar e carregar (ETL). Ao usar gatilhos, você pode projetar uma estrutura de jobs e crawlers dependentes.

Como adicionar um gatilho (console)

- 1) Faça login no Console de gerenciamento da AWS e abra o console do AWS Glue em <https://console.aws.amazon.com/glue/>
- 2) No painel de navegação, em ETL, selecione Triggers (Gatilhos). Escolha Add trigger (Adicionar gatilhos).
- 3) Forneça as seguintes propriedades:
 - a. Name: Atribua um nome exclusivo ao gatilho.
 - b. Tipos de gatilho: Especifique um dos seguintes:
 - i. **Schedule:** O gatilho ativa-se a uma frequência e hora específicas.
 - ii. **Job events:** Um gatilho condicional. O gatilho é acionado quando um ou todos os trabalhos na lista correspondem ao seu status designado. Para que o gatilho seja acionado, os trabalhos monitorados devem ter sido iniciados por um gatilho. Para qualquer trabalho que você escolher, só é possível observar um evento de trabalho (status de conclusão).
 - iii. **On-demand:** O gatilho dispara quando é ativado.
- 4) Executar o assistente de gatilho. Na página Review (Revisar) você pode ativar gatilhos Schedule (Programados) e de Job events (Eventos de trabalho) (condicionais) imediatamente selecionando Enable trigger on creation (Habilitar gatilho na criação).

Para ativar ou desativar um gatilho:

- 1) No painel de navegação, em ETL, selecione Triggers (Gatilhos).
- 2) Marque a caixa de seleção ao lado do gatilho desejado e, no menu Action (Ação), escolha Enable trigger (Habilitar gatilho) para ativar o gatilho ou Disable trigger (Desabilitar gatilho) para desativar o gatilho.

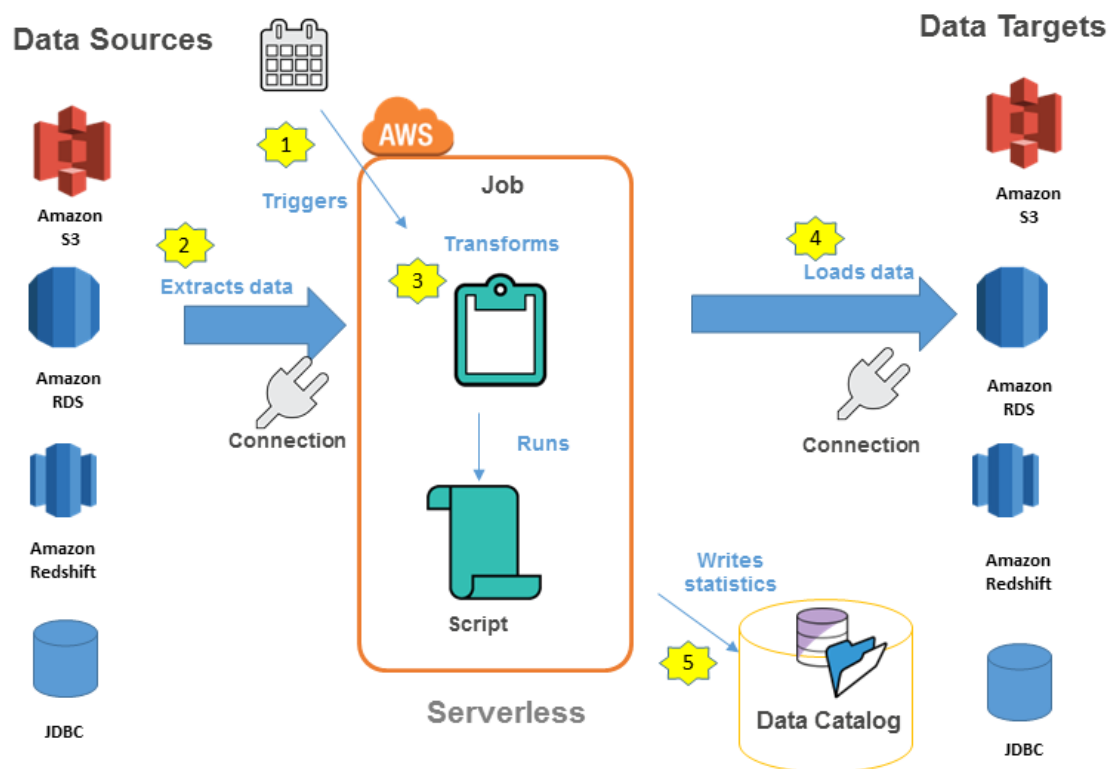
Executar e monitorar o AWS Glue

Você pode automatizar a execução dos seus jobs de ETL. O AWS Glue também fornece métricas para crawlers e jobs que você pode monitorar. Depois de configurar o Catálogo de dados do AWS Glue com os metadados necessários, o AWS Glue fornecerá estatísticas sobre a integridade do seu ambiente. Você pode automatizar a invocação de crawlers e jobs com uma

programação de cron baseada em hora. Você também pode ativar jobs quando um gatilho baseado em eventos é acionado.

O principal objetivo do AWS Glue é fornecer uma maneira mais fácil de extrair e transformar seus dados de origem em dados de destino. Para atingir esse objetivo, um trabalho de ETL segue estas etapas típicas (exibidas no diagrama a seguir):

- 1) Um gatilho é acionado para iniciar uma execução de job. Este evento pode ser configurado em uma programação recorrente ou satisfazer uma dependência.
- 2) O job extrai os dados da sua fonte. Se necessário, as propriedades de conexão serão usadas para acessar sua fonte.
- 3) O job transforma seus dados usando um script que você criou e os valores de quaisquer argumentos. O script contém o código Scala ou PySpark Python que transforma seus dados.
- 4) Os dados transformados são carregados nos seus destinos de dados. Se necessário, as propriedades de conexão serão usadas para acessar o destino.
- 5) As estatísticas sobre a execução do job são coletadas e gravadas no seu Data Catalog.



AWS Glue ETL – notebook

O AWS Glue pode criar um ambiente, conhecido como endpoint de desenvolvimento, que você pode usar para desenvolver e testar iterativamente seus scripts de extração, transformação e carregamento (ETL). Você pode criar, editar e excluir endpoints de desenvolvimento usando o console ou a API do AWS Glue.

Ao criar um endpoint de desenvolvimento, você fornece valores de configuração para provisionar o ambiente de desenvolvimento. Esses valores informam ao AWS Glue como configurar a rede para que você possa acessar o endpoint de forma segura e o endpoint possa acessar seus armazenamentos de dados.

Depois, é possível criar um notebook que se conecte ao endpoint e usá-lo para criar e testar seu script de ETL. Quando estiver satisfeito com os resultados do seu processo de desenvolvimento, você poderá criar um job de ETL que execute seu script. Com esse processo, você pode adicionar funções e depurar seus scripts de forma interativa.

Para usar um endpoint de desenvolvimento do AWS Glue, você pode seguir este fluxo de trabalho:

- 1) Crie um endpoint de desenvolvimento usando o console ou a API. O endpoint é executado em uma nuvem privada virtual (VPC) com seus grupos de segurança definidos.
- 2) O console ou a API sonda o endpoint de desenvolvimento até que ele seja provisionado e esteja pronto para o trabalho. Quando estiver pronto, conecte-se ao endpoint de desenvolvimento usando um dos seguintes métodos para criar e testar scripts do AWS Glue:
 - a. Instale um notebook Apache Zeppelin na sua máquina local, conecte-o a um endpoint de desenvolvimento e, depois, desenvolva com base nele usando seu navegador.
 - b. Crie um servidor de notebook Zeppelin na sua própria instância do Amazon EC2 (na sua conta) usando o console do AWS Glue e, depois, conecte-se a ele usando seu navegador.
 - c. Você pode criar um notebook SageMaker em sua conta usando o console do AWS Glue.
 - d. Abra uma janela de terminal para se conectar diretamente a um endpoint de desenvolvimento.
 - e. Se você tiver a edição profissional do JetBrains Python IDE do PyCharm, conecte-a a um endpoint de desenvolvimento e use-a para desenvolver de forma interativa. Se você inserir instruções pydevd no script, o PyCharm poderá oferecer suporte a pontos de interrupção remotos.
- 3) Ao concluir a depuração e o teste no seu endpoint de desenvolvimento, você poderá excluí-lo.

Aqui vamos usar o método (a). Siga o Tutorial abaixo para realizar a configuração de seu ambiente de desenvolvimento:

Tutorial: configurar um notebook Apache Zeppelin local para testar e depurar scripts de ETL:
https://docs.aws.amazon.com/pt_br/glue/latest/dg/dev-endpoint-tutorial-local-notebook.html

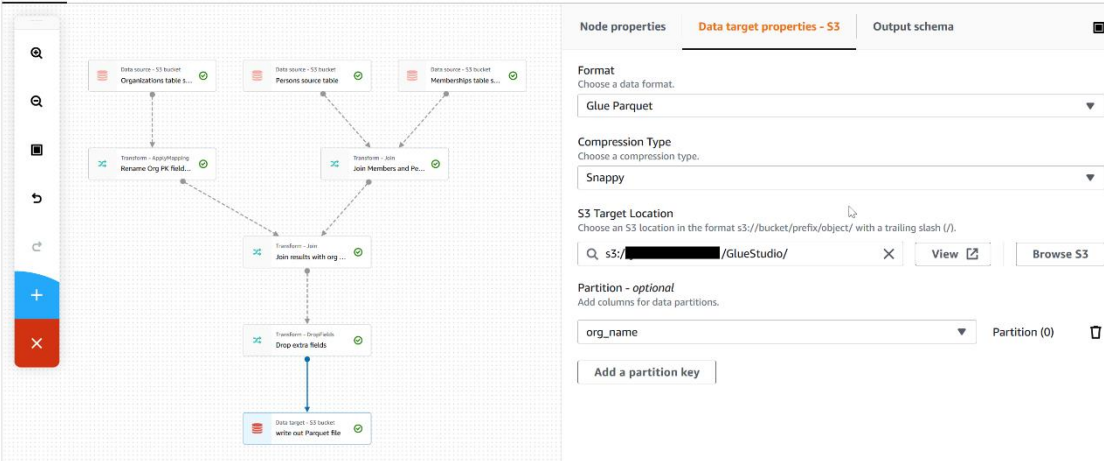
AWS Glue Studio

O AWS Glue Studio é uma nova interface gráfica que facilita a criação, a execução e o monitoramento de trabalhos de extração, transformação e carregamento (ETL) no AWS Glue. Você pode compor visualmente fluxos de trabalho de transformação de dados e executá-los perfeitamente no mecanismo AWS Glue de ETL sem servidor baseado em Apache Spark.

Combine legislators data

Last Saved at 10/8/2020, 4:07:35 PM Save Run

Visual Script Job details Run details



Node properties Data target properties - S3 Output schema

Format
Choose a data format.
Glue Parquet

Compression Type
Choose a compression type.
Snappy

S3 Target Location
Choose an S3 location in the format s3://bucket/prefix/object/ with a trailing slash (/).
s3://[redacted]/GlueStudio/ View Browse S3

Partition - optional
Add columns for data partitions.
org_name Partition (0) +

Add a partition key

Exercícios:

- 4) AWS Glue Tutorial | Getting Started with AWS Glue ETL - <https://www.youtube.com/watch?v=Qpv7BzOM-Ul> Crie exatamente o que o instrutor está sugerindo no vídeo em um Bucket S3 de origem chamado "EMR-helloworld-<seu nome>"como apoio estou disponibilizando o passo a passo para o exercício <https://www.edureka.co/blog/aws-glue/>
- 5) Faça um Job no Glue para processar o mesmo conjunto de dados do exercício 3 de EMR sobre os Resultados da inspeção do Departamento de Saúde em King County.
- 6) Faça um agendamento de Job AWS Glue ETL

AWS Athena

O Amazon Athena Amazon Simple Storage Service é um serviço de consulta interativo que facilita a análise de dados diretamente no Amazon S3 usando SQL. Com algumas ações no Console de Gerenciamento da AWS, você pode apontar o Athena para os dados armazenados no Amazon S3 e começar a usar o SQL padrão para executar consultas ad-hoc e receber resultados em segundos.

O Athena não possui servidor, portanto, não há infraestrutura para configurar ou gerenciar, e você paga apenas pelas consultas executadas. O Athena — escala automaticamente — executando consultas em paralelo para que os resultados sejam rápidos, mesmo com conjuntos de dados grandes e consultas complexas.

Quando devo usar o Athena?

O Athena ajuda a analisar dados desestruturados, semiestruturados e estruturados armazenados no Amazon S3. Entre os exemplos estão formatos de dados CSV, JSON ou colunares, como Apache Parquet e Apache ORC. Você pode usar o Athena para executar consultas ad-hoc com o SQL ANSI, sem necessidade de agregar ou carregar os dados no Athena.

O Athena se integra ao Amazon QuickSight para oferecer fácil visualização de dados. Você pode usar o Athena para gerar relatórios ou explorar dados com ferramentas de business intelligence ou clientes SQL conectados com um driver JDBC ou ODBC.

O Athena se integra ao Catálogo de dados do AWS Glue, que oferece um armazenamento de metadados persistente para os dados no Amazon S3. Isso permite criar tabelas e consultar dados no Athena com base em um armazenamento de metadados central disponível em toda a conta da AWS e integrado com o ETL e os recursos de descoberta de dados do AWS Glue.

Você pode acessar o Athena usando o Console de gerenciamento da AWS ou uma conexão JDBC ou ODBC, a API do Athena, a CLI da Athena, o SDK da AWS ou o AWS Tools para Windows PowerShell.

No Athena, tabelas e bancos de dados são contêineres para as definições de metadados que definem um esquema para dados de origem subjacentes. Para cada conjunto de dados, deve existir uma tabela no Athena. Os metadados na tabela informam ao Athena onde os dados estão localizados no Amazon S3 e especificam a estrutura dos dados, por exemplo, nomes de coluna, tipos de dados e o nome da tabela. Os bancos de dados são um agrupamento lógico de tabelas e também mantêm somente metadados e informações do esquema de um conjunto de dados

Para cada conjunto de dados que você deseja consultar, o Athena deve ter uma tabela subjacente que usará para obter e retornar os resultados da consulta. Por isso, antes de consultar dados, uma tabela deve ser registrada no Athena. O registro ocorre quando você cria tabelas automática ou manualmente.

Independentemente de como as tabelas são criadas, o processo de criação de tabelas registra o conjunto de dados no Athena. Esse registro ocorre **no AWS Glue Data Catalog** e permite que o Athena execute consultas nos dados.

O Data Catalog do AWS Glue é acessível em toda a sua conta da AWS. Outros serviços da AWS podem compartilhar o Data Catalog do AWS Glue, para que você veja bancos de dados e tabelas criados em toda a organização usando o Athena e vice-versa. Para criar uma tabela manualmente:

- Use o console do Athena para executar o Create Table Wizard (Assistente de Criação de tabela).
- Use o console do Athena para escrever instruções DDL do Hive no Query Editor (Editor de consultas).
- Use a API ou a CLI do Athena para executar uma string de consulta SQL com instruções DDL.
- Use o driver JDBC ou ODBC do Athena

Quando você cria tabelas e bancos de dados manualmente, o Athena usa instruções de linguagem de definição de dados (DDL) do HiveQL, como CREATE TABLE, CREATE DATABASE e DROP TABLE, nos bastidores para criar tabelas e bancos de dados no AWS Glue Data Catalog.

Quando você consulta uma tabela existente, o Amazon Athena usa o Presto (<https://prestodb.io/>), um mecanismo distribuído de SQL para BigData.

Quando você acessa o Athena pelo Console da AWS você pode fazer o seguinte:

- Criar ou selecionar um banco de dados.
- Criar, visualizar e excluir tabelas.
- Filtrar tabelas começando a digitar os nomes delas.
- Visualizar tabelas e gerar CREATE TABLE DDL para elas.
- Mostrar as propriedades da tabela.
- Executar consultas em tabelas, salvar e formatar consultas e visualizar o histórico de consultas.
- Criar até dez consultas usando diferentes guias de consulta no editor de consultas. Para abrir uma nova aba, clique no sinal de adição.
- Exibir, salvar e exportar os resultados da consulta.
- Acessar o Catálogo de dados do AWS Glue.
- Visualizar e alterar as configurações, como visualizar o local do resultado da consulta, configurar o preenchimento automático e criptografar os resultados da consulta.

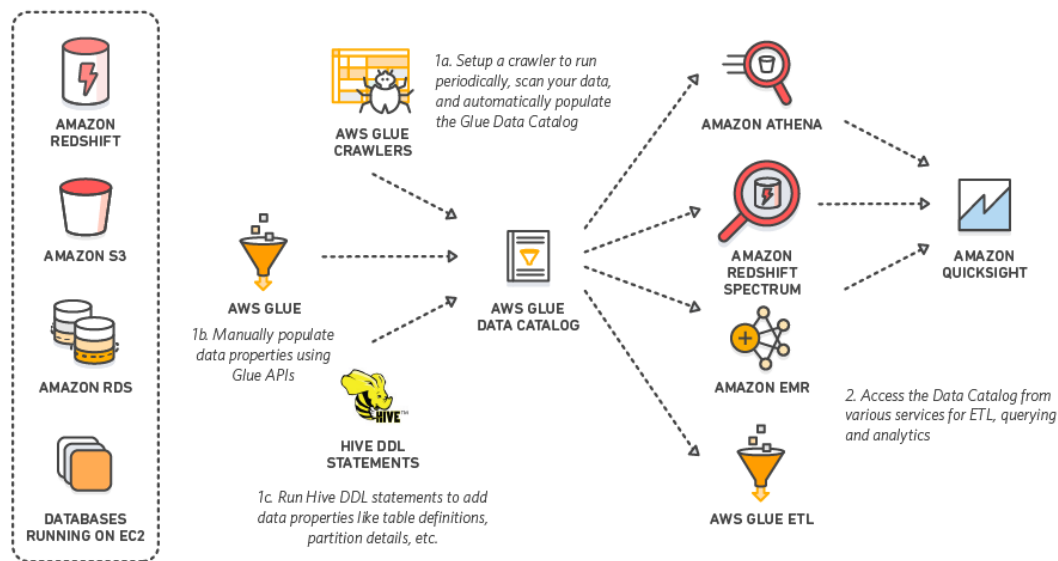
No painel direito, o Query Editor exibe uma tela de introdução que solicita a criação de sua primeira tabela. Você pode visualizar suas tabelas em Tables no painel esquerdo.

Veja a visão geral de alto nível das ações disponíveis para cada tabela:

- Preview tables (Visualizar tabelas) – visualizar a sintaxe de consulta no Editor de consulta à direita.
- Show properties (Mostrar propriedades) – mostrar o nome de uma tabela, sua localização no Amazon S3, os formatos de entrada e saída, a biblioteca de serialização (SerDe) utilizada e se a tabela tem dados criptografados.

- Delete table (Excluir tabela) – excluir tabela.
- Generate CREATE TABLE DDL (Gerar CREATE TABLE DDL) – gerar a consulta por trás de uma tabela e visualizá-la no editor de consultas.

O Athena oferece suporte nativo à consulta de conjuntos de dados e fontes de dados que são registrados com o Catálogo de dados do AWS Glue. Ao executar consultas em DML (Data Manipulation Language – Linguagem de manipulação de dados) no Athena com o Data Catalog como origem, você está usando o esquema do Data Catalog para derivar informações do conjunto de dados. Ao executar consultas DDL (Data Definition Language), o esquema é atualizado no Catálogo de dados do AWS Glue. No Athena, também é possível executar um crawler do AWS Glue em uma fonte de dados para criar um esquema no Catálogo de dados do AWS Glue.



Local da tabela no Amazon S3

Ao executar uma consulta CREATE TABLE no Athena, você registra a tabela no Data Catalog do AWS Glue. Use a propriedade LOCATION para especificar o caminho para os dados no Amazon S3, conforme mostrado no exemplo a seguir:

```
CREATE EXTERNAL TABLE `test_table`(  
  ...  
)  
ROW FORMAT ...  
STORED AS INPUTFORMAT ...  
OUTPUTFORMAT ...  
LOCATION s3://bucketname/folder/
```

O LOCATION no Amazon S3 especifica todos os arquivos que representam a sua tabela. O Athena lê todos os dados armazenados no s3://bucketname/folder/. Se você tiver dados que não quer que o Athena leia, não armazene esses dados na mesma pasta do Amazon S3. Se

you are using data partitioning, to ensure that Athena verifies data within a partition, your WHERE filter must include the partition.

When specifying the LOCATION in the CREATE TABLE instruction, use the following guidelines:

- Use a slash at the end.

Use:

```
s3://bucketname/folder/
```

Local e partições de tabela

Your data can be grouped into folders in Amazon S3 called **partitions**, based on a set of columns. For example, these columns can represent the year, month, and day the record was created.

When creating a table, you **can** opt to make it partitioned. When Athena executes a SQL query on a non-partitioned table, it uses the LOCATION property of the table definition as the base path to list and verify all available files. However, for a partitioned table to be queryable, you must update the AWS Glue Data Catalog with information about the partition. This information represents the schema of files in the partition and the LOCATION of files in Amazon S3 for the partition.

When Athena executes a query on a partitioned table, it checks if any partitioned columns were used in the WHERE clause of the query. If the partitioned columns were used, Athena requests the AWS Glue Data Catalog to return the partition specification corresponding to the columns in the specified partition. The partition specification includes the LOCATION property, which indicates to Athena which prefix of Amazon S3 to use to read the data. In this case, only the data stored in that prefix is verified. If you do not use partitioned columns in the WHERE clause, Athena verifies **all** files that belong to the table's partitions.

Tips for maximizing Amazon Athena performance:

<https://aws.amazon.com/pt/blogs/big-data/top-10-performance-tuning-tips-for-amazon-athena/>

Partitionar dados

Partitioning your data, you can restrict the number of data files verified in each query, **improving performance and reducing cost**. Athena uses Apache Hive for data partitioning. You can divide your data into partitions using any key. A common practice is to partition data by time, typically creating a partitioning schema at various levels. For example, a client that has data coming in hourly can choose to partition by year, month, day, and hour. Another client, who has data from many different sources, but loaded once a day, can partition by a source identifier and date.

Condições e limitações do Particionamento

Ao usar o particionamento, lembre-se dos seguintes pontos:

- Se você consultar uma tabela particionada e especificar a partição na cláusula WHERE, o Athena verificará somente os dados dessa partição.
- Se você emitir consultas em buckets do Amazon S3 com um grande número de objetos, e os dados não estiverem particionados, essas consultas poderão afetar os limites da taxa de solicitações **GET no Amazon S3 e resultar em exceções do Amazon S3. Para evitar erros, particione seus dados.** Além disso, considere ajustar suas taxas de solicitações do Amazon S3. Para obter mais informações, consulte Melhores práticas e padrões de design: otimização do desempenho do Amazon S3 (https://docs.aws.amazon.com/pt_br/AmazonS3/latest/userguide/optimizing-performance.html).
- Os locais de partição a serem usados com o Athena devem usar o protocolo do s3 (por exemplo, **s3://bucket/folder/**). No Athena, os locais que usam outros protocolos (por exemplo, **s3a://bucket/folder/**) resultarão em falhas na consulta quando consultas MSCK REPAIR TABLE forem executadas nas tabelas que às contem.
- Como o MSCK REPAIR TABLE verifica uma pasta e suas subpastas para localizar um esquema de partição correspondente, **mantenha os dados de tabelas separadas em hierarquias de pastas separadas.** Por exemplo, suponha que você tenha dados para a tabela A no **s3://table-a-data** e dados para a tabela B no **s3://table-a-data/table-b-data**. Se ambas as tabelas forem particionadas por string, o MSCK REPAIR TABLE adicionará as partições da tabela B à tabela A. Para evitar isso, use estruturas de pasta separadas, como **s3://table-a-data** e **s3://table-b-data**. Observe que esse comportamento é consistente com o Amazon EMR e o Apache Hive.

Formatos de armazenamento colunar

Apache Parquet e ORC são formatos de armazenamento colunar otimizados para recuperação rápida de dados e usados em aplicações analíticas da AWS.

Os formatos de armazenamento colunar têm as seguintes características que os tornam adequados para o uso com o Athena:

- Compactação por coluna, com algoritmo de compactação selecionado para o tipo de dados colunar para **economizar espaço** de armazenamento no Amazon S3, **reduzir o E/S** e espaço de disco durante o processamento de consultas.
- A aplicação de predicados em Parquet e ORC permite que as consultas do Athena obtenham **somente os blocos de que se precisa**, melhorando o desempenho das consultas. Quando uma consulta do Athena obtém valores de coluna específicos de seus dados, ela usa estatísticas de predicados de bloco de dados, como valores máximos e mínimos, para determinar se deseja ler ou ignorar o bloco.
- A divisão de dados em Parquet e ORC permite que o Athena divida a leitura de dados **entre vários leitores e aumente o paralelismo** durante o processamento da consulta

Para converter seus dados brutos existentes de outros formatos de armazenamento para Parquet ou ORC, é possível executar consultas CREATE TABLE AS SELECT (CTAS) no Athena e

especificar um formato de armazenamento físico de dados como Parquet ou ORC, ou usar o crawler do AWS Glue.

Exemplos:

- a. Criar e carregar uma tabela com dados particionados:

Para criar uma tabela que usa partições, você deve defini-la durante a instrução CREATE TABLE. Use PARTITIONED BY para definir as chaves segundo as quais particionar os dados, como no exemplo a seguir. LOCATION especifica o local raiz dos dados particionados.

```
CREATE EXTERNAL TABLE users (  
  first string,  
  last string,  
  username string  
)  
PARTITIONED BY (id string)  
STORED AS parquet  
LOCATION 's3://bucket/folder/'
```

Depois de criar a tabela, carregue os dados nas partições para consulta. Para dados compatíveis com o Hive, execute MSCK REPAIR TABLE. Para dados não compatíveis com Hive, use ALTER TABLE ADD PARTITION para adicionar as partições manualmente.

- b. Preparar dados particionados e não particionados para consulta

As seções a seguir abordam dois cenários:

- i. Os dados já estão particionados, armazenados no Amazon S3 e você precisa acessar os dados no Athena.
- ii. Os dados não são particionados.

Cenário 1: os dados já estão particionados e armazenados no S3 no formato Hive

As partições são armazenadas em pastas separadas no Amazon S3. Por exemplo, aqui está a listagem parcial para impressões de anúncio de exemplo:

```
aws s3 ls s3://elasticmapreduce/samples/hive-ads/tables/impressions/
```

```
PRE dt=2009-04-12-13-00/  
PRE dt=2009-04-12-13-05/  
PRE dt=2009-04-12-13-10/  
PRE dt=2009-04-12-13-15/  
PRE dt=2009-04-12-13-20/  
PRE dt=2009-04-12-14-00/  
PRE dt=2009-04-12-14-05/  
PRE dt=2009-04-12-14-10/  
PRE dt=2009-04-12-14-15/  
PRE dt=2009-04-12-14-20/  
PRE dt=2009-04-12-15-00/  
PRE dt=2009-04-12-15-05/
```

Aqui, os logs são armazenados com o nome da coluna (dt) definido igual a incrementos de data, hora e minuto. Quando fornece uma DDL com o local da pasta pai, o esquema e o nome da coluna particionada, o Athena pode consultar dados nessas subpastas.

Para criar uma tabela a partir desses dados, crie uma partição com 'dt', como na seguinte instrução DDL do Athena

```
CREATE EXTERNAL TABLE impressions (  
    requestBeginTime string,  
    adId string,  
    impressionId string,  
    referrer string,  
    userAgent string,  
    userCookie string,  
    ip string,  
    number string,  
    processId string,  
    browserCookie string,  
    requestEndTime string,  
    timers struct<modelLookup:string, requestTime:string>,  
    threadId string,  
    hostname string,  
    sessionId string)  
PARTITIONED BY (dt string)  
ROW FORMAT serde 'org.apache.hive.hcatalog.data.JsonSerDe'  
LOCATION 's3://elasticmapreduce/samples/hive-ads/tables/impressions/' ;
```

Esta tabela usa o SerDe (Serializador/desserializador) JSON nativo do Hive para ler dados **JSON** armazenados no Amazon S3. Um SerDe (Serializador/desserializador) é uma maneira como o Athena interage com dados em vários formatos. É a SerDe que você especifica, e não a DDL, e que define o esquema de tabela. Em outras palavras, o SerDe pode substituir a configuração de DDL especificada em Athena ao criar a tabela.

Para usar um SerDe ao criar uma tabela no Athena, use um dos seguintes métodos:

- Use instruções DDL para descrever como ler e gravar dados na tabela e não especifique um ROW FORMAT, como neste exemplo. Isso omite a listagem do tipo SerDe real, e o LazySimpleSerDe nativo é usado por padrão. Em geral, o Athena usará o LazySimpleSerDe, se você não especificar um ROW FORMAT, ou se especificar ROW FORMAT DELIMITED.

```
ROW FORMAT
DELIMITED FIELDS TERMINATED BY ','
ESCAPED BY '\\'
COLLECTION ITEMS TERMINATED BY '|'
MAP KEYS TERMINATED BY ':'
```

- Especifique explicitamente o tipo de SerDe que o Athena deve usar ao ler e gravar dados na tabela. Além disso, especifique propriedades adicionais em SERDEPROPERTIES, como neste exemplo.

```
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = ',',
  'field.delim' = ',',
  'collection.delim' = '|',
  'mapkey.delim' = ':',
  'escape.delim' = '\\'
)
```

O Athena oferece suporte à criação de tabelas e à consulta de dados nos formatos CSV, TSV, delimitado personalizado e JSON; dados de formatos relacionados ao Hadoop: Logs ORC, Apache Avro e Parquet; logs do Logstash, logs do AWS CloudTrail e logs do Apache WebServer.

Depois de executar a instrução anterior em Athena, escolha New Query (Nova consulta) e execute o seguinte comando: **MSCK REPAIR TABLE impressions**

O Athena carrega os dados nas partições.

Agora consulte os dados da tabela de impressões usando a coluna de partição. Veja um exemplo abaixo:

```
SELECT dt,impressionid
```


FROM impressions

WHERE dt<'2009-04-12-14-00' and dt>='2009-04-12-13-00'

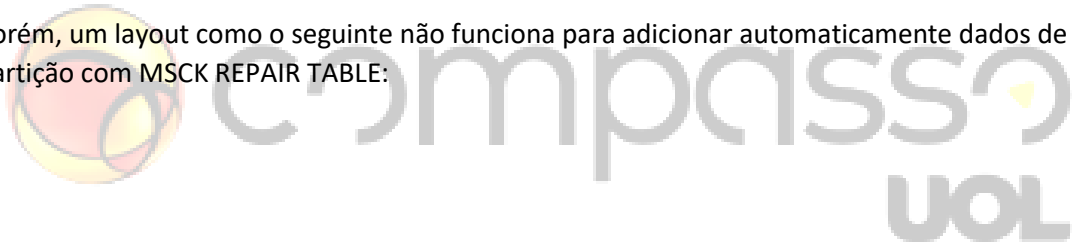
ORDER BY dt DESC LIMIT 100

Esta consulta deve mostrar dados semelhantes aos seguintes:

2009-04-12-13-20	ap3HcVKAwfXtgIPu6WpuUfAfL0DQEc
2009-04-12-13-20	17uchtodoS9kdeQP1x0XThK15IuRsV
2009-04-12-13-20	JOUf1SCtRwviGw8sVcghqE5h0nkgtp
2009-04-12-13-20	NQ2XP0J0dvVbCXJ0pb4XvqJ5A4QxxH
2009-04-12-13-20	fFAItiBMsgqro9kRdIwbeX60SR0axr
2009-04-12-13-20	V4og4R9W6G3QjHHwF7gI1cSqi5D1G
2009-04-12-13-20	hPEPtBwk45msmwWTxPVVo1kVu4v11b
2009-04-12-13-20	v0SkfxegheD90gp31UCr6Fp1nKpx6i
2009-04-12-13-20	1iD9odVg0Ii4QWkwHMcOhmwTkWDKfj
2009-04-12-13-20	b31tJiIA25CK8eDHQrHnbcknfSndUk

Cenário 2: Os dados não são particionados no formato Hive

Porém, um layout como o seguinte não funciona para adicionar automaticamente dados de partição com MSCK REPAIR TABLE:



```
aws s3 ls s3://athena-examples-myregion/elb/plaintext/ --recursive

2016-11-23 17:54:46 11789573 elb/plaintext/2015/01/01/part-r-00000-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:46 8776899 elb/plaintext/2015/01/01/part-r-00001-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:46 9309800 elb/plaintext/2015/01/01/part-r-00002-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:47 9412570 elb/plaintext/2015/01/01/part-r-00003-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:47 10725938 elb/plaintext/2015/01/01/part-r-00004-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:46 9439710 elb/plaintext/2015/01/01/part-r-00005-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:47 0 elb/plaintext/2015/01/01_$folder$
2016-11-23 17:54:47 9012723 elb/plaintext/2015/01/02/part-r-00006-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:47 7571816 elb/plaintext/2015/01/02/part-r-00007-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:47 9673393 elb/plaintext/2015/01/02/part-r-00008-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:48 11979218 elb/plaintext/2015/01/02/part-r-00009-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:48 9546833 elb/plaintext/2015/01/02/part-r-00010-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:48 10960865 elb/plaintext/2015/01/02/part-r-00011-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:48 0 elb/plaintext/2015/01/02_$folder$
2016-11-23 17:54:48 11360522 elb/plaintext/2015/01/03/part-r-00012-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:48 11211291 elb/plaintext/2015/01/03/part-r-00013-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:48 8633768 elb/plaintext/2015/01/03/part-r-00014-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:49 11891626 elb/plaintext/2015/01/03/part-r-00015-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:49 9173813 elb/plaintext/2015/01/03/part-r-00016-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:49 11899582 elb/plaintext/2015/01/03/part-r-00017-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:49 0 elb/plaintext/2015/01/03_$folder$
2016-11-23 17:54:50 8612843 elb/plaintext/2015/01/04/part-r-00018-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:50 10731284 elb/plaintext/2015/01/04/part-r-00019-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:50 9984735 elb/plaintext/2015/01/04/part-r-00020-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:50 9290089 elb/plaintext/2015/01/04/part-r-00021-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:50 7896339 elb/plaintext/2015/01/04/part-r-00022-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 8321364 elb/plaintext/2015/01/04/part-r-00023-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 0 elb/plaintext/2015/01/04_$folder$
2016-11-23 17:54:51 7641062 elb/plaintext/2015/01/05/part-r-00024-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 10253377 elb/plaintext/2015/01/05/part-r-00025-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 8502765 elb/plaintext/2015/01/05/part-r-00026-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 11518464 elb/plaintext/2015/01/05/part-r-00027-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 7945189 elb/plaintext/2015/01/05/part-r-00028-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 7864475 elb/plaintext/2015/01/05/part-r-00029-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 0 elb/plaintext/2015/01/05_$folder$
2016-11-23 17:54:51 11342140 elb/plaintext/2015/01/06/part-r-00030-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 8063755 elb/plaintext/2015/01/06/part-r-00031-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 9387508 elb/plaintext/2015/01/06/part-r-00032-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 9732343 elb/plaintext/2015/01/06/part-r-00033-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 11510326 elb/plaintext/2015/01/06/part-r-00034-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 9148117 elb/plaintext/2015/01/06/part-r-00035-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 0 elb/plaintext/2015/01/06_$folder$
2016-11-23 17:54:52 8402024 elb/plaintext/2015/01/07/part-r-00036-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 8282860 elb/plaintext/2015/01/07/part-r-00037-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 11575283 elb/plaintext/2015/01/07/part-r-00038-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:53 8149059 elb/plaintext/2015/01/07/part-r-00039-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:53 10037269 elb/plaintext/2015/01/07/part-r-00040-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:53 10019678 elb/plaintext/2015/01/07/part-r-00041-ce65fca5-d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:53 0 elb/plaintext/2015/01/07_$folder$
2016-11-23 17:54:53 0 elb/plaintext/2015/01_$folder$
2016-11-23 17:54:53 0 elb/plaintext/2015_$folder$
```

Neste caso, você precisaria usar ALTER TABLE ADD PARTITION para adicionar cada partição manualmente.

Por exemplo, para carregar os dados em **s3://athena-examples-myregion/elb/plaintext/2015/01/01/**, você pode executar o seguinte. Observe que não é necessária uma coluna de partição separada para cada pasta do Amazon S3 e que o valor da chave de partição pode ser diferente da chave do Amazon S3

```
ALTER TABLE elb_logs_raw_native_part ADD PARTITION (dt='2015-01-01') location 's3://athena-examples-us-west-1/elb/plaintext/2015/01/01/'
```

Projeção de partições com o Amazon Athena

Na projeção de partições, os valores e locais de partição são calculados a partir da configuração, em vez de lidos em um repositório como o Catálogo de dados do AWS Glue. Como as operações na memória costumam ser mais rápidas do que as operações remotas, a projeção de partições pode reduzir o tempo de execução de consultas em tabelas altamente particionadas. Dependendo das características específicas da consulta e dos dados subjacentes, a projeção de partições pode reduzir significativamente o tempo de execução para consultas restritas na recuperação de metadados de partição.

Redução e projeção para tabelas extremamente particionadas

A redução de partição coleta metadados e os “reduz” para apenas as partições que se aplicam à sua consulta. Isso geralmente acelera as consultas. O Athena usa a redução de partição para todas as tabelas com colunas de partição, incluindo as tabelas configuradas para projeção de partições.

Normalmente, ao processar consultas, o Athena faz uma chamada GetPartitions para o Catálogo de dados do AWS Glue antes de executar a redução de partição. Se uma tabela tiver um grande número de partições, o uso de GetPartitions poderá afetar negativamente o desempenho. Para evitar isso, você pode usar a projeção de partições. A projeção de partições permite que o Athena evite chamar GetPartitions porque a configuração de projeção de partições fornece ao Athena todas as informações necessárias para criar as partições propriamente ditas.

Para usar a projeção de partições, especifique os intervalos de valores de partição e os tipos de projeção para cada coluna de partição nas propriedades da tabela no Catálogo de dados do AWS Glue ou no metastore do Hive externo. Essas **propriedades personalizadas** na tabela permitem que o Athena saiba quais padrões de partição esperar quando executa uma consulta na tabela. Durante a execução da consulta, o Athena usa essas informações para projetar os valores de partição em vez de recuperá-los do Catálogo de dados do AWS Glue ou do metastore do Hive externo. Isso não só reduz o tempo de execução da consulta, mas também automatiza o gerenciamento de partições porque remove a necessidade de criar manualmente partições no Athena, no AWS Glue ou no metastore do Hive externo.

Casos de uso

Os cenários em que a projeção de partições é útil incluem o seguinte:

- As consultas em uma tabela altamente particionada não são concluídas com a rapidez desejada.
- Você adiciona partições regularmente a tabelas à medida que novas partições de data ou hora são criadas em seus dados. Com a projeção de partições, você configura intervalos de datas relativos que podem ser usados à medida que novos dados chegam.

- Você tem dados altamente particionados no Amazon S3. Os dados não são práticos para modelar no Catálogo de dados do AWS Glue ou no metastore do Hive, e suas consultas leem apenas pequenas partes deles.

Estruturas de partições projetáveis

A projeção de partições é mais facilmente configurada quando as partições seguem um padrão previsível como, mas não limitado ao seguinte:

- **Inteiros** – qualquer sequência contínua de inteiros, como [1, 2, 3, 4, ..., 1000] ou [0500, 0550, 0600, ..., 2500].
- **Datas** – qualquer sequência contínua de datas ou datas e horas, como [20200101, 20200102, ..., 20201231] ou [1-1-2020 00:00:00, 1-1-2020 01:00:00, ..., 12-31-2020 23:00:00].
- **Valores enumerados** – um conjunto finito de valores enumerados, como códigos de aeroporto ou regiões da AWS
- **Logs de serviço da AWS** – os logs de serviço da AWS normalmente têm uma estrutura conhecida cujo esquema de partição pode ser especificado no AWS Glue e que o Athena pode, portanto, usar para projeção de partições.

As seguintes considerações se aplicam:

- A projeção de partições elimina a necessidade de especificar partições manualmente no AWS Glue ou em um metastore do Hive externo.
- Quando você habilita a projeção de partições em uma tabela, o Athena ignora todos os metadados de partição no Catálogo de dados do AWS Glue ou no metastore do Hive externo para essa tabela.
- Se uma partição projetada não existir no Amazon S3, o Athena ainda irá projetar a partição. O Athena não lança um erro, mas nenhum dado é retornado. No entanto, se muitas partições estiverem vazias, o desempenho poderá ser mais lento em comparação com as partições tradicionais do AWS Glue. Se mais da metade das partições projetadas estiver vazia, é recomendado usar partições tradicionais.
- A projeção de partições pode ser usada somente quando a tabela é consultada pelo Athena. Se a mesma tabela for lida por meio de outro serviço, como o Amazon Redshift Spectrum ou Amazon EMR, os metadados de partição padrão serão usados.
- Como a projeção de partições é um recurso somente DML, SHOW PARTITIONS não lista as partições projetadas pelo Athena, mas não registradas no catálogo do AWS Glue ou metastore do Hive externo.
- As visualizações no Athena não usam propriedades de configuração de projeção.

Exemplo de Projeção de Partições:

×

Edit table details

Table name

flights_parquet

Input format

org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat

Output format

Table properties

Key	Value	
last_modified_time	1582588443	×
EXTERNAL	TRUE	×
last_modified_by	hadoop	×
projection.year.type	Integer	×
projection.year.range	2000,2016	×
		×

Table properties

Key	Value	
last_modified_time	1582588443	×
EXTERNAL	TRUE	×
last_modified_by	hadoop	×
projection.year.type	Integer	×
projection.year.range	2000,2016	×
projection.enabled	true	×

Apply

Resultados de consulta, arquivos de saída e histórico de consultas

O Amazon Athena armazena automaticamente os resultados da consulta e as informações de metadados para cada consulta executada em um local de resultados da consulta que você pode especificar no Amazon S3. Se necessário, você pode acessar os arquivos nesse local para trabalhar com eles. Também é possível fazer download dos arquivos de resultados da consulta diretamente no console do Athena.

Os arquivos de saída são salvos automaticamente para cada consulta executada, independentemente de a consulta em si ter sido salva ou não. Para acessar e visualizar arquivos de saída da consulta, as entidades principais do IAM (usuários e funções) precisam de permissão para a ação Amazon S3 `GetObject` do local de resultados da consulta, bem como permissão para a ação `GetQueryResults` do Athena. O local de resultados da consulta pode ser criptografado. Se o local estiver criptografado, os usuários deverão ter as permissões de chave apropriadas para criptografar e descriptografar o local de resultados da consulta.

Views no Athena

Uma view ou visualização no Amazon Athena é uma tabela lógica, não física. A consulta que define uma exibição é executada sempre que a exibição é referenciada em uma consulta.

Você pode criar uma view a partir de uma consulta `SELECT` e, em seguida, fazer referência a essa view em futuras consultas. Views podem ser usadas quando:

- Consultar um subconjunto de dados. Por exemplo, você pode criar uma view com um subconjunto de colunas da tabela original para simplificar a consulta de dados.
- Combinar várias tabelas em uma consulta. Se você tem várias tabelas e deseja combiná-las com `UNION ALL`, é possível criar uma view com essa expressão para simplificar consultas em tabelas combinadas.
- Ocultar a complexidade de consultas básicas existentes e simplificar as consultas executadas pelos usuários. As consultas básicas geralmente incluem junções entre tabelas, expressões na lista de colunas e outra sintaxe de SQL que dificultam o entendimento e a depuração delas. Você pode criar uma view que oculta a complexidade e simplifica consultas.
- Experimente com técnicas de otimização e crie consultas otimizadas. Por exemplo, se você encontrar uma combinação de condições `WHERE`, ordem de cláusulas de `JOIN` ou outras expressões que demonstram o melhor desempenho, você poderá criar uma view com essas cláusulas e expressões. As aplicações podem realizar consultas relativamente simples contra essa view. Posteriormente, se você encontrar uma melhor maneira de otimizar a consulta original, quando você recriar a view, todas as aplicações aproveitarão imediatamente a consulta básica otimizada.
- Oculte a tabela subjacente e os nomes de coluna e minimize os problemas de manutenção se esses nomes forem alterados. Nesse caso, recrie a exibição usando os novos nomes. Todas as consultas que usam a exibição ao invés de tabelas subjacentes continuarão em execução sem alterações.

Limitações de views

- Os nomes de views no Athena não podem conter caracteres especiais, além de sublinhado (`_`).

- Evite usar palavras-chave reservadas para nomear views. Se você usar palavras-chave reservadas, use aspas duplas para delimitar as palavras-chave reservadas em suas consultas em views.
- Não é possível usar views com fontes de dados federadas, metastores do Hive externos ou UDFs.
- Não é possível usar views com funções geoespaciais.
- Não é possível usar views para gerenciar o controle de acesso a dados no Amazon S3.

Exercício:

- 7) Hello World Athena: https://docs.aws.amazon.com/pt_br/athena/latest/ug/getting-started.html
- 8) Converter dados para o Parquet usando um cluster do EMR:
https://docs.aws.amazon.com/pt_br/athena/latest/ug/convert-to-columnar.html
- 9) Crie Views com Athena para o conjunto de dados do exercício anterior. O link a seguir pode ser usado como exemplo:
https://docs.aws.amazon.com/pt_br/athena/latest/ug/create-view.html

AWS QuickSight

O Amazon QuickSight é um serviço de inteligência de negócios rápido e baseado na nuvem que facilita a entrega de insights a todos em sua organização.

Como um serviço gerenciado, o QuickSight permite que você crie e publique facilmente painéis interativos que incluem o Insights de Machine Learning (ML). Os painéis podem ser acessados de qualquer dispositivo e incorporados a aplicativos, portais e sites.

Com a nossa definição de preço de pagamento por sessão, o QuickSight permite que todos acessem os dados de que precisam e paguem apenas pela utilização.

- Descubra informações ocultas com **ML insights**

O ML Insights aproveita os recursos comprovados de aprendizado de máquina (ML) e capacidades de linguagem natural da AWS para ajudá-lo a obter informações mais detalhadas de seus dados. Esses recursos poderosos e prontos para uso facilitam a descoberta de tendências e discrepâncias ocultas, a identificação de fatores-chave de negócios e a realização de análises e previsões hipotéticas sem necessidade de conhecimento técnico ou experiência em ML.

- Pague apenas pelo que usar

O QuickSight oferece um modelo único de pay-per-session para leitores de dashboards e usuários que consomem dashboards que outros criaram. Em vez de pagar um custo fixo de licença por mês, os leitores recebem US\$ 0,30 por uma sessão de 30 minutos, até uma cobrança máxima de US\$ 5 / leitor / mês para uso ilimitado. Esse modelo de preços permite que todos os usuários acessem painéis interativos seguros e relatórios de e-mail com base em pagamento por sessão, sem custos iniciais ou planejamento de capacidade complexo.

- Entregue dashboards interativos avançados para seus leitores

O QuickSight torna fácil e rápido a criação de dashboards e relatórios interativos para seus usuários. Você pode compartilhar com segurança esses dashboards com qualquer pessoa da sua organização por meio de navegadores ou dispositivos móveis. Com o QuickSight, você pode escolher entre uma ampla biblioteca de visualizações, gráficos e tabelas, adicionar recursos interativos, como drill-downs e filtros, e executar atualizações automáticas de dados para criar dashboards interativos. O QuickSight também permite que você agende relatórios automáticos baseados em email, para que você possa receber as principais informações na sua caixa de entrada.

- Explore, analise e colabore

O QuickSight oferece aos usuários e analistas de self-service business intelligence (BI), para que eles possam responder suas próprias perguntas, colaborar e compartilhar insights. Com o QuickSight, seus usuários podem se conectar a fontes de dados, criar / editar conjuntos de dados, criar análises visuais, convidar colegas de trabalho para colaborar nas análises e publicar dashboards e relatórios.

- SPICE (super-fast, parallel, in-memory, calculation engine)

Com o SPICE, o mecanismo de cálculo na memória do QuickSight, você obtém um desempenho incrivelmente rápido em escala. O SPICE replica automaticamente os dados para alta disponibilidade, permitindo que milhares de usuários realizem análises interativas e rápidas, protegendo sua infraestrutura de dados subjacente, economizando tempo e recursos.

- Embed dashboards and APIs

Com o QuickSight, você pode criar e incorporar facilmente visualizações e dashboards interativos em seus aplicativos e portais da Web usando logon único e APIs - sem escrever código ou pagar por dashboards incorporados não utilizados.

- Conecte-se aos seus dados, onde quer que estejam

O QuickSight permite conectar e importar diretamente dados de uma ampla variedade de fontes de dados na nuvem e on-premise. Isso inclui aplicativos SaaS como Salesforce, Square, ServiceNow, Twitter, Github e JIRA; Bancos de dados de terceiros, como Teradata, MySQL, Postgres e SQL Server; serviços nativos da AWS, como Redshift, Athena, S3, RDS e Aurora; e sub-redes de VPC privadas. Você também pode fazer upload de vários tipos de arquivos, incluindo Excel, CSV, JSON e Presto.

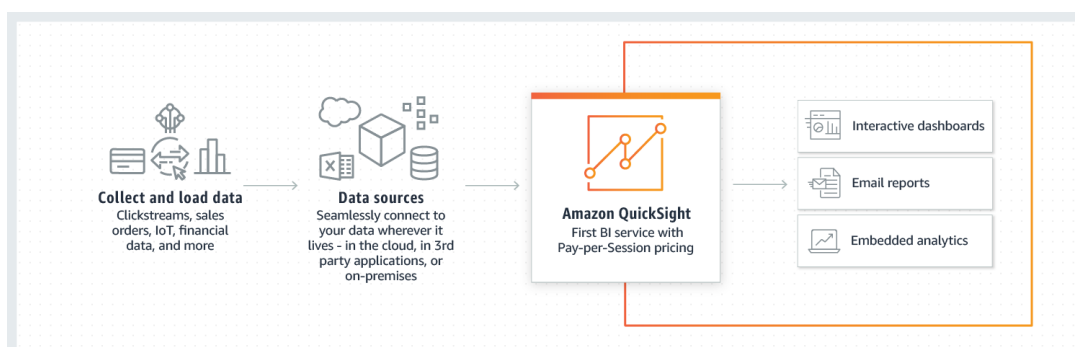
- Uma solução Global

Como um serviço nativo da AWS com clientes em todo o mundo, o QuickSight foi projetado e construído como um produto global desde o início. O aplicativo QuickSight está localizado em 10 idiomas principais, incluindo: inglês, alemão, espanhol, francês, italiano, português, japonês, coreano, chinês simplificado e chinês tradicional. O QuickSight também está disponível em várias regiões da AWS, incluindo: N. Virginia, Oregon, Ohio, Dublin, Japão, Cingapura e Sydney.

- Obtenha segurança e conformidade

O QuickSight fornece uma plataforma segura, permitindo distribuir dashboards e insights com segurança a dezenas de milhares de usuários. Além da disponibilidade em várias regiões e redundância integrada, o QuickSight permite gerenciar com segurança seus usuários e conteúdo por meio de um conjunto abrangente de recursos de segurança, incluindo controle de acesso baseado em função, integração de diretório ativo, auditoria do CloudTrail, logon único (IAM, terceiros), sub-redes VPC privadas e backup de dados. O QuickSight também é compatível com FedRamp, HIPAA, PCI PSS, ISO e SOC para ajudá-lo a atender a quaisquer requisitos regulamentares ou específicos do setor.

Como Funciona:



Veja o vídeo introdutório sobre o Quicksight: <https://youtu.be/2V1bHRLRG-w>

Exercícios:

- 10) Criar uma análise com um único visual usando dados de exemplo -
https://docs.aws.amazon.com/pt_br/quicksight/latest/user/quickstart-createanalysis.html

AWS LakeFormation

AWS Lake Formation é um serviço totalmente gerenciado que facilita a criação, segurança e gestão de Data Lakes. Lake Formation simplifica e automatiza muitos dos passos manuais complexos que são normalmente necessários para criar Data Lakes. Estes passos incluem a coleta, limpeza, transferência e catalogação de dados e a disponibilização segura desses dados para análise e aprendizagem de máquina. Você aponta Lake Formation nas suas fontes de dados e Lake Formation aponta essas fontes de dados e transfere os dados para o seu novo Data Lake em Amazon Simple Storage Service (Amazon S3).

Lake Formation fornece o seu próprio modelo de permissões que aumenta o AWS Identity and Access Management (IAM) modelos de permissões. Este modelo de permissões definidas centralmente permite um acesso restrito aos dados armazenados em Data Lakes através de um mecanismo simples de concessão/revogação.

As permissões do Lake Formation são aplicadas ao nível da tabela e coluna em todo o portfólio de análises da AWS e serviços de aprendizagem de máquina.

Os seguintes serviços AWS integram-se com AWS Lake Formation e honra Lake Formation permissões:

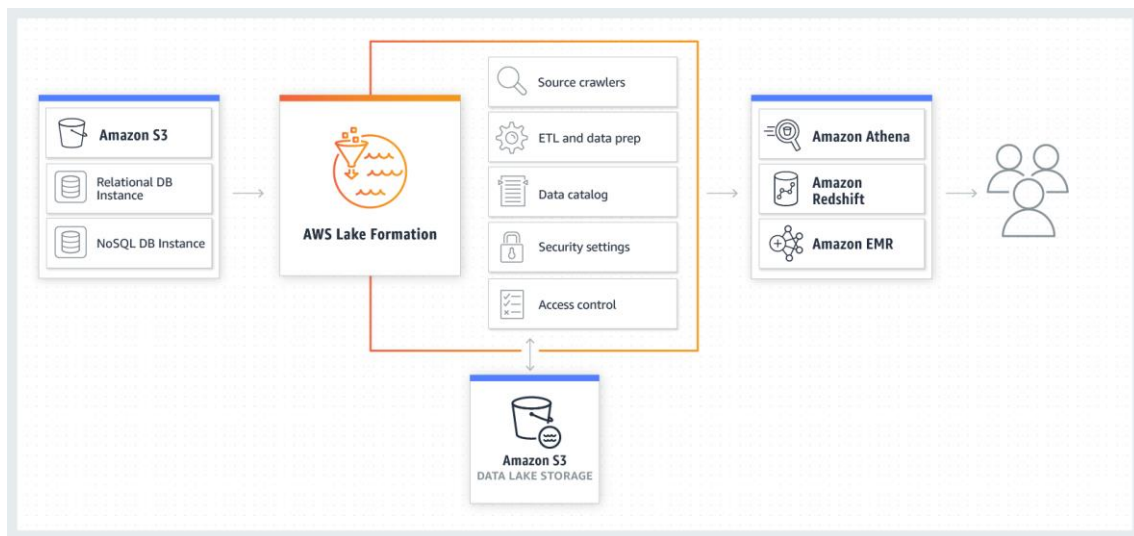
Serviço da AWS	Como está integrada
AWS Glue	AWS Glue e Lake Formation compartilham o mesmo Data Catalog. Para operações no console (como ver a lista de tabelas) e todas as operações API, usuários do AWS Glue podem acessar apenas às bases de dados e tabelas nas quais têm permissões definida no Lake Formation. AWS Glue não suporta Lake Formation permissões de coluna.
Amazon Athena	Quando usuários selecionam o AWS Glue Data Catalog no Amazon Athena no editor de consultas, podem consultar apenas as bases de dados, tabelas e colunas que têm permissões definidas em Lake Formation. As consultas que utilizam manifestos não são suportadas. Adicionalmente para quem se autentica no Athena através de AWS Identity and Access Management (IAM), Lake Formation suporta usuários que se conectam no Athena através do driver JDBC ou ODBC e se autenticam através de SAML. Os fornecedores de SAML suportados incluem a Okta e o Serviço da Federação do Microsoft Active Directory (AD FS).
Amazon Redshift Spectrum	Quando usuários do Amazon Redshift criam um esquema externo num banco de dados no AWS Glue Data Catalog, eles podem consultar apenas as tabelas e colunas nesse esquema em que têm permissões no Lake Formation. As consultas que utilizam manifestos não são suportadas.
Amazon QuickSight Enterprise Edition	Quando um usuário do Amazon QuickSight Enterprise Edition consulta um conjunto de dados numa localização Amazon S3 que está registada com Lake Formation, o utilizador deve ter o Lake Formation SELECT permissão nos dados.
Amazon EMR	As permissões do Lake Formation são aplicadas quando Jobs do Apache Spark são submetidas utilizando o Apache Zeppelin ou os Notebooks do EMR.

Como ele funciona

AWS Lake Formation permite-lhe criar, proteger e gerir mais facilmente Data Lakes. Lake Formation ajuda-o a fazer o seguinte, seja diretamente ou através de outros serviços da AWS:

- Registe o bucket Amazon Simple Storage Service (Amazon S3) e caminhos onde o seu Data Lake irá residir.
- Orquestrar fluxos de dados que ingerem, limpem, transformem e organizem os dados brutos.
- Criar e gerir um Data Catalog contendo metadados sobre fontes de dados de origem e dados no Data Lake.
- Definir políticas granulares de acesso aos dados através de um modelo de permissões de concessão/revogação sobre os metadados.

O diagrama seguinte ilustra como os dados são carregados e protegidos Lake Formation.



Terminologia do Lake Formation

Seguem-se alguns termos importantes que encontrará neste guia.

Data Lake

O Data Lake é onde os seus dados persistentes são armazenados em Amazon S3 e gerido pelo Lake Formation usando um Data Catalog. Um Data Lake armazena normalmente:

- Dados estruturados e não estruturados
- Dados em bruto e dados transformados

Para um caminho do Amazon S3 estar dentro de um Data Lake, isto deve ser registado com Lake Formation.

Acesso aos dados

O Lake Formation fornece acesso seguro e granular aos dados através de um novo modelo de permissões de concessão/revogação que amplia o AWS Identity and Access Management (IAM).

Analistas e cientistas de dados podem usar o portfólio completo de serviços analíticos e de aprendizado de máquina da AWS, como Amazon Athena, para acessar os dados. As configurações de políticas de segurança do Lake Formation ajudam a garantir que os usuários possam acessar apenas os dados que estão autorizados a acessar.

Blueprint

Um blueprint é um template de gerenciamento de dados que permite inserir dados facilmente em um Data Lake. O Lake Formation fornece vários blueprints, cada um para um tipo de fonte predefinido, como um banco de dados relacional ou logs do AWS CloudTrail. A partir de um blueprint, você pode criar um work flow ou fluxo de trabalho. Os fluxos de trabalho consistem em crawlers, jobs e gatilhos do AWS Glue que são gerados para orquestrar o carregamento e a atualização dos dados. Os blueprints usam a fonte de dados de origem, o destino dos dados e a programação como input para configurar o fluxo de trabalho.

Workflow

Um workflow é um contêiner para um conjunto de jobs, crawlers e gatilhos relacionados ao AWS Glue. Você cria o workflow em Lake Formation e ele é executado no serviço AWS Glue. O Lake Formation pode rastrear o status de um workflow como uma entidade única.

Quando você define um workflow, você seleciona o blueprint no qual ele se baseia. Você pode então executar workflows sob demanda ou em uma programação.

Os workflows que você cria em Lake Formation são visíveis no console do AWS Glue como um gráfico acíclico direcionado (Directed Acyclic Graph - DAG). Usando o DAG, você pode acompanhar o andamento do workflow e solucionar problemas.

Data Catalog

O Catálogo de Dados é seu repositório de metadados persistente. É um serviço gerenciado que permite armazenar, anotar e compartilhar metadados na nuvem AWS da mesma forma que você faria em um metastore Apache Hive. Ele fornece um repositório uniforme onde sistemas distintos podem armazenar e localizar metadados para rastrear dados em silos de dados e, em seguida, usar esses metadados para consultar e transformar os dados. Lake Formation usa o AWS Glue Data Catalog para armazenar metadados sobre Data Lakes, fontes de dados, transformações e destinos.

Os metadados sobre fontes e destinos de dados estão na forma de bancos de dados e tabelas. As tabelas armazenam informações de esquema, informações de localização e muito mais. Os bancos de dados são coleções de tabelas. Lake Formation fornece uma hierarquia de permissões para controlar o acesso a bancos de dados e tabelas no Catálogo de Dados.

Cada conta da AWS tem um catálogo de dados por região da AWS.

Dados subjacentes

Dados subjacentes referem-se às fontes de dados ou aos dados de dentro dos Data Lakes que as tabelas do Data Catalog apontam.

Principal

Um Principal é um usuário. Um principal é um usuário ou função do AWS Identity and Access Management (IAM) ou um usuário do Active Directory.

Administrador do Data Lake

Um Administrador do Data Lake é o usuário principal que pode conceder a qualquer usuário (incluindo a si mesmo) qualquer permissão em qualquer recurso do Data Catalog ou local de dados. Defina um administrador do Data Lake como o primeiro usuário do Catálogo de Dados. Esse usuário pode então conceder permissões mais granulares de recursos a outros usuários e principais.

Adicionando um local do Amazon S3 ao Data Lake

Para adicionar um local do Amazon Simple Storage Service (Amazon S3) como armazenamento em seu Data Lake, você registra o local com AWS Lake Formation. Você pode então usar as permissões do Lake Formation para controle refinado de acesso para objetos do catálogo de dados AWS Glue que apontam para este local e para os dados subjacentes no local.

Quando você registra um local, esse caminho do Amazon S3 e todas as pastas sob esse caminho são registrados.

Por exemplo, suponha que você tenha um caminho do Amazon S3 como por exemplo:

`/mybucket/accounting/sales/`

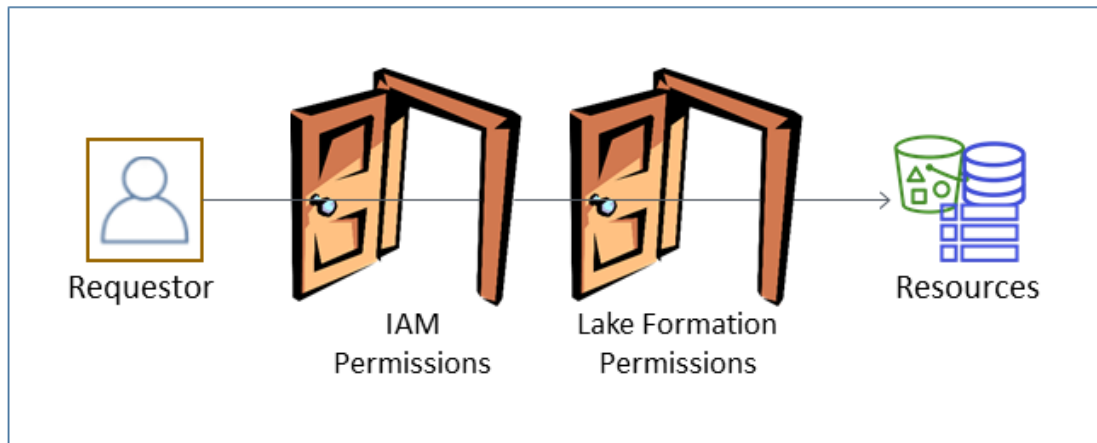
Se você registrar S3: `//mybucket/accounting`, a pasta `sales` também será registrada e sob o gerenciamento de Lake Formation.

Visão Geral do controle de Acesso do Lake Formation

O controle de acesso na AWS Lake Formation é dividido nas duas áreas:

- **Controle de acesso a metadados** - Permissões em recursos do Catálogo de Dados (permissões do Catálogo de Dados). Essas permissões permitem que os principais criem, leiam, atualizem e excluam metadados de bancos de dados e tabelas no Catálogo de Dados.
- **Controle de acesso a dados subjacente (herdados)** - Permissões em locais no Amazon Simple Storage Service (Amazon S3) (permissões de acesso a dados e permissões de localização de dados). As permissões de acesso a dados permitem que os principais leiam e gravem dados em locais subjacentes do Amazon S3. As permissões de localização de dados permitem que os principais criem bancos de dados de metadados e tabelas que apontam para locais específicos do Amazon S3.

Para ambas as áreas, Lake Formation usa uma combinação de permissões de Lake Formation e permissões AWS Identity and Access Management (IAM). O modelo de permissões IAM consiste em políticas IAM. O modelo de permissões de Lake Formation é implementado como comandos GRANT / REVOKE no estilo DBMS. Quando um principal faz uma solicitação para acessar recursos do Catálogo de Dados ou dados subjacentes, para que a solicitação seja bem-sucedida, ele deve passar nas verificações de permissão do IAM e do Lake Formation.



As permissões de Lake Formation controlam o acesso aos recursos do Data Catalog, locais do Amazon S3 e os dados subjacentes nesses locais. As permissões de IAM controlam o acesso às APIs e recursos do Lake Formation e AWS Glue. Portanto, embora você possa ter a permissão Lake Formation para criar uma tabela de metadados no Catálogo de Dados (`CREATE_TABLE`), sua operação falhará se você não tiver a permissão IAM (`glue:CreateTable`). Assim, o IAM concede acesso ao usuário aos serviços AWS e as permissões do Lake Formation concede ou restringe os acessos aos objetos (Banco de Dados, Tabelas e Colunas) do Data Lake.

Com um Data Lake, o objetivo é ter um controle de acesso refinado aos dados. No lake Formation, isso significa controle de acesso refinado aos recursos do Data Catalog e aos locais do Amazon S3. Você pode obter controle de acesso refinado com um dos seguintes métodos:

Método	Permissões Lake Formation	Permissões IAM	Comentários
Método 1	Aberto	Refinado	<p>Este é o método padrão para compatibilidade com versões anteriores com AWS Glue.</p> <ul style="list-style-type: none"> Aberto significa que a permissão especial <code>Super</code> é concedida ao grupo <code>IAMAllowedPrincipals</code>, onde <code>IAMAllowedPrincipals</code> é criado automaticamente e inclui todos os usuários e funções IAM que têm acesso aos recursos do Catálogo de Dados por suas políticas IAM, e a permissão <code>Super</code> permite que um principal execute todas as operações suportadas pelo Lake Formation. Isso efetivamente faz com que o acesso aos recursos do Data Catalog e aos locais do Amazon S3 sejam controlados exclusivamente por políticas de IAM. Refinado significa que as políticas de IAM controlam todo o acesso aos recursos do Data Catalog e aos buckets individuais do Amazon S3. <p>No console do Lake Formation, esse método aparece como Use only IAM access control.</p>
Método 2	Refinado	Granular	Este é o método recomendado

			<ul style="list-style-type: none"> • Refinado significa conceder permissões limitadas do Lake Formation a usuários principais em recursos individuais do Catálogo de Dados, locais do Amazon S3 e os dados subjacentes nesses locais. • Granular significa permissões mais amplas em operações individuais e no acesso aos locais do Amazon S3. Por exemplo, uma política IAM granular pode incluir 'glue:*' ou 'glue: Create*' em vez de 'glue: CreateTable', deixando as permissões do Lake Formation para controlar se um usuário principal pode ou não criar objetos no Catálogo de Dados.
--	--	--	--

Controle de acesso a Metadados

Para controlar o acesso aos recursos do Catálogo de Dados, assume-se utilizar o Método 2 de controle de acesso refinado com permissões de Lake Formation e controle de acesso granular com políticas IAM.

As concessões de permissões do Lake Formation seguem este formato:

```
Grant <permissões> to <usuário> on <recurso> [with grant option]
```

With the grant option, você pode permitir que o donatário conceda as permissões a outros usuários.

A tabela a seguir resume as permissões disponíveis no Lake Formation para os recursos do Catálogo de Dados. Os títulos das colunas indicam o recurso no qual a permissão é concedida.

Catálogo	Banco de Dados	Tabela
CREATE_DATABASE	CREATE_TABLE	ALTER
	ALTER	DROP
	DROP	DESCRIBE
	DESCRIBE	SELECT*
		INSERT*
		DELETE*

Por exemplo, a permissão CREATE_TABLE é concedida em um banco de dados. Isso significa que o usuário principal tem permissão para criar tabelas nesse banco de dados.

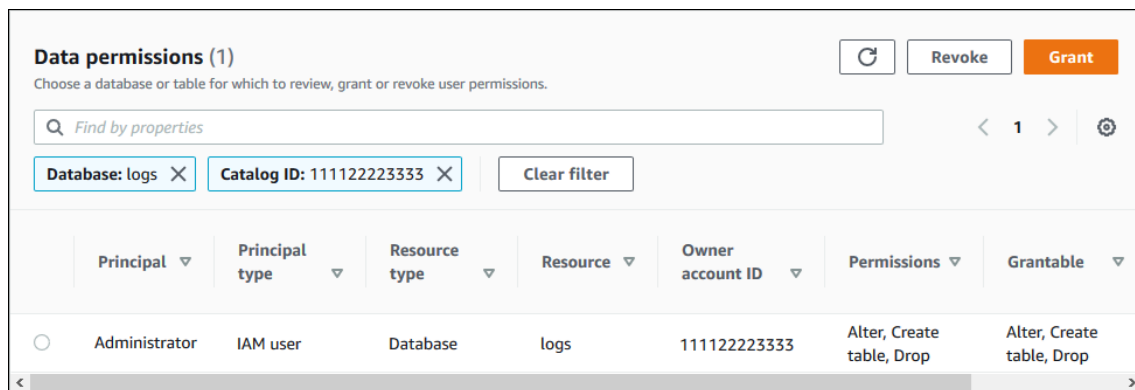
As permissões com um asterisco (*) são concedidas aos recursos do Catálogo de Dados, mas se aplicam aos dados subjacentes. Por exemplo, a permissão DROP em uma tabela de metadados permite que você exclua a tabela do Catálogo de Dados. No entanto, a permissão DELETE concedida na mesma tabela permite que você exclua os dados subjacentes da tabela no Amazon S3, usando, por exemplo, uma instrução SQL DELETE. Com essas permissões, você também pode visualizar a tabela no console do Lake Formation e recuperar informações sobre a tabela com a API AWS Glue. Portanto, SELECT, INSERT e DELETE são permissões do Catálogo de Dados e de acesso aos dados.

Ao conceder SELECT em uma tabela, você pode adicionar um filtro que inclui ou exclui uma ou mais colunas. Isso permite o controle de acesso refinado nas colunas da tabela de metadados,

limitando as colunas que os usuários de serviços integrados podem ver ao executar consultas. Esse recurso **não** está disponível usando **apenas** políticas IAM.

Também existe uma permissão especial chamada **Super**. A permissão Super permite que um usuário principal execute todas as operações suportadas pelo Lake Formation no banco de dados ou tabela na qual é concedida. Essa permissão pode coexistir com as outras permissões do Lake Formation. Por exemplo, você pode conceder Super, SELECT e INSERT em uma tabela de metadados. O usuário principal pode executar todas as ações com suporte na tabela e, quando você revoga Super, as permissões SELECT e INSERT permanecem.

É importante saber que para poder ver uma tabela do Catálogo de Dados criada por outro usuário, você deve ter pelo menos uma permissão do Lake Formation na tabela. Se você tiver pelo menos uma permissão concedida na tabela, também poderá ver o banco de dados que contém a tabela. Abaixo um exemplo de como ficam as permissões no Lake Formation



Principal	Principal type	Resource type	Resource	Owner account ID	Permissions	Grantable
<input type="radio"/> Administrator	IAM user	Database	logs	111122223333	Alter, Create table, Drop	Alter, Create table, Drop

Exercícios:

- 11) Configurando o AWS Lake Formation - <https://docs.aws.amazon.com/lake-formation/latest/dg/getting-started-setup.html>
- 12) Criação de um Data Lake a partir de uma fonte AWS CloudTrail - <https://docs.aws.amazon.com/lake-formation/latest/dg/getting-started-cloudtrail-tutorial.html>

Referências

- [1] <https://medium.com/@dayysonlima/voc%C3%AA-sabe-o-que-%C3%A9-arquitetura-serverless-1f6dd1184e5b>
- [2] <https://serverless-stack.com/chapters/pt/what-is-serverless.html>
- [3] <https://aws.amazon.com/pt/serverless/>
- [4] <https://aws.amazon.com/pt/lambda/>

