aws
certified
Solutions
Architect
Professional
DOLFINed

HYBRID CLOUD AND DATA MIGRATION SERVICE

APPLICATION DISCOVERY SERVICE

Amazon Application Discovery Service

- What it is
- How it works

What is it?

- AWS Application Discovery Service helps in planning migration to the AWS cloud by collecting usage and configuration data about on-premises servers.
- Application Discovery Service is integrated with AWS Migration Hub, which simplifies migration tracking.
- After performing discovery, the discovered servers can be viewed and grouped into applications, and then track the migration status of each application from the Migration Hub console.
- The discovered data can be exported for analysis in Microsoft Excel or AWS analysis tools such as Amazon Athena and Amazon QuickSight.

Collecting on-premise data using App Discovery Service

Two ways of performing discovery and collecting data about on-premises servers can be used:

- **Agentless discovery Connector**
  - It can be performed by deploying the AWS Agentless Discovery Connector (OVA file) through the on premise VMware vCenter.
    - After the Discovery Connector is configured, it identifies virtual machines (VMs) and hosts associated with vCenter.
    - The Discovery Connector collects the following static configuration data:
      - ➢ Server hostnames, IP addresses, MAC addresses, disk resource allocations.
      - ➢ Additionally, it collects the utilization data for each VM and computes average and peak utilization for metrics such as CPU, RAM, and Disk I/O.
  - The collected data is sent to the Application Discovery Service using Secure Sockets Layer (SSL) encryption.

- For VMware vCenter environments, the discovery connector can not "look inside" each of the VMs, and as such, cannot figure out what processes are running on each VM nor what network connections exist.
  - Therefore, if this level of detail is needed and it is required to take a closer look at some of the existing VMs in order to assist in planning your migration, then install the Discovery Agent on an as-needed basis.

Collecting on-premise data using App Discovery Service

Two ways of performing discovery and collecting data about on-premises servers can be used:

- **Agent-based discovery**
  - Can be performed by deploying the AWS Application Discovery Agent on each of the VMs and physical servers.
  - The agent installer is available for both Windows and Linux operating systems.
  - It collects static configuration data, detailed time-series system-performance information, inbound and outbound network connections, and processes that are running.
  - Data is transmitted securely by the Discovery Agents to Application Discovery Service using Transport Layer Security (TLS) encryption

- Also, for VMs hosted on VMware, both the Discovery Connector and Discovery Agent can be used to perform discovery simultaneously.

Comparison of the two discovery methods

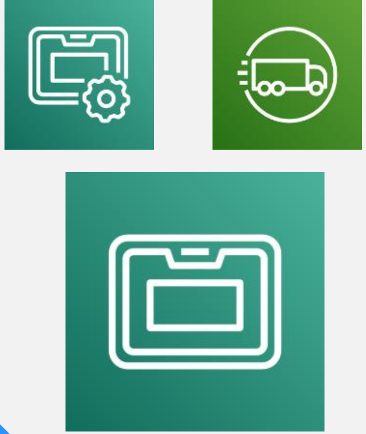
	Discovery Connector	Discovery Agent
Supported Server Types	VMs only	Both physical and VM
Deployment	Per vCenter (not per VM)	Per Server/VM
Collected Data	Static Configuration data, VM utilization metrics	VM utilization metrics, Timed performance metrics, Network in/out connections,
Supported OS	Any OS running on Vmware VMs	Windows and Linux

Data Exploration in Amazon Athena

- Data Exploration in Amazon Athena allows for the analysis of the data collected from all the discovered on premises servers by Discovery Agents in one place.

- Once Data Exploration in Amazon Athena is enabled from the Migration Hub console (or by using the StartContinousExport API) and the data collection for agents is turned on,
  - Data collected by agents will automatically get stored in the designated S3 bucket at regular intervals.
  - Amazon Athena can then be used to run pre-defined queries to analyze the time-series system performance for each server, the type of processes that are running on each server and the network dependencies between different servers.
  - Customers' own custom queries can be written using Amazon Athena, upload additional existing data sources such as configuration management database (CMDB) exports, and associate the discovered servers with the actual business applications.
  - The Athena database can be integrated with Amazon QuickSight to visualize the query outputs and perform additional analysis

- Data exploration in Athena must be turned on before seeing and starting to explore the discovered data.



AWS SNOW FAMILY

Snowball family – Why do we need it?

- Some clients need to move massive amounts of Data from on premises data centers into the cloud
  - This could be for backup purposes (Disaster Recovery perspectives)
  - This can also be as a step to migrate fully into the cloud

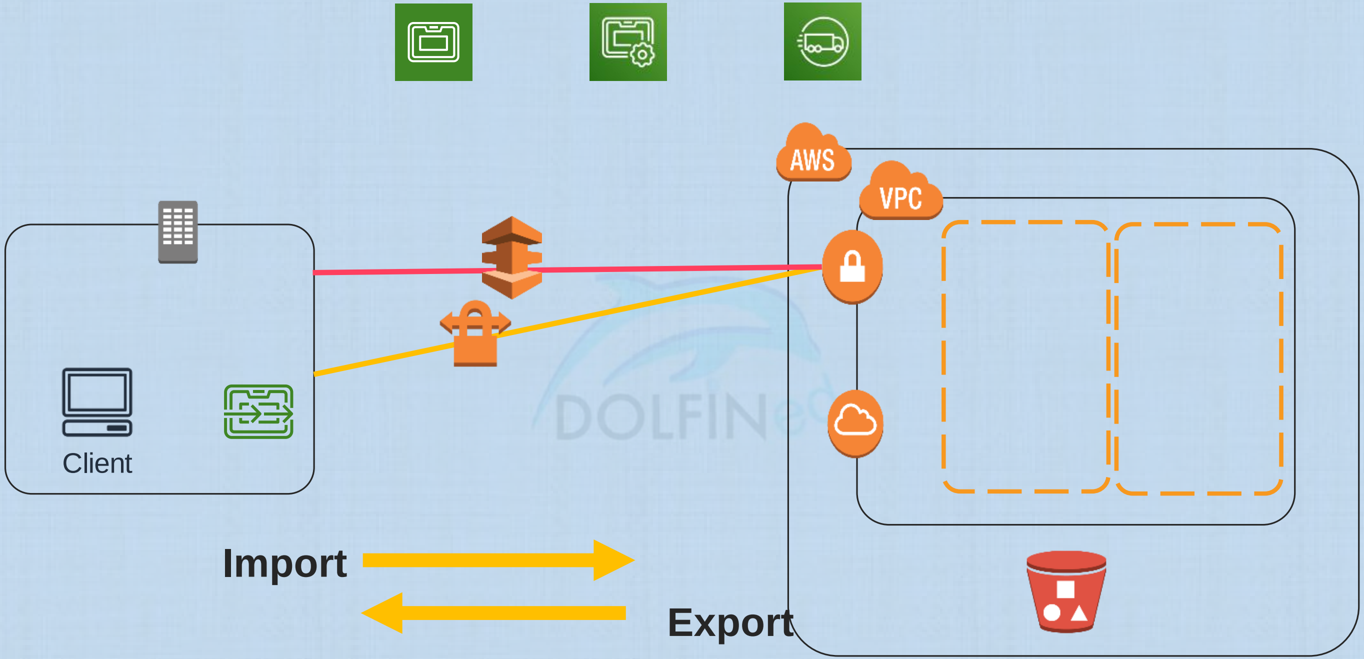
- When the existing communication link between On Premises data centers do not exist, or are not fast enough to do the migration of data in time, then we need faster means to do that.
  - This is where the Snowball family comes in handy

- The AWS Snowball service uses physical storage devices to transfer large amounts of data between Amazon Simple Storage Service (Amazon S3) and your onsite data storage location at faster-than-internet speeds.

- Snowball devices are physically rugged devices that are protected by the AWS Key Management Service (AWS KMS). They secure and protect your data in transit.

- Snowball is intended for transferring large amounts of data. If you want to transfer less than 10 TB of data between your on-premises data centers and Amazon S3, **Snowball might not be your most economical choice.**

Snowball family - Scenarios



The data is encrypted in the local memory of the client workstation before it is transferred to the Snowball. The Snowball never contains any discoverable encryption keys.

Quick refreshes on Units of Data and Data Rates

- All depends on the scenario and the requirements
  - Also what is the available link bandwidth between AWS and On premise

- Pay attention to Bytes and Bits in the scenario
  - On GB is equal to 1024 MB
  - One MB is equal to 1024 KB
  - One KB is equal to 1024 Bytes
  - One Byte is equal to 8 Bits
    - A link speed of 100 Mbps is equivalent to 12.5MB/s
    - A link speed of 1Gbps is equivalent to 0.128 GB/s or 128 MB/s

- Data speeds/rates : for example –
- 1 MB/s means ->
  - 1 x 60 MB/minute = 60 MB/min ->> Multiply by 60 to change from sec to min
    - 1 x 60 x 60 MB/hour = 3600 MB/Hr = 3.6 GB/hr ->>> Multiple by 60 to convert min into Hrs
      - ➢ 24 x 3.6 GB/day = 86.4 GB/day ->>>> Multiply by 24 to convert Hrs into Days
        - ❖ 30 x 86.4 GB/month = 2592 GB/Mo or 2.592 TB /month ->>>>> Multiple by 30 to covert Days into Months

Transfer Data – Can the existing connection do the job?

- You have a 150 TB and want to transfer it into AWS in no more than 30 days, you have 1 Gbps Internet link,
  - Can this job be done over the existing link?
  - You need to ensure the same units are used GB and the link is Gbps
    - This means the link speed in GB is 1/8 or 0.125 GB/s -> **128 MB/s**
  - 150 TB per month means 150/30 or 5 TB per day
  - 5 TB per day means 5000 GB/24 or 208.3 GB/ Hr
  - 208.3 GB/ Hr means 208.3/60 or 3.47 GB/min
  - 3.47 GB/min means 3.47/60 GB/second or .057 GB/s -> **57 MB/s**

Snowball family

There are three services within this family

- **Snowball** – a 50/80 TB appliance that gets shipped to the client by AWS
  - Multiple devices can be used when needed
- **Snowball Edge** – a 100TB device with compute (EC2 or Lambda) appliance that gets shipped to the client by AWS
  - Multiple Devices can be used when needed
- **Snowmobile** – When Exabyte scale data is to be moved
  - This is an actual truck with a huge storage trailer that come to the customer site

AWS SnowBall

- Snowball
- Snowball Edge
- Snowmobile

AWS Snowball

- It is a petabyte-scale data transport solution that uses secure appliances to transfer large amounts of data into and out of AWS. It is usually cheaper than High speed internet

- It solves the common challenge of moving large-scale data to/from AWS
  - These challenges include high network costs, long transfer times, and security concerns.

- Snowball uses multiple layers of security designed to protect your data including:
  - Tamper-resistant enclosures,
  - 256-bit encryption,
  - An industry-standard Trusted Platform Module (TPM) to ensure security and full chain of custody of the transferred data,
  - A software erasure is done by AWS once the data has been processed and the appliance is returned

- Comes in 50 (42)TB in USA only, and 80 (72)TB sizes available in all Regions

Snowball Import and Export Jobs

- Import jobs, Snowballs and jobs have a one-to-one relationship, meaning that each job has exactly one Snowball associated with it.
  - If you need additional Snowballs, you can create new import jobs or clone existing ones.

- In importing data to S3, each file becomes an object in Amazon S3 and each directory becomes a prefix.

- An export job is the transfer of any amount of data (located in Amazon S3), copied onto any number of Snowballs, and then moved one Snowball at a time into your on-premises data destination.

- When an Export job is created, it gets split into job parts.
  - Each job part is no more than 72 TB in size, and each job part has exactly one Snowball associated with it.
  - The data source for an export job is one or more amazon S3 buckets.

- In either case, AWS suggests that clients do not change, update, or delete the source data, until the destination data is verified to be completely copies and is not corrupted.

Snowball Edge

Some of the key features:

- Large amounts of storage capacity or compute functionality for devices
- Network adapters with transfer speeds of up to 100 GB/second.
- Encryption is enforced, protecting the data at rest and in physical transit.
- Snowball Edge devices support running EC2 instances using AMIs on the device

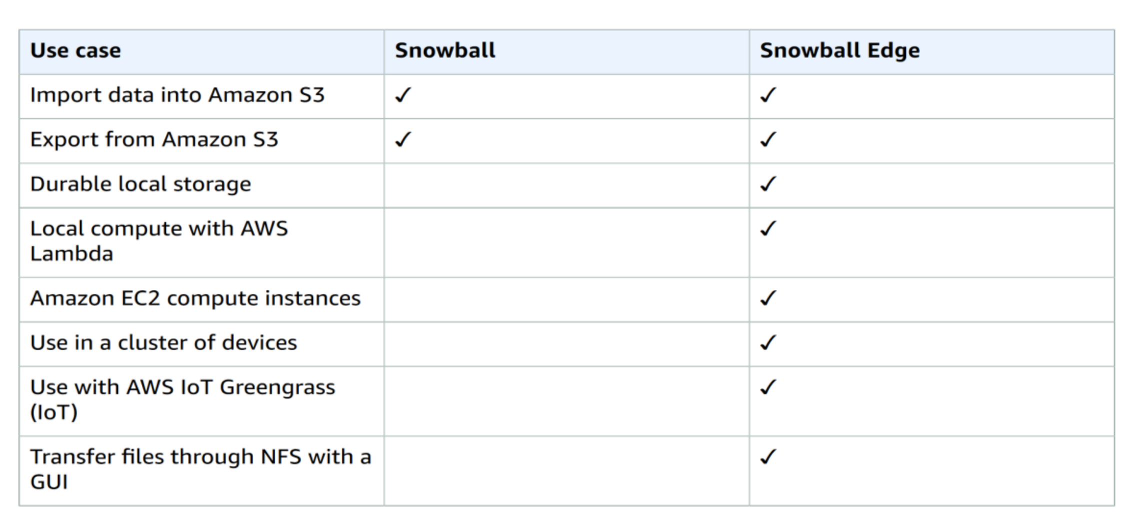
- Snowball Edge devices can be clustered for local storage and compute jobs to achieve 99.999 percent data durability across 5–10 devices, and to locally grow and shrink storage on demand.

- A file interface (NFS) can be used to read and write data to an AWS Snowball Edge device through a file share or NFS mount point.

  - Python-language Lambda functions can be associated with Amazon S3 buckets when you create an AWS Snowball Edge device job.
    - You can trigger Lambda functions based on Amazon S3 storage actions (PUT Object) made on an AWS Snowball Edge device.

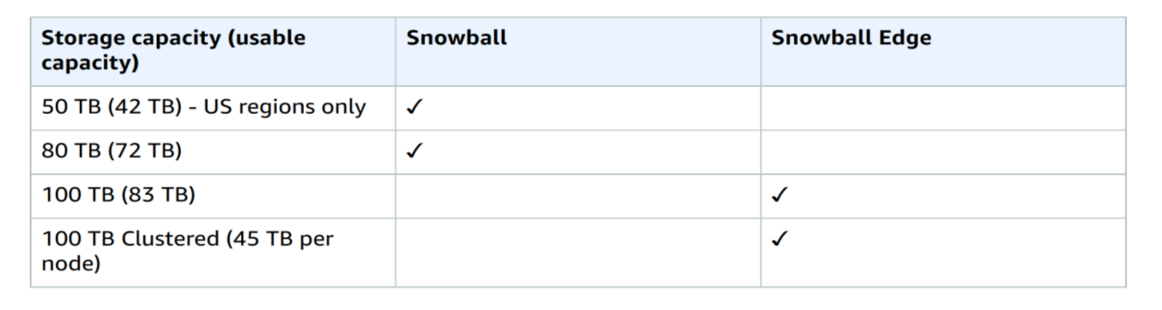
Snowball vs Snowball Edge

- Use case



Use case	Snowball	Snowball Edge
Import data into Amazon S3	✓	✓
Export from Amazon S3	✓	✓
Durable local storage		✓
Local compute with AWS Lambda		✓
Amazon EC2 compute instances		✓
Use in a cluster of devices		✓
Use with AWS IoT Greengrass (IoT)		✓
Transfer files through NFS with a GUI		✓

- Storage Capacity



Storage capacity (usable capacity)	Snowball	Snowball Edge
50 TB (42 TB) - US regions only	✓	
80 TB (72 TB)	✓	
100 TB (83 TB)		✓
100 TB Clustered (45 TB per node)		✓

AWS Snowmobile

- AWS Snowmobile is an exabyte (is 1000 Petabyte, or 1,000,000 Terabyte) scale data transfer service used to move extremely large amounts of data to AWS.
  - Up to 100 PB can be transferred per Snowmobile,
  - The snow is a 45-foot long ruggedized shipping container, pulled by a semi-trailer truck.
- Use cases: moving Video libraries, image repositories, or even a complete data center migration to the Cloud.
- Once is back at AWS, the data is imported into Amazon S3 or Glacier.
- To protect your data snow mobile supports dedicated security personnel, GPS tracking, alarm monitoring, 24/7 video surveillance, and an optional escort security vehicle while in transit.
- Transferring data with Snowmobile is secure, fast, and cost effective.
  - All data is encrypted with 256-bit encryption keys managed through AWS KMS

AWS MIGRATION HUB

AWS Migration Hub

- What is it?
- How it works

AWS Migration Hub

- AWS Migration Hub provides a single place to discover your existing DC/On-premise servers, plan migrations, and track the status of each application migration.
- The AWS Migration Hub provides visibility into the application portfolio and streamlines planning and tracking.
- The status of the servers and databases that make up each of the applications you are migrating can be seen, regardless of which migration tool is being used.
- AWS Migration Hub gives the choice to start migrating right away and group servers while migration is underway, or to first discover servers and then group them into applications.
  - Either way, each server in an application can be migrated, and progress tracking can be done from each tool in the AWS Migration Hub.
- It supports migration status updates from AWS Database Migration Service, AWS Server Migration Service, and other third party migration tools (ATAData, CloudEndure, RiverMeadow).

Discover
Migrate
Track

AWS Migration Hub

- Migration Hub monitors the status of the migrations in all AWS regions,
  - This requires that the migration tools are available in that region.

- The migration tools that integrate with Migration Hub send migration status to Migration Hub in uswest-2 where the status is aggregated and visible in a single location.
  - The migration tools do not send status unless you have authorized (that is, connected) them.

- There are three ways that can get data about the on-premise servers and applications into the AWS Migration Hub console for migration tracking;
  - Migration Hub import,
    - Uses AWS Application Discovery Service
    - It can import information about the on premises servers and applications into Migration Hub
  - The AWS Agentless Discovery Connector,
    - The Discovery Connector is a VMware appliance that can collect information about VMware VMs.
  - The AWS Application Discovery Agent.
    - The Discovery Agent is AWS software that gets installed on the on premises physical servers and VMs to capture system configuration, system performance, running processes…etc

AWS SERVER MIGRATION SERVICE

AWS Server Migration Service (SMS)

- What is it?
- Benefits
- Required IAM Permissions
- Limitations

AWS Server Migration Service (SMS)

- AWS Server Migration Service automates the migration of your on-premises **VMware** vSphere, **Microsoft Hyper-V/SCVMM**, and **Azure virtual machines** to the AWS Cloud.

- AWS SMS incrementally replicates your server VMs as cloud-hosted Amazon Machine Images (AMIs) ready for deployment on Amazon EC2.

- Working with AMIs, you can easily test and update your cloud-based images before deploying them in production.

- There is no additional fee to use Server Migration Service.
  - Customers pay the standard fees for the S3 buckets, EBS volumes, and data transfer used during the migration process, and for the EC2 instances used to launch the AMIs on AWS.

- The Server Migration Connector is a FreeBSD VM that you install in your on-premises virtualization environment.
  - There is a connector for each of Vmware, MS HyperV and Azure.
  - It uses port 443 to securely replicate the VMs to AWS
  - Once configured it can import your on premise servers which then becomes the SMS Servers Catalog

Benefits

- Simplify the cloud migration process.
  - After initiating the migration, AWS SMS manages the migration process fully.
  - SMS automatically replicates volumes of live servers to AWS.
  - SMS periodically creates new AMIs (When replication Job is configured as such, not a one time job)

- Orchestrate multi-server migrations.
  - You can schedule replication and track progress of a group of servers (VMs) that together make an application.
  - Using the SMS Console, you can
    - Schedule initial replications,
    - Configure replication intervals, and
    - Track progress for each server using the console.
    - When you launch a migrated application, you can apply customized configuration scripts that run during startup.

- Test server migrations incrementally
  - With support for incremental replication, AWS SMS allows fast, scalable testing of migrated servers.

Benefits

- Because AWS SMS replicates incremental changes to your on-premises servers and transfers only the delta to the cloud, you can test small changes iteratively and save on network bandwidth.
- SMS Supports Windows, as well as several major Linux distributions.
- Minimize downtime.
  - Incremental AWS SMS replication minimizes the business impact associated with application downtime during the final cutover.
- SNS can be used to send notifications to a list of recipients when the replication job has completed, failed, or been deleted.
- Enable AMI Encryption, if chosen when configuring the replication
  - AWS SMS encrypts the generated AMIs. Your default CMK is used unless a non-default CMK is specified.

Required IAM Permissions & Service Limits

- Admin permissions on SMS to configure SMS and schedule jobs…etc
- IAM role for the SMS Connector to access AWS
- IAM Service Role for AWS SMS service itself,
  - To allow SMS to place migrated resources into your Amazon EC2 account

- 50 concurrent VM migrations per account (Soft Limit)
- 90 days of service usage per VM (not per account) (Soft Limit),
  - It begins from the initial replication of a VM.
  - AWS terminates an ongoing replication after 90 days unless a customer requests a limit increase.
- 50 concurrent application migrations per account, with a limit of 10 groups and 50 servers in each application

AWS SMS – Migrating Applications

- How are applications defined for SMS
- Grouping VMs into application component groups
- AWS SMS and Migration Hub
- Monitoring and Logging

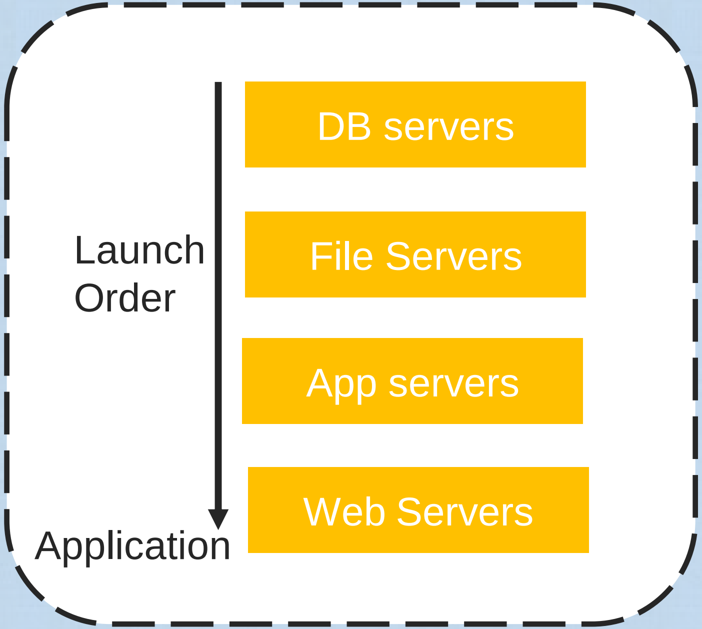
Migrating Applications with AWS SMS

- AWS Server Migration Service supports the automated migration of multi-server application stacks from on-premises data center to Amazon EC2.
  - ○ Please note that SMS will not distinguish between the different actual applications,
    - ▪ Rather, it treats the application based on the group of VMs that constitute the application tiers

- Application migration replicates all the servers in an application as AMIs and generates an AWS CloudFormation template to launch them in a coordinated fashion.

- Applications can be further subdivided into groups that allow you to launch tiers of servers in a defined order.

- The on-premises servers can be replicated to AWS for up to 90 days per server. After which it gets terminated by AWS

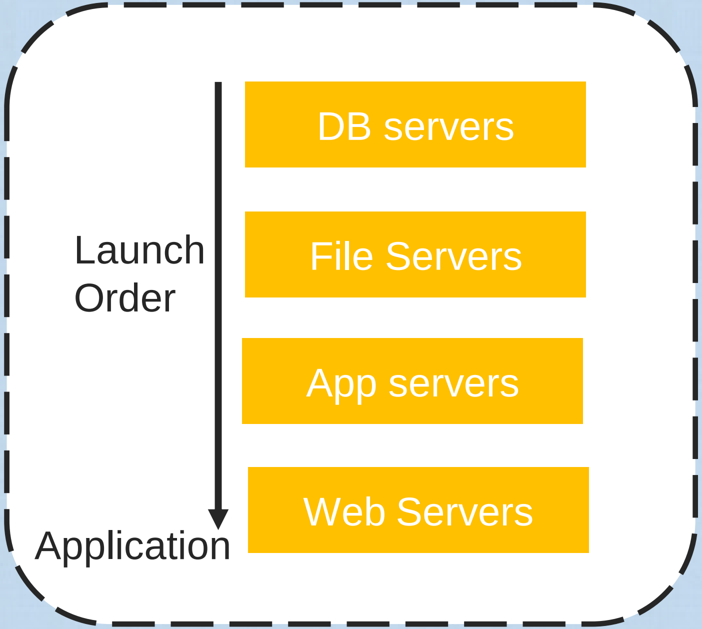


Migrating Applications with AWS SMS

- The application to be migrated where,
  - The application is divided into four groups, each with a number of servers.
  - The AWS CloudFormation template starts the servers in the following order (Launch Order):
    - Databases, file servers, App servers, and web servers.

- After the servers are organized into applications and launch groups, a replication frequency can be specified, configuration scripts can be provided, and a target VPC/Subnet/Security Groups/Internet accessibility, can be configured to launch them.

- When launching an application, AWS SMS configures it based on the generated template.



AWS SMS Monitoring and Logging

- AWS SMS integrates with Cloud Trail for API calls logging
- AWS SMS integrates with AWS CloudWatch
  - Amazon CloudWatch Events can be used with AWS Server Migration Service to automate actions based on the migration workflow.
  - This requires the creation of an IAM policy for Lambda to assume, a Lambda function to handle the event, and a CloudWatch Events rule that matches incoming events and routes them to the Lambda function.

AWS SMS and Migration HUB

- AWS SMS Application Migration supports the import and migration of applications discovered by AWS Migration Hub.

- Application Import from Migration Hub
  - SMS imports application-related servers from AWS Migration Hub only if they exist in the SMS Server Catalog.
    - As a result, some applications may only be partially migrated.
      - If none of the servers in a Migration Hub application exist in the SMS Server Catalog, the import will fail silently and the application will not be visible in SMS.
    - Imported applications can be migrated but cannot be edited in SMS.
      - They can, however, be edited in Migration Hub.

- If integration between AWS SMS and AWS Migration Hub is enabled, the SMS server catalog will be also visible on Migration Hub.

AWS DATABASE MIGRATION SERVICE (DMS)

AWS DataBase Migration Service (DMS)

- What is it?
- Features and Benefits

What is it?

- AWS DMS is a web service that can be used to migrate data from a source database/datastore to a target database/datastore. These two data stores are called (in AWS DMS terms) source and target.

- AWS DMS allows for migrating between source and target endpoints that use the same database engine, or different database engines.

- To use AWS DMS it is required that one of the two endpoints (Source or Target) **must be on an AWS service.**
  - Hence, AWS DMS can’t be used to migrate from an on-premises database to another on-premises database directly.

What is it? (Cont.)

- AWS DMS facilitates migrations of:
  - Relational Databases
  - Data Warehouses
  - NoSQL Databases
  - Other types of datastores

- AWS DMS can be used to migrate data:
  - From On-premise into AWS
  - Indirectly between different on-premise instances, through the use of an AWS-based setup
  - Between combinations of Cloud and On-premises setups

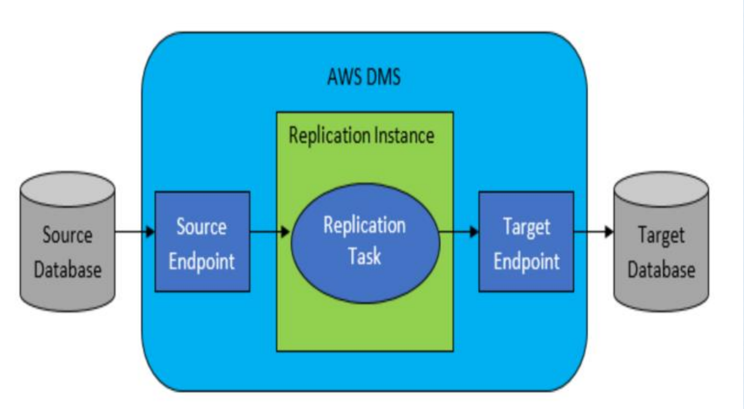
- AWS DMS can be used to perform one-time migrations, OR
  - Ongoing replication such that the ongoing changes are sync'd between the source and target.

Features and benefits..

- AWS DMS fully and automatically manages the deployment, management, and monitoring of all hardware and software needed for your migration (faster and easier migrations).
- AWS DMS allows for scaling migration resources up or down based on migration needs.
- Only pay for involved resources only while you use them, no upfront or licensing costs.
- Automatic failover. If the primary replication server fails for any reason, AWS DMS provides a backup replication server automatically which will take over with little or no interruption of migration service.
- AWS DMS can help in switching to another DB engine that is most cost-effective, or better performing.
  - AWS DMS allows to move to RDS, DynamoDB, Redshift, S3 as targets.
  - It also allows to maintain like-to-like DB engine between Source and Target.
- AWS DMS supports data security both in-transit and at rest.
  - Encryption at rest is supported using AWS KMS
  - SSL can be used to encrypt/protect data in-transit
- Supports Multi-AZ deployment of two Replication instances across two AZ’s in the same region.
- AWS DMS sources (as of now) include most popular DBMS engines as data sources, including
  - Oracle, Microsoft SQL Server, MySQL, MariaDB, PostgreSQL, Db2 LUW, SAP, MongoDB, and Amazon Aurora.
- AWS DMS targets includes a broad coverage of available target engines including
  - Oracle, Microsoft SQL Server, PostgreSQL, MySQL, Amazon Redshift, SAP ASE, Amazon S3, and Amazon DynamoDB.
- You can migrate from any of the supported data sources to any of the supported data targets.
  - AWS DMS supports fully heterogeneous data migrations between the supported engines.

How it works?

- AWS DMS is an AWS instance/server (Replication Instance), in the AWS Cloud, that AWS DMS configures to run the replication software.
- Source and Target Endpoints are nothing more than connection information into the respective database.
- Configuring/Creating source and target connections (to source and target endpoints) is what AWS DMS requires to know where to extract the data from and where to load it to.
- Scheduling replication tasks on the DMS instance, will move the data from source to target.
  - ○ You can run one or more tasks on a replication instance
  - ○ Replication instances come with 50GB or 100GB of data storage
- You can create the target tables and primary keys manually, or AWS DMS can take care of that (using AWS SCT. AWS DMS can create some or all of the tables, indexes, views, … at the target).
- Creating a *replication task that* specifies the actual data tables to migrate and data transformation rules to apply.
- AWS DMS manages running the replication task and provides the status on the migration process.



How it works? (cont.)

- To perform a database migration,
    - AWS DMS connects to the source data store,
    - Reads the source data, and formats the data for consumption by the target data store.
    - It then loads the data into the target data store.
    - Most of this processing happens in memory, though large transactions might require some buffering to disk.
    - Cached transactions and log files are also written to disk.

- An AWS DMS task can consist of three major phases:
    - The full load of existing data
    - The application of cached changes
    - Ongoing replication

AWS DMS and AWS Schema Conversion Tool (SCT)

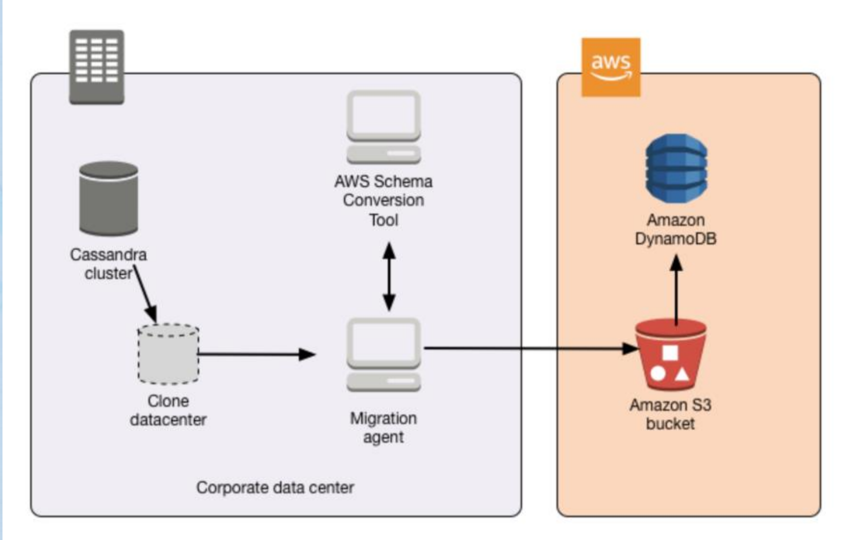
- AWS DMS doesn't perform schema or code conversion.
- AWS DMS attempts to create the target schema.
- AWS DMS doesn't create a target Oracle schema for security reasons.
- For MySQL database targets, extra connection attributes can be used to have AWS DMS migrate all objects to the specified database and schema or create each database and schema as it finds the schema on the source.
- In most cases, when performing a migration, most or all of the source schema is also migrated.

- In a **homogeneous migration** (between two databases of the same engine type),
  - The schema can be migrated using your engine’s native tools to export and import the schema itself, without any data.
  - AWS DMS will move the data
- In a **heterogeneous migration** (between two databases that use different engine types),
  - AWS Schema Conversion Tool (AWS SCT) can be used to generate a complete target schema for you.
  - It can create a target schema and also can generate and create an entire schema: tables, indexes, views, and so on.

AWS DMS and Schema Conversion Tool

- If the migration involves changing the database engine,
  - ○ AWS Schema Conversion Tool (AWS SCT) can be used to translate the existing database schema to the new platform,
  - ○ Then AWS SCT uses AWS DMS to migrate the data.

- **Example Use case:**
  - ○ Migrating **Apache Cassandra (NoSQL)** from On-Premise to **DynamoDB** on AWS.
    - ➢ Using the AWS SCT data extraction agent (runs on EC2 instance) for Cassandra.
    - ➢ SCT agent extracts Cassandra data (Clone DC) and writes it to S3
    - ➢ Then using AWS DMS, SCT orchestrates the workflow, reading Cassandra data from S3, creating the DynamoDB tables, transform the data, then write it to DynamoDB



AWS DataBase Migration Service (DMS)

- AWS DMS Components – Replication Instance
- Network Setup for the Replication Instance
- AWS DMS Multi AZ feature

AWS DMS Components – Replication Instance

- AWS DMS replication instance is an AWS EC2 that hosts one or more replication tasks.

- AWS DMS provides a variety of replication instances to choose from, depending on the requirements

- Depending on the Amazon EC2 instance class selected, the replication instance comes with either 50 GB or 100 GB of data storage.
  - This amount is usually sufficient for most customers.
  - If the migration involves large transactions or a high-volume of data changes, you can increase the base storage allocation.
  - Change data capture (CDC) may cause data to be written to disk, depending on how fast the target can write the changes.

Different use cases for VPC and Networking involved with AWS DMS

- Source, Replication instance and Target are all in one VPC
- Source in a VPC and Target in another, replication instance can be in either one
  - Use VPC peering to connect the two VPCs
- Source (or Target) is on-premise, Replication Instance and Target (or source) in an AWS VPC
  - Connect through VPN or Direct connect
  - Also connection can be through direct internet (security concerns?)

AWS DMS Multi-AZ

- AWS DMS supports high availability and failover using a Multi-AZ deployment.

- In a Multi-AZ deployment, AWS DMS automatically provisions and maintains a standby replica of the replication instance in a different Availability Zone.

- In a Multi-AZ AWS DMS deployment, the primary replication instance is synchronously replicated to the standby replica.

- In the event the primary replication instance fails or becomes unresponsive, the standby resumes any running tasks with minimal interruption.

- However, since the primary is constantly replicating its state to the standby, Multi-AZ deployment are expected to incur some performance overhead.

AWS DMS Replication Instance and Virtual Private Cloud (VPC)

- AWS creates the replication instance in an AWS VPC of your choice (Default VPC, Custom VPC)
  - The VPC must have two subnets in at least one Availability Zone.
- The security group associated with the Replication Instance ENI must allow all egress traffic from your VPC (allow all outbound),
  - This will ensure that the replication instance can initiate any required communication to the source/target database endpoints.
- Similarly on your source and target DBs' associated security groups and NACLs, where applicable, ensure that inbound communication from the replication instance is allowed
- The source and target endpoints access the replication instance that is inside the VPC either by connecting to the VPC or by being inside the VPC.
  - Depending on the network configuration you are using, you can:
    - Use the replication instance VPC security group,
    - Use the replication instance's private or public IP address, or
    - Use the NAT gateway's public IP address.
- These connections form the network medium that facilitates (is used for) data migration.

AWS DataBase Migration Service (DMS)

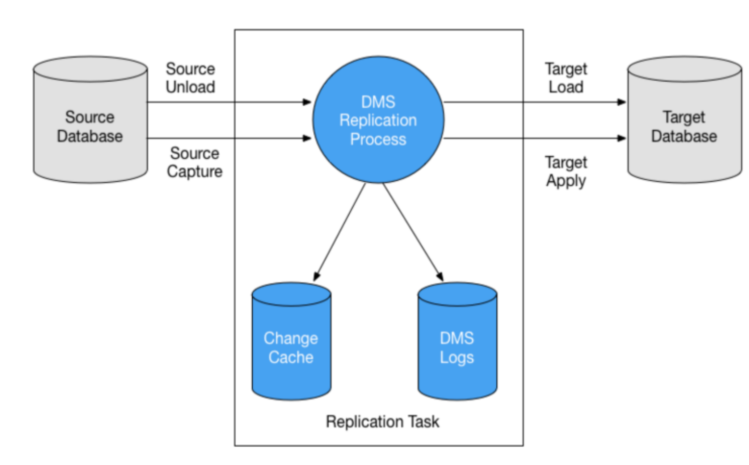
- AWS DMS Components – Endpoints and Replication Tasks

AWS DMS Components - Endpoints

- An endpoint provides connection, data store type, and location information about your data store.
- AWS DMS uses an endpoint to access the source or target data store.
- The specific connection information depends on the data store,
- A single endpoint can be used by more than one replication task.
- Example use case:
  - It is required to move a database existing on-premise; the database has two distinct Apps using the DB as the source DB. It is required to migrate the applications and separate the DB to two, one for each app.
    - Create separate AWS DMS tasks, one for each application (to migrate the respective set of tables for each app)
    - The same AWS DMS endpoint can be used since it is the same source DB

AWS DMS Components – Replication Tasks

- You use an AWS DMS replication task to move a set of data from the source endpoint to the target endpoint. Creating a replication task is the last step you need to take before you start a migration.
- When you create a replication task, you specify the following task settings:
  - Replication instance – the instance to host and run the task
  - Source endpoint
  - Target endpoint
  - Migration type options
  - Target table preparation mode options
  - LOB (Large binary OBjects) mode options
  - Table mappings
  - Data Transformation
  - Data validation
  - Amazon CloudWatch logging



AWS DMS – Task Migration Options

- **Full load (Migrate existing data)**
  - This option migrates the data from the source database to the target database, creating tables when necessary.
  - If the use case can afford an outage long enough to copy the existing data, this option is a good one to choose.

- **Full load + CDC (Migrate existing data and replicate ongoing changes)**
  - It does a full data load while capturing changes on the source.
  - After full load is complete, the data captured changes are applied to the target.
  - With ongoing application of changes, this would reach a steady state.
  - Once steady state is reached, applications can be shutdown, and restarted to point to the target

- **CDC only (Replicate data changes only)**
  - If another tool/method is used to copy the data from source to target (AWS DMS not used for that)
    - In this case, AWS DMS can be used to replicate changes to the source that happened starting when the buld load started, which will bring and keep the source and target databases in sync.
  - In homogeneous migration (like to like DB migration), using native export/import tools might be more efficient at loading the bulk data.

AWS DataBase Migration Service (DMS)

- AWS DMS Supported Sources
- AWS DMS Supported Targets

Supported Source/Targets for AWS DMS

DB or Data Store	Source for DMS?	Source Located at	Target for DMS?	Target located at
MS Azure SQL	Yes	MS Azure	No	
MongoDB	Yes	On premise, EC2 instance,	No	
DB2	Yes	On premise, EC2 instance,	No	
Redshift	**No**		Yes	AWS
ElasticSearch (ES)	**No**		Yes	AWS
Kinesis Data Streams	**No**		Yes	AWS
DynamoDB (w/mongoDB compatibility)	**No**		Yes	AWS
DocumentDB	**No**		Yes	AWS
Amazon S3	Yes		Yes	AWS
MariaDB (Supported as MySQL compatible)	Yes	On premise, EC2 instance, RDS	Yes	On Premise, EC2 instance, RDS
Aurora	Yes	RDS (Amazon Aurora with MySQL compatibility, and Amazon Aurora w/ PostgreSQL compatibility)	Yes	RDS (Amazon Aurora with MySQL compatibility, and Amazon Aurora w/ PostgreSQL compatibility)
Oracle	Yes	On premise, EC2 instance, RDS	Yes	On Premise, EC2 instance, RDS
MySQL		On Premise, EC2 instance, RDS	Yes	On Premise, EC2 instance, RDS
MS SQL Server	Yes	On Premise, EC2 instance, RDS	Yes	On Premise, EC2 instance, RDS
PostgreSQL	Yes	On premise, EC2 instance, RDS,		On Premise, EC2 instance
SAP Adaptive Server Enterprise (ASE)	Yes	On Premise, EC2 instance	Yes	On Premise, EC2 instance

MS Azure SQL Server as source/Target

As a Source:

- With AWS DMS, you can use Microsoft Azure SQL Database as a source in much the same way as you do SQL Server. AWS DMS supports, as a source, the same list of database versions that are supported for SQL Server running on-premises or on an Amazon EC2 instance.
- AWS DMS doesn't support change data capture operations (CDC) with Azure SQL Database.
- Not supported as an AWS DMS Target

As a Target:

- Not supported as an AWS DMS Target

Using IBM Db2 LUW as source

- AWS DMS can be used to migrate data from an IBM Db2 for Linux, Unix, and Windows (Db2 LUW) database to any supported target database using AWS Database Migration Service (AWS DMS).

- Secure Sockets Layer (SSL) can be used to encrypt connections between the source Db2 LUW endpoint and the replication instance.

MongoDB as Source for AWS DMS

Important MongoDB database concepts:

- A record in MongoDB is a *document*, which is a data structure composed of field and value pairs.
  - The value of a field can include other documents, arrays, and arrays of documents. A document is roughly equivalent to a row in a relational database table.

- A *collection* in MongoDB is a group of documents, and is roughly equivalent to a relational database table.

- Internally, a MongoDB document is stored as a binary JSON (BSON) file in a compressed format that includes a type for each field in the document. Each document has a unique ID.

AWS DMS supports MongoDB versions 2.6.x and 3.x as a database source.

MongoDB as Source

As a Source:

- Supported as a source on premise or on EC2 instances.
- AWS DMS supports two migration modes when using MongoDB as a source.
  - **Document mode**

In document mode, the MongoDB document is migrated as is, meaning that the document data is consolidated into a single column in a target table.

Document mode is the default for DMS.

- **Table mode**

In table mode, AWS DMS transforms each top-level field in

a MongoDB document into a column in the target table.

If a field is nested, AWS DMS flattens the nested values into a single column. AWS DMS then adds a key field and Data types to the target table's column set.

For each MongoDB document, AWS DMS adds each key and type to the target table’s column set.

- Not supported as a Target

AWS DataBase Migration Service (DMS)

- AWS DMS Supported Targets that can Not be Sources for DMS

Using Amazon DocumentDB as Target

DocumentDB is NOT supported as source

As a Target:
AWS DMS can be used to migrate data to Amazon DocumentDB (with MongoDB compatibility) from any of the source data engines that AWS DMS supports.

The source engine can be on:

- An Amazon-managed service such as Amazon RDS, Aurora, or Amazon S3.
- A self-managed database ex.**MongoDB running on Amazon EC2 or on-premises**.

AWS DMS can be used to replicate source data to Amazon DocumentDB databases, collections, or documents.

If the source endpoint is a relational database, AWS DMS maps database objects to Amazon DocumentDB as follows:

- A relational database, or database schema, maps to an Amazon DocumentDB database.
- Tables within a relational database map to collections in Amazon DocumentDB.
- Records in a relational table map to documents in Amazon DocumentDB.
  - ○ Each document is constructed from data in the source record.
- If the source endpoint is Amazon S3, then the resulting Amazon DocumentDB objects correspond to AWS DMS mapping rules for Amazon S3.

Amazon DynamoDB as a Target for AWS DMS

- Can NOT be a source for AWS DMS

As Target:

- AWS DMS can be used to migrate data to an Amazon DynamoDB table.
- AWS DMS supports using **a relational database or MongoDB as a source**.
- Object mapping is used to migrate data from a source database to a target DynamoDB table.
  - Object mapping helps in determining where the source data is located in the target.
- When creating tables in the target endpoint, AWS DMS creates as many tables as in the source database endpoint. It also configures DynamoDB parameters, for ex. It sets the default WCU/RCU to 200.
- **When migrating from a Relational Database to a DynamoDB Table ,** AWS DMS supports migrating data to DynamoDB scalar data types.
  - Restructuring how the data is stored might be required in this case
  - In the time being, AWS DMS supports single table to single table restructuring to DynamoDB scalar type attributes.

Redshift as a Target for AWS DMS

- Can NOT be a source for AWS DMS

As Target:

- With an Amazon Redshift database as a target for AWS DMS, you can migrate data from all of the other supported source databases.
- The Amazon Redshift target cluster must be in the same AWS account and Region as the replication instance.
- During migration, AWS DMS first moves data to an Amazon S3 bucket.
  - Once the files are uploaded to the S3 bucket, AWS DMS then transfers them to the proper tables in the Amazon Redshift data warehouse.
- AWS DMS creates the S3 bucket in the same AWS Region as the Redshift target.
- Encrypting the target data pushed to Amazon S3 before it is copied to Amazon Redshift is supported.
  - Create and use custom AWS KMS keys for that.
- The Amazon Redshift endpoint provides full automation for the following:
  - Schema generation and data type mapping, Full load of source database tables
  - Incremental load of changes made to source tables
  - Application of schema changes made to the source tables
  - Synchronization between full load and change data capture (CDC) processes.

Redshift as Target (cont.)

- Enhanced VPC Routing affects the way that Amazon Redshift accesses other resources,
- If Enhanced VPC Routing is used with your Amazon Redshift target, all COPY traffic between your Amazon Redshift cluster and your data repositories goes through your VPC.
- COPY commands might fail if the VPC has not been configured correctly.

Using Kinesis data streams as Target for AWS DMS

- Can NOT be a source for AWS DMS

As Target:

- AWS DMS can be used to migrate data to an Amazon Kinesis data stream.
- Kinesis data streams can be used to collect and process large streams of data records in real time.
- AWS Database Migration Service publishes records to a Kinesis data stream using JSON.
- During conversion, AWS DMS serializes each record from the source database into an attribute-value pair in JSON format.
- You use object mapping to migrate your data from any supported data source to a target stream.
- With object mapping, you determine how to structure the data records in the stream. You also define a partition key for each table, which Kinesis Data Streams uses to group the data into its shards.

Using Amazon Elasticsearch Service Cluster as Target

- Can NOT be a source for AWS DMS

As a Target:

- Amazon ES is a managed service that makes it easy to deploy, operate, and scale an Elasticsearch cluster.
- You can use AWS DMS to migrate data to Amazon Elasticsearch Service (Amazon ES).
- Elasticsearch uses indexes and documents.
  - An *index* is a collection of documents,
  - *A document* is a JSON object containing scalar values, arrays, and other objects.
- Elasticsearch provides a JSON-based query language, so that you can query data in an index and retrieve the corresponding documents.
- When AWS DMS creates one index for each table from the source endpoint.
- Configure your Elasticsearch cluster with compute and storage resources that are appropriate for the scope of your migration. We recommend that you consider the following factors, depending on the replication task you want to use:

Using Amazon Elasticsearch Service Cluster as Target

- Can NOT be a source for AWS DMS

As a Target:

- Amazon ES is a managed service that makes it easy to deploy, operate, and scale an Elasticsearch cluster.
- You can use AWS DMS to migrate data to Amazon Elasticsearch Service (Amazon ES).
- In Elasticsearch, you work with indexes and documents. An *index* is a collection of documents, and a *document* is a JSON object containing scalar values, arrays, and other objects. Elasticsearch provides a JSON-based query language, so that you can query data in an index and retrieve the corresponding documents.
- When AWS DMS creates indexes for a target endpoint for Amazon Elasticsearch Service, it creates one index for each table from the source endpoint. The cost for creating an Elasticsearch index depends on several factors. These are the number of indexes created, the total amount of data in these indexes, and the small amount of metadata that Elasticsearch stores for each document.
- Configure your Elasticsearch cluster with compute and storage resources that are appropriate for the scope of your migration. We recommend that you consider the following factors, depending on the replication task you want to use:
- For a full data load, consider the total amount of data that you want to migrate, and also the speed of the transfer.
- For replicating ongoing changes, consider the frequency of updates, and your end-to-end latency requirements.
- Also, configure the index settings on your Elasticsearch cluster, paying close attention to the shard and replica count.

MySQL as Source/Target for AWS DMS

As a Source:

- Data can be migrated from any MySQL-compatible database (MySQL, MariaDB, or Amazon Aurora MySQL) using AWS Database Migration Service.
- All AWS-managed MySQL databases (Amazon RDS for MySQL, Amazon RDS for MariaDB, Amazon Aurora MySQL) are supported as sources for AWS DMS.
- Use SSL to encrypt connections between your MySQL-compatible endpoint and the replication instance.

Migrating from MySQL to MySQL Using AWS DMS

- For a homogeneous migration, when migrating from a MySQL database to a MySQL database, native tools can be more effective.
- For a heterogeneous migration, where you are migrating from a database engine other than MySQL to a MySQL database, AWS DMS is almost always the best migration tool to use.

As a Target:

- Data can be migrated to any MySQL-compatible database using AWS DMS, from any of the source data engines that AWS DMS supports.
- When migrating **to an on-premises** MySQL-compatible database, then AWS DMS requires that the source engine reside within the AWS ecosystem.
  - The source DB can be on
    - Amazon RDS, Amazon Aurora, Amazon EC2 self managed DB, or Amazon S3.
- Secure Sockets Layer (SSL) can be used to encrypt connections between your MySQL-compatible endpoint and the replication instance.

MySQL as Source (cont.)

- AWS DMS can migrate data from, for example, a source MySQL database that is on premises to a target Amazon RDS for MySQL or Amazon Aurora (MySQL) instance.

- AWS recommends that you use native MySQL database migration tools (such as mysqldump) to carry the following:
  - Migrate data from an existing MySQL or MariaDB database to an Amazon RDS MySQL or MariaDB DB.
  - Copy the database with mysqldump and pipe it directly into the Amazon RDS MySQL or MariaDB DB instance.
  - The mysqldump command-line utility is commonly used to make backups and transfer data from one MySQL or MariaDB server to another.

ORACLE DB as Source/Target for AWS DMS

ORACLE DB as a Source:

- Data can be migrated from one or many Oracle databases using AWS DMS to any of the Targets supported by AWS DMS
  - This includes:
    - Self managed Oracle DB source
    - AWS Managed Oracle DB source

- Secure Sockets Layer (SSL) can be used to encrypt connections between the Oracle source endpoint and the replication instance.

- AWS DMS supports Oracle transparent data encryption (TDE) tablespace encryption and AWS Key Management Service (AWS KMS) encryption (when used with Oracle LogMiner, an Oracle tool for querying logged changes to the DB).
  - All other forms of encryption aren't supported

ORACLE DB as Target:

- You can migrate data to Oracle database targets using AWS DMS, either from another Oracle database or from one of the other supported databases.

- Secure Sockets Layer (SSL) can be used to encrypt connections between Oracle target endpoint and the replication instance.

- AWS DMS does not create schema on the target Oracle database.
  - Client team has to create any schemas required on the target Oracle database.

Microsoft SQL Server as Source/Target for AWS DMS

As a source:

- With a SQL Server database as a source, data can be migrated to either another SQL Server database or one of the other supported databases.
- AWS DMS supports, as a source, **on-premises** and **Amazon EC2 instance** databases for Microsoft SQL Server.
  - The Enterprise, Standard, Workgroup, and Developer editions are supported.
  - **The Web and Express editions aren't supported.**
- AWS DMS supports, as a source, **Amazon RDS DB instance databases for SQL**
  - The Enterprise and Standard editions are supported.
- Secure Sockets Layer (SSL) can be used to encrypt connections between your SQL Server endpoint and the replication instance.

As a Target:

- You can migrate data to PostgreSQL databases using AWS DMS, either from another PostgreSQL database or from one of the other supported databases.
- PostgreSQL versions 9.4 and later are supported for on-premises, Amazon RDS, Amazon Aurora with PostgreSQL compatibility, and EC2 instance databases.

Using PostgreSQL server as Source/Target for AWS DMS

As a Source:

- You can migrate data from one or many PostgreSQL databases using AWS DMS.
- With a PostgreSQL database as a source, you can migrate data to either another PostgreSQL database or one of the other supported databases.
- AWS DMS supports a PostgreSQL version 9.4 and later database as a source for
  - ○ **On-premises databases,**
  - ○ **DB on an EC2 instance,**
  - ○ **Databases on an Amazon RDS DB** instance, and
  - ○ **Databases on an Amazon Aurora DB instance with PostgreSQL compatibility.**
- Secure Sockets Layer (SSL) can be used to encrypt connections between your PostgreSQL endpoint and the replication instance.

As a Target:

- AWS DMS can migrate data from, for example, a source PostgreSQL database that is **on premises** to a target **Amazon RDS** for PostgreSQL or **Amazon Aurora (PostgreSQL) instance.**
- You can use an Amazon RDS for PostgreSQL DB instance or **Read Replica as a source for AWS DMS.**
- For homogeneous migration for the entire DB, AWS recommends that you use native PostgreSQL database migration tools such as pg_dump.
- For a heterogeneous migration, where migrating from a database engine other than PostgreSQL to a PostgreSQL database, AWS DMS is almost always the best migration tool to use. But for a homogeneous migration, where you are migrating from a PostgreSQL database to a PostgreSQL database, native tools can be more effective.
- Secure Sockets Layer (SSL) can be used to encrypt connections between your PostgreSQL endpoint and the replication instance.

SAP Adaptive Server Enterprise (ASE) as Source/Target for AWS DMS

As a Source:

- You can migrate data from an SAP Adaptive Server Enterprise (ASE) database (formerly Sybase) using AWS DMS.
- With an SAP ASE database as a source, you can migrate data to any of the other supported AWS DMS target databases.

As a Target:

- With an SAP ASE database as a target, you can migrate data to any of the other supported AWS DMS target databases.

S3 as Source/Target for AWS DMS

As a Source:

- You can migrate data from an Amazon Simple Storage Service bucket using AWS DMS.
- In addition to the data files, you must also provide an external table definition. An external table definition is a JSON document that describes how AWS DMS should interpret the data from Amazon S3.
- The source data files must be in comma-separated value (CSV) format
- Defining External Tables for Amazon S3 as a Source for AWS DMS
- The source S3 bucket must be in the same AWS Region as the DMS replication instance that migrates your data.
- The AWS account you use for the migration must have read access to the source bucket.

As a target:

- Amazon S3 can be used as a target for AWS DMS from any of the supported database sources.
- Both full load and change data capture (CDC) data is written to comma-separated value (.csv) format by default.
- The S3 bucket you are using as a target must be in the same region as the DMS replication instance you are using to migrate your data.
- The AWS account you use for the migration has an IAM role with write and delete access to the S3 bucket you are using as a target.
  - This role has tagging access so you can tag any S3 objects written to the target bucket.
- You can create and use custom AWS KMS keys to encrypt your Amazon S3 target objects.
- Specify different target endpoints (buckets) if your data source is from the same table.

References

aws certified
Solutions Architect
Professional
DOLFINed

References

http://www.amazon.com

http://docs.amazon.com

Amazon AWS re:Invent Videos

AWS FAQs