

Migration & Transfer

Server Migration service, Database Migration Service, Application discovery

AWS Certified Solutions Architect– Professional (SAP-C01)

Study notes - Sep'2019

Migrating Applications to the cloud

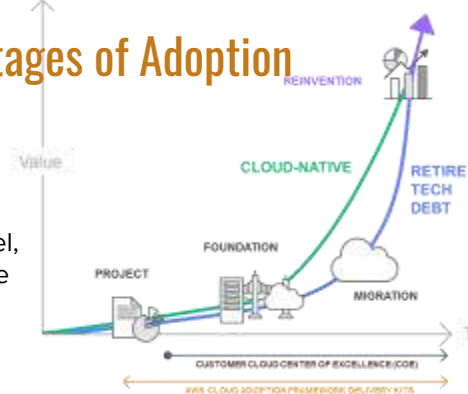
<https://aws.amazon.com/cloud-migration/>

Business Drivers - Operational Costs, Workforce Productivity, Cost Avoidance, Operational Resilience, Business Agility

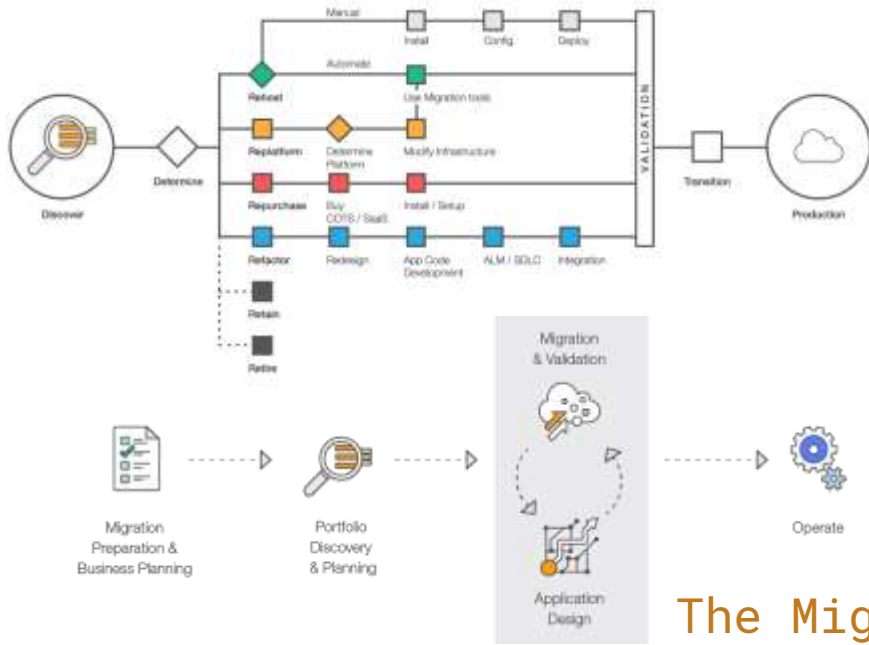
Cloud Stages of Adoption

Foundation setup:

Landing zone
CCoE, Operations model,
Security and compliance



Six Common Application Migration Strategies



- 1.Rehosting "Lift and shift"** - Use Automated tools (AWS VM Import/Export, Racemi). Applications are easier to optimize/re-architect once they're already running in the cloud as teams gain expertise.
- 2.Replatforming "lift-tinker-and-shift."** - Do targeted changes such as moving DBPaaS, Use light weight app servers, Use appropriate Cloud Native services etc.
- 3.Repurchasing** - Move to a different product typically SaaS such as Salesforce, Drupal CMS or Workday HR
- 4.Refactoring / Re-architecting** - Develop using cloud native features, Move from monolith to MSA, Most expensive and complex for development and migration
- 5.Retire** - About 10% no longer used based on functional reqms
- 6.Retain** - Do not do anything

The Migration Process

AWS Server Migration Service

<https://aws.amazon.com/server-migration-service/>

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 currently supports the migration of **on-premises virtual machines (VMs)** as an **agentless** service using a **virtual appliance**.

- SMS has an on-premises appliance, the **SMS Connector**, which talks to the service in AWS. The Connector **incrementally transfers volumes of running Hyper-V VMs to the SMS service**, and the service creates the AMI incrementally from the transferred volume
- The SMS connector is a preconfigured FreeBSD virtual machine available in OVA format for deployment in the VMware environment.
- Output is a Amazon Custom AMI which can be used to initialize new EC2 instance
- Supports VMware vCenter and MS Hyper-V environments
- OS support for Windows Server, Windows, RHEL and Most Linux distributions
- SMS is a significant enhancement of EC2 VM Import. It supports automated, live incremental server replication and AWS Console support.
- Captures VM inventory info from vCenter and replicates server volumes to AWS
- Now offers **multi-server migration** support. You can group servers into applications, replicate the entire application together, and monitor its migration status centrally from the console. Supports up to 50 concurrent VM migrations per account.
- Manages and tracks the progress of server migration through a UI

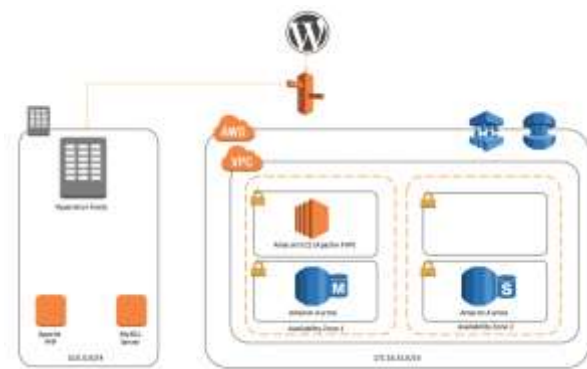
• Pre-requisites

- Before you deploy the SMS connector, remember to re-configure the firewall to permit stateful outbound connections for DHCP, DNS, HTTPS, ICMP, and NTP services
- You'll also need to create a vCenter service account and an IAM user with the appropriate policy and permissions for AWS SMS

There is no additional fee to use SMS. You pay the standard fees for the S3 buckets, EBS volumes, and data transfer used during the migration process, and for the EC2 instances that you run.

Typical steps for migrating a WP blog

1. Establish your AWS environment.
2. Replicate your database.
3. Download the SMS Connector from AWS
4. Configure AWS SMS and Hypervisor permissions.
5. Install and configure the SMS Connector appliance.
6. Import your virtual machine inventory and create a replication job.
7. Launch your Amazon EC2 instance.
8. Change your DNS records to resolve the WordPress blog to your EC2 instance.



AWS Server Migration Service

Getting Started with AWS Server Migration Service

1. Configure AWS SMS Permissions and Roles:

If your IAM user account, group, or role is assigned administrator permissions, then you already have access to AWS SMS. To call the AWS SMS API with the credentials of an IAM user that does not have administrative access to your AWS account:

- Create a custom inline policy
- Apply it to the IAM user.

2. Configure an IAM User for Server Migration Connector

- Create a new IAM user for your connector to communicate with AWS.
- Attach the managed IAM policy `ServerMigrationConnector` to the IAM user.

3. Configure a Service Role for AWS SMS

Create an IAM role that grants permissions to AWS SMS to place migrated resources into your Amazon EC2 account. Attach managed policy `'ServerMigrationServiceRole'`

4. Configure a Launch Role for AWS SMS

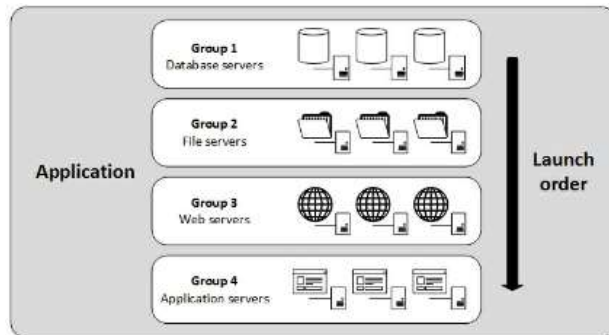
If you plan to launch applications, you need an AWS SMS launch role. You assign this role using the `PutAppLaunchConfiguration` API. When the `LaunchApp` API is called, the role is used by AWS CloudFormation. Attach managed policy `'ServerMigrationServiceLaunchRole'`

Migrating Applications with AWS

SMS AWS Server Migration Service supports the automated migration of **multi-server application stacks** from your on-premises data center to Amazon EC2.

Where server migration is accomplished by replicating a single server as an Amazon Machine Image (AMI), application migration replicates all of 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.



AWS Database Migration Service

Use DMS to migrate your data into AWS, between on-premises instances (through an AWS Cloud setup), or between combinations of cloud and on-premises setups.

- AWS DMS is a server in AWS that runs replication software.
- You create a source and target connection to tell AWS DMS where to extract from and load to.
- Then you schedule a task that runs on this server to move data.
- DMS creates the tables and associated primary keys if they don't exist on the target.
- You can precreate the target tables manually, if you prefer.
- AWS DMS doesn't perform schema or code conversion. You can use tools such as Oracle SQL Developer, MySQL Workbench, or pgAdmin III to move your schema if your source and target are the same database engine.
- If you want to change database engines, you can use the AWS Schema Conversion Tool (AWS SCT) to create some or all of the target tables, indexes, views, triggers. You then use DMS to migrate the data.
- Support for data validation, to ensure that your data was migrated accurately from the source to the target.

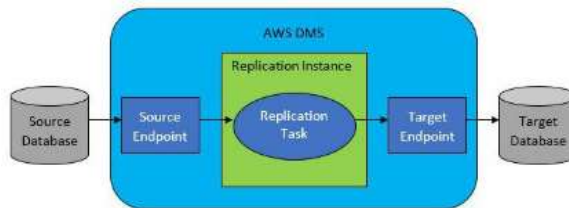
Supported sources:

Oracle, SQL Server, MySQL, MariaDB, PostgreSQL, Db2 LUW, SAP, MongoDB, Aurora, S3, RDS, Azure SQL



Supported targets:

Oracle, SQL Server, PostgreSQL, MySQL, Redshift, SAP ASE, S3, DynamoDB, RDS, AWS ElasticSearch, Kinesis, DocumentDB (w/ Mongo), Aurora (w/ Mysql & pg)



Supports **one-time migrations** AND **ongoing changes** to keep sources and targets in sync.

High-Level Steps of AWS DMS

1. Create a **replication server (managed EC2)**
2. Create **source and target endpoints** that have connection information about your data stores.
3. Create **one or more migration tasks** to migrate data between the source and target data stores. A task can consist of 3 major phases:
 - **Full load (migrate existing data)**
Typically used if you can bear downtime
 - **Full load + CDC (Migrate existing data and replicate ongoing changes)**
 - **CDC only (Replicate data changes only)**
Initial load could be done by using native tools, only changes are then applied using DMS

Monitoring AWS DMS Tasks

Task status (running/ stopped/failed), percent complete, elapsed time and table statistics (# of inserts, deletions, and updates)
Metrics by Host, Task and Table
Replication logs to CloudWatch
AWS DMS API Calls with AWS CloudTrail

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

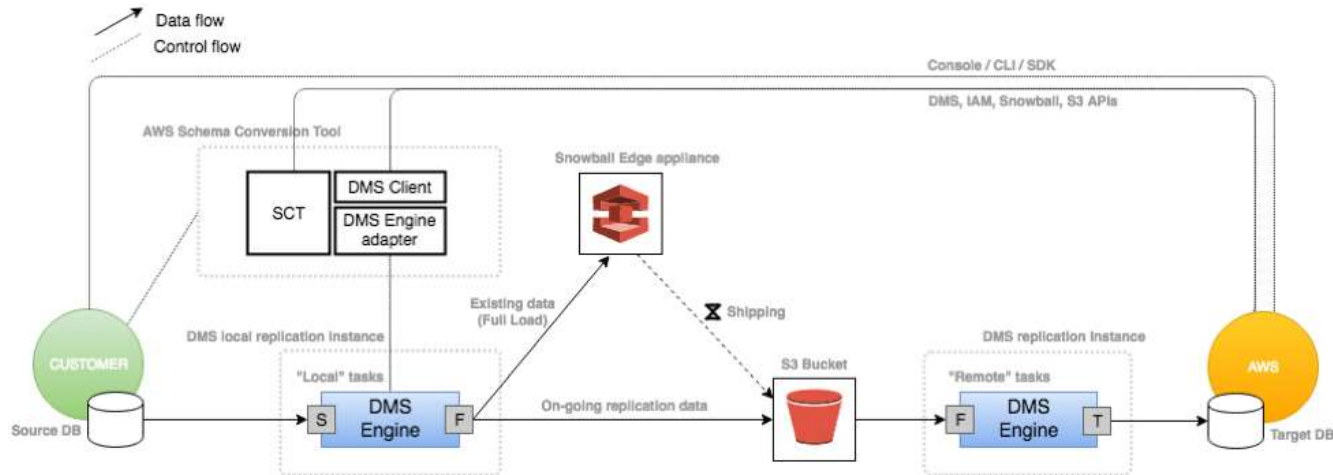
Migrating Large Data Stores Using AWS Database Migration Service and AWS Snowball Edge

https://docs.aws.amazon.com/dms/latest/userguide/CHAP_LargeDBs.html

Larger data migrations can include many terabytes of information. This process can be cumbersome due to network bandwidth limits or just the sheer amount of data. DMS can use [Snowball Edge](#) and S3 to migrate large databases more quickly than by other methods.

When you're using an Edge device, the data migration process has the following stages:

1. You use the AWS Schema Conversion Tool (AWS SCT) to extract the data locally and move it to an Edge device.
2. You ship the Edge device or devices back to AWS.
3. After AWS receives your shipment, the Edge device automatically loads its data into an Amazon S3 bucket.
4. AWS DMS takes the files and migrates the data to the target data store. If you are using change data capture (CDC), those updates are written to the Amazon S3 bucket and then applied to the target data store.



Best Practices for AWS Database Migration Service

https://docs.aws.amazon.com/dms/latest/userguide/CHAP_BestPractices.html

- Improving the Performance of an AWS DMS Migration
 - a. Load Multiple Tables in Parallel (8 default)
 - b. Working with Indexes, Triggers and Referential Integrity Constraints
 - i. For full load > disable all, For full load + CDC > Add secondary indexes before load
 - c. Disable Backups and Transaction Logging
 - d. Use Multiple Tasks (Migrate tables in multiple tasks in parallel that are not related to each other)
 - e. Optimizing Change Processing (use batch optimised, disable referential integrity)
- Choosing the Optimum Size for a Replication Instance
- Reducing the Load on Your Source Database
- Using the Task Log to Troubleshoot Migration Issues
- Converting Schema
- Migrating Large Binary Objects (LOBs)
 - a. Limited LOB (default) or Full LOB mode
 - b. Full load supported by Oracle, SQL Server, ODBC (Source)→ Oracle, SQL Server (Target)
- Ongoing Replication
- Changing the User and Schema for an Oracle Target
- Changing Table and Index Tablespaces for an Oracle Target
- Improving Performance When Migrating Large Tables
 - a. If you want to improve the performance when migrating a large table, you can break the migration into more than one task. To break the migration into multiple tasks using row filtering, use a key or a partition key. For example, if you have an integer primary key ID from 1 to 8,000,000, you can create eight tasks using row filtering to migrate 1 million records each.

AWS Application Discovery Service

<https://docs.aws.amazon.com/application-discovery/latest/userguide/appdiscovery-ug.pdf>

- Helps in collecting **usage and configuration data** about your on-premises servers.
- Its' **integrated with AWS Migration Hub**, which simplifies your migration tracking.
- After discovery, you can **view the discovered servers, group them into applications**, and **track the migration status** of applications from the Migration Hub console.
- The discovered data can be exported for **analysis in Microsoft Excel** or AWS analysis tools such as **Athena** and **QuickSight**.
- Export the **system performance** and **utilization data** for your discovered servers.
- You can input this data into your **cost model** to compute the expected cost.
- **Export the network connections** and **process data** that exist between servers. This will help you determine the network dependencies between servers and group them into applications for migration planning.

IAM - An administrator account will by default inherit all the policies required for accessing ADS. For non-administrative account, you need to add below policies:

A **Service-linked role** is a unique type of IAM role that is linked directly to ADS. Service-linked roles are predefined by ADS and include all the permissions that the service requires to call other AWS services on your behalf.

Simple discovery
on VMware?

Detailed discovery
on windows /linux?

Agentless discovery -

- Deploy the AWS **Agentless Discovery Connector** (OVA file) through your **VMware vCenter**. It identifies virtual machines (VMs) and hosts associated with vCenter.
- Data collected: **Server hostnames, IP & MAC addresses, disk allocations**. Additionally, it collects **utilization data** for each VM and **computes average and peak utilization** for metrics such as CPU, RAM, and Disk I/O.
- Export a summary of the system performance information for all the VMs associated with a given VM host and perform a **cost analysis** of running them in AWS.

Agent based discovery -

- Deploy the AWS **Application Discovery Agent** on each of your VMs and physical servers. The agent installer is available for both Windows and Linux.
- Data collected: **static configuration data**, detailed **time-series system-performance information, inbound and outbound network connections**, and **processes** that are running.
- Export this data to perform a detailed **cost analysis** and to identify **network connections between servers** for grouping servers as applications.

AWS Application Discovery Service

AWS Discovery Connector & AWS Discovery Agent

AWS Discovery Connector is a VMware appliance that can collect information only about VMware (VMs)

This mode doesn't require you to install a connector on each host. You install the Connector as a VM in your vCenter environment using an Open Virtualization Archive (OVA) file.

Because the Discovery Connector relies on VMware metadata to gather server information regardless of operating system, it minimizes the time required for initial on-premises infrastructure assessment.

After you deploy and configure the Discovery Connector, it registers with the ADS endpoint, <https://arsenal.us-west-2.amazonaws.com/>, and pings the service (w/ SSL) at every 60 minutes, for config information.

When you start the connector's data collecting process, it connects to VMware vCenter Server where it collects information about all the VMs and hosts managed by this specific vCenter.

AWS Discovery Agent is AWS software that you install on on-premises servers and VMs targeted for discovery and migration.

Agents support most Linux and Windows operating systems, and you can deploy them on physical on-premises servers, Amazon EC2 instances, and virtual machines.

The Discovery Agent runs in your local environment and requires root privileges.

When you start the Discovery Agent, it connects securely with arsenal.us-west-2.amazonaws.com and registers with ADS. Then it pings the service (w/ TLS) at 15 minute intervals for configuration information.

When you send a command that tells an agent to start data collection, it starts collecting data for the host or VM where it resides.

In addition to system specifications, Discovery Agents collect more detailed data including time series data for system performance information, inbound and outbound network connections, and processes running on the server

AWS Application Discovery Service

Migration Hub Import - Alternative to Discovery Connector & Agent

Migration Hub Import allows you to **import details of your on-premises environment directly into Migration Hub without using the Discovery Connector or Agent**. This gives you option to perform migration assessment and planning directly from your imported data. You can also group your devices as applications and track their migration status

To initiate an import request, first download the specially-formatted, **CSV import template**, **populate it** with your existing on-premises server data, and **upload it to Migration Hub** using the Migration Hub console, AWS CLI or one of the AWS SDKs.

At any given time, by using the console or import APIs, you can check the status of your import requests. After an import request is complete, you can **view the details of individual imported records**. View utilization data, tags, and application mappings directly from within the Migration Hub console.

Discovery Tools

In order to plan the migration from your data centers to AWS, you can either import data from an existing source or perform a detailed discovery using Discovery Connector or Agent.

 **Import**
Use the [Import template](#) to upload data about servers and applications (group of servers) from existing data sources. [Learn more](#)
[Import](#) [View imports](#)

 **Discovery Connector**
Deploy an agentless Discovery Connector in each VMware vCenter. The connectors collect specification and summarized performance about your VMs. [Learn more](#)
[Download connector](#)

 **Discovery Agent**
Install a Discovery Agent on each of your VMs or physical servers. The agents collect running processes and network connections besides specification and summarized performance. [Learn more](#)
[Download agent](#)

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

AWS Application Discovery Service

View Collected Data

After data collection, you can use the console to view collected data about your servers and VMs. Data appears in the console approximately 15 minutes after turning on data collection. This data can also be viewed in a csv format by bulk exporting the collected data by making API calls through the AWS CLI.



Data export from >

- **Discovery Agent** - you can export the collected data either from the console or from the AWS CLI.
- **Discovery Connector** - you can only export the collected data from the AWS CLI.

Data Exploration in Amazon Athena

A better alternative for data analysis is using Amazon Athena. Once Data Exploration in Athena is enabled from the Migration Hub console and data collection is turned on, data collected will automatically get stored in your S3 bucket.

You can then visit Athena 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. In addition, you can **write your own custom queries** using Amazon Athena, **upload additional existing data sources** such as CMDB exports, and associate the discovered servers with the actual business applications.

Visualize Amazon Athena Data

To visualize your data, a query can be ported to a visualization program such as Amazon QuickSight or other open-source visualization tools such as Cytoscape, yEd, or Gephi. Use these tools to render network diagrams, summary charts, and other graphical representations.

Sample csv export files:

- `<awsaccid>_destinationProcessConnection.csv`
- `<awsaccid>_networkInterface.csv`
- `<awsaccid>_osInfo.csv`
- `<awsaccid>_process.csv`
- `<awsaccid>_sourceProcessConnection.csv`
- `<awsaccid>_systemPerformance.csv`

AWS Snowball - Data migration service

<https://docs.aws.amazon.com/snowball/latest/ug/AWSSnowball-ug.pdf>

Snowball vs Snowball Edge vs SnowMobile

AWS Snowball is a service that accelerates transferring large amounts of data into and out of AWS using physical storage devices, bypassing the Internet.

Note: If you want to transfer less than 10 TB data, Snowball might not be your most economical choice.

Features:

- Import and export data between your on-premises locations and S3.
- Snowball has an 80 TB model available in all regions, and a 50 TB model only available in the US regions.
- Encryption is enforced protecting data at rest and in physical transit.
- You don't have to buy or maintain your own hardware devices.
- You can manage your jobs through the AWS Snowball Management Console, or programmatically with the job management API.
- You can perform local data transfers between your on-premises data center and a Snowball. These transfers can be done through the Snowball client, a standalone downloadable client, or programmatically using Amazon S3 REST API calls with the downloadable Amazon S3 Adapter for Snowball.
- The Snowball is its own shipping container, and its E Ink display changes to show your shipping label when the Snowball is ready to ship.

The **Snowball** and the **Snowball Edge** are two different devices. Both devices allow you to move huge amounts of data into and out of Amazon S3, they both have the same job management API, and they both use the same console. However, the two devices differ in hardware specifications, some features, what transfer tools are used, and price.

Use case	SB	SBE
Import data into Amazon S3	Y	Y
Export from Amazon S3	Y	Y
Durable local storage		Y
Local compute with AWS Lambda		Y
Amazon EC2 compute instances		Y
Use in a cluster of devices		Y
Use with AWS IoT Greengrass (IoT)		Y
Transfer files w/ NFS with a GUI		Y
Storage capacity	50/80 TB	100 TB

AWS Snowball - Data migration service

AWS Snowball Tool Differences



	Snowball tools	Snowball Edge Tools
Snowball client	<ul style="list-style-type: none">• Download in local workstation• Can Transfer to or from snowball	<ul style="list-style-type: none">• Download in local workstation• Must be used to unlock the Snowball Edge or the cluster of Snowball Edge devices.• Can't be used to transfer data
Amazon S3 Adapter for Snowball	<ul style="list-style-type: none">• Encrypt data in workstation before transfer	<ul style="list-style-type: none">• Is already installed on Snowball edge by default. No download needed• Can Transfer to or from snowball edge• Encrypts data on the Snowball Edge while the data is transferred to the device.
File interface with Snowball Edge		<ul style="list-style-type: none">• Is already installed on Snowball edge by default. No download needed• Can transfer data by dragging /dropping files up to 150 GB from your computer to the buckets on the Snowball Edge through an easy-to-configure NFS mount point.• Encrypts data on the Snowball Edge while the data is transferred to the device.
AWS IoT Greengrass console with Snowball Edge		You can use the AWS IoT Greengrass console to update your AWS IoT Greengrass group and the core running on the Snowball Edge.

AWS Snowball - Data migration service



IMPORT JOB:

- Each import job uses a single Snowball appliance. After you create a job in the AWS Snowball Management Console or the job management API, we ship you a Snowball.
- When it arrives in a few days, you'll connect the Snowball to your network and transfer the data that you want imported into Amazon S3 onto that Snowball using the Snowball client or the Amazon S3 Adapter for Snowball.
- When you're done transferring data, ship the Snowball back to AWS, and we'll import your data into Amazon S3.

EXPORT JOB:

- Each export job can use any number of Snowball appliances. After you create a job in the AWS Snowball Management Console or the job management API, a listing operation starts in S3.
- This listing operation splits your job into parts. Each job part can be up to about 80 TB in size, and each job part has exactly one Snowball associated with it.
- After your job parts are created, your first job part enters the Preparing Snowball status.
- Soon after that, we start exporting your data onto a Snowball.
- Typically, exporting data takes one business day.
- Once the export is done, AWS gets the Snowball ready for pickup by your region's carrier.
- When the Snowball arrives at your data center or office in a few days, connect the Snowball to your network and transfer the data that you want exported to your servers by using the Snowball client or the Amazon S3 Adapter for Snowball.
- After transferring data, ship the Snowball back to AWS.
- Once we receive a returned Snowball for your export job part, we perform a complete erasure of the Snowball. This erasure follows the National Institute of Standards and Technology (NIST) 800-88 standards. This step marks the completion of that particular job part.

Filter or search by job name

« < Viewing 1 - 1 of 1 items > »

Job name

Created date

Job type

Status

April finance logs

10/04/2015

Import

Job created

Job status: Job created

Read the following information for more details

[View job details](#)


Job created



Your job has been created, and AWS has queued your job request for processing. If there's a problem with your request, you can cancel your job now. Once AWS begins processing your job, you cannot cancel it. To cancel this job now, from **Actions** choose **Cancel job**.

We recommend that you download and familiarize yourself with the Snowball client. It is the tool you'll use to transfer data from your on-premises source storage device to the Snowball through a computer workstation. [Get client](#).

Calculation for data transfer based on available network bandwidth OR When to opt for snowball/snowmobile vs network transfer ?



Mbps	Mb/day	MB/day	GB/day	TB/day
1	86400	10800	10,54688	
10	864000	108000	105,4688	
50	4320000	540000	527,3438	
100	8640000	1080000	1054,688	1
500	43200000	5400000	5273,438	5
1024	88473600	11059200	10800	10

AWS Snowmobile - exabyte scale Data migration service

AWS Snowmobile is the **first exabyte-scale data migration service** that allows you to move very large datasets from on-premises to AWS. Each Snowmobile is a **secured data truck with up to 100PB storage capacity** that can be dispatched to your site and connected directly to your network backbone to perform high-speed data migration. You can **quickly migrate an exabyte of data with ten Snowmobiles in parallel** from a single location or multiple data centers. **Snowmobile is offered by AWS as a managed service.**

Use cases include migrating 100's of petabytes of data (such as video libraries, genomic sequences, seismic data, satellite images), and financial records to run big data analytics on AWS, or shutting down legacy data centers and moving all local data in exabytes to AWS.

Q: How should I choose between Snowmobile and Snowball? Snowball < 10PB > Snowmobile

Low bandwidth

High bandwidth

To migrate large datasets of 10PB or more in a single location, you should use Snowmobile. For datasets less than 10PB or distributed in multiple locations, you should use Snowball. In addition, you should evaluate the amount of available bandwidth in your network backbone. If you have a high speed backbone with hundreds of Gb/s of spare throughput, then you can use Snowmobile to migrate the large datasets all at once. If you have limited bandwidth on your backbone, you should consider using multiple Snowballs to migrate the data incrementally.

Snowmobile does not support data export. It is designed to let you quickly, easily, and more securely migrate exabytes of data to AWS. **When you need to export data from AWS, you can use AWS Snowball Edge** to quickly export up to 100TB per appliance and run multiple export jobs in parallel as necessary.

Whitepaper - Migrating AWS Resources to a New AWS Region

<https://d1.awsstatic.com/whitepapers/aws-migrate-resources-to-new-region.pdf>

