AWS Cloud Migration Services

# AWS Cloud Migration Services

* AWS Cloud Migration services help to address a lot of common use cases such as
  + cloud migration,
  + disaster recovery,
  + data center decommission, and
  + content distribution.
* For migrating data from On Premises to AWS, the major aspect for considerations are
  + amount of data and network speed
  + data security in transit
  + existing application knowledge for recreation

A screenshot of a computer

Description automatically generated with medium confidence

## ****NOTE: Topic mainly for Professional Exam Only****

## Application & Database Migration Services

### AWS EC2 VM Import/Export

* allows easy import of virtual machine images from existing environment to EC2 instances and export them back to on-premises environment
* allows leveraging of existing investments in the virtual machines, built to meet compliance requirements, configuration management and IT security by bringing those virtual machines into EC2 as ready-to-use instances
* Common usages include
  + **Migrate Existing Applications and Workloads to EC2, allows to preserve software and settings that configured in the existing VMs**
  + Copy Your VM Image Catalog to Amazon EC2
  + Create a Disaster Recovery Repository for your VM images

### AWS Application Discovery Service

* helps enterprise customers plan migration projects by gathering information about their on-premises data centers.
* collects and presents server specification information, performance data, and details of running processes and network connections
* provides protection for the collected data by encrypting it both in transit to AWS and at rest within the Application Discovery Service data store.

Graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

### AWS Server Migration Service (SMS)

* is an agentless service which makes it easier and faster to migrate thousands of on-premises workloads to AWS.
* helps automate, schedule, and track incremental replications of live server volumes, making it easier to coordinate large-scale server migrations.
* currently supports migration of virtual machines from VMware vSphere,  Windows Hyper-V and Azure VM to AWS
* supports migrating Windows Server 2003, 2008, 2012, and 2016, and Windows 7, 8, and 10; Red Hat Enterprise Linux (RHEL), SUSE/SLES, CentOS, Ubuntu, Oracle Linux, Fedora, and Debian Linux OS
* replicates each server volume, which is saved as a new AMI, which can be launched as an EC2 instance
* is a significant enhancement of EC2 VM Import/Export service
* is used to Re-host Timeline

  Description automatically generated with low confidence

Diagram

Description automatically generated

### AWS Database Migration Service (DMS)

* helps migrate databases to AWS quickly and securely.
* source database remains fully operational during the migration, minimizing downtime to applications that rely on the database.
* supports homogeneous migrations such as Oracle to Oracle, as well as heterogeneous migrations between different database platforms, such as Oracle or Microsoft SQL Server to Amazon Aurora.
* monitors for replication tasks, network or host failures, and automatically provisions a host replacement in case of failures that can’t be repaired
* supports both one-time data migration into RDS and EC2-based databases as well as for continuous data replication
* supports continuous replication of the data with high availability and consolidate databases into a petabyte-scale data warehouse by streaming data to Amazon Redshift and Amazon S3
* provides free AWS Schema Conversion Tool (SCT) that automates the conversion of Oracle PL/SQL and SQL Server T-SQL code to equivalent code in the Amazon Aurora / MySQL dialect of SQL or the equivalent PL/pgSQL code in PostgreSQL
* A screenshot of a computer

  Description automatically generated with medium confidence

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

## Data Transfer Services

Timeline

Description automatically generated

### VPN

* connection utilizes IPSec to establish encrypted network connectivity between on-premises network and VPC over the Internet.
* **connections can be configured in minutes** and a good solution for an immediate need, have low to modest bandwidth requirements, and can tolerate the inherent variability in Internet-based connectivity.
* still requires internet and be configured using VGW and CGW

### AWS Direct Connect

* provides a dedicated physical connection between the corporate network and AWS Direct Connect location with no data transfer over the Internet.
* **helps bypass Internet service providers** (ISPs) in the network path
* helps reduce network costs, increase bandwidth throughput, and provide a more consistent network experience than with Internet-based connection
* takes time to setup and involves third parties
* **are not redundant** and would need another direct connect connection or a VPN connection
* Security
  + provides a dedicated physical connection without internet
  + For additional security can be used with VPN

### AWS Import/Export (upgraded to Snowball)

* accelerates moving large amounts of data into and out of AWS using secure Snowball appliances
* AWS transfers the data directly onto and off of the storage devices using Amazon’s high-speed internal network, bypassing the Internet
* **Data Migration**
  + for significant data size, AWS Import/Export is faster than Internet transfer is and more cost-effective than upgrading the connectivity
  + **if loading the data over the Internet would take a week or more, AWS Import/Export should be considered**
  + data from appliances can be imported to S3, Glacier and EBS volumes and exported from S3
  + **not suitable for applications that cannot tolerate offline transfer time**
* **Security**
  + Snowball uses an industry-standard Trusted Platform Module (TPM) that has a dedicated processor designed to detect any unauthorized modifications to the hardware, firmware, or software to physically secure the AWS Snowball device.

### Snow Family

Text

Description automatically generated

Text

Description automatically generated

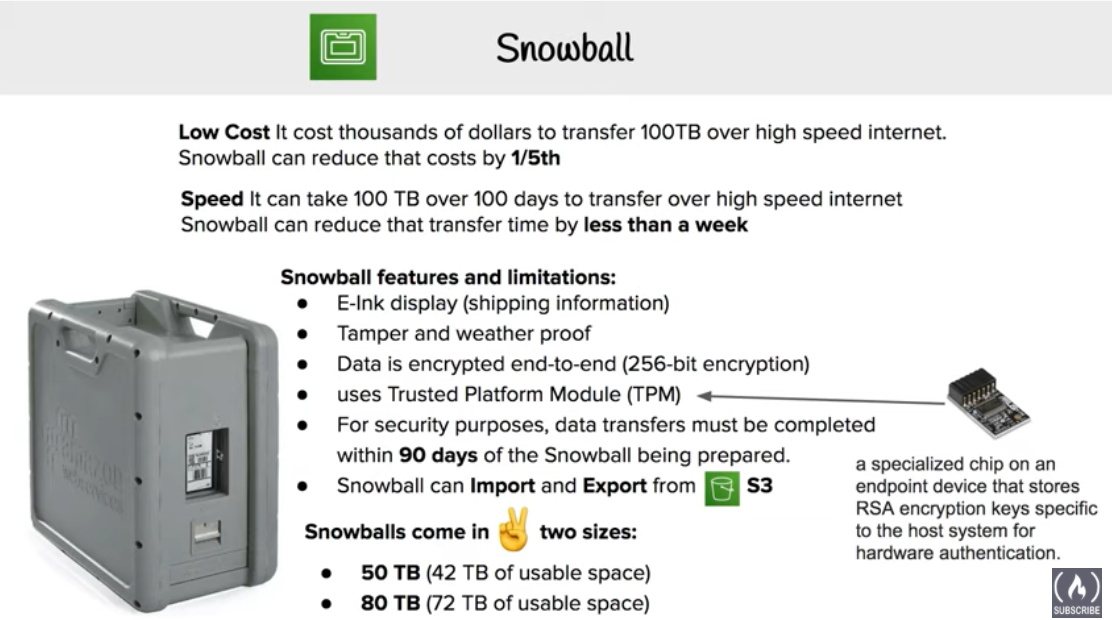
Text

Description automatically generated

Text

Description automatically generated

* AWS Snowball



* + is a petabyte-scale data transfer service built around a secure suitcase-sized device that moves data into and out of the AWS Cloud quickly and efficiently.
  + transfers the data to S3 bucket
  + transfer times are about a week from start to finish.
  + are commonly used to ship terabytes or petabytes of analytics data, healthcare and life sciences data, video libraries, image repositories, backups, and archives as part of data center shutdown, tape replacement, or application migration projects.
* AWS Snowball Edge devices



* + contain slightly larger capacity and an embedded computing platform that helps perform simple processing tasks.
  + can be rack shelved and may also be clustered together, making it simpler to collect and store data in extremely remote locations.
  + commonly used in environments with intermittent connectivity (such as manufacturing, industrial, and transportation); or in extremely remote locations (such as military or maritime operations) before shipping them back to AWS data centers.
  + delivers serverless computing applications at the network edge using AWS Greengrass and Lambda functions.
  + common use cases include capturing IoT sensor streams, on-the-fly media transcoding, image compression, metrics aggregation and industrial control signaling and alarming.
* AWS Snowmobile



* + moves up to 100PB of data (equivalent to 1,250 AWS Snowball devices) in a 45-foot long ruggedized shipping container and is ideal for multi-petabyte or Exabyte-scale digital media migrations and datacenter shutdowns.
  + arrives at the customer site and appears as a network-attached data store for more secure, high-speed data transfer. After data is transferred to Snowmobile, it is driven back to an AWS Region where the data is loaded into S3.
  + is tamper-resistant, waterproof, and temperature controlled with multiple layers of logical and physical security — including encryption, fire suppression, dedicated security personnel, GPS tracking, alarm monitoring, 24/7 video surveillance, and an escort security vehicle during transit.

### AWS Storage Gateway

* connects an on-premises software appliance with cloud-based storage to provide seamless and secure integration between an organization’s on-premises IT environment and the AWS storage infrastructure
* provides low-latency performance by maintaining frequently accessed data on-premises while securely storing all of the data encrypted in S3 or Glacier.
* for disaster recovery scenarios, Storage Gateway, together with EC2, can serve as a cloud-hosted solution that mirrors the entire production environment
* **Data Migration**
  + with gateway-cached volumes, S3 can be used to hold primary data while frequently accessed data is cached locally for faster access reducing the need to scale on premises storage infrastructure
  + with gateway-stored volumes, entire data is stored locally while asynchronously backing up data to S3
  + with gateway-VTL, offline data archiving can be performed by presenting existing backup application with an iSCSI-based VTL consisting of a virtual media changer and virtual tape drives
* **Security**
  + Encrypts all data in transit to and from AWS by using SSL/TLS.
  + All data in AWS Storage Gateway is encrypted at rest using AES-256.
  + Authentication between the gateway and iSCSI initiators can be secured by using Challenge-Handshake Authentication Protocol (CHAP).

### S3

* Data Transfer
  + Files up to 5GB can be transferred using single operation
  + Multipart uploads can be used to upload files up to 5 TB and speed up data uploads by dividing the file into multiple parts
  + **transfer rate still limited by the network speed**
* Security
  + Data in transit can be secured by using SSL/TLS or client-side encryption.
  + Encrypt data at-rest by performing server-side encryption using Amazon S3-Managed Keys (SSE-S3), AWS Key Management Service (KMS)-Managed Keys (SSE-KMS), or Customer Provided Keys (SSE-C). Or by performing client-side encryption using AWS KMS–Managed Customer Master Key (CMK) or Client-Side Master Key.

## 

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

## AWS Certification Exam Practice Questions

1. Your must architect the migration of a web application to AWS. The application consists of Linux web servers running a custom web server. You are required to save the logs generated from the application to a durable location. What options could you select to migrate the application to AWS? (Choose 2)
   1. Create an AWS Elastic Beanstalk application using the custom web server platform. Specify the web server executable and the application project and source files. Enable log file rotation to Amazon Simple Storage Service (S3). (EB does not work with Custom server executable)
   2. Create Dockerfile for the application. Create an AWS OpsWorks stack consisting of a custom layer. Create custom recipes to install Docker and to deploy your Docker container using the Dockerfile. Create custom recipes to install and configure the application to publish the logs to Amazon CloudWatch Logs (although this is one of the option, the last sentence mentions configure the application to push the logs to S3, which would need changes to application as it needs to use SDK or CLI)
   3. Create Dockerfile for the application. Create an AWS OpsWorks stack consisting of a Docker layer that uses the Dockerfile. Create custom recipes to install and configure Amazon Kinesis to publish the logs into Amazon CloudWatch. (Kinesis not needed)
   4. **Create a Dockerfile for the application. Create an AWS Elastic Beanstalk application using the Docker platform and the Dockerfile. Enable logging the Docker configuration to automatically publish the application logs. Enable log file rotation to Amazon S3.**(Use Docker configuration with awslogs and EB with Docker)
   5. **Use VM import/Export to import a virtual machine image of the server into AWS as an AMI. Create an Amazon Elastic Compute Cloud (EC2) instance from AMI, and install and configure the Amazon CloudWatch Logs agent. Create a new AMI from the instance. Create an AWS Elastic Beanstalk application using the AMI platform and the new AMI.**(Use VM Import/Export to create AMI and CloudWatch logs agent to log)
2. Your company hosts an on-premises legacy engineering application with 900GB of data shared via a central file server. The engineering data consists of thousands of individual files ranging in size from megabytes to multiple gigabytes. Engineers typically modify 5-10 percent of the files a day. Your CTO would like to migrate this application to AWS, but only if the application can be migrated over the weekend to minimize user downtime. You calculate that it will take a minimum of 48 hours to transfer 900GB of data using your company’s existing 45-Mbps Internet connection. After replicating the application’s environment in AWS, which option will allow you to move the application’s data to AWS without losing any data and within the given timeframe?
   1. Copy the data to Amazon S3 using multiple threads and multi-part upload for large files over the weekend, and work in parallel with your developers to reconfigure the replicated application environment to leverage Amazon S3 to serve the engineering files. (Still limited by 45 Mbps speed with minimum 48 hours when utilized to max)
   2. Sync the application data to Amazon S3 starting a week before the migration, on Friday morning perform a final sync, and copy the entire data set to your AWS file server after the sync completes. (Works best as the data changes can be propagated over the week and are fractional and downtime would be know)
   3. Copy the application data to a 1-TB USB drive on Friday and immediately send overnight, with Saturday delivery, the USB drive to AWS Import/Export to be imported as an EBS volume, mount the resulting EBS volume to your AWS file server on Sunday. (Downtime is not known when the data upload would be done, although Amazon says the same day the package is received)
   4. Leverage the AWS Storage Gateway to create a Gateway-Stored volume. On Friday copy the application data to the Storage Gateway volume. After the data has been copied, perform a snapshot of the volume and restore the volume as an EBS volume to be attached to your AWS file server on Sunday. (Still uses the internet)
3. You are tasked with moving a legacy application from a virtual machine running inside your datacenter to an Amazon VPC. Unfortunately this app requires access to a number of on-premises services and no one who configured the app still works for your company. Even worse there’s no documentation for it. What will allow the application running inside the VPC to reach back and access its internal dependencies without being reconfigured? (Choose 3 answers)
   1. **An AWS Direct Connect link between the VPC and the network housing the internal services**
   2. An Internet Gateway to allow a VPN connection. (Virtual and Customer gateway is needed)
   3. An Elastic IP address on the VPC instance
   4. **An IP address space that does not conflict with the one on-premises**
   5. Entries in Amazon Route 53 that allow the Instance to resolve its dependencies’ IP addresses
   6. **A VM Import of the current virtual machine**
4. An enterprise runs 103 line-of-business applications on virtual machines in an on-premises data center. Many of the applications are simple PHP, Java, or Ruby web applications, are no longer actively developed, and serve little traffic. Which approach should be used to migrate these applications to AWS with the LOWEST infrastructure costs?
   1. Deploy the applications to single-instance AWS Elastic Beanstalk environments without a load balancer.
   2. Use AWS SMS to create AMIs for each virtual machine and run them in Amazon EC2.
   3. **Convert each application to a Docker image and deploy to a small Amazon ECS cluster behind an Application Load Balancer.**
   4. Use VM Import/Export to create AMIs for each virtual machine and run them in single-instance AWS Elastic Beanstalk environments by configuring a custom image.