**Table of Content**

* Migration Objective
* Migration Checklist
* Migration Methods
* Migration Approaches
* AWS Migration: Steps
* AWS Cloud Data Migration Services
* Post Migration Stage

**Migration Objective:**

Here are some advantages that cause businesses to migrate to AWS:

* **Cost ━**AWS is famous for its pay-per-use pricing, but in addition, it provides a large variety of services that allow you to consume exactly the level of services you need, to control costs.
* **Scalability ━**AWS offers infinite scalability and also provides many ways to manage that scalability━from auto-scaling on EC2 instances to containerized deployments, serverless functions, and more.
* **Agility ━**AWS provides very comprehensive automation capabilities, integrates with DevOps workflows and allows teams to instantly spin up development, test and production environments to iterate quickly. Amazon also makes it easy to provide self-service IT and allows organizational units to set up their own infrastructure.
* High availability**and**disaster recovery ━ Amazon offers multiple data centers around the world, with a Multi-AZ feature that lets you run cloud services in multiple data centers, so even in the case of a catastrophe that affects one geographical region, your services would continue to run.

## Migration Checklist:

### 1. ****Train Staff****

Map out staff to cloud services and infrastructure and understand who needs to know what. Use the following methods to get staff up to speed on the AWS services they will be using on an ongoing basis:

* + - **Practicing in sandboxes━**give employees a small budget so they can play around on AWS. They can spin up machines, integrate them with other services, break things and get their hands dirty. This can be the fastest and most effective form of training.
    - **Cloud Center of Excellence (CCoE)━**many industry experts agree that cloud migration should be supported by an organizational unit that shares knowledge about cloud technology and practices, and promotes learning and experimentation. Identify your most cloud-savvy employees, encourage them to create a Cloud Center and become champions of your cloud effort.

### 2. ****Consider Security Needs and Access Management****

Symantec provides several important steps you can take to ensure security during and after your AWS migration:

* + - **Unified policies━**users may access cloud applications from many different locations and may need to grant third-parties access to data. Legacy applications may be restrictive in the level of access they grant users. Some web or cloud systems should be locked out and not used by any employees or third-parties. Regardless of the situation, there must be a unified policy that governs access and control for all cloud applications.
    - **Ensure flexibility━**cloud applications make dynamic use of data. Data may be encrypted at rest but may be decrypted when used and transferred between multiple cloud services. Understand data flows in your new cloud deployment and ensure you have security measures in place that do not disrupt essential data flows.
    - **Set up security brokers━**a Cloud Access Security Broker (CASB) can take log information about user interactions with endpoints and make it available for compliance, security and operations purposes. It provides visibility into activity at important cloud destinations, and lets you set policies to limit access and control shadow IT.

### 3. ****Establish Cloud Performance KPIs****

NewRelic provides a list of simple KPIs you can use to monitor your cloud. Cloud performance KPIs are critical because, in the cloud, performance is not guaranteed and can vary between deployments, data centers, Amazon services and times of the day.

**Cloud Performance KPIs examples:**

|  |  |
| --- | --- |
| **Category** | **Sample KPIs** |
| **User experience** | Page load time Response time Session duration |
| **Application performance** | Error rates Throughput Availability |
| **Infrastructure performance** | CPU Usage % Disk performance Memory usage |
| **Business metrics** | Bounce rates and dwell time Conversion rates Engagement rates |

It is important to:

* + - **Establish a baseline━**measure performance of the current on-premise deployment, to be able to compare it with the performance in the cloud. This can also help diagnose specific performance issues. Set a baseline measure for each KPI, and choose the shortest baseline period that gives you statistically significant data.
    - **Test and measure performance━**create a realistic production-like environment in the cloud and measure each application for each metric, comparing to the baseline.
    - **Diagnose problems━**if your performance lags behind your goals or the original performance of the on-premise app, identify where the problem lies: in the machine instance, the performance of the underlying cloud services, data issues, or simply insufficient cloud resource allocation.
    - **Optimize and retest━**make changes to the environment and test again to ensure performance is up to your standards. Doing this now at the pre-migration stage will save major headaches later.

### 4.****Plan Database Migration****

Amazon offers a few important considerations for migrating databases to AWS.

Take the following aspects into account:

* + - * **Data transfer━**consider how to manage the actual data transfer, will you need Extract, Transform and Load (ETL) and how will you guarantee consistency. Consider the size of the database, how long data transfer will take, and whether you need specialized services like Amazon Direct Connect. Knowing the layout of your database can help you define a migration project and speed up your data copy phase significantly.
    - **Moving to a different database━**it is common in cloud migrations to switch to a different database, especially in modernization projects. Alternatively, you may be switching to a service version of the same database, like Amazon RDS. Consider what changes need to happen at the application level and test carefully to ensure your data is there and that your critical application flows are working.
    - **Plan time for refactoring━**if you do need to refactor applications, this can be time-consuming. Leave time in your migration plan for several iterations of code changes and testing, as this may take weeks to months for complex applications.
    - **Very large tables and large objects━**if you have tables larger than 200 gigabytes and with hundreds of millions of rows, handle them with care. Amazon Database Management Service can help by loading the data in parallel. If you have large objects, they can be painful to migrate, take into account longer migration time and allocate more data to your replication server.

### 5. ****Plan Data Migration for Static Websites and Unstructured Data****

In our blog post about AWS data migration, we outlined several important considerations before moving data volumes to the cloud:

* **Amazon S3 bucket name restrictions━**every S3 bucket must have a unique name, and for static websites, the bucket name must be the same as the domain name. You must ensure that you keep bucket names available to support all the buckets used in your migration.
* **Set up CDN with Amazon CloudFront━**to provide fast global access to data assets, especially those on S3, use CloudFront, in combination with Route53 for DNS. Furthermore, an added benefit is that CloudFront enables SSL, and AWS provides free SSL certificates to be used with its services.
* **Define cache policy━**Amazon CloudFront caches your content at the network edge. Define your cache policy smartly to ensure content updates are visible to your users. Use a short Time to Live (TTL) or periodically invalidate all caches, so you can push updates to your users.
* **Data consistency on S3━**when you upload new objects they are available immediately, but update or delete operations can take time to resolve due to S3’s eventual consistency mechanism. When updated content doesn’t load for a user, it may not be a caching issue, but a data consistency issue.
* **Setting HTTP headers for web pages━**when migrating from IIS to Amazon S3, a major problem is that you can’t set HTTP response headers on all pages in bulk. You can use Amazon’s CloudFront Lambda@Edge and define a custom index page for each folder.
* **S3 security━**if you are leveraging CloudFront, it will take care of SSL connectivity. However, you also need to secure access to Amazon buckets. Use Original Access Identity or ACL and set up AWS WAF to monitor and block malicious incoming requests.
* **Select S3 storage type━**Amazon S3 offers three storage classes for static website hosting, each with a different SLA and cost. Select storage classes carefully to save costs while ensuring that you provide appropriate performance for cold/warm/hot data.

## Migration Methods:

## Lift and Shift Meaning

### Advantages of the Lift and Shift Approach

Let us examine some of the key benefits of using Lift & Shift approach for workload migration to the cloud:

* The lift and shift cloud migration approach does not demand any application-level changes as it is merely being rehosted on the cloud.
* Workloads that demand specialized hardware, say for example, graphical cards or HPC, can be directly moved to specialized VMs in cloud, which will provide similar capabilities.
* A lift and shift allows you to migrate your on-premises identity services components such as Active Directory to the cloud along with the application.
* Security and compliance management in a lift and shift cloud migration is relatively simple as you can translate the requirements to controls that should be implemented against compute, storage, and network resources.
* The lift and shift approach uses the same architecture constructs even after the migration to the cloud takes place. That means there are no significant changes required in terms of the business processes associated with the application as well as monitoring and management interfaces.

### Lift and Shift vs. Other Migration Methods

The drivers for choosing the cloud migration approach could be anything from using the least-disruptive approach, application compatibility, risk management, ROI, performance and HA requirements etc. The different components in the application architecture and how they interact with each other over different interfaces should be considered while deciding the approach.  
  
**Lift and Shift vs. PaaS migrations**: PaaS migrations involve significant effort in refactoring the application to fit into the platform offered by service provider. Architecture changes might be required to place new components or replace old components. Lift and shift data center migration on the other hand is straightforward and can be done after an analysis of the infrastructure support matrix in cloud.  
  
**Lift and Shift vs. SaaS migrations**: Migrating to a SaaS is even more challenging, as it is more about migrating from one application experience to another than it is about migrating to the cloud. All aspects, including access control, data management, security, etc., must be reconsidered and adapted to the constructs of the SaaS. A lift and shift provides the original application experience and most often can use the same authentication and security mechanisms that were used on-premises.  
  
**Migration Approaches**

When migrating from Hyper-V to AWS you need to consider the following key questions:

* Network connectivity: Are you connecting via VPN over your internet break out point or via Direct Connect?
* Network bandwidth: How much capacity do you have and what is the current utilization?
* Volume of data: How much data do you have to move and how often is this data changing?
* Downtime: How much time downtime will the business allow?
* How long do you have? Are there time constraints that will force the migration to be completed by a specified date?
* How are your Hyper-V VM drives configured? Are you using VHD or VHDX formats?

**Main Hyper-V Migration Options**

Without transforming the application and making it cloud optimized, here are the main migration options:

### ****1. AWS VM Import Service****

The AWS VM Import Service allows you to convert your VMs into the AMI format and upload into AWS. Once the VM is uploaded, the VM can then be used within EC2.

This service can be very useful if you have a standardized server build that you also want to use within your EC2 environment.

### ****2. Build Fresh VMs, Migrate Data****

This approach allows you to deploy a standardized EC2 installation and then use the data transfer method that best matches your application data volume. AWS provides a number of managed data transfer services that allow you to migrate large volumes of data that would not usually be feasible to migrate over the network.

These include:

* AWS Import/Export Disk: You load the data onto a disk and ship this to Amazon who will then transfer the data into your S3 bucket. This supports up to 16TB per disk; however, you must ensure that the disk you select is supported by Amazon.
* AWS Snowball: Snowball is designed for petabyte-scale data transportation that uses Amazon supplied appliances.
* AWS Snowmobile: This exabyte-scale data transfer service is used to move extremely large amounts of data to AWS. Each Snowmobile will support up to 100PB. It will arrive as a 45-foot long ruggedized shipping container, pulled by a semi-trailer truck so you must plan carefully how you will use this service.
* AWS Direct Connect: This service provides private connectivity between AWS and your data center, which offers a more consistent network experience than Internet-based connections. Direct Connect is available in different port speeds ranging from 50M to 10G.

Once the application is configured and working you can then carry out a final sync of the data and make the EC2 instance the live instance.

Utilizing this approach reduces the data that must be migrated. You build and install all the applications in EC2, and then migrate the application data from the Hyper-V environment to EC2. As an example, with an SQL instance you could build the VM and install SQL, then backup the on-prem SQL database and restore onto the new EC2 instance.

### ****3. System Center Virtual Machine Manager Plugin****

AWS provides a plugin into System Center Virtual Machine manager that allows you to carry out basic management of EC2 instances and migrate virtual machines from Hyper-V into EC2. Once the virtual machines are in EC2, you can continue to carry out basic management of these machines:

* Launch
* View
* Reboot
* Stop and start instances
* Terminate instances

### ****4. Third-Party AWS Marketplace Migration Tools****

Within the AWS Marketplace there are numerous tools available that allow you to migrate the VMs in a controlled and optimized manner. They typically fall into two camps:

* VM Replication: Allows you to orchestrate the replication of any supported workload running on Hyper-V VMs to EC2, then make the EC2 instance the live node with minimal downtime.
* Data Migration: Provides optimal methods to migrate data to S3. These often utilize existing on-prem infrastructure; for example, Cloud Volumes ONTAP (formerly ONTAP Cloud) for AWS allows the use of SnapMirror to replicate data from your data center to AWS.

The third-party tools are designed to optimize how data is migrated to AWS. In many cases, they provide a method of being able to orchestrate the migration of workloads in a more efficient manner than is possible if you were doing it manually.

Before selecting the right approach, you must understand the pros and cons of each method. The following table provides a high-level summary of the pros and cons.

|  |  |  |
| --- | --- | --- |
| **Approach** | **Pros** | **Cons** |
| AWS VM Import Service | * Free service from AWS * AWS supported approach * Opportunity to script as CLI tools available | * AWS Windows License Key used * VHD file format only supported, VHDX drives will have to be converted to VHD * Extended service outage due to the time required to convert and then upload into AWS * The process is not scalable across large volume of VMs * Network capacity required to move data into AWS |
| **Build Fresh VMs, Migrate Data** | * Opportunities to upgrade OS to highest level app supports. * Allows testing and validation without impacting existing service. * Data sync can be phased and will reduce the network impact. * VM deployment which can be automated, will help drive standardization. * Reduced downtime window. * AWS provides services that can help with the data migration:   o   Snowball  o   Snowmobile  o   Amazon S3 Transfer   Acceleration  o   Direct Connect | * Application by application understanding of how to carry out final sync data is required prior to cut over. * Some applications may not support a method of being able to sync data. * Network capacity required to sync data into AWS. |
| System Center Virtual Machine Manager Plugin | * Can be managed via the SCVMM console | * Linux VMs not supported * AWS have not updated plugin to support SCVMM 2016 * Requires SCVMM deployment * Extended service outage due to migration time to AWS. * Process not scalable across large volume of VMs * Network capacity required to move data into AWS. |
| **3rd Party AWS Marketplace Migration Tools** | * Ability to sync VMs and data in a network efficient manner. * Minimal downtime. * Simplified approach to scheduling the migration of a high volume of VMs * Orchestration capabilities * Provides extra capabilities that are not available natively within EC2 | * Network capacity required to move data into AWS. * Additional cost for migration tools. |

To help you understand how long it will take you to migrate your data over the network using native tools, AWS has provided an example calculation:

Number of Days = (Total Bytes)/(Megabits per second \* 125 \* 1000 \* Network Utilization \* 60 seconds \* 60 minutes \* 24 hours)

For example, if you have a T1 connection (1.544Mbps) and 1TB (1024 \* 1024 \* 1024 \* 1024 bytes), to move in or out of AWS, the theoretical minimum time it would take to load over your network connection at 80% network utilization is 82 days.

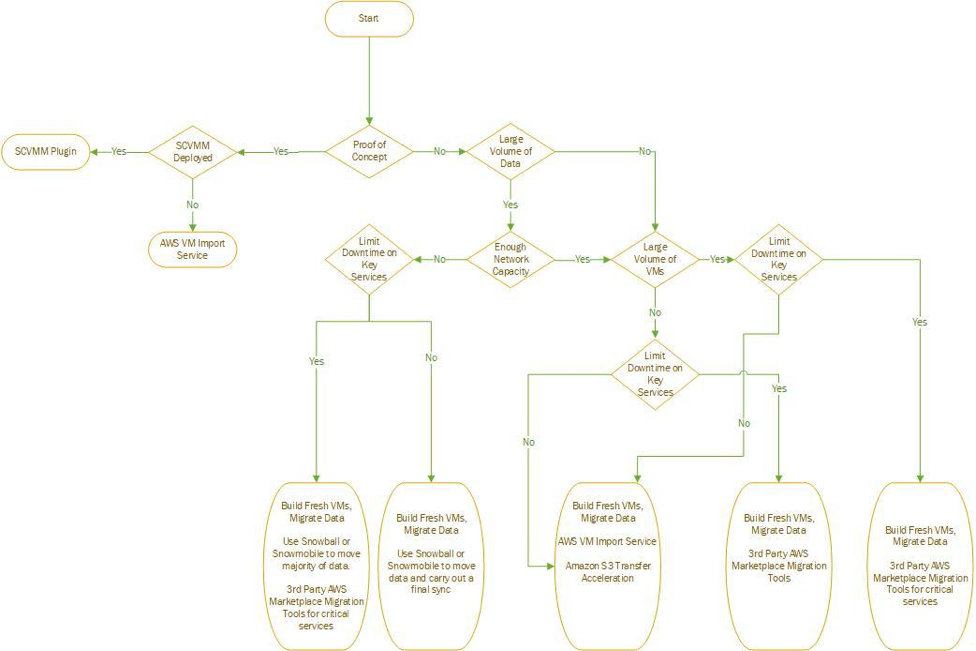
If your Hyper-V environment is using the VHDX format, you will also need to factor in the conversion time. If you are intending to use the AWS VM Import Service, the VM would have to be offline during the conversion.

## Getting Started on Your AWS Migration Strategy

As you can see, there are multiple approaches for migrating VMs from Hyper-V to AWS. Typically, a migration project will end up using some combination of the above.

To start off, carry out a proof of concept to check that the AWS design works and is ready to accept the migrated VMs. Use the AWS Import Service or the SCVMM plugin to migrate the proof of concept VMs, use this as an opportunity to also understand the network impact as you migrate the proof of concept VMs.

To help you understand the best approach for your project, take a look at this decision tree that will guide you to the best approach for your project:



It can be a time-consuming and complex task if you don’t plan your Hyper-V to AWS migration strategy carefully. So, remember:

* If you have a large volume of data and limited network bandwidth, utilize the AWS data migration services for the majority of systems and Third-Party AWS Marketplace Migration Tools for key services that must have limited downtime.
* If you have adequate network bandwidth and a large number of VMs to migrate, use Third-Party AWS Marketplace Migration Tools, as this will simplify the migration of a large number of VMs and optimize the traffic being sent over the network.
* If you have a small number of VMs, a minimal amount of data, and the network capacity to support the migration, use AWS VM Import Service.

### During Migrating to AWS

Before further moving to how of the cloud migration process, here’s a formula suggested by AWS to determine **how much data can be transferred and how fast.**

Number of Days = (Total Bytes)/(Megabits per second \* 125 \* 1000 \* Network Utilization \* 60 seconds \* 60 minutes \* 24 hours)

## AWS Migration: Steps

#### ****Following are the AWS Migration steps:****

1. Planning and Assessment
2. Migration Tools
3. AWS Cloud Storage Options
4. Migration Strategies
5. Application Migration Options

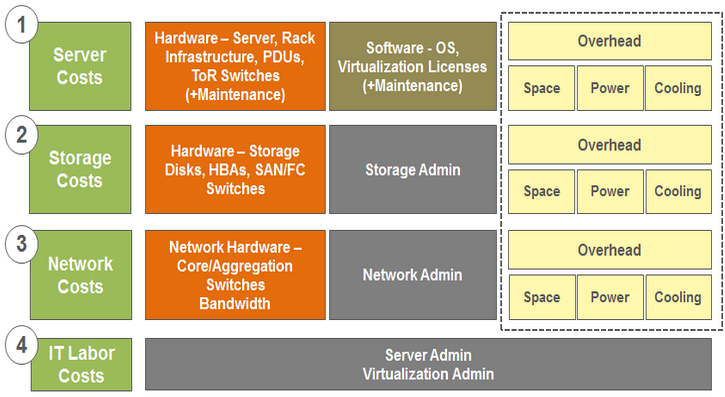
### 1. Planning and Assessment

The planning and assessment phase is divided into:

1. Financial Assessment
2. Security & Compliance Assessment
3. Technical and Functional assessment

#### 1.1 Financial Assessment

Before deciding on-prem to cloud migration, you need to estimate the cost of moving data to the AWS cloud. A careful and detailed analysis is required to weigh the financial considerations of on-premises center versus employing a cloud-based infrastructure.



**P.S. You also need to evaluate the on-premises costs which include server cost, storage cost, network cost, and IT labor costs.**

#### 1.2 Security and Compliance Assessment

If you are wondering about:

* Overall risk tolerance
* Main concerns around availability, durability, and confidentiality of your data.
* Security threats
* Options available to retrieve all data back from the cloud

#### 1.3 Technical and Functional Assessment

Assessing the need to understand which applications are more suited to the cloud strategically and architecturally. This helps you decide:

* Which application/data to move into the cloud first?
* Which data to transfer later?
* Which applications should remain in-house?

### ****2. Migration Tools****

There are physical limitations when it comes to migrating data from on-premises locations into the cloud. That’s where migration tools come to rescue. The following tools will help you move data through roads, networks, and technology partners.

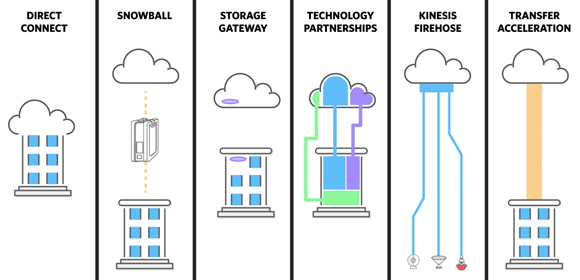
#### ****2.1 Unmanaged Cloud Data Migration Tools****

If you need easy, one-and-done methods to transfer data at small scales, go for the following tools:

* **Glacier command line interface-**On-premises data → Glacier vaults
* **S3 command line interface-**Write commands → Data moves directly into S3 buckets
* **Rsync-**Open source tool combined with 3rd party file system tools. Copy data directly → S3 buckets

### 2.2 Amazon Managed Cloud Data Migration tools

Based on optimizing or replacing the internet and friendly interfaces to S3, there are the following tools you can leverage:



#### A. Optimizing or Replacing the Internet

Ideal for moving data lakes, extensive archives and more.

|  |  |  |
| --- | --- | --- |
|  | **Ideal for** | **Data Migration Tool to Be Used** |
| 1. | Migrate petabytes of data in batches to the cloud | AWS Import/Export Snowball |
| 2. | Migrate exabytes of data in batches to the cloud | AWS Snowmobile |
| 3. | Connect directly into an AWS regional data center | AWS Direct Connect |
| 4. | Migrate recurring jobs (with incremental changes over long distances) | Amazon S3 Transfer Acceleration |

#### ****B. Friendly Interfaces to S3****

Makes it simple to use S3 with existing native applications. Helps you to integrate existing process flows like recovery, backup, etc.

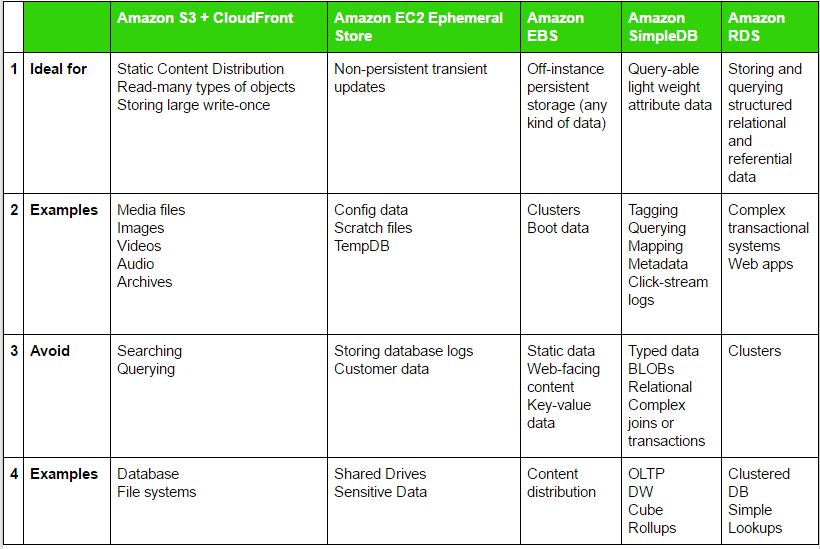
|  |  |  |
| --- | --- | --- |
|  | **Ideal for** | **Data Migration Tool to Be Used** |
| 1. | Push backups or archives to the cloud with least disruption | Technology Partnerships |
| 2. | Cache data locally in a hybrid model | Gateways (AWS or Partner) |
| 3. | Collect and ingest multiple streaming data sources | Amazon Kinesis Firehose |
| 4. | Migrate petabytes of data in batches + apply onboard storage + compute capabilities | AWS Snowball Edge |

### 3. Various Storage Options Available in the AWS Cloud

Decide which storage option is feasible for you based on:

1. Cost,
2. Durability,
3. Latency performance (response time),
4. Availability,
5. Size of the object stored (large, small),
6. Accessibility,
7. Cache-ability,
8. Consistency (eventual, strict),
9. Relational (SQL joins)
10. Update Frequency

**Which Storage Option to Use?**



## 4. 2 Major Strategies for AWS Migration

Here are two strategies that will help you move part of or an entire system to the cloud without disrupting the current business:

### 1. Forklift Migration Strategy

Self-contained applications, tightly coupled applications, or stateless applications might be better served by this approach. “Pick it all up at once and move it to the cloud” approach.

**Pros**

* Shrinking IT infrastructure footprint: Using this approach for specific application types, you have to worry less about the IT infrastructure.
* Focus on Other Important Resources: With this approach, you will be able to focus on your core and differentiating resources to be ahead of the competition.

**Cons**

* Might not be able to take immediate advantage of scalability and elasticity of the cloud

### 2. Hybrid Migration Strategy

Considering some parts of an application and moving them to the cloud while leaving other parts of the application in place. Ideal for large systems involving several applications.

#### Pros

* Low-risk approach to the migration of applications to the cloud.
* Parts can be moved and optimized one at a time.
* Reduced risk of unexpected behavior after migration.

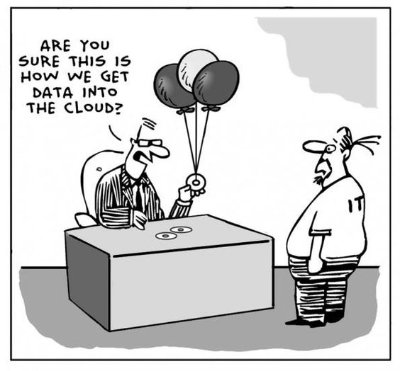
#### Cons

* Time-consuming.

#### ****Configuring and Creating AMI****

* AMI provides the information needed to launch an instance. This is provided by AWS or solution provider.
* You will need to create an AMI for each component designed to run in a separate Amazon EC2 instance.
* Create an automated or semi-automated deployment process to reduce efforts and time.
* Think of a process for configuration management to ensure your servers running in the cloud are included in your process.

## 5. Application Migration Options



Well… here are some appropriate application migration options available:

#### 1. Live Migration

The process of moving a running application from physical machines to cloud without disconnecting the application. Memory, network connectivity, and storage of the virtual machine are replicated from the physical device to cloud.

#### 2. Host Cloning

It is cloning the Operating System image and typically one-time migration.

#### 3. Data Migration

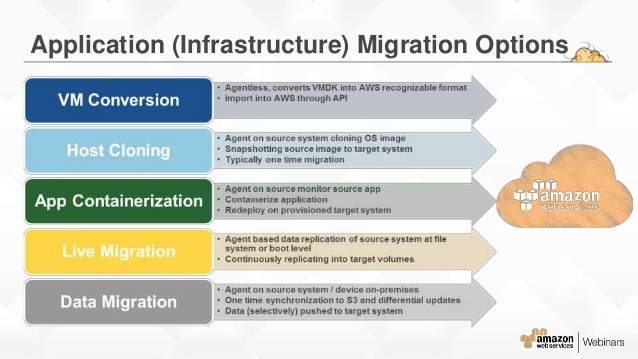
Synchronizing the data between computer storage types or file formats to the cloud. The data is selectively pushed to AWS Cloud.

#### 4. App Containerization

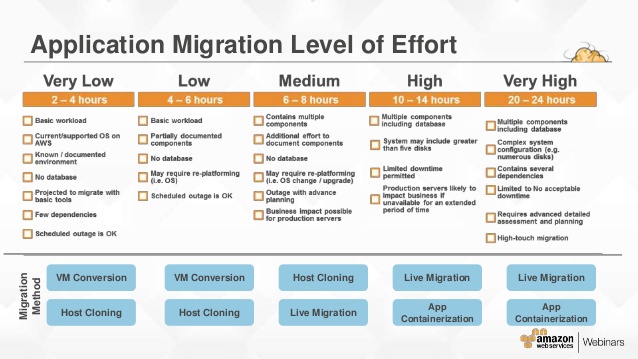
An OS-level virtualization method for deploying and running distributed applications.

#### 5. VM Conversion

Converts Virtual Machine Disk (VMDK) into AWS recognizable format. The data is transferred via API.



**The level of Efforts Required with Each Migration Method:**



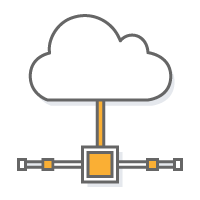
## AWS Cloud Data Migration Services

The suite of data transfer services created by AWS includes many methods that help you migrate your data more effectively. You can think about them in two categories:

* Online data transfer and hybrid cloud storage. These methods make it simple to create a network link to your VPC, transfer data to AWS, or use S3 for hybrid cloud storage with your existing on-premises applications. These services can help you both lift and shift large datasets once, as well as help you integrate existing process flows like backup and recovery or continuous data streams directly with cloud storage.
* Offline data migration to Amazon S3. One should never underestimate the bandwidth of a semi truck filled with 100 Petabytes of hard drives, or a 100 TB suitcase-sized device. These offline data migration services that use shippable, ruggedized devices are ideal for moving large archives, data lakes, or in situations where bandwidth and data volumes cannot pass over your networks within your desired time frame.

## Online data transfer and hybrid cloud storage

These migration methods enhance or replace the Internet to lift-and-shift data from your current location straight into Amazon's datacenters. No development work or APIs are supported, and transfers may impact your existing on-premises processes.

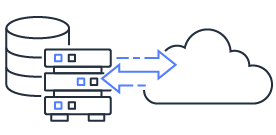


Explore our Direct Connect Partner Bundles that help extend on-premises technologies to the cloud.

### AWS Direct Connect

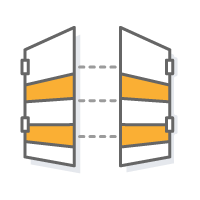
Customer select a Direct Connect dedicated physical connection to accelerate network transfers between their datacenters and ours.

AWS Direct Connect lets you establish a dedicated network connection between your network and one of the AWS Direct Connect locations. Using industry standard 802.1q VLANs, this dedicated connection can be partitioned into multiple virtual interfaces. This allows you to use the same connection to access public resources such as objects stored in Amazon S3 using public IP address space, and private resources such as Amazon EC2 instances running within an Amazon Virtual Private Cloud (VPC) using private IP space, while maintaining network separation between the public and private environments. Virtual interfaces can be reconfigured at any time to meet your changing needs.



### AWS DataSync

AWS DataSync is a data transfer service that makes it easy for you to automate moving data between on-premises storage and Amazon S3 or Amazon Elastic File System (Amazon EFS). DataSync automatically handles many of the tasks related to data transfers that can slow down migrations or burden your IT operations, including running your own instances, handling encryption, managing scripts, network optimization, and data integrity validation. You can use DataSync to transfer data at speeds up to 10 times faster than open-source tools. You can use DataSync to copy data over AWS Direct Connect or internet links to AWS for one-time data migrations, recurring data processing workflows, and automated replication for data protection and recovery.



### AWS Storage Gateway and Partner Gateways

A gateway sits on-premises and links your environment to the AWS cloud. It's an ideal solution for hybrid scenarios where some storage is needed locally for performance or compliance reasons, but some may be offloaded to S3.

Consider combining the AWS Direct Connect service with your gateway to ensure optimal performance.

#### AWS Storage Gateway

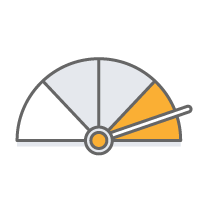
The AWS Storage Gateway service simplifies on-premises adoption of AWS storage. Your existing applications connect to a local gateway via industry-standard block and tape storage protocols to store data in Amazon S3 and Amazon Glacier.

* Data is compressed and securely transferred to AWS.
* The File Gateway presents SMB or NFS file shares for on-premises applications to store files as S3 objects and access them with traditional file interfaces.
* The Volume Gateway stores or caches block volumes locally, with point-in-time backups as EBS snapshots. These snapshots may be recovered in the cloud.
* The Tape Gateway virtual tape library (VTL) configuration seamlessly integrates with your existing backup software for cost effective tape replacement in Amazon S3 and long term archival in S3 Glacier, and S3 Glacier Deep Archive.

#### APN Partner Products

AWS has partnered with a number of industry vendors on physical gateway appliances that bridge the gap between traditional backup and cloud. Link existing on-premises data to Amazon's cloud to make the move without impacting performance and preserving existing backup catalogs.

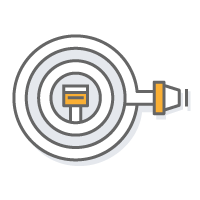
* Seamlessly integrates into existing infrastructure
* May offer deduplication, compression, encryption or WAN acceleration
* Cache recent backups locally, vault everything to the AWS cloud



### Amazon S3 Transfer Acceleration

Amazon S3 Transfer Acceleration makes public Internet transfers to Amazon S3 faster. You can maximize your available bandwidth regardless of distance or varying Internet weather, and there are no special clients or proprietary network protocols. Simply change the endpoint you use with your S3 bucket and acceleration is automatically applied.

This is ideal for recurring jobs that travel across the globe, such as media uploads, backups, and local data processing tasks that are regularly sent to a central location.

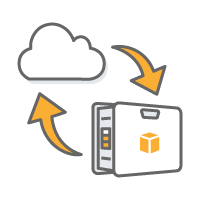


### Amazon Kinesis Data Firehose

Amazon Kinesis Data Firehose is the easiest way to load streaming data into AWS. It can capture and automatically load streaming data into Amazon S3 and Amazon Redshift, enabling near real-time analytics with existing business intelligence tools and dashboards you’re already using today. It is a fully managed service that automatically scales to match the throughput of your data and requires no ongoing administration. It can also batch, compress, and encrypt the data before loading it, minimizing the amount of storage used at the destination and increasing security. You can easily create a Firehose delivery stream from the AWS Management Console, configure it with a few clicks, and start sending data to the stream from hundreds of thousands of data sources to be loaded continuously to AWS – all in just a few minutes.

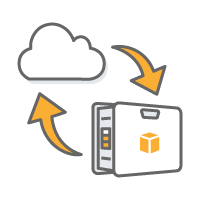
## Offline data migration to Amazon S3

Sometimes adequate bandwidth or even network are not available for the data transfers or migrations you need to make to keep a data center closure or remote site data transfer job on schedule. Here are options keep your data moving - over the roadways.



### AWS Snowball

Snowball is a petabyte-scale data transport solution that uses secure appliances to transfer large amounts of data into and out of AWS. Using Snowball addresses common challenges with large-scale data transfers including limited network bandwidth, long transfer times, and security concerns. Transferring data with Snowball is simple, fast, and secure.



### AWS Snowball Edge

Snowball Edge is a petabyte-scale data transfer device with on-board storage and compute capabilities. You can use Snowball Edge to move large amounts of data into and out of AWS, as a temporary storage tier for large local datasets, or to support local workloads in remote or offline locations.



### AWS Snowmobile

Snowmobile is an exabyte-scale data transport solution that uses a secure semi 40-foot shipping container to transfer large amounts of data into and out of AWS. Using Snowmobile addresses common challenges with large-scale data transfers including high network costs, long transfer times, and security concerns. Transferring data with Snowmobile is done through a custom engagment, is fast, secure, and can be as little as one-fifth the cost of high-speed Internet.

The Common Cloud Data Migration Challenge

The daunting realities of data transport apply to most projects. How do you gracefully move from your current location to your new cloud, with minimal disruption, cost and time? What is the smartest way to actually move your GB, TB or PB of data?

It's a basic underlying problem: how much data can move how far how fast? For a best case scenario use this formula:

*Number of Days = (Total Bytes)/(Megabits per second \* 125 \* 1000 \* Network Utilization \* 60 seconds \* 60 minutes \* 24 hours)*

For example, if you have a T1 connection (1.544Mbps) and 1TB (1024 \* 1024 \* 1024 \* 1024 bytes) to move in or out of AWS the theoretical minimum time it would take to load over your network connection at 80% network utilization is 82 days.

Relax. We’ve done this before. We've found that customers approach this in two ways: they use very basic unmanaged migration tools to move their data or they select one of Amazon's suite of services noted above.

As a general rule of thumb, for best results we suggest:

|  |  |  |
| --- | --- | --- |
| Connection | Data Scale | Method |
| Less than 10Mbps | Less than 500GB | Unmanaged |
| More than 10Mbps | More than 500GB | Managed Service |

### Post Migration Stage

#### 1. Leveraging the Cloud

After migrating your application, don’t forget to run the necessary tests, and confirm everything is in place. Invest time and resources to explore the additional benefits of the AWS cloud. You must:

* Leverage AWS Enterprise Support
* Leverage other AWS services like Auto Scaling Service, Amazon CloudFront, and Amazon Elastic MapReduce.

#### 2. Monitor and Optimize

**Understand → Monitor → Examine → Observe**

Follow this to know your load patterns and manage the cloud environment more effectively. Since AWS charges only for the infrastructure (having utility pricing structure) that has been used, you can cut cost here by optimizing your systems.

#### 3. Use Cloud Monitoring Tools

**There are various tools available that help in application-level insights and monitoring on AWS. Some of them are:**

* New Relic
* AWS CloudWatch Logs
* APPDYNAMICS