**AWS Application Migration Service (AWS MGN):**

*Situation*: Migrate the Source Servers AS-IS, means lift and shift to another AWS region

*Task*: Replicate the Source Servers into another AWS region, test your website on the new target server. If the testing is successful then perform a server cut over

*Action*: Use AWS Application Migration Service (AWS MGN) which is a highly automated lift-and-shift (re-host) solution to perform the server migration

*Results*: AWS Application Migration Service (MGN) simplifies, expedites, and reduces the cost of migrating applications to AWS. It enables companies to lift-and-shift a large number of physical, virtual, or cloud servers without compatibility issues, performance disruption, or long cutover windows. When you're ready, it automatically converts and launches your servers on AWS so you can quickly benefit from the cost savings, productivity, resilience, and agility of the Cloud. Once your applications are running on AWS, you can leverage AWS services and capabilities to quickly and easily re-platform or refactor those applications – which makes lift-and-shift a fast route to modernization.

*AWS Application Migration Service Architecture (AWS MGN):*
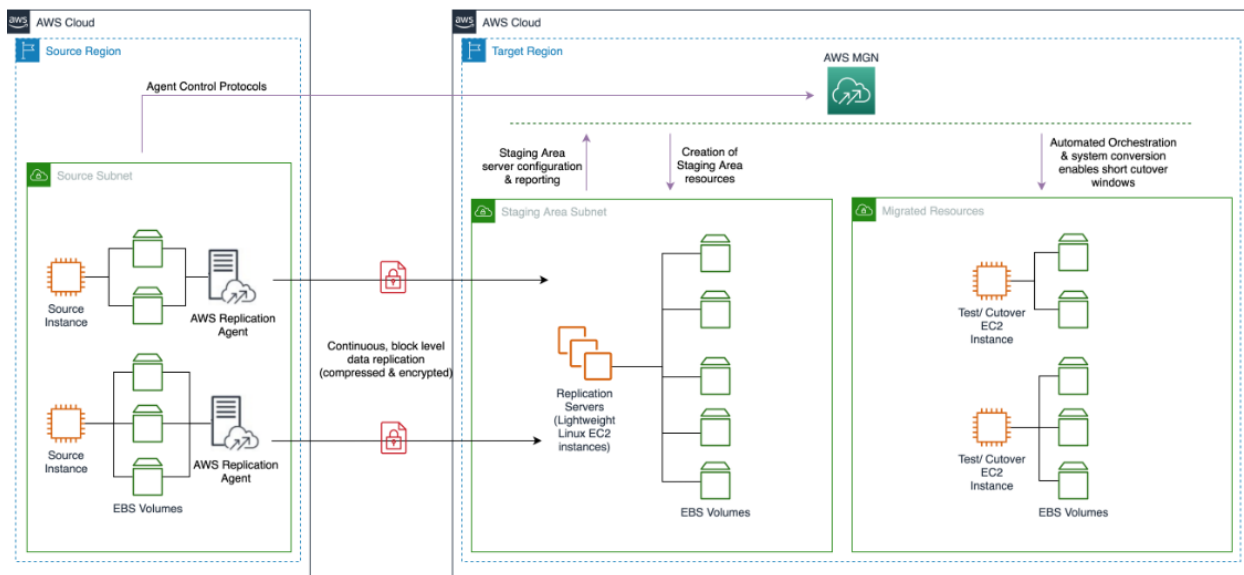


*Steps to migrate/ replicate Source Servers to another AWS region using AWS MGN*:

- Launch an EC2 instance with public IP enabled and inbound ports 22 (SSH) and 80 (HTTP) open in any AWS region of your choice to be copied over. This will be your source region and server.
- SSH into your Linux EC2 instance and switch to root (sudo) and install enable and start httpd as 1. Install httpd => yum install httpd 2. Enable httpd => sudo systemctl enable httpd 3. Start httpd => sudo systemctl start httpd
- Create a test html file on the EC2 instance as index.html
- Place your index.html file in /var/www/html/ and some the html text.
- Browse html file using instances' public IP
- ***Initialize the AWS MGN***
- In order to use the AWS MGN service, the service must first be initialized for any AWS region in which you plan to use AWS MGN and copy the server over, which is your target region
- Go to AWS MGN => Getting Started => Set up an Application Migration Service
- Leave all the options AS-IS and click on "Create Template" (this will allow MGN to create the IAM roles and the security groups required to facilitate data replication, and the launching of your migrated servers)
- ***Generate the required AWS credentials***
- Go to IAM => User => Add User

- Provide a user name (based on your choice)
- Select the Access Type as Programmatic
- Choose the attach existing policies directly and attach – *AWSApplicationMigrationAgentPolicy*
- After the user is created store/ copy the username, access key id, and the secret access key
- ***Install the replication agent on the Source Servers***
- SSH into your ***Source Servers***, and download the agent installer on your Source Servers using the command - sudo *wget -O ./aws-replication-installer-init.py [https://aws-application-migration-service-us-west-2.s3.amazonaws.com/latest/linux/aws-replication-installer-init.py](https://aws-application-migration-service-us-west-2.s3.amazonaws.com/latest/linux/aws-replication-installer-init.py)* (here us-west-2 is the target server)

[ec2-user@ip-▮▮▮▮▮ html]$ sudo wget -O ./aws-replication-installer-init.py https://aws-application-migration-service-us-west-2.s3.amazonaws.com/latest/linux/aws-replication-installer-init.py

--2022-04-02 01:22:50--  https://aws-application-migration-service-us-west-2.s3.amazonaws.com/latest/linux/aws-replication-installer-init.py

Resolving aws-application-migration-service-us-west-2.s3.amazonaws.com (aws-application-migration-service-us-west-2.s3.amazonaws.com)... ▮▮▮▮▮▮▮▮

Connecting to aws-application-migration-service-us-west-2.s3.amazonaws.com (aws-application-migration-service-us-west-2.s3.amazonaws.com)|▮▮▮▮▮▮▮▮▮... connected.

HTTP request sent, awaiting response... 200 OK

Length: 13740 (13K) [binary/octet-stream]

Saving to: './aws-replication-installer-init.py'

100%[=====================================>] 13,740      --.-K/s   in 0s

2022-04-02 01:22:50 (112 MB/s) - './aws-replication-installer-init.py' saved [13740/13740]

- Install the agent after the agent installer has successfully downloaded using the command - *sudo python3 aws-replication-installer-init.py*

[ec2-user@ip-▮▮▮▮▮ html]$ sudo python3 aws-replication-installer-init.py

The installation of the AWS Replication Agent has started.

AWS Region Name: us-west-2

AWS Access Key ID: ▮▮▮▮▮▮▮▮▮▮▮▮

AWS Secret Access Key:

Identifying volumes for replication.

Choose the disks you want to replicate. Your disks are: /dev/sda,/dev/xvda

To replicate some of the disks, type the path of the disks, separated with a comma (for example, /dev/sda,/dev/sdb). To replicate all disks, press Enter:

Identified volume for replication: /dev/xvda of size 8 GiB

All volumes for replication were successfully identified.

Downloading the AWS Replication Agent onto the source server... Finished.

Installing the AWS Replication Agent onto the source server... Finished.

Syncing the source server with the Application Migration Service Console... Finished.

The following is the source server ID: s-51e9ba7173f604b3d.

You now have 1 active source server out of a total quota of 20.

Learn more about increasing source servers limit at https://docs.aws.amazon.com/mgn/latest/ug/MGN-service-limits.html

The AWS Replication Agent was successfully installed.

- Provide the target region (for an example, us-west-2) where AWS MGN is initialized and where the server has to be replicated
- Provide the user's Access Key ID and Secret Access Key
- Choose to replicated all disks ('Enter' to replicated all disks)
- Agent should be downloaded, installed, and the server should be synced with AWS MGN of the selected region, and visible via the AWS MGN console
- ***Provision AWS MGN Replication Server***
- Click on the Source Servers and monitor the life cycle progress, replication initiation steps
- Go to EC2 and look for AWS MGN service replication server. As per your template a server will be provisioned
- ***Edit launch settings and template (If you would like to change the launch settings and the template – it's an optional steps) Note: While creating the template create new SG using inbound port 80 & 22, and also make sure the instance has the public IP***

| Inbound rules | Outbound rules | Tags |
|---|---|---|

> ⓘ You can now check network connectivity with Reachability Analyzer       **Run Reachability Analyzer**     ✕

**Inbound rules (2)**          ⟳   Manage tags   **Edit inbound rules**

🔍 Filter security group rules                                          ‹ 1 ›   ⚙

| ▽ | Security group rule... ▽ | IP version ▽ | Type ▽ | Protocol ▽ | Port ra |
|---|---|---|---|---|---|
| | sgr-09d461707e8d12... | IPv4 | HTTP | TCP | 80 |
| | sgr-0ef87b208c7cb7348 | IPv4 | SSH | TCP | 22 |

- Click on the Source Servers => Launch settings => EC2 Launch template
- Enter the launch template name
- Go to EC2 => Launce template => Go to the corresponding launch template
- In case if you would like to make changes, then
- Go to Actions => Modify Template => Make required changes and click on => Create Template Version
- Once the version is created, select the version from the dropdown and go to Actions => Set Default Version
- ***Launch the Test Instance***
- Select the Source Servers => Test and Cutover => Launch Test Instance
- A test instance should be launched as per your Launch Template
- Click on the Source Servers => Lifecycle => Click on the job id and monitor the progress
- Go to EC2 => A new AWS MGN service conversion server should be launched
- Once the conversion is complete a test instance should be available
- Test the application by browsing index.html using the test instances' public IP
- After the testing is complete => Select the Source Servers => Test and Cutover => Mark Ready for Cutover

- ***Launch the Cutover Instance & Archive***
- Select the Source Servers => Test and Cutover => Launch Cutover Instance
- A cutover instance should be launched as per your launch template
- Click on the Source Servers => Lifecycle => Click on the job id and monitor the progress
- Go to EC2 => A new AWS MGN service conversion server should be launched
- Once the conversion is complete a cutover instance should be available
- Test the application by browsing index.html using the cutover instances' public IP
- After the testing is complete => Select the Source Servers => Test and Cutover => Finalize Cutover
- Once the cutover is complete => go to the Source Serverss => Action => Mark archived ( to archive the Source Servers. Archiving servers prevents them being shown in the servers list page.)

| Migration dashboard | Server info | Tags | Disks settings | Replication settings | Launch settings |
|---|---|---|---|---|---|

### Lifecycle  Info

Not ready — Ready for testing — Test in progress — Ready for cutover — Cutover in progress — **Cutover complete**

**Launch status**
Launched
First boot: Succeeded
View in EC2 console

**Last test**
Job ID: mgnjob-59327d83640c683fe
Started: April 01, 2022, 19:40 (UTC-7:00)

**Cutover**
Job ID: mgnjob-5adcdbf9308737e37
Started: April 01, 2022, 19:56 (UTC-7:00)
Finalized: April 01, 2022, 20:15 (UTC-7:00)

***AWS Links***:

https://docs.aws.amazon.com/mgn/latest/ug/linux-agent.html

https://docs.aws.amazon.com/mgn/latest/ug/what-is-application-migration-service.html