MIGRATION PLAYBOOK · MULTI-CLOUD

How to migrate between clouds or bring workloads back on premises

A practical approach to moving servers, databases and storage from AWS to Azure, from Azure to AWS, or back to your own data centre or colocation.

Updated 4 Nov 2026 6 phases 24 detailed steps 8 official sources

START HERE

How do you migrate between clouds or bring workloads back on premises?

SHORT ANSWER

This moves servers, databases and storage from AWS to Azure, from Azure to AWS, or back to your own hardware (repatriation). Plain servers move with the target cloud's own replication tool, while managed services such as databases and storage each need their own method.

- **Recommended route:** Rehost servers with the target cloud's native tool, and replatform databases and storage
- **Order:** Measure what you run and what it will cost to move it out, set up identity and networking in the target cloud, start the long data copies early, then cut over one application at a time with its database.
- **6 phases:** List what you run, Plan the data move, Replicate the servers, Move databases and storage, Cut over and Clean up the source.
- **You're done when:** Every source resource has a target, a replacement managed service, or a recorded retire or retain decision.

The time and cost are driven by how much data leaves the source cloud, because data transfer out (egress) is charged, so measure it before you set dates.

RECOMMENDED ROUTE

Rehost servers with the target cloud's native tool, and replatform databases and storage

Azure Migrate (AWS to Azure) and AWS Transform MGN (Azure to AWS) replicate servers continuously and let you test before cutover.

▼ **Other options and when to use them**

- **Rebuild on the target cloud**
  : Workloads already defined in code, containers or serverless functions, and images that cannot be replicated, such as Amazon Linux.
- **Repatriate to on-premises or colocation**
  : Steady, predictable workloads you have decided to run on your own hardware. Size it from measured cloud use.

TOOLS FOR THIS PLAN

Are you ready?
22-question readiness check

Plan your dates
Dated countdown you can add to your calendar

Compare options
Cloud platforms compared

Check your DNS
MX, SPF, DKIM and DMARC lookup

New to the jargon?
Plain-English migration glossary

ORDER TO MIGRATE IN, AND WHY

Measure what you run and what it will cost to move it out, set up identity and networking in the target cloud, start the long data copies early, then cut over one application at a time with its database. Each step depends on the one before: servers cannot sign in or reach their data until roles, secrets and network links exist in the target.

1. **Inventory and egress estimate (data transfer out of the source cloud)**
   Every byte that leaves the source cloud is charged, so the size and daily change of each data store decide the method and the budget.
2. **Target landing zone: network, link between the clouds, roles, managed identities and secrets**
   Replication needs network paths and firewall rules, and migrated servers need identities and secrets to start their applications.
3. **Bulk copies of large databases and buckets, started early**
   The largest stores take longest to seed; later passes then copy only changes, which keeps cutover short and egress lower.

4 **Pilot: one simple application with its server and data**

Proves the replication tool, the data tool and the cutover steps end to end before the big applications.

5 **Application waves: each application's servers, database and storage cut over in one window**

An application running in one cloud while its database sits in the other pays egress on every query and adds delay.

6 **Clean up the source after each rollback window**

You pay for both clouds while workloads overlap, but deleted resources cannot be recovered, so cleanup waits for sign-off.

SETTINGS CHECKLIST: DECIDE THESE BEFORE YOU COPY

Built-in and third-party tools treat these differently. Tick each one off once you have decided.

Download this checklist (Excel)

☐ **Data transfer out (egress)**

Which data is copied at all, and how many full copies and top-up passes you can afford.

▼ How each tool handles it

Built-in method	Every method: the source cloud charges for data that leaves it, whichever tool copies it. Each full copy, retry, replication change and top-up pass counts. AzCopy copies S3 objects server to server, so data does not pass through your computer, but it still leaves AWS.
AWS DataSync	Lists transferring data in or out of Azure Blob Storage among the costs of a transfer.

Tip: Delete old snapshots, logs and backup histories with owner approval before the first copy, and check the source bill daily during waves.

Microsoft Learn: Copy data from Amazon S3 to Azure Storage by using AzCopy

☐ **Server replication method**

Which tool replicates each server, and where its appliance or agent runs.

▼ How each tool handles it

Built-in method	AWS to Azure: Azure Migrate agent-based replication treats EC2 instances as physical servers, with a replication

appliance in the source VPC and the Mobility service on each server. Azure to AWS: AWS Transform MGN installs the AWS Replication Agent in each VM.

Tip: Open the replication ports (443 and 9443 for Azure Migrate, 443 and 1500 for MGN) before installing anything.

Microsoft Learn: Migrate machines as physical servers to Azure

- [ ] **Boot type: BIOS or UEFI**

Which target VM generation or instance type suits each server.

▼ **How each tool handles it**

Built-in method

AWS Transform MGN: launches with the same boot mode as the source; UEFI works only on Nitro instance types. Azure Migrate (EC2 instances as physical servers): check UEFI and Secure Boot rules in Microsoft's physical server support matrix, and test a UEFI server in your pilot. For VMware sources, Microsoft says agent-based migration does not support Secure Boot.

Tip: Turn Secure Boot back on after migration where the target supports it, such as an Azure Trusted Launch VM.

AWS docs: Boot mode (AWS Transform MGN)

- [ ] **Drivers and cloud agents**

Which source cloud agents to remove and which target agents to install.

▼ **How each tool handles it**

Built-in method

AWS Transform MGN: before launch, a conversion server changes the bootloader, injects drivers and installs AWS cloud tools. Azure Migrate agent-based: needs the Mobility service in each server; which guest changes it makes is not stated in Microsoft's support matrix: test it in your pilot. Amazon Linux cannot move to Azure as-is.

Tip: After cutover, remove the source cloud's agents and install the target's (for AWS, the Systems Manager and CloudWatch agents).

AWS docs: AWS Transform MGN FAQ

- [ ] **IP addresses and DNS**

Which servers must keep their private IP, and how clients find the new addresses.

▼ **How each tool handles it**

Built-in method	AWS Transform MGN: with Copy private IP on, MGN checks the instance uses the same private IP as the source (IPv4 only). Azure Migrate: servers take new private IPs from the Azure subnet you choose.

Tip: Lower DNS TTLs (how long other systems cache a record) well before cutover, and search configs for hard-coded IPs.

AWS docs: Configuring launch settings (AWS Transform MGN)

- [ ] **Object metadata and bucket names**

Whether object metadata and tags must survive, and how renamed containers are handled.

▼ **How each tool handles it**

Built-in method	AzCopy (Amazon S3 to Azure Blob): by default leaves out metadata whose key name is invalid in Azure and logs a warning; FailIfInvalid skips those objects instead. Bucket names with full stops or repeated hyphens are renamed (my----bucket becomes my-4-bucket).
AWS DataSync	Azure Blob to Amazon S3: copies object tags by default. Objects with tags or metadata over 2 KB, or over 5 TB in size, are not transferred, and copying tags fails on accounts with a hierarchical namespace.
rclone	With --metadata, maps metadata to Azure Blob headers, user metadata and tags. Stores modification time as metadata (mtime), because it cannot set Azure's Last-Modified.

Tip: Test one bucket with --s2s-handle-invalid-metadata=FailIfInvalid and fix names before the real copy.

Microsoft Learn: Copy data from Amazon S3 to Azure Storage by using AzCopy

- [ ] **Storage tiers and archived objects**

Which archived objects must be restored before they can be copied.

▼ **How each tool handles it**

Built-in method	AzCopy: handling of Amazon S3 Glacier storage classes is not stated on Microsoft's S3 page: test it in your pilot.
AWS DataSync	Reads hot and cool tier blobs. Archive tier blobs must be rehydrated (moved back to an online tier) first, and the cold tier is not supported.

rclone	Cannot read blobs in the archive tier until they are rehydrated.

Tip: List objects by storage class or tier during inventory and start rehydration early, because it can take hours.

Microsoft Learn: Copy data from Amazon S3 to Azure Storage by using AzCopy

- [ ] **Top-up copies (delta passes)**

How often to run a top-up copy (only what changed since the last copy) before cutover.

▼ **How each tool handles it**

Built-in method	AzCopy from Amazon S3: incremental copies are not described on Microsoft's S3 page: test your chosen option in your pilot.
AWS DataSync	Can optionally delete destination objects that are not in the source. How it detects changed objects: not stated on its Azure Blob location page: test it in your pilot.
rclone	Copies in either direction between Azure Blob, Amazon S3 and S3-compatible storage; later passes copy only changes.

Tip: Seed the bulk weeks ahead and run the final delta pass inside the cutover window after writes stop.

Microsoft Learn: Copy data from Amazon S3 to Azure Storage by using AzCopy

- [ ] **Managed databases**

Which method moves each database, and how long its outage can be.

▼ **How each tool handles it**

Built-in method	Each target cloud has its own database tools; plain server replication does not move a managed database such as Amazon RDS or Azure SQL Database.
AWS Database Migration Service	Full load plus ongoing changes into AWS. Creates tables, primary keys and sometimes unique indexes, but not secondary indexes, other constraints, defaults or user accounts.
Azure Database for PostgreSQL migration service	Migrates Amazon RDS and Aurora PostgreSQL offline or online. Online needs primary keys on all tables.

Tip: Script indexes, constraints and users from the source, and apply them at the stage your tool documents.

AWS docs: Sources for AWS DMS

- [ ] **Large objects in databases**

How large text and binary columns (LOBs) are copied with AWS DMS.

▼ How each tool handles it

Built-in method	Native backup and restore copies LOB columns with the rest of the database.
AWS Database Migration Service	Limited LOB mode truncates any LOB larger than the size you set. Full LOB mode copies them whole, more slowly. Inline LOB mode works only during the full load.

Tip: Measure the largest LOB in each table before choosing a DMS LOB mode.

- [ ] **Roles, identities, secrets and certificates**

Who recreates each role, managed identity, secret and certificate in the target cloud.

▼ How each tool handles it

Built-in method	Azure Migrate and AWS Transform MGN copy server disks only. IAM roles, managed identities, secret stores and certificates are cloud resources outside the disk, so they are rebuilt by hand or from code.

Tip: Test each application's sign-in and secret access on a test launch or test migration before cutover.

- [ ] **Finishing and rollback**

How long the rollback window is before replication is stopped for good.

▼ How each tool handles it

Built-in method	AWS Transform MGN: Finalize cutover removes replication and uninstalls the agent. Azure Migrate: Complete migration stops replication and cannot be reversed. Until then, you can stop the target and restart the source.

Tip: Keep source resources stopped but intact until the owner signs off, then delete them.

AWS CLI reference: finalize-cutover

THE PLAN, PHASE BY PHASE

1 List what you run

Before you set any dates

Export every resource from each source account, Region or subscription with the AWS or Azure CLI.

Group resources by application and give each one an owner.

Classify each workload as rehost, replatform, rebuild, repurchase, retire, retain or relocate in

the 7R matrix.

Record the size and daily change of every database, file share and bucket.

List every role, managed identity, secret and certificate each application uses.

Watch out: Queues, serverless functions, managed databases, load balancers and identity do not move with the servers.

WAYS AROUND IT

- Give each managed service its own line with a target equivalent and a test, and budget refactoring time.

TOOLS THAT CAN DO IT

AWS Database Migration Service
Copies Azure SQL Database, Azure Database for PostgreSQL and MySQL, and Google Cloud SQL databases into AWS.

Azure Database for PostgreSQL migration service

Migrates Amazon RDS and Aurora PostgreSQL, and Google Cloud SQL for PostgreSQL, offline or online. Online needs primary keys on all tables.

Watch out: Amazon Linux cannot be migrated to Azure as-is, because it is only supported on AWS.

WAYS AROUND IT

- Mark it rebuild, build a supported Linux VM in Azure and move the application and data.

Show the detailed steps (3) →

2 Plan the data move

Get owner approval, then delete or leave behind old snapshots, logs and backup histories.

Choose a transfer path for each data store: internet, site-to-site VPN, private

link or offline device.

Apply the source provider's current transfer rates to the data per wave, including replication changes.

Ask your provider account team in writing about any exit credits or waivers.

Plan to copy the bulk early and sync

only changes during cutover.

> **Watch out:** Whichever tool you use, the source cloud charges for data leaving it: every full copy, retry, replication change and top-up sync counts.
>
> **WAYS AROUND IT**
>
> - Clean up before copying, seed once, sync only changes, and watch billing daily.

> **Watch out:** You pay for both clouds while workloads overlap, including disks and snapshots of stopped servers.
>
> **WAYS AROUND IT**
>
> - Move in waves with a fixed rollback window, then delete each wave's source resources promptly.

Show the detailed steps (4) →

3 Replicate the servers

AWS to Azure: build the Azure Migrate replication appliance on a Windows Server 2022 instance in the source VPC.

Allow TCP 443 and 9443 from the sources to that appliance, and 443 outbound to Azure.

Turn off AWS-specific cloud-init settings (Linux start-up configuration) before replicating.

Azure to AWS: install the AWS Replication Agent in each VM, allowing

outbound TCP 443 and 1500.

Run a test migration or test launch, and check identities and secrets work in the target.

> **Watch out:** Replication fails quietly when a port is missing in a security group, NACL or NSG (cloud firewall rules).
>
> **WAYS AROUND IT**
>
> - Add the rules before installing agents, scoped to the appliance or staging subnet only.

Show the detailed steps (4) →

4 Move databases and storage

Choose a method for each database (Azure SQL Managed

Instance Log Replay Service, which restores SQL Server backup files, AWS DMS, or backup and restore) and test it.

Copy Amazon S3 buckets into Azure Blob storage with AzCopy, then check object counts.

Recreate roles, managed identities, secrets and certificates in

the target cloud.

Lower the TTL (caching time) on DNS records you will change, well before cutover.

Watch out: AzCopy by default leaves out S3 metadata whose key name is invalid in Azure, and renames buckets with full stops or repeated hyphens.

WAYS AROUND IT

- Test a bucket with --s2s-handle-invalid-metadata=FailIfInvalid, and fix names before the real copy.

Watch out: AzCopy cannot write to Amazon S3, so it does not help with Azure to AWS or repatriation copies.

WAYS AROUND IT

- Test your chosen tool on one container first, checking counts, metadata and tags.

TOOLS THAT CAN DO IT

AWS DataSync
Copies Azure Blob Storage into Amazon S3 with object tags. Rehydrate archive tier first; the cold tier and objects with over 2 KB of metadata are not supported.

rclone
Open-source tool that copies between Azure Blob, Amazon S3 and S3-compatible storage in either direction, with later passes copying only changes.

Check your DNS records before and after the change

Show the detailed steps (3) →

5 Cut over

In each wave's change window

- Freeze changes and stop writes to the source application.
- Run the final syncs, including a final delta copy of object storage.
- Switch DNS and endpoints to the target, then run acceptance tests and get owner sign-off.

Remove the source cloud's agents and install the target cloud's ones (for AWS, the SSM and CloudWatch agents).

Stop the source resources, but do not delete them.

> **Watch out:** Finalising in MGN or completing migration in Azure Migrate stops replication, so you cannot simply resume.
>
> **WAYS AROUND IT**
>
> - Finalise or complete only after the rollback window.
> - To roll back before then, stop writes in the target, restart the source and point DNS back.

> **Watch out:** Once users write to the target, rolling back means copying those changes back.
>
> **WAYS AROUND IT**

- Export the changes and apply them to the source, or get the data owner to accept the loss in writing.

Turn this phase into dated tasks with the cutover planner

Show the detailed steps (3) →

6 Clean up the source

After the rollback window

Delete source compute, disks, snapshots, backups and buckets you no longer need.

Watch the source bill until it shows no new charges for the migrated workloads.

Close the source accounts or subscriptions

once billing is clear.

> **Watch out:** Deleted source resources, snapshots and buckets cannot be recovered.
>
> **WAYS AROUND IT**
>
> - Delete only after each wave's rollback window has closed and the owner has signed off.

Show the detailed steps (3) →

You are finished when

- Every source resource has a target, a replacement managed service, or a recorded retire or retain decision.
- Migrated servers pass status checks, and object counts, sizes and database checksums match the source.
- DNS returns the target addresses and every application owner has signed off.
- The source bill shows no new charges for the migrated workloads.

Tools named in this plan are independent third-party products, not endorsed by the platform vendor and not paid placements. Check the tool's own documentation and run a pilot first.

CLICK BY CLICK

What does each step involve?

Open a part when you reach it in the plan. Each step says where to go, what to do and what you should see.

About this migration — What it covers, limits at a glance

Moving between clouds is a migration like any other, with two extra problems: managed services that have no direct equivalent, and data transfer charges for everything that leaves the source cloud. The servers are usually the easy part.

This guide covers the approach, the native tools from each provider and how to plan the data movement. It does not include prices. Check each provider's current data transfer pricing and ask your account team about any exit arrangements.

AWS TO AZURE SERVERS
Azure Migrate, treating EC2 instances as physical servers

AZURE TO AWS SERVERS
AWS Transform MGN with the AWS Replication Agent in each VM

DATABASES
Azure Database Migration Service, AWS DMS, or native backup and restore

REPATRIATION
Rebuild, restore from backup, or agent-based replication to your hypervisor

BIGGEST HIDDEN FACTOR
Data transfer out of the source cloud; check current pricing

1 What you need before step one

Need	Details	Where to get it
Read access to the source cloud	To list every resource, bucket and database in each account or subscription.	AWS Management Console or Azure portal
Admin access to the target cloud	To create networks, the migration project or MGN service, and IAM roles or managed identities.	AWS Management Console or Azure portal
AWS CLI and Azure CLI	For the inventory commands and scripted steps. Both clouds also offer a browser shell.	Install the AWS CLI; Install the Azure CLI
AWS to Azure servers	An Azure Migrate project and a replication appliance on a Windows server in AWS.	Set up the Azure Migrate appliance
Azure to AWS servers	The AWS Replication Agent in each VM.	Install the AWS Replication Agent

Need	Details	Where to get it
Object storage copy tool	AzCopy copies from Amazon S3 into Azure Blob storage.	Copy files with AzCopy
Current data transfer pricing	Egress rates for each source region.	Each provider's pricing page; check current pricing

2 Inventory services, not just servers

3 steps

1

Export the resource list

Where: AWS CloudShell (or a terminal with the AWS CLI) for each AWS account and Region; Azure Cloud Shell (or the Azure CLI) for each subscription

List every resource in the source accounts or subscriptions, grouped by application. Include managed databases, object storage, queues, serverless functions, identity, DNS and monitoring. Repeat the AWS command for each Region you use, and run `az account set` for each subscription.

```
aws resourcegroupstaggingapi get-resources --region ap-southeast-2 --output json > aws-resources.json
az resource list --output json > azure-resources.json
```

✓ **You should see:** A JSON file per account or subscription with every tagged resource, which you load into the application inventory.

2

Classify each workload

Where: Workload 7R decision matrix template

Use the 7 Rs: rehost (lift and shift VMs), replatform (managed database on the other side), refactor, repurchase (SaaS), retire, retain or relocate. Managed services usually need replatforming because they do not move as-is.

✓ **You should see:** Every workload has one R and an owner.

3

Measure data volumes

Where: Application inventory, filled from the S3 console (bucket > Metrics tab), Azure portal (storage account > Monitoring > Metrics) and each database's size

Record the size of each database, file share and bucket, and the daily change rate. These numbers drive both the timeline and the data transfer bill.

✓ **You should see:** A size and daily change figure for every data store.

3 Plan data movement and egress

4 steps

Data leaving a cloud provider is usually charged per amount transferred. We do not quote rates because they change. Get them from the provider's current pricing page for your source region.

1

Reduce what moves

Where: Source cloud consoles (S3 lifecycle rules, EBS or managed disk snapshots, log groups) and your inventory

Delete or archive data you do not need, exclude old snapshots and logs, and compress where possible. Only move the latest backup set, not the full history, unless retention rules require it. Get the data owner's approval before deleting anything.

✓ **You should see:** An approved list of data to leave behind, and a smaller total in the inventory.

2

Choose the transfer path

Where: Your Network and DNS worksheet

Small and medium volumes can go over the internet or a site-to-site VPN. Large volumes may justify a private interconnect or an offline transfer device. Compare transfer time against your cutover window.

✓ **You should see:** A transfer path and an estimated transfer time for each data store.

3

Seed first, then sync changes

Where: Your Migration wave plan template

Copy the bulk of the data ahead of time and sync only the changes during cutover. Replication tools (Azure Migrate, MGN, DMS) do this automatically for servers and databases. For object storage, schedule a final delta copy in the cutover window.

✓ **You should see:** Each wave has a seed date and a delta sync step in its cutover runsheet.

4

Ask about exit arrangements

Where: An email to your source provider account team, filed with the business case

Ask your provider account team in writing whether any credits or waivers apply to data transferred out as part of leaving, and what conditions apply. Keep the answer with the business case.

✓ **You should see:** A written answer from the provider.

Budget for double running. You pay for both clouds during the migration. Keep the overlap short by moving in waves and decommissioning each wave promptly.

4 AWS to Azure: servers

4 steps

1

Prepare EC2 instances for discovery

Where: Amazon EC2 console > Security Groups > the instances' group > Edit inbound rules; and on Linux sources, /etc/ssh/sshd_config

Microsoft migrates AWS VMs as physical servers. For discovery, allow WinRM on port 5985 for Windows, and enable SSH password authentication for Linux if you use the discovery appliance. Allow these only from the appliance's IP address.

✓ **You should see:** The Azure Migrate appliance validates the credentials and discovers each instance.

2

Deploy the replication appliance in AWS

Where: Amazon EC2 console > Launch instance (Windows Server 2022), then Azure Migrate project > Execute > Migration > Start execution > From replication appliance (Physical or Others)

Deploy a separate replication appliance on an EC2 instance running Windows Server 2022, ideally in the same VPC as the sources. Allow HTTPS 443 and TCP 9443 from the sources to the appliance, and HTTPS 443 outbound to Azure. In the portal select the target region and **Create resources**, then download the installer and run `DRInstaller.ps1` as administrator on the appliance.

```
.\DRInstaller.ps1
```

✓ **You should see:** The appliance configuration manager opens, registers with the replication appliance key and lists your EC2 instances once you add them.

3

Handle AWS-specific images

Where: Each source EC2 instance (cloud-init configuration on Linux) and your inventory

Amazon Linux cannot be migrated as-is because it is only supported on AWS. Build a supported Linux VM in Azure and move the application instead. Disable AWS-specific cloud-init configuration before migrating.

✓ **You should see:** Amazon Linux workloads are marked rebuild in the 7R matrix, and the remaining instances have no AWS-only cloud-init settings.

4

Replicate, test and migrate

Where: Azure Migrate project > Execute > Migration > Start execution, then Execute > Migrations > the server > Testing and Completion

Follow the agent-based flow in Azure Migrate: replicate, run a test migration into an isolated virtual network, then migrate and complete. See the VMware to Azure guide for the detailed steps.

✓ **You should see:** Each server reaches the Completion stage and runs as an Azure VM.

5 Azure to AWS: servers

4 steps –

1

Install the AWS Replication Agent in each VM

Where: AWS Transform MGN console in the target Region (initialise once), then each Azure VM as Administrator or root

AWS Transform MGN replicates from an agent inside the operating system, so the same process applies to Azure VMs as to on-premises servers. Check the source OS is on the MGN supported list first, initialise the service, then follow install the AWS Replication Agent on each VM.

```
aws mgn initialize-service --region ap-southeast-2
```

✓ **You should see:** Each VM appears on the MGN Source servers page and starts initial sync.

2

Allow outbound replication traffic

Where: Azure portal > each VM's network security group > Outbound security rules > Add (and any Azure Firewall)

Allow the Azure VMs outbound TCP 443 to the MGN endpoint and Amazon S3, and TCP 1500 to the replication servers in your AWS staging subnet. Use a site-to-site VPN if you do not want replication over the internet.

✓ **You should see:** The servers move past initial sync to Ready for testing with Healthy replication.

3

Remove Azure-specific agents after cutover

Where: Each launched EC2 instance: Control Panel > Programs and Features (Windows) or the package manager (Linux)

Uninstall the Azure VM agent and Azure monitoring extensions from the launched EC2 instances, and install the AWS equivalents (SSM agent, CloudWatch agent). MGN post-launch actions can install the SSM agent for you.

✓ **You should see:** No Azure agents remain, and the instance appears as a managed node in AWS Systems Manager.

4

Test and cut over

Where: AWS Transform MGN console > Source servers > Test and Cutover menu

Launch test instances, validate, then cut over and finalise. The VMware to AWS guide has the console and CLI steps.

✓ **You should see:** Each server reaches the CUTOVER lifecycle state after finalising.

6 Move databases and storage

3 steps –

1

Databases

Where: The target database service (Azure SQL Managed Instance or Amazon RDS) and the matching guide

For SQL Server into Azure, Azure SQL Managed Instance supports sources such as Amazon EC2 and Amazon RDS for SQL Server through Log Replay Service. Into AWS, use AWS DMS or native backup and restore. See the Azure SQL guide and the Amazon RDS guide.

✓ **You should see:** Each database has a chosen method and a test restore completed in the target.

2

Object storage: Amazon S3 to Azure Blob

Where: A terminal with AzCopy (copy files with AzCopy), signed in with azcopy login

Set the AWS access key as environment variables, then copy the bucket to a Blob container. Microsoft says the copy runs server to server, so it does not use your machine's bandwidth. For the other direction (Azure to S3), use a copy tool that can read from Azure and write to S3. Verify object counts and checksums after the copy, and run a final delta copy during cutover.

```
$env:AWS_ACCESS_KEY_ID="<access-key>"
$env:AWS_SECRET_ACCESS_KEY="<secret-access-key>"
azcopy copy 'https://s3.amazonaws.com/mybucket'
'https://mystorageaccount.blob.core.windows.net/mycontainer' --recursive=true
```

✓ **You should see:** AzCopy reports the job as Completed, and the object count in the container matches the bucket.

3

Identity and DNS

Where: Target cloud IAM (roles or managed identities), the secrets store, and your DNS host

Recreate IAM roles or managed identities, secrets and certificates in the target. Lower DNS TTLs well before cutover (lower DNS TTL).

✓ **You should see:** Every identity and secret in the inventory has a target equivalent, and DNS records show the lowered TTL.

7 Repatriation to on-premises or colocation

3 steps

1

Size and buy the target platform

Where: TCO and cost comparison template, filled from CloudWatch or Azure Monitor utilisation data

Size hosts and storage from actual cloud utilisation, not provisioned sizes. Allow for growth, redundancy and the hypervisor's cluster requirements.

✓ **You should see:** A host and storage specification based on measured peaks, with headroom documented.

2

Pick the move method per workload

Where: Workload 7R decision matrix template, then a test host on your hypervisor

Options are a clean rebuild with data restored from backup, exporting disk images and importing them to your hypervisor, or using backup software that can restore cloud VM backups to on-premises hypervisors. Test each method on one workload first.

✓ **You should see:** Each workload has a method that has been tested once end to end.

3

Replace managed services

Where: Your target-state architecture, one row per managed service in the inventory

Managed databases, load balancers, queues and object storage need self-managed equivalents, plus the patching, backup and monitoring the provider used to do for you.

✓ **You should see:** Every managed service has a named replacement and an owner for patching and backup.

8 Cut over and decommission

3 steps −

1

Run a cutover per wave

Where: Your Cutover runsheet, the migration tool console and your DNS host

Freeze changes, stop writes, run final syncs, switch DNS and endpoints, run acceptance tests and get sign-off.

✓ **You should see:** Every runsheet line is ticked and the application owner has signed off.

2

Keep a rollback window

Where: Source cloud console (stop, do not terminate or delete)

Keep source resources stopped but not deleted until the rollback window closes.

✓ **You should see:** Source instances show as stopped, with their disks intact.

3

Delete source resources

Where: Source cloud console and billing (AWS Billing and Cost Management or Azure Cost Management)

Delete compute, disks, snapshots, backups and buckets you no longer need, and close accounts or subscriptions once billing shows no remaining charges.

✓ **You should see:** The source billing shows no new charges for the migrated workloads.

WORTH KNOWING

What can go wrong, and how do you avoid it?

The detail behind the plan: options compared, and what experienced engineers check first. Tap any item.

Compare all the ways to do this

4 options –

Rehost servers with the target cloud's native tool

MEDIUM EFFORT

Best for: Virtual machines that can run unchanged on the other cloud

+ AWS to Azure uses Azure Migrate with EC2 instances treated as physical servers
+ Azure to AWS uses AWS Transform MGN with the AWS Replication Agent in each VM
+ Continuous replication and test launches before cutover
– Managed services do not move with the servers
– Some images, such as Amazon Linux, have to be rebuilt
– Every replicated byte counts towards data transfer out of the source cloud

Read more →

Replatform managed databases and storage

MEDIUM EFFORT

Best for: Managed databases, object storage and other services with an equivalent on the target cloud

+ Azure Database Migration Service, AWS DMS or native backup and restore move the data
+ AzCopy copies Amazon S3 buckets to Azure Blob Storage directly
+ You stay on managed services, with no new patching work
– S3 metadata and bucket names can change in transit
– Each service needs its own method, testing and cutover

Read more →

Rebuild on the target cloud

HIGH EFFORT

Best for: Workloads built with infrastructure as code, containers or serverless functions, and images that cannot be replicated

+ Clean build with no source cloud agents to remove
+ Chance to adopt target-native services
– Deployment code, identities and secrets must be rewritten for the target
– Data still has to be copied and synchronised separately

Repatriate to on-premises or colocation

HIGH EFFORT

Best for: Steady, predictable workloads you have decided to run on your own hardware

+ You choose the hypervisor and hardware
+ Methods include a clean rebuild, disk image export and import, or backup software that restores cloud VM backups to your hypervisor
– Managed services need self-managed replacements, plus the patching, backup and monitoring the provider did
– Hardware has to be sized, bought and installed before the move

Read more →

OUR ADVICE

Classify each workload first: most estates use more than one approach. Rehost plain VMs with the target's native tool, replatform managed databases and storage, and rebuild anything already defined in code. Measure data volumes before you choose, because data transfer out of the source cloud often decides the order and the timeline.

Measure egress before you commit to a date

COST LEVER –

Data transfer out of the source cloud is the cost most often missed, and it scales with every copy, retry and delta sync.

WHAT TO DO

Total the data per wave (Step 1), add the expected change during replication, and apply the source provider's current transfer rates. Get any exit credit terms in writing before the first copy.

Clean up before you copy

COST LEVER

Old snapshots, log archives, backup histories and abandoned buckets often outweigh the live data. Moving them costs egress and time and then costs storage in the target.

WHAT TO DO

List snapshots, log groups and buckets with no recent access, get owner sign-off, and delete or leave them behind. Keep only the backup sets your retention rules require.

Amazon Linux has to be rebuilt

SAVES TIME

Microsoft says Amazon Linux cannot be migrated to Azure as-is because it is only supported on AWS.

WHAT TO DO

Find Amazon Linux instances in the inventory early, mark them rebuild, and plan time to build a supported distribution in Azure and move the application and data.

Microsoft Learn: Discover, assess and migrate AWS VMs to Azure

Open the right ports in security groups and NSGs

SAVES TIME

Agent-based replication fails quietly when ports are missing: Azure Migrate needs 443 and 9443 to its replication appliance, and AWS says MGN needs 443 to the endpoint and 1500 to the replication servers.

WHAT TO DO

Add the rules to the source security groups, NACLs or NSGs before installing agents, scoped to the appliance or staging subnet only.

AWS docs: Network requirements for AWS Transform MGN

Managed services do not move with the servers

REDUCES RISK

Queues, serverless functions, managed databases, load balancers and identity have no lift-and-shift path between clouds.

WHAT TO DO

Give each managed service its own line in the 7R matrix with a target equivalent and a test, and budget refactoring time for those with no direct equivalent.

TOOLS THAT CAN GET AROUND THIS

Managed databases are the exception: database migration services copy the data from one cloud's managed database into the other, online or offline. Queues, functions, load balancers and identity still need the refactoring plan above.

AWS Database Migration Service

Supports Azure SQL Database, Azure Database for PostgreSQL and MySQL flexible server, and Google Cloud SQL for MySQL and PostgreSQL as sources.

Azure Database for PostgreSQL migration service

Migrates Amazon RDS for PostgreSQL, Amazon Aurora PostgreSQL and Google Cloud SQL for PostgreSQL, offline or online. Online needs primary keys on all tables.

Third-party tools, not endorsed by the platform vendor. Confirm in the tool's documentation and run a pilot first.

S3 metadata and bucket names can change in transit

REDUCES RISK

Microsoft says AzCopy drops S3 metadata keys that are invalid in Azure by default, and changes bucket names that contain periods or repeated hyphens. Applications that rely on metadata or exact names can break.

WHAT TO DO

Check whether applications read object metadata. If so, run AzCopy with `--s2s-handle-invalid-metadata=FailIfInvalid` on a test bucket first, and fix names before the real copy.

Microsoft Learn: Copy data from Amazon S3 to Azure Storage by using AzCopy

Recreate identities and secrets before the test migration

REDUCES RISK

IAM roles, managed identities, key vault secrets and certificates are cloud-specific. A server that boots fine can still fail every call it makes.

WHAT TO DO

Map each identity and secret in the inventory to its target equivalent and test them during the test migration, not at cutover.

Remove source cloud agents after cutover

PERFORMANCE

Agents from the source cloud keep trying to reach their old endpoints, fill logs and can slow boot. Missing target agents leave the server unmanaged.

WHAT TO DO

After cutover uninstall the source agents (Azure VM agent, AWS SSM and CloudWatch agents) and install the target ones, and disable source-specific cloud-init settings.

Keep the double-running period short

COST LEVER

You pay for both clouds while workloads overlap, including stopped instances' disks and snapshots.

WHAT TO DO

Set a fixed rollback window per wave, then delete source compute, disks, snapshots and buckets, and watch the source bill until it shows no charges.

Pick an object copy tool for the Azure to AWS direction

SAVES TIME

AzCopy copies directly from Amazon S3 into Azure Storage, but it has no Amazon S3 destination. For Azure to AWS moves, or repatriation to your own S3-compatible storage, it would mean downloading blobs to a server and uploading them again.

WHAT TO DO

Choose the transfer tool for each direction in Step 1 and test it on one container or bucket, checking object counts, metadata and tags, before the real copy.

TOOLS THAT CAN GET AROUND THIS

AWS DataSync copies Azure Blob Storage straight into Amazon S3, and rclone copies between Azure Blob Storage, Amazon S3 and S3-compatible storage in either direction, including storage in your own data centre.

AWS DataSync

Transfers block, page and append blobs from Azure Blob Storage to Amazon S3, without an agent in Enhanced mode, and copies object tags by default. Archive tier objects must be rehydrated first, the cold tier is not supported, and objects with more than 2 KB of metadata do not transfer to S3.

rclone

Open-source command-line tool with Azure Blob Storage and Amazon S3 (and S3-compatible) backends, so rclone sync can copy a container to a bucket and later passes copy only changes.

Third-party tools, not endorsed by the platform vendor. Confirm in the tool's documentation and run a pilot first.

Microsoft Learn: Copy data from Amazon S3 to Azure Storage by using AzCopy

FINISH SAFELY

How do you check it worked, or roll back?

Full validation checklist

8 checks

Run these before you declare the migration complete. Record the results in the data validation report.

Check	How to check	What good looks like
Every resource in the inventory has a target or a decision	Compare the `aws-resources.json` or `azure-resources.json` export from Step 1 with the Workload 7R decision matrix and a fresh export from the target cloud (`az resource list` or `aws resourcegroupstaggingapi get-resources`).	Each source resource maps to a target resource, a managed service replacement, or a recorded retire or retain decision.

Check	How to check	What good looks like
Migrated servers boot and run cleanly	In the target console (Azure portal VM > Overview and Boot diagnostics, or the Amazon EC2 console > Instances > Status checks), check each migrated server, then sign in and check the application services are running.	All status checks pass, the application services are running, and source cloud agents have been replaced with the target ones.
Object storage counts and sizes match	For an S3 source, run `aws s3 ls s3://mybucket --recursive --summarize`. For the Azure target, run `azcopy list 'https://mystorageaccount.blob.core.windows.net/mycontainer' --running-tally`. Compare totals and review the AzCopy job log for failed transfers.	Object count and total size match after the final delta copy; any failed or skipped objects are listed and explained.
Databases match the source	Compare row counts and checksums for key tables on source and target, using the checks in the Azure SQL or Amazon RDS guide, and record them in the data validation report.	Counts and checksums match for every key table at the cutover point.
DNS points at the new endpoints	From a PC outside your network, run `Resolve-DnsName app.contoso.com` (or `nslookup app.contoso.com`) for each record you changed, and check the TTL has expired since the change.	Every record returns the target address and no client still resolves the source address.
Identities, secrets and certificates work in the target	Check application logs in the target for authentication failures, and test each managed identity or IAM role by running the job or call that uses it.	No access denied errors; every secret and certificate in the inventory is in the target store and not expired.
Users and integrations work end to end	Run the acceptance tests in your Cutover runsheet: user sign-in, key transactions, scheduled jobs, inbound and outbound integrations, and email the application sends.	Every test passes and each application owner signs off.

Check	How to check	What good looks like
The source is no longer receiving writes	In the source cloud, check Amazon CloudWatch or Azure Monitor metrics for network in, disk writes and database connections on the stopped or frozen resources after cutover.	Metrics show no traffic or writes after the cutover time.

Rollback plan

If it goes wrong

ROLL BACK IF

- Acceptance tests fail on a critical application and cannot be fixed within the cutover window
- Data validation shows missing or mismatched data that a final sync cannot fix
- Users or integrations cannot reach the application after DNS changes
- Performance in the target is far below the agreed baseline under real load

POINT OF NO RETURN

Each wave can be rolled back while the source resources are stopped but not deleted and no new data has been written only to the target. Two things end that: users writing to the target (rolling back then means copying those changes back), and deleting the source compute, disks, snapshots, backups and buckets. Finalising cutover in AWS Transform MGN or completing migration in Azure Migrate also stops replication and removes the replicated data, so you cannot simply resume.

1

Stop writes to the target

Where: The application, in the target cloud

Put the application into maintenance mode or stop its services so no more data is written to the target while you roll back.

✓ **You should see:** No new transactions in the target logs or database.

2

Restart the source resources

Where: Source cloud console (Amazon EC2 console > Instances > Instance state > Start, or Azure portal > the VM > Start)

Start the stopped source servers and databases in the order your runsheet gives (databases first, then application tiers).

✓ **You should see:** Source status checks pass and the application responds on its old address.

3

Revert the cutover in the migration tool

Where: AWS Transform MGN console > Source servers, or Azure Migrate project > Execute > Migrations

If you have not finalised the cutover in MGN, use the Test and Cutover menu to revert the server so replication continues. In Azure Migrate, if you have not selected Complete migration, replication is still in place and you can migrate again later.

✓ **You should see:** The server shows a replicating state again, ready for a new test or cutover.

4

Point DNS back to the source

Where: Your DNS host

Change each record back to the source address. Because you lowered TTLs before cutover, clients follow within the low TTL.

✓ **You should see:** `Resolve-DnsName app.contoso.com` returns the source address.

5

Copy back any data written to the target

Where: The database or storage tool used for the wave (AWS DMS, Log Replay Service, AzCopy)

If users wrote to the target before rollback, export those changes and apply them to the source before reopening it, or agree with the data owner to accept the loss in writing.

✓ **You should see:** The source holds every transaction made since cutover, or the data owner has signed off on the gap.

6

Record what cannot be undone

Where: Your Rollback plan and Cutover runsheet

Note that deleted source resources, snapshots and buckets cannot be recovered. Only delete them, and close accounts or subscriptions, after the rollback window for that wave has closed.

✓ **You should see:** Deletion is scheduled after the rollback window with owner sign-off.

Agree the triggers and who decides before the go / no-go meeting. Use the rollback plan template.

Common problems and fixes

5 errors −

`Azure Migrate replication of EC2 instances stalls`

Cause: TCP 9443 or HTTPS 443 is blocked between the sources and the replication appliance, or a stale duplicate appliance is registered.

Fix: Open the ports in the security groups and NACLs and remove stale appliances from the Azure Migrate project.

`Amazon Linux VM fails to boot in Azure`

Cause: Amazon Linux is only supported on AWS.

Fix: Build a supported Linux distribution in Azure and migrate the application and data instead.

`Migrated VM has no network after boot`

Cause: Cloud-init or the guest network configuration is specific to the source cloud.

Fix: Disable source-specific cloud-init configuration before migrating and use DHCP in the guest.

`Unexpectedly large data transfer bill`

Cause: Full histories, snapshots or logs were copied, or data was copied more than once.

Fix: Clean up data before transfer, seed once and sync only changes. Monitor billing daily during the migration.

`Application fails after cutover with authentication errors`

Cause: Roles, managed identities or secrets were not recreated in the target cloud.

Fix: Map every identity and secret during inventory and test them in the test migration.

Search the full error library →

FAQ

Frequently asked questions

Can Azure Migrate move AWS EC2 instances?

Yes. Microsoft documents migrating AWS VMs to Azure by treating them as physical servers, using a separate replication appliance.

Can AWS Transform MGN move Azure VMs?

MGN replicates from an agent inside the operating system, so it can replicate VMs in other clouds that run a supported OS and can reach AWS.

How do I estimate egress costs?

Measure the data you will move, including the final delta sync, and apply the source provider's current data transfer rates for that region.

Is repatriation cheaper than cloud?

It depends on utilisation, staffing and hardware lifecycle. Compare full costs, including data centre, support and staff time, with current cloud pricing.

Do managed databases move with the servers?

No. Managed databases need a database migration tool or native backup and restore into the target service.

SOURCES

Official sources

8 official pages this guide is based on

Vendors change limits and features, so confirm details on the source before you rely on them. Last checked 4 Nov 2026.

- Microsoft Learn: Discover, assess and migrate AWS VMs to Azure
- Microsoft Learn: Migrate machines as physical servers to Azure
- Microsoft Learn: SQL Server to Azure SQL Managed Instance migration overview
- AWS docs: Installing the AWS Replication Agent on Linux servers
- AWS docs: AWS Application Migration Service endpoints and quotas
- AWS: AWS Database Migration Service FAQs
- AWS docs: Network requirements for AWS Transform MGN

Microsoft Learn: Copy data from Amazon S3 to Azure Storage by using AzCopy