Cloud migration rehearsal and cutover runbook

A practical worksheet that links task order and data authority to cutover and rollback decisions

A practical IXC resource for adoption decisions and delivery

Published 2026.09 · English edition · PDF + Excel worksheets

Using this resource

Sets the migration order and verification evidence for a live system and records decisions to start, hold or roll back the cutover.

Technology leaders approving a cloud migration, engineers carrying out the work and staff verifying business functions

Contents

01 Define the business functions and dependencies that move together 3

02 Record rehearsal durations and blocking conditions 4

03 Define the evidence needed to move to the next stage 5

04 Separate rollback paths before and after new writes 6

05 Approve business recovery and source retirement separately 7

06 Define the migration engagement scope in the runbook 8

How to complete and reuse it

Read the criteria in this PDF, then fill the blank fields with your actual conditions and anything still unconfirmed. The Excel file for the same practice area has a worksheet for each resource. Keep the worked examples separate from real customer cases and confirmed quotes.

With attribution, you may quote and adapt this material for internal review and your own work. The latest files are at www.ixc.co.kr/en/assets.

Define the business functions and dependencies that move together

A migration unit includes files, scheduled jobs and external integrations as well as servers and databases. Identify who reads and writes the data and the impact of a business disruption, then define the groups to cut over together.

Migration group definitions

Target and connection	Authority before cutover	Authority after cutover	Reviewer
Worked example: documents and versions	Source version DB	Destination version DB	App owner
Worked example: approval state	Source approval history	Destination approval history	Business owner
Worked example: attachments and permissions	Source storage and permission table	Destination storage and permission table	Data owner
Worked example: notification webhook	Source delivery queue	Destination delivery queue	Integration owner

If document versions and approval states differ, what users reviewed diverges from what was actually approved. Compare the document API, approval history, attachment access permissions and approval notification webhooks within the same migration group. Define write owners and workers so that approval requests and notification deliveries do not run in duplicate across both environments during cutover.

> **Worked example: B2B document approval and collaboration**
>
> In a fictional team collaboration service, an author submits a document and a designated approver checks the version and approves it. The times, counts and results in the tables are worked examples, and production execution remains not performed. In a real plan, fill in the versions, approval history, permission scope, change window and evidence checked.

Record rehearsal durations and blocking conditions

A rehearsal plan needs test data, the external actions to block, a starting state and expected results. Keep measured results separate from conditions not yet confirmed in production.

Fictional rehearsal record	Start→end	Expected result	Example record
Document and approval synchronization	21:02→21:08	40 documents, approval history matches	Final approval event AP-040
Attachment and permission comparison	21:08→21:10	12 attachments, allow/deny results match	Hashes and role-based access checked
Document approval flow	21:10→21:12	v3 approved by designated approver	Author's approval attempt denied
Notification webhook preparation	21:12→21:13	Delivery owned by one environment	Source stopped, destination on standby

Rehearsal results and production differences log

Task and verification	Execution time and evidence	Conditions unlike production	Remediation owner and deadline
Synchronization and comparison			
Permissions and external integrations			
Traffic and business verification			
Rollback and follow-up reconciliation			

The 11-minute total in the fictional record is not a committed duration for the production migration. In an actual rehearsal, record the number of document versions, the volume of approval changes and attachment sizes, along with the permission and webhook conditions. If external notifications went to a test receiver, confirming delivery to production receivers remains a separate open item. Classify the status as not performed, passed, failed or blocked.

Define the evidence needed to move to the next stage

Cutover transfers real user requests and authoritative writes to the destination. Approve the next task only after confirming each stage's results; do not cut over just because time has passed.

Stage	Action	Condition for next stage	Role
Readiness check	Compare versions, roles and contacts	No open blocking items	Lead
Write control	Control submissions, approvals and webhooks	No new writes in the source	App and operations
Final synchronization	Compare documents, approvals and permissions	Missing items and mismatches resolved	Data
Traffic switch	Change to the approved route	Destination responses confirmed	Infrastructure
Business check	Verify versions, approvals and attachments	Allow and deny results match	QA and business
Enable writes	Make the destination the sole authority	Source writes stay blocked	Lead and operations

> **Worked example: hold for missing approval history**
>
> Version v3 of document DOC-018 was approved in the source but shows as awaiting approval in the destination. Source approval event AP-041 is missing in the destination, which reflects events only up to AP-040. Even though 40 documents and 12 attachments match, approval to proceed is put on hold. Decide after rechecking the approval history and role permissions.

Compare not only document counts but also the current version, the version submitted for approval, the approval state and the event ID. Matching attachment hashes are not a pass if permissions have broadened. Test the example policy, which allows the designated approver and denies self-approval by the author, on the same document. Set the actual access policy from the approval conditions of the organization concerned.

Separate rollback paths before and after new writes

Once the destination holds new business data, routing only the requests back to the source leaves those changes behind. Choose rollback or a forward fix based on whether new writes and external processing have occurred.

Current situation	Check first	Decision options	Approving role
Before new writes begin	Source environment and data state	Consider rollback to the source route	Technical lead
New versions or approvals created	Approved version and reverse synchronization feasibility	Consider rollback after comparison	Business and data
Notification webhook delivered	Approval event ID and duplicate risk	Consider follow-up notice and forward fix	Business and integration
Not enough time to roll back	Remaining work and business impact	Switch to the contingency plan	Technical and business leads

Worked example: rollback decision deadline

Assuming the change window ends at 21:30, the rollback procedure takes 8 minutes and business checks after rollback take 5 minutes, the last decision time is 21:17. If mandatory verification has not finished at 21:17, do not approve continuing with the original plan. Check whether the 13 minutes are available and the state of the data, then choose the rollback or contingency procedure agreed in advance.

Proceed, hold and rollback decision log

Decision time and stage	Observed values and new writes	Decision and executor	Approval and follow-up check
Worked example: 21:12	AP-041 missing, no new writes	Hold, data owner	Recheck approval history
Worked example: 21:17	Mandatory permission verification incomplete	Stop proceeding, lead	Check rollback readiness

Approve business recovery and source retirement separately

Even after connections to the destination succeed, confirm that document versions, approval states, attachment permissions and notification webhook results match. Retiring the source is a separate decision that reviews the conditions for retention, rollback and open items.

Business recovery and source retirement checks

What to check	Evidence required	Results and open items	Reviewer
Critical business functions and permissions	Reproduction record per business function		
Changed data	Source and destination comparison results		
Notifications and external integrations	Event IDs and processing status		
Monitoring and contacts	Destination monitoring and contact test		
Source retirement	Retention, contract and return lists		

Worked example: a closing record that keeps conditions separate

The fictional record confirmed document v3, approval history, attachment permissions and a single running notification worker. It keeps the result of the test receiver applying the same approval event once. Delivery to production receivers is left unconfirmed, and the record states that business resumes after an approval that has reviewed its impact and temporary handling.

While writes to the source stay blocked, assign owners for lookups, retention and cost. Do not handle service resumption and account deletion with the same checkbox. Link the resources to decommission, the backups and logs to keep, the contracts to end and the permissions to revoke to the operations handover list.

Define the migration engagement scope in the runbook

Give whoever runs the migration the business functions that will change and the completion criteria, not only the current configuration. Deciding who owns design, data movement, verification and stabilization lets you compare proposals on the same scope.

Worked example: requested scope	Internal team	Requested IXC review	Results to keep
Document approval service migration	Roles and business schedule	Dependencies and migration sequencing	Migration groups and task sheet
Moving versions and approval history	Comparison scope and access approval	Synchronization and rehearsal	State comparison and open items
Attachment and task permission verification	Allow and deny policy	Verification scope and retest	Scenarios and verdicts
Notifications and operational stabilization	Delivery conditions and retention decisions	Webhook comparison and follow-up actions	Operations handover record

IXC's Cloud Migration service covers dependency assessment, migration sequencing, rehearsal, cutover and stabilization. The technical review and the actual delivery scope are agreed based on the runbook's undecided items. Ownership of assets, permissions and recovery after the migration carries over to the cloud operations handover checklist.

> **Applying the official guidance**
>
> AWS cutover guidance describes setting the task order and owners, along with verification and rollback plans, in advance. Rolling back after new data has arrived must also account for how that data is handled. The document approval service worked example in this resource does not assume any specific provider's tools or templates.

IXC

Publication and use

Working from the business functions in scope, your current environment and your data change paths, we review with you the rehearsal, cutover and rollback decisions as well as the stabilization scope.

IXC Corp.

CEO Nam Kichul | Business registration no. 109-86-56732
4F 408, Chuncheon ICT Venture Center
7 Huseok-ro 462beon-gil, Chuncheon-si, Gangwon-do, Republic of Korea
+82-33-242-0210 | info@ixc.co.kr

www.ixc.co.kr

www.ixc.co.kr/en/services/cloud-migration

www.ixc.co.kr/en/assets/cloud-migration-runbook

References

AWS Prescriptive Guidance: Pre-cutover stage

https://docs.aws.amazon.com/prescriptive-guidance/latest/best-practices-migration-cutover/pre-cutover-stage.html

AWS Prescriptive Guidance: Cutover stage

https://docs.aws.amazon.com/prescriptive-guidance/latest/best-practices-migration-cutover/cutover-stage.html

IXC Cloud Migration

https://www.ixc.co.kr/en/services/cloud-migration

IXC: Building a deployment you can undo — code, configuration, database, and traffic together

https://www.ixc.co.kr/en/insights/rollback-ready-deployment