



CHECKLIST

AWS Virtual Machine Migration Zero-Downtime Checklist

A practical, vendor-neutral checklist for planning and executing zero-downtime VM migrations in AWS, with validation, cutover, and rollback steps.

AWS Virtual Machine Migration Zero-Downtime Checklist

Use this checklist to plan a production VM migration in AWS with minimal user-visible interruption. It is designed for stateless, lightly stateful, and strongly stateful workloads and focuses on cutover readiness, validation, and rollback safety.

1) Classify the workload

- ☐ Identify whether the workload is stateless, lightly stateful, or strongly stateful.
- ☐ Document all upstream and downstream dependencies:
- ☐ Database
- ☐ File shares / object storage
- ☐ Identity provider / authentication
- ☐ API gateways / reverse proxies
- ☐ DNS
- ☐ Batch jobs / schedulers
- ☐ Monitoring / alerting
- ☐ Confirm whether the application can tolerate:
- ☐ Brief connection resets



- ☐ Session re-establishment
- ☐ Replication lag
- ☐ Short write freeze during cutover
- ☐ Define the acceptable user impact target:
- ☐ No visible outage
- ☐ Brief reconnect only
- ☐ Limited maintenance window

2) Choose the migration pattern

- ☐ Select the primary migration approach:
- ☐ Live replication with cutover
- ☐ Blue-green replacement
- ☐ Incremental coexistence / phased migration
- ☐ Confirm why this pattern fits the workload:
- ☐ Fast rollback required
- ☐ Data consistency is the main constraint
- ☐ Multiple dependent systems must move together
- ☐ Traffic can be switched through a reversible mechanism
- ☐ Verify the pattern supports the chosen rollback method.
- ☐ Record the cutover trigger criteria and stop conditions.

3) Pre-stage the target environment

- ☐ Build the target VM with matching or approved-equivalent settings.
- ☐ Confirm operating system version, patches, and required agents.



- ☐ Recreate storage layout, mount points, and filesystem behavior as needed.
- ☐ Apply network policy, security controls, and firewall rules.
- ☐ Confirm IAM roles, instance profiles, and access boundaries.
- ☐ Set up monitoring, logs, and alert routing on the target.
- ☐ Validate time synchronization and hostname / DNS expectations.
- ☐ Confirm any required encryption, certificates, and secrets are available.

4) Replicate and synchronize data

- ☐ Start block, file, or application-level replication as appropriate.
- ☐ Measure initial sync time and steady-state replication lag.
- ☐ Confirm application writes are being captured correctly.
- ☐ Verify that data drift is within the acceptable threshold.
- ☐ Identify any data that requires a final sync or freeze window.
- ☐ Confirm the source can remain in standby until the target is stable.

5) Validate the target before traffic shift

- ☐ Test application startup on the target VM.
- ☐ Confirm services and daemons start cleanly.
- ☐ Validate dependency reachability from the target:
 - ☐ Database connectivity
 - ☐ File share access
 - ☐ Authentication / identity
 - ☐ External API reachability
- ☐ Check application configuration for environment-specific values.



- ☐ Confirm logs show no critical errors or unexpected retries.
- ☐ Run health checks and synthetic transactions.
- ☐ Validate user authentication and session behavior.
- ☐ Confirm performance is within expected thresholds.
- ☐ Verify that security controls still work after migration.

6) Prepare the cutover plan

- ☐ Choose a reversible traffic mechanism:
- ☐ Load balancer
- ☐ Reverse proxy
- ☐ DNS change
- ☐ Route or policy update
- ☐ Define the exact cutover steps in order.
- ☐ Assign owners for:
- ☐ Change execution
- ☐ Application validation
- ☐ Infrastructure validation
- ☐ Communication
- ☐ Rollback decision
- ☐ Set a freeze period for unrelated changes.
- ☐ Confirm the source system stays ready for rollback.
- ☐ Prepare user communications and support coverage if needed.
- ☐ Establish the rollback threshold before the move starts.

7) Execute the cutover



- ☐ Perform a final replication sync or consistency checkpoint.
- ☐ Confirm the target is healthy immediately before traffic shift.
- ☐ Shift a small portion of traffic first if a staged move is possible.
- ☐ Move production traffic using the selected reversible method.
- ☐ Watch for:
 - ☐ Connection resets
 - ☐ Authentication failures
 - ☐ Latency spikes
 - ☐ Elevated 4xx / 5xx errors
 - ☐ Session loss
- ☐ Keep the source available until the target proves stable.

8) Validate after cutover

- ☐ Confirm real user traffic is reaching the target.
- ☐ Check application availability from multiple user paths.
- ☐ Review error rates, latency, CPU, memory, and I/O.
- ☐ Confirm database and file dependency behavior remains correct.
- ☐ Validate key business transactions end to end.
- ☐ Confirm logs, alerts, and dashboards are functioning.
- ☐ Watch for delayed issues caused by session handling or caches.
- ☐ Keep the rollback window open until the stability threshold is met.

9) Rollback readiness

- ☐ Verify rollback steps are documented and tested.



- ☐ Confirm you can restore traffic to the source quickly.
- ☐ Define what conditions require rollback:
- ☐ Sustained error-rate increase
- ☐ Failed authentication
- ☐ Data inconsistency
- ☐ Critical dependency failure
- ☐ Performance degradation beyond threshold
- ☐ Confirm rollback will not overwrite unreconciled data.
- ☐ Keep source state intact until post-cutover signoff.

10) Post-migration closeout

- ☐ Record the actual cutover time and observed impact.
- ☐ Confirm the source VM can be decommissioned or retained temporarily.
- ☐ Update documentation, runbooks, and dependency maps.
- ☐ Review alerts triggered during migration and tune thresholds if needed.
- ☐ Capture lessons learned:
- ☐ What validated well
- ☐ What caused risk or delay
- ☐ What should change next time
- ☐ Schedule a follow-up review for any deferred cleanup tasks.

Quick decision guide

- Choose blue-green when rollback speed matters most and the workload can be validated on a parallel target.



- Choose live replication when data sync and cutover timing are the main concerns.
- Choose phased coexistence when multiple dependent systems must move in stages.

Final sign-off

- ☐ Workload classified
- ☐ Target environment staged
- ☐ Replication verified
- ☐ Validation completed
- ☐ Cutover plan approved
- ☐ Rollback plan ready
- ☐ Production traffic shifted successfully
- ☐ Stability confirmed
- ☐ Documentation updated

Migration owner: _____

Validation owner: _____

Cutover date: _____

Rollback window end: _____