

# The Ultimate Checklist for Upgrading to MarkLogic 11/12

A step-by-step guide to ensure a smooth, secure, and future-ready upgrade.



# Why This Checklist Matters

Upgrading to MarkLogic 11/12 brings performance, security, and compliance benefits, but without proper preparation, even minor issues can cause significant delays.

This checklist has been created by Datavid's experts based on real-world upgrade projects to help you:



**Prepare your platform and team**



**Avoid common pitfalls**



**Ensure a smooth upgrade from version 9/10 to 11/12**



## 0. Big-Picture Prep

- Read the MarkLogic 11/12 Release Notes & Compatibility Matrix and highlight breaking changes
- Look for any changes that might cause reindexing and plan for the same (baseline language change, geo indexes .etc. )
- Identify deprecated APIs / libraries in your codebase; create migration tickets
- Choose between an in-place rolling upgrade or a new, clean cluster
- Verify the underlying OS, Java, drivers, storage, and load-balancer versions are supported by ML 11/12
- Book a maintenance window and notify business owners of the cut-over plan
- Define rollback criteria, plan and owner
- Freeze production schema changes during the upgrade window
- Plan to freeze and catch up with ingestions pre and post upgrade



## 1. Dev / Local & QA

- Create a feature branch for ML 11/12 upgrades in each repo
- Upgrade Roxy/roxy-gradle or ml-gradle scripts to target ML 11/12; run unit tests
- Spin up local ML 11/12 containers / VMs; load sample data
- Run application smoke-tests & automation; fix issues
- Identify the current DHF version and check compatibility with ML 11/12
- Review DHF release notes for breaking changes or migration steps
- Upgrade DHF libraries/tools in local/dev environment to a supported version for ML 11/12
- Run DHF data flows and tests against ML 11/12 local environment; fix any issues
- Document any custom DHF extensions or custom steps that require changes for ML 11/12
- Promote to a QA environment/cluster on ML 11/12 and let QA do regression tests and sign off



## 2. Staging / UAT Environment

- Full backup of existing ML 10 staging forests (keep off-box)
- Provision a new ML 11/12 staging cluster
- Export configuration from ML 10 via Configuration Manager / Admin API
- Import configuration into ML 11/12 cluster; create app-servers & databases
- Restore data from backup
- Deploy the latest ML 11/12-ready application packages
- Create fresh staging endpoints / DNS entries
- Run functional, integration, and end-to-end tests
- Stakeholder UAT & change board approval
- Upgrade DHF in the staging environment; validate flows, mappings, and security roles
- Run specific performance tests for areas of known incompatibilities



### 3. Production Cut-over

- Announce outage; switch all prod clusters to read-only
- Take final incremental & full backups of ML 10 databases
- Provision a new ML 11/12 production cluster
- Export ML 10 config → import into ML 11/12; validate forest mounts
- Disable rebalancing and indexing before restoring
- Restore the latest data backups to ML 11/12
- Deploy ML 11/12 application builds; run sanity scripts
- Synchronize delta data from ML 10 → ML 11/12
- Redirect traffic/swap DNS or load-balancer target to ML 11/12
- Disable scheduled tasks, CPF flows, and cron jobs on the old cluster; enable on the new one
- Monitor logs, CPU, IOPS, and cache hit rates
- Obtain production sign-off; decommission the ML 10 cluster
- In production cut-over, coordinate DHF upgrade/deployment with MarkLogic upgrade steps
- Post-upgrade, run DHF data ingestion, harmonization, and mastering flows to verify success



### 4. Disaster-Recovery / High-Availability

- Back up ML 11/12 production; store in DR location
- Provision ML 11/12 DR cluster in secondary region / Availability Zone
- Restore config + data; establish database replication from prod → DR
- Run fail-over / fail-back exercise, update runbooks



### 5. Post-Upgrade Validation & Hardening

- Run automated regression suite; record new baseline response times
- Verify security roles, TLS certificates, and external auth still function
- Re-enable rebalancing, tiered-storage rules, and CPF/Task Server queues
- Reconnect BI/reporting or monitoring tools that rely on JMX or REST metrics
- Tune caches for new workload patterns
- Update internal documentation & topology diagrams



### 6. Close-out

- Send 'upgrade complete' memo to stakeholders
- Schedule a 30-day review to assess performance, bug backlog, and patch cadence
- Plan routine minor upgrades to stay current

# Need Help Upgrading with Confidence?

Dataavid helps organizations modernize their MarkLogic stack with minimal risk and zero disruption.

Learn more:

-  [MarkLogic Consulting](#)
-  [MarkLogic Managed Services](#)

[\*\*Book a free consultation\*\*](#)

