DOSSIER MODERNISATION · PLATEFORME DATA

MODERNISER UNE PLATEFORME DATA PENDANT QUE L'ENTREPRISE CONTINUE À TRAVAILLER

Publié le 10 novembre 2026 · Version 1.0.0 · Mise à jour le 10 novembre 2026
Article en ligne :
https://datanalysis.re/ressources/data-engineering/moderniser-plateforme-data-continuite-activite

Une nouvelle plateforme peut être prête techniquement alors que les applications et les équipes qui dépendent de ses données ne sont pas prêtes à changer. La date de livraison du nouvel environnement et la date de bascule des usages sont deux décisions différentes.

Pour préparer la seconde, il faut regarder ce qui produit les données, ce qui les transforme et ce qui les utilise. La méthode ci-dessous propose cinq décisions à documenter sur un premier périmètre. Elle ne prescrit ni une migration complète ni un outil particulier.

1. Partir des usages qui doivent continuer

Un jeu de données peut alimenter une application, un traitement de facturation, un export partenaire et un reporting. Ces consommateurs n'ont pas forcément les mêmes contraintes de délai, de format ou de disponibilité.

La cartographie doit donc relier applications, flux, règles et responsables. Elle doit également signaler les usages encore inconnus. Un export repris manuellement ou un traitement historique peu documenté peut ne pas apparaître dans les premières vues techniques.

Les recommandations AWS sur la préparation d'une bascule insistent sur les dépendances, les tests, les responsabilités et le déroulé de transition. [1] Pour notre cadrage, le résultat attendu est une liste exploitable : quel usage doit être préservé, qui le valide et quelle preuve montrera qu'il fonctionne après le changement ?

2. Choisir un périmètre cohérent

La première étape n'est pas nécessairement la plus petite table ou l'application la moins visible. Elle doit former un ensemble que l'équipe peut migrer et vérifier sans laisser des dépendances essentielles de côté.

Une évolution progressive peut limiter le périmètre de chaque bascule, mais elle ajoute une période de coexistence. Il faut alors maintenir les interfaces, comprendre quel système fait autorité et gérer les éventuelles différences entre les deux chemins. Une bascule groupée peut être nécessaire lorsque des composants sont fortement couplés. Le choix dépend de

l'architecture et des contraintes de l'activité. [2]

La décision utile n'est donc pas « tout migrer ou ne rien faire », mais « quels usages peuvent avancer ensemble, avec quels contrôles et quelles conditions d'arrêt ? »

3. Comparer à partir de règles et de données identifiées

Comparer deux totaux ne suffit pas si les périodes, les règles ou les données de départ diffèrent. Avant la recette, l'équipe doit fixer la référence temporelle, les filtres, les unités et les cas particuliers qui définissent le résultat attendu.

Un écart peut venir du nouveau traitement, d'une mise à jour de la source ou d'une règle métier qui a été modifiée pendant le projet. Sans référence explicite, l'équipe risque de traiter tous ces changements comme une seule anomalie.

Les contrôles doivent couvrir les résultats, mais aussi les délais de disponibilité, les droits et le comportement des applications consommatrices. Une donnée correcte dans la plateforme peut encore être lue dans la mauvaise unité ou arriver trop tard pour un usage.

4. Attribuer la décision de bascule

Le feu vert ne doit pas se résumer à « les traitements sont verts ». Il faut savoir quelles validations techniques et métier sont attendues, lesquelles sont obtenues et quelles réserves restent ouvertes.

Décisions de bascule et éléments à documenter

Décision	Élément à documenter
Périmètre	Applications, flux, usages et exclusions.
Référence	Jeux de données, périodes et règles servant à la comparaison.
Validation	Résultat attendu et responsable de chaque contrôle.
Feu vert	Personne habilitée à décider, critères et réserves acceptables.
Surveillance	Contrôles immédiats après bascule, durée d'observation et destinataires des alertes.
Arrêt ou retour	Déclencheurs, responsable et traitement des données déjà produites.

Le déroulé doit préciser l'ordre des opérations et les points où l'on peut interrompre le changement. Une répétition sur un environnement adapté permet de confronter le plan aux dépendances réelles ; elle ne transforme pas les hypothèses restantes en garanties.

5. Préparer le retour avec les nouvelles écritures

Revenir à l'ancien code ne suffit pas toujours. Des opérations ont pu être enregistrées après la bascule. Il faut décider comment elles seront conservées, rapprochées ou rejouées, sans perte ni doublon.

La stratégie doit également couvrir les sorties déjà diffusées vers les applications et partenaires. Une correction dans la plateforme ne retire pas automatiquement un fichier envoyé ou une décision prise à partir d'un résultat incorrect.

Les guides de bascule soulignent cette nécessité de prévoir le retour et la gestion des données modifiées. [1][2] Le projet doit identifier le point au-delà duquel un simple retour n'est plus adapté et une procédure de reprise ou de correction devient nécessaire.

Transformer la carte en première étape de travail

La carte de migration associée sert à préparer une réunion entre responsables applicatifs, équipe data, exploitation et métiers. Elle rassemble le périmètre, les preuves attendues, les validations et les conditions de retour.

Elle peut faire apparaître qu'un problème présenté comme une migration concerne d'abord une définition, un flux ou une dépendance mal maîtrisée. Traiter ce point peut devenir la première étape du projet, sans présumer de la trajectoire complète.

Examiner un périmètre de modernisation avec DATANALYSIS. Apportez les applications concernées, les usages à préserver et le changement envisagé. Nous pourrons cadrer les dépendances, les validations et une première étape cohérente.

→ Examiner mon périmètre de modernisation : https://datanalysis.re/contact?lang=fr

→ Carte remplissable (PDF) : https://datanalysis.re/downloads/data-modernisation-carte.fr.pdf

→ Carte HTML : https://datanalysis.re/ressources/data-engineering/moderniser-plateforme-data-continuite-activite/carte

Dans le même dossier

Quand une donnée change de sens : préparer l'évolution d'un flux entre applications : https://datanalysis.re/ressources/data-engineering/donnee-change-de-sens-evolution-flux

Reprendre un flux sans multiplier ses effets : https://datanalysis.re/ressources/data-engineering/reprendre-flux-sans-multiplier-effets

Expertises associées

Data Engineering : https://datanalysis.re/expertises/data-engineering?lang=fr

Data Governance : https://datanalysis.re/expertises/data-governance?lang=fr

Sources

Références consultées le 10 novembre 2026. La carte et les formulations sont une adaptation pédagogique de principes de préparation et de contrôle. Elles ne constituent pas un plan de migration exécutable sans examen de votre environnement.

[1] AWS Prescriptive Guidance | Migration cutover best practices: pre-cutover stage (préparation de la bascule : dépendances, tests, responsabilités, déroulé de transition et retour)
https://docs.aws.amazon.com/prescriptive-guidance/latest/best-practices-migration-cutover/pre-cutover-stage.html

[2] AWS Prescriptive Guidance | Migration cutover best practices: cutover stage (déroulé de la bascule, retour et gestion des données modifiées)
https://docs.aws.amazon.com/prescriptive-guidance/latest/best-practices-migration-cutover/cutover-stage.html

Exemple fictif à visée pédagogique ; aucun résultat client.