



AWS Base de référence en matière de résilience des startups

AWS Conseils prescriptifs



AWS Conseils prescriptifs: AWS Base de référence en matière de résilience des startups

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

Introduction	1
Champ d'application des directives	2
Modèle de responsabilité partagée	2
Étape 1 : Fixer des objectifs	4
Étape 2 : Conception et mise en œuvre	6
Étape 3 : Évaluer et tester	8
Étape 4 : opérer	10
Étape 5 : Réagir et apprendre	12
Étapes suivantes	14
Ressources	15
AWS Framework Well-Architected	15
AWS Directives prescriptives	15
Autres AWS ressources	15
Collaborateurs	16
Conception	16
Révision	16
Rédaction technique	16
Historique de la documentation	17
.....	xviii

AWS Base de référence en matière de résilience des startups

Amazon Web Services ([contributeurs](#))

Mars 2026 ([historique du document](#))

Chaque fondateur de start-up connaît la tension : l'équipe s'empresse de proposer de nouvelles fonctionnalités alors que les clients s'attendent à ce que chaque version fonctionne. Vous recevez une notification concernant la panne survenue la nuit dernière, les plaintes des clients affluent et votre feuille de route contient déjà de nombreuses fonctionnalités promises. C'est la réalité quotidienne des startups : trouver un équilibre entre la pression d'innovation et la nécessité de maintenir des services stables.

L'instinct de prioriser les nouvelles fonctionnalités est naturel. Les investisseurs veulent de la croissance, les clients demandent davantage de capacités et les concurrents ne restent pas immobiles. Pourtant, l'expérience montre que les problèmes de résilience peuvent faire dérailler même les startups les plus innovantes. Une interruption de service ne signifie pas seulement une perte de revenus ; elle érode la confiance que vous avez tant travaillé pour établir avec vos clients. En fait, une fiabilité à toute épreuve peut devenir votre facteur de différenciation silencieux sur un marché surpeuplé.

Renforcer la résilience de votre start-up ne signifie pas ralentir ou investir massivement dans l'infrastructure. Il s'agit de faire des choix intelligents dès le début afin de prévenir les problèmes plus tard. Pensez-y comme si vous construisiez une maison. Il est beaucoup plus facile de poser des bases solides dès le départ que de régler les problèmes structurels une fois que tout est construit. En intégrant des pratiques résilientes à votre architecture et à vos opérations initiales, vous créez un avantage concurrentiel. Vous ne vous contentez pas de livrer un produit ; vous offrez une expérience fiable qui fidélise les clients.

La AWS Startup Resiliency Baseline (AWS SRB) a été créée spécifiquement pour les équipes confrontées à ce défi. Il fournit une solution pratique pour renforcer la fiabilité de vos systèmes tout en maintenant la vitesse qui fait la particularité des startups. Pas de processus lourds ni de complexité à l'échelle de l'entreprise, juste des modèles pratiques qui évoluent avec votre entreprise.

Ce guide explique comment implémenter le AWS SRB dans un environnement de démarrage. Il vous aide à identifier ce qui compte vraiment pour la résilience, où concentrer vos ressources limitées et

comment mettre en place des pratiques qui évoluent au fur et à mesure de la croissance de votre entreprise.

Champ d'application des directives

Lorsque vous étudiez les étapes pratiques du renforcement de la résilience des startups, il est important de comprendre dans quelle mesure ces conseils s'inscrivent dans la trajectoire de croissance globale de votre organisation. La base de référence de résilience des AWS startups est alignée sur le [cadre du cycle de vie de résilience](#). Il constitue une feuille de route initiale, conçue pour vous aider à mettre en œuvre des mesures de résilience fondamentales avec un minimum de frais de développement lors du déploiement de vos premières applications AWS. Considérez-le comme votre kit de départ pour créer des systèmes fiables capables de se remettre efficacement en cas de panne.

Le [AWS Well-Architected](#) Framework et ce guide de résilience se complètent pour vous aider à créer une infrastructure sécurisée et fiable. Ce guide approfondit les pratiques de résilience, tandis que le AWS Well-Architected Framework fournit un ensemble plus large de meilleures pratiques architecturales. Vous pouvez évaluer votre architecture par rapport à ces pratiques en utilisant le [AWS Well-Architected Tool](#) in your Compte AWS.

Modèle de responsabilité partagée

Il est essentiel de clarifier un aspect important de la création de systèmes résilients dans le cloud : le partenariat entre AWS et votre start-up. La résilience dans le cloud fonctionne selon un [modèle de responsabilité partagée](#), chaque équipe jouant un rôle distinct pour garantir la résilience du système.

AWS

AWS assume la responsabilité de la résilience de l'infrastructure cloud sous-jacente qui alimente ses services. Considérez cela comme la base sur laquelle vous construisez vos applications. Cela inclut le maintien de la résilience des centres de données, des composants réseau et du cœur Services AWS que vous utilisez dans votre architecture.

La responsabilité de votre start-up se concentre sur la résilience des systèmes que vous créez et déployez sur cette base. Les recommandations de ce guide relèvent de votre domaine de responsabilité. En mettant en œuvre ces pratiques de manière réfléchie, vous pouvez créer des applications qui utilisent une AWS infrastructure fiable et intègrent des fonctionnalités de restauration robustes.

Ce modèle signifie que vous ne développez jamais la résilience de manière isolée. Vous vous appuyez sur une base fiable et éprouvée tout en concentrant vos efforts sur les aspects qui ont

un impact direct sur vos clients et vos activités commerciales. Ce guide explique comment tirer le meilleur parti de ce modèle de responsabilité partagée, en utilisant une AWS infrastructure fiable tout en mettant en œuvre des mesures pratiques pour garantir que vos applications se rétablissent harmonieusement après les interruptions.

Étape 1 : Fixer des objectifs

Imaginez que votre équipe effectue le sprint final avant le lancement d'un produit majeur. Les nouvelles fonctionnalités sont révolutionnaires et l'enthousiasme des investisseurs ne cesse de croître. Ensuite, lors d'un déploiement de routine, votre service principal tombe en panne. Au fur et à mesure que les plaintes des clients envahissent vos e-mails, deux questions deviennent extrêmement claires : combien de temps pouvez-vous vous permettre de rester hors ligne ? Quelles données pouvez-vous vous permettre de perdre ?

Espérer que tout va bien fonctionner n'est pas une bonne stratégie. Vous avez besoin d'une méthode systématique pour décider où la résilience est la plus importante et où elle ne l'est pas. C'est là qu'une analyse d'impact commercial (BIA) devient essentielle. Il vous aide à prendre des décisions éclairées sur les domaines dans lesquels investir dans la résilience. Un BIA vous aide à comprendre quelles parties de votre système ont réellement besoin d'une fiabilité à toute épreuve et lesquelles peuvent tolérer une certaine flexibilité.

Commencez par cartographier les principaux parcours de vos utilisateurs. Pour chacune d'entre elles, posez-vous les questions suivantes :

- Quel est l'impact si cela est perturbé ?
- Dans combien de temps devons-nous rétablir le service ?
- Quelles sont les données qu'il est essentiel de protéger ?

Il ne s'agit pas simplement d'un exercice technique ; il vous aide à comprendre l'impact commercial des problèmes de fiabilité. La perte de revenus n'est qu'un début. Réfléchissez à la façon dont les pannes peuvent éroder la confiance des clients, enfreindre les exigences réglementaires ou donner un avantage aux concurrents.

À partir de cette analyse, vous allez déduire deux chiffres critiques pour chaque parcours utilisateur : l'objectif de temps de restauration (RTO) et l'objectif de point de reprise (RPO). Le RTO définit la rapidité avec laquelle vous devez rétablir ce parcours. Le RPO définit le niveau de perte de données que vos clients peuvent tolérer. Ces objectifs commerciaux orientent ensuite les composants que vous choisissez et la manière dont vous les concevez, sans surdimensionner chaque partie du système.

L'avantage de cette approche est qu'elle vous permet de concentrer vos ressources limitées là où elles comptent le plus. Le traitement de vos transactions de base nécessite peut-être une restauration

quasi instantanée et aucune perte de données, mais votre moteur de recommandation peut tolérer des temps d'arrêt plus longs. En définissant des objectifs clairs, vous créez un cadre qui vous permet de poursuivre le développement rapide de fonctionnalités tout en renforçant la résilience de manière stratégique.

Documentez clairement ces objectifs. Ils ne sont pas réservés à votre équipe d'ingénierie. Lorsque vous vous adressez à des entreprises clientes ou que vous soumettez un audit technique à des investisseurs, cette documentation montre que vous avez fait preuve d'un esprit critique en matière de continuité des activités.

Ces cibles évoluent au fur et à mesure que votre start-up grandit. Les besoins en résilience de vos mille premiers utilisateurs sont différents de ceux de votre première entreprise cliente. Commencez par des objectifs que vous pouvez atteindre de façon réaliste dès aujourd'hui, mais planifiez la manière dont ils se resserreront à mesure que vous évoluerez.

Ce guide explique comment mettre en œuvre des mesures de résilience qui répondent à ces objectifs. La définition de ces objectifs est votre première étape cruciale. Ils sont votre boussole pour surmonter la tension constante entre innovation et stabilité, en vous aidant à créer un système qui apporte de la valeur à vos clients de manière fiable.

Étape 2 : Conception et mise en œuvre

Cette section explique comment transformer vos objectifs de résilience en réalité. Vous avez défini ce qui compte le plus pour votre entreprise, et il est maintenant temps de le développer. Comment renforcer la résilience sans ralentir l'innovation ?

Considérez les services AWS gérés comme un raccourci vers la résilience. Au lieu de perdre de précieuses heures d'ingénierie à entretenir l'infrastructure, utilisez des services qui s'occupent de la redondance à votre place. Prenons l'exemple d'[Amazon Simple Storage Service \(Amazon S3\)](#). Il stocke automatiquement plusieurs copies de vos données dans une Région AWS soucieuse de durabilité. Il ne nécessite pas de code supplémentaire ni de droits de téléavertisseur tard le soir.

Qu'en est-il des principaux composants de votre application ? Des choix intelligents peuvent accroître l'impact de votre équipe. Pensez à une base de données qui constitue l'épine dorsale de votre service. Au lieu de créer votre propre système de réplication, pensez à utiliser [Amazon Aurora](#), qui gère automatiquement le basculement. Ces fonctionnalités peuvent coûter plus cher, mais elles permettent à votre équipe de se concentrer sur la résolution des problèmes commerciaux plutôt que sur la maintenance de l'infrastructure. Ce coût peut être compensé par une diffusion plus rapide des fonctionnalités et par des pertes de revenus évitées en cas de panne.

Parfois, les startups ont besoin de créer des solutions personnalisées. C'est la nature des startups innovantes. Lorsque vous le faites, restez simple mais intelligent. Répartissez votre application sur plusieurs zones de disponibilité à l'aide des [groupes Elastic Load Balancing](#) et [Amazon EC2 Auto Scaling](#). Définissez la capacité minimale du groupe Auto Scaling pour gérer votre trafic de base même en cas de défaillance d'une zone de disponibilité. Cela garantit la résilience face aux défaillances localisées sans modèles architecturaux complexes. À mesure que votre start-up se développe et que les clients exigent une plus grande résilience, vous pouvez adopter des approches plus sophistiquées.

Nous vous recommandons de séparer vos environnements de production et de développement Comptes AWS. Il est tentant de les mélanger lorsque vous vous déplacez rapidement, mais cette limite constitue votre filet de sécurité. Cela empêche qu'une expérience bien intentionnée ne mette fin à votre service de production. Considérez-le comme une assurance pour votre culture de développement qui consiste à « agir vite et à innover », c'est-à-dire à innover dans le développement, à maintenir la stabilité de la production.

Si votre application dépend de services tiers, planifiez leurs défaillances. Lorsque votre processeur de paiement rencontre des problèmes, votre système peut-il les gérer correctement ? Construisez

des disjoncteurs simples et des options de secours. Vous pouvez mettre ces transactions en file d'attente au lieu d'afficher des messages d'erreur. Vos clients apprécieront que vous ayez fait en sorte que les choses fonctionnent, même si ce n'est pas parfaitement le cas.

Documentez au fur et à mesure que vous créez, mais restez pratique. Concentrez-vous sur l'enregistrement des raisons qui sous-tendent les décisions clés et créez des plans de rétablissement simples. Il est important de les avoir à portée de main en cas d'incident.

Vous ne bâtissez pas pour une résilience parfaite ; vous bâtissez pour une résilience appropriée. Chaque heure consacrée à l'ingénierie excessive de la résilience est une heure non consacrée aux fonctionnalités demandées par les clients. Utilisez les services AWS gérés comme base, ajoutez une résilience ciblée là où c'est le plus important et créez des moyens clairs pour renforcer la résilience au fur et à mesure que votre entreprise se développe.

Le chapitre suivant explique comment valider ces choix de conception sans épuiser les ressources d'ingénierie. Pour les startups, les tests doivent être une solution raisonnable et un investissement intelligent dans la résilience de votre application.

Étape 3 : Évaluer et tester

Vous avez construit une base résiliente, mais comment savez-vous qu'elle fonctionne réellement ? Tester la résilience peut sembler un luxe lorsque vous vous efforcez de prouver l'adéquation du produit au marché. Cependant, il existe un moyen intelligent de le faire sans faire dérailler le développement de vos fonctionnalités. Ce chapitre décrit les tests pratiques et allégés qui s'adaptent au rythme d'une start-up.

Commencez par [AWS Resilience Hub](#), et considérez-le comme un outil d'évaluation initiale de l'architecture. Il fournit un examen de base utile des fondements de résilience d'une architecture. Il vous aide à évaluer si la configuration de base de l'infrastructure correspond à vos objectifs de restauration en vérifiant les modèles de configuration courants et les points de défaillance uniques potentiels. Il peut signaler des lacunes évidentes, telles que l'absence de plusieurs configurations de zone de disponibilité ou des politiques de sauvegarde incomplètes. Resilience Hub complète, mais ne remplace pas, les révisions d'architecture réfléchies et les tests ciblés des chemins critiques.

Pour valider vos objectifs de restauration documentés, planifiez des tests de restauration mensuels [AWS Backup](#) dans votre environnement de développement. Même si cela demande du temps d'ingénierie, cela peut s'avérer moins coûteux que de découvrir que vos sauvegardes ne fonctionnent pas lors d'un véritable incident. Intégrez-le à votre cycle de développement habituel, par exemple en exécutant des tests unitaires ou des révisions de code. L'objectif n'est pas la perfection ; c'est la certitude que vous pouvez récupérer quand vous en avez besoin.

Au fur et à mesure que votre start-up grandit et que les clients commencent à dépendre de plus en plus de vous, améliorez progressivement votre jeu de test. Lorsque vous déployez de nouvelles fonctionnalités, incluez des contrôles de résilience de base dans votre pipeline. Essayez des expériences de chaos simples en utilisant [AWS Fault Injection Service](#). Commencez dans votre environnement de préproduction et commencez modestement. Testez la façon dont votre application gère une réponse API différée pendant le développement avant d'envisager des expériences de production. Au fur et à mesure que votre confiance augmente, étendez progressivement ces tests, mais validez toujours d'abord en préproduction. Pour une start-up, casser des objets en production est suffisamment risqué sans le faire intentionnellement.

La clé, c'est l'équilibre. Chaque heure consacrée aux tests est une heure non consacrée à la création de nouvelles fonctionnalités. Cependant, quelques tests stratégiques peuvent prévenir les types de pannes susceptibles de faire perdre la confiance des clients. Utilisez les outils automatisés fournis AWS pour effectuer le gros du travail et concentrez-vous sur les tests les plus importants pour vos

clients. Cela vous permet de renforcer la confiance dans la résilience de votre application sans ralentir l'innovation.

Le chapitre suivant explique comment faire évoluer cette base au fur et à mesure que votre start-up évolue.

Étape 4 : opérer

Vous avez créé une application résiliente et l'avez testée. Aujourd'hui, la réalité quotidienne est de le faire fonctionner. Mais dans une start-up, vous ne pouvez pas surveiller toutes les opérations, et vous ne devriez pas essayer de le faire. L'essentiel est de rester attentif à ce qui compte, sans fournir trop de statistiques ni surcharger votre équipe.

Commencez par le point de vue du client. Les canaris [Amazon CloudWatch Synthetics](#) agissent comme des clients automatisés. Ils testent en permanence les parcours critiques des utilisateurs. Demandez-leur de se connecter, de simuler des achats à l'aide de comptes de test ou d'accéder à des fonctionnalités clés, en particulier pendant les heures les plus chargées. Cela vous permet de comprendre l'expérience client et de détecter les problèmes avant que les vrais utilisateurs ne le fassent. Lorsqu'un canari échoue, vous savez immédiatement que quelque chose ne va pas du point de vue du client.

Tirez parti de cette base en surveillant de manière ciblée l'infrastructure de soutien. Quels signaux vous indiquent qu'il y a un problème ? [Amazon](#) vous CloudWatch aide à créer des tableaux de bord qui permettent de suivre ces signes. Ne vous contentez pas de surveiller les indicateurs techniques, associez-les à l'impact commercial. Par exemple, une utilisation élevée du processeur est importante, mais c'est parce qu'elle peut dégrader l'expérience client que vous suivez avec des canaris.

Concrètement, associez votre surveillance aux parcours de vos clients. Si vous utilisez une plateforme SaaS (Software as a Service), vous vous souciez probablement des temps de réponse des API, des taux de réussite de l'authentification et de la disponibilité des fonctionnalités de base. Configurez des alertes qui vous avertissent lorsque ces indicateurs dérivent. Soyez toutefois sélectif. Chaque alerte doit exiger une action. Si votre équipe commence à ignorer les alertes parce que « ce n'est probablement rien », c'est que vous en avez défini trop ou que vous suivez les mauvais indicateurs.

Transférez ces alertes par le biais d'outils que votre équipe utilise déjà. Si vos ingénieurs utilisent une application de messagerie en particulier, envoyez-y des alertes. L'objectif est une prise de conscience rapide sans créer de nouveau processus. Lorsqu'une alerte se déclenche, votre équipe doit savoir exactement ce que cela signifie et comment y remédier.

Veillez à ce que votre documentation opérationnelle soit simple et pratique. Stockez des runbooks contenant votre code dans le contrôle de version, mais n'oubliez pas qu'il ne s'agit pas de romans. En cas de panne, votre équipe a besoin de mesures claires et concrètes. Chaque alerte doit renvoyer à un runbook correspondant, et chaque runbook doit répondre à trois questions :

- Qu'est-ce qui s'est cassé ?
- Pourquoi est-ce important ?
- Comment puis-je résoudre ce problème ?

Mettez en œuvre un processus simple de gestion des incidents. Vous n'avez pas besoin de cadres complexes, mais simplement de définitions claires de ce qui constitue un incident et des personnes à contacter en cas d'escalade. Conservez les journaux d'incidents, car ils vous aident à améliorer la résilience de votre application.

L'essentiel est de trouver le juste équilibre entre vigilance et frais généraux. Utilisez AWS des outils pour automatiser ce que vous pouvez, concentrez-vous sur le suivi des indicateurs qui ont un impact sur les clients et maintenez vos processus suffisamment légers pour évoluer au fur et à mesure de votre croissance.

Le chapitre suivant explore comment favoriser un état d'esprit de résilience sans pour autant sacrifier la rapidité et l'innovation qui font la particularité des startups. En fin de compte, la résilience concerne autant les personnes que la technologie.

Étape 5 : Réagir et apprendre

Lorsque vous dirigez une start-up, des processus post-mortem complexes peuvent ralentir votre équipe. Ce chapitre explique comment tirer des leçons des incidents sans les transformer en exercices bureaucratiques.

Intégrez l'apprentissage par incidents à vos rythmes existants. Si votre équipe se réunit déjà régulièrement, consacrez dix minutes à discuter des incidents récents. Concentrez-vous sur des questions pratiques, telles que :

- Les runbooks vous ont-ils aidé ?
- Les alertes ont-elles eu lieu au bon moment ?
- Les services AWS gérés auraient-ils pu empêcher cela ?

Concentrez-vous sur les actions, pas sur le blâme. Dans une start-up, vous ne construisez pas un système parfait ; vous en créez un qui s'améliore chaque fois que quelque chose ne va pas.

Vous pouvez utiliser votre système de billetterie pour suivre les incidents ; aucun outil spécialisé n'est nécessaire. Créez un modèle simple qui inclut la chronologie des incidents, l'impact sur le client, les mesures de reprise prises et les leçons apprises. Cela peut devenir une mémoire institutionnelle si vous l'utilisez activement. Passez en revue les incidents passés lors de l'intégration afin de mettre les nouveaux ingénieurs au courant. Référez-les dans les revues d'architecture lors de la conception de systèmes similaires. Intégrez-les aux journées de jeu pour créer des scénarios d'échec réalistes basés sur des événements réels. Le modèle capture ce qui s'est passé, et une utilisation régulière le transforme en apprentissage organisationnel.

Au fur et à mesure que les startups se développent, des modèles apparaissent. Il se peut que certains composants échouent plus souvent ou que certains types de modifications posent des problèmes. Utilisez ces modèles pour orienter les investissements dans la résilience. Si les basculements de base de données posent problème, envisagez d'améliorer la configuration de vos zones de disponibilité multiples. Si les interruptions des services fournis par des tiers sont un thème courant, envisagez d'améliorer les disjoncteurs.

L'objectif n'est pas d'empêcher tous les échecs possibles. C'est impossible et cela vous ralentirait trop. L'objectif est d'apprendre rapidement, de s'adapter rapidement et de maintenir la fiabilité de l'application pendant que vous vous développez rapidement. Profitez de chaque incident pour rendre votre système un peu plus résilient, votre équipe un peu plus compétente et vos clients un peu

plus confiants dans votre service. Pour les entreprises en démarrage, la rapidité et l'apprentissage surpassent la perfection. Créez des processus légers qui vous aident à tirer les leçons des incidents sans ralentir l'innovation. Les meilleures pratiques de résilience sont celles que votre équipe utilise réellement.

Prochaines étapes pour les startups

Vous vous souvenez de ce déploiement nocturne qui a fait battre votre cœur ? Ou le moment où un client important vous a posé des questions sur vos mesures de résilience ? Le parcours de résilience d'une start-up ne consiste pas à éliminer ces moments, mais à les affronter en toute confiance.

Ce guide décrit une voie pratique vers la résilience, notamment : définir des objectifs de reprise réalistes, utiliser des services AWS gérés, mettre en œuvre des tests allégés, surveiller ce qui compte et tirer les leçons des incidents. Cette approche est puissante pour les startups, car elle évolue avec vous. Le même cadre qui aide votre start-up à se remettre des problèmes d'aujourd'hui constitue la base des services qu'elle pourra offrir aux entreprises clientes de demain.

Pensez à la résilience comme à une feuille de route pour un produit ; vous ne créez pas tout en même temps. Commencez par protéger vos principaux services. Ajoutez une surveillance pour les parcours clients critiques. Mettez en œuvre des tests de base qui détectent les problèmes évidents. Documentez ce que vous apprenez. Chaque étape est gérable et pratique, et surtout, elle ne vous empêche pas d'utiliser les fonctionnalités d'expédition.

Au fur et à mesure qu'une start-up grandit, ses besoins en matière de résilience évoluent. Les entreprises clientes peuvent poser des questions plus difficiles. Les pics de trafic peuvent atteindre de nouveaux sommets. L'expansion internationale pose de nouveaux défis. Parce que vous avez intégré la résilience à votre base, vous êtes prêt à vous adapter.

L'objectif des startups n'est pas une disponibilité parfaite ou l'absence d'incidents. L'objectif est de créer un système suffisamment fiable pour gagner la confiance des clients tout en maintenant la vitesse qui rend les startups spéciales. En suivant cette approche, vous pouvez créer quelque chose de plus précieux que la création d'un système parfait ; vous pouvez en créer un qui s'améliore chaque jour en tirant des leçons de chaque défi tout en suivant le rythme des ambitions de votre start-up.

Le parcours de résilience ne s'arrête pas. Il évolue en fonction de chaque fonctionnalité que vous proposez, de chaque client que vous gagnez et de chaque incident que vous gérez. Ce guide vous aide à créer un cadre qui vous aidera à avancer de manière sûre et pratique. En fin de compte, le succès d'une start-up n'est pas une question de choix entre résilience et rapidité. Il s'agit de trouver des moyens intelligents de fournir les deux.

Ressources

AWS Framework Well-Architected

- [Reliability Pillar](#)
 - [REL05- BP01 Implémentez une dégradation progressive pour transformer les dépendances matérielles applicables en dépendances souples](#)
 - [REL06- BP01 Surveillez tous les composants en fonction de la charge de travail](#)
 - [REL06- BP03 Envoyer des notifications \(traitement en temps réel et alarmes\)](#)
 - [REL08- BP03 Intégrez des tests de résilience dans le cadre de votre déploiement](#)
- [Modèle de responsabilité partagée pour la résilience](#)

AWS Directives prescriptives

- [Approches de sauvegarde et de restauration sur AWS](#)
- [Cadre du cycle de vie de la résilience : une approche continue de l'amélioration de la résilience](#)

Autres AWS ressources

- [AWS Limites d'isolation des défauts](#) (AWS livres blancs)
- [Définition d'objectifs RPO et RTO pour les applications cloud](#) (article de AWS blog)
- [Présentation des options de déploiement sur AWS](#) (livres AWS blancs)

Collaborateurs

Les personnes suivantes ont contribué à ce guide.

Conception

- Dylan McCarroll, architecte de solutions associé, AWS
- Igor Fil, architecte de solutions de démarrage, AWS
- Parth Shah, architecte de solutions senior, AWS
- Vrajesh Prajapati, architecte de solutions, AWS

Révision

- Clark Richey, technologue principal, AWS
- Bruno Emer, technologue principal, AWS

Rédaction technique

- Lilly AbouHarb, rédactrice technique senior, AWS

Historique du document

Le tableau suivant décrit les modifications importantes apportées à ce guide. Pour être averti des mises à jour à venir, abonnez-vous à un [fil RSS](#).

Modification	Description	Date
Mises à jour importantes	Recommandations mises à jour et utilisation de l' Services AWS ensemble.	6 mars 2026
Publication initiale	—	24 janvier 2024

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.