

AWS Livre blanc

Principes fondamentaux de l'architecture SaaS



Principes fondamentaux de l'architecture SaaS: AWS Livre blanc

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

Résumé et introduction	i
Introduction	1
Êtes-vous bien architecturé ?	2
Le SaaS est un modèle économique	3
Vous êtes un service, pas un produit	5
Version initiale	6
Passer à une expérience unifiée	9
Plan de contrôle ou plan d'application	12
Principales services	14
Redéfinir la mutualisation	16
Le cas extrême	18
Suppression du terme de locataire unique	20
Présentation du silo et du pool	20
Silo et pool complets	22
SaaS ou fournisseur de services gérés (MSP)	24
Migration vers le SaaS	26
identité SaaS	30
Isolement des locataires	31
Partitionnement de données	33
Comptage, mesures et facturation	35
SaaS B2B et B2C	37
Conclusion	38
Suggestions de lecture	39
Collaborateurs	40
Révisions du document	41
Avis	42
Glossaire AWS	43
.....	xliv

Principes fondamentaux de l'architecture SaaS

Date de publication : 3 août 2022 ([Révisions du document](#))

La portée, les objectifs et la nature de la gestion d'une entreprise dans un modèle de logiciel en tant que service (SaaS) peuvent être difficiles à définir. La terminologie et les modèles utilisés pour caractériser les SaaS varient en fonction de leur origine. L'objectif de ce document est de mieux définir les éléments fondamentaux du SaaS et de créer une image plus claire des modèles, des termes et des systèmes de valeurs appliqués lors de la conception et de la fourniture d'un système SaaS sur AWS. L'objectif général est de fournir un ensemble d'informations de base qui fournissent aux clients une vision plus claire des options qu'ils devraient envisager lorsqu'ils envisagent d'adopter un modèle de prestation SaaS.

Ce document s'adresse aux concepteurs et aux architectes du SaaS qui en sont au début de leur parcours vers le SaaS, ainsi qu'aux constructeurs plus expérimentés qui souhaitent affiner leur compréhension des concepts SaaS de base. Certaines de ces informations peuvent également être utiles aux propriétaires de produits SaaS et aux stratèges qui souhaitent se familiariser avec le paysage SaaS.

Introduction

Le terme logiciel en tant que service (SaaS) est utilisé pour décrire un modèle commercial et de prestation. Le défi, cependant, est que ce que signifie être SaaS n'est pas universellement compris.

Bien qu'il existe un certain consensus sur certains des piliers fondamentaux du SaaS, une certaine confusion persiste quant à ce que signifie être SaaS. Il est normal qu'il y ait des variations dans la façon dont les équipes perçoivent le SaaS. Dans le même temps, le manque de clarté des concepts et des termes du SaaS peut créer une certaine confusion pour ceux qui explorent un modèle de prestation SaaS.

Ce document se concentre sur la description de la terminologie utilisée pour décrire les concepts de base du SaaS. Le fait d'avoir un état d'esprit commun autour de ces concepts permet d'avoir une image claire des éléments fondamentaux d'une architecture SaaS, ce qui vous permet de disposer d'un vocabulaire commun pour décrire les constructions de l'architecture SaaS. Cela est particulièrement utile lorsque vous recherchez du contenu supplémentaire basé sur ces thèmes.

Ce livre blanc revient sur les détails de l'architecture du multitenant et explore la manière dont nous avons défini les principes fondamentaux du SaaS. Dans l'idéal, cela fournira également un ensemble

de terminologie plus clair qui permettra aux entreprises de s'aligner plus rapidement sur la nature et la nature de leurs solutions SaaS.

Êtes-vous bien architecturé ?

Le [AWS Well-Architected Framework](#) vous aide à comprendre les avantages et les inconvénients des décisions que vous prenez lors de la création de systèmes dans le cloud. Les six piliers du Framework vous permettent d'apprendre les meilleures pratiques architecturales pour concevoir et exploiter des systèmes fiables, sécurisés, efficaces, rentables et durables. À l'aide du [AWS Well-Architected Tool](#), disponible gratuitement dans le [AWS Management Console](#), vous pouvez examiner vos charges de travail par rapport à ces meilleures pratiques en répondant à une série de questions pour chaque pilier.

Du point de [vue du SaaS](#), nous nous concentrons sur les meilleures pratiques pour l'architecture de vos charges de travail SaaS (logiciel en tant que service). AWS

[Pour obtenir des conseils d'experts supplémentaires et les meilleures pratiques concernant votre architecture cloud \(déploiements d'architecture de référence, diagrammes et livres blancs\), consultez le Centre d'architecture. AWS](#)

Le SaaS est un modèle économique

Pour définir ce que signifie être SaaS, il faut d'abord s'entendre sur un principe clé : le SaaS est un modèle commercial. Cela signifie que, avant tout, l'adoption d'un modèle de prestation SaaS est directement motivée par un ensemble d'objectifs commerciaux. Oui, la technologie sera utilisée pour atteindre certains de ces objectifs, mais le SaaS consiste à mettre en place un état d'esprit et un modèle qui ciblent un ensemble spécifique d'objectifs commerciaux.

Examinons de plus près certains des principaux objectifs commerciaux associés à l'adoption d'un modèle de prestation SaaS.

- **Agilité** — Ce terme résume l'objectif plus large du SaaS. Les entreprises SaaS prospères reposent sur l'idée qu'elles doivent être prêtes à s'adapter continuellement à la dynamique du marché, des clients et de la concurrence. Les grandes entreprises SaaS sont structurées de manière à adopter en permanence de nouveaux modèles de tarification, de nouveaux segments de marché et de nouveaux besoins des clients.
- **Efficacité opérationnelle** — Les entreprises SaaS s'appuient sur l'efficacité opérationnelle pour promouvoir l'évolutivité et l'agilité. Cela implique de mettre en place une culture et des outils axés sur la création d'une empreinte opérationnelle qui favorise la publication fréquente et rapide de nouvelles fonctionnalités. Cela signifie également disposer d'une expérience unique et unifiée qui vous permet de gérer, d'exploiter et de déployer tous les environnements clients de manière collective. L'idée de prendre en charge des versions et des personnalisations uniques est révolue. Les entreprises SaaS accordent une grande importance à l'efficacité opérationnelle en tant que pilier essentiel de leur capacité à développer et à faire évoluer leur activité avec succès.
- **Intégration fluide** : pour gagner en agilité et faire face à la croissance, vous devez également accorder la priorité à la réduction des frictions dans le processus d'intégration des clients locataires. Cela s'applique universellement aux clients interentreprises (B2B) et business-to-customer (B2C). Quel que soit le segment ou le type de client que vous soutenez, vous devez toujours vous concentrer sur le délai de rentabilisation pour vos clients. Le passage à un modèle centré sur les services exige que l'activité SaaS se concentre sur tous les aspects de l'expérience client, en mettant particulièrement l'accent sur la répétabilité et l'efficacité du cycle de vie global d'intégration.
- **Innovation** — La transition vers le SaaS ne consiste pas uniquement à répondre aux besoins des clients actuels ; il s'agit également de mettre en place les éléments fondamentaux qui vous permettent d'innover. Vous souhaitez réagir et répondre aux besoins des clients dans votre modèle SaaS. Toutefois, vous souhaitez également utiliser cette agilité pour stimuler les innovations future

qui vous permettront de débloquer de nouveaux marchés, de nouvelles opportunités et de gagner en efficacité pour vos clients.

- Réaction du marché — Le SaaS s'éloigne de la notion traditionnelle de versions trimestrielles et de plans biennaux. Elle s'appuie sur son agilité pour donner à l'organisation la capacité de réagir et de répondre à la dynamique du marché en temps quasi réel. L'investissement dans les éléments organisationnels, techniques et culturels du SaaS permet de faire pivoter la stratégie commerciale en fonction des clients émergents et de la dynamique du marché.
- Croissance — Le SaaS est une stratégie commerciale centrée sur la croissance. L'alignement de tous les éléments mobiles de l'organisation autour de l'agilité et de l'efficacité permet aux organisations SaaS de cibler un modèle de croissance. Cela implique de mettre en place les mécanismes qui favorisent l'adoption rapide de votre offre SaaS.

Vous remarquerez que chacun de ces éléments est axé sur un résultat commercial. Il existe un large éventail de stratégies et de modèles techniques qui peuvent être utilisés pour créer un système SaaS. Cependant, rien dans ces stratégies techniques ne change l'histoire globale de l'entreprise.

Lorsque nous discutons avec des organisations et que nous leur demandons ce qu'elles essaient de réaliser dans le cadre de leur adoption du SaaS, nous commençons toujours par cette discussion axée sur l'entreprise. Les choix technologiques sont importants, mais ils doivent être réalisés dans le contexte de ces objectifs commerciaux. Le fait d'être mutualisé sans gagner en agilité, en efficacité opérationnelle ou sans intégration fluide, par exemple, compromettrait le succès de votre activité SaaS.

Dans ce contexte, essayons de formaliser cela en une définition plus concise du SaaS conforme aux principes décrits précédemment :

Le SaaS est un modèle commercial et de fourniture de logiciels qui permet aux entreprises de proposer leurs solutions selon un modèle axé sur les services et à faible friction qui maximise la valeur pour les clients et les fournisseurs. Elle repose sur l'agilité et l'efficacité opérationnelle en tant que piliers d'une stratégie commerciale qui favorise la croissance, la portée et l'innovation.

Vous devriez voir l'alignement entre les objectifs commerciaux et la façon dont ils reposent sur une expérience partagée pour tous les clients. Le passage au SaaS implique en grande partie de s'éloigner des personnalisations ponctuelles qui peuvent faire partie d'un modèle logiciel traditionnel. Tout effort visant à proposer une spécialisation aux clients nous éloigne généralement des valeurs fondamentales que nous essayons d'atteindre avec le SaaS.

Vous êtes un service, pas un produit

L'adoption d'un modèle de « service » ne se limite pas au marketing ou à la terminologie. Dans un esprit de service, vous vous éloignerez des aspects de l'approche traditionnelle du développement basée sur les produits. Bien que les caractéristiques et les fonctionnalités soient certainement importantes pour chaque produit, le SaaS met davantage l'accent sur l'expérience que les clients pourront vivre avec votre service.

Qu'est-ce que cela signifie ? Dans un modèle centré sur les services, vous réfléchissez davantage à la manière dont les clients sont intégrés à votre service, à la rapidité avec laquelle ils obtiennent de la valeur et à la rapidité avec laquelle vous pouvez introduire des fonctionnalités qui répondent à leurs besoins. Les détails associés à la manière dont votre service est conçu, exploité et géré échappent à la vue de votre client.

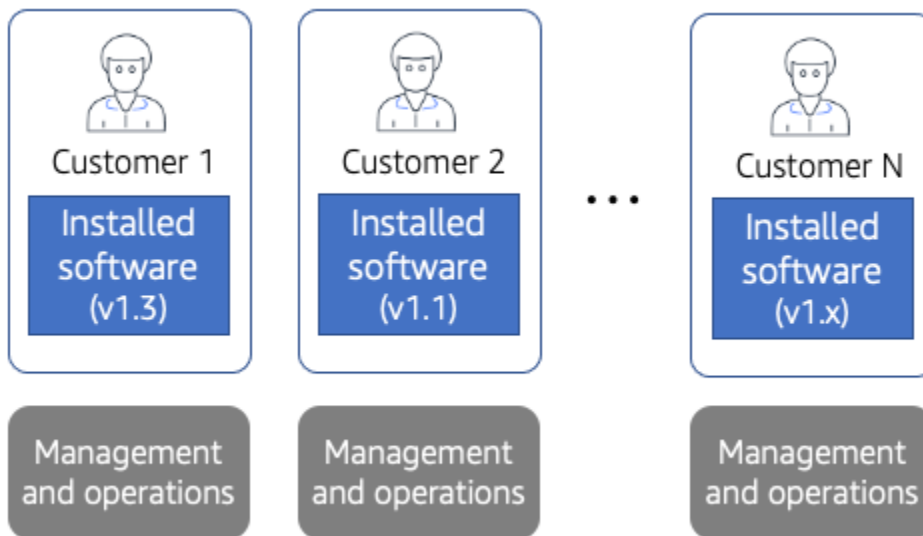
Dans ce mode, nous pensons à ce service SaaS comme nous le ferions pour n'importe quel autre service que nous pourrions consommer. Si nous sommes dans un restaurant, nous nous soucions certainement de la nourriture, mais aussi du service. La rapidité avec laquelle votre serveur arrive à votre table, la fréquence à laquelle il fait le plein d'eau, la rapidité avec laquelle les plats arrivent : ce sont autant de mesures de l'expérience de service. C'est le même état d'esprit et le même système de valeurs qui devraient façonner la façon dont nous envisageons de créer un service SaaS.

Ce as-a-service modèle devrait avoir une forte influence sur la façon dont vous constituez vos équipes et votre service. Votre carnet de travail placera désormais ces attributs d'expérience sur un pied d'égalité, voire plus, que les fonctionnalités et les fonctions. L'entreprise considérera également ces éléments comme essentiels à la croissance et au succès à long terme de votre offre SaaS.

Version initiale

Pour comprendre le SaaS, commençons par une notion assez simple de ce que nous cherchons à atteindre lors de la création d'une entreprise SaaS. Le meilleur point de départ est d'examiner comment les logiciels traditionnels (non SaaS) ont été créés, gérés et exploités.

Le schéma suivant fournit une vue conceptuelle de la manière dont plusieurs fournisseurs ont packagé et livré leurs solutions.



Le modèle classique pour l'emballage et la fourniture de solutions logicielles

Dans ce schéma, nous avons décrit un ensemble d'environnements clients. Ces clients représentent les différentes entreprises ou entités qui ont acheté le logiciel d'un fournisseur. Chacun de ces clients fonctionne essentiellement dans un environnement autonome dans lequel il a installé le produit d'un fournisseur de logiciels.

Dans ce mode, l'installation de chaque client est traitée comme un environnement autonome dédié à ce client. Cela signifie que les clients se considèrent comme les propriétaires de ces environnements et peuvent demander une personnalisation ponctuelle ou des configurations uniques répondant à leurs besoins.

L'un des effets secondaires courants de cet état d'esprit est que les clients contrôleront la version du produit qu'ils utilisent. Cela peut se produire pour différentes raisons. Les clients peuvent être inquiets

face aux nouvelles fonctionnalités ou s'inquiéter des perturbations liées à l'adoption d'une nouvelle version.

Vous pouvez imaginer l'impact de cette dynamique sur l'empreinte opérationnelle du fournisseur de logiciels. Plus vous permettez aux clients de disposer d'environnements uniques, plus il devient difficile de gérer, de mettre à jour et de prendre en charge les différentes configurations de chaque client.

Ce besoin d'environnements ponctuels oblige souvent les entreprises à créer des équipes dédiées qui fournissent une expérience de gestion et d'exploitation distincte à chaque client. Bien que certaines de ces ressources puissent être partagées entre les clients, ce modèle introduit généralement des dépenses supplémentaires pour chaque nouveau client intégré.

Le fait que chaque client exécute ses solutions dans son propre environnement (dans le cloud ou sur site) a également un impact sur les coûts. Bien que vous puissiez essayer de faire évoluer ces environnements, la mise à l'échelle sera limitée à l'activité d'un seul client. Pour l'essentiel, votre optimisation des coûts se limite à ce que vous pouvez réaliser dans le cadre d'un environnement client individuel. Cela signifie également que vous pourriez avoir besoin de stratégies de mise à l'échelle distinctes pour tenir compte des variations d'activité entre les clients.

Dans un premier temps, certains éditeurs de logiciels considéreront ce modèle comme une construction puissante. Ils utilisent la possibilité de fournir une personnalisation ponctuelle en tant qu'outil de vente, permettant aux nouveaux clients d'imposer des exigences propres à leur environnement. Bien que le nombre de clients et la croissance de l'entreprise restent modestes, ce modèle semble parfaitement durable.

Cependant, à mesure que les entreprises commencent à connaître un succès croissant, les contraintes de ce modèle commencent à créer de véritables défis. Imaginez, par exemple, un scénario dans lequel votre entreprise atteint un pic de croissance significatif qui vous amène à attirer de nombreux nouveaux clients à un rythme rapide. Cette croissance va commencer à alourdir les frais d'exploitation, la complexité de la gestion, les coûts et toute une série d'autres problèmes.

En fin de compte, les frais généraux et l'impact collectifs de ce modèle peuvent commencer à compromettre fondamentalement le succès d'une entreprise de logiciels. Le premier point sensible pourrait être l'efficacité opérationnelle. L'augmentation du personnel et des coûts associés à l'acquisition de clients commence à éroder les marges de l'entreprise.

Toutefois, les problèmes opérationnels ne constituent qu'une partie du défi. Le véritable problème est que ce modèle, à mesure qu'il évolue, commence à avoir un impact direct sur la capacité de

l'entreprise à lancer de nouvelles fonctionnalités et à suivre le rythme du marché. Lorsque chaque client possède son propre environnement, les fournisseurs doivent trouver un équilibre entre la mise à jour, la migration et les exigences du client lorsqu'ils tentent d'introduire de nouvelles fonctionnalités dans leur système.

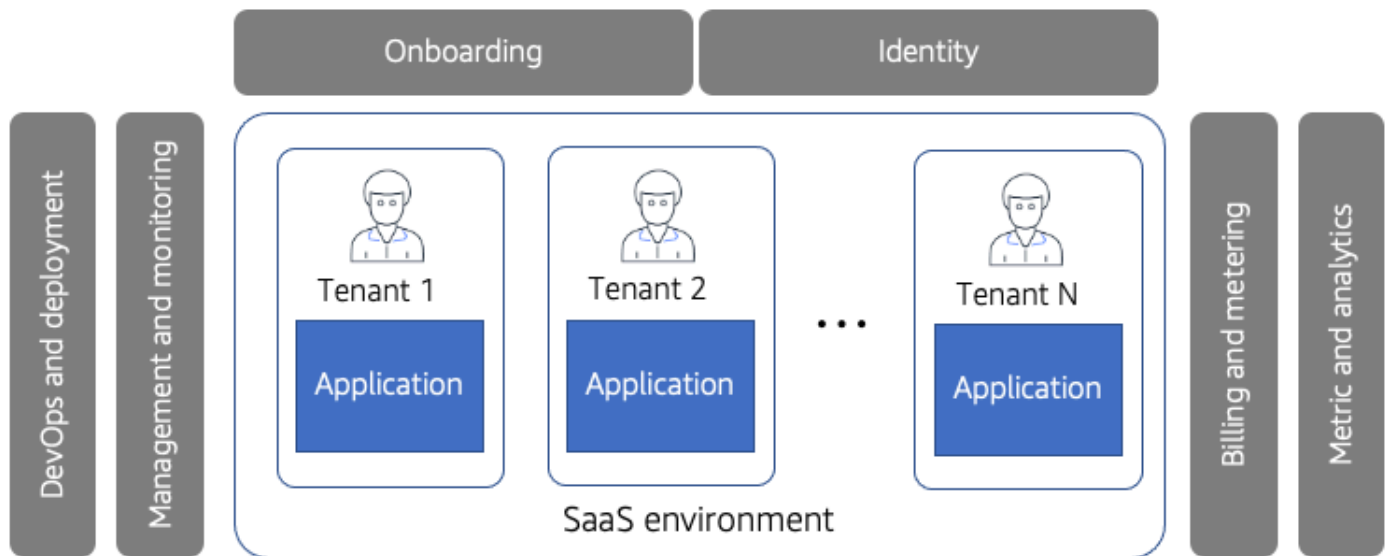
Cela entraîne généralement des cycles de publication plus longs et plus complexes, ce qui tend à réduire le nombre de versions effectuées chaque année. Plus important encore, cette complexité incite les équipes à consacrer plus de temps à l'analyse de chaque nouvelle fonctionnalité bien avant qu'elle ne soit mise à la disposition du client. Les équipes commencent à se concentrer davantage sur la validation des nouvelles fonctionnalités et moins sur la rapidité de livraison. Les coûts liés au lancement de nouvelles fonctionnalités deviennent si importants que les équipes se concentrent davantage sur les mécanismes de test et moins sur les nouvelles fonctionnalités qui sous-tendent l'innovation de leur offre.

Dans ce mode plus lent et plus prudent, les équipes ont tendance à avoir de longs cycles, ce qui crée un écart de plus en plus important entre le début d'une idée et le moment où elle arrive entre les mains d'un client. Dans l'ensemble, cela peut nuire à votre capacité à réagir à la dynamique du marché et aux pressions concurrentielles.

Passer à une expérience unifiée

Pour répondre à ce dilemme logiciel classique, les entreprises se tournent vers un modèle qui leur permet de créer une expérience unique et unifiée qui permet de gérer et d'exploiter les clients de manière collective.

Le schéma suivant fournit une vue conceptuelle d'un environnement dans lequel tous les clients sont gérés, intégrés, facturés et exploités via un modèle partagé.



Une vue conceptuelle d'un environnement dans lequel tous les clients sont gérés, intégrés, facturés et exploités via un modèle partagé

À première vue, cela peut ne pas sembler très différent du modèle précédent. Cependant, en approfondissant un peu plus nos recherches, vous constaterez qu'il existe des différences fondamentales et significatives entre ces deux approches.

Tout d'abord, vous remarquerez que les environnements clients ont été renommés locataires. Cette notion de locataire est à la base du SaaS. L'idée de base est que vous disposez d'un environnement SaaS unique et que chacun de vos clients est considéré comme un locataire de cet environnement, consommant les ressources dont il a besoin. Un locataire peut être une entreprise comptant de nombreux utilisateurs, ou il peut être directement lié à un utilisateur individuel.

Pour mieux comprendre l'idée d'un locataire, considérez l'idée d'un appartement ou d'un immeuble commercial. L'espace de chacun de ces bâtiments est loué à des locataires individuels. Les

locataires dépendent de certaines des ressources partagées du bâtiment (eau, électricité, etc.) et payent ce qu'ils consomment.

Les locataires SaaS suivent un schéma similaire. Vous disposez de l'infrastructure de votre environnement SaaS et de locataires qui utilisent l'infrastructure de cet environnement. La quantité de ressources consommée par chaque locataire peut varier. Ces locataires sont également gérés, facturés et exploités collectivement.

Si vous revenez au diagramme, vous verrez la notion de location prendre vie. Ici, les locataires n'ont plus leur propre environnement. Au lieu de cela, tous les locataires sont hébergés et gérés dans les murs d'un environnement SaaS collectif.

Le diagramme inclut également une gamme de services partagés qui entourent votre environnement SaaS. Ces services s'adressent à tous les locataires de votre environnement SaaS à l'échelle mondiale. Cela signifie que l'intégration et l'identité, par exemple, sont partagées par tous les locataires de cet environnement. Il en va de même pour la gestion, les opérations, le déploiement, la facturation et les métriques.

Cette idée d'un ensemble unifié de services appliqués de manière universelle à tous vos locataires est à la base du SaaS. En partageant ces concepts, vous êtes en mesure de relever un certain nombre de défis associés au modèle classique décrit ci-dessus.

Un autre élément clé, quelque peu subtil, de ce diagramme est que tous les locataires de cet environnement exécutent la même version de votre application. L'idée d'exécuter des versions distinctes et uniques pour chaque client est révolue. Le fait que tous les locataires exécutent la même version représente l'un des attributs distinctifs fondamentaux d'un environnement SaaS.

En faisant en sorte que tous les clients utilisent la même version de votre produit, vous n'êtes plus confronté à de nombreux défis liés à un modèle de logiciel installé classique. Dans le modèle unifié, les nouvelles fonctionnalités peuvent être déployées auprès de tous les locataires par le biais d'un processus unique et partagé.

Cette approche vous permet d'utiliser une interface opérationnelle unique capable de gérer et d'exploiter tous les locataires. Cela vous permet de gérer et de surveiller vos locataires grâce à une expérience opérationnelle commune, ce qui permet d'ajouter de nouveaux locataires sans ajouter de frais opérationnels supplémentaires. Il s'agit d'un élément essentiel de la proposition de valeur du SaaS qui permet aux équipes de réduire les dépenses opérationnelles et d'améliorer l'agilité globale de l'organisation.

Imaginez ce que cela signifierait d'ajouter 100 ou 1 000 nouveaux clients à ce modèle. Au lieu de vous inquiéter de la façon dont ces nouveaux clients pourraient éroder vos marges et ajouter de la complexité, vous pouvez considérer cette croissance comme une opportunité qu'elle représente.

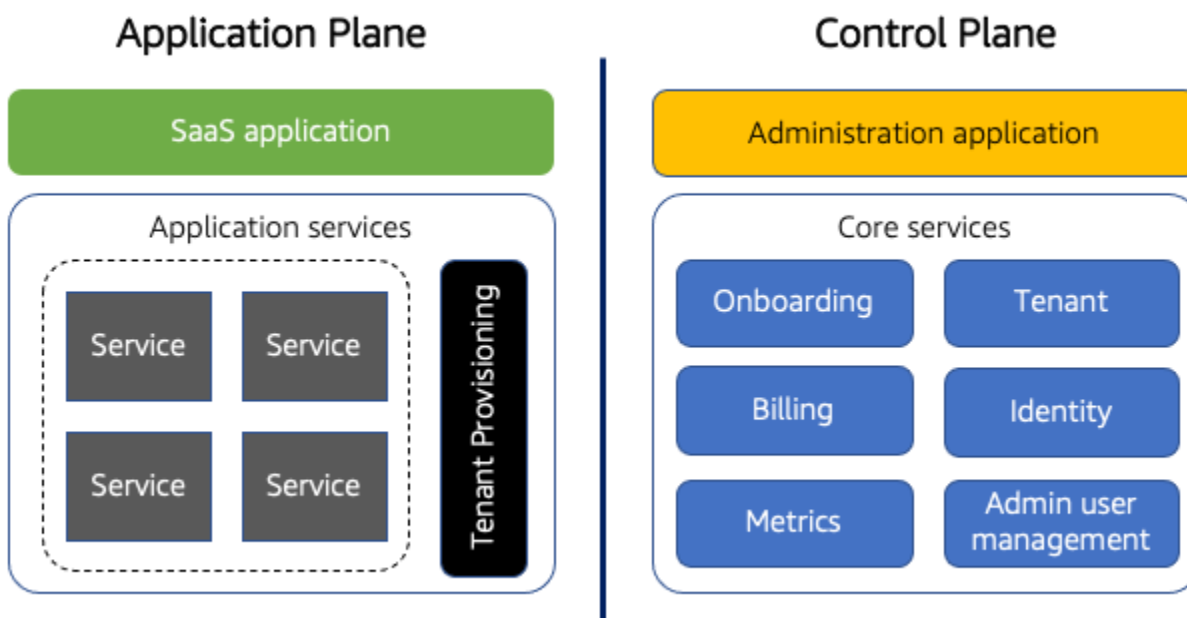
En général, le SaaS se concentre sur la manière dont l'application au milieu de ce modèle est mise en œuvre. Les entreprises souhaitent se concentrer sur la manière dont les données sont stockées, sur la manière dont les ressources sont partagées, etc. Cependant, la réalité est que, bien que ces détails soient indéniablement importants, il existe de nombreuses façons de créer votre application tout en se présentant comme une solution SaaS pour vos clients.

Ce qui est essentiel, c'est l'objectif plus large de disposer d'une expérience unique et unifiée qui entoure les environnements de vos locataires. Cette expérience partagée vous permet de stimuler la croissance, l'agilité et l'efficacité opérationnelle qui sont liées aux objectifs généraux d'une entreprise SaaS.

Plan de contrôle ou plan d'application

Le schéma précédent fournit une vue conceptuelle des principaux concepts de l'architecture SaaS. Examinons maintenant cela plus en profondeur et définissons mieux la façon dont votre environnement SaaS se décompose en couches distinctes. Cette vision plus claire des frontières entre les concepts SaaS facilitera la description des éléments évolutifs d'une solution SaaS.

Le schéma suivant divise votre environnement SaaS en deux plans distincts. Le plan de contrôle est à droite. Cette partie du diagramme inclut toutes les fonctionnalités et tous les services utilisés pour intégrer, authentifier, gérer, exploiter et analyser un environnement mutualisé.



Plan de contrôle ou plan d'application

Ce plan de contrôle est à la base de tout modèle SaaS multi-tenant. Chaque solution SaaS, indépendamment du déploiement des applications et du schéma d'isolation, doit inclure les services qui vous permettent de gérer et d'exploiter vos clients via une expérience unique et unifiée.

Dans le plan de contrôle, nous l'avons ensuite décomposé en deux éléments distincts. Les services principaux présentés ici représentent l'ensemble des services utilisés pour orchestrer votre expérience mutualisée. Nous avons inclus certains des exemples courants de services qui font généralement partie du noyau, en reconnaissant que les services de base peuvent varier d'une solution SaaS à l'autre.

Vous remarquerez également que nous présentons une application d'administration distincte. Cela représente l'application (une application Web, une interface de ligne de commande ou une API) qui peut être utilisée par un fournisseur SaaS pour gérer son environnement mutualisé.

Une mise en garde importante est que le plan de contrôle et ses services ne sont pas réellement mutualisés. La fonctionnalité ne fournit pas les attributs fonctionnels réels de votre application SaaS (qui doit être mutualisée). Si vous examinez l'un des services principaux, par exemple, vous ne trouverez pas l'isolation des locataires ni les autres structures faisant partie de la fonctionnalité de votre application mutualisée. Ces services s'adressent à tous les locataires dans le monde entier.

Le côté gauche du diagramme fait référence au plan d'application d'un environnement SaaS. C'est là que réside la fonctionnalité multi-tenant de votre application. Ce qui apparaît dans le diagramme doit rester quelque peu vague, car chaque solution peut être déployée et décomposée différemment en fonction des besoins de votre domaine, de l'empreinte de votre technologie, etc.

Le domaine d'application est séparé en deux éléments. Il existe l'application SaaS qui représente l'expérience/l'application du locataire pour votre solution. Il s'agit de la surface que les locataires touchent pour interagir avec votre application SaaS. Ensuite, il y a les services dorsaux qui représentent la logique métier et les éléments fonctionnels d'une solution SaaS. Il peut s'agir de microservices ou d'autres packages de vos services applicatifs.

Vous remarquerez également que nous avons réparti le provisionnement. Cela permet de mettre en évidence le fait que tout approvisionnement de ressources pour les locataires lors de l'intégration ferait partie de ce domaine d'application. Certains diront que cela a sa place dans le plan de contrôle. Nous l'avons toutefois placé dans le domaine de l'application, car les ressources qu'il doit provisionner et configurer sont connectées plus directement aux services créés et configurés dans le plan de l'application.

En divisant cela en plans distincts, il est plus facile de réfléchir au paysage global d'une architecture SaaS. Plus important encore, cela met en évidence la nécessité de disposer d'un ensemble de services qui sortent totalement du cadre des fonctionnalités de votre application.

Principales services

Le plan de contrôle référencé précédemment mentionne une série de services de base qui représentent les services typiques utilisés pour intégrer, gérer et exploiter un environnement SaaS. Il peut être utile de souligner davantage le rôle de certains de ces services afin de mettre en évidence leur portée et leur objectif dans un environnement SaaS. Vous trouverez ci-dessous un bref résumé de chacun de ces services :

- **Intégration** — Chaque solution SaaS doit fournir un mécanisme fluide pour intégrer de nouveaux locataires dans votre environnement SaaS. Il peut s'agir d'une page d'inscription en libre-service ou d'une expérience gérée en interne. Quoi qu'il en soit, une solution SaaS doit faire tout ce qui est en son pouvoir pour éliminer les frictions internes et externes liées à cette expérience et garantir la stabilité, l'efficacité et la répétabilité de ce processus. Il joue un rôle essentiel dans le soutien à la croissance et à l'échelle d'une entreprise SaaS. En général, ce service orchestre d'autres services pour créer des utilisateurs, des locataires, des politiques d'isolement, des ressources de provisionnement et par locataire.
- **Locataire** : le service aux locataires permet de centraliser les politiques, les attributs et l'état des locataires. L'essentiel est que les locataires ne sont pas des utilisateurs individuels. En fait, un locataire est probablement associé à de nombreux utilisateurs.
- **Identité** : les systèmes SaaS ont besoin d'un moyen clair de connecter les utilisateurs aux locataires afin d'intégrer le contexte des locataires à l'expérience d'authentification et d'autorisation de leurs solutions. Cela influence à la fois l'expérience d'intégration et la gestion globale des profils utilisateurs.
- **Facturation** — Dans le cadre de l'adoption du SaaS, les entreprises adoptent souvent de nouveaux modèles de facturation. Ils peuvent également explorer l'intégration avec des fournisseurs de facturation tiers. Ce service de base vise principalement à faciliter l'intégration des nouveaux locataires et à collecter des données de consommation et d'activité qui sont utilisées pour générer des factures pour les locataires.
- **Métriques** : les équipes SaaS s'appuient largement sur leur capacité à capturer et à analyser de riches données métriques qui leur permettent de mieux comprendre comment les locataires utilisent leur système, comment ils consomment les ressources et comment leurs locataires interagissent avec leurs systèmes. Ces données sont utilisées pour façonner les stratégies opérationnelles, de produits et commerciales.
- **Gestion des utilisateurs administrateurs** : les systèmes SaaS doivent prendre en charge à la fois les utilisateurs locataires et les utilisateurs administrateurs. Les utilisateurs administrateurs

représentent les administrateurs d'un fournisseur SaaS. Ils se connecteront à votre expérience opérationnelle pour surveiller et gérer votre environnement SaaS.

Redéfinir la mutualisation

Les termes « multi-tenancy » et « SaaS » sont souvent étroitement liés. Dans certains cas, les organisations décrivent le SaaS et la mutualisation comme une seule et même chose. Bien que cela puisse sembler naturel, le fait d'assimiler le SaaS à la mutualisation tend à amener les équipes à adopter une vision purement technique du SaaS alors qu'en réalité, le SaaS est davantage un modèle commercial qu'une stratégie d'architecture.

Pour mieux comprendre ce concept, commençons par la vision classique de la mutualisation. Dans cette vision purement axée sur l'infrastructure, la mutualisation est utilisée pour décrire la manière dont les ressources sont partagées par les locataires afin de promouvoir l'agilité et la rentabilité.

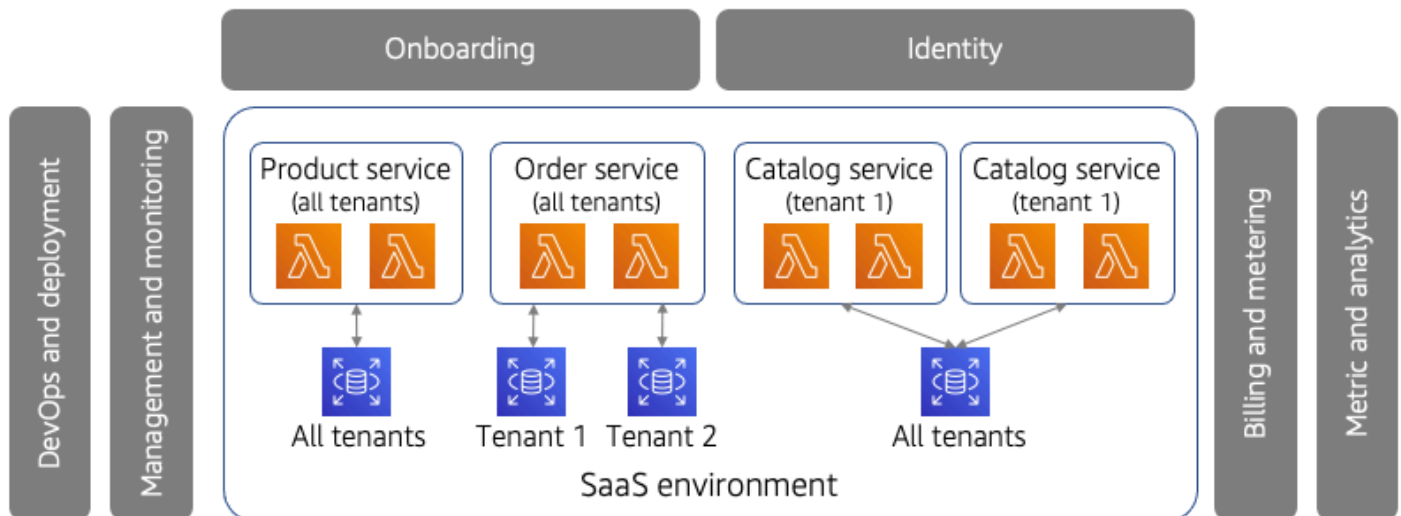
Supposons, par exemple, que vous disposiez d'un microservice ou d'une instance [Amazon Elastic Compute Cloud](#) (Amazon EC2) utilisée par plusieurs locataires de votre système SaaS. Ce service serait considéré comme fonctionnant dans un modèle multi-tenant, car les locataires partagent l'utilisation de l'infrastructure qui exécute ce service.

Le défi de cette définition est qu'elle associe trop directement la notion technique de mutualisation au SaaS. Cela suppose que la caractéristique déterminante du SaaS est qu'il doit disposer d'une infrastructure partagée et mutualisée. Cette vision du SaaS commence à s'effondrer lorsque nous examinons les différentes manières dont le SaaS est réalisé dans différents environnements.

Le schéma suivant donne une vue d'un système SaaS qui expose certains des défis que pose la définition de la mutualisation.

Vous découvrirez ici le modèle SaaS classique décrit précédemment, avec une série de services applicatifs entourés de services partagés qui vous permettent de gérer et d'exploiter vos locataires collectivement.

Ce qui est nouveau, ce sont les microservices que nous avons inclus. Le diagramme inclut trois exemples de microservices : produit, commande et catalogue. Si vous examinez attentivement le modèle de location de chacun de ces services, vous remarquerez qu'ils utilisent tous des modèles de location légèrement différents.



SaaS et mutualisation

Le service produit partage toutes ses ressources (calcul et stockage) avec tous les locataires. Cela correspond à la définition classique de la mutualisation. Toutefois, si vous examinez le service de commande, vous constaterez qu'il dispose d'un calcul partagé, mais d'un stockage distinct pour chaque locataire.

Le service de catalogue ajoute une autre variante dans laquelle le calcul est distinct pour chaque locataire (déploiement de microservices distinct pour chaque locataire), mais partage le stockage pour tous les locataires.

Des variations de cette nature sont courantes dans les environnements SaaS. [Voisin bruyant](#), modèles de hiérarchisation, besoins d'isolation : voici quelques-unes des raisons pour lesquelles vous pourriez partager ou cloisonner certaines parties de votre solution SaaS de manière sélective.

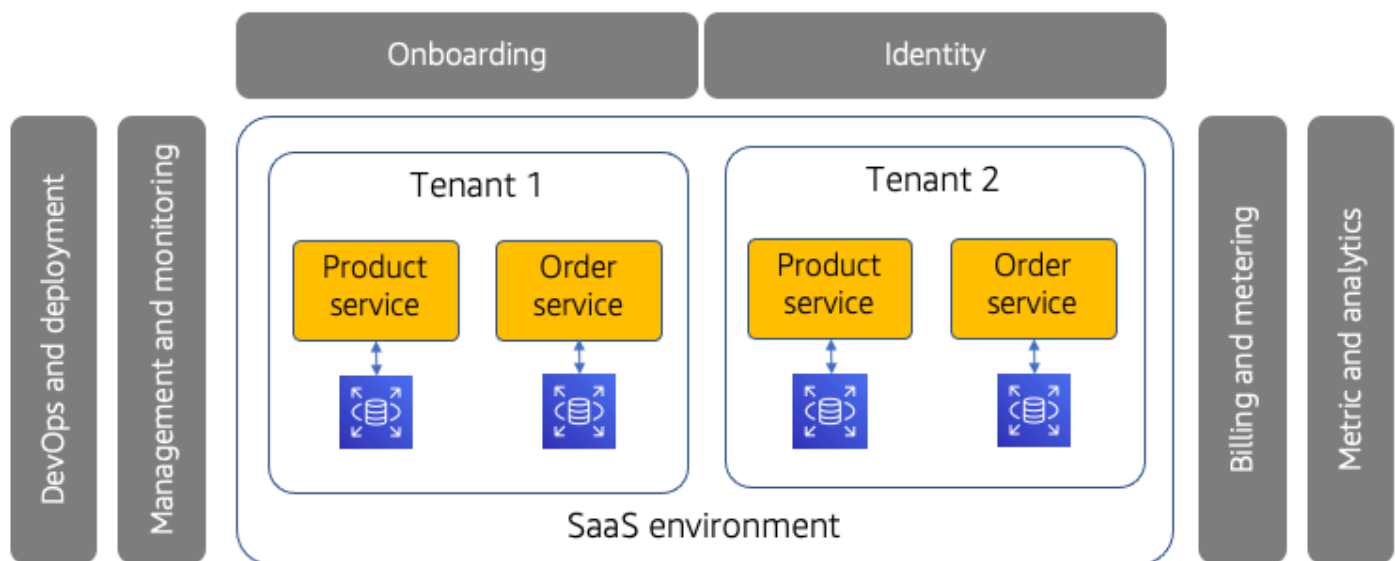
Compte tenu de ces variations, et de nombreuses autres possibilités, il devient de plus en plus difficile de déterminer comment utiliser le terme « multi-locataire » pour caractériser cet environnement. Dans l'ensemble, en ce qui concerne le client, il s'agit d'un environnement à locataires multiples. Cependant, si nous utilisons la définition la plus technique, certaines parties de cet environnement sont mutualisées et d'autres ne le sont pas.

C'est pourquoi il devient nécessaire de ne plus utiliser le terme « multi-tenant » pour caractériser les environnements SaaS. Nous pouvons plutôt parler de la manière dont la mutualisation est mise en œuvre au sein de votre application, mais évitez de l'utiliser pour qualifier une solution de SaaS. Si le terme multi-tenant doit être utilisé, il est plus logique de l'utiliser pour décrire l'ensemble de

l'environnement SaaS comme étant multi-tenant, sachant que certaines parties de l'architecture peuvent être partagées et d'autres non. Dans l'ensemble, vous exploitez et gérez toujours cet environnement dans le cadre d'un modèle multi-tenant.

Le cas extrême

Pour mieux mettre en évidence cette notion de location, examinons un modèle SaaS dans lequel les locataires ne partagent aucune ressource. Le schéma suivant fournit un exemple d'environnement SaaS utilisé par certains fournisseurs de SaaS.



Pile par locataire

Dans ce schéma, vous verrez que nous avons toujours notre environnement commun autour de ces locataires. Cependant, chaque locataire est déployé avec un ensemble de ressources dédié. Rien n'est partagé par les locataires dans ce modèle.

Cet exemple remet en question ce que signifie être multilocataire. S'agit-il d'un environnement mutualisé même si aucune des ressources n'est partagée ? Les locataires qui utilisent ce système ont les mêmes attentes que vous pourriez avoir à l'égard d'un environnement SaaS avec des ressources partagées. En fait, ils n'ont peut-être aucune idée de la manière dont leurs ressources sont déployées dans le cadre de l'environnement SaaS.

Même si ces locataires fonctionnent dans une infrastructure cloisonnée, ils sont toujours gérés et exploités collectivement. Ils partagent une expérience unifiée en matière d'intégration, d'identité, de

statistiques, de facturation et d'exploitation. De plus, lorsqu'une nouvelle version est publiée, elle est déployée sur tous les locataires. Pour que cela fonctionne, vous ne pouvez pas autoriser une personnalisation ponctuelle pour des locataires individuels.

Cet exemple extrême fournit un bon modèle pour tester la notion de SaaS à locataires multiples. Même s'il ne permet pas de tirer pleinement parti de l'efficacité d'une infrastructure partagée, il s'agit d'un environnement SaaS multi-locataires entièrement valide. Pour certains clients, leur domaine peut imposer à certains clients ou à tous d'utiliser ce modèle. Cela ne signifie pas qu'ils ne sont pas des SaaS. S'ils utilisent ces services partagés et que tous les locataires utilisent la même version, cela reste conforme aux principes fondamentaux du SaaS.

Compte tenu de ces paramètres et de cette définition plus large du SaaS, vous pouvez constater la nécessité de faire évoluer l'utilisation du terme « multi-tenant ». Il est plus logique de désigner tout système SaaS géré et exploité collectivement comme un système multi-tenant. Vous pouvez ensuite utiliser une terminologie plus précise pour décrire la manière dont les ressources sont partagées ou dédiées dans le cadre de la mise en œuvre d'une solution SaaS.

Suppression du terme de locataire unique

Dans le cadre de l'utilisation du terme multitenant, il est tout à fait naturel que les utilisateurs souhaitent utiliser le terme locataire unique pour décrire les environnements SaaS. Toutefois, compte tenu du contexte décrit précédemment, l'expression locataire unique prête à confusion.

Le schéma précédent est-il un environnement à locataire unique ? Bien que chaque locataire dispose techniquement de son propre parc, ces locataires sont toujours exploités et gérés dans le cadre d'un modèle multitenant. C'est pourquoi le terme locataire unique est généralement évité. Au lieu de cela, tous les environnements sont caractérisés comme mutualisés, car ils mettent simplement en œuvre une variante de location dans laquelle certaines ou toutes les ressources sont partagées ou dédiées.

Présentation du silo et du pool

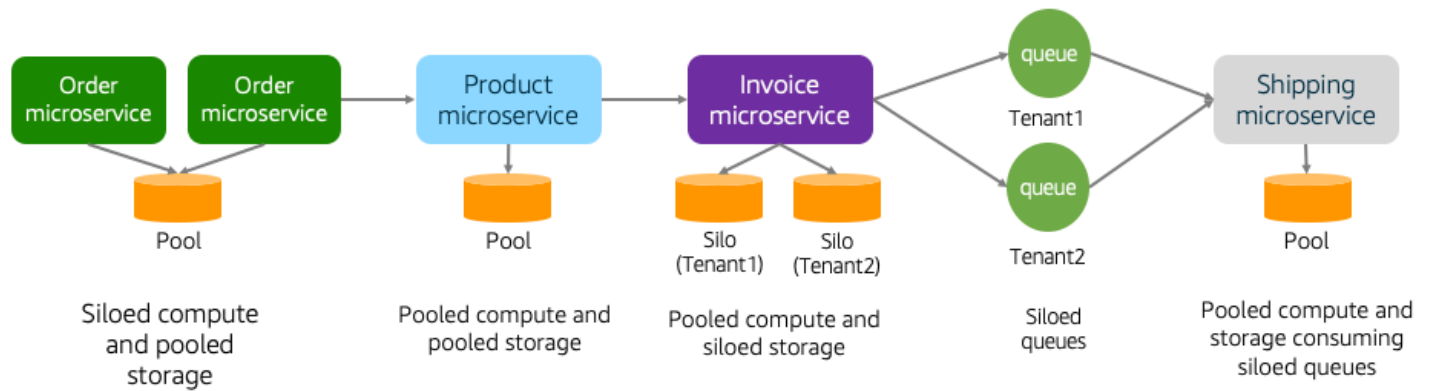
Compte tenu de toutes ces variations de modèles et des défis liés au terme multitenant, nous avons introduit une terminologie qui nous permet de capturer et de décrire avec plus de précision les différents modèles utilisés lors de la création d'un SaaS.

Les deux termes que nous utilisons pour caractériser l'utilisation des ressources dans un environnement SaaS sont silo et pool. Ces termes nous permettent d'étiqueter la nature des environnements SaaS, en utilisant le multi-tenant comme description globale qui peut être appliquée à un certain nombre de modèles sous-jacents.

Au niveau le plus élémentaire, le terme silo est destiné à décrire des scénarios dans lesquels une ressource est dédiée à un locataire donné. Inversement, le modèle de pool est utilisé pour décrire des scénarios dans lesquels une ressource est partagée par des locataires.

Lorsque nous examinons la façon dont les termes silo et pool sont utilisés, il est important de préciser que silo et pool ne sont pas all-or-nothing des concepts. Le silo et le pool peuvent s'appliquer à l'ensemble des ressources d'un locataire, ou ils peuvent être appliqués de manière sélective à des parties de votre environnement SaaS global. Ainsi, si nous disons qu'une ressource utilise un modèle de silo, cela ne signifie pas que toutes les ressources de cet environnement sont cloisonnées. Il en va de même pour la façon dont nous utiliserions le terme « groupé ».

Le schéma suivant fournit un exemple de la manière dont les modèles cloisonnés et groupés sont utilisés de manière plus granulaire dans un environnement SaaS :



Modèles de silos et de piscines

Ce diagramme comprend une série d'exemples destinés à illustrer la nature plus ciblée des modèles de silo et de pool. Si vous suivez cette procédure de gauche à droite, vous verrez que nous commençons par un microservice de commande. Ce microservice dispose d'un calcul cloisonné et d'un stockage groupé. Il interagit avec un produit et un service qui met en commun le calcul et le stockage.

Le service produit interagit ensuite avec un microservice de facturation qui dispose d'un traitement groupé et d'un stockage cloisonné. Ce service envoie des messages via des files d'attente au service d'expédition. Les files d'attente sont déployées dans un modèle cloisonné.

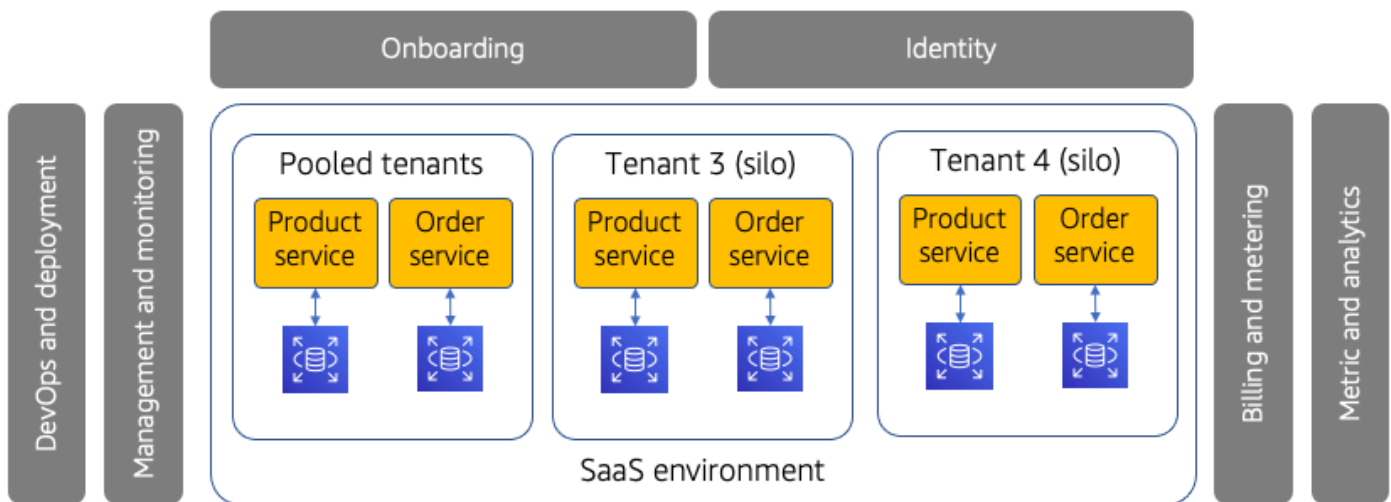
Enfin, le microservice d'expédition acquiert les messages provenant des files d'attente cloisonnées. Il utilise des capacités de calcul et de stockage groupées.

Bien que cela puisse sembler un peu compliqué, l'objectif est de mettre en évidence la nature granulaire des concepts de silo et de pool. Lorsque vous envisagez de concevoir et de créer votre solution SaaS, vous êtes censé prendre ces décisions en matière de silo et de pool en fonction des besoins de votre domaine et de vos clients.

Le bruit du voisinage, l'isolement, la hiérarchisation et de nombreuses autres raisons peuvent influencer la manière et le moment où vous choisissez d'appliquer le modèle en silo ou en pool.

Silo et pool complets

Le silo et le pool peuvent également être utilisés pour décrire une pile SaaS complète. Dans cette approche, toutes les ressources d'un locataire sont déployées de manière dédiée ou partagée. Le schéma suivant fournit un exemple de la façon dont cela peut se produire dans un environnement SaaS.



Modèles de silos et de pools à empilage complet

Dans ce diagramme, vous verrez qu'il existe trois modèles différents pour vos déploiements de locataires complets. Tout d'abord, vous verrez qu'il existe un environnement de pool complet. Les locataires de ce pool partagent toutes les ressources (calcul, stockage, etc.).

Les deux autres piles présentées sont destinées à représenter des environnements locataires cloisonnés à pile complète. Dans ce cas, les locataires 3 et 4 sont présentés comme ayant chacun leurs propres piles dédiées où aucune ressource n'est partagée avec d'autres locataires.

Ce mélange de silos et de modèles groupés dans le même environnement SaaS n'est pas si atypique. Imaginez, par exemple, que vous ayez un ensemble de locataires de niveau de base qui paient un prix modéré pour l'utilisation de votre système. Ces locataires sont placés dans l'environnement mutualisé.

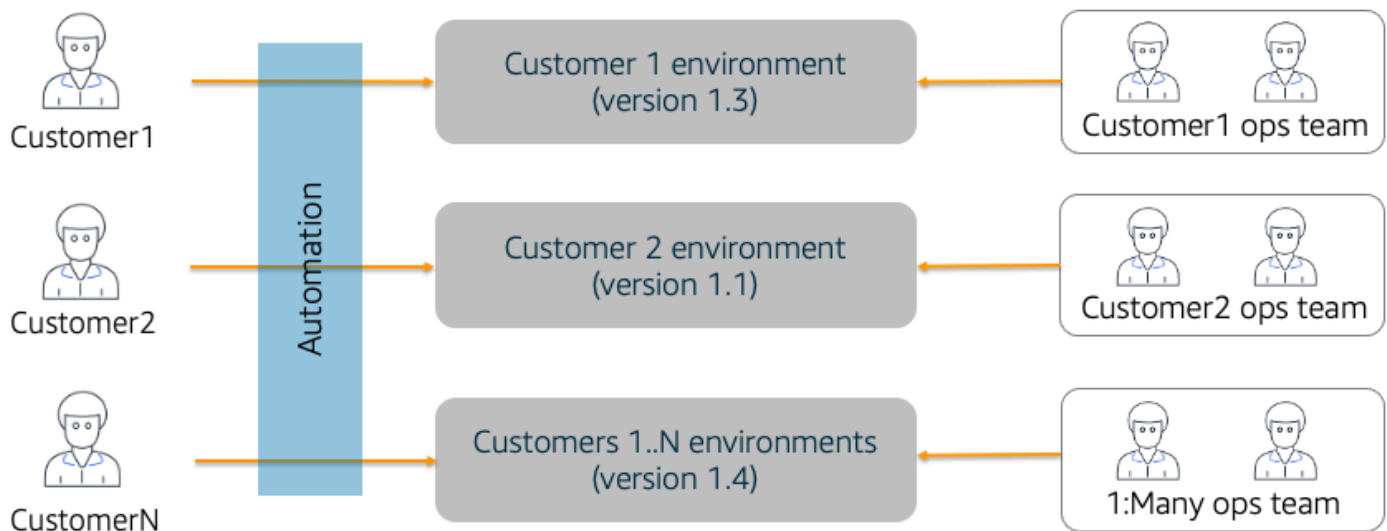
Dans le même temps, vous pouvez également avoir des locataires haut de gamme prêts à payer plus cher pour avoir le privilège de fonctionner dans un silo. Ces clients sont déployés avec des piles distinctes (comme indiqué dans le schéma).

Même dans ce modèle, où vous avez peut-être autorisé les locataires à fonctionner dans leur propre silo complet, il serait essentiel que ces silos ne permettent aucune variation ou personnalisation ponctuelle pour ces locataires. À tous égards, chacune de ces piles doit exécuter la même configuration de la pile, avec la même version du logiciel. Lorsqu'une nouvelle version est publiée, elle est déployée dans l'environnement mutualisé et dans chacun des environnements cloisonnés.

SaaS ou fournisseur de services gérés (MSP)

Une certaine confusion entoure également les frontières entre les modèles SaaS et les modèles de fournisseurs de services gérés (MSP). Si vous examinez un modèle MSP, il peut sembler avoir des objectifs similaires à ceux du modèle SaaS.

Cependant, si vous étudiez un peu plus le MSP, vous constaterez que le MSP et le SaaS sont en fait différents. Le schéma suivant fournit une vue conceptuelle d'un environnement MSP.



Modèle de fournisseur de services gérés (MSP)

Ce diagramme représente une approche du modèle MSP. Sur la gauche, vous verrez les clients qui utilisent le modèle MSP. En général, l'approche consiste à utiliser n'importe quelle automatisation disponible pour approvisionner l'environnement de chaque client et installer le logiciel pour ce client.

Sur la droite se trouve une approximation de l'empreinte opérationnelle que le MSP fournirait pour prendre en charge ces environnements clients.

Il est important de noter que le MSP installe et gère souvent une version du produit qu'un client souhaite exécuter. Tous les clients peuvent utiliser la même version, mais cela n'est généralement pas obligatoire dans un modèle MSP.

La stratégie générale consiste à simplifier la vie d'un fournisseur de logiciels en lui confiant l'installation et la gestion de ces environnements. Bien que cela simplifie la vie du fournisseur, cela ne correspond pas directement aux valeurs et à l'état d'esprit qui sont essentiels à une offre SaaS.

L'accent est mis sur le délestage de la responsabilité de gestion. Cette décision ne revient pas à faire fonctionner tous les clients sur la même version avec une expérience de gestion et d'exploitation unique et unifiée. Au lieu de cela, MSP autorise souvent des versions distinctes et traite souvent chacun de ces environnements comme étant distinct sur le plan opérationnel.

Il existe certainement des domaines dans lesquels le MSP pourrait commencer à se chevaucher avec le SaaS. Si le MSP exigeait essentiellement que tous ses clients exécutent la même version et qu'il était en mesure d'intégrer, de gérer, d'exploiter et de facturer tous les locataires de manière centralisée via une seule expérience, il se pourrait qu'il s'agisse d'une solution SaaS plutôt que d'une solution MSP.

Le thème général est que l'automatisation de l'installation des environnements n'équivaut pas à disposer d'un environnement SaaS. Ce n'est que lorsque vous ajouterez toutes les autres mises en garde évoquées précédemment que cela représentera davantage un véritable modèle SaaS.

Si nous nous éloignons des aspects technologiques et opérationnels de cette histoire, la frontière entre MSP et SaaS devient encore plus nette. En général, en tant qu'entreprise SaaS, le succès de votre offre dépend de votre capacité à vous impliquer profondément dans tous les aspects évolutifs de l'expérience.

Cela implique généralement de prendre le pouls de l'expérience d'intégration, de comprendre l'impact des événements opérationnels sur les locataires, de suivre les indicateurs et les analyses clés, et d'être proche de votre client. Dans un modèle MSP où cela est transféré à quelqu'un d'autre, vous risquez de vous retrouver à un niveau éloigné des détails clés qui sont essentiels à l'exploitation d'une entreprise SaaS.

Migration vers le SaaS

De nombreux fournisseurs qui adoptent le SaaS migrent vers le SaaS à partir d'un modèle de logiciel installé traditionnel (décrit précédemment). Pour ces fournisseurs, il est particulièrement important de bien s'aligner sur les principes fondamentaux du SaaS.

Encore une fois, c'est là qu'il peut y avoir une certaine confusion quant à ce que signifie migrer vers un modèle SaaS. Certains, par exemple, considèrent la migration vers le cloud comme une migration vers le SaaS. D'autres considèrent que l'automatisation de leur processus d'installation et de provisionnement constitue une réalisation de la migration.

Il va sans dire que chaque organisation peut partir d'une situation différente, avoir des considérations liées à son héritage et être confrontée à des pressions différentes du marché et de la concurrence. Cela signifie que chaque migration aura un aspect différent.

Bien que chaque voie soit différente, il existe des domaines dans lesquels les principes fondamentaux qui façonnent les stratégies de migration présentent des différences. Un bon alignement sur les concepts et les principes peut avoir un impact significatif sur la réussite globale de votre migration SaaS.

Sur la base des concepts décrits précédemment, il doit être clair que le passage au SaaS commence par la stratégie et les objectifs de l'entreprise. Ce point peut être oublié dans un contexte de migration où la pression s'exerce pour passer au SaaS le plus rapidement possible.

Dans ce mode, les entreprises considèrent souvent la migration principalement comme un exercice technique. En réalité, chaque migration SaaS doit commencer par une vision claire des clients cibles, de l'expérience de service, des objectifs opérationnels, etc. Le fait de vous concentrer davantage sur ce à quoi doit ressembler votre activité SaaS aura un impact profond sur la forme, les priorités et la voie que vous emprunterez pour migrer votre solution vers le SaaS.

Le fait d'avoir cette vision claire dès le début de votre migration jette les bases de la manière dont vous migrez à la fois la technologie et l'entreprise dans le cadre de votre transition vers le SaaS. Au fur et à mesure que vous vous engagez dans cette voie, concentrez-vous sur les questions qui peuvent vous en dire le plus sur la direction que vous prenez.

Le tableau suivant donne un aperçu de la nature contrastée des mentalités techniques et commerciales en matière de migration.

Tableau 1 — Migration d'abord technique et migration d'abord vers l'entreprise

L'état d'esprit axé sur la technologie	L'esprit d'entreprise d'abord
Comment isoler les données des locataires ?	Comment le SaaS peut-il nous aider à développer notre activité ?
Comment connecter les utilisateurs aux locataires ?	Quels segments ciblons-nous ?
Comment éviter le bruit des voisins ?	Quels sont la taille et le profil de ces segments ?
Comment effectuons-nous un exemple de ce à quoi pourrait ressembler la réponse ?	Quels niveaux devons-nous prendre en charge ?
Comment pouvons-nous nous adapter en fonction de la charge de locataires ?	Quelle expérience de service ciblons-nous ?
Quel fournisseur de facturation devons-nous utiliser ?	Quelle est notre stratégie en matière de prix et d'emballage ?

Voici un exemple de ce à quoi pourrait ressembler la réponse de ce à quoi pourrait ressembler la réponse. L'équipe d'ingénierie se concentre sur les sujets classiques du multitenant qui sont certainement importants pour toute architecture SaaS.

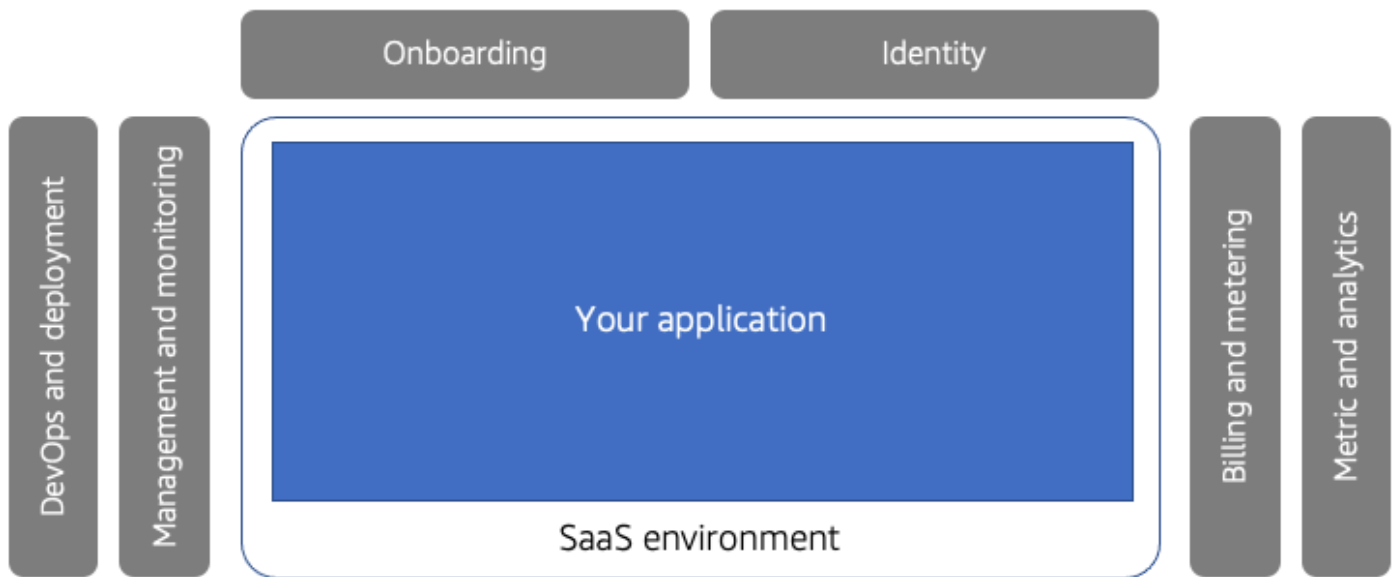
Le problème est que les réponses à de nombreuses questions de gauche sont souvent directement influencées par les réponses à la question de droite. Il est peu probable que ce point soit nouveau pour quiconque s'intéresse à la migration. Cependant, la réalité est que de nombreuses organisations commencent par rechercher la rentabilité et la rentabilité, en supposant que les aspects commerciaux se débrouilleront d'eux-mêmes.

Dans le cadre de cette stratégie de migration, il peut également y avoir confusion quant à la manière dont votre environnement existant pourrait être évolué pour s'adapter à un modèle SaaS. Il s'agit également d'un domaine dans lequel il existe une multitude d'options pour migrer vers le SaaS. Cependant, il existe un système de valeurs communes que nous préconisons généralement pour toute migration.

Dans notre discussion précédente sur les principes du SaaS, nous avons décrit différents modèles et terminologies utilisés pour décrire les environnements SaaS. Un thème commun à toutes ces

solutions est l'idée de partager des services autour de votre application. Identité, intégration, métriques, facturation : tous ces éléments sont considérés comme des éléments communs au cœur de tout environnement SaaS.

Maintenant que nous examinons la migration, vous constaterez que ces mêmes services partagés jouent un rôle clé dans tout processus de migration. Le schéma suivant fournit une vue conceptuelle du paysage migratoire.



Migration vers le SaaS

Ce diagramme représente l'expérience cible pour n'importe quel chemin de migration. Il inclut les mêmes services partagés que ceux décrits précédemment. Au milieu se trouve un espace réservé à votre application.

L'idée principale est que vous pouvez placer n'importe quel nombre de modèles d'application au milieu de cet environnement. Lors de la première étape de la migration, chaque locataire peut fonctionner dans son propre silo. Vous pouvez également avoir une architecture hybride dans laquelle les éléments sont cloisonnés et d'autres fonctionnalités sont traitées par le biais d'un ensemble de microservices modernisés.

L'ampleur de la modernisation initiale de votre application varie en fonction de la nature de votre environnement existant, des besoins du marché, des considérations de coûts, etc. Ce qui ne change pas, c'est l'introduction de ces services partagés.

Toute migration SaaS doit prendre en charge ces services partagés fondamentaux pour permettre à votre entreprise de fonctionner dans un modèle SaaS. Toutes les variantes de l'architecture de l'application ont besoin d'une identité SaaS, par exemple. Vous aurez besoin d'opérations tenant compte des locataires pour gérer et surveiller votre solution SaaS.

La mise en place de ces services partagés dès le début de votre migration vous permet de proposer une expérience SaaS à vos clients, même si l'application sous-jacente fonctionne toujours dans un silo complet pour chaque locataire.

L'objectif général est de faire fonctionner votre application dans un modèle SaaS. Vous pouvez ensuite vous concentrer sur la modernisation et le perfectionnement de votre application. Cette approche vous permet également de faire évoluer les autres secteurs de votre entreprise (marketing, ventes, support, etc.) à un rythme plus rapide. Plus important encore, cela vous permet de commencer à engager et à recueillir les commentaires des clients qui peuvent être utilisés pour façonner la modernisation continue de votre environnement.

Il est important de noter que les services partagés que vous mettez en place peuvent ne pas inclure toutes les fonctionnalités ou tous les mécanismes dont vous aurez finalement besoin. L'objectif principal est de créer les mécanismes partagés nécessaires au début de votre migration. Cela vous permet de vous concentrer sur les éléments du système qui sont essentiels à l'évolution de votre architecture applicative et à votre évolution opérationnelle.

identité SaaS

Le SaaS ajoute de nouvelles considérations au modèle d'identité de votre application. Lorsque chaque utilisateur est authentifié, il doit être connecté à un contexte de locataire spécifique. Ce contexte de locataire fournit des informations essentielles sur votre locataire qui sont utilisées dans l'ensemble de votre environnement SaaS.

Ce lien entre les locataires et les utilisateurs est souvent appelé identité SaaS de votre application. Lorsque chaque utilisateur s'authentifie, votre fournisseur d'identité émet généralement un jeton qui inclut à la fois l'identité de l'utilisateur et l'identité du locataire.

La connexion des locataires aux utilisateurs représente un aspect fondamental de votre architecture SaaS qui a de nombreuses implications en aval. Le jeton issu de ce processus d'identité est transmis aux microservices de votre application et est utilisé pour créer des journaux tenant compte des locataires, enregistrer des métriques, facturer des compteurs, renforcer l'isolement des locataires, etc.

Il est essentiel d'éviter les scénarios qui reposent sur des mécanismes distincts et autonomes qui mettent en correspondance les utilisateurs et les locataires. Cela peut compromettre la sécurité de votre système et créer souvent des goulots d'étranglement dans votre architecture.

Isolement des locataires

Plus vous ferez passer les clients à un modèle multi-tenant, plus ils seront préoccupés par la possibilité qu'un locataire accède aux ressources d'un autre locataire. Les systèmes SaaS incluent des mécanismes explicites qui garantissent que les ressources de chaque locataire, même si elles s'exécutent sur une infrastructure partagée, sont isolées.

C'est ce que nous appelons l'isolement des locataires. L'idée qui sous-tend l'isolation des locataires est que votre architecture SaaS introduit des structures qui contrôlent étroitement l'accès aux ressources et bloquent toute tentative d'accès aux ressources d'un autre locataire.

Notez que l'isolement des locataires est distinct des mécanismes de sécurité généraux. Votre système prendra en charge l'authentification et l'autorisation ; toutefois, le fait qu'un utilisateur locataire soit authentifié ne signifie pas que votre système est parvenu à s'isoler. L'isolation est appliquée séparément de l'authentification et de l'autorisation de base qui peuvent faire partie de votre application.

Pour mieux comprendre cela, imaginez que vous ayez utilisé un fournisseur d'identité pour authentifier l'accès à votre système SaaS. Le jeton issu de cette expérience d'authentification peut également inclure des informations sur le rôle d'un utilisateur qui peuvent être utilisées pour contrôler l'accès de cet utilisateur à une fonctionnalité d'application spécifique. Ces constructions assurent la sécurité, mais pas l'isolation. En fait, un utilisateur peut être authentifié et autorisé, tout en continuant à accéder aux ressources d'un autre locataire. Rien en matière d'authentification et d'autorisation ne bloquera nécessairement cet accès.

L'isolation des locataires se concentre exclusivement sur l'utilisation du contexte des locataires pour limiter l'accès aux ressources. Il évalue le contexte du locataire actuel et utilise ce contexte pour déterminer quelles ressources sont accessibles pour ce locataire. Il applique cette isolation à tous les utilisateurs de ce tenant.

Cela devient de plus en plus difficile à mesure que nous examinons comment l'isolation des locataires est réalisée dans les différents modèles d'architecture SaaS. Dans certains cas, l'isolation peut être réalisée en consacrant des piles entières de ressources à un locataire, alors que des politiques réseau (ou plus grossières) empêchent l'accès entre locataires. Dans d'autres scénarios, vous pouvez avoir regroupé des ressources (éléments d'un tableau [Amazon DynamoDB](#)) qui nécessitent des politiques plus précises pour contrôler l'accès aux ressources.

Toute tentative d'accès à une ressource locataire doit être limitée aux ressources qui appartiennent à ce locataire. Il incombe aux développeurs et aux architectes SaaS de déterminer quelle combinaison d'outils et de technologies répondra aux exigences d'isolation de votre application spécifique.

Partitionnement de données

Le partitionnement des données est utilisé pour décrire les différentes stratégies utilisées pour représenter les données dans un environnement mutualisé. Ce terme est largement utilisé pour couvrir une gamme d'approches et de modèles différents qui peuvent être utilisés pour associer différents ensembles de données à des locataires individuels.

Notez qu'il est souvent tentant de considérer le partitionnement des données et l'isolation des locataires comme interchangeable. Ces deux concepts ne sont pas censés être équivalents. Lorsque nous parlons de partitionnement des données, nous parlons de la manière dont les données des locataires sont stockées pour chaque locataire. Le partitionnement des données ne garantit pas leur isolement. L'isolation doit toujours être appliquée séparément pour garantir qu'un locataire ne puisse pas accéder aux ressources d'un autre locataire.

Chaque technologie AWS de stockage apporte son propre ensemble de considérations à la stratégie de partitionnement des données. Par exemple, l'isolation de données dans Amazon DynamoDB sera très différente de l'isolation de données avec [Amazon Relational Database Service \(Amazon RDS\)](#).

En général, lorsque vous pensez au partitionnement des données, vous commencez par vous demander si les données seront cloisonnées ou regroupées. Dans un modèle cloisonné, vous disposez d'une structure de stockage distincte pour chaque locataire, sans données mélangées. Pour le partitionnement groupé, les données sont mélangées et partitionnées sur la base d'un identifiant de locataire qui détermine quelles données sont associées à chaque locataire.

Par exemple, avec Amazon DynamoDB, un modèle cloisonné utilise une table distincte pour chaque locataire. Le regroupement des données dans Amazon DynamoDB s'effectue en stockant l'identifiant du locataire dans la clé de partition de chaque table Amazon DynamoDB qui gère les données de tous les locataires.

Vous pouvez imaginer comment cela peut varier en fonction de la gamme de AWS services, chacun introduisant ses propres concepts, ce qui peut nécessiter une approche différente pour réaliser des modèles de stockage en silo et groupé pour chaque service.

Bien que le partitionnement des données et l'isolation des locataires soient des sujets distincts, les stratégies de partitionnement des données que vous choisissez sont susceptibles d'être influencées par le modèle d'isolation de vos données. Par exemple, vous pouvez cloisonner une partie du stockage car cette approche correspond le mieux aux exigences de votre domaine ou de vos clients. Vous pouvez également choisir le silo car le modèle de pool peut ne pas vous permettre de renforcer l'isolation avec le niveau de granularité requis par votre solution.

Un voisin bruyant peut également avoir une incidence sur votre approche de l'isolement. Certaines charges de travail ou certains cas d'utilisation de votre application peuvent devoir être séparés pour limiter les impacts par rapport aux autres locataires ou pour respecter les accords de niveau de service (SLA).

Comptage, mesures et facturation

Les discussions sur le SaaS ont également tendance à inclure les notions de comptage, de métriques et de facturation. Ces concepts sont souvent regroupés en un seul concept. Cependant, il est important de distinguer les différents rôles que jouent le comptage, les métriques et la facturation dans un environnement SaaS.

Le défi de ces concepts est qu'ils ont souvent des utilisations du même mot qui se chevauchent. Par exemple, nous pouvons parler de compteurs utilisés pour générer votre facture. Dans le même temps, nous pouvons également parler de comptage utilisé pour suivre la consommation interne de ressources qui n'est pas liée à la facturation. Nous parlons également de métriques et de SaaS dans de nombreux contextes qui peuvent être intégrés à cette discussion.

Pour vous aider à résoudre ce problème, associons des concepts spécifiques à chacun de ces termes (sachant qu'il n'y a pas d'absolu ici).

- **Comptage** — Ce concept, bien qu'il comporte de nombreuses définitions, convient le mieux au domaine de la facturation SaaS. L'idée est de mesurer l'activité ou la consommation de ressources des locataires pour collecter les données nécessaires à l'établissement d'une facture.
- **Métriques** : les métriques représentent toutes les données que vous capturez pour analyser les tendances dans vos domaines commerciaux, opérationnels et technologiques. Ces données sont utilisées dans de nombreux contextes et rôles au sein de l'équipe SaaS.

Cette distinction n'est pas essentielle, mais elle permet de simplifier la façon dont nous envisageons le rôle des mesures et des métriques dans un environnement SaaS.

Maintenant, si nous associons ces deux concepts à des exemples, vous pouvez envisager d'instrumenter votre application à l'aide d'événements de mesure spécifiques utilisés pour obtenir les données nécessaires à l'établissement d'une facture. Il peut s'agir du nombre de demandes, du nombre d'utilisateurs actifs ou d'un agrégat de consommation (demandes, processeur, mémoire) correspondant à une unité qui convient à vos clients.

Dans votre environnement SaaS, vous publierez ces événements de facturation depuis votre application et ils seront ingérés et appliqués par la structure de facturation utilisée par votre système SaaS. Il peut s'agir d'un système de facturation tiers ou d'un système personnalisé.

En revanche, l'objectif qui sous-tend les métriques est de capturer les actions, les activités, les habitudes de consommation, etc. qui sont essentielles à l'évaluation de l'empreinte sanitaire et

opérationnelle imposée à votre système par les différents locataires. Les indicateurs que vous publiez et agrégez ici sont davantage dictés par les besoins de différentes personnes (équipes opérationnelles, propriétaires de produits, architectes, etc.). Ici, ces données métriques sont publiées et agrégées dans des outils d'analyse qui permettent à ces différents utilisateurs de créer des vues de l'activité du système qui analysent les aspects du système qui correspondent le mieux à leur personnalité. Le propriétaire d'un produit peut souhaiter comprendre comment les différents locataires utilisent les fonctionnalités. Un architecte peut avoir besoin de vues qui l'aident à comprendre comment les locataires consomment les ressources de l'infrastructure, etc.

SaaS B2B et B2C

Les offres SaaS sont créées pour les marchés B2B et B2C. Bien que les marchés et les clients aient certainement des dynamiques différentes, les principes généraux du SaaS ne modifient pas d'une manière ou d'une autre chacun de ces marchés.

Si vous examinez l'onboarding, par exemple, les clients B2B et B2C peuvent avoir des expériences d'intégration différentes. Il est vrai que les systèmes B2C peuvent se concentrer davantage sur un flux d'intégration en libre-service (bien que le système B2B puisse également le prendre en charge).

Bien qu'il puisse y avoir des différences dans la manière dont l'intégration est proposée aux clients, les valeurs sous-jacentes fondamentales de l'intégration sont pour la plupart les mêmes. Même si votre solution B2B repose sur un processus d'intégration interne, nous nous attendons à ce que ce processus soit aussi fluide et automatisé que possible. Être B2B ne signifie pas que nous modifions nos attentes au fil du temps afin de créer de la valeur pour nos clients.

Conclusion

L'objectif de ce document est de décrire les concepts fondamentaux de l'architecture SaaS, en fournissant des précisions sur les modèles et la terminologie utilisés pour caractériser les modèles et les stratégies SaaS. L'espoir est que cela permettra aux organisations d'avoir une vision plus claire du paysage global du SaaS.

La plupart des sujets abordés ici ont porté sur ce que signifie être SaaS, en mettant particulièrement l'accent sur la création d'un environnement qui vous permet de gérer et d'exploiter tous vos clients SaaS grâce à une expérience unifiée. Cela correspond à l'idée de base selon laquelle le SaaS est avant tout un modèle commercial. L'architecture SaaS que vous créez vise à promouvoir ces objectifs commerciaux fondamentaux.

Suggestions de lecture

Un certain nombre de ressources abordent plus en détail les modèles d'architecture SaaS conformes aux modèles décrits ici.

Pour plus d'informations, voir :

- [Stratégies d'isolation des locataires SaaS](#) (AWS Livre blanc)
- [Stratégies de stockage SaaS](#) (AWS Livre blanc)
- [Objectif SaaS Well-Architected \(livre blanc\)](#) AWS

Collaborateurs

Les personnes et organisations suivantes ont contribué à l'élaboration du présent document :

- Tod Golding, architecte principal des solutions partenaires, SaaS Factory AWS

Révisions du document

Pour recevoir une notification concernant des mises à jour de ce livre blanc, abonnez-vous au flux RSS.

Modification	Description	Date
Publication initiale	Livre blanc publié.	3 août 2022

Avis

Les clients sont tenus de faire leur propre évaluation indépendante des informations contenues dans ce document. Ce document : (a) est fourni à titre informatif uniquement, (b) représente les offres de AWS produits et les pratiques actuelles, qui sont susceptibles d'être modifiées sans préavis, et (c) ne crée aucun engagement ni aucune garantie de la part AWS de ses filiales, fournisseurs ou concédants de licence. AWS les produits ou services sont fournis « tels quels » sans garantie, représentation ou condition d'aucune sorte, qu'elle soit expresse ou implicite. Les responsabilités et obligations AWS de ses clients sont régies par AWS des accords, et le présent document ne fait partie d'aucun accord entre AWS et ses clients et ne le modifie pas.

© 2023 Amazon Web Services, Inc. ou ses filiales. Tous droits réservés.

Glossaire AWS

Pour connaître la terminologie la plus récente d'AWS, consultez le [Glossaire AWS](#) dans la Référence Glossaire AWS.

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.