Implémentation de l'infrastructure en tant que produit (iAP) sur AWS

AWS Conseils prescriptifs

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

AWS Conseils prescriptifs: Implémentation de l'infrastructure en tant que produit (iAP) sur AWS

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
Pourquoi gérer l'infrastructure en tant que produits ? 1
Résultats commerciaux ciblés 1
Utilisation AWS Service Catalog pour gérer iAP 3
Support pour la modularité et la réutilisation du code 4
Options de programmation pour définir les produits dans Service Catalog 5
CloudFormation écriture de scripts 6
Approche programmatique avec le AWS CDK 6
Intégration aux processus et flux de travail de provisionnement externes 8
Spécifications relatives à l'approvisionnement des produits 8
DevSecOps support du cycle de vie 9
Réutilisation personnalisée et provisionnement spécifique au compte 9
Définition et gestion des ressources des produits Service Catalog en tant qu'applications 9
Gestion des stocks 10
Utilisation d' AWS Service Catalog outils 12
Service Catalog Puppet 12
Support pour le provisionnement des flux de travail 13
Modes de provisionnement 13
mise en cache 15
DevSecOps support du cycle de vie 15
Maturité, exhaustivité et support 16
Service Catalog Factory 17
Résumé et prochaines étapes 18
Ressources 19
Historique du document 20
..... xxi

Implémentation de l'iAP sur AWS

Kirsten Kissmeyer, Amazon Web Services ()AWS

Janvier 2023 (historique du document)

Ce guide explore les approches permettant de gérer votre AWS infrastructure en tant que produit (iAP). L'iAP fournit un niveau d'abstraction et de contrôle supérieur à celui de l'infrastructure en tant que code (IaC), mais utilise les méthodes IaC pour atteindre ses objectifs. Le guide explore également les Services AWS outils de gestion de l'iAP et met en évidence la manière dont chaque outil peut vous aider à atteindre vos objectifs en matière de gestion de votre infrastructure. Les informations contenues dans ce guide sont basées sur les enseignements tirés d'une initiative d' AWS Service Catalog habilitation pour une très grande entreprise du secteur financier.

Ce guide est destiné aux utilisateurs qui souhaitent développer des services AWS Cloud d'infrastructure fonctionnels pouvant être facilement alloués et autorisés selon les besoins des différents utilisateurs de l'organisation, des unités commerciales et des tiers.

Pourquoi gérer l'infrastructure en tant que produits ?

L'avantage de gérer les ressources de votre infrastructure sous forme de produits est que vous pouvez regrouper les fonctionnalités destinées aux consommateurs sous la forme d'un ensemble de ressources dotées de définitions et de configurations standardisées. Les produits constituent un moyen pratique pour une organisation de gérer et de contrôler la manière dont les AWS capacités sont allouées et consommées. Un produit peut être limité aux seules unités organisationnelles désignées (OUs) ou aux personnes ayant besoin de ces capacités fonctionnelles. Un produit peut également être limité à un produit spécifique Régions AWS .

Un modèle de provisionnement de produit vous permet également d'encapsuler et de mettre à jour la définition d'un produit à partir d'un emplacement central. Vous pouvez ensuite distribuer les mises à jour du produit de manière ponctuelle ou planifiée, au fur et à mesure que sa mise en œuvre évolue au fil du temps.

Résultats commerciaux ciblés

Organisations recherchent toujours de meilleurs moyens de gérer et de provisionner leur AWS infrastructure. Vos objectifs peuvent inclure :

- Atteindre un haut degré d'agilité, de fiabilité, de tolérance aux pannes et de contrôle centralisé, où les points de configuration uniques répondent à l'évolution des normes internes et externes.
- Un mécanisme facile à utiliser ou à bouton-poussoir permettant de distribuer l'infrastructure de manière centralisée, tout en permettant un accès en libre-service en cas de besoin pour des équipes ou des individus spécifiques.
- La capacité de fournir une AWS infrastructure et des services au personnel interne, aux comptes clients et aux comptes des unités d'organisation partenaires. Vous souhaiterez peut-être également contrôler quelles OUs organisations ont accès à des composants d'infrastructure spécifiques dans des régions spécifiques.
- Si vous utilisez des outils tiers (tels que ServiceNow) ou des outils personnalisés pour gérer les demandes d'accès et de provisionnement des actifs et de l'infrastructure de votre entreprise, l'intégration entre votre AWS infrastructure et ces outils est facilitée.
- La capacité de fournir une AWS infrastructure à des dizaines, voire à des centaines de comptes cibles en même temps.
- Support pour le provisionnement de plusieurs AWS ressources afin de fournir une fonctionnalité unique.
- La possibilité de créer de nouveaux comptes avec l'infrastructure requise dans un calendrier serré.
- Accès à un inventaire de l'infrastructure que vous avez mise en service et possibilité de mettre à jour ou de supprimer des composants de l'infrastructure.
- Approches et technologies qui rendent le processus de provisionnement et de maintenance plus facile, plus rapide, plus sûr et plus fiable.

Utilisation AWS Service Catalog pour gérer iAP

AWS fournit un service appelé AWS Service Catalogqui prend en charge la gestion et le provisionnement de AWS l'infrastructure en tant que produit. Vous pouvez utiliser Service Catalog pour définir rapidement l'infrastructure que vous devez fournir sous la forme d'un ensemble de produits, accorder l'autorisation d'utiliser ces produits aux parties souhaitées et mettre en œuvre les modèles de provisionnement et de mise à jour requis pour les produits individuels.

Service Catalog est soutenu par AWS CloudFormation. Les portefeuilles, les produits et les modèles de provisionnement du Service Catalog sont gérés comme des CloudFormation piles. Vous pouvez définir ces piles de quatre manières :

- En utilisant des CloudFormation modèles standard.
- En utilisant la bibliothèque Cloud Development Kit AWS (AWS CDK)et la bibliothèque Service Catalog Construct avec le langage de programmation pris en charge que vous préférez.
- En utilisant un framework fourni par un outil tiers pour générer les définitions de CloudFormation pile à partir de métadonnées déclaratives décrivant les piles.
- En utilisant l'API Service Catalog. Cette API fournit des méthodes pour tout sauf pour créer le produit. Vous pouvez ajouter des produits à des portefeuilles, supprimer des produits des portefeuilles, étiqueter des produits et des portefeuilles, définir des actions administratives et opérationnelles de service des produits, et parcourir et rechercher des définitions de portefeuilles et de produits.

À la base, un produit Service Catalog est un ensemble d'une ou de plusieurs AWS ressources configurées pour fournir une fonctionnalité collective personnalisable (par le biais du paramétrage). Par exemple, vous pouvez définir un produit Service Catalog pour approvisionner un compartiment Amazon Simple Storage Service (Amazon S3) privé dans un compte cible. Le compartiment S3 est un produit qui peut comporter des paramètres d'entrée tels que le nom du compartiment, une plage d'adresses Internet à partir de laquelle l'accès est autorisé, un ensemble d'utilisateurs autorisés à accéder au compartiment, une politique de hiérarchisation du cycle de vie ou une spécification de version du compartiment. Vous pouvez également définir un rôle Gestion des identités et des accès AWS (IAM) pour donner accès au bucket dans le cadre du produit.

Vous pouvez ajouter un produit Service Catalog à un ou plusieurs portefeuilles. Un portefeuille de Services Catalog est un ensemble de produits regroupés, généralement parce qu'ils répondent à un

objectif similaire (par exemple, analyse, développement, services d'accès client, services d'accès aux partenaires, etc.).

Vous autorisez un utilisateur, un groupe ou un rôle à accéder à l'approvisionnement d'un produit au niveau du portefeuille. Pour le provisionnement, les produits sont associés soit à un rôle IAM de lancement (pour lancer le produit en libre-service à toute personne susceptible d'assumer ce rôle), soit à un stack set qui définit un ou plusieurs comptes auxquels le produit peut être approvisionné. Pour utiliser un stack set, vous devez définir un rôle d'administrateur de Service Catalog dans le compte hub de Service Catalog et un rôle d'exécution du provisionnement des produits Service Catalog dans chaque compte cible du stack set.

Les sections suivantes décrivent plus en détail les fonctionnalités iAP de Service Catalog.

Rubriques

- Support pour la modularité et la réutilisation du code
- Options de programmation pour définir les produits dans Service Catalog
- Intégration aux processus et flux de travail de provisionnement externes
- Spécifications relatives à l'approvisionnement des produits
- DevSecOps support du cycle de vie
- Réutilisation personnalisée et provisionnement spécifique au compte
- Définition et gestion des ressources des produits Service Catalog en tant qu'applications
- Gestion des stocks

Support pour la modularité et la réutilisation du code

Vous pouvez assembler un produit à partir de nombreuses AWS ressources différentes ou même d'autres produits. Idéalement, vous définissez les ressources de manière modulaire afin de pouvoir les réutiliser dans plusieurs produits. La réutilisation au niveau des ressources vous permet d'apporter les modifications futures en un seul endroit plutôt que sur tous les produits utilisant ce type de ressource.

Service Catalog fournit une fonctionnalité appelée chaînage pour favoriser la réutilisabilité au niveau du produit. Vous pouvez associer un produit à un ou plusieurs autres produits. Par exemple, vous souhaiterez peut-être associer un produit de compartiment de journalisation S3 à un produit de surveillance de niveau supérieur. Bien que le chaînage favorise la modularité, il impose certaines

complexités opérationnelles car vous devez gérer les dépendances. Service Catalog ne gère pas automatiquement la gestion des versions entre les produits en chaîne. Il ne peut donc pas garantir que les modifications apportées à un produit ne nuisent pas aux autres produits qui en dépendent. Utilisez le chaînage avec prudence et développez vos propres mécanismes pour garantir le versionnement et maintenir les dépendances.

Service Catalog l'utilise CloudFormation nativement pour déployer un modèle de provisionnement de produit sous forme de stack. CloudFormation Service Catalog impose toutefois certaines limites au CloudFormation déploiement de la pile de produits. En particulier, le provisionnement du Service Catalog ne prend pas en charge la CloudFormation `include` macro permettant d'insérer des segments de script réutilisables ou de référencer des CloudFormation scripts imbriqués (ou des piles) à plusieurs niveaux. Ces restrictions du Service Catalog limitent la possibilité de définir des produits à partir de CloudFormation modèles ou de composants réutilisables, ce qui est une bonne pratique standard lorsque vous définissez des piles de manière native dans. CloudFormation

> **Note**
>
> Service Catalog vous permet de définir avec succès des produits à l'aide de modèles de provisionnement utilisant ces CloudFormation structures. Toutefois, vous risquez de rencontrer des erreurs de mise en service si vous utilisez la `include` macro ou si vous imbriquez plusieurs niveaux de scripts dans un modèle de Service Catalog CloudFormation .

Ces restrictions peuvent rendre difficile l'implémentation de produits modulaires et réutilisables dans Service Catalog. Si la modularité est une exigence, vous pouvez envisager d'utiliser le AWS CDK pour implémenter vos produits et leurs modèles de provisionnement, ou d'utiliser les flux de travail et le moteur de provisionnement dans le projet Labs Service AWS Catalog Tools. Les deux alternatives sont décrites plus loin dans ce guide.

Options de programmation pour définir les produits dans Service Catalog

Deux options de programmation permettant d'utiliser Service Catalog pour provisionner AWS l'infrastructure sont les CloudFormation modèles ou le AWS CDK. Il n'existe actuellement aucun mécanisme déclaratif ou sans code pour définir un produit Service Catalog.

CloudFormation écriture de scripts

CloudFormation est un langage de script natif IaC éprouvé pour le provisionnement AWS de l'infrastructure. Vous pouvez développer un CloudFormation script dans Console de gestion AWS ou à l'aide d'un outil de développement tel que Visual Studio Code (ou un simple éditeur de texte) et le AWS Command Line Interface (AWS CLI).

Pour plus d'informations, consultez la documentation CloudFormation. Pour plus d'informations sur l'utilisation d'un CloudFormation modèle pour spécifier un produit Service Catalog, consultez la AWS::ServiceCatalog::CloudFormationProduct ressource dans la CloudFormation documentation.

Approche programmatique avec le AWS CDK

AWS CDK Il fournit un cadre de programmation orienté objet élégant et puissant pour définir et maintenir l' AWS infrastructure en utilisant une sélection de langages de programmation. Vous pouvez utiliser le AWS CDK pour développer des personnalisations et des extensions précises et orientées objet pour le cadre de classes. AWS AWS CDK Il s'adresse aux utilisateurs qui souhaitent personnaliser leur infrastructure Services AWS pour répondre à des besoins d'infrastructure plus sophistiqués et qui possèdent les compétences et l'expérience de programmation requises.

Pour implémenter des solutions Service Catalog à l'aide de AWS CDK, vous utilisez les classes Service Catalog intégrées pour définir vos produits et portefeuilles. Ces classes sont fournies par le module AWS CDK aws-cdk-lib.aws_servicecatalog.

Vous pouvez utiliser le AWS CDK pour implémenter des produits de nombreuses manières. Pour éviter d'avoir à écrire le modèle de provisionnement d'un produit CloudFormation et pour garantir la réutilisabilité, nous vous recommandons d'utiliser la AWS CDK ProductStackclasse pour représenter le modèle de provisionnement. Une `ProductStack` instance est une AWS CDK pile à laquelle vous ajoutez des ressources par programmation. Par exemple, vous pouvez ajouter un compartiment S3, des rôles IAM ou un CloudWatch journal Amazon. Lorsque vous ajoutez l'`ProductStack`instance à une `servicecatalog.CloudFormationProduct` instance définie en tant que modèle de provisionnement en appelant`servicecatalog.CloudFormationTemplate.fromProductStack (<ProductStack instance>)`, le modèle génère AWS CDK automatiquement le CloudFormation modèle.

Voici un exemple d'`ProductStack`implémentation Java pour un produit Amazon S3.

```
import * as s3 from 'aws-cdk-lib/aws-s3';
```

```
import * as cdk from 'aws-cdk-lib';

class S3BucketProduct extends servicecatalog.ProductStack {
  constructor(scope: Construct, id: string) {
    super(scope, id);

    new s3.Bucket(this, 'BucketProduct');
  }
}

const product = new servicecatalog.CloudFormationProduct(this, 'Product', {
  productName: "My Product",
  owner: "Product Owner",
  productVersions: [
    {
      productVersionName: "v1",
      cloudFormationTemplate:
 servicecatalog.CloudFormationTemplate.fromProductStack(new S3BucketProduct(this,
 'S3BucketProduct')),
    },
  ],
});
```

AWS CDK Il fournit des pipelines intégrés d'intégration continue et de déploiement continu (CI/CD). Vous pouvez personnaliser ces pipelines intégrés et ces processus de cycle de vie du développement logiciel (SDLC) pour répondre à vos propres normes et objectifs en matière de processus.

AWS CDK Les classes personnalisées peuvent hériter d'autres classes pour fournir des fonctions spécialisées, et une classe peut être composée d'instances d'autres classes. Si vous utilisez des structures de AWS CDK classes partagées pour implémenter plusieurs produits Service Catalog, prenez en compte les implications en termes de version ou de compatibilité, en particulier au sein de plusieurs équipes de développement. Vous devrez vous assurer que les modifications sont rétrocompatibles ou que vous disposez d'un schéma de version qui est suivi afin que les modifications de classe que vous apportez à un produit ne nuisent pas à un autre produit.

Pour plus d’informations, consultez la documentation AWS CDK.

Intégration aux processus et flux de travail de provisionnement externes

Vous pouvez interagir avec les composants du Service Catalog à l'aide du AWS SDK APIs ou du AWS CLI. Vous pouvez utiliser l'API AWS SDK Service Catalog pour gérer les produits Service Catalog à partir de n'importe quel outil capable d'intégrer les appels d'API Service Catalog. L'API couvre tous les aspects de la création et de la gestion du Service Catalog. Par exemple, Terraform prend en charge le lancement (provisionnement) des produits Service Catalog en appelant l'API SDK AWS Service Catalog dans son Launch Wizard. Pour plus d'informations, consultez la section Lancer AWS Service Catalog des produits avec Terraform dans la AWS documentation.

Vous pouvez également appeler les commandes AWS CLI Service Catalog pour effectuer des actions sur Service Catalog. Pour plus d'informations sur les commandes prises en charge, consultez le catalogue de services dans la AWS CLI référence des commandes.

Spécifications relatives à l'approvisionnement des produits

Service Catalog lance le processus de provisionnement sous la forme d' CloudFormation un déploiement en série des ressources spécifiées dans le modèle de CloudFormation provisionnement. (Le modèle peut être créé directement dans CloudFormation ou généré par la AWS CDK `ProductStack` construction.) Le provisionnement des produits Service Catalog est un processus fermé : vous ne pouvez pas le personnaliser pour ajouter des étapes préliminaires ou post-processus, ni le régler. Vous pouvez toutefois modifier le modèle de provisionnement pour ajouter des étapes sous forme de spécifications de CloudFormation ressources. Il peut s'agir AWS Lambda de ressources personnalisées soutenues par Lambda qui effectuent des étapes préliminaires (telles que le démarrage personnalisé pour configurer un hôte bastion utilisé lors du provisionnement) et des étapes ultérieures (telles que le démontage de l'hôte bastion). AWS Step Functions Cette méthode de mise en œuvre des étapes de pré-approvisionnement et de post-provisionnement est soumise aux mêmes restrictions de pile `include` imbriquées que le modèle de provisionnement.

Vous pouvez définir les comptes cibles en tant que comptes individuels, et non en tant qu'unités organisationnelles (OUs). Vous pouvez créer une ressource ou une fonction personnalisée pour contourner cette limitation. La plupart des entreprises fournissent des portefeuilles de produits à des comptes individuels OUs et non à des comptes individuels, car elles automatisent la génération de comptes et ne souhaitent pas gérer les listes de comptes manuellement.

DevSecOps support du cycle de vie

À l'heure actuelle, les produits fournis avec des CloudFormation scripts Service Catalog ne disposent pas d'un support intégré pour les CI/CD processus. Nous vous recommandons de créer un CI/CD processus AWS CodePipeline ou d'autres DevOps outils pour développer, tester et commercialiser un produit dans des environnements de cycle de vie tels que le développement, les tests, le stade et la production.

Le AWS CDK fournit un CI/CD support intégré pour les produits, comme indiqué précédemment dans ce guide.

Réutilisation personnalisée et provisionnement spécifique au compte

Les produits doivent être réutilisables pour autant d'usages personnalisés que possible. Service Catalog favorise la réutilisabilité grâce aux paramètres du produit. Vous pouvez fournir ces paramètres en tant qu'entrée dans un produit au moment de l'approvisionnement.

Vous pouvez également spécifier ces paramètres sous forme de valeurs du magasin de AWS Systems Manager paramètres au niveau du CloudFormation modèle, afin d'appliquer des valeurs spécifiques au compte et à l'unité d'organisation. Il s'agit d'une bonne pratique pour la conception de modèles de CloudFormation provisionnement. La valeur du paramètre nommé dans le compte cible est appliquée lorsque le produit est approvisionné. Par exemple, vous pouvez spécifier un paramètre de sous-réseau en tant que valeur du magasin de paramètres et appliquer ce sous-réseau au moment de la mise en service du produit pour un compte d'unité d'organisation spécifique. Pour plus d'informations sur les valeurs du Parameter Store en tant que paramètres de CloudFormation modèle, consultez la section Utilisation de références dynamiques pour spécifier des valeurs de modèle dans la CloudFormation documentation.

Définition et gestion des ressources des produits Service Catalog en tant qu'applications

AWS Service Catalog AppRegistry fournit des fonctionnalités de recherche, de création de rapports et de gestion centralisées dans les applications. Une AppRegistry application peut inclure une ou plusieurs piles de produits provisionnées ainsi que des CloudFormation piles indépendantes de Service Catalog. Vous pouvez regrouper et afficher toutes vos collections de ressources

d'applications sur celles Comptes AWS que vous définissez comme cibles de déploiement. Ces comptes peuvent être vos comptes de développement, de test et de cycle de vie de production.

Vous pouvez également l'utiliser AppRegistry pour associer des attributs de métadonnées à une application. Vous pouvez attribuer des groupes d'attributs réutilisables contenant des ensembles d'attributs. Vous pouvez ensuite rechercher et agir sur les ressources d'application qui possèdent les attributs donnés en utilisant AppRegistry ou en intégrant des services. Ces services intégrés incluent :

- Gestionnaire d'applications, une fonctionnalité permettant d' AWS Systems Managerétudier et de résoudre les problèmes liés aux AWS ressources dans le contexte de vos applications et de vos clusters
- AWS Resource Access Manager, pour partager des applications et des groupes d'attributs avec les responsables de AWS l'organisation
- AWS services qui fonctionnent avec Groupes de ressources AWS
- AWS Resilience Hubpour la découverte de la structure du produit et l'évaluation de la résilience
- Connecteur AWS Service Managementpour déclarer et configurer des connexions à ServiceNow JIRA et à d'autres outils populaires

Pour plus d'informations sur AppRegistry, consultez les rubriques suivantes :

- AppRegistry Guide de l'administrateur.
- Améliorez la visibilité et la gouvernance des applications grâce à AWS Service Catalog AppRegistry un article de blog. Cet article fournit une vue d'ensemble de la manière de l'utiliser AppRegistry dans le cadre de la gouvernance de l'infrastructure, avec des exemples en ligne de commande pour enregistrer votre infrastructure en tant qu'applications dans. AppRegistry
- Gérez vos applications de manière centralisée à l'aide AppRegistry d'un article de blog d'Application Manager. Cet article fournit une vue d'ensemble avec un didacticiel expliquant comment s'inscrire AppRegistry pour enregistrer une application Web LAMP sur le Console de gestion AWS et la gérer à l'aide du Gestionnaire d'applications.

Gestion des stocks

Service Catalog possède sa propre fonctionnalité interne de gestion des stocks qui enregistre les produits lorsqu'ils sont approvisionnés par le biais du partage de produits et du libre-service.

Toutefois, nous vous recommandons d'utiliser AWS Configou AppRegistryles services associés pour gérer les ressources mises à disposition par le produit. Ces outils fournissent une approche plus complète et intégrée pour gérer les produits Service Catalog que vous avez fournis avec le reste de votre AWS infrastructure. AWS Config vous permet d'inventorier et d'exécuter des actions sur les produits provisionnés sur la console ou à l'aide de l'API du AWS SDK. AppRegistry, qui est intégré à Application Manager, assure également la gestion de l'inventaire des produits approvisionnés par Service Catalog.

Utilisation d' AWS Service Catalog outils

Si vous souhaitez fournir à vos produits iAC des flux de travail de provisionnement personnalisés de manière plus déclarative, vous souhaiterez peut-être augmenter certaines parties de la fonctionnalité Service Catalog. AWS fournit plusieurs outils pour répondre à ces exigences. Deux outils populaires sont fournis dans le projet AWS Labs : Service Catalog Puppet et Service Catalog Factory.

Rubriques

- Service Catalog Puppet
- Service Catalog Factory

Service Catalog Puppet

Service Catalog Puppet est implémenté en Python à l'aide de l'API AWS Boto3. Cet outil propose plusieurs fonctionnalités puissantes pour configurer et approvisionner les produits Service Catalog. Les développeurs peuvent configurer les informations de provisionnement des produits et des portefeuilles de Service Catalog à l'aide de modèles YAML qui servent de manifestes. Les flux de travail de provisionnement de Service Catalog Puppet prennent en charge les produits qui nécessitent des processus de déploiement plus complexes que Service Catalog. Ils permettent également d'optimiser les performances afin de fournir des produits à grande échelle dans des délais serrés.

Service Catalog Puppet accède aux CloudFormation modèles de Service Catalog pour le provisionnement des produits au moment du déploiement. Il appelle CloudFormation directement au déploiement de la pile de modèles de provisionnement pour un produit et contourne les restrictions imposées par le processus de provisionnement des ensembles de piles de Service Catalog. Si le modèle de provisionnement utilise des macros pour inclure d'autres CloudFormation scripts ou utilise des CloudFormation scripts imbriqués, vous devez fournir l'accès à ces scripts dans le compte cible dans la partie d'amorçage du flux de travail de provisionnement.

Pour plus d’informations, consultez:

- Consultez la documentation et le GitHubréférentiel du Service Catalog Puppet.
- Si vous souhaitez utiliser le SDK Service Catalog Puppet pour interagir avec l'outil par programmation afin de lancer le provisionnement de produits et de portefeuilles, consultez la documentation du SDK.

- GitOpsest le mécanisme par défaut pour gérer l'environnement Service Catalog Puppet.

Service Catalog Puppet est assez facile à apprendre pour les développeurs. Il faut être familiarisé avec la mise en œuvre CloudFormation de modèles de provisionnement de produits et de modèles YAML pour implémenter des manifestes. Il existe de bons ateliers pour familiariser les nouveaux développeurs, tels que des ateliers adaptés à leur propre rythme.

Support pour le provisionnement des flux de travail

Service Catalog Puppet utilise le moteur d'orchestration de tâches Python Luigi pour implémenter des flux de travail d'amorçage et de provisionnement. Toutes les étapes de ces flux de travail sont mises en œuvre en tant que tâches de flux de travail Luigi. Pour un aperçu de Luigi et de sa comparaison avec d'autres outils de flux de travail populaires, consultez Airflow vs. Luigi vs. Argo vs. MLFlow vs. KubeFlow sur le blog Data Revenue.

Luigi permet à Service Catalog Puppet de contrôler le nombre de travailleurs associés aux tâches du flux de travail et de contrôler d'autres aspects des flux de travail, pour une meilleure évolutivité et de meilleures performances. Service Catalog Puppet fournit également un mécanisme depends_on pour gérer les dépendances entre les produits et les étapes, et pour orchestrer le provisionnement des produits. Cette fonctionnalité vous aide à implémenter et à gérer de manière opérationnelle des définitions de produits précises et des dépendances complexes.

Modes de provisionnement

Service Catalog Puppet prend en charge trois modes d'exécution : hub, spoke et async. Les trois modes fournissent des produits au sein de portefeuilles déjà définis dans Service Catalog. Ils s'appuient sur le partage des produits Service Catalog avec les comptes cibles et utilisent les rôles d'administrateur et de lancement de Service Catalog pour effectuer le provisionnement sur ces cibles. Service Catalog Puppet exécute les étapes de démarrage au sein de la même organisation en fonction des configurations de rôles fournies dans les fichiers de configuration YAML. L'outil prend également en charge le provisionnement auprès de plusieurs organisations à partir d'un seul compte hub. Dans ce scénario, le démarrage doit être effectué manuellement dans les organisations externes pour permettre à Service Catalog Puppet d'effectuer les actions de provisionnement requises dans les comptes de l'organisation externe.

Dans tous les modes de provisionnement, Service Catalog Puppet implémente le provisionnement des produits directement sans appeler le processus de provisionnement de Service Catalog. Vous pouvez configurer un manifeste de provisionnement pour utiliser les spécifications du rôle et du

compte cible dans une contrainte d'ensemble de piles Service Catalog existante. Service Catalog Puppet utilise ces informations pour effectuer son propre approvisionnement avec les flux de travail Luigi.

Vous pouvez définir des objectifs pour le provisionnement du portefeuille de produits sur la base d'une approche de marquage des comptes, en plus de spécifier directement OUs les comptes. Dans le cadre du provisionnement basé sur les balises de compte, un produit de portefeuille est fourni à tous les comptes qui possèdent toutes les balises du jeu de balises de provisionnement du manifeste spécifié. Par exemple, si vous souhaitez émettre un produit de portefeuille pour tous les comptes de production institutionnels des régions de l'est des États-Unis, vous pouvez spécifier les balises `type:prod partition:us-east`, et`scope:institutional-client`. Vous pouvez également déclarer des exclusions de comptes et d'unités d'organisation afin d'interdire le provisionnement de comptes OUs ou de comptes dotés des balises que vous spécifiez, ou de comptes membres des cibles spécifiées par l'unité d'organisation. Pour plus d'informations sur le balisage des comptes, consultez la documentation des outils Service Catalog.

Mode hub

En mode de provisionnement du hub, tous les flux de travail Luigi pour les comptes Spoke sont gérés à partir du compte central du hub désigné. Le compte du hub assume un rôle IAM qui lui permet d'effectuer des actions dans un compte Spoke, mais la gestion des tâches se fait depuis le compte du hub. Le compte hub attend de manière synchrone que toutes les tâches de provisionnement du compte Spoke soient terminées, avec ou sans succès. Il indique ensuite l'état final. Le mode compte hub est le mode de provisionnement le plus ancien et le plus mature. Cependant, de nombreux utilisateurs sont passés au mode de provisionnement en étoile pour améliorer la simultanéité et la rapidité du provisionnement.

Mode Spoke

En mode Spoke, le compte Hub Service Catalog lance les flux de travail Luigi à exécuter dans les comptes Spoke bootstrap désignés. Le compte hub est averti lorsque les flux de travail parlés sont terminés. Les défaillances d'un compte Spoke se répercutent sur le compte hub. Le compte hub interroge le compte Spoke pour voir s'il est terminé et pour déterminer son statut.

Le mode Spoke est le moins susceptible de nécessiter des augmentations de Service AWS quotas, car presque tout fonctionne dans des comptes Spoke distincts. Le mode Spoke offre également une simultanéité bien supérieure à celle du mode hub tout en conservant le contrôle central. Il peut améliorer la vitesse de provisionnement de 800 % par rapport au mode hub. Le mode Spoke

prend en charge le chaînage des produits grâce aux `DependsOn` relations entre les produits, ce qui garantit qu'un produit dont on dépend a déjà été approvisionné. Un produit qui comprend des produits enchaînés peut également fournir un produit chaîné de composants. Vous pouvez également utiliser des appels de AWS Lambda fonctions spécialisés pour effectuer les étapes requises. Les défauts d'un rayon sont isolés des autres rayons.

Le mode Spoke est utilisé par les entreprises qui possèdent plus de 980 comptes dans un maximum de 7 régions. Ces entreprises sont généralement en mesure de fournir un produit à toutes les régions et à tous les comptes de leur infrastructure en une heure.

> **Note**
>
> Ces résultats peuvent varier en fonction de facteurs tels que l'infrastructure réseau, la charge de travail et les quotas mis en place pour les comptes hub et spoke de AWS l'organisation. Ils dépendent également des ressources du produit mises en service, de leur temps de création inhérent et de leur dépendance à l'égard d'autres ressources.

Mode asynchrone

Le mode asynchrone lance les flux de travail de provisionnement dans les comptes satellites, mais il n'attend ni ne reçoit de réponses de fin de la part des rayons.

mise en cache

Un autre mécanisme utilisé par Service Catalog Puppet pour optimiser la vitesse des flux de travail consiste à mettre en cache les tâches courantes qui représentent les étapes du flux de travail. Lorsqu'une tâche mise en cache est terminée, elle écrit sa sortie dans Amazon Simple Storage Service (Amazon S3). La prochaine fois que la tâche est appelée dans la même session avec les mêmes paramètres, Service Catalog Puppet utilise les valeurs mises en cache au lieu de réexécuter la tâche. Pour plus d'informations, consultez la section Mise en cache dans la documentation de Service Catalog Puppet.

DevSecOps support du cycle de vie

Service Catalog Puppet inclut un support pour la gestion du DevSecOps pipeline. Vous pouvez utiliser les actions des outils Service Catalog (comme illustré dans la présentation de Service Catalog Puppet) pour automatiser les tests et promouvoir les produits sur l'ensemble de vos comptes de AWS

cycle de vie, y compris le compte Canary recommandé. Pour plus d'informations, consultez la section Gestion de vos environnements dans la documentation de Service Catalog Puppet.

Pour garantir que tout problème lié à une modification du produit est détecté avant une utilisation généralisée en production, Service Catalog Puppet nécessite au moins un compte Canary pour le déploiement initial. Après avoir testé la nouvelle version et gagné en confiance, vous pouvez la promouvoir sur des comptes de production autres que Canary. Si vous identifiez des problèmes, vous pouvez annuler la version et la réintroduire une fois les problèmes résolus. Lorsque vous utilisez cette approche, des problèmes de production peuvent survenir si vous publiez une version de Canary présentant un problème avec les comptes de production. Comme autre approche, vous pouvez exécuter des tests de régression complets pour chaque modification de produit avant de passer la modification à la production. Cela introduit une surcharge supplémentaire dans le CI/CD processus, mais permet d'éviter les problèmes de production. Il est de la responsabilité de l' DevSecOps administrateur de déterminer les meilleurs scénarios et approches de publication des fonctionnalités pour ses équipes de développement.

Service Catalog Puppet permet à plusieurs équipes de développer et de tester simultanément le provisionnement des solutions de produits Service Catalog. En tant que bonne pratique, un produit ne doit pas être modifié par plusieurs développeurs en même temps. Au lieu de cela, vous pouvez diviser les produits en composants plus fins pour des modifications distinctes et simultanées.

Service Catalog Puppet permet également d'automatiser les tests grâce à une instruction d'assertion qui fournit des fonctionnalités de test statique et unitaire. Vous pouvez tester les politiques de contrôle des services (SCPs) et les politiques IAM à l'aide de simulateurs de politiques. Il s'agit de end-to-end tests techniques, mais ils peuvent être utilisés dans des environnements de test d'intégration de systèmes (SIT). Pour plus d'informations, consultez les sections Utilisation de simulations de politiques et Application de politiques de contrôle de service dans la documentation Service Catalog Puppet.

Maturité, exhaustivité et support

Bien que Service Catalog Puppet ne soit pas officiellement pris en charge Service AWS, il a été largement adopté. Cet outil a été utilisé par de grandes entreprises au cours des dernières années pour fournir avec succès et de manière centralisée des produits à des centaines de comptes UO dans les délais de provisionnement souhaités. Il s'est avéré capable de fournir des produits tolérants aux pannes à grande échelle. Les utilisateurs qui rencontrent des problèmes avec Service Catalog Puppet peuvent les enregistrer dans le GitHub référentiel afin que les contributeurs à cette solution AWS Labs les résolvent.

Service Catalog Factory

Service Catalog Factory est un autre outil fourni par AWS Labs. Il est similaire à AWS Control Tower : il génère des comptes et appelle Service Catalog (éventuellement via Puppet) pour fournir de l'iAP au sein de ces comptes. Il utilise bon nombre des mêmes mécanismes que Service Catalog Puppet pour implémenter ses fonctionnalités. Service Catalog Factory peut appeler Service Catalog ou Service Catalog Puppet pour fournir l'infrastructure des produits d'un compte. Cet outil prend également en charge la génération de comptes dans plusieurs Régions AWS organisations. Pour plus d'informations, consultez la documentation et le GitHub référentiel de Service Catalog Factory.

Résumé et prochaines étapes

Service Catalog vous aide à provisionner rapidement et de manière fiable votre infrastructure en tant que produit. Vous pouvez utiliser une infrastructure en libre-service à partir d'un catalogue de produits défini ou envoyer des produits vers des comptes cibles désignés dans un hub-and-spoke modèle. Vous pouvez définir les produits Service Catalog et leurs modèles de provisionnement à l'aide CloudFormation de scripts ou du. AWS CDK Dans les deux approches, Service Catalog approvisionne un produit en appelant CloudFormation pour déployer une pile représentant le modèle de provisionnement du produit. La pile est déployée sur tous les comptes cibles désignés au sein d'un ensemble de CloudFormation piles.

L' AWS CDK approche du développement du Service Catalog favorise une plus grande modularisation et une meilleure réutilisation CloudFormation, car vous pouvez définir les produits et leurs ressources en utilisant des classes de produits et de portefeuilles Service Catalog prédéfinies ainsi que des types de ressources prédéfinis. Une AWS CDK implémentation nécessite des compétences de programmation plus avancées. Cela peut être justifié si votre organisation souhaite établir son propre cadre de produits réutilisable avec des configurations de ressources et des comportements standardisés comme base pour le développement de votre AWS infrastructure.

Vous pouvez utiliser Service Catalog Puppet et Service Catalog Factory pour améliorer les fonctionnalités de Service Catalog, principalement pour le provisionnement. Service Catalog Puppet propose des spécifications de provisionnement de produits déclaratives et basées sur des balises ; des flux de travail de provisionnement intégrés, personnalisables, performants et spécialement conçus ; et des pipelines SDLC intégrés, personnalisables, basés sur l'action. CI/CD En utilisant la gestion des dépendances des flux de travail et les fonctionnalités intégrées d'automatisation des tests, vous pouvez enchaîner les produits Service Catalog avec moins de risques opérationnels. Service Catalog Puppet vous aide à fournir des produits à des centaines de comptes de manière fiable dans des délais serrés. Service Catalog Factory est similaire à AWS Control Tower. Il génère des comptes et appelle Service Catalog pour approvisionner iAP au sein de ces comptes.

Les outils Service Catalog et Service Catalog fournissent des fonctionnalités étendues pour vous aider à gérer iAP on AWS. Service Catalog et ces outils font l'objet d'améliorations constantes. Pour les dernières fonctionnalités, consultez les AWS Service Catalog fonctionnalités et le référentiel AWS Service Catalog des produits.

Ressources

Références

- Documentation du Service Catalog
- API Service Catalog
- AppRegistry
- CloudFormation documentation
- CloudFormation ensembles de piles
- AWS::ServiceCatalog::CloudFormationProduct ressource
- Lancez AWS Service Catalog des produits avec Terraform
- Cloud Development Kit AWS (AWS CDK)
- Bibliothèque Service Catalog Construct
- AWS CDK ProductStackclasse
- AWS Organizations documentation

Outils

- Documentation du Service Catalog Puppet
- Référentiel Service Catalog Puppet GitHub
- Documentation de Service Catalog Factory
- GitHubRéférentiel Service Catalog Factory

AWS Modèles de directives prescriptives

- Gérez AWS Service Catalog les produits de manière multiple Comptes AWS et Régions AWS
- Copiez AWS Service Catalog des produits sur différents Comptes AWS et Régions AWS
- Automatisez le déploiement du AWS Service Catalog portefeuille et des produits en utilisant AWS CDK

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
Publication initiale	—	30 janvier 2023

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.