



AWS Lambda の 予約済同時実行数の設定による テナント多層化戦略の実装

Honami Miyake

Solutions Architect

Amazon Web Services Japan G.K.

三宅 穂波 (Miyake Honami)

アマゾンウェブサービスジャパン
技術統括本部 ソリューションアーキテクト

好きな食べ物:

- ラーメン

好きな AWS サービス:

- AWS Lambda

趣味:

- サウナ、サーフィン

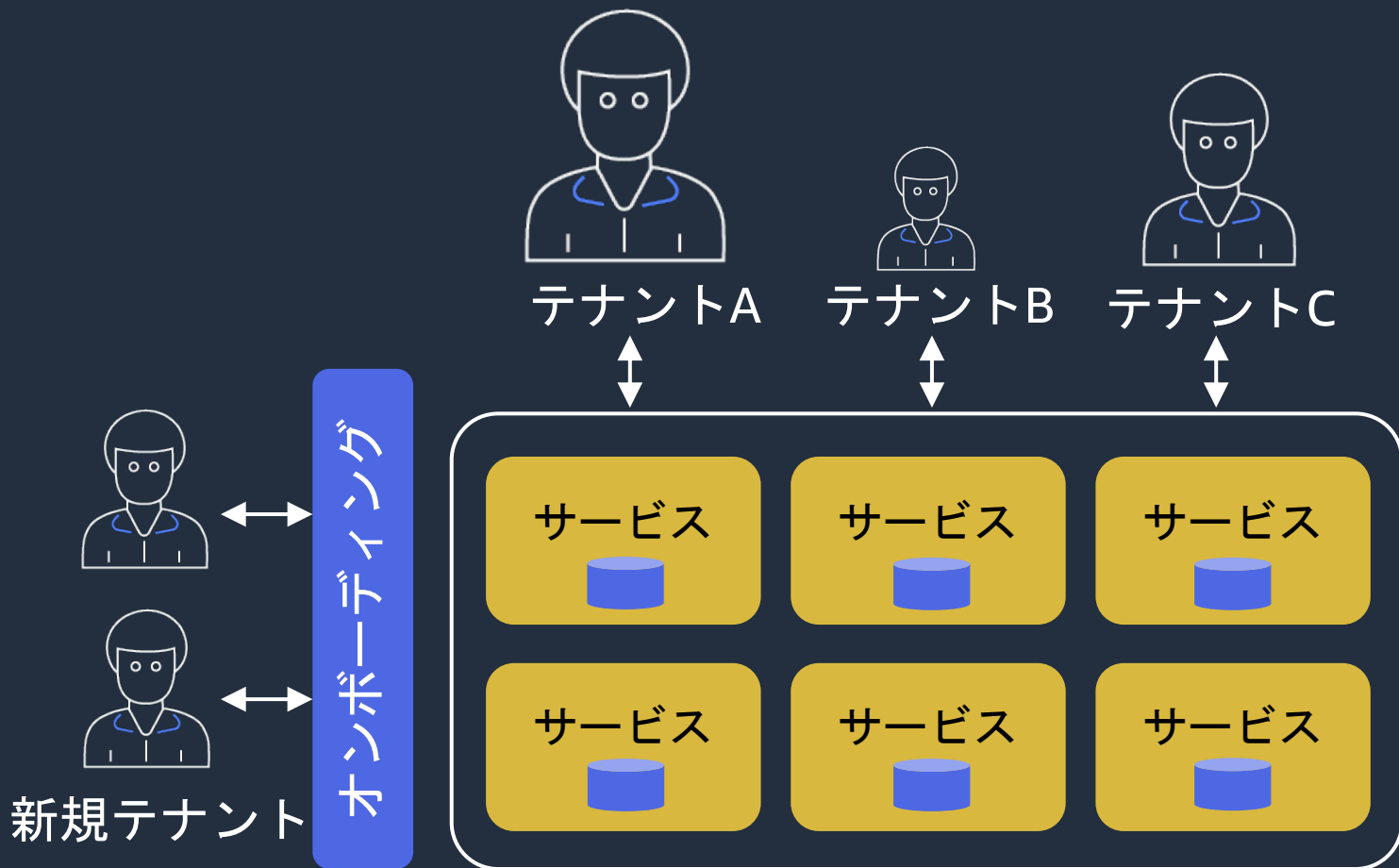


Agenda

- マルチテナントにおける課題
- テナントの多層化戦略
- AWS Lambdaの予約済同時実行数の設定
- SaaSソリューション
- デモ
- まとめ

マルチテナントにおける課題

マルチテナントとは



- 全てのテナントで環境を共有
- **プールモデル**とも呼ばれる

メリット

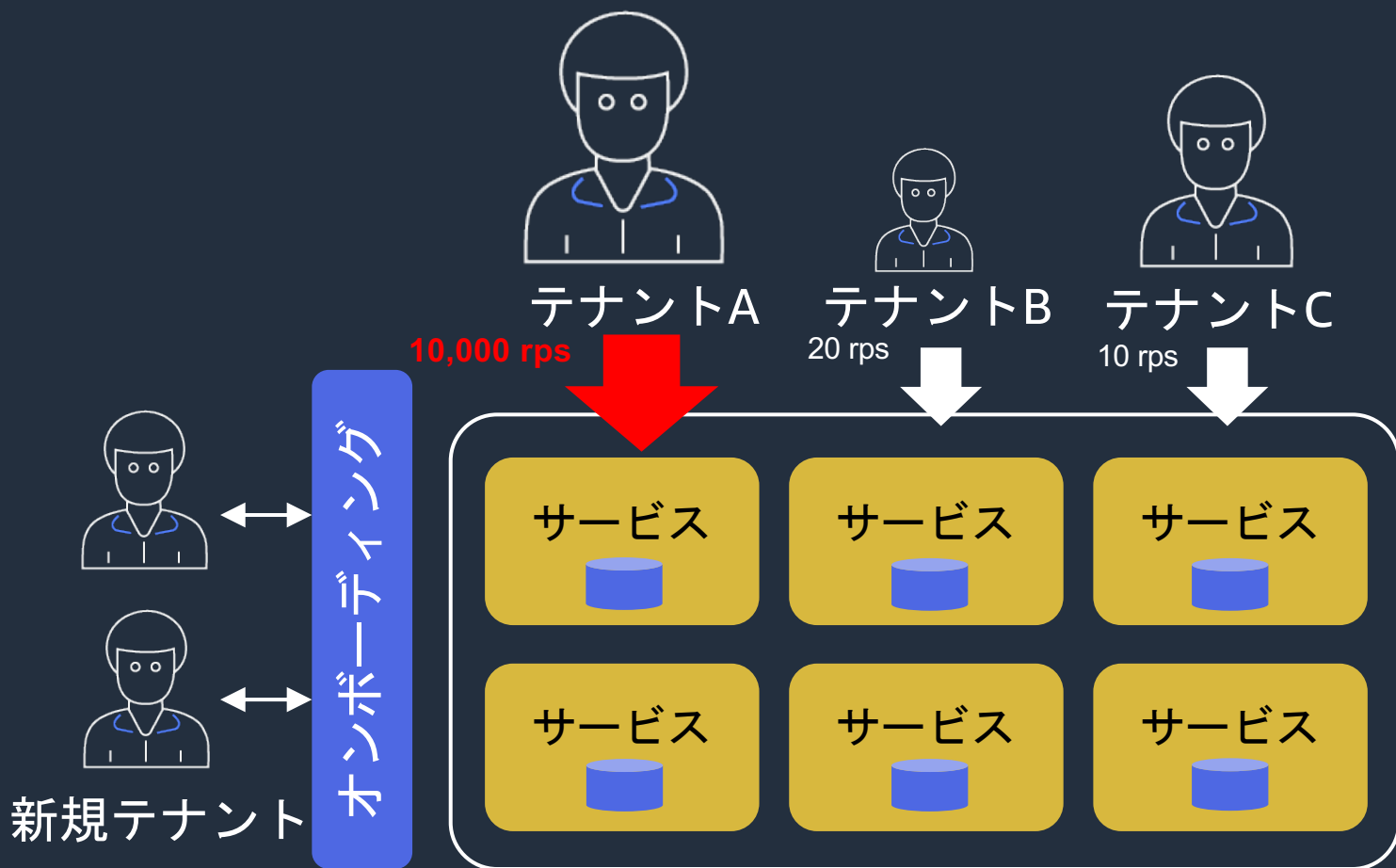
- コスト削減
- アジリティ
- 運用管理の集約
- シンプルなデプロイ
- 横断的なデータ分

析

デメリット

- コンプライアンス
- ノイジーネイバー
- 単一の可用性
- 実装の複雑化

マルチテナントのパフォーマンス観点での課題



- テナントごとに負荷の大きさやピークは異なる
- 一部のテナントが他テナントの分までリソースを使い尽くすリスクがある (**ノイジーネイバー問題**)
- リソース割り当て、スロットリングの仕組みが必要

テナントの多層化戦略

求める要件はテナントによって様々

テナント A : 数千名規模のエンタープライズ企業



- データは他のテナントとは物理的に分離してほしい
- プライベートなネットワークでサービスを利用したい
- 可用性は 99.999 %以上が必要
- コストは惜しまない

SaaS

- セキュリティは最低限でOK
- 可用性は 99 % 以上であればよい
- なるべくコストを抑えたい
- 一度に大量のデータをアップロードできるようにしてほしい



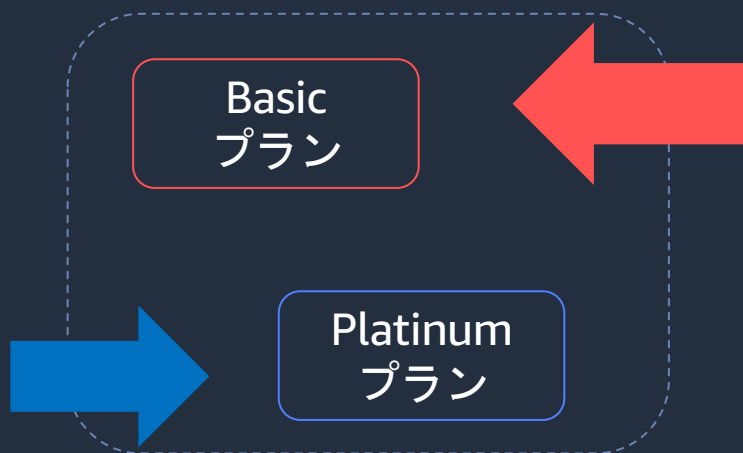
テナント B : 10~15 名のスタートアップ

求める要件はテナントによって様々

テナント A : 数千名規模のエンタープライズ企業



- データは他のテナントとは物理的に分離してほしい
- プライベートなネットワークでサービスを利用したい
- 可用性は 99.999 %以上が必要
- コストは惜しまない



- セキュリティは最低限でOK
- 可用性は 99 % 以上であればよい
- なるべくコストを抑えたい
- 一度に大量のデータをアップロードできるようにしてほしい



テナント B : 10~15 名のスタートアップ

求める要件によって体験を差別化



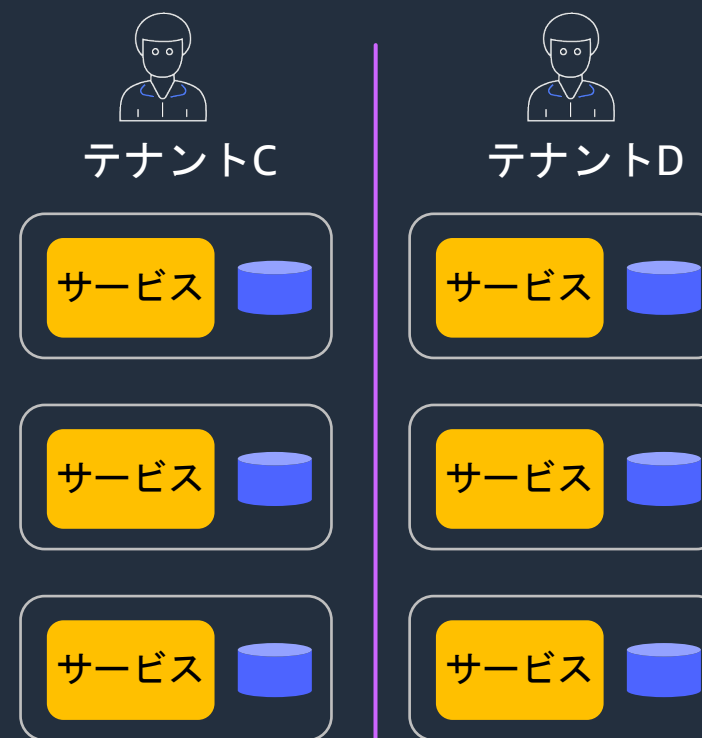
テナントの多層化戦略

Basicプラン



全テナントマルチテナント
(プールモデル)

Platinumプラン



各テナントシングルテナント
(サイロモデル)

AWS Lambdaの 予約済同時実行数の設定

AWS Lambdaとは

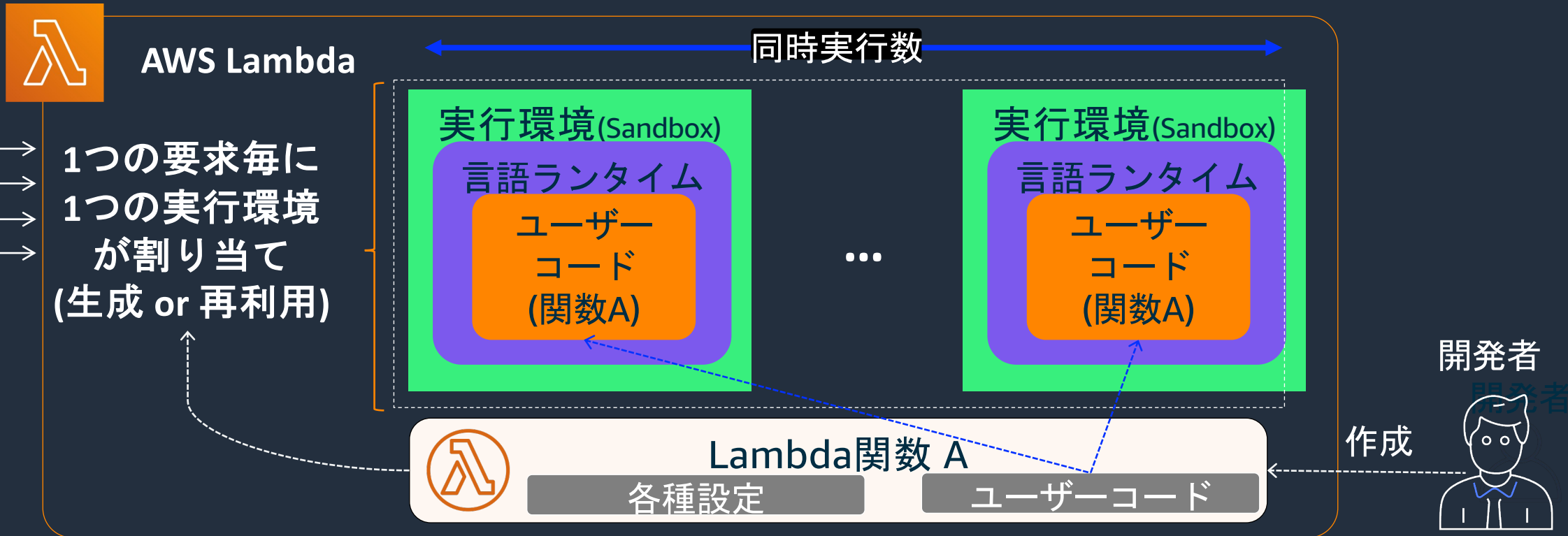
お客様が開発した **関数** を **フルマネージドな環境で実行** するサービス
→ コスト効率と管理性に優れたイベントドリブンのプログラム実行基盤



- 基盤側の可用性 設計済
- 仮想サーバー(OS)管理不要
- 使用量に応じた課金

AWS Lambdaの実行モデル

Lambda関数は、1つの実行要求毎に、1つの実行環境で実行される
同時実行数のクォータは1000(上限緩和可能)



AWS Lambdaの予約済み同時実行数

同時実行の予約機能

他Lambdaの同時実行により、
予期していた同時実行数が出ないことを防ぐ機能

※設定に料金は発生せず、実際に実行された関数に対して料金が発生



reserved = 5

同時実行数を確保
他のLambdaに使われない

AWS Lambdaの予約済み同時実行数



The diagram shows a 2x2 grid of dashed boxes, each labeled "Microservice". Each box contains two Lambda function icons (orange lambda symbol) and one icon representing a different AWS service: a blue database icon, a green server icon, a blue database icon with a star, and a blue database icon with a cylinder.

Basic プラン
予約済み同時実行数設定なし

マルチテナント
(プールモデル)



The diagram shows a 2x2 grid of dashed boxes, each labeled "Microservice". Each box contains two Lambda function icons (orange lambda symbol) and one icon representing a different AWS service: a blue database icon, a green server icon, a blue database icon with a star, and a blue database icon with a cylinder.

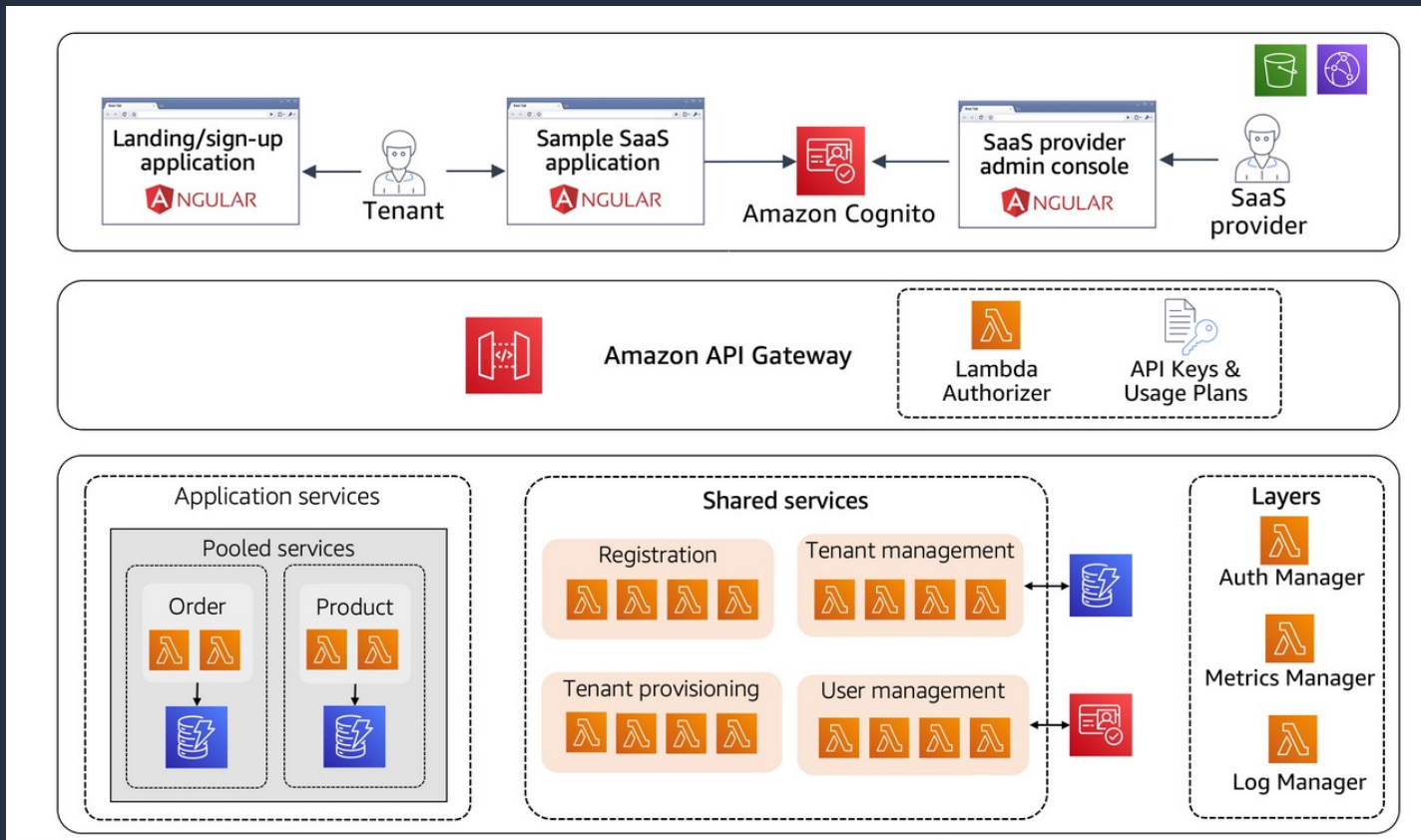
Platinum プラン
予約済み同時実行数=20

シングルテナント
(サイロモデル)

SaaS ソリューション



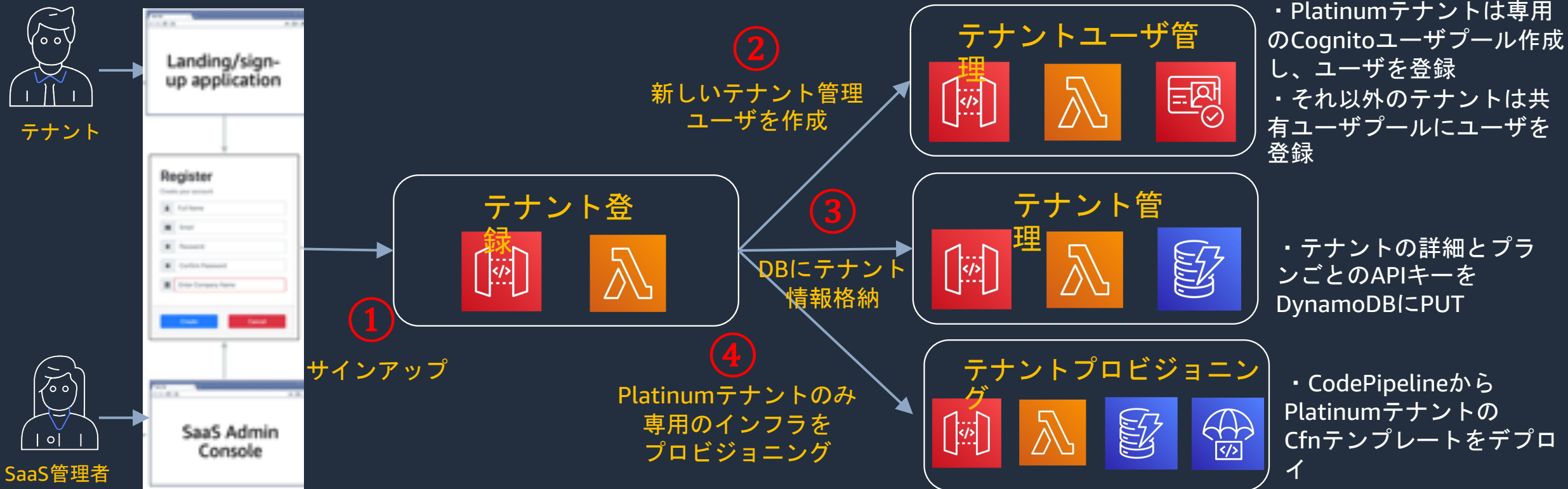
SaaS ソリューション



- eコマースアプリケーション
- 商品や注文を作成・更新

<https://github.com/aws-samples/aws-saas-factory-ref-solution-serverless-saas>

SaaS ソリューション (プロビジョニング)



- ① プランを選択して、サインアップします。
- ② ユーザー管理サービスを呼び出し、新しいテナント管理ユーザーを作成します。
- ③ DBにテナントの詳細を保存します。テナントプランに基づいて、API キーをテナントに関連付けます。
- ④ Platinumテナント専用のインフラをプロビジョニングします。

デモ

まとめ

- マルチテナントにはメリット・デメリットがある
 - デメリットの1つにノイジーネイバー問題
- テナントごとの多様な要件に合うSaaSサービスにするために多層化戦略を検討してみる
- マイクロサービスの場合、AWS Lambdaの予約済み同時実行数の設定によって実現できる
- AWS SaaS FactoryのSaaS ソリューションのご紹介

<https://github.com/aws-samples/aws-saas-factory-ref-solution-serverless-saas>



Thank you!