

サーバレスアーキテクチャ概要

サーバレスとは

サーバレスのイメージ

- ・サーバはサービスを提供するインフラ
- ・サーバレスはインフラがないこと(インフラを持たないこと)
- ・そもそもなぜインフラを持たない状態を選択するのか？

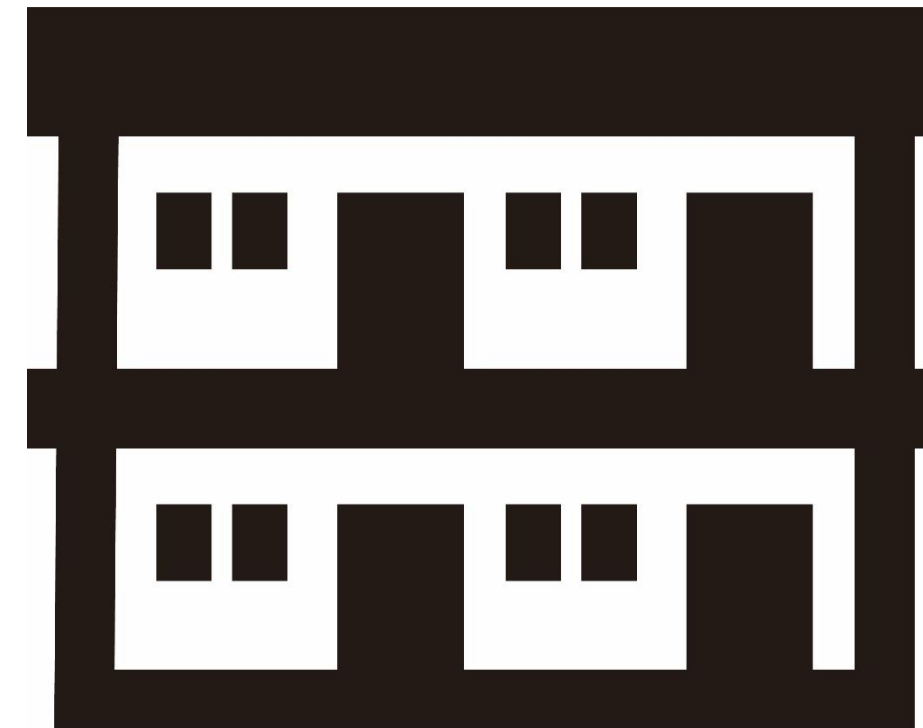
サーバレスとは

サーバレスのイメージ

- ・インフラの代表例は「家」
- ・みなさんのお家は「持ち家」でしょうか、「賃貸」でしょうか
- ・なぜそれを選びましたか？



持ち家



賃貸

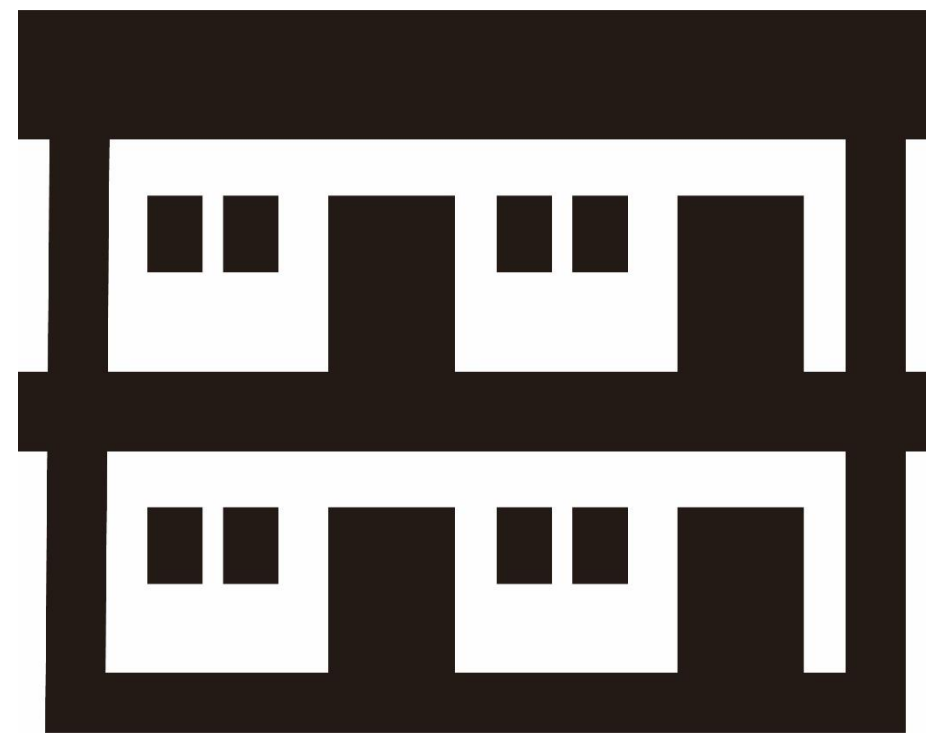
サーバレスとは

サーバレスのイメージ

- ・その土地に長く住み続けたいなら、「持ち家」
- ・転勤があつて住む場所が変わる可能性があるなら、「賃貸」
- ・ではあなたが東京に住んでいて、東京で大阪の友人とたこ焼きパーティを
するとなった場合、自宅が使えないときどこでパーティをしますか？



持ち家



賃貸



??

サーバレスとは

サーバレスのイメージ

- ・数時間のパーティなら、「レンタルルーム」
- ・用途とコスパに応じてインフラを選ぶ
- ・サーバレスは使った分だけ費用を払うレンタルルームのようなもの



レンタルルーム

サーバレスとは

サーバレスの特徴

- ・オンプレミス（持ち家）、IaaS（賃貸）と比較したサーバレス（レンタルルーム）の特徴は以下の通り

		オンプレミス	IaaS	サーバレス
物理インフラ（NW、サーバ等）構築要否		要	不要 ※初期設定（OS選定やCPUメモリ容量選定等）は要	不要 ※初期設定（OS選定やCPUメモリ容量選定等）も不要
費用		構築するシステムに応じた一括課金	サーバ稼働に応じた従量課金	プログラム実行に応じた従量課金
運用管理範囲	プログラム	○	○	○
	ミドルウェア	○	○	—
	OS	○	○	—
	ハードウェア	○	—	—
リソース（CPU、メモリ等）調整可否		否 ※調整できないことはないが費用面の兼ね合いで難しい	可 ※半自動で調整可（調整の為の個別設定要）	可 ※自動で調整可（調整の為の個別設定不要）

サーバレスとは

サーバレスの代表サービス

- ・サーバレスな環境を提供するサービスはFaaS (Function as a Service) と呼ばれる
- ・FaaSはREST APIのような関数 (Function) を実行するための環境を提供する
- ・代表的なFaaSサービスは以下の通り
- ・各サービスごとに対応するプログラミング言語の種類や無料利用枠 (リクエスト数の上限) などに差がある

No	提供元	サービス名
1	AWS	AWS Lambda
2	Google	Google Cloud Functions
3	Microsoft	Azure Functions
4	IBM	IBM Cloud Functions

サーバレスのメリット・デメリット

サーバレスのメリット・デメリット

- ・サーバレスのメリット・デメリットは以下の通り

<メリット>

- ・費用がプログラムの実行分(リクエスト数分)だけで済む

<デメリット>

- ・サーバが常に起動しているわけではないため即応性が低い



<サーバレスアーキテクチャに適するシステム、適さないシステム>

- ・実行頻度の少ないWebシステムなどはサーバレスに適している
(例)一定以上の気温になったらスマホに通知するスマート農業アプリ
- ・逆に実行頻度が多く常にサーバを起動しておく必要のあるシステムは適さない