AWS

SUMMIT

Amazon Robotics フルフィルメントセンターでのパフォーマンスの向上のためのエンジニアリング

Andy Pollock, Sr. Software Development Manager,
Amazon Robotics

2017年6月2日

amazon web services

© 2016, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

アジェンダ

- Amazon Robotics とは？
- モバイルフルフィルメントシステムについて
- Amazon Robotics ソフトウェアについて
- スケールパフォーマンスの設計について

amazon robotics

We Reimagine Now

Eコマースのフルフィルメントセンターのためのロボットシステム

2003年に設立され、注文のから配送までの効率化を推進

Amazon.com 傘下の会社

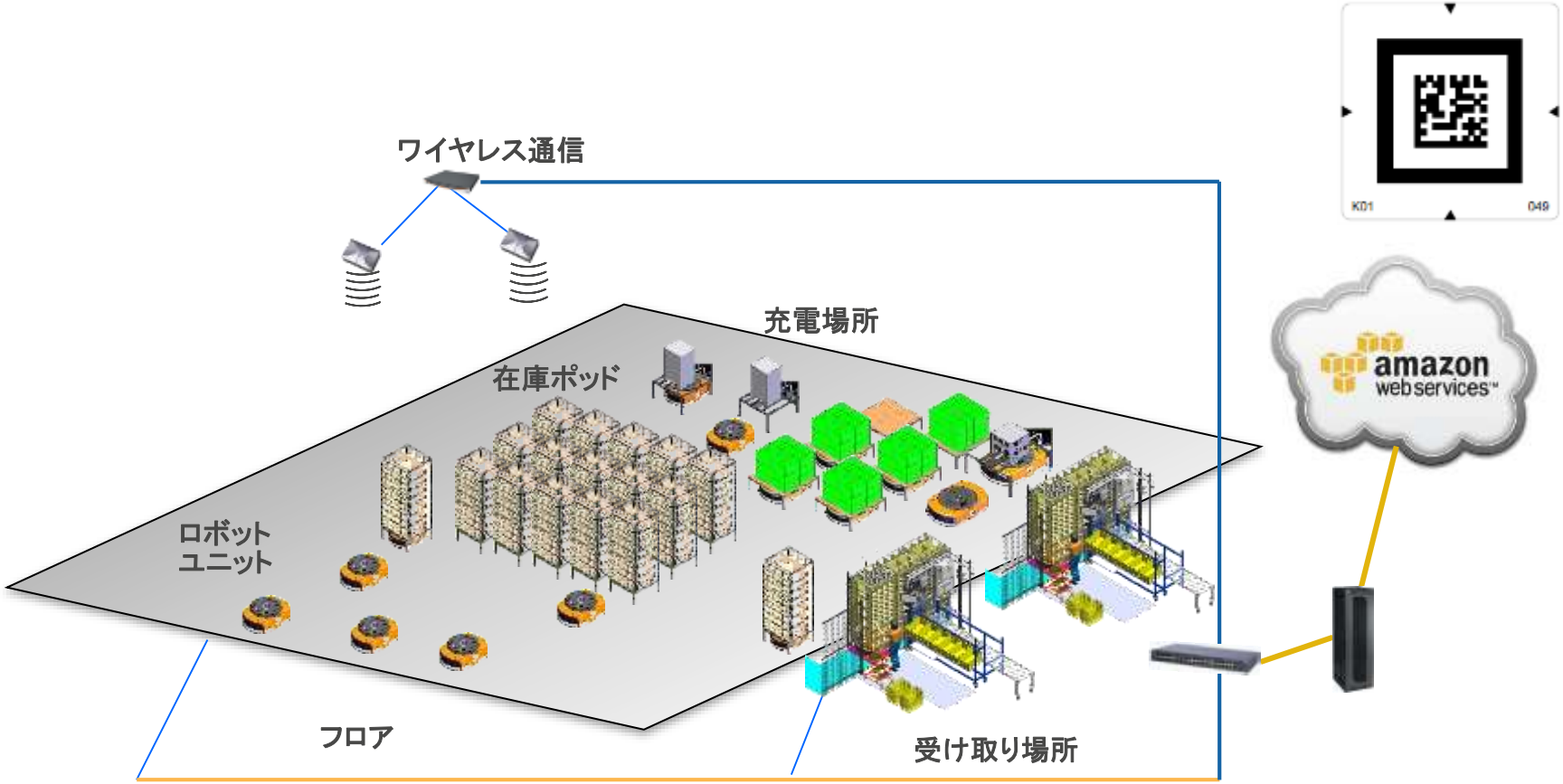
ロボットを使用することにより、人の生産性を上げ、アマゾンフルフィルメントセンターの効率化を図る

ロボティクスとマテリアルハンドリングにおける未解決の課題に取り組んでいる

Amazon Robotics モバイル フルフィルメントシステム

amazon web services

Amazon Robotics システムの詳細



モバイルポッドと移動せずに集める人

ポッドに保管されている在庫

ロボットで移動した棚

密度と耐久性を考慮した設計

受け取り場所に人を配置

注文のために製品を集める

在庫をポッドに補充する

在庫品質を確認する

ロボットの詳細

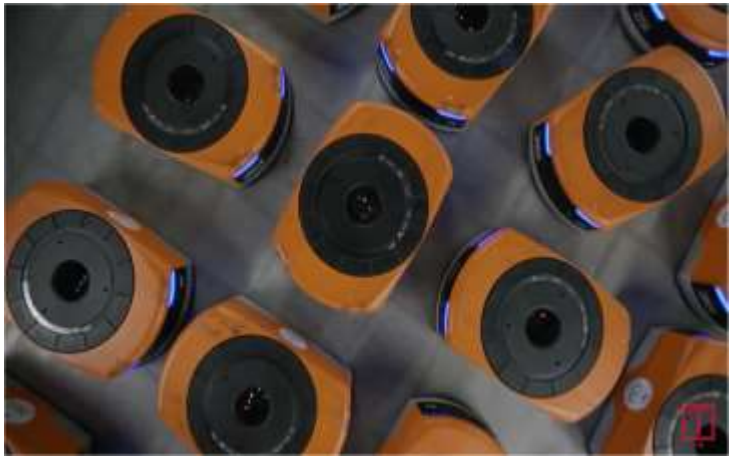
- 3,000ポンド(約1,360キロ)まで運べる
- WiFiネットワーク接続
- 床を検出するカメラを装備
- 障害物検出用カメラを装備
- 充電用ドック
- 頑丈に設計

バランスをとり、流動的に動きます

ロボットは分散し、独立して作業を行う

ロボットを待たず、人が常に仕事をし続けられるよう設計

共有リソースの使用を最適化



Amazon Robotics ソフトウェア

amazon web services

リソース使用率の最適化

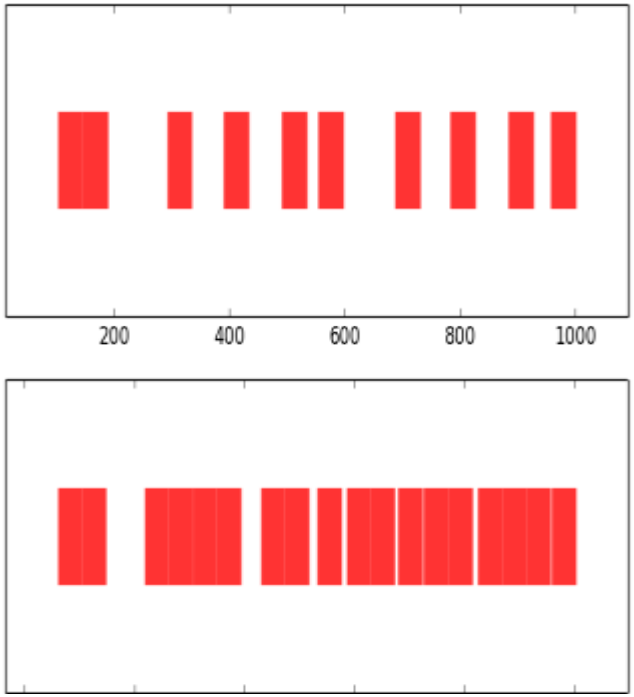
いくつかロボット、人、在庫があるとします。

いくつかの命令を実行してみます

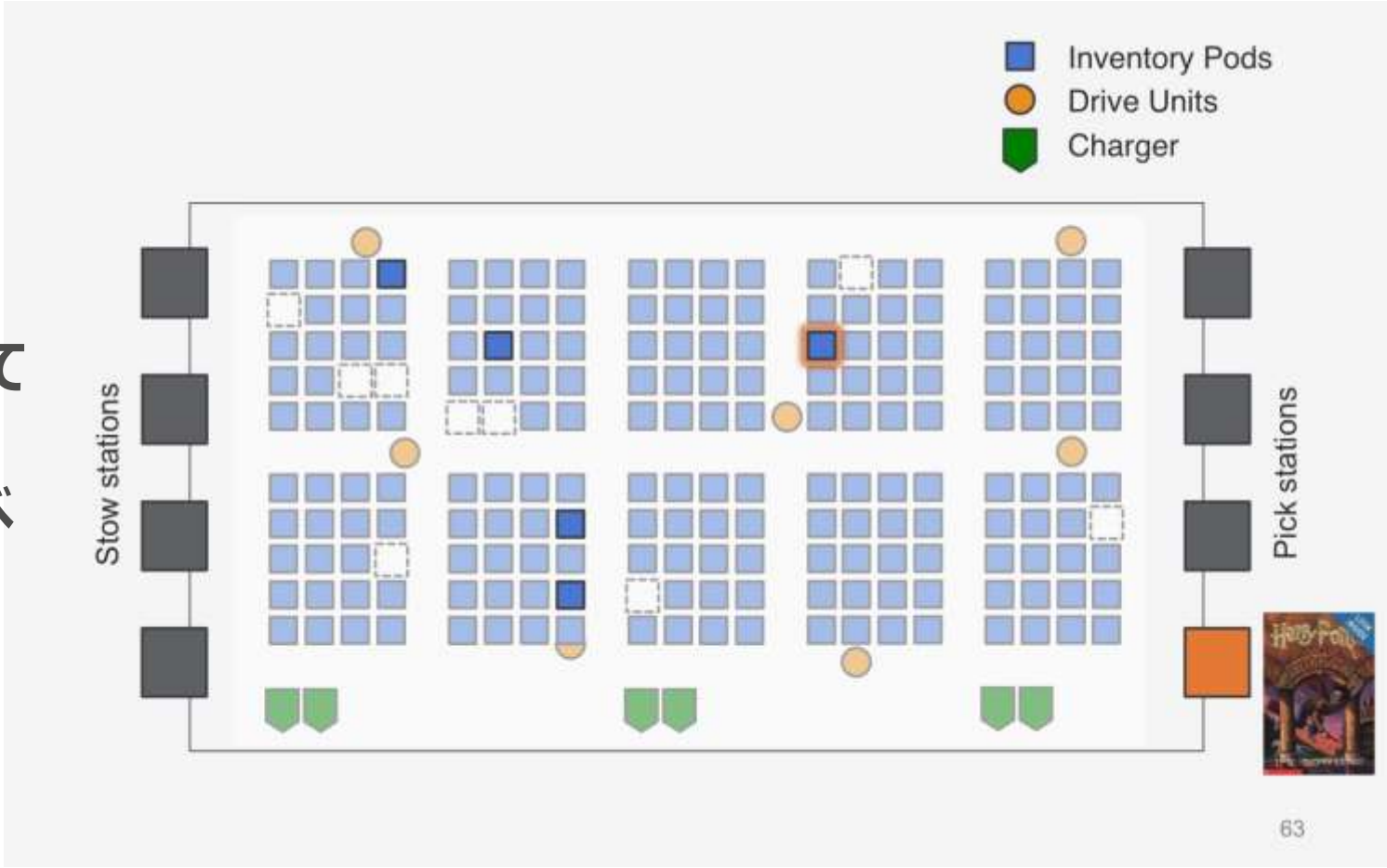
人は最適に動いていますか？

ロボットは最適に使用されていますか？

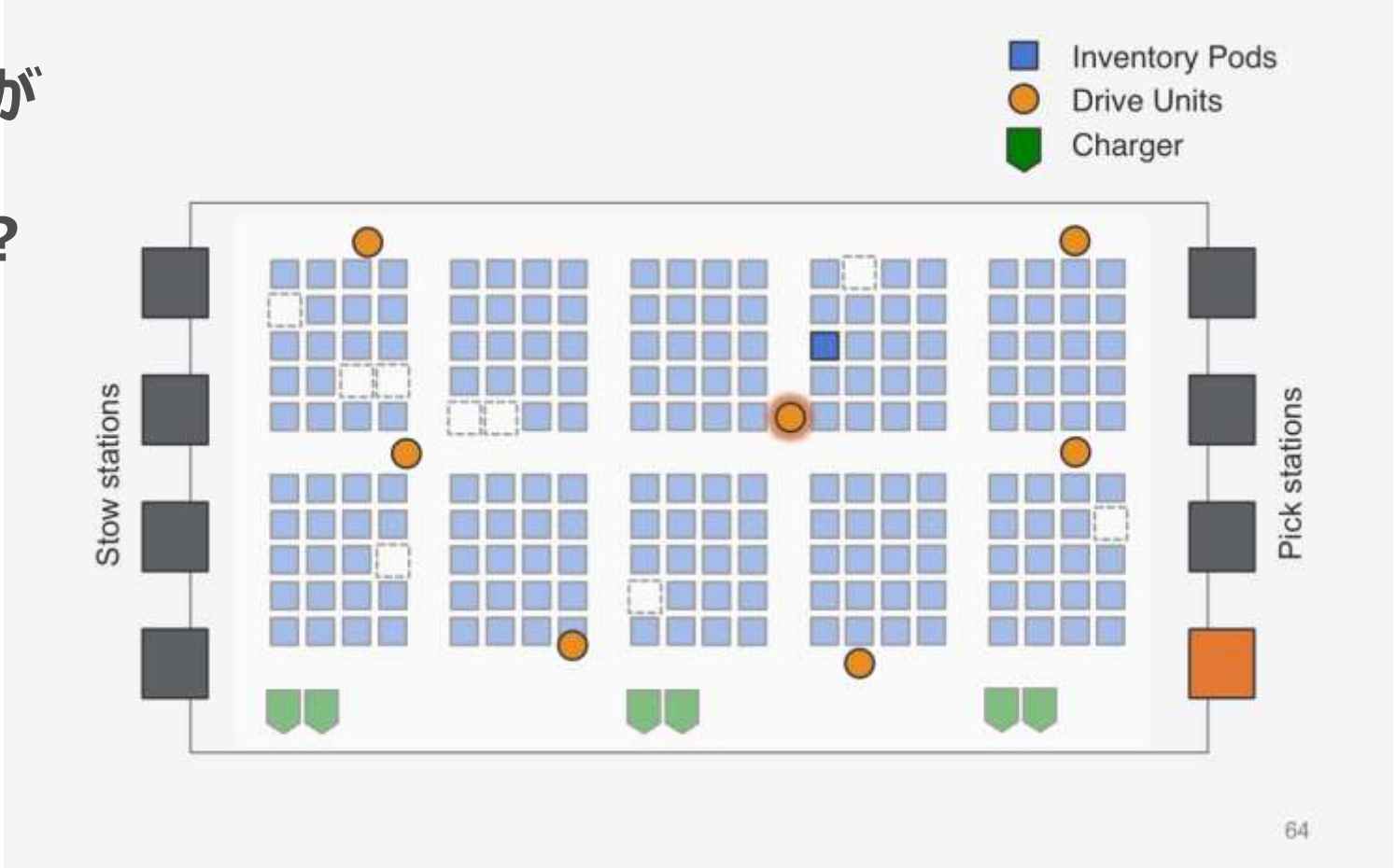
ロボットと人間が最も効率よくなるようにしていますか？



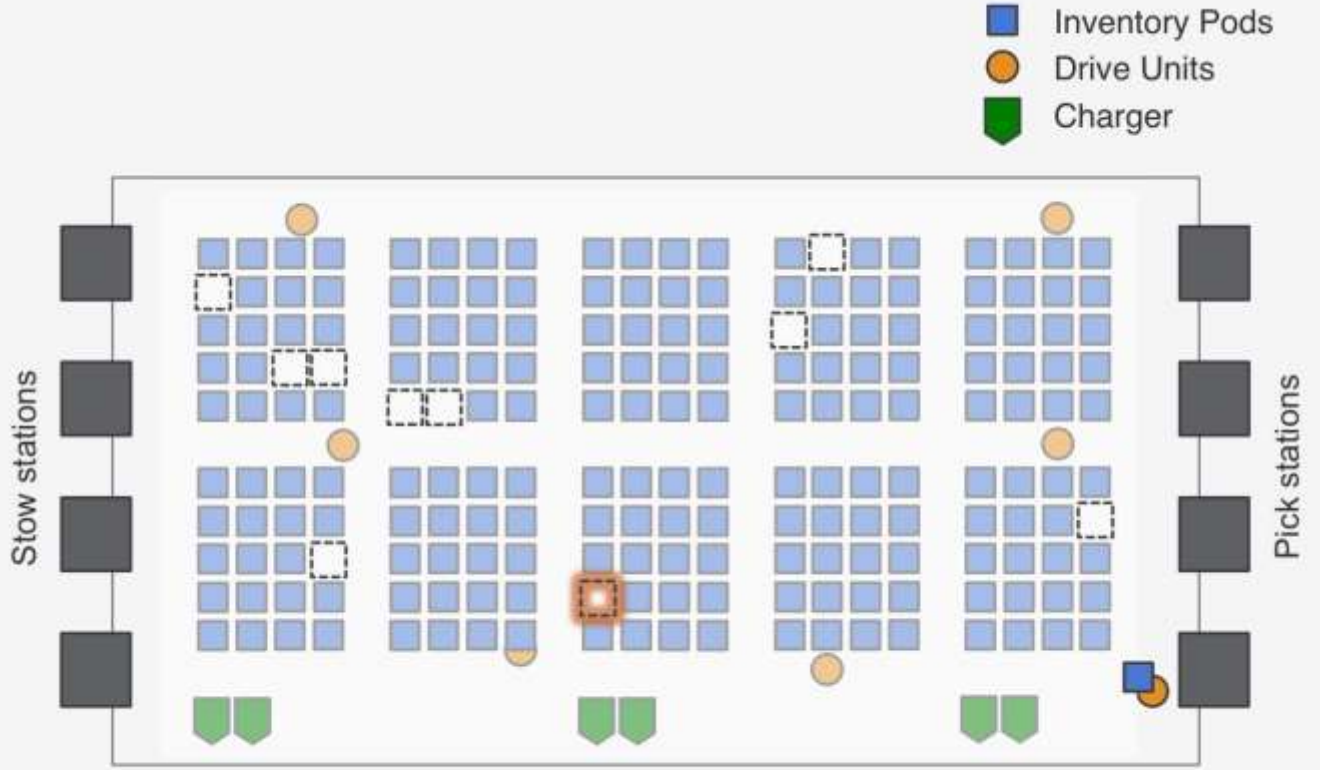
受注された本は、多くのポッドに収納されています。
どこから選ぶべきでしょうか？



どのロボットが
取りに行くべ
きでしょうか?



ポッドが在庫を運び終えました。次は何をすべきでしょうか?



とてつもないスケールを実現するロボット

1,000人以上の人

1,000台以上のロボット

10,000個以上の在庫ポッド

1,000,000個以上の製品

スケールパフォーマンスの設計

amazon web services

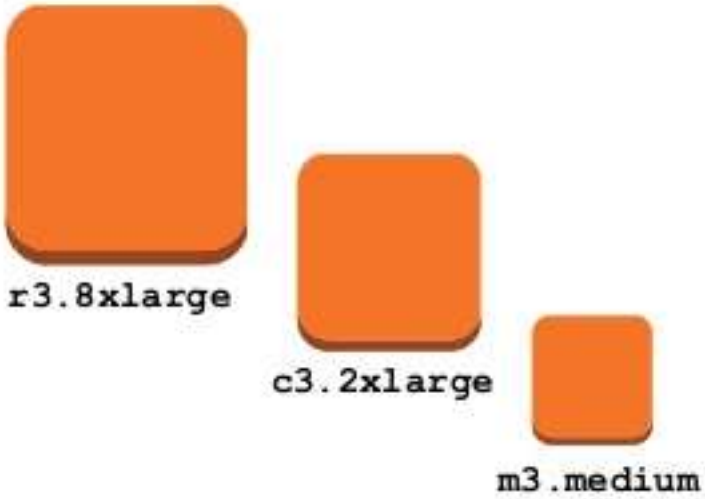
AWS以前のオンプレミスでの設計

垂直方向に拡大 -「大きなシステムを作ろう」

開始する前にはシステムをダウンする必要があった

共有されたリレーショナルデータベース

年間需要のピーク時に合わせてシステムが作られていた



スケールできる限界を伸ばさなければならない

計算能力は固定されたが、顧客の欲求は制限なし

今年、多くの新しいフルフィルメントセンターを建設したとしたら？

フルフィルメントセンターの規模を倍増させたら？

10個の新しい計算能力が必要な機能を追加したとしたら？

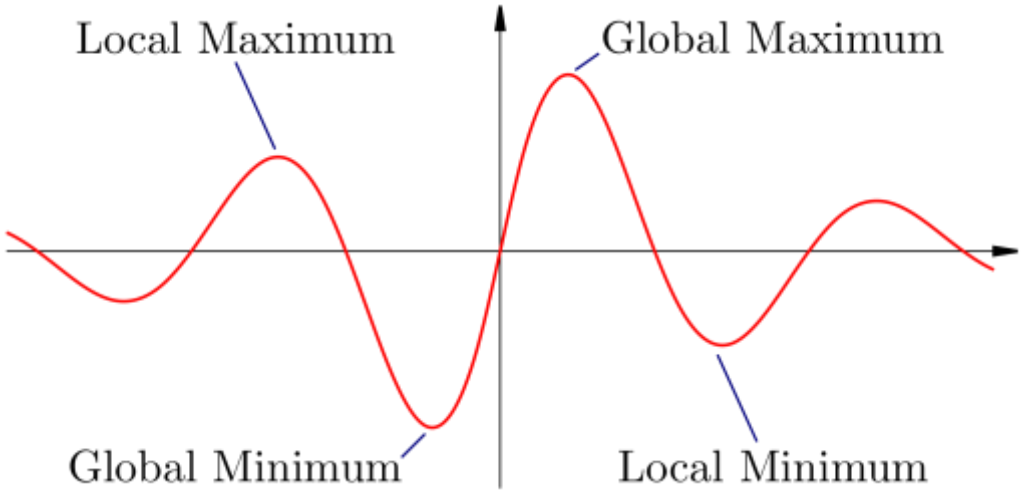
ドライブの数を3倍にしたら？

大きな視点で視る必要がある

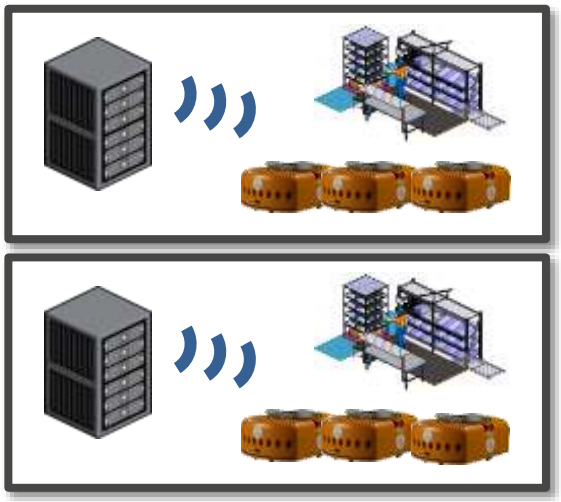
複数の注文をより効率的に処理するにはどうすればよいか？

最適な意思決定を行うためにネットワーク全体を見渡すことができたら？

1つのフルフィルメントセンターまたは1つのゾーンからオーダー全体を取り扱うことができるか？

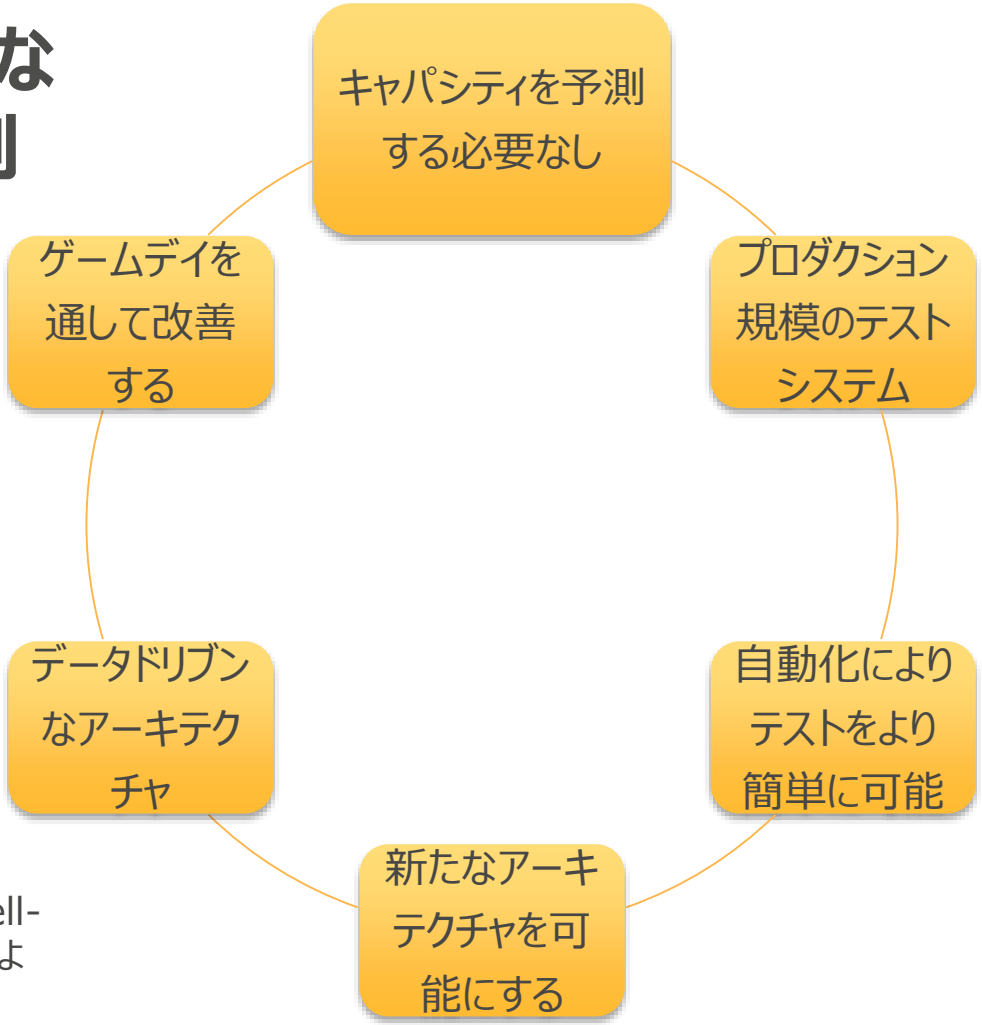


AWSクラウドアーキテクチャで課題を解決



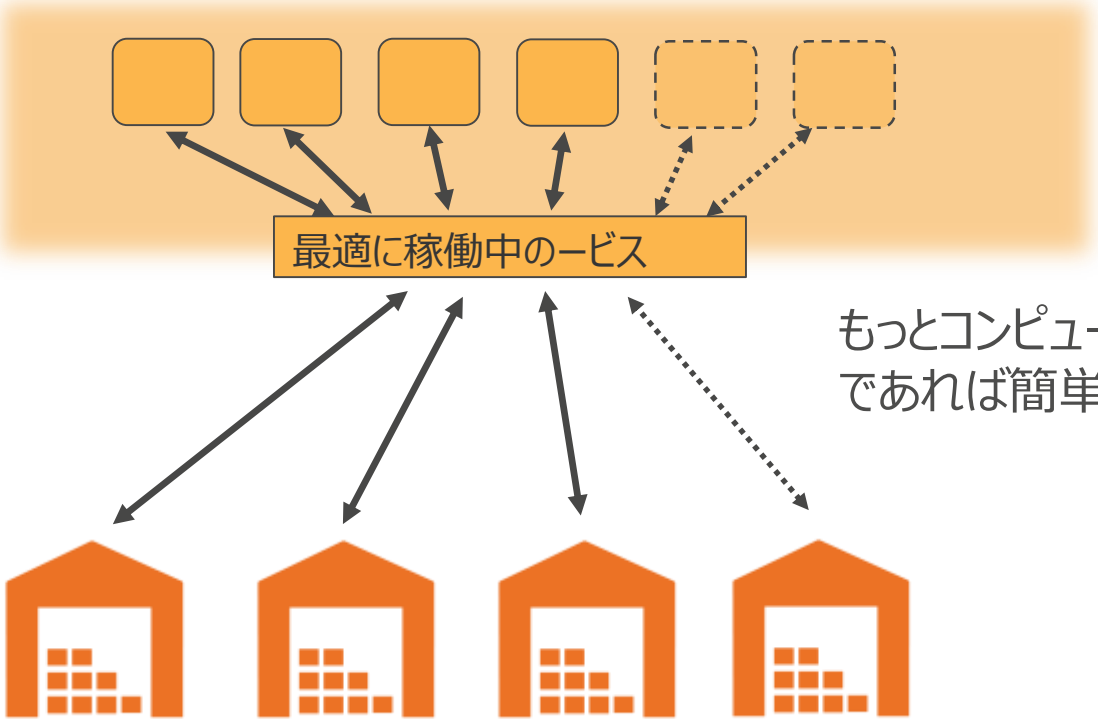
垂直スケーリングの制限	水平にスケール可能
ローカルでの視点	よりグローバルな視点
アップグレード=ダウンタイム	停止時間なし
弾力性の制限	クラウドファーストで、一部のローカルコンポーネント

AWS 基本的なデザインの原則



*2 2016年11月のAWS Well-Architected Frameworkより抜粋

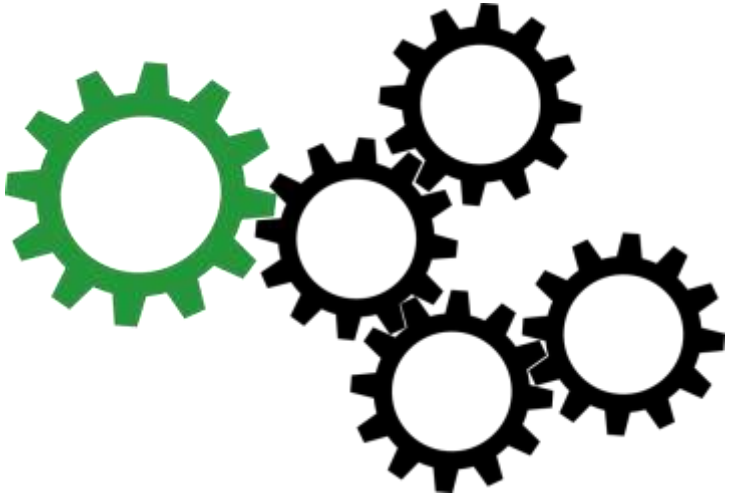
キャパシティを予測する必要なし



もっとコンピューティングパワーが必要であれば簡単に増加可能

大きく成長するのであればクラウドが最適です。

プロダクション規模のテストシステム- バーチャルシステム



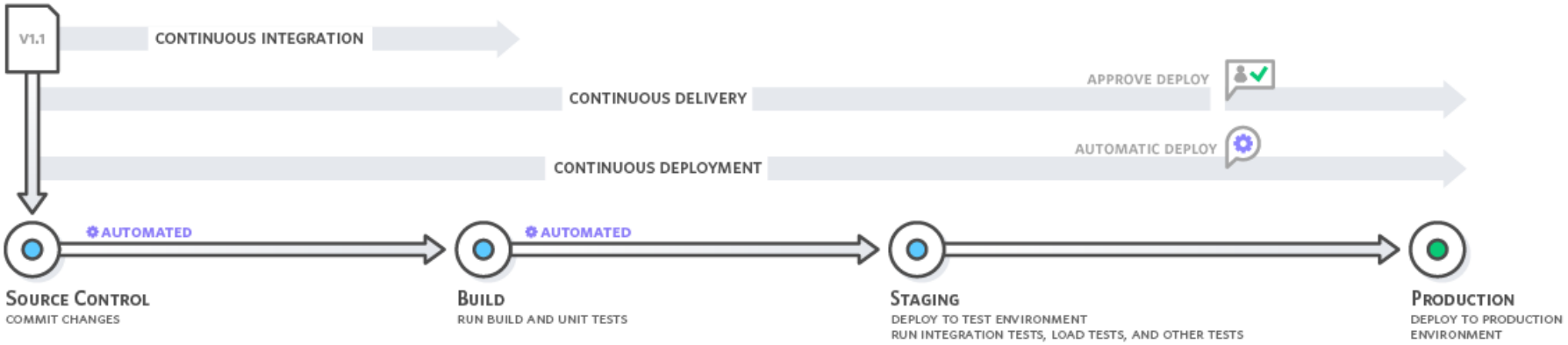
負荷テストをすべてのソフトウェアコンポーネントで実行

スケールのテストにもシステムの視点を取り入れる必要がある

複雑な実環境をエミュレートする必要がある

テスト境界を慎重に設定する必要がある

自動化によりテストをより簡単に可能



継続的な導入によりエンジニアは迅速な対応が可能

テストを世界中の顧客に影響与えないように実施する必要がある

統合されたユニット、負荷、ワンボックステスト

自動ロールバック

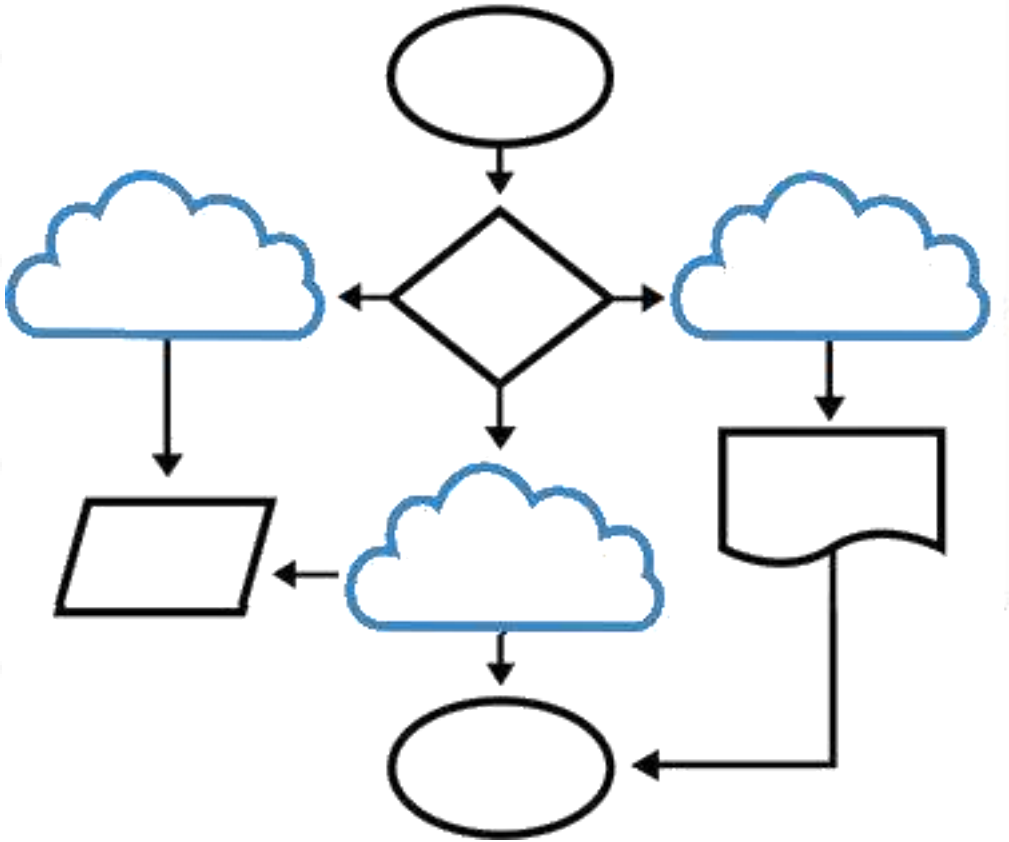
進化したアーキテクチャを可能にする

AWSは不確実性なことにも対応できる

要件は不完全であり、同じ機能で移行することが十分とはいえない

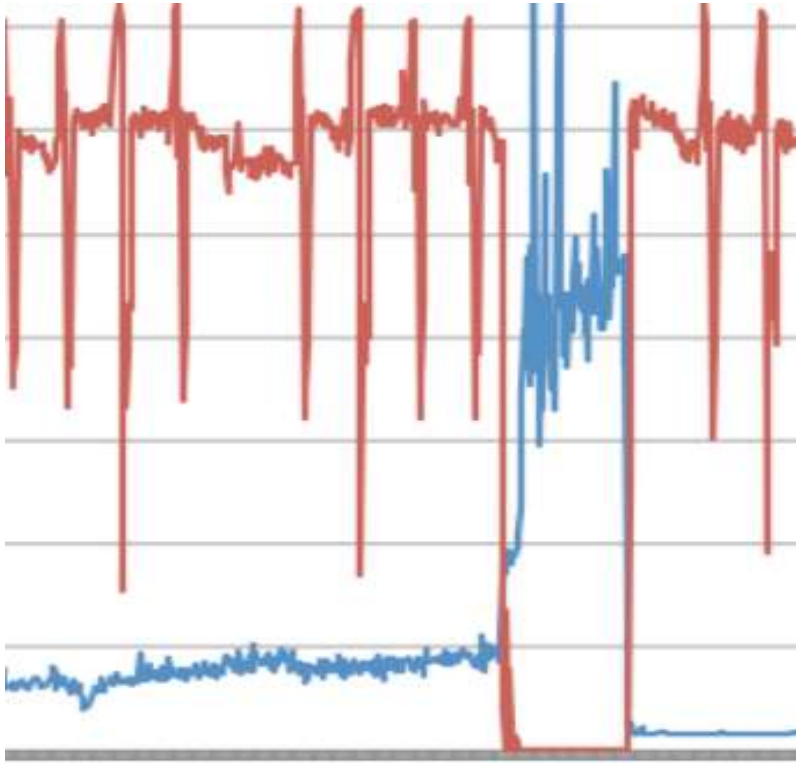
適切なパーシスタンスレイヤーとは何ですか？

どのくらいの粒度でサービスアーキテクチャを設計すべきか？



データドリブンなアーキテクチャ

- あらゆるレベルでの計測
- 1秒あたりのサービストランザクション数
- API使用時のSLA
- システム性能の特性
- メトリクスの計算に基づき、しきい値を超えた時にアラーム
- メトリクスによるアーキテクチャの選択とソフトウェアの改善



Game Day

プリプロダクションでのテストがうまくいっても、実環境では差が生まれる

実稼働中のイベントをシミュレートする

実稼働中のソフトウェア、ネットワークイベントをシミュレートする

天候が嵐のときに船を出す

最終結果: 80,000台のロボットをAWSで稼働

水平スケーラビリティと弾力性

冗長性、高い可用性、フェールオーバー

NoSQLデータストアの水平スケーリング

分析のためのリレーショナルデータベース

大量データを保管

コンポーネント間のメッセージング、パブリケーション/サブスクリプション

Amazon VPC

Amazon EC2

AWS Lambda

Elastic Load Balancing

Amazon DynamoDB

Amazon RDS

Amazon Redshift

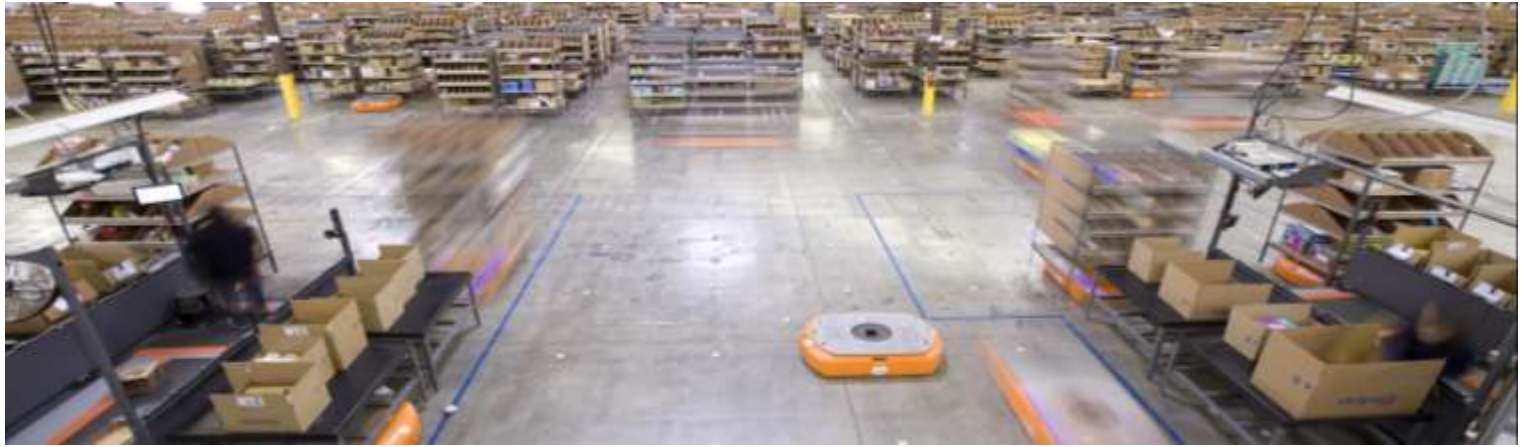
Amazon S3

AWS Snowball

Amazon SNS

Amazon SQS

Amazon CloudWatch



AWS

SUMMIT

amazon web services