

# クラウドネイティブとは

## - CCoE・SREの導入と実践 -

# クラウドネイティブとは

- **クラウドの特性を最適に活用することを目指すアプローチ**
  - クラウドの特性を最大限に活用し、システムのスケーラビリティやレジリエンスを高め、アクセス増加や障害などの事態に迅速に対応することや、開発速度を向上させるアプローチのこと
- **クラウドコンピューティングの普及とともに大きく発展・変化**
  - 初期のクラウドサービスは、従来のオンプレミス環境をクラウドに移行することや、IaaSとしてクラウドサービスを利用できることを重視
  - 現在はアプリケーションレイヤまでクラウドでネイティブに設計することが重要視されている



クラウドネイティブとは

クラウドサービスを利用し、クラウドインフラ環境に最適化されたアプリケーションの開発や運用を目指すシステム、考え方のこと。

- **企業のデジタル変革を支える重要な要素である**
  - 迅速にPoCを実施し、サービスリリースを加速させてビジネス環境変化に追随することが企業にとって非常に重要な取り組みになった
  - 開発サイクルが短縮され、リリース頻度を高めることができると、ビジネス創出などのイノベーションが促進される
- **従来のアプリケーション開発や運用が直面していた課題の解決に繋がる**
  - 予期せぬアクセス量の増加などに対して、リソースの自動スケールを導入することで運用負荷が軽減できる
  - マイクロサービスアーキテクチャを採用することで、変更や障害により強いシステムを構築できる
- **パブリッククラウドやコンテナ技術の普及が進んでいる**
  - フルマネージドなクラウドサービスの積極的な活用が進んでいる
  - コンテナ技術やマイクロサービスを駆使したシステム開発事例が増えている

- システム開発時のメリット

- **アジリティ向上**

- マイクロサービスやコンテナ技術を活用することで、開発プロセスの加速が期待できる。複雑な依存関係を避け、機能追加やバグ修正などのリリース頻度を高めることで、迅速なビジネス検証やイノベーションの促進に繋がる。

- **スケーラビリティ向上**

- 自動スケーリング機能などのクラウドネイティブ技術を利用することで、需要に応じた柔軟なリソース調整が可能になる。急なトラフィックの増加にも迅速に対応でき、コスト効率の良いシステム運用が実現できる。

- システム運用時のメリット

- **可用性向上**

- クラウドネイティブなシステムは、自動復旧/スケール/冗長構成など高い可用性を持つ構成が可能。マイクロサービスでは、一部の障害が全体に波及しないような設計も実現できる。

- **運用自動化**

- コンテナ化されたアプリケーションは、各環境の移行をスムーズに実施できる。CI/CDパイプラインを組み込むことで、テストやデプロイが自動化され、継続的なデリバリーが可能。

▶ **クラウドネイティブなシステムは、開発にスピードと柔軟性を、運用に効率化と安定性をもたらします**

# クラウドネイティブを導入する際のポイント

- **クラウドセンターオブエクセレンス (CCoE) の設立**

- CCoEは、クラウド戦略の策定、ベストプラクティスの共有、および組織全体のクラウドへの取り組みを支援するための専門チーム
- CCoEの設立により、クラウドネイティブへの取り組みに対する技術や知見が組織全体で共有・統一され、各開発に対して横断的に展開することが可能
- CCoEは、クラウドガバナンスの確立、コスト管理、セキュリティポリシーの策定など、クラウド導入に関わるあらゆる側面をサポートする役割

- **サイトリライアビリティエンジニアリング (SRE) の採用**

- SREは、システムの信頼性と効率を維持するためのエンジニアリングアプローチ
- 運用にソフトウェアエンジニアリングの手法を取り入れ、様々な開発や運用の自動化、効果的なデータ収集や可視化を行う
- SREチームは、開発プロセスの自動化、モニタリング、インシデント管理などを通じて、クラウドネイティブなシステムの安定性と信頼性を維持する役割

- **段階的な導入**

- まずは小規模なプロジェクトや特定のサービスから始め、徐々に範囲を拡大していくことが重要。
- 段階的なアプローチにより、組織が新しい技術やプラクティスに適応し、リスクを最小限に抑えながら、クラウドネイティブへの移行を進めることができる。

- **組織文化の変革**

- クラウドネイティブなシステムの導入は、技術的な変化だけでなく、組織文化の変革も伴う。
- エンジニアやスタッフが新しい技術やプラクティスを受け入れ、活用するためには、専門的な技術サポートや教育が不可欠。
- イノベーションを促進するため、クラウドネイティブなアプローチを採用することの重要性を組織全体で共有する必要がある。



これらの要素を意識し、適切に組み合わせることで、クラウドネイティブなシステムの構築運用を実施することができる。

また、組織文化の変革によって、迅速なシステム開発への意識が高まり、ビジネス競争力の向上にも繋がる。

# 代表的なクラウドネイティブ技術

- **Docker**

- アプリケーションをコンテナ化するためのプラットフォームで、アプリケーションとその依存関係をパッケージ化し、環境に依存しない形で実行できるようにしたもの
- アプリケーションの環境移行がスムーズになり、デプロイのプロセスが非常に簡単になる

- **Kubernetes (K8s)**

- コンテナオーケストレーションのためのプラットフォームで、多数のコンテナがある場合に、自動スケーリング、ローリングアップデート、負荷分散など、高度な管理機能を実現することが出来る
- コミュニティが非常に発達しており、エコシステムが豊富であることから、さまざまなOSSと連携して機能を拡張することが出来る

- **Istio**

- マイクロサービス間の通信を管理するためのサービスメッシュと呼ばれるツールで、セキュリティ、監視、トラフィック管理などの機能を提供し、マイクロサービスの高度な運用が可能
- Kubernetes と合わせて活用されることも多く、コンテナ間の通信を透過的に制御し、多数のアプリケーションがある場合にも制御や管理をしやすいになる

- **Observability (o11y)**

- システムの内部状態を理解し、問題を迅速に特定および解決するための力のこと
- アプリケーションやインフラから様々なデータ(ログ・メトリクス・トレースなど)を収集し、システムのパフォーマンスと健全性を一元的に監視する
- OSSではPrometheusやGrafanaなど、SaaSではDataDogやNewRelicなどのツールが、Observabilityの実現に広く使用されている

- **CI/CD**

- 開発プロセスの自動化を通じて、ソフトウェアの品質向上と迅速なリリースを実現する手法
- CI(Continuous Integration)では、開発者がコードをリポジトリにコミットすると、自動的にビルドやテストが実行され、コードの品質や適合性を確認できる
- CD(Continuous Deployment)では、テストをパスしたコードが自動的に本番環境にデプロイ出来るような仕組みを構築する
- Jenkins、Travis CI、CircleCIなどのツールがCI/CDの実現に利用され、AWSのCodePipelineやGoogle CloudのCode Build、GitHub Actionsなどのサービスも広く利用されている

# クラウドネイティブアプローチの事例

- **メルカリ**

- メルカリでは高度なSREチームを組織し、クラウドネイティブやマイクロサービスのシステムを構築、運用を実施している
- サービスのグローバル展開や信頼性の向上に向けて、SREチームが共通的なインフラを整備し、開発速度の向上を目指してマイクロサービス構成を採用している
- 参考：<https://speakerdeck.com/kazeburo/mercari-infrastructure-and-software>

- **メルペイ**

- メルペイではマイクロサービスアーキテクチャを採用し、リリースの短サイクル化に取り組んでいる
- Kubernetesなどクラウドネイティブ技術を利用して開発を進める中で、技術スタックだけでなく、システムを開発・運用する組織文化の変革も重要であると考察している
- 参考：<https://speakerdeck.com/tjun/cloudnative-days-kansai2019>

- **Netflix**

- クラウドネイティブなアプリケーションを通じて、世界中の顧客にストリーミングメディアサービスを提供しており、2016年にはほぼ全てのインフラをクラウドに移行している
- 大部分にAWSを利用し、世界中に配置されたAWSクラウドのリージョンを複数利用することによりグローバルインフラの対応能力が拡大し、世界中でより快適なストリーミング視聴を提供可能にした
- 参考：<https://about.netflix.com/ja/news/completing-the-netflix-cloud-migration>

- **Spotify**

- クラウドネイティブな技術を駆使して音楽ストリーミングサービスを提供しており、特にkubernetesの自動スケーリングなどを活用している
- 突発的なアクセス増加に対してエンジニアが手作業でスケールする必要がなくなり、ユーザ体験の向上だけでなく、エンジニアがアプリケーションの開発に集中できるメリットがあった
- 参考：<https://kubernetes.io/ja/case-studies/spotify/>

クラウドネイティブ支援のご相談がございましたら  
次のお問い合わせ先にご連絡ください。

3>SHAKE

お問い合わせ先：

株式会社スリーシェイク

住所： 東京都新宿区大京町22-1

URL: <https://sreake.com/contact/>

Email: [business@3-shake.com](mailto:business@3-shake.com)

# 会社概要

**会社名** 株式会社スリーシェイク

**設立日** 2015/1/15

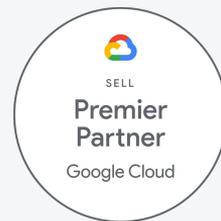
**代表者** 代表取締役社長 吉田 拓真

**所在地** 東京都新宿区大京町22-1  
グランファースト新宿御苑3F・4F

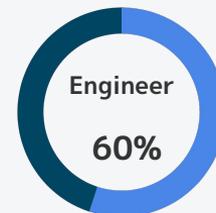
**Mission:** インフラをシンプルにして  
イノベーションが起きやすい世界を作る

**Vision:** 労苦 (Toil) を無くすサービスを適正な価格で提供し続ける

**Value:** エンジニアリングレイヤーに横たわる人、手法、ツールが  
サイロ化されて労苦が発生しているプロセスをシンプルにし  
サービス機能開発に集中できるソリューション  
(SRE、DevSecOps、DataOps、HROps) を提供する



Google Cloud、AWSの両方に強みを持ち  
SREを軸にご支援





## SRE/DevOps



- ・SRE総合支援からセキュリティ対策を全方位支援
- ・Geminiを用いた生成AIの活用支援

## BizOps



- ・クラウド型ETL/データパイプラインSaaSの決定版
- ・あらゆるSaaSをノーコードで連携

## SecOps



- ・ワンストップで脆弱性診断を行うセキュリティ対策SaaS

## HR



- ・ハイスキルフリーランスエンジニア紹介エージェント



IT内製化 / 高度化

クラウドネイティブ化

モダナイゼーション

ITアジリティ向上

# Thank You