戦略的必要性としてのカオスエンジニアリングへの投資

AWS 規範ガイダンス

AWS 規範ガイダンス: 戦略的必要性としてのカオスエンジニアリングへの投資

Table of Contents

戦略的必要性としてのカオスエンジニアリングへの投資

Adrian Hornsby、Amazon Web Services

2025 年 1 月 (ドキュメント履歴)

カオスエンジニアリングプラクティスでは、制御された中断を使用して、システムの問題と、停止やその他のインシデントを防ぐ機会を特定します。カオスエンジニアリングは回復力のあるシステムの改善に不可欠となっていますが、幅広い導入は、誤解、文化的抵抗、リソース、ビジネス価値を定量化する方法に関するハードルに直面しています。初期目標を設定すると、カオスエンジニアリングの取り組みを開始するのに役立ちますが、投資収益率 (ROI) を定量化すると、特に経済的なプレッシャーの中で、継続的な投資が正当化されます。

この戦略ドキュメントでは、量的運用上の改善と質的組織上の利点の両方を把握するための包括的なアプローチの概要を説明します。最終的な目的は、カオスエンジニアリングをサイバーセキュリティに似た戦略的必要性として扱い、継続的なコスト正当化の演習として扱うことではありません。

ダウンタイムコストとカオスエンジニアリングの出現

情報技術インテリジェンスコンサルティング (ITIC) は、企業の 90% がダウンタイムの 1 時間あたり 300,000 USD を超えるコストに直面し、41% が 1 時間あたり 1,500 万 USD を超えると推定しています。ダウンタイムは、収益の即時損失に加えて、コンプライアンスの失敗、株価の低下、大幅な緩和コスト、さらにはブランドの損害など、長期的な問題につながる可能性があります。

ダウンタイムは一般的に収益を生み出すオンラインシステムに関連していますが、悪影響はそれよりもはるかに大きくなります。すべての大企業や組織は、主要な収益モデルに関係なく、人事や給与などの内部システムの可用性に大きく依存しています。

これらのコア内部サービスに影響を与えるダウンタイムは、企業の機能を妨げる可能性があり、大幅な運用の中断や財務上の影響につながる可能性があります。結果として生じる問題には、次のようなものがあります。

- 従業員やベンダーへの支払いの遅延
- 顧客の注文または取引を処理できない
- 侵害されたセキュリティシステムによって許可される機密データの侵害
- 生産性と収益機会の喪失
- コンプライアンス違反に対する規制上の罰則
- ブランド評価の低下

カオスエンジニアリングは、制御された中断を意図的に導入します。カオスエンジニアリングを使用して障害に対するシステムの対応を理解または検証することは、システムの耐障害性を向上させるための重要なプラクティスとなっています。カオスエンジニアリングにより、組織は問題を事前に発見し、レジリエンスメカニズムを検証し、最終的に計画外のダウンタイムやそれに関連するコストを削減できます。カオスエンジニアリングの利点は次のとおりです。

- 技術的負債の公開
- 操作筋の運動
- システムへの信頼の構築
- 障害ポイントの特定
- モニタリングとオブザーバビリティの向上
- 実験ベースの学習のサポート
- 耐障害性の向上によるダウンタイムの短縮

システムが複雑になり、顧客の期待が高まるにつれて、カオスエンジニアリングの重要性が高まっています。ガートナーは、組織が予期しないダウンタイムを短縮し、回復力を向上させるための重要なプラクティスとしてカオスエンジニアリングを推奨しています。

カオスエンジニアリングの導入の課題

カオスエンジニアリングはシステムのレジリエンスを向上させるためにますます重要なプラクティスですが、カオスエンジニアリングの導入は以下の障害に直面する可能性があります。

- リスクに関する誤解 – 一般的な誤解は、カオスエンジニアリングが本番環境でのみ行われ、過剰なリスクに関する懸念につながることです。この認識は、カオスエンジニアリングプラクティスの体系的で制御された性質を理解していないことから生じます。AWS Well-Architected フレームワークで説明されているように、本番稼働用以外の環境で障害シミュレーションを最初に実行します。
- 長期的なビジネス価値 - カオスエンジニアリングの利点は徐々に蓄積されるため、ビジネス価値を定量化し、初期投資を正当化することは困難です。ROI が遅くなると、組織はカオスエンジニアリングに優先順位を付け、維持することが困難になります。
- スキルと専門知識のギャップ - カオスエンジニアリングには、組織内ですぐには利用できない可能性のある独自のスキルと専門知識が必要です。この専門知識を構築または獲得することは、特に実践を初めて行う組織やリソースが限られている組織にとって、大きな障壁となる可能性があります。

この戦略ドキュメントの残りの部分では、主にカオスエンジニアリングのビジネス価値を示す 2 番目の課題に焦点を当てます。

カオスエンジニアリングの蓄積効果

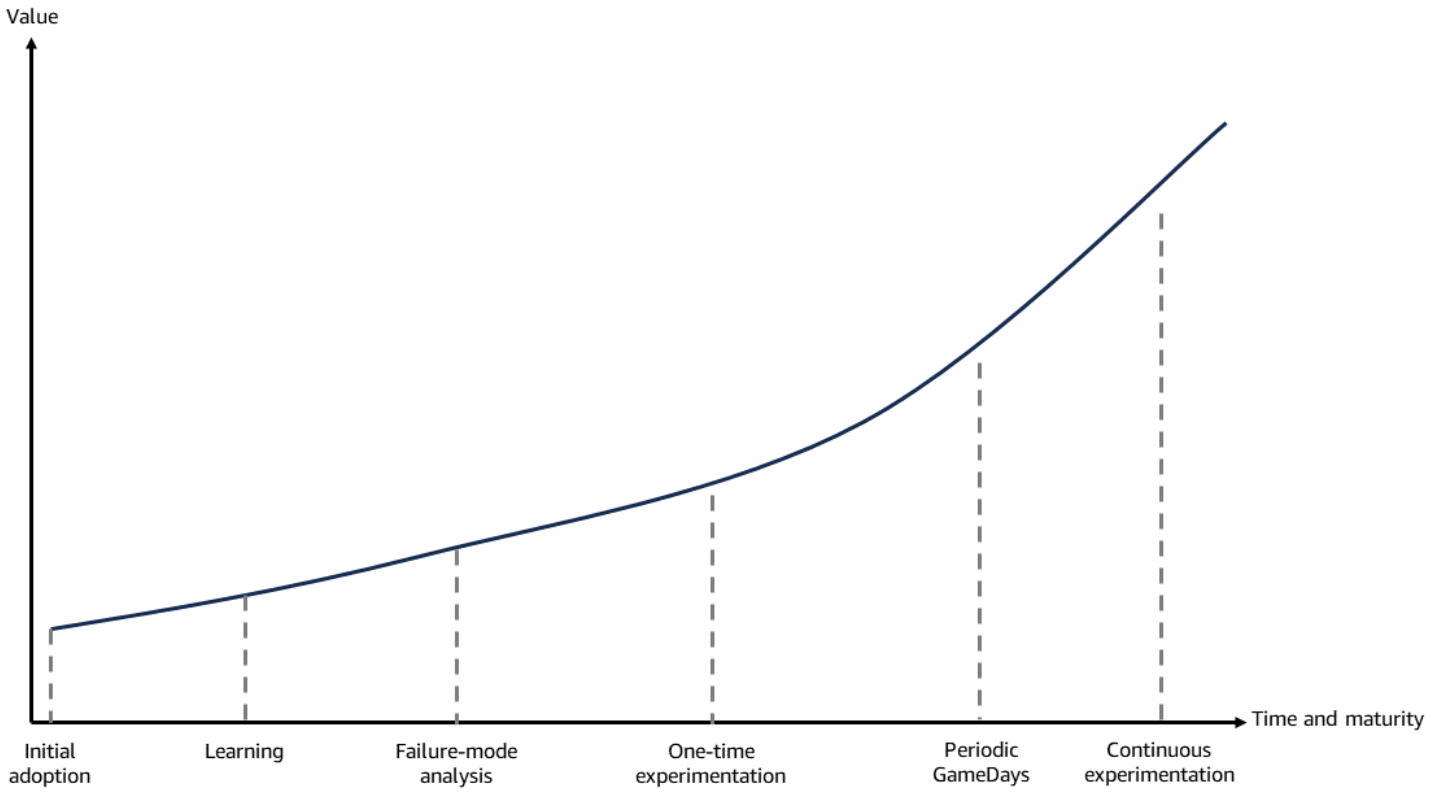
開始日と終了日が明確に定義された従来のテクノロジープロジェクトとは異なり、カオスエンジニアリングは継続的な学習とシステムの回復力の継続的な改善の継続的な実践です。時間の経過に伴うカオスエンジニアリング複合の利点。

システムが進化し、より複雑になるにつれて、新しい障害モードが登場します。潜在的な問題を特定するには、さらに多くのカオス実験が必要です。特に複雑なシステムやプロセスを持つ大企業や、障害が外部のサービスプロバイダーによって所有されている場合、問題の修正には数か月かかることがあります。

学習と改善の機会としての失敗を受け入れる文化のシフトは、何年もかけて成長し、組織に根付いています。カオスエンジニアリング実験を自動化し、サポートツールを開発するための投資は、カオスエンジニアリングの実践を合理化し、強化し続けています。この組織の知識を構築し、システムのレジリエンスを理解することは、時間の経過とともに蓄積される段階的なプロセスです。カオスエンジニアリングによって開発された知識、プロセス、ツールは、継続的に進化するシステムとともに実践が成熟するにつれて価値が増大します。

次の図は、カオス導入が次の段階に進むにつれて、時間の経過とともに価値がどのように向上するかを示しています。

- 初期導入
- 学習
- 障害モード分析
- 1 回限りの実験
- 定期的な GameDays
- 継続的な実験



図に示すように、カオスエンジニアリングの利点は、障害がシステムに注入される前に開始されることがよくあります。カオス実験自体を計画および設計するプロセスは、すぐに価値をもたらします。潜在的な障害シナリオ、単一障害点、システム内の不確実性領域を特定すると、改善につながります。

例えば、障害シナリオを記述し、潜在的なカスケード効果、障害モードと効果分析 (FMEA) と呼ばれるプロセスについて話し合うと、見落とされた可能性のある明らかな弱点やギャップを明らかにするのに役立ちます。組織は、システムが意図的な中断にさらされる前であっても、これらの問題に積極的に対処できます。詳細については、「レジリエンス分析フレームワーク」を参照してください。

さらに、カオスエンジニアリングイニシアチブに伴うシステムオブザーバビリティとモニタリングへの注目が高まり、すぐにメリットがもたらされ始めます。システム動作と障害モードの可視性を向上させることで、チームはシステムの通常の動作状態をよりよく理解できます。また、可視性を高めると、チームは、オペレーション条件が制限にプッシュされたときにどのように低下、適応、失敗するかを理解できます。

1 回限りの実験モードと定期的な GameDay モードはどちらも、連続実験モードよりも手動によるアプローチです。エンジニアが観察や実験を通じて仮説を積極的に形成し、改良する、実践的で探索的なプロセスが必要です。

一方、継続的実験モードは、本質的により自動化されています。このモードは、制御された反復的な方法で承認済みおよび検証済みの仮説を実行することに重点を置いています。専用のカオスパイプラインを通じて開発プロセスで自動化と統合を使用して、一貫性のある反復可能な実験を確保します。

Grassroots カオスエンジニアリングイニシアチブ

カオスエンジニアリングジャーニーは、多くの場合、エンジニアリングチームがニーズを特定し、カオスエンジニアリングを個別に試し始める草の根レベルから始まります。

この草の根アプローチでは、チームはカオスエンジニアリングプラクティスを実験、学習、改良します。カオスエンジニアリングの価値は、以下の具体的な結果を通じて実証できます。

- インシデントの削減
- オブザーバビリティの向上
- 復旧時間の短縮
- システムの耐障害性と継続的な耐障害性の向上

草の根カオスエンジニアリングイニシアチブは、通常、特定の組織条件下で発生します。これには高度なエンジニアリングの自律性を備えた環境が必要です。この環境では、チームは過剰な官公庁の障壁なしに自由に実験やイノベーションを行うことができます。レジリエンスエンジニアリングまたは分散システムに関するローカルの専門知識は、カオス実験を理解して実装するための技術的基盤を提供するため、不可欠です。最も重要なのは、これらのイニシアチブはカオスチャンピオン、つまりカオスエンジニアリングの価値を理解する熱心な個人に依存していることです。カオスチャンピオンは、カオスエンジニアリングの導入を支持し、同僚を教育し、初期実験を推進します。組織の自由、技術的専門知識、意欲的なチャンピオンがいなければ、潜在的な利点に関係なく、草の根のカオスエンジニアリングの取り組みが根付くことはめったにありません。

カオスエンジニアリングの導入における目標の役割

最初の目標は、組織内の草の根カオスエンジニアリングの取り組みから有機的に出現することが一般的です。これらのチームやグループは、自身の繰り返し発生する問題に対処する必要があるため、多くの場合、より高いレベルの明示的な承認や優先順位付けなしにカオスエンジニアリングのプラクティスを検討します。

チームはこれらの結果を使用して、より広範な組織導入のための説得力のあるケースを構築し、他のチームにとって実質的に実証の場になります。

草の根の取り組みによるメリットが大きすぎて無視できないほどになると、これらのチームは取り組みと知識をリーダーシップに昇格させ、目標を設定できます。この可視性の向上により、組織全体のレジリエンス目標の導入が容易になり、カオスエンジニアリングの実装に必要なサポートとリソースにつながります。

目標、特にリーダーシップによって推進され、大規模な停止に対応して確立された目標は、カオスエンジニアリングプラクティスの採用を促進する上で重要な役割を果たします。一般的な目標のタイプは次のとおりです。

- 単一障害点 (SPOF) を特定して削減するための可用性目標
- 中断や障害から回復する能力を向上させるためのサービス復旧目標
- 特定のサービスレベル目標 (SLOs) を達成するためのユーザーエクスペリエンス目標
- 既知の可用性リスクの軽減と推奨されるレジリエンス対策の実施の進行状況を追跡するためのメトリクス主導の目標
- 運用レジリエンスを示すための規制とコンプライアンスの目標

これらの目標タイプの一部と、カオスエンジニアリングの導入中に Amazon およびその他の組織が目標をどのように使用したかの詳細については、付録 A を参照してください。

これらの目標は説得力のある根拠として機能し、カオスエンジニアリングの導入を推進するためのターゲットを絞った実用的なアプローチを提供します。当初、目標は従来の ROI メトリクスのプロキシとして機能します。この目標は、定量化可能なレジリエンス ROI 計算の取得が困難な場合に、説得力のある理論的根拠を提供します。このような目標の導入の早い段階では、カオスエンジニアリングプラクティスは、その有効性を示し、より広範な組織の賛同を得られないリスクがあります。

目標から ROI の測定への移行

プラクティスが成熟し、初期目標が達成されると、焦点は最終的に目標の設定からカオスエンジニアリングの具体的な財務上の利点である投資収益率 (ROI) の定量化へと移行します。このシフトは、主に次の 2 つの理由から発生します。

- 経済的な考慮事項
- カスタマーエクスペリエンスと信頼の維持

経済的な考慮事項

経済成長と健全な財務の時代、企業はカオスエンジニアリング戦略の具体的な目標を設定するための広範な根拠を必要としないことがよくあります。ただし、財務状況の変化により、多くの組織が投資を再評価し、カオスエンジニアリングの実装は定量化された ROI を提供する必要があります。

これらの企業は、カオスエンジニアリングプラクティスの価値と影響を示すために、明確で従来の ROI メトリクスを設定することが求められています。この課題は、防止パラドックスによってさらに複雑になります。防止パラドックスは、インシデントの防止が成功すると、ステークホルダーが回避された災害を過小評価する傾向があるため、投資を正当化するのが難しくなる場合に発生します。運用上の優秀性の文化が深く根付いている組織でも、ROI メトリクスを使用してカオスエンジニアリングの継続的な導入を正当化するというプレッシャーに直面しています。

カスタマーエクスペリエンスと信頼の維持

目標主導のレジリエンスを維持することは、長期的には難しい場合があります。復旧時間目標の達成などの初期目標を達成した後、カオスエンジニアリングの継続的な投資を正当化することは、次の大規模な停止まで困難になります。投資のフローとエッブは、事後対応型の鋸歯サイクルを作成します。新しい停止ごとに、根本的な原因に対処する新しい目標とともに、レジリエンスへの投資が急増します。新しい目標が達成されると、次のインシデントまで投資が中断され、リアクティブループが再開されます。

この事後対応型アプローチを推進する停止は、お客様に悪影響を及ぼします。重要な質問: 顧客がサービスプロバイダーを放棄し、より回復力のある競争相手を優先する前に、顧客が許容する重大な停止の数

カオスエンジニアリングの ROI の定量化

現在、カオスエンジニアリングの長期投資収益率 (ROI) を定量化するための包括的な方法論や実際のデータを提供する公開リソースはほとんどありません。

ホワイトペーパー「The Business Case for Chaos Engineering」では、Netflix はカオスエンジニアリングの ROI を計算するための貴重な方程式を提供しています。この方程式は、カオスエンジニアリングジャーニーに着手する組織にとって出発点となります。

方程式では、以下のコストを正確に見積もる必要があります。

- 予防可能な停止と予防不可能な停止
- カオスエンジニアリングプログラムの実装コスト
- カオスによる損害コスト

カオスによる損害とは、カオスエンジニアリング実験の一環としてシステム内に障害や乱流状態を意図的に注入することによる悪影響や中断を指します。この方程式では、予防可能な停止と予防不可能な停止のコスト、カオスエンジニアリングプログラムの実装コスト、カオスによる損害コストを見積もる必要があります。

カオスエンジニアリングプログラムによってどの問題が防止できたかを確実に判断するのは難しい作業です。これには、問題の根本原因を調べ、カオスエンジニアリング実験がそれらを特定するのにどのように役立ったかを特定することを含む架空の分析が必要です。最新のシステムは非常に複雑で、さまざまなコンポーネント、サービス、サードパーティーライブラリ間の多くの相互依存関係と相互作用があるため、この分析は困難です。さらに、システムの障害は多くの場合非決定的であり、障害を引き起こす条件を後から完全に理解するのは難しい場合があります。

Netflix によって提案されるアプローチにはいくつかの制限がありますが、組織がカオスエンジニアリングを検討し始めるための良い基盤となります。この方程式は、コストと潜在的な利点を見積もるのに役立ちます。これにより、このようなプログラムの実装を決定するのに役立ちます。ただし、組織がカオスエンジニアリングジャーニーをさらに進めるにつれて、ROI 評価を拡張してより包括的な視点を組み込むことが重要です。

この包括的なアプローチは、停止とエンジニアリングコストの削減による直接的な利点を捉えるだけでなく、組織の全体的なレジリエンスに対する長期的で変革的な影響を強調します。カオスエンジニアリングの複合的な利点とより広範な組織効果をキャプチャし、カオスエンジニアリングの真の価値と影響をより正確に表現します。

ROI の定量化に対する包括的なアプローチ

全体的な ROI 評価では、 定量的測定だけでなく、定性的要因も考慮する必要があります。包括的なアプローチには、カオスエンジニアリングを長期間大規模に実践する組織からの実際のデータが必要です。草の根プロジェクトと目標から取得したデータは、収集した方程式アプローチ ROI データを通じて使用できます。

量的測定は、数量または頻度に焦点を当てます。測定値は目的であり、統計的に分析できます。例としては、アンケート、実験、データ分析などがあります。量的対策には以下が含まれます。

- インシデントメトリクス
- コスト
- 改良点
- コンプライアンス
- 導入率
- 顧客満足度

定量的測定を追跡することで、カオスエンジニアリングの直接的な運用上の利点を示すことができます。

定性的尺度は記述的であり、経験や意見の理解に焦点を当てています。多くの場合、主観的であり、数値的に簡単に測定することはできません。カオスエンジニアリングの場合、定性的指標はより広範な組織への影響を捉えます。定性的対策には以下が含まれます。

- 従業員の信頼度
- 文化シフト
- コラボレーション
- トレーニングの有効性
- 人材保持
- ブランドの評価
- 競争上の利点

量的財務的影響と質的組織的利益の両方を考慮することで、レジリエンス文化を育むと同時に、カオスエンジニアリング投資の継続についてより多くの情報に基づいた意思決定を行うことができます。

これらの対策とそれに関連するインシデント分類フレームワークの詳細については、付録 B と付録 C を参照してください。

戦略的必要性としての ROI からカオスエンジニアリングへの移行

ROI をモニタリングする誘惑はありますが、カオスエンジニアリングの価値を測定する際の課題により、組織は戦略的レジリエンスへの投資よりも迅速で短期的な効率性を優先することがよくあります。このアプローチは、カオスエンジニアリングを回復力の主要な推進要因として見落とし、停止を回避することの競争上の利点を見落とします。カオスエンジニアリングの真の価値は、将来の障害を防ぐことです。カオスエンジニアリングは、長期的なビジネス継続性をサポートします。

ROI に焦点を当てるのではなく、カオスエンジニアリングをサイバーセキュリティのように扱います。Forbes の記事「戦略的投資としてのサイバーセキュリティ: ROI 最適化がより安全な未来につながる方法」で説明されているように、サイバーセキュリティは、堅牢なサイバーセキュリティ対策が時間の経過とともに提供できる戦略的価値を認識しないため、組織のコストセンターまたは義務的支出と見なすべきではありません。代わりに、サイバーセキュリティを競争上の利点をもたらす長期的な投資として扱う視点をシフトすることで、組織はそれぞれの市場内でイノベーション、運用効率、差別化のための新しい道を開くことができると主張しています。このアプローチを採用することで、筆者は最高情報セキュリティ責任者 (CISOs。その後、ますますリスクの高いサイバー状況で、競合企業よりも勝つように企業を配置できます。このサイバーセキュリティの長期的で戦略的な価値創造は、カオスエンジニアリングプラクティスに内在する継続的な改善と並行しています。

セキュリティはアセットを運用および保護する組織の能力を保護しますが、カオスエンジニアリングはコアシステムとサービスの可用性、信頼性、回復性を確保するのに役立ちます。長期的な価値と競争上の優位性を実現するには、カオスエンジニアリングを、継続的な根拠を必要とするイニシアチブとしてではなく、中核的な能力と戦略的規範として扱います。

次の図は、草の根から目標と ROI へのカオスエンジニアリングの進化を示しています。

草の根レベルでは、通常、個々のチームはローカルのニーズに応じて個別に実験を行います。これらの実験は、インシデントの軽減とオブザーバビリティの向上を通じて価値を示す熱心なエンジニアによって推進されています。

これらの取り組みが成功すると、チームは学習内容をリーダーシップに昇格させることができます。この可視性により、取り組みは目標駆動型フェーズに移行します。組織は、より広範な実装のためのリソースとサポートに支えられ、回復力と回復のための正式な目標を設定します。

最後に、カオスエンジニアリングは、サイバーセキュリティと同様に、継続的な ROI の根拠が戦略的必要性として認識される必要以上に成熟します。この段階では、カオスエンジニアリングは組織のプロセスに完全に統合されます。実装は、短期的なメトリクスではなく、長期的なレジリエンスに焦点を当てています。カオスエンジニアリングは、競争上の優位性と顧客の信頼を維持するために不可欠なコア機能として扱われます。

カオスエンジニアリングを組織に統合する

カオスエンジニアリングをセキュリティと同じレベルの重要度に昇格させるには、次の提案を検討してください。

- カオスエンジニアリングを交渉不可能なプラクティスとして確立する – サイバーセキュリティが組織の基本的な要件と見なされているのと同様に、カオスエンジニアリングをシステムの耐障害性と信頼性を確保するための必須プラクティスと見なしてください。カオスエンジニアリングをオプションまたは任意のアクティビティとしてではなく、組織のプロセス、ツール、文化に統合します。詳細については、「レジリエンスライフサイクルフレームワークガイド」を参照してください。
- 経営陣レベルの安全な賛同とサポート – セキュリティイニシアチブと同様に、カオスエンジニアリングの取り組みには経営陣からの賛同と積極的なサポートが必要です。これには、組織全体でカオスエンジニアリングプラクティスを実装して維持するための専用のリソース、予算、人員の割り当てが含まれます。
- ガバナンスと監視を実装する – CISO とセキュリティガバナンスフレームワークと同様に、専用のカオスエンジニアリングチームまたは最高レジリエンス責任者を確立します。このチームまたはロールは、さまざまなチームやビジネスユニットにわたるカオスエンジニアリングの取り組みを監督し、調整する責任があります。
- カオスエンジニアリングを開発サイクルと運用サイクルに統合する - セキュリティプラクティスがソフトウェア開発とデプロイプロセスに統合されているのと同様に、カオスエンジニアリングをソフトウェア開発と配信ライフサイクルのシームレスな一部にします。
- 定期的なカオスエンジニアリングドリルとシミュレーションを実施する – セキュリティ侵害シミュレーションやインシデント対応ドリルと同様に、定期的なカオスエンジニアリング実験を実施してインシデント対応機能を検証し、潜在的な死角を事前に特定します。

- カオスエンジニアリングを使用してランブックを維持する – セキュリティレビューの実施と同様に、カオスエンジニアリング実験を使用して、インシデント対応と復旧のためのランブックの有効性と正確性を検証します。さらに、カオスエンジニアリング実験は、オンコールエンジニアがランブック手順の実行を練習するための現実的なシミュレーションとして機能します。シミュレーションは、エンジニアが実際のインシデントを処理するための運用上の記憶力と準備を維持するのに役立ちます。
- レジリエンスの文化を育む – セキュリティ意識トレーニングと同様に、カオスエンジニアリング教育と知識共有イニシアチブに投資し、レジリエンスの文化を育みます。カオスエンジニアリングプラクティスを採用するチームのトレーニングプログラム、部門間のコラボレーション、インセンティブを含めます。
- レジリエンスメトリクスの測定と報告 - レジリエンスメトリクスを定期的にモニタリングし、ステークホルダーに報告します。このドキュメントで説明されている量的および質的メトリクスを出発点として使用します。
- レジリエンスを競争上の優位性として扱う – サイバーセキュリティ対策は競争上の優位性を提供することができます。同様に、カオスエンジニアリングとレジリエンス機能を、より信頼性が高く信頼できるサービスを顧客に提供するのに役立つ差別化要因として見てください。

エグゼクティブの賛同を得る

カオスエンジニアリングには、経営幹部の従来の責任内で明確な所有者がいないことがよくあります。CEO は、成長、生産性向上、市場リーダーシップを重視しています。CFO は、財務パフォーマンス、コスト管理、リスク管理に焦点を当てています。CTO は、テクノロジー戦略、製品ロードマップ、エンジニアリングの優秀性を優先します。CISO はセキュリティとコンプライアンスを監督します。

真のレジリエンスを所有する単一のエグゼクティブがいない場合、賛同とサポートを得ることは難しいことがよくあります。ただし、システムの障害は、収益、顧客満足度、ブランド評価に影響します。これは CEO と CFO にとって懸念事項です。CTO と CISO はレジリエンス対策の実装を任されていますが、組織の義務がない可能性があります。このあいまいさは、戦略的投資を行い、組織を共通のレジリエンス戦略に合わせる妨げになる可能性があります。

このあいまいさにより、カオスエンジニアリングなどのレジリエンスイニシアチブに対する経営陣の賛同を得ることも難しくなります。結局のところ、経営幹部レベルのリーダーは、成長、イノベーション、カスタマーエクスペリエンス、コンプライアンスなど、さまざまな戦略的優先事項に対処しています。

カオスエンジニアリングの価値を C レベルのエグゼクティブに効果的に伝えるには、次のアプローチを検討してください。

- 経営幹部の主な懸念事項と決定要因を決定します。

  例えば、経営幹部は顧客離れ、規制コンプライアンス、コスト削減、競争圧力について懸念していますか？ カオスエンジニアリングを、会社の固有の課題と目標に沿った力の乗数として配置します。

- 共有目標と戦略的成果を特定します。

  カオスエンジニアリング戦略は、組織全体の成長戦略、カスタマーエクスペリエンス、市場機会、運用効率をどのようにサポートしていますか？ 目標、ビジネスへの影響、ROI、およびイニシアチブを行わないリスクに基づいて、イニシアチブに優先順位を付けます。

- 主要なレジリエンス指標を使用して、カオスエンジニアリング戦略の有効性を定量化可能な言葉で伝達します。

  可用性、検出時間、応答時間、復旧時間という 4 つの主要なレジリエンス指標から始めます。これらを収益、コスト削減、ブランド評価などのビジネス成果に直接結び付けます。

- 技術的な詳細が失われることはありません。

  全体的な感情と測定可能なビジネスへの影響に焦点を当てます。経営幹部は、成長を促進し、顧客の信頼を高め、イノベーションを促進する成果を重視します。

防止パラドックス

障害が出現する前に問題が正常に軽減されると、ステークホルダーに講じられた予防策の価値と必要性を確信させることが難しくなります。この現象は防止パラドックスと呼ばれます。予防パラドックスは、カオスエンジニアリングを戦略的必要性として統合する上で最大の障害であり、人間の認識に内在するバイアスから生じます。

Y2K バグはこのパラドックスの優れた例として機能します。何年もの準備と数十億ドルが世界中のコンピュータシステムの更新に費やされました。ただし、2000 年へのスムーズな移行は、Y2K の懸念の誇張された性質の証明として多くの によって解釈されました。行われた予防の取り組みの成功はほとんど認識されませんでした。

この防止パラドックスは、現在もカオスエンジニアリングに投資する組織に課題を与え続けています。プロアクティブな対策によって潜在的な停止が正常に回避されると、壊滅的な事態がまったく

存在しないため、予防に費やされたリソースを逆説的に正当化することが困難になる可能性があります。

この現象の根本原因は、情報を処理するために心をワイヤリングする方法にあります。人間の認知プロセスは、実際のイベントや目に見える結果に対応し、記憶することを目指しています。災害を防ぐと、劇的な説明文を保持したり共有したりすることはありません。防止パラドックスのもう 1 つの側面は、後見バイアスです。イベントが起こらなかった後、個人は何も起こらなかったと結論付ける傾向があるため、実際の問題ではありません。適切な予防策によって実際の問題が妨げられた可能性は認識されません。この心理的な死角は、組織に永続的な課題をもたらします。防止とレジリエンスで成功すればするほど、遡及的に作業が不要に見えます。

防止のパラドックスに対処するために、組織は目に見えない防止作業を目に見える、測定可能な、価値のあるものにするための特定のステップを実行できます。考えられる手順は次のとおりです。

- 予防策なしで何が起こったかをドキュメント化してシミュレートします。
- 予防対策が潜在的な災害を回避したイベントのストーリーを共有します。
- 準備をせず、結果として結果につながったピア組織をポイントします。
- 防止している潜在的な影響を考慮して、防止コストを提示します。
- 防止の取り組みを目に見えるマイルストーンとアチーブメントに分解します。
- 予防策が存在する理由とその履歴の重要性に関する機関メモリを構築します。
- レジリエンスとカオスエンジニアリングプラクティスの価値について、ステークホルダーを定期的に教育します。

結論

カオスエンジニアリングは、組織にとって戦略的に不可欠なものです。導入ジャーニーでは、誤解、文化的な抵抗、リソースの制約などの課題に直面する可能性がありますが、リーダーシップ主導の明確な目標を設定すると、プロセスが悪化する可能性があります。プラクティスが成熟したら、量的な運用上の改善と質的な組織上の利点の両方を捉えた包括的なアプローチを通じて、投資収益率を定量化します。全体的なアプローチは、経済的なプレッシャーの中で特に重要です。

この戦略的必要性を現実のものにするには、まず組織の現在の成熟度レベルを評価します。組織は草の根実験段階、目標駆動段階、またはその中間にありますか？ この評価に基づいて、以下を達成するためのカスタマイズされたロードマップを作成します。

- カオスエンジニアリングガバナンスを確立する (最高レジリエンス責任者を任命するなど) 。
- カオスプラクティスを開発ワークフローに統合します。
- 定期的なトレーニングプログラムを実装します。
- 包括的なレジリエンスメトリクスを開発します。

この変換は夜間には行われません。ただし、これらの具体的なステップを実行することで、継続的なエグゼクティブサポートを確保しながら、カオスエンジニアリングをサイバーセキュリティと同じ戦略的レベルに引き上げることができます。サイバーセキュリティと同様に、カオスエンジニアリングは組織の運用上のDNAとプロセスの不可欠な部分になる可能性があります。

リソース

- ITIC 2021 グローバルサーバーハードウェア、サーバー OS 信頼性アンケートの結果
- カオスエンジニアリングのビジネスケース
- 戦略的投資としてのサイバーセキュリティ: ROI 最適化がより安全な未来につながる方法
- カオスエンジニアリングに関する I&O リーダーガイド
- AWS Resilience Hub スコアの使用方法
- AWS Resilience Hub コンソールを使用した推奨実験の実装

付録 A – カオスエンジニアリングの目標タイプ

以下の目標タイプの説明には、Amazon や他の組織がカオスエンジニアリングの目標を設計した実例が含まれています。

回復力のあるアーキテクチャの目標

カオスエンジニアリングを採用するための最初の推進要因の 1 つは、システムとインフラストラクチャ全体で単一障害点 (SPOF) を特定して削減することです。目標は、特に新しいサービスやアプリケーションについて、重要なシステムとアーキテクチャの耐障害性を検証するように設定されています。

レジリエントアーキテクチャの目標には、サービスの依存関係の障害をシミュレートするカオス実験の実行が含まれます。実験では、タイムアウト、再試行、キャッシュ動作、サーキットブレーカー設定が正しく機能しているかどうかを確認します。これらの実験は、修復の問題を発見し、顧客に影響を与えるインシデントを防ぐのに役立ちます。例については、「カオスエンジニアリングを使用した Prime Video での回復力のあるサービスの構築」を参照してください。

サービス復旧の目標

サービス復旧の目標は、運用の中断やインフラストラクチャの障害から回復する能力の向上に重点を置いています。たとえば、組織は、停止が発生した場合にコアサービスの特定の復旧時間目標 (RTO) を達成することを目指している場合があります。チームはカオス実験を設計して、退避戦略、フェイルオーバーメカニズム、自動復旧プロセスを検証および最適化できます。最適化により、最終的にサービスの復元に必要な時間が短縮されます。例については、AWS Lambda「: Resilience under-the-hood」を参照してください。

ユーザーエクスペリエンスの目標

特にトラフィックの多い時間帯や重要なイベントでは、一貫性があり信頼性の高いユーザーエクスペリエンスを維持することが最優先事項です。このような場合は、特定のサービスレベル目標 (SLOs) を達成することを中心とした目標を設定します。この顧客中心のアプローチにより、障害や障害が発生した場合でも、レジリエンスの取り組みが優れたユーザーエクスペリエンスの提供と直接連携します。例については、「エンジニアリングレジリエンス: Amazon Search のカオスエンジニアリングジャーニーからの教訓」を参照してください。

メトリクス駆動型の目標

実証済みのレジリエンスベストプラクティスを採用するサービスにポイントを与えることで計算されるレジリエンススコアなど、定量的メトリクスに基づいて目標を設定できます。その後、特定のカオス実験を使用してレジリエンススコアを決定できます。このスコアは、チームが既知の可用性リスクを軽減し、推奨されるレジリエンス対策を実装する際の進捗状況を追跡するための指標として役立ちます。ただし、このようなスコアを慎重に解釈し、より広範な耐障害性目標を犠牲にして単一のメトリクスを強調しすぎないようにすることが重要です。例については、「障害耐性スコアについて」を参照してください。

規制コンプライアンスの目標

金融サービス業界は、主に堅牢なレジリエンス機能を必要とする厳格な規制要件によって推進され、カオスエンジニアリングを採用するフロントランナーとして浮上しています。規制では、金融機関が重要なシステムやプロセスの脆弱性を事前に特定、テスト、修復することが要求されます。これらの規制には以下が含まれます。

- 米国連邦機関が発行した運用レジリエンスを強化するためのサウンドプラクティスに関する機関間ホワイトペーパー
- 欧州中央銀行の運用レジリエンスに関するガイドライン
- デジタル運用レジリエンス法 (DORA) に関する欧州委員会の提案

組織が金融機関である場合は、包括的なテストおよび検証戦略を通じて運用レジリエンスを実証するための明示的な目標を設定することで、これらの規制に従ってください。例については、「London Stock Exchange Group uses chaos engineering on AWS to improve resilience」を参照してください。

付録 B – 定量的および定性的尺度

このセクションでは、運用上の改善を追跡するための定量的メトリクスと、カオスエンジニアリングプラクティスによるより広範な組織的成果を評価するための定性的対策の概要を説明します。

量的測定

以下の定量的測定は、カオスエンジニアリングプラクティスを通じて達成された直接的なインシデントと運用上の改善を示すことができる主要なメトリクスを追跡するためのフレームワークを提供します。

- インシデント：
  - インシデント頻度 - インシデント分類フレームワーク内のインシデントの数を追跡し、一定期間にわたって重要度 (重大、重大、マイナー) で分類します。インシデント分類フレームワークの詳細については、付録 C を参照してください。
  - ダウンタイムと低下 - 各インシデント分類のダウンタイムまたはサービス低下の合計期間を測定します。
  - インシデント対応メトリクス – インシデントを理解するには、各インシデント分類の検出時間、特定時間、緩和時間、復旧時間、エスカレーション時間、およびその他の関連メトリクスを測定します。
  - 顧客に影響を与えるインシデント - 顧客に影響を与えるインシデントの数、または顧客に影響を与える前に含まれていたインシデントの割合を追跡します。
  - ランブックの変更 - カオス実験によって得られたインサイトから生じるランブックの更新またはリビジョンの数を追跡します。ランブックには、特定のタイプのインシデントから復旧するための特定のオペレーションまたは手順を実行するための詳細な手順が記載されています。
- コスト：
  - インフラストラクチャコスト – 耐障害性を向上させるために実行されたアクションに必要なクラウドコンピューティングリソースや冗長性対策など、インフラストラクチャコストに関するデータを収集します。
  - 顧客への影響 - システム障害やダウンタイムに関連するカスタマーエクスペリエンス、解約率、収益損失への影響を測定します。
  - スタッフの生産性 - エンジニアリングチームと運用チームが、インシデント対応、戦闘、事後作業、システム障害に関連するその他の事後対応タスクに費やした時間を追跡します。

- 継続的なシステム改善 - カオス実験からのインサイトの直接的な結果として実装されたプロセス改善、アーキテクチャ変更、または自動復旧メカニズムの数をカウントします。
- コンプライアンス - コストを追跡し、運用レジリエンスに関連する規制要件または業界標準を満たすように作業します。
- 導入 - 組織全体のカオスプラクティスの導入率を追跡します。
- 顧客満足度 - 顧客満足度メトリクスの変化を測定し、システムの信頼性の向上がビジネスにどのように影響するかを測定します。

定性的対策

以下の定性的測定は、カオスエンジニアリングプラクティスを通じて達成されたより広範な組織の結果を追跡するためのフレームワークを提供します。

- 従業員の信頼と準備状況：
  - チームを定期的に調査して、実際のインシデントを処理する際の信頼度と、オンコールローテーションに対する認識された準備状況を測定します。
  - トレーニングの一環としてカオス実験に参加したオンコールエンジニアの割合を追跡します。
- 文化シフト：
  - アンケート、フィードバックセッション、または監査を通じて、レジリエンスの考え方が組織にどの程度浸透したかを評価します。
  - カオスエンジニアリングプラクティスを積極的に推進および提唱するチームの数を追跡します。
- 部門間のコラボレーションと知識共有：
  - カオスエンジニアリング学習に関連するチーム間の知識共有セッションやワークショップの頻度と出席状況を追跡します。
  - 複数のチームまたは部門が関与する共同カオスエンジニアリングイニシアチブの数を追跡します。
- トレーニングの有効性：
  - トレーニング後のアンケートや評価を実施して、カオスエンジニアリングトレーニングプログラムの有効性を評価します。
  - カオスエンジニアリングトレーニングプログラムに参加し、事後分析を読むエンジニアの数を追跡します。
- 人材の誘致と保持：

  - カオスエンジニアリングプログラムが、停止の修正に費やす時間と労力を削減することで、トップエンジニアリングの人材を惹きつけ、定着させるのに役立つかどうかを評価します。
- ブランドの評価：
  - 組織の運用レジリエンスに対するコミットメントの実証に関連するブランド認識や評価の変化を追跡します。
- 競争上の利点：
  - システムの可用性の観点から、同業他社に対する競争上の優位性を追跡します。

付録 C – インシデント分類

分類フレームワーク内のインシデントを追跡することは重要です。フレームワークは、システムに影響を与える障害タイプと問題の全体像を提供するためです。組織がインフラストラクチャの障害など、インシデントを 1 つのクラス内でのみ追跡する場合、インサイトを見逃したり、他の分野を改善したりする可能性があります。複数のクラスにわたるインシデントを追跡することで、実行するさまざまなカオス実験をよりよく理解できます。この視点は潜在的な死角を特定し、エンジニアリング範囲の拡大をサポートし、より回復力があり耐障害性のあるシステムにつながります。

推奨されるインシデント分類フレームワークは、インシデントの性質と潜在的な影響に基づいてインシデントを分類するのに役立つように設計されています。インシデントを 8 つの主要なカテゴリにグループ化する高レベルの分類を使用します。

- デプロイの問題：
  - 失敗したデプロイ
  - ロールバックの失敗
  - デプロイ中の設定の問題
- ソフトウェアのバグとリグレッション：
  - 機能バグ
  - 統合の問題
  - パフォーマンスの問題
  - クォータの問題
  - 障害耐性メカニズムの問題 (再試行、タイムアウト)
  - データ整合性の問題
- テストの問題：
  - 欠落しているテスト
  - 無効なテスト
  - フレアテスト
- インフラストラクチャの障害：
  - ハードウェア障害 (サーバー、ネットワークデバイス、ストレージ)
  - スケーリングの問題
  - 依存関係の障害 (サードパーティーのサービス、APIs)
  - ネットワーク接続の問題

- 運用上の問題：
  - ヒューマンエラー (設定ミス、偶発的な変更)
  - 障害のモニタリングとアラート
  - キャパシティプランニングの問題
  - バックアップと復元の失敗
- セキュリティインシデント：
  - 許可されていないアクセスの試行
  - データ侵害
  - サービス拒否 (DoS) 攻撃
- サードパーティーサービスの停止：
  - クラウドプロバイダーの停止
  - DNS 障害
  - 外部 API とサービスの中断
- 環境要因：
  - 自然災害 (地震、火災、洪水、停電)
  - 気象関連の問題

これは、特定のニーズや組織に合わせて調整できる非決定的な分類フレームワークの例です。システムの進化や新しいタイプのインシデントの発生に応じて、分類フレームワークを定期的に確認および更新することをお勧めします。

ドキュメント履歴

以下の表は、本ガイドの重要な変更点について説明したものです。今後の更新に関する通知を受け取る場合は、RSS フィード をサブスクライブできます。

変更	説明	日付
初版発行	—	2025 年 1月 28 日

AWS 規範ガイダンス用語集

以下は、 AWS 規範ガイダンスが提供する戦略、ガイド、パターンで一般的に使用される用語です。エントリを提案するには、用語集の最後のフィードバックの提供リンクを使用します。

数字

7 Rs

アプリケーションをクラウドに移行するための 7 つの一般的な移行戦略。これらの戦略は、ガートナーが 2011 年に特定した 5 Rs に基づいて構築され、以下で構成されています。

- リファクタリング/アーキテクチャの再設計 — クラウドネイティブ特徴を最大限に活用して、俊敏性、パフォーマンス、スケーラビリティを向上させ、アプリケーションを移動させ、アーキテクチャを変更します。これには、通常、オペレーティングシステムとデータベースの移植が含まれます。例: オンプレミスの Oracle データベースを Amazon Aurora PostgreSQL 互換エディションに移行します。
- リプラットフォーム (リフトアンドリシェイプ) — アプリケーションをクラウドに移行し、クラウド機能を活用するための最適化レベルを導入します。例: オンプレミスの Oracle データベースを の Oracle 用 Amazon Relational Database Service (Amazon RDS) に移行します AWS クラウド。
- 再購入 (ドロップアンドショップ) — 通常、従来のライセンスから SaaS モデルに移行して、別の製品に切り替えます。例: カスタマーリレーションシップ管理 (CRM) システムを Salesforce.com に移行します。
- リホスト (リフトアンドシフト) — クラウド機能を活用するための変更を加えずに、アプリケーションをクラウドに移行します。例: オンプレミスの Oracle データベースを の EC2 インスタンス上の Oracle に移行します AWS クラウド。
- 再配置 (ハイパーバイザーレベルのリフトアンドシフト) – 新しいハードウェアを購入したり、アプリケーションを書き換えたり、既存の運用を変更したりすることなく、インフラストラクチャをクラウドに移行できます。サーバーをオンプレミスプラットフォームから同じプラットフォームのクラウドサービスに移行します。例: Microsoft Hyper-Vアプリケーションを に移行します AWS。
- 保持 (再アクセス) — アプリケーションをお客様のソース環境で保持します。これには、主要なリファクタリングを必要とするアプリケーションや、お客様がその作業を後日まで延期したいアプリケーション、およびそれらを移行するためのビジネス上の正当性がないため、お客様が保持するレガシーアプリケーションなどがあります。

- 使用停止 — お客様のソース環境で不要になったアプリケーションを停止または削除します。

A

ABAC

属性ベースのアクセスコントロールを参照してください。

抽象化されたサービス

「マネージドサービス」を参照してください。

ACID

アトミック性、一貫性、分離性、耐久性を参照してください。

アクティブ - アクティブ移行

(双方向レプリケーションツールまたは二重書き込み操作を使用して) ソースデータベースとターゲットデータベースを同期させ、移行中に両方のデータベースが接続アプリケーションからのトランザクションを処理するデータベース移行方法。この方法では、1 回限りのカットオーバーの必要がなく、管理された小規模なバッチで移行できます。より柔軟ですが、アクティブ/パッシブ移行よりも多くの作業が必要です。

アクティブ - パッシブ移行

ソースデータベースとターゲットデータベースを同期させながら、データがターゲットデータベースにレプリケートされている間、接続しているアプリケーションからのトランザクションをソースデータベースのみで処理するデータベース移行の方法。移行中、ターゲットデータベースはトランザクションを受け付けません。

集計関数

行のグループで動作し、グループの単一の戻り値を計算する SQL 関数。集計関数の例には、SUMおよび が含まれますMAX。

AI

「人工知能」を参照してください。

AIOps

「人工知能オペレーション」を参照してください。

匿名化

データセット内の個人情報を完全に削除するプロセス。匿名化は個人のプライバシー保護に役立ちます。匿名化されたデータは、もはや個人データとは見なされません。

アンチパターン

繰り返し起こる問題に対して頻繁に用いられる解決策で、その解決策が逆効果であったり、効果がなかったり、代替案よりも効果が低かったりするもの。

アプリケーションコントロール

マルウェアからシステムを保護するために、承認されたアプリケーションのみを使用できるようにするセキュリティアプローチ。

アプリケーションポートフォリオ

アプリケーションの構築と維持にかかるコスト、およびそのビジネス価値を含む、組織が使用する各アプリケーションに関する詳細情報の集まり。この情報は、ポートフォリオの検出と分析プロセス の需要要素であり、移行、モダナイズ、最適化するアプリケーションを特定し、優先順位を付けるのに役立ちます。

人工知能 (AI)

コンピューティングテクノロジーを使用し、学習、問題の解決、パターンの認識など、通常は人間に関連づけられる認知機能の実行に特化したコンピュータサイエンスの分野。詳細については、「人工知能 (AI) とは何ですか?」を参照してください。

AI オペレーション (AIOps)

機械学習技術を使用して運用上の問題を解決し、運用上のインシデントと人の介入を減らし、サービス品質を向上させるプロセス。 AWS 移行戦略での AIOps の使用方法については、オペレーション統合ガイド を参照してください。

非対称暗号化

暗号化用のパブリックキーと復号用のプライベートキーから成る 1 組のキーを使用した、暗号化のアルゴリズム。パブリックキーは復号には使用されないため共有しても問題ありませんが、プライベートキーの利用は厳しく制限する必要があります。

原子性、一貫性、分離性、耐久性 (ACID)

エラー、停電、その他の問題が発生した場合でも、データベースのデータ有効性と運用上の信頼性を保証する一連のソフトウェアプロパティ。

属性ベースのアクセス制御 (ABAC)

部署、役職、チーム名など、ユーザーの属性に基づいてアクセス許可をきめ細かく設定する方法。詳細については、 AWS Identity and Access Management (IAM) ドキュメントの「 の ABAC AWS」を参照してください。

信頼できるデータソース

最も信頼性のある情報源とされるデータのプライマリーバージョンを保存する場所。匿名化、編集、仮名化など、データを処理または変更する目的で、信頼できるデータソースから他の場所にデータをコピーすることができます。

アベイラビリティーゾーン

他のアベイラビリティーゾーンの障害から AWS リージョン 隔離され、同じリージョン内の他のアベイラビリティーゾーンへの低コストで低レイテンシーのネットワーク接続を提供する 内の別の場所。

AWS クラウド導入フレームワーク (AWS CAF)

のガイドラインとベストプラクティスのフレームワークは、組織がクラウドへの移行を成功させるための効率的で効果的な計画を立て AWS るのに役立ちます。 AWS CAF は、ビジネス、人材、ガバナンス、プラットフォーム、セキュリティ、運用という 6 つの重点分野にガイダンスを整理します。ビジネス、人材、ガバナンスの観点では、ビジネススキルとプロセスに重点を置き、プラットフォーム、セキュリティ、オペレーションの視点は技術的なスキルとプロセスに焦点を当てています。例えば、人材の観点では、人事 (HR)、人材派遣機能、および人材管理を扱うステークホルダーを対象としています。この観点から、 AWS CAF は、クラウド導入を成功させるための組織の準備に役立つ人材開発、トレーニング、コミュニケーションのガイダンスを提供します。詳細については、AWS CAF ウェブサイト と AWS CAF のホワイトペーパー を参照してください。

AWS ワークロード認定フレームワーク (AWS WQF)

データベース移行ワークロードを評価し、移行戦略を推奨し、作業見積もりを提供するツール。AWS WQF は AWS Schema Conversion Tool （ AWS SCT) に含まれています。データベーススキーマとコードオブジェクト、アプリケーションコード、依存関係、およびパフォーマンス特性を分析し、評価レポートを提供します。

B

不正なボット

個人や組織を混乱させたり、損害を与えたりすることを意図したボット。

BCP

「事業継続計画」を参照してください。

動作グラフ

リソースの動作とインタラクションを経時的に示した、一元的なインタラクティブビュー。Amazon Detective の動作グラフを使用すると、失敗したログオンの試行、不審な API 呼び出し、その他同様のアクションを調べることができます。詳細については、Detective ドキュメントのData in a behavior graphを参照してください。

ビッグエンディアンシステム

最上位バイトを最初に格納するシステム。エンディアン性も参照してください。

二項分類

バイナリ結果 (2 つの可能なクラスのうちの 1 つ) を予測するプロセス。例えば、お客様の機械学習モデルで「この E メールはスパムですか、それともスパムではありませんか」などの問題を予測する必要があるかもしれません。または「この製品は書籍ですか、車ですか」などの問題を予測する必要があるかもしれません。

ブルームフィルター

要素がセットのメンバーであるかどうかをテストするために使用される、確率的でメモリ効率の高いデータ構造。

ブルー/グリーンデプロイ

2 つの異なる同一の環境を作成するデプロイ戦略。現在のアプリケーションバージョンを 1 つの環境 (青) で実行し、新しいアプリケーションバージョンを別の環境 (緑) で実行します。この戦略は、最小限の影響で迅速にロールバックするのに役立ちます。

ボット

インターネット経由で自動タスクを実行し、人間のアクティビティややり取りをシミュレートするソフトウェアアプリケーション。インターネット上の情報のインデックスを作成するウェブクローラーなど、一部のボットは有用または有益です。悪質なボットと呼ばれる他のボットの中には、個人や組織を混乱させたり、損害を与えたりすることを意図したものもあります。

ボットネット

マルウェアに感染し、ボットハーダーまたはボットオペレーターとして知られる、単一の当事者によって制御されているボットのネットワーク。ボットは、ボットとその影響をスケールするための最もよく知られているメカニズムです。

ブランチ

コードリポジトリに含まれる領域。リポジトリに最初に作成するブランチは、メインブランチといいます。既存のブランチから新しいブランチを作成し、その新しいブランチで機能を開発したり、バグを修正したりできます。機能を構築するために作成するブランチは、通常、機能ブランチと呼ばれます。機能をリリースする準備ができたら、機能ブランチをメインブランチに統合します。詳細については、「ブランチの概要」(GitHub ドキュメント) を参照してください。

ブレークグラスアクセス

例外的な状況では、承認されたプロセスを通じて、ユーザーが AWS アカウント 通常アクセス許可を持たない にすばやくアクセスできるようになります。詳細については、 Well-Architected ガイダンスの「ブレークグラス手順の実装」インジケータ AWS を参照してください。

ブラウンフィールド戦略

環境の既存インフラストラクチャ。システムアーキテクチャにブラウンフィールド戦略を導入する場合、現在のシステムとインフラストラクチャの制約に基づいてアーキテクチャを設計します。既存のインフラストラクチャを拡張している場合は、ブラウンフィールド戦略とグリーンフィールド戦略を融合させることもできます。

バッファキャッシュ

アクセス頻度が最も高いデータが保存されるメモリ領域。

ビジネス能力

価値を生み出すためにビジネスが行うこと (営業、カスタマーサービス、マーケティングなど)。マイクロサービスのアーキテクチャと開発の決定は、ビジネス能力によって推進できます。詳細については、ホワイトペーパー AWSでのコンテナ化されたマイクロサービスの実行 の ビジネス機能を中心に組織化 セクションを参照してください。

ビジネス継続性計画 (BCP)

大規模移行など、中断を伴うイベントが運用に与える潜在的な影響に対処し、ビジネスを迅速に再開できるようにする計画。

C

CAF

AWS 「クラウド導入フレームワーク」を参照してください。

Canary デプロイ

エンドユーザーへのバージョンのスローリリースと増分リリース。確信が持てば、新しいバージョンをデプロイし、現在のバージョン全体を置き換えます。

CCoE

「Cloud Center of Excellence」を参照してください。

CDC

「データキャプチャの変更」を参照してください。

変更データキャプチャ (CDC)

データソース (データベーステーブルなど) の変更を追跡し、その変更に関するメタデータを記録するプロセス。CDC は、ターゲットシステムでの変更を監査またはレプリケートして同期を維持するなど、さまざまな目的に使用できます。

カオスエンジニアリング

障害や破壊的なイベントを意図的に導入して、システムの耐障害性をテストします。AWS Fault Injection Service （ AWS FIS ） を使用して、 AWS ワークロードにストレスを与え、その応答を評価する実験を実行できます。

CI/CD

継続的インテグレーションと継続的デリバリーを参照してください。

分類

予測を生成するのに役立つ分類プロセス。分類問題の機械学習モデルは、離散値を予測します。離散値は、常に互いに区別されます。例えば、モデルがイメージ内に車があるかどうかを評価する必要がある場合があります。

クライアント側の暗号化

ターゲットがデータ AWS のサービス を受信する前のローカルでのデータの暗号化。

Cloud Center of Excellence (CCoE)

クラウドのベストプラクティスの作成、リソースの移動、移行のタイムラインの確立、大規模変革を通じて組織をリードするなど、組織全体のクラウド導入の取り組みを推進する学際的なチーム。詳細については、 AWS クラウド エンタープライズ戦略ブログの CCoE 投稿を参照してください。

クラウドコンピューティング

リモートデータストレージと IoT デバイス管理に通常使用されるクラウドテクノロジー。クラウドコンピューティングは、一般的にエッジコンピューティングテクノロジーに接続されています。

クラウド運用モデル

IT 組織において、1 つ以上のクラウド環境を構築、成熟、最適化するために使用される運用モデル。詳細については、「クラウド運用モデルの構築」 を参照してください。

導入のクラウドステージ

組織が に移行するときに通常実行する 4 つのフェーズ AWS クラウド :

- プロジェクト — 概念実証と学習を目的として、クラウド関連のプロジェクトをいくつか実行する
- 基礎固め — お客様のクラウドの導入を拡大するための基礎的な投資 (ランディングゾーンの作成、CCoE の定義、運用モデルの確立など)
- 移行 — 個々のアプリケーションの移行
- 再発明 — 製品とサービスの最適化、クラウドでのイノベーション

これらのステージは、 AWS クラウド エンタープライズ戦略ブログのブログ記事「クラウドファーストへのジャーニー」と「導入のステージ」で Stephen Orban によって定義されました。移行戦略との関連性については、 AWS「移行準備ガイド」を参照してください。

CMDB

「設定管理データベース」を参照してください。

コードリポジトリ

ソースコードやその他の資産 (ドキュメント、サンプル、スクリプトなど) が保存され、バージョン管理プロセスを通じて更新される場所。一般的なクラウドリポジトリには、 GitHubまたは が含まれますBitbucket Cloud。コードの各バージョンはブランチと呼ばれます。マイクロサービスの構造では、各リポジトリは 1 つの機能専用です。1 つの CI/CD パイプラインで複数のリポジトリを使用できます。

コールドキャッシュ

空である、または、かなり空きがある、もしくは、古いデータや無関係なデータが含まれているバッファキャッシュ。データベースインスタンスはメインメモリまたはディスクから読み取る必要があり、バッファキャッシュから読み取るよりも時間がかかるため、パフォーマンスに影響します。

コールドデータ

めったにアクセスされず、通常は過去のデータです。この種類のデータをクエリする場合、通常は低速なクエリでも問題ありません。このデータを低パフォーマンスで安価なストレージ階層またはクラスに移動すると、コストを削減することができます。

コンピュータビジョン (CV)

機械学習を使用してデジタルイメージやビデオなどのビジュアル形式から情報を分析および抽出する AI の分野。例えば、Amazon SageMaker AI は CV 用の画像処理アルゴリズムを提供します。

設定ドリフト

ワークロードの場合、設定は想定状態から変化します。ワークロードが非準拠になる可能性があり、通常は段階的かつ意図的ではありません。

構成管理データベース (CMDB)

データベースとその IT 環境 (ハードウェアとソフトウェアの両方のコンポーネントとその設定を含む) に関する情報を保存、管理するリポジトリ。通常、CMDB のデータは、移行のポートフォリオの検出と分析の段階で使用します。

コンフォーマンスパック

コンプライアンスチェックとセキュリティチェックをカスタマイズするためにアセンブルできる AWS Config ルールと修復アクションのコレクション。YAML テンプレートを使用して、コンフォーマンスパックを AWS アカウント および リージョンの単一のエンティティとしてデプロイすることも、組織全体にデプロイすることもできます。詳細については、 AWS Config ドキュメントの「コンフォーマンスパック」を参照してください。

継続的インテグレーションと継続的デリバリー (CI/CD)

ソフトウェアリリースプロセスのソース、ビルド、テスト、ステージング、本番の各ステージを自動化するプロセス。CI/CD は一般的にパイプラインと呼ばれます。プロセスの自動化、生産性の向上、コード品質の向上、配信の加速化を可能にします。詳細については、「継続的デリバ

リーの利点」を参照してください。CD は継続的デプロイ (Continuous Deployment) の略語でもあります。詳細については「継続的デリバリーと継続的なデプロイ」を参照してください。

CV

「コンピュータビジョン」を参照してください。

D

保管中のデータ

ストレージ内にあるデータなど、常に自社のネットワーク内にあるデータ。

データ分類

ネットワーク内のデータを重要度と機密性に基づいて識別、分類するプロセス。データに適した保護および保持のコントロールを判断する際に役立つため、あらゆるサイバーセキュリティのリスク管理戦略において重要な要素です。データ分類は、 AWS Well-Architected フレームワークのセキュリティの柱のコンポーネントです。詳細については、データ分類を参照してください。

データドリフト

実稼働データと ML モデルのトレーニングに使用されたデータとの間に有意な差異が生じたり、入力データが時間の経過と共に有意に変化したりすることです。データドリフトは、ML モデル予測の全体的な品質、精度、公平性を低下させる可能性があります。

転送中のデータ

ネットワーク内 (ネットワークリソース間など) を活発に移動するデータ。

データメッシュ

一元管理とガバナンスを備えた分散型の分散型データ所有権を提供するアーキテクチャフレームワーク。

データ最小化

厳密に必要なデータのみを収集し、処理するという原則。でデータ最小化を実践 AWS クラウドすることで、プライバシーリスク、コスト、分析のカーボンフットプリントを削減できます。

データ境界

AWS 環境内の一連の予防ガードレール。信頼された ID のみが、期待されるネットワークから信頼されたリソースにアクセスできるようにします。詳細については、「 でのデータ境界の構築 AWS」を参照してください。

データの前処理

raw データをお客様の機械学習モデルで簡単に解析できる形式に変換すること。データの前処理とは、特定の列または行を削除して、欠落している、矛盾している、または重複する値に対処することを意味します。

データ出所

データの生成、送信、保存の方法など、データのライフサイクル全体を通じてデータの出所と履歴を追跡するプロセス。

データ件名

データを収集、処理している個人。

データウェアハウス

分析などのビジネスインテリジェンスをサポートするデータ管理システム。データウェアハウスには通常、大量の履歴データが含まれており、通常はクエリや分析に使用されます。

データベース定義言語 (DDL)

データベース内のテーブルやオブジェクトの構造を作成または変更するためのステートメントまたはコマンド。

データベース操作言語 (DML)

データベース内の情報を変更 (挿入、更新、削除) するためのステートメントまたはコマンド。

DDL

「データベース定義言語」を参照してください。

ディープアンサンブル

予測のために複数の深層学習モデルを組み合わせる。ディープアンサンブルを使用して、より正確な予測を取得したり、予測の不確実性を推定したりできます。

ディープラーニング

人工ニューラルネットワークの複数層を使用して、入力データと対象のターゲット変数の間のマッピングを識別する機械学習サブフィールド。

多層防御

一連のセキュリティメカニズムとコントロールをコンピュータネットワーク全体に層状に重ねて、ネットワークとその内部にあるデータの機密性、整合性、可用性を保護する情報セキュリ

ティの手法。この戦略を採用するときは AWS、 AWS Organizations 構造の異なるレイヤーに複数のコントロールを追加して、リソースの安全性を確保します。たとえば、多層防御アプローチでは、多要素認証、ネットワークセグメンテーション、暗号化を組み合わせることができます。

委任管理者

では AWS Organizations、互換性のあるサービスが AWS メンバーアカウントを登録して組織のアカウントを管理し、そのサービスのアクセス許可を管理できます。このアカウントを、そのサービスの委任管理者と呼びます。詳細、および互換性のあるサービスの一覧は、 AWS Organizations ドキュメントのAWS Organizationsで使用できるサービスを参照してください。

デプロイ

アプリケーション、新機能、コードの修正をターゲットの環境で利用できるようにするプロセス。デプロイでは、コードベースに変更を施した後、アプリケーションの環境でそのコードベースを構築して実行します。

開発環境

「環境」を参照してください。

検出管理

イベントが発生したときに、検出、ログ記録、警告を行うように設計されたセキュリティコントロール。これらのコントロールは副次的な防衛手段であり、実行中の予防的コントロールをすり抜けたセキュリティイベントをユーザーに警告します。詳細については、Implementing security controls on AWSのDetective controlsを参照してください。

開発バリューストリームマッピング (DVSM)

ソフトウェア開発ライフサイクルのスピードと品質に悪影響を及ぼす制約を特定し、優先順位を付けるために使用されるプロセス。DVSM は、もともとリーンマニュファクチャリング・プラクティスのために設計されたバリューストリームマッピング・プロセスを拡張したものです。ソフトウェア開発プロセスを通じて価値を創造し、動かすために必要なステップとチームに焦点を当てています。

デジタルツイン

建物、工場、産業機器、生産ラインなど、現実世界のシステムを仮想的に表現したものです。デジタルツインは、予知保全、リモートモニタリング、生産最適化をサポートします。

ディメンションテーブル

スタースキーマでは、ファクトテーブル内の量的データに関するデータ属性を含む小さなテーブル。ディメンションテーブル属性は通常、テキストフィールドまたはテキストのように動作する

離散数値です。これらの属性は、クエリの制約、フィルタリング、結果セットのラベル付けに一般的に使用されます。

ディザスタ

ワークロードまたはシステムが、導入されている主要な場所でのビジネス目標の達成を妨げるイベント。これらのイベントは、自然災害、技術的障害、または意図しない設定ミスやマルウェア攻撃などの人間の行動の結果である場合があります。

ディザスタリカバリ (DR)

災害によるダウンタイムとデータ損失を最小限に抑えるために使用する戦略とプロセス。詳細については、 AWS Well-Architected フレームワークの「 でのワークロードのディザスタリカバリ AWS: クラウドでのリカバリ」を参照してください。

DML

「データベース操作言語」を参照してください。

ドメイン駆動型設計

各コンポーネントが提供している変化を続けるドメイン、またはコアビジネス目標にコンポーネントを接続して、複雑なソフトウェアシステムを開発するアプローチ。この概念は、エリック・エヴァンスの著書、Domain-Driven Design: Tackling Complexity in the Heart of Software (ドメイン駆動設計:ソフトウェアの中心における複雑さへの取り組み) で紹介されています (ボストン: Addison-Wesley Professional、2003)。strangler fig パターンでドメイン駆動型設計を使用する方法の詳細については、コンテナと Amazon API Gateway を使用して、従来の Microsoft ASP.NET (ASMX) ウェブサービスを段階的にモダナイズ を参照してください。

DR

「ディザスタリカバリ」を参照してください。

ドリフト検出

ベースライン設定からの偏差を追跡します。たとえば、 AWS CloudFormation を使用してシステムリソースのドリフトを検出したり、 を使用して AWS Control Tower 、ガバナンス要件のコンプライアンスに影響を与える可能性のあるランディングゾーンの変更を検出したりできます。

DVSM

「開発値ストリームマッピング」を参照してください。

E

EDA

「探索的データ分析」を参照してください。

EDI

「電子データ交換」を参照してください。

エッジコンピューティング

IoT ネットワークのエッジにあるスマートデバイスの計算能力を高めるテクノロジー。クラウドコンピューティングと比較すると、エッジコンピューティングは通信レイテンシーを短縮し、応答時間を短縮できます。

電子データ交換 (EDI)

組織間のビジネスドキュメントの自動交換。詳細については、「電子データ交換とは」を参照してください。

暗号化

人間が読み取り可能なプレーンテキストデータを暗号文に変換するコンピューティングプロセス。

暗号化キー

暗号化アルゴリズムが生成した、ランダム化されたビットからなる暗号文字列。キーの長さは決まっておらず、各キーは予測できないように、一意になるように設計されています。

エンディアン

コンピュータメモリにバイトが格納される順序。ビッグエンディアンシステムでは、最上位バイトが最初に格納されます。リトルエンディアンシステムでは、最下位バイトが最初に格納されます。

エンドポイント

「サービスエンドポイント」を参照してください。

エンドポイントサービス

仮想プライベートクラウド (VPC) 内でホストして、他のユーザーと共有できるサービス。を使用してエンドポイントサービスを作成し AWS PrivateLink 、他の AWS アカウント または AWS Identity and Access Management (IAM) プリンシパルにアクセス許可を付与できます。これら

のアカウントまたはプリンシパルは、インターフェイス VPC エンドポイントを作成することで、エンドポイントサービスにプライベートに接続できます。詳細については、Amazon Virtual Private Cloud (Amazon VPC) ドキュメントの「エンドポイントサービスを作成する」を参照してください。

エンタープライズリソースプランニング (ERP)

エンタープライズの主要なビジネスプロセス (会計、MES、プロジェクト管理など) を自動化および管理するシステム。

エンベロープ暗号化

暗号化キーを、別の暗号化キーを使用して暗号化するプロセス。詳細については、 AWS Key Management Service （ AWS KMS) ドキュメントの「エンベロープ暗号化」を参照してください。

環境

実行中のアプリケーションのインスタンス。クラウドコンピューティングにおける一般的な環境の種類は以下のとおりです。

- 開発環境 — アプリケーションのメンテナンスを担当するコアチームのみが使用できる、実行中のアプリケーションのインスタンス。開発環境は、上位の環境に昇格させる変更をテストするときに使用します。このタイプの環境は、テスト環境と呼ばれることもあります。
- 下位環境 — 初期ビルドやテストに使用される環境など、アプリケーションのすべての開発環境。
- 本番環境 — エンドユーザーがアクセスできる、実行中のアプリケーションのインスタンス。CI/CD パイプラインでは、本番環境が最後のデプロイ環境になります。
- 上位環境 — コア開発チーム以外のユーザーがアクセスできるすべての環境。これには、本番環境、本番前環境、ユーザー承認テスト環境などが含まれます。

エピック

アジャイル方法論で、お客様の作業の整理と優先順位付けに役立つ機能カテゴリ。エピックでは、要件と実装タスクの概要についてハイレベルな説明を提供します。例えば、 AWS CAF セキュリティエピックには、ID とアクセスの管理、検出コントロール、インフラストラクチャセキュリティ、データ保護、インシデント対応が含まれます。 AWS 移行戦略のエピックの詳細については、プログラム実装ガイド を参照してください。

ERP

「エンタープライズリソース計画」を参照してください。

探索的データ分析 (EDA)

データセットを分析してその主な特性を理解するプロセス。お客様は、データを収集または集計してから、パターンの検出、異常の検出、および前提条件のチェックのための初期調査を実行します。EDA は、統計の概要を計算し、データの可視化を作成することによって実行されます。

F

ファクトテーブル

星スキーマの中央テーブル。事業運営に関する量的データを保存します。通常、ファクトテーブルには、メジャーを含む列とディメンションテーブルへの外部キーを含む列の 2 つのタイプの列が含まれます。

フェイルファスト

開発ライフサイクルを短縮するために頻繁で段階的なテストを使用する哲学。これはアジャイルアプローチの重要な部分です。

障害分離の境界

では AWS クラウド、アベイラビリティーゾーン AWS リージョン、コントロールプレーン、データプレーンなどの境界で、障害の影響を制限し、ワークロードの耐障害性を向上させるのに役立ちます。詳細については、AWS 「障害分離境界」を参照してください。

機能ブランチ

「ブランチ」を参照してください。

特徴量

お客様が予測に使用する入力データ。例えば、製造コンテキストでは、特徴量は製造ラインから定期的にキャプチャされるイメージの可能性もあります。

特徴量重要度

モデルの予測に対する特徴量の重要性。これは通常、Shapley Additive Deskonations (SHAP) や積分勾配など、さまざまな手法で計算できる数値スコアで表されます。詳細については、「 を使用した機械学習モデルの解釈可能性 AWS」を参照してください。

機能変換

追加のソースによるデータのエンリッチ化、値のスケーリング、単一のデータフィールドからの複数の情報セットの抽出など、機械学習プロセスのデータを最適化すること。これにより、機械

学習モデルはデータの恩恵を受けることができます。例えば、「2021-05-27 00:15:37」の日付を「2021 年」、「5 月」、「木」、「15」に分解すると、学習アルゴリズムがさまざまなデータコンポーネントに関連する微妙に異なるパターンを学習するのに役立ちます。

数ショットプロンプト

同様のタスクの実行を求める前に、タスクと必要な出力を示す少数の例を LLM に提供します。この手法は、プロンプトに埋め込まれた例 (ショット) からモデルが学習するコンテキスト内学習のアプリケーションです。少数ショットプロンプトは、特定のフォーマット、推論、またはドメインの知識を必要とするタスクに効果的です。「ゼロショットプロンプト」も参照してください。

FGAC

「きめ細かなアクセスコントロール」を参照してください。

きめ細かなアクセス制御 (FGAC)

複数の条件を使用してアクセス要求を許可または拒否すること。

フラッシュカット移行

段階的なアプローチを使用する代わりに、変更データキャプチャによる継続的なデータレプリケーションを使用して、可能な限り短時間でデータを移行するデータベース移行方法。目的はダウンタイムを最小限に抑えることです。

FM

「基盤モデル」を参照してください。

基盤モデル (FM)

一般化データとラベル付けされていないデータの大規模なデータセットでトレーニングされている大規模な深層学習ニューラルネットワーク。FMs は、言語の理解、テキストと画像の生成、自然言語の会話など、さまざまな一般的なタスクを実行できます。詳細については、「基盤モデルとは」を参照してください。

G

生成 AI

大量のデータでトレーニングされ、シンプルなテキストプロンプトを使用してイメージ、動画、テキスト、オーディオなどの新しいコンテンツやアーティファクトを作成できる AI モデルのサブセット。詳細については、「生成 AI とは」を参照してください。

ジオブロッキング

地理的制限を参照してください。

地理的制限 (ジオブロッキング)

特定の国のユーザーがコンテンツ配信にアクセスできないようにするための、Amazon CloudFront のオプション。アクセスを許可する国と禁止する国は、許可リストまたは禁止リストを使って指定します。詳細については、CloudFront ドキュメントのコンテンツの地理的ディストリビューションの制限を参照してください。

Gitflow ワークフロー

下位環境と上位環境が、ソースコードリポジトリでそれぞれ異なるブランチを使用する方法。Gitflow ワークフローはレガシーと見なされ、トランクベースのワークフローはモダンで推奨されるアプローチです。

ゴールデンイメージ

そのシステムまたはソフトウェアの新しいインスタンスをデプロイするためのテンプレートとして使用されるシステムまたはソフトウェアのスナップショット。例えば、製造では、ゴールデンイメージを使用して複数のデバイスにソフトウェアをプロビジョニングし、デバイス製造オペレーションの速度、スケーラビリティ、生産性を向上させることができます。

グリーンフィールド戦略

新しい環境に既存のインフラストラクチャが存在しないこと。システムアーキテクチャにグリーンフィールド戦略を導入する場合、既存のインフラストラクチャ (別名ブラウンフィールド) との互換性の制約を受けることなく、あらゆる新しいテクノロジーを選択できます。既存のインフラストラクチャを拡張している場合は、ブラウンフィールド戦略とグリーンフィールド戦略を融合させることもできます。

ガードレール

組織単位 (OU) 全般のリソース、ポリシー、コンプライアンスを管理するのに役立つ概略的なルール。予防ガードレールは、コンプライアンス基準に一致するようにポリシーを実施します。これらは、サービスコントロールポリシーと IAM アクセス許可の境界を使用して実装されます。検出ガードレールは、ポリシー違反やコンプライアンス上の問題を検出し、修復のためのアラートを発信します。これらは AWS Config、、Amazon GuardDuty AWS Security Hub、、 AWS Trusted Advisor Amazon Inspector、およびカスタム AWS Lambda チェックを使用して実装されます。

H

HA

「高可用性」を参照してください。

異種混在データベースの移行

別のデータベースエンジンを使用するターゲットデータベースへお客様の出典データベースの移行 (例えば、Oracle から Amazon Aurora)。異種間移行は通常、アーキテクチャの再設計作業の一部であり、スキーマの変換は複雑なタスクになる可能性があります。AWS は、スキーマの変換に役立つ AWS SCTを提供します。

ハイアベイラビリティ (HA)

課題や災害が発生した場合に、介入なしにワークロードを継続的に運用できること。HA システムは、自動的にフェイルオーバーし、一貫して高品質のパフォーマンスを提供し、パフォーマンスへの影響を最小限に抑えながらさまざまな負荷や障害を処理するように設計されています。

ヒストリアンのモダナイゼーション

製造業のニーズによりよく応えるために、オペレーションテクノロジー (OT) システムをモダナイズし、アップグレードするためのアプローチ。ヒストリアンは、工場内のさまざまなソースからデータを収集して保存するために使用されるデータベースの一種です。

ホールドアウトデータ

機械学習モデルのトレーニングに使用されるデータセットから保留される、ラベル付きの履歴データの一部。モデル予測をホールドアウトデータと比較することで、ホールドアウトデータを使用してモデルのパフォーマンスを評価できます。

同種データベースの移行

お客様の出典データベースを、同じデータベースエンジンを共有するターゲットデータベース (Microsoft SQL Server から Amazon RDS for SQL Server など) に移行する。同種間移行は、通常、リホストまたはリプラットフォーム化の作業の一部です。ネイティブデータベースユーティリティを使用して、スキーマを移行できます。

ホットデータ

リアルタイムデータや最近の翻訳データなど、頻繁にアクセスされるデータ。通常、このデータには高速なクエリ応答を提供する高性能なストレージ階層またはクラスが必要です。

ホットフィックス

本番環境の重大な問題を修正するために緊急で配布されるプログラム。緊急性が高いため、通常の DevOps のリリースワークフローからは外れた形で実施されます。

ハイパーケア期間

カットオーバー直後、移行したアプリケーションを移行チームがクラウドで管理、監視して問題に対処する期間。通常、この期間は 1～4 日です。ハイパーケア期間が終了すると、アプリケーションに対する責任は一般的に移行チームからクラウドオペレーションチームに移ります。

I

IaC

「Infrastructure as Code」を参照してください。

ID ベースのポリシー

AWS クラウド 環境内のアクセス許可を定義する 1 つ以上の IAM プリンシパルにアタッチされたポリシー。

アイドル状態のアプリケーション

90 日間の平均的な CPU およびメモリ使用率が 5～20% のアプリケーション。移行プロジェクトでは、これらのアプリケーションを廃止するか、オンプレミスに保持するのが一般的です。

IIoT

「産業用モノのインターネット」を参照してください。

イミュータブルインフラストラクチャ

既存のインフラストラクチャを更新、パッチ適用、または変更する代わりに、本番環境のワークロード用に新しいインフラストラクチャをデプロイするモデル。イミュータブルインフラストラクチャは、本質的にミュータブルインフラストラクチャよりも一貫性、信頼性、予測性が高くなります。詳細については、 AWS 「 Well-Architected フレームワーク」の「イミュータブルインフラストラクチャを使用したデプロイ」のベストプラクティスを参照してください。

インバウンド (受信) VPC

AWS マルチアカウントアーキテクチャでは、アプリケーションの外部からネットワーク接続を受け入れ、検査し、ルーティングする VPC。AWS Security Reference Architecture では、アプリ

ケーションとより広範なインターネット間の双方向のインターフェイスを保護するために、インバウンド、アウトバウンド、インスペクションの各 VPC を使用してネットワークアカウントを設定することを推奨しています。

増分移行

アプリケーションを 1 回ですべてカットオーバーするのではなく、小さい要素に分けて移行するカットオーバー戦略。例えば、最初は少数のマイクロサービスまたはユーザーのみを新しいシステムに移行する場合があります。すべてが正常に機能することを確認できたら、残りのマイクロサービスやユーザーを段階的に移行し、レガシーシステムを廃止できるようにします。この戦略により、大規模な移行に伴うリスクが軽減されます。

インダストリー 4.0

2016 年に Klaus Schwab によって導入された用語で、接続、リアルタイムデータ、オートメーション、分析、AI/ML の進歩によるビジネスプロセスのモダナイゼーションを指します。

インフラストラクチャ

アプリケーションの環境に含まれるすべてのリソースとアセット。

Infrastructure as Code (IaC)

アプリケーションのインフラストラクチャを一連の設定ファイルを使用してプロビジョニングし、管理するプロセス。IaC は、新しい環境を再現可能で信頼性が高く、一貫性のあるものにするため、インフラストラクチャを一元的に管理し、リソースを標準化し、スケールを迅速に行えるように設計されています。

産業分野における IoT (IIoT)

製造、エネルギー、自動車、ヘルスケア、ライフサイエンス、農業などの産業部門におけるインターネットに接続されたセンサーやデバイスの使用。詳細については、「Building an industrial Internet of Things (IIoT) digital transformation strategy」を参照してください。

インスペクション VPC

AWS マルチアカウントアーキテクチャでは、VPC (同一または異なる AWS リージョン) 、インターネット、オンプレミスネットワーク間のネットワークトラフィックの検査を管理する一元化された VPCs。AWS Security Reference Architecture では、アプリケーションとより広範なインターネット間の双方向のインターフェイスを保護するために、インバウンド、アウトバウンド、インスペクションの各 VPC を使用してネットワークアカウントを設定することを推奨しています。

IoT

インターネットまたはローカル通信ネットワークを介して他のデバイスやシステムと通信する、センサーまたはプロセッサが組み込まれた接続済み物理オブジェクトのネットワーク。詳細については、「IoT とは」を参照してください。

解釈可能性

機械学習モデルの特性で、モデルの予測がその入力にどのように依存するかを人間が理解できる度合いを表します。詳細については、「 を使用した機械学習モデルの解釈可能性 AWS」を参照してください。

IoT

「モノのインターネット」を参照してください。

IT 情報ライブラリ (ITIL)

IT サービスを提供し、これらのサービスをビジネス要件に合わせるための一連のベストプラクティス。ITIL は ITSM の基盤を提供します。

IT サービス管理 (ITSM)

組織の IT サービスの設計、実装、管理、およびサポートに関連する活動。クラウドオペレーションと ITSM ツールの統合については、オペレーション統合ガイド を参照してください。

ITIL

「IT 情報ライブラリ」を参照してください。

ITSM

「IT サービス管理」を参照してください。

L

ラベルベースアクセス制御 (LBAC)

強制アクセス制御 (MAC) の実装で、ユーザーとデータ自体にそれぞれセキュリティラベル値が明示的に割り当てられます。ユーザーセキュリティラベルとデータセキュリティラベルが交差する部分によって、ユーザーに表示される行と列が決まります。

ランディングゾーン

ランディングゾーンは、スケーラブルで安全な、適切に設計されたマルチアカウント AWS 環境です。これは、組織がセキュリティおよびインフラストラクチャ環境に自信を持ってワークロー

ドとアプリケーションを迅速に起動してデプロイできる出発点です。ランディングゾーンの詳細については、安全でスケーラブルなマルチアカウント AWS 環境のセットアップ を参照してください。

大規模言語モデル (LLM)

大量のデータに対して事前トレーニングされた深層学習 AI モデル。LLM は、質問への回答、ドキュメントの要約、テキストの他の言語への翻訳、文の完了など、複数のタスクを実行できます。詳細については、LLMs」を参照してください。

大規模な移行

300 台以上のサーバの移行。

LBAC

「ラベルベースのアクセスコントロール」を参照してください。

最小特権

タスクの実行には必要最低限の権限を付与するという、セキュリティのベストプラクティス。詳細については、IAM ドキュメントの最小特権アクセス許可を適用するを参照してください。

リフトアンドシフト

「7 Rs」を参照してください。

リトルエンディアンシステム

最下位バイトを最初に格納するシステム。エンディアン性も参照してください。

LLM

「大規模言語モデル」を参照してください。

下位環境

「環境」を参照してください。

M

機械学習 (ML)

パターン認識と学習にアルゴリズムと手法を使用する人工知能の一種。ML は、モノのインターネット (IoT) データなどの記録されたデータを分析して学習し、パターンに基づく統計モデルを生成します。詳細については、「機械学習」を参照してください。

メインブランチ

「ブランチ」を参照してください。

マルウェア

コンピュータのセキュリティまたはプライバシーを侵害するように設計されたソフトウェア。マルウェアは、コンピュータシステムの中断、機密情報の漏洩、不正アクセスにつながる可能性があります。マルウェアの例としては、ウイルス、ワーム、ランサムウェア、トロイの木馬、スパイウェア、キーロガーなどがあります。

マネージドサービス

AWS のサービス はインフラストラクチャレイヤー、オペレーティングシステム、プラットフォーム AWS を運用し、エンドポイントにアクセスしてデータを保存および取得します。Amazon Simple Storage Service (Amazon S3) と Amazon DynamoDB は、マネージドサービスの例です。これらは抽象化されたサービスとも呼ばれます。

製造実行システム (MES)

生産プロセスを追跡、モニタリング、文書化、制御するためのソフトウェアシステムで、原材料を工場の完成製品に変換します。

MAP

「移行促進プログラム」を参照してください。

メカニズム

ツールを作成し、ツールの導入を推進し、調整を行うために結果を検査する完全なプロセス。メカニズムは、動作中にそれ自体を強化して改善するサイクルです。詳細については、 AWS 「 Well-Architected フレームワーク」の「メカニズムの構築」を参照してください。

メンバーアカウント

組織の一部である管理アカウント AWS アカウント 以外のすべて AWS Organizations。アカウントが組織のメンバーになることができるのは、一度に 1 つのみです。

MES

「製造実行システム」を参照してください。

メッセージキューイングテレメトリトランスポート (MQTT)

リソースに制約のある IoT デバイス用の、パブリッシュ/サブスクライブパターンに基づく軽量 machine-to-machine (M2M) 通信プロトコル。

マイクロサービス

明確に定義された API を介して通信し、通常は小規模な自己完結型のチームが所有する、小規模で独立したサービスです。例えば、保険システムには、販売やマーケティングなどのビジネス機能、または購買、請求、分析などのサブドメインにマッピングするマイクロサービスが含まれる場合があります。マイクロサービスの利点には、俊敏性、柔軟なスケーリング、容易なデプロイ、再利用可能なコード、回復力などがあります。詳細については、AWS「サーバーレスサービスを使用したマイクロサービスの統合」を参照してください。

マイクロサービスアーキテクチャ

各アプリケーションプロセスをマイクロサービスとして実行する独立したコンポーネントを使用してアプリケーションを構築するアプローチ。これらのマイクロサービスは、軽量 API を使用して、明確に定義されたインターフェイスを介して通信します。このアーキテクチャの各マイクロサービスは、アプリケーションの特定の機能に対する需要を満たすように更新、デプロイ、およびスケーリングできます。詳細については、「 でのマイクロサービスの実装 AWS」を参照してください。

Migration Acceleration Program (MAP)

組織がクラウドに移行するための強力な運用基盤を構築し、移行の初期コストを相殺するのに役立つコンサルティングサポート、トレーニング、サービスを提供する AWS プログラム。MAP には、組織的な方法でレガシー移行を実行するための移行方法論と、一般的な移行シナリオを自動化および高速化する一連のツールが含まれています。

大規模な移行

アプリケーションポートフォリオの大部分を次々にクラウドに移行し、各ウェーブでより多くのアプリケーションを高速に移動させるプロセス。この段階では、以前の段階から学んだベストプラクティスと教訓を使用して、移行ファクトリー チーム、ツール、プロセスのうち、オートメーションとアジャイルデリバリーによってワークロードの移行を合理化します。これは、AWS 移行戦略 の第 3 段階です。

移行ファクトリー

自動化された俊敏性のあるアプローチにより、ワークロードの移行を合理化する部門横断的なチーム。移行ファクトリーチームには、通常、運用、ビジネスアナリストおよび所有者、移行エンジニア、デベロッパー、およびスプリントで作業する DevOps プロフェッショナルが含まれます。エンタープライズアプリケーションポートフォリオの 20～50% は、ファクトリーのアプローチによって最適化できる反復パターンで構成されています。詳細については、このコンテンツセットの移行ファクトリーに関する解説とCloud Migration Factory ガイドを参照してください。

移行メタデータ

移行を完了するために必要なアプリケーションおよびサーバーに関する情報。移行パターンごとに、異なる一連の移行メタデータが必要です。移行メタデータの例としては、ターゲットサブネット、セキュリティグループ、 AWS アカウントなどがあります。

移行パターン

移行戦略、移行先、および使用する移行アプリケーションまたはサービスを詳述する、反復可能な移行タスク。例: AWS Application Migration Service を使用して Amazon EC2 への移行をリホストします。

Migration Portfolio Assessment (MPA)

に移行するためのビジネスケースを検証するための情報を提供するオンラインツール AWS クラウド。MPA は、詳細なポートフォリオ評価 (サーバーの適切なサイジング、価格設定、TCO 比較、移行コスト分析) および移行プラン (アプリケーションデータの分析とデータ収集、アプリケーションのグループ化、移行の優先順位付け、およびウェーブプランニング) を提供します。MPA ツール (ログインが必要) は、すべての AWS コンサルタントと APN パートナーコンサルタントが無料で利用できます。

移行準備状況評価 (MRA)

AWS CAF を使用して、組織のクラウド準備状況に関するインサイトを取得し、長所と短所を特定し、特定されたギャップを埋めるためのアクションプランを構築するプロセス。詳細については、移行準備状況ガイド を参照してください。MRA は、AWS 移行戦略の第一段階です。

移行戦略

ワークロードを に移行するために使用するアプローチ AWS クラウド。詳細については、この用語集の「7 Rs エントリ」と「組織を動員して大規模な移行を加速する」を参照してください。

ML

???「機械学習」を参照してください。

モダナイゼーション

古い (レガシーまたはモノリシック) アプリケーションとそのインフラストラクチャをクラウド内の俊敏で弾力性のある高可用性システムに変換して、コストを削減し、効率を高め、イノベーションを活用します。詳細については、「」の「アプリケーションをモダナイズするための戦略 AWS クラウド」を参照してください。

モダナイゼーション準備状況評価

組織のアプリケーションのモダナイゼーションの準備状況を判断し、利点、リスク、依存関係を特定し、組織がこれらのアプリケーションの将来の状態をどの程度適切にサポートできるかを決定するのに役立つ評価。評価の結果として、ターゲットアーキテクチャのブループリント、モダナイゼーションプロセスの開発段階とマイルストーンを詳述したロードマップ、特定されたギャップに対処するためのアクションプランが得られます。詳細については、『』の「アプリケーションのモダナイゼーション準備状況の評価 AWS クラウド」を参照してください。

モノリシックアプリケーション (モノリス)

緊密に結合されたプロセスを持つ単一のサービスとして実行されるアプリケーション。モノリシックアプリケーションにはいくつかの欠点があります。1 つのアプリケーション機能エクスペリエンスの需要が急増する場合は、アーキテクチャ全体をスケーリングする必要があります。モノリシックアプリケーションの特徴を追加または改善することは、コードベースが大きくなると複雑になります。これらの問題に対処するには、マイクロサービスアーキテクチャを使用できます。詳細については、モノリスをマイクロサービスに分解する を参照してください。

MPA

「移行ポートフォリオ評価」を参照してください。

MQTT

「Message Queuing Telemetry Transport」を参照してください。

多クラス分類

複数のクラスの予測を生成するプロセス (2 つ以上の結果の 1 つを予測します)。例えば、機械学習モデルが、「この製品は書籍、自動車、電話のいずれですか?」 または、「このお客様にとって最も関心のある商品のカテゴリはどれですか?」と聞くかもしれません。

ミュータブルインフラストラクチャ

本番ワークロードの既存のインフラストラクチャを更新および変更するモデル。Well-Architected AWS フレームワークでは、一貫性、信頼性、予測可能性を向上させるために、イミュータブルインフラストラクチャの使用をベストプラクティスとして推奨しています。

O

OAC

「オリジンアクセスコントロール」を参照してください。

OAI

「オリジンアクセスアイデンティティ」を参照してください。

OCM

「組織変更管理」を参照してください。

オフライン移行

移行プロセス中にソースワークロードを停止させる移行方法。この方法はダウンタイムが長くなるため、通常は重要ではない小規模なワークロードに使用されます。

OI

「 オペレーションの統合」を参照してください。

OLA

「運用レベルの契約」を参照してください。

オンライン移行

ソースワークロードをオフラインにせずにターゲットシステムにコピーする移行方法。ワークロードに接続されているアプリケーションは、移行中も動作し続けることができます。この方法はダウンタイムがゼロから最小限で済むため、通常は重要な本番稼働環境のワークロードに使用されます。

OPC-UA

「Open Process Communications - Unified Architecture」を参照してください。

オープンプロセス通信 - 統合アーキテクチャ (OPC-UA)

産業用オートメーション用のmachine-to-machine (M2M) 通信プロトコル。OPC-UA は、データの暗号化、認証、認可スキームを備えた相互運用性標準を提供します。

オペレーショナルレベルアグリーメント (OLA)

サービスレベルアグリーメント (SLA) をサポートするために、どの機能的 IT グループが互いに提供することを約束するかを明確にする契約。

運用準備状況レビュー (ORR)

インシデントや潜在的な障害の理解、評価、防止、または範囲の縮小に役立つ質問とそれに関連するベストプラクティスのチェックリスト。詳細については、 AWS Well-Architected フレームワークの「Operational Readiness Reviews (ORR)」を参照してください。

運用テクノロジー (OT)

産業オペレーション、機器、インフラストラクチャを制御するために物理環境と連携するハードウェアおよびソフトウェアシステム。製造では、OT と情報技術 (IT) システムの統合が、Industry 4.0 変換の主な焦点です。

オペレーション統合 (OI)

クラウドでオペレーションをモダナイズするプロセスには、準備計画、オートメーション、統合が含まれます。詳細については、オペレーション統合ガイド を参照してください。

組織の証跡

組織 AWS アカウント 内のすべての のすべてのイベント AWS CloudTrail をログに記録する、 によって作成された証跡 AWS Organizations。証跡は、組織に含まれている各 AWS アカウント に作成され、各アカウントのアクティビティを追跡します。詳細については、CloudTrail ドキュメントの組織の証跡の作成を参照してください。

組織変更管理 (OCM)

人材、文化、リーダーシップの観点から、主要な破壊的なビジネス変革を管理するためのフレームワーク。OCM は、変化の導入を加速し、移行問題に対処し、文化や組織の変化を推進することで、組織が新しいシステムと戦略の準備と移行するのを支援します。 AWS 移行戦略では、クラウド導入プロジェクトに必要な変化のスピードのため、このフレームワークは人材アクセラレーションと呼ばれます。詳細については、OCM ガイド を参照してください。

オリジンアクセスコントロール (OAC)

Amazon Simple Storage Service (Amazon S3) コンテンツを保護するための、CloudFront のアクセス制限の強化オプション。OAC は AWS リージョン、すべての S3 バケット、 AWS KMS (SSE-KMS) によるサーバー側の暗号化、S3 バケットへの動的 PUT および DELETEリクエストをサポートします。

オリジンアクセスアイデンティティ (OAI)

CloudFront の、Amazon S3 コンテンツを保護するためのアクセス制限オプション。OAI を使用すると、CloudFront が、Amazon S3 に認証可能なプリンシパルを作成します。認証されたプリンシパルは、S3 バケット内のコンテンツに、特定の CloudFront ディストリビューションを介してのみアクセスできます。OACも併せて参照してください。OAC では、より詳細な、強化されたアクセスコントロールが可能です。

ORR

「運用準備状況レビュー」を参照してください。

OT

「運用テクノロジー」を参照してください。

アウトバウンド (送信) VPC

AWS マルチアカウントアーキテクチャでは、アプリケーション内から開始されたネットワーク接続を処理する VPC。AWS Security Reference Architecture では、アプリケーションとより広範なインターネット間の双方向のインターフェイスを保護するために、インバウンド、アウトバウンド、インスペクションの各 VPC を使用してネットワークアカウントを設定することを推奨しています。

P

アクセス許可の境界

ユーザーまたはロールが使用できるアクセス許可の上限を設定する、IAM プリンシパルにアタッチされる IAM 管理ポリシー。詳細については、IAM ドキュメントのアクセス許可の境界を参照してください。

個人を特定できる情報 (PII)

直接閲覧した場合、または他の関連データと組み合わせた場合に、個人の身元を合理的に推測するために使用できる情報。PII の例には、氏名、住所、連絡先情報などがあります。

PII

個人を特定できる情報を参照してください。

プレイブック

クラウドでのコアオペレーション機能の提供など、移行に関連する作業を取り込む、事前定義された一連のステップ。プレイブックは、スクリプト、自動ランブック、またはお客様のモダナイズされた環境を運用するために必要なプロセスや手順の要約などの形式をとることができます。

PLC

「プログラム可能なロジックコントローラー」を参照してください。

PLM

「製品ライフサイクル管理」を参照してください。

ポリシー

アクセス許可を定義 (アイデンティティベースのポリシーを参照)、アクセス条件を指定 (リソースベースのポリシーを参照)、または の組織内のすべてのアカウントに対する最大アクセス許可を定義 AWS Organizations （ サービスコントロールポリシーを参照) できるオブジェクト。

多言語の永続性

データアクセスパターンやその他の要件に基づいて、マイクロサービスのデータストレージテクノロジーを個別に選択します。マイクロサービスが同じデータストレージテクノロジーを使用している場合、実装上の問題が発生したり、パフォーマンスが低下する可能性があります。マイクロサービスは、要件に最も適合したデータストアを使用すると、より簡単に実装でき、パフォーマンスとスケーラビリティが向上します。詳細については、マイクロサービスでのデータ永続性の有効化 を参照してください。

ポートフォリオ評価

移行を計画するために、アプリケーションポートフォリオの検出、分析、優先順位付けを行うプロセス。詳細については、「移行準備状況ガイド」を参照してください。

述語

`true` または を返すクエリ条件。一般的に`false`は `WHERE`句にあります。

述語プッシュダウン

転送前にクエリ内のデータをフィルタリングするデータベースクエリ最適化手法。これにより、リレーショナルデータベースから取得して処理する必要があるデータの量が減少し、クエリのパフォーマンスが向上します。

予防的コントロール

イベントの発生を防ぐように設計されたセキュリティコントロール。このコントロールは、ネットワークへの不正アクセスや好ましくない変更を防ぐ最前線の防御です。詳細については、Implementing security controls on AWSのPreventative controlsを参照してください。

プリンシパル

アクション を実行し AWS 、リソースにアクセスできる のエンティティ。このエンティティは通常、、IAM AWS アカウントロール、または ユーザーのルートユーザーです。詳細については、IAM ドキュメントのロールに関する用語と概念内にあるプリンシパルを参照してください。

プライバシーバイデザイン

開発プロセス全体を通じてプライバシーを考慮するシステムエンジニアリングアプローチ。

プライベートホストゾーン

1 つ以上の VPC 内のドメインとそのサブドメインへの DNS クエリに対し、Amazon Route 53 がどのように応答するかに関する情報を保持するコンテナ。詳細については、Route 53 ドキュメントの「プライベートホストゾーンの使用」を参照してください。

プロアクティブコントロール

非準拠リソースのデプロイを防ぐように設計されたセキュリティコントロール。これらのコントロールは、プロビジョニング前にリソースをスキャンします。リソースがコントロールに準拠していない場合、プロビジョニングされません。詳細については、 AWS Control Tower ドキュメントの「 コントロールリファレンスガイド」および「 セキュリティコントロールの実装」の「プロアクティブコントロール」を参照してください。 AWS

製品ライフサイクル管理 (PLM)

設計、開発、発売から成長と成熟まで、製品のデータとプロセスのライフサイクル全体にわたる管理。

本番環境

「環境」を参照してください。

プログラム可能なロジックコントローラー (PLC)

製造では、マシンをモニタリングし、製造プロセスを自動化する、信頼性の高い適応可能なコンピュータです。

プロンプトの連鎖

1 つの LLM プロンプトの出力を次のプロンプトの入力として使用して、より良いレスポンスを生成します。この手法は、複雑なタスクをサブタスクに分割したり、事前レスポンスを繰り返し改善または拡張したりするために使用されます。これにより、モデルのレスポンスの精度と関連性が向上し、より詳細でパーソナライズされた結果が得られます。

仮名化

データセット内の個人識別子をプレースホルダー値に置き換えるプロセス。仮名化は個人のプライバシー保護に役立ちます。仮名化されたデータは、依然として個人データとみなされます。

パブリッシュ/サブスクライブ (pub/sub)

マイクロサービス間の非同期通信を可能にするパターン。スケーラビリティと応答性を向上させます。たとえば、マイクロサービスベースの MES では、マイクロサービスは他のマイクロサー

ビスがサブスクライブできるチャネルにイベントメッセージを発行できます。システムは、公開サービスを変更せずに新しいマイクロサービスを追加できます。

Q

クエリプラン

SQL リレーショナルデータベースシステムのデータにアクセスするために使用される手順などの一連のステップ。

クエリプランのリグレッション

データベースサービスのオプティマイザーが、データベース環境に特定の変更が加えられる前に選択されたプランよりも最適性の低いプランを選択すること。これは、統計、制限事項、環境設定、クエリパラメータのバインディングの変更、およびデータベースエンジンの更新などが原因である可能性があります。

R

RACI マトリックス

責任、説明責任、相談、通知 (RACI) を参照してください。

RAG

「取得拡張生成」を参照してください。

ランサムウェア

決済が完了するまでコンピュータシステムまたはデータへのアクセスをブロックするように設計された、悪意のあるソフトウェア。

RASCI マトリックス

責任、説明責任、相談、情報 (RACI) を参照してください。

RCAC

「行と列のアクセスコントロール」を参照してください。

リードレプリカ

読み取り専用に使用されるデータベースのコピー。クエリをリードレプリカにルーティングして、プライマリデータベースへの負荷を軽減できます。

再設計

「7 Rs」を参照してください。

目標復旧時点 (RPO)

最後のデータリカバリポイントからの最大許容時間です。これにより、最後の回復時点からサービスが中断されるまでの間に許容できるデータ損失の程度が決まります。

目標復旧時間 (RTO)

サービスの中断から復旧までの最大許容遅延時間。

リファクタリング

「7 Rs」を参照してください。

リージョン

地理的エリア内の AWS リソースのコレクション。各 AWS リージョン は、耐障害性、安定性、耐障害性を提供するために、他の とは独立しています。詳細については、AWS リージョン「アカウントで使用できる を指定する」を参照してください。

回帰

数値を予測する機械学習手法。例えば、「この家はどれくらいの値段で売れるでしょうか?」という問題を解決するために、機械学習モデルは、線形回帰モデルを使用して、この家に関する既知の事実 (平方フィートなど) に基づいて家の販売価格を予測できます。

リホスト

「7 Rs」を参照してください。

リリース

デプロイプロセスで、変更を本番環境に昇格させること。

再配置

「7 Rs」を参照してください。

プラットフォーム変更

「7 Rs」を参照してください。

再購入

「7 Rs」を参照してください。

回復性

中断に抵抗または回復するアプリケーションの機能。高可用性とディザスタリカバリは、 で回復性を計画する際の一般的な考慮事項です AWS クラウド。詳細については、AWS クラウド「レジリエンス」を参照してください。

リソースベースのポリシー

Amazon S3 バケット、エンドポイント、暗号化キーなどのリソースにアタッチされたポリシー。このタイプのポリシーは、アクセスが許可されているプリンシパル、サポートされているアクション、その他の満たすべき条件を指定します。

実行責任者、説明責任者、協業先、報告先 (RACI) に基づくマトリックス

移行活動とクラウド運用に関わるすべての関係者の役割と責任を定義したマトリックス。マトリックスの名前は、マトリックスで定義されている責任の種類、すなわち責任 (R) 、説明責任 (A) 、協議 (C) 、情報提供 (I) に由来します。サポート (S) タイプはオプションです。サポートを含めると、そのマトリックスは RASCI マトリックスと呼ばれ、サポートを除外すると RACI マトリックスと呼ばれます。

レスポンシブコントロール

有害事象やセキュリティベースラインからの逸脱について、修復を促すように設計されたセキュリティコントロール。詳細については、Implementing security controls on AWSのResponsive controlsを参照してください。

保持

「7 Rs」を参照してください。

廃止

「7 Rs」を参照してください。

取得拡張生成 (RAG)

LLM がレスポンスを生成する前にトレーニングデータソースの外部にある信頼できるデータソースを参照する生成 AI テクノロジー。たとえば、RAG モデルは、組織のナレッジベースまたはカスタムデータのセマンティック検索を実行する場合があります。詳細については、「RAG とは」を参照してください。

ローテーション

攻撃者が認証情報にアクセスすることをより困難にするために、シークレットを定期的に更新するプロセス。

行と列のアクセス制御 (RCAC)

アクセスルールが定義された、基本的で柔軟な SQL 表現の使用。RCAC は行権限と列マスクで構成されています。

RPO

「目標復旧時点」を参照してください。

RTO

目標復旧時間を参照してください。

ランブック

特定のタスクを実行するために必要な手動または自動化された一連の手順。これらは通常、エラー率の高い反復操作や手順を合理化するために構築されています。

S

SAML 2.0

多くの ID プロバイダー (IdP) が使用しているオープンスタンダード。この機能を使用すると、フェデレーティッドシングルサインオン (SSO) が有効になるため、ユーザーは組織内のすべてのユーザーを IAM で作成しなくても、 AWS Management Console にログインしたり AWS 、 API オペレーションを呼び出すことができます。SAML 2.0 ベースのフェデレーションの詳細については、IAM ドキュメントのSAML 2.0 ベースのフェデレーションについてを参照してください。

SCADA

「監視コントロールとデータ取得」を参照してください。

SCP

「サービスコントロールポリシー」を参照してください。

シークレット

暗号化された形式で保存する AWS Secrets Managerパスワードやユーザー認証情報などの機密情報または制限付き情報。シークレット値とそのメタデータで構成されます。シークレット値は、バイナリ、1 つの文字列、または複数の文字列にすることができます。詳細については、Secrets Manager ドキュメントの「Secrets Manager シークレットの内容」を参照してください。

設計によるセキュリティ

開発プロセス全体でセキュリティを考慮するシステムエンジニアリングアプローチ。

セキュリティコントロール

脅威アクターによるセキュリティ脆弱性の悪用を防止、検出、軽減するための、技術上または管理上のガードレール。セキュリティコントロールには、予防的、検出的、応答的、プロアクティブの 4 つの主なタイプがあります。

セキュリティ強化

アタックサーフェスを狭めて攻撃への耐性を高めるプロセス。このプロセスには、不要になったリソースの削除、最小特権を付与するセキュリティのベストプラクティスの実装、設定ファイル内の不要な機能の無効化、といったアクションが含まれています。

Security Information and Event Management (SIEM) システム

セキュリティ情報管理 (SIM) とセキュリティイベント管理 (SEM) のシステムを組み合わせたツールとサービス。SIEM システムは、サーバー、ネットワーク、デバイス、その他ソースからデータを収集、モニタリング、分析して、脅威やセキュリティ違反を検出し、アラートを発信します。

セキュリティレスポンスの自動化

セキュリティイベントに自動的に応答または修復するように設計された、事前定義されたプログラムされたアクション。これらの自動化は、セキュリティのベストプラクティスを実装するのに役立つ検出的または応答的な AWS セキュリティコントロールとして機能します。自動応答アクションの例としては、VPC セキュリティグループの変更、Amazon EC2 インスタンスへのパッチ適用、認証情報の更新などがあります。

サーバー側の暗号化

送信先にあるデータの、それ AWS のサービス を受け取る による暗号化。

サービスコントロールポリシー (SCP)

AWS Organizationsの組織内の、すべてのアカウントのアクセス許可を一元的に管理するポリシー。SCP は、管理者がユーザーまたはロールに委任するアクションに、ガードレールを定義したり、アクションの制限を設定したりします。SCP は、許可リストまたは拒否リストとして、許可または禁止するサービスやアクションを指定する際に使用できます。詳細については、 AWS Organizations ドキュメントの「サービスコントロールポリシー」を参照してください。

サービスエンドポイント

のエントリポイントの URL AWS のサービス。ターゲットサービスにプログラムで接続するには、エンドポイントを使用します。詳細については、AWS 全般のリファレンスの「AWS のサービス エンドポイント」を参照してください。

サービスレベルアグリーメント (SLA)

サービスのアップタイムやパフォーマンスなど、IT チームがお客様に提供すると約束したものを明示した合意書。

サービスレベルインジケータ (SLI)

エラー率、可用性、スループットなど、サービスのパフォーマンス側面の測定。

サービスレベルの目標 (SLO)

サービスレベルのインジケータによって測定される、サービスの状態を表すターゲットメトリクス。

責任共有モデル

クラウドのセキュリティとコンプライアンス AWS について と共有する責任を説明するモデル。AWS はクラウドのセキュリティを担当しますが、お客様はクラウドのセキュリティを担当します。詳細については、責任共有モデルを参照してください。

SIEM

セキュリティ情報とイベント管理システムを参照してください。

単一障害点 (SPOF)

システムを中断する可能性のあるアプリケーションの 1 つの重要なコンポーネントの障害。

SLA

「サービスレベルの契約」を参照してください。

SLI

「サービスレベルインジケータ」を参照してください。

SLO

「サービスレベルの目標」を参照してください。

スプリットアンドシードモデル

モダナイゼーションプロジェクトのスケーリングと加速のためのパターン。新機能と製品リリースが定義されると、コアチームは解放されて新しい製品チームを作成します。これにより、お客様の組織の能力とサービスの拡張、デベロッパーの生産性の向上、迅速なイノベーションのサポートに役立ちます。詳細については、『』の「アプリケーションをモダナイズするための段階的アプローチ AWS クラウド」を参照してください。

SPOF

単一障害点を参照してください。

スタースキーマ

1 つの大きなファクトテーブルを使用してトランザクションデータまたは測定データを保存し、1 つ以上の小さなディメンションテーブルを使用してデータ属性を保存するデータベース組織構造。この構造は、データウェアハウスまたはビジネスインテリジェンスの目的で使用するように設計されています。

strangler fig パターン

レガシーシステムが廃止されるまで、システム機能を段階的に書き換えて置き換えることにより、モノリシックシステムをモダナイズするアプローチ。このパターンは、宿主の樹木から根を成長させ、最終的にその宿主を包み込み、宿主に取って代わるイチジクのつるを例えています。そのパターンは、モノリシックシステムを書き換えるときのリスクを管理する方法として Martin Fowler により提唱されました。このパターンの適用方法の例については、コンテナと Amazon API Gateway を使用して、従来の Microsoft ASP.NET (ASMX) ウェブサービスを段階的にモダナイズを参照してください。

サブネット

VPC 内の IP アドレスの範囲。サブネットは、1 つのアベイラビリティーゾーンに存在する必要があります。

監視制御とデータ収集 (SCADA)

製造では、ハードウェアとソフトウェアを使用して物理アセットと本番稼働をモニタリングするシステム。

対称暗号化

データの暗号化と復号に同じキーを使用する暗号化のアルゴリズム。

合成テスト

ユーザーとのやり取りをシミュレートして潜在的な問題を検出したり、パフォーマンスをモニタリングしたりする方法でシステムをテストします。Amazon CloudWatch Synthetics を使用して、これらのテストを作成できます。

システムプロンプト

LLM にコンテキスト、指示、またはガイドラインを提供して動作を指示する手法。システムプロンプトは、コンテキストを設定し、ユーザーとのやり取りのルールを確立するのに役立ちます。

T

tags

AWS リソースを整理するためのメタデータとして機能するキーと値のペア。タグは、リソースの管理、識別、整理、検索、フィルタリングに役立ちます。詳細については、「AWS リソースのタグ付け」を参照してください。

ターゲット変数

監督された機械学習でお客様が予測しようとしている値。これは、結果変数 のことも指します。例えば、製造設定では、ターゲット変数が製品の欠陥である可能性があります。

タスクリスト

ランブックの進行状況を追跡するために使用されるツール。タスクリストには、ランブックの概要と完了する必要のある一般的なタスクのリストが含まれています。各一般的なタスクには、推定所要時間、所有者、進捗状況が含まれています。

テスト環境

「環境」を参照してください。

トレーニング

お客様の機械学習モデルに学習するデータを提供すること。トレーニングデータには正しい答えが含まれている必要があります。学習アルゴリズムは入力データ属性をターゲット (お客様が予測したい答え) にマッピングするトレーニングデータのパターンを検出します。これらのパターンをキャプチャする機械学習モデルを出力します。そして、お客様が機械学習モデルを使用して、ターゲットがわからない新しいデータでターゲットを予測できます。

トランジットゲートウェイ

VPC とオンプレミスネットワークを相互接続するために使用できる、ネットワークの中継ハブ。詳細については、 AWS Transit Gateway ドキュメントの「トランジットゲートウェイとは」を参照してください。

トランクベースのワークフロー

デベロッパーが機能ブランチで機能をローカルにビルドしてテストし、その変更をメインブランチにマージするアプローチ。メインブランチはその後、開発環境、本番前環境、本番環境に合わせて順次構築されます。

信頼されたアクセス

ユーザーに代わって AWS Organizations およびそのアカウントで組織内でタスクを実行するために指定したサービスにアクセス許可を付与します。信頼されたサービスは、サービスにリンクされたロールを必要なときに各アカウントに作成し、ユーザーに代わって管理タスクを実行します。詳細については、 ドキュメントの「 を他の AWS のサービス AWS Organizations で使用する AWS Organizations 」を参照してください。

チューニング

機械学習モデルの精度を向上させるために、お客様のトレーニングプロセスの側面を変更する。例えば、お客様が機械学習モデルをトレーニングするには、ラベル付けセットを生成し、ラベルを追加します。これらのステップを、異なる設定で複数回繰り返して、モデルを最適化します。

ツーピザチーム

2 枚のピザで養うことができるくらい小さな DevOps チーム。ツーピザチームの規模では、ソフトウェア開発におけるコラボレーションに最適な機会が確保されます。

U

不確実性

予測機械学習モデルの信頼性を損なう可能性がある、不正確、不完全、または未知の情報を指す概念。不確実性には、次の 2 つのタイプがあります。認識論的不確実性は、限られた、不完全なデータによって引き起こされ、弁論的不確実性 は、データに固有のノイズとランダム性によって引き起こされます。詳細については、深層学習システムにおける不確実性の定量化 ガイドを参照してください。

未分化なタスク

ヘビーリフティングとも呼ばれ、アプリケーションの作成と運用には必要だが、エンドユーザーに直接的な価値をもたらさなかったり、競争上の優位性をもたらしたりしない作業です。未分化なタスクの例としては、調達、メンテナンス、キャパシティプランニングなどがあります。

上位環境

???「環境」を参照してください。

V

バキューミング

ストレージを再利用してパフォーマンスを向上させるために、増分更新後にクリーンアップを行うデータベースのメンテナンス操作。

バージョンコントロール

リポジトリ内のソースコードへの変更など、変更を追跡するプロセスとツール。

VPC ピアリング

プライベート IP アドレスを使用してトラフィックをルーティングできる、2 つの VPC 間の接続。詳細については、Amazon VPC ドキュメントの「VPC ピア機能とは」を参照してください。

脆弱性

システムのセキュリティを脅かすソフトウェアまたはハードウェアの欠陥。

W

ウォームキャッシュ

頻繁にアクセスされる最新の関連データを含むバッファキャッシュ。データベースインスタンスはバッファキャッシュから、メインメモリまたはディスクからよりも短い時間で読み取りを行うことができます。

ウォームデータ

アクセス頻度の低いデータ。この種類のデータをクエリする場合、通常は適度に遅いクエリでも問題ありません。

ウィンドウ関数

現在のレコードに何らかの形で関連する行のグループに対して計算を実行する SQL 関数。ウィンドウ関数は、移動平均の計算や、現在の行の相対位置に基づく行の値へのアクセスなどのタスクの処理に役立ちます。

ワークロード

ビジネス価値をもたらすリソースとコード (顧客向けアプリケーションやバックエンドプロセスなど) の総称。

ワークストリーム

特定のタスクセットを担当する移行プロジェクト内の機能グループ。各ワークストリームは独立していますが、プロジェクト内の他のワークストリームをサポートしています。たとえば、ポートフォリオワークストリームは、アプリケーションの優先順位付け、ウェーブ計画、および移行メタデータの収集を担当します。ポートフォリオワークストリームは、これらの設備を移行ワークストリームで実現し、サーバーとアプリケーションを移行します。

WORM

「Write Once」、「Read Many」を参照してください。

WQF

AWS 「ワークロード認定フレームワーク」を参照してください。

Write Once, Read Many (WORM)

データを 1 回書き込み、データの削除や変更を防ぐストレージモデル。承認されたユーザーは、必要な回数だけデータを読み取ることができますが、変更することはできません。このデータストレージインフラストラクチャはイミュータブルと見なされます。

Z

ゼロデイエクスプロイト

ゼロデイ脆弱性を利用する攻撃、通常はマルウェア。

ゼロデイ脆弱性

実稼働システムにおける未解決の欠陥または脆弱性。脅威アクターは、このような脆弱性を利用してシステムを攻撃する可能性があります。開発者は、よく攻撃の結果で脆弱性に気付きます。

ゼロショットプロンプト

LLM にタスクを実行する手順を提供しますが、タスクのガイドに役立つ例 (ショット) はありません。LLM は、事前トレーニング済みの知識を使用してタスクを処理する必要があります。ゼロショットプロンプトの有効性は、タスクの複雑さとプロンプトの品質によって異なります。「数ショットプロンプト」も参照してください。

ゾンビアプリケーション

平均 CPU およびメモリ使用率が 5% 未満のアプリケーション。移行プロジェクトでは、これらのアプリケーションを廃止するのが一般的です。