ForkMe!

ご利用マニュアル

3.レポート活用

バージョン１

INDEX

3-1.レポート活用（設計の全体像を把握する）

レポートの概要タブは、設計の特徴を素早く把握できるよう構成されています。

ポイント１）設計の複雑性をチェックする

本システムはクラウドの利用要件を、誰が（アクター）、何を（リソース）、どのように利用するか（コンテクスト）という観点で整理します。よってそれらの数を見ることで、概ね**システムの複雑性**が確認できます。

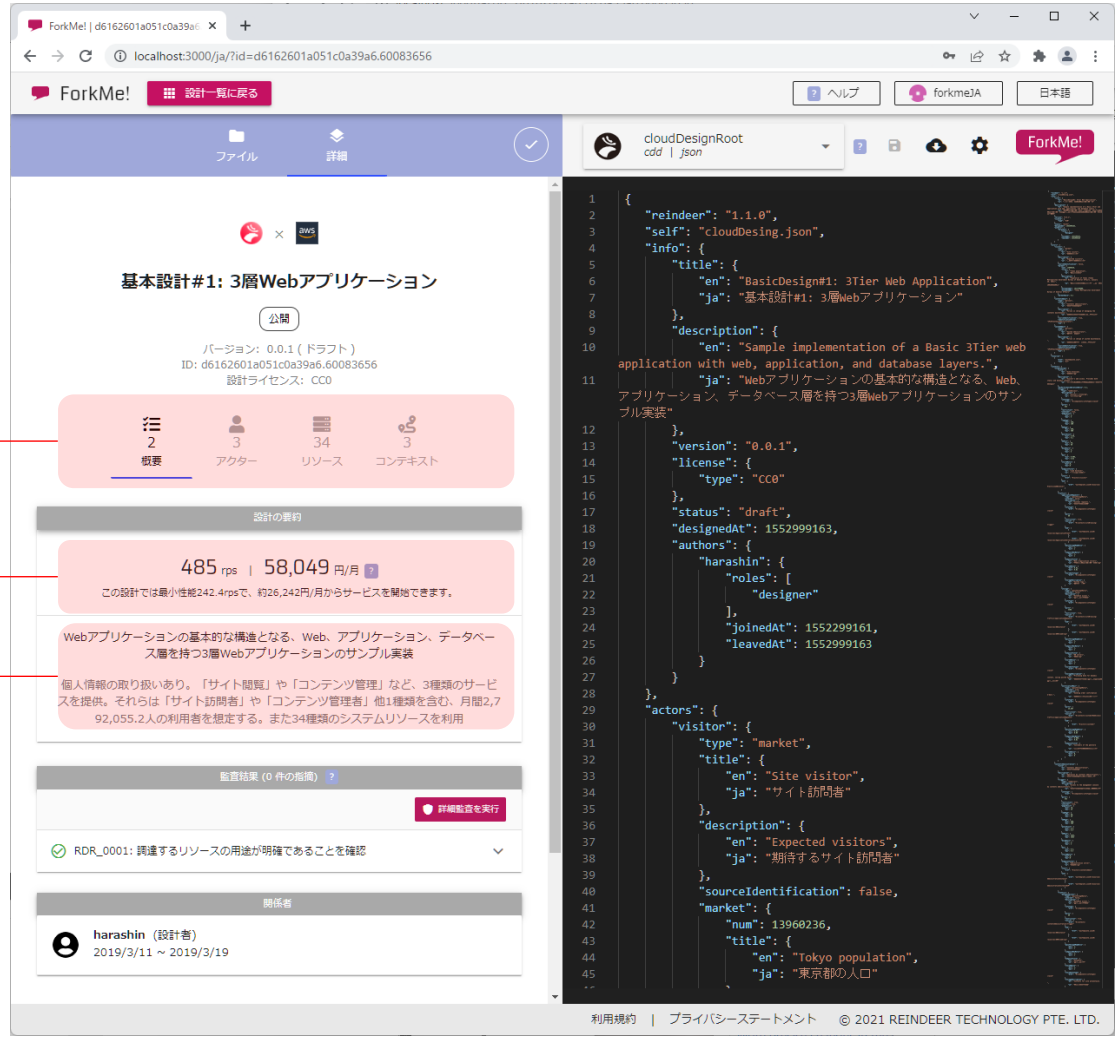
・アクターが多い：多様な利用者に機能提供するシステム
・リソースが多い：多様なクラウドリソースを利用するシステム
・コンテキストが多い：多様な機能を提供するシステム

ポイント２）期待性能とコストをチェックする

本システムは期待性能（rps: 秒間リクエスト数）と必要コスト（月間のクラウド利用費）を自動算定します。本システムは対応リソースを順次拡充していますが、実際の値とは異なる点にご注意ください。
よってこれらの値は、過去の設計／他者の設計を比較する際の、**相対的な指標**としてご利用ください。

ポイント３）機能概要をチェックする

個人情報の取り扱いを意図した環境か、どのくらいの利用者を想定するかといった要件のポイントが要約されます。同要約には設計の**特徴を素早く把握**するためのキーワードが含まれます。



3-2. レポート活用（セキュリティを監査する）

ForkMe!は要件と設計データをワンストップで確認できるため、設計者によるセキュリティのセルフチェックや情報システム部門による監査の、品質とスピードが向上します。

ポイント）トリアージと、デフォルトセーフからの逸脱確認

監査不足による事業リスクを低減しつつ、その作業負荷を持続可能なレベルに抑えるには、医療現場のようなトリアージ（選別）が有効です。 はじめにセキュリティリスクをすばやく評価し、どの程度の監査が必要かを判断しましょう。次にデフォルトセーフ（可能な限りシステムの気密性を維持し、不要な通信経路を開かない）が守られていることを確認し、基本的なリスク確認を行いましょう。

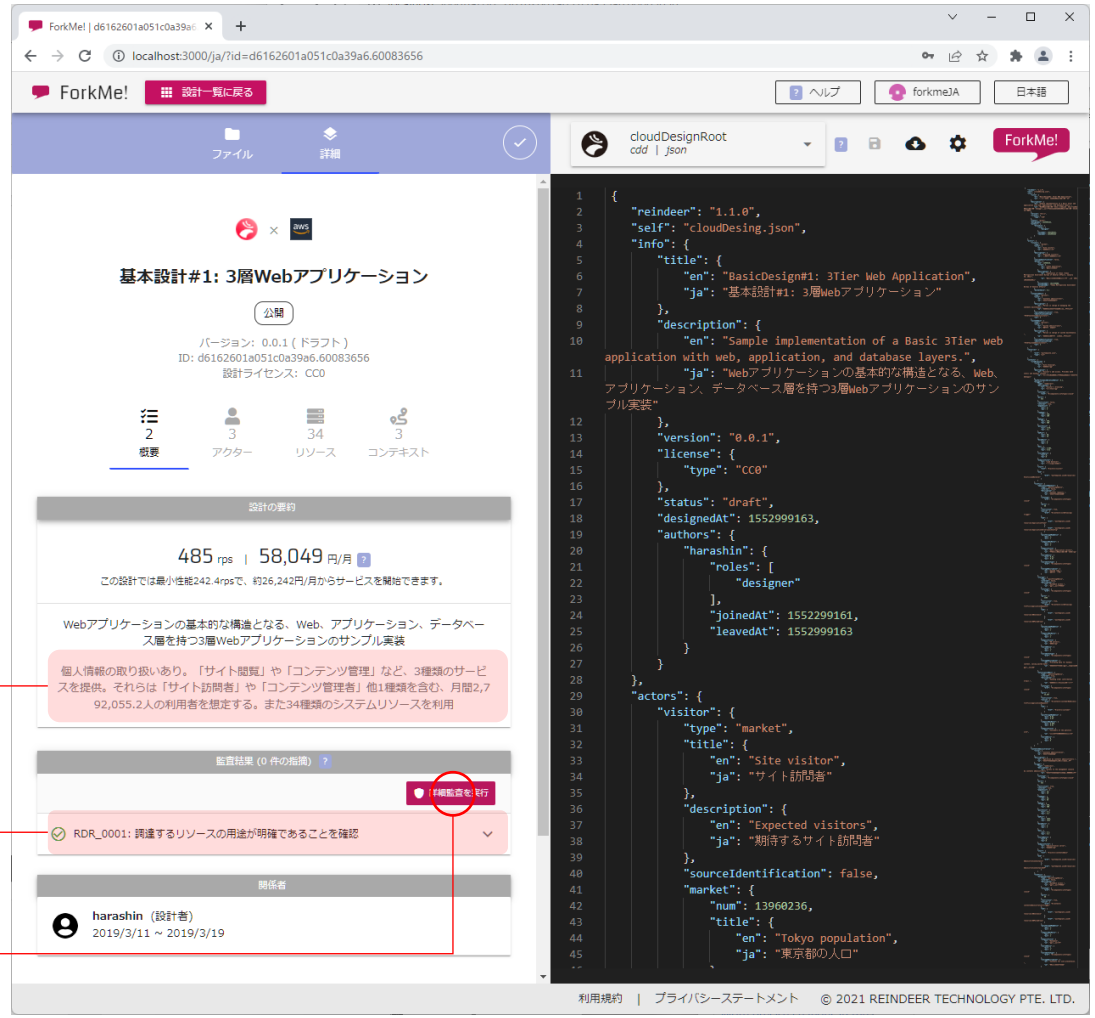
ステップ１）セキュリティリスクの評価

概要タブで以下の情報を確認し、リスクを評価します。

１－１）個人情報の取り扱い有無とそのボリュームを確認
個人情報の取り扱いボリュームが大きいほど、問題発生時の事業影響は高まります。よってまずは、要約文に記載される個人情報の取り扱い有無と取扱量を確認し、取扱量が多い場合は、詳細な監査に進みましょう。

１－２）使途不明なリソースの有無を確認
監査結果に表示される「RDR_0001」に緑のチェックマークがついていない場合には、登録された要件データに不備があります。不備があるとセキュリティリスクを正しく評価できないため、設計者に修正を依頼してください。

１－３）静的解析を実行
一般的に推奨される設定からの逸脱が表示されます。逸脱範囲が許容できない場合には、設計者に逸脱の理由を確認します。



3-2. レポート活用（セキュリティを監査する）

ステップ２）不要な通信経路の発見

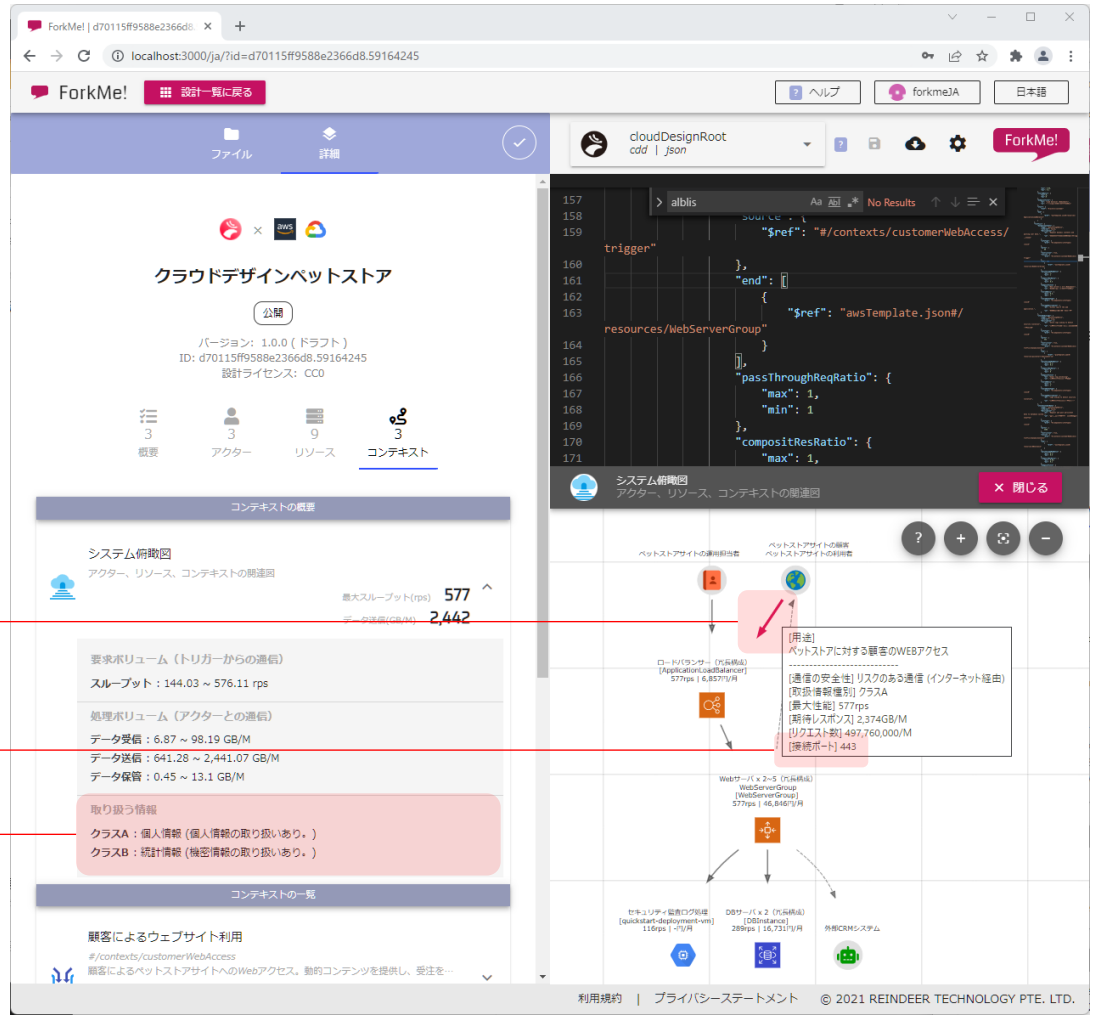
コンテキストタブで以下の情報を確認し、不要な通信経路を発見します。

２－１）リスクのある通信に注目
「システム俯瞰図」をクリックして図面を開き、図中に赤の矢印があるかを確認します。同矢印は、開かれた通信経路で個人情報あるいは機密情報が扱われる場合（不特定多数にログイン機能を提供するなど）に表示されます。その必要性が明確でない場合には、同経路を許容すべきでありません。
なお、各結線は以下を意味します。

→　リスクのある通信 (重要な情報が通信制限のない経路で送受信される)
→　安全性を問わない通信 (通信制限のない経路で、公開または機密性の必要がない情報が送受信される)
→　安全な通信 (通信制限のある経路で情報が送受信される)
破線　システム内からアクターへのアウトバウンド通信

２－２）暗号化されないポートに注目
赤矢印にカーソルをあわせ、利用ポートを確認します。22, 443, 465, 995以外の場合には通信が暗号化されていない可能性があります。同経路の暗号化を設計者に依頼しましょう。

２－３）取り扱う情報種別を確認
設計者が要件データとして登録している情報種別が表示されます。本区分が監査部門が用いる情報種別（機微な個人情報をクラスAとするなど）と異なる場合にはセキュリティリスクを正しく評価できないため、設計者に修正を依頼してください。



3-3. レポート活用（コストを評価する）

ForkMe!は要件と設計データをワンストップで確認できるため、設計者によるコストのセルフチェックや企画者によるコスト評価の、精度とスピードが向上します。

ポイント）費用対効果の確認

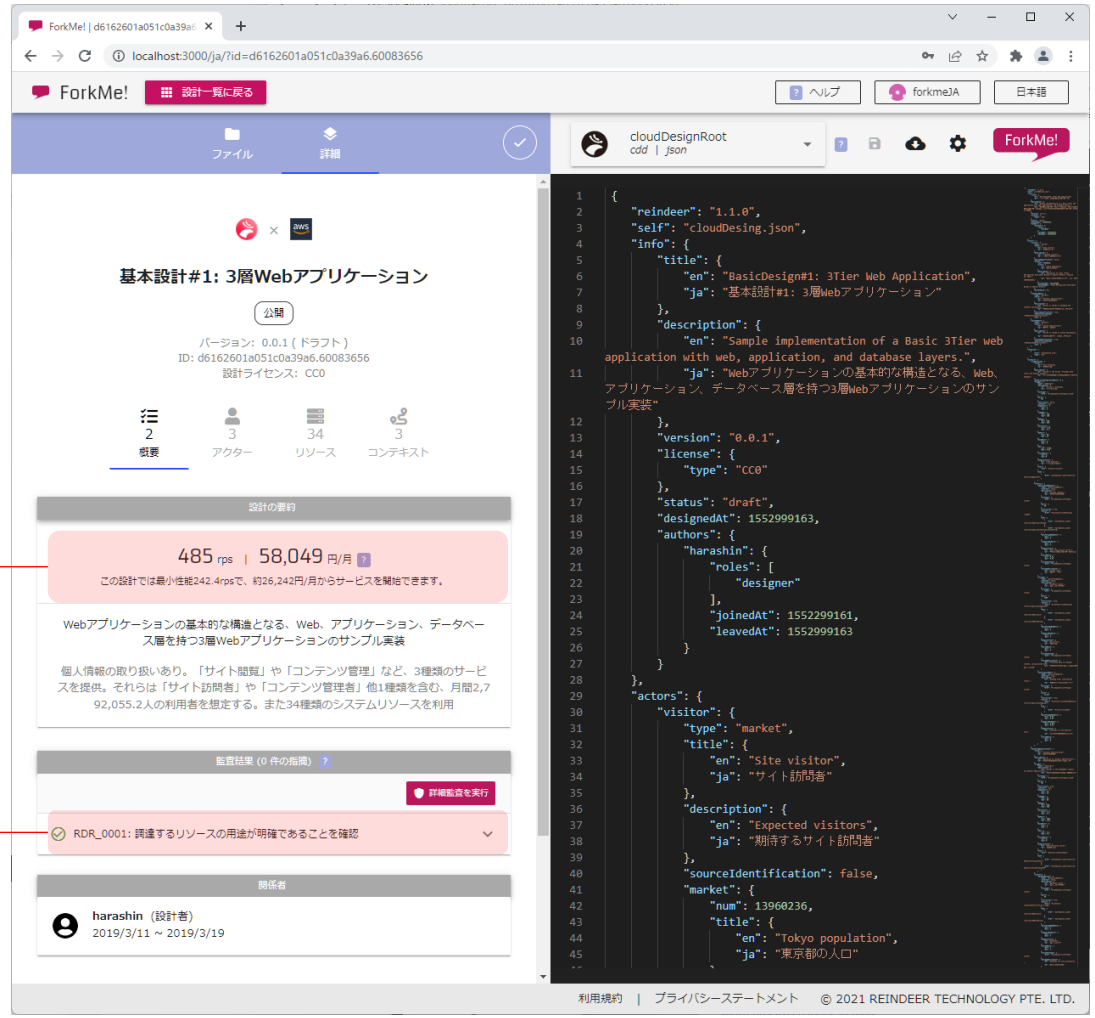
コストの妥当性は単純な価格の高低ではなく、用途の重要性に見合っているかがポイントです。よって重要性の低い用途に、相対的に大きなコストがかかっていないかを確認しましょう。
注意：本システムのコスト算定機能は、現時点ですべてのクラウドリソースを網羅していません。対応するリソースの種類は、コスト右に表示される？アイコンをクリックしてご確認ください。

ステップ１）基本的なコスト感を確認

概要タブで以下の情報を確認し、コスト感を確認します。

１－１）詳細なコスト評価の必要性を確認
概要タブで稼働費と通信費を含む概算コストの総計を確認します。本値が充分に低いと考えられる場合、本システムをつかった詳細評価の必要はありません。そうでない場合には、詳細評価に進みましょう。

１－２）使途不明なリソースの有無を確認
監査結果に表示される「RDR_0001」に緑のチェックマークがついていない場合には、登録された要件データに不備があります。不備があるとコストを正しく評価できないため、設計者に修正を依頼してください。



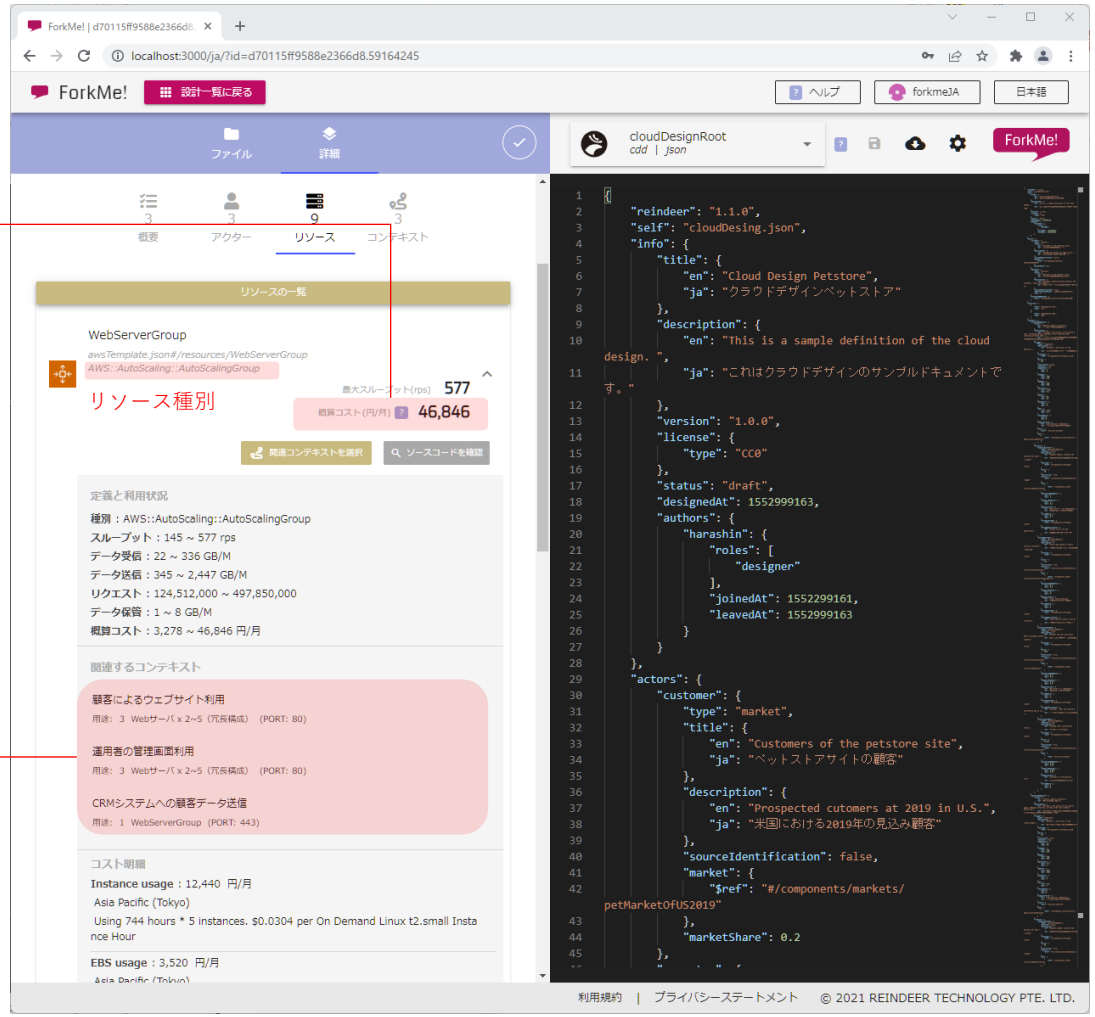
3-3. レポート活用（コストを評価する）

ステップ２）無駄な占有リソースの発見

リソースタブで以下の情報を確認し、無駄な占有リソースを発見します。

２－１）相対的にコストの高いリソースから詳細を確認
リソースの一覧はコストの高いリソースが昇順に並ぶため、上から順にリソースをクリックしてその詳細を確認すると効率的です。
ただし本システムがコスト計算に対応していないリソースには、コストが表示されません。同コストの確認が必要な場合には、表示されるリソース種別をもとに、公式ドキュメントを確認してください。

２－２）リソースの用途を確認
リソースを利用する用途（コンテキスト）が１つしか記載されていない場合、当該リソースは特定の用途に占有されていることを意味します。
当該用途の利用頻度が極めて低い（最大スループットが1未満）、あるいは少数の関係者に向けたものである場合には、コストの妥当性を検討すべきです。
例えば月に数回しか利用しない用途で専用サーバが常設されている場合、またミッションクリティカルではない運用者向け機能にもかかわらず、用途ごとに専用サーバが用意されているなどの場合には、占有リソースを用いる理由を設計者に確認しましょう。
リソースの兼用やサーバレス化によって、同コストを圧縮できる可能性があります。



3-4. レポート活用（障害リスクを評価する）

ForkMe!は要件と設計データをワンストップで確認できるため、設計者による可用性のセルフチェックや企画者による障害リスク評価の、精度とスピードが向上します。

ポイント）クォータと単一障害点の確認

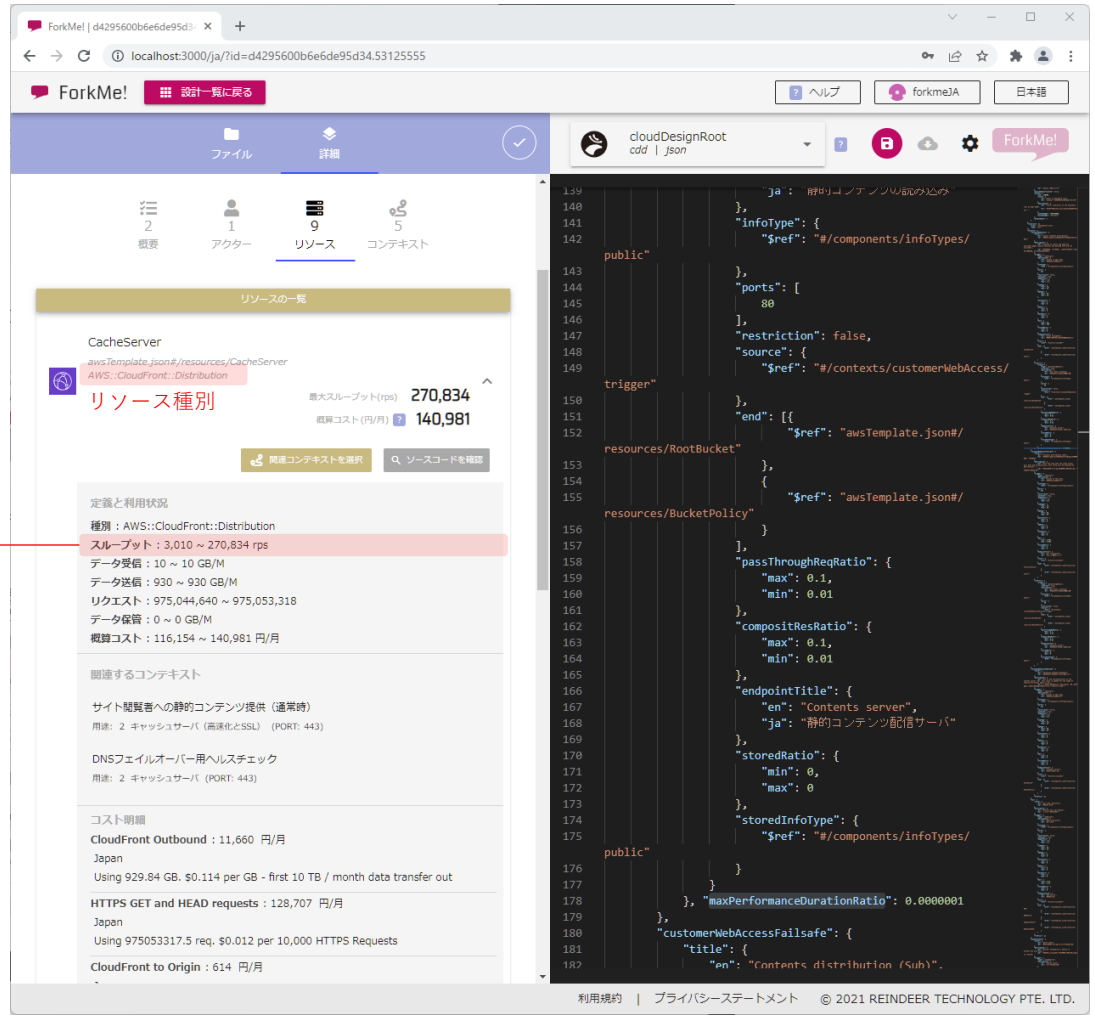
各リソースや外部システムが許容範囲（クォータまたはレートリミット）を超えたアクセスを受けた場合、また他で代替不能なリソース（単一障害点）がダウンした場合には、システムの一部または全体が停止します。それらのリスクと影響範囲を事前に把握し、必要に応じた対策を検討しましょう。

ステップ１）リソースの上限緩和申請を行う

リソースタブで以下の情報を確認し、上限緩和申請を行います。

１－１）上限緩和申請の必要性を確認
各リソースの詳細を開き、ベンダーが定める上限（クウォータ）を超えた利用ボリュームが想定されていないかを確認します。各リソースの上限については、ベンダー公式ドキュメントを参照してください。
右の例では、キャッシュサーバの想定スループットがAWS:CloudFrontの利用上限（クォータ）である250,000rpsを超えるため、AWSに対する緩和申請が必要となります。

１－２）上限緩和の申請
クラウドベンダーとの契約を管理する管理者に、上限緩和申請を依頼してください。なお、申請は必ず承認されるとは限りません。承認されなかった場合、複数のリソースに通信を分散させるなどの、設計の見直しが必要です。



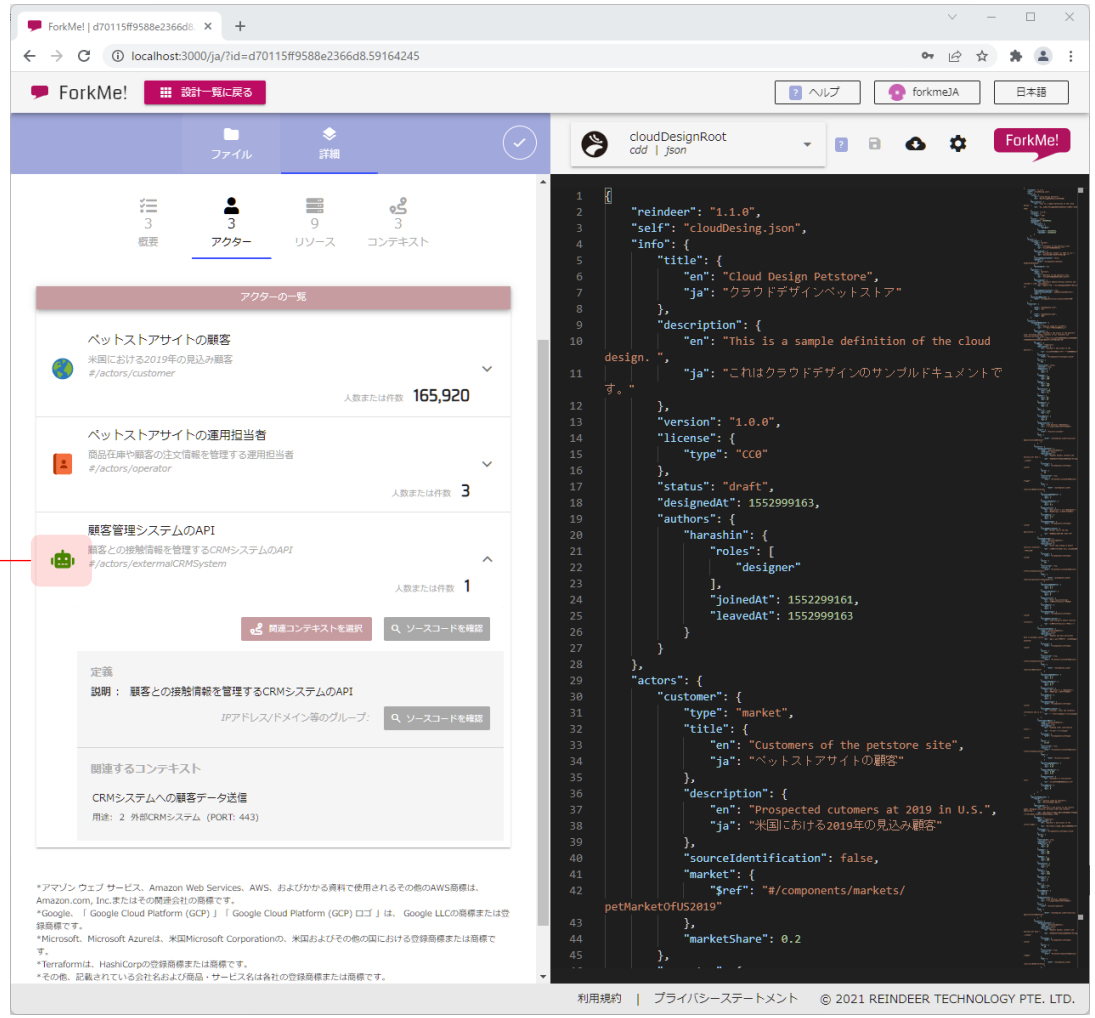
3-4. レポート活用（障害リスクを評価する）

ステップ２）外部システムの上限緩和申請（クォータ対策）を行う

アクタータブで以下の情報を確認し、上限緩和申請を行います。

１－１）上限緩和申請の必要性を確認
APIやPOSTリクエスト先など、関連する外部システムはロボットのアイコンで表示されます。当該アクターの想定スループットが大きい場合には、外部システムが設定する利用上限を超えている可能性があります。その場合、データの一部が欠損する、速度が低下するなどの影響を受けるため、同システムを提供するベンダーへの上限緩和申請が必要になる可能性があります。
各リソースの上限については、各外部システムにお問い合わせください。

１－２）上限緩和の申請
上限緩和申請は一般的に、外部システムとの契約を管理するアカウント管理者が手続きを行う必要があります。なお、申請は必ず承認されるとは限りません。
承認されなかった場合には、キャッシュを利用して通信量を減らすなどの、設計の見直しが必要です。



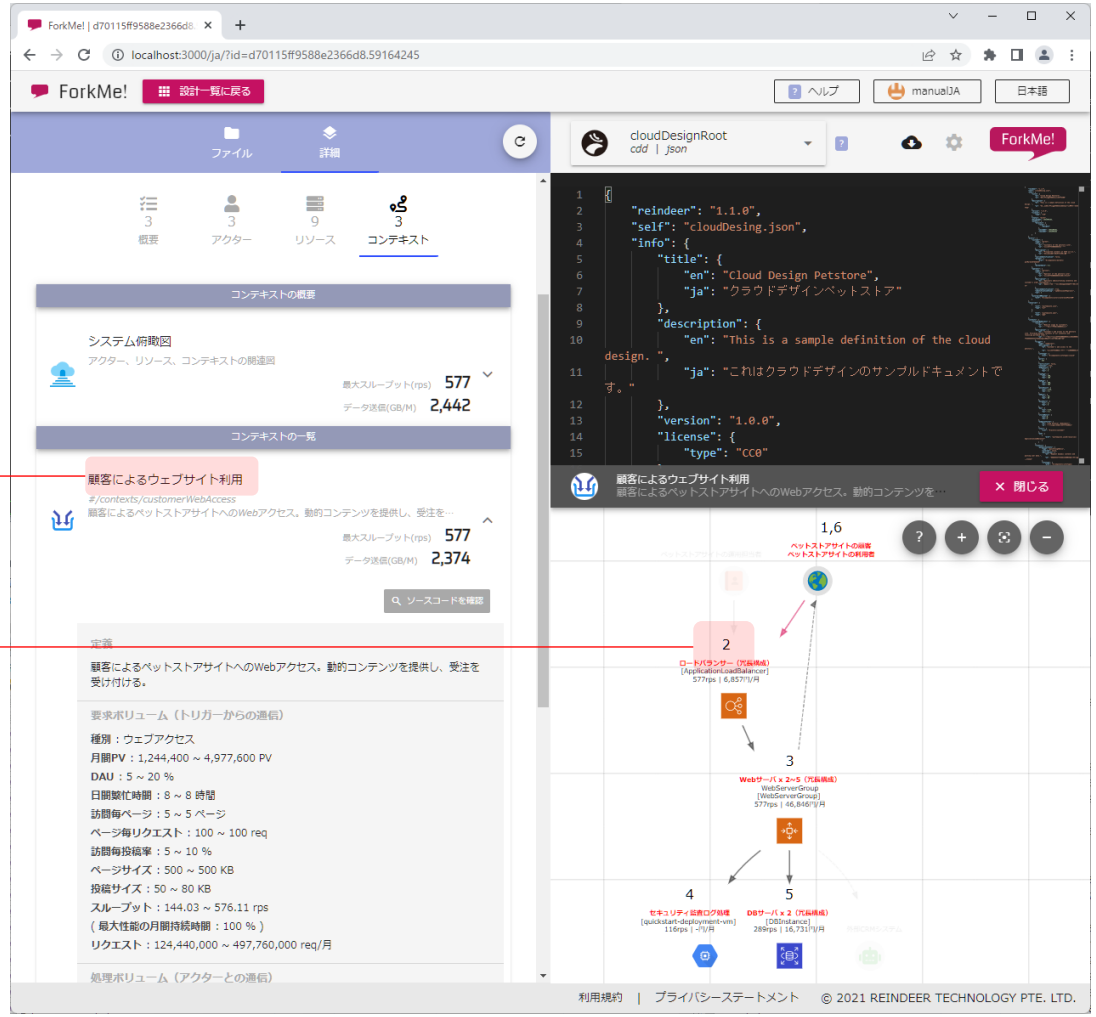
3-4. レポート活用（障害リスクを評価する）

ステップ３）リスクの高い単一障害点の発見

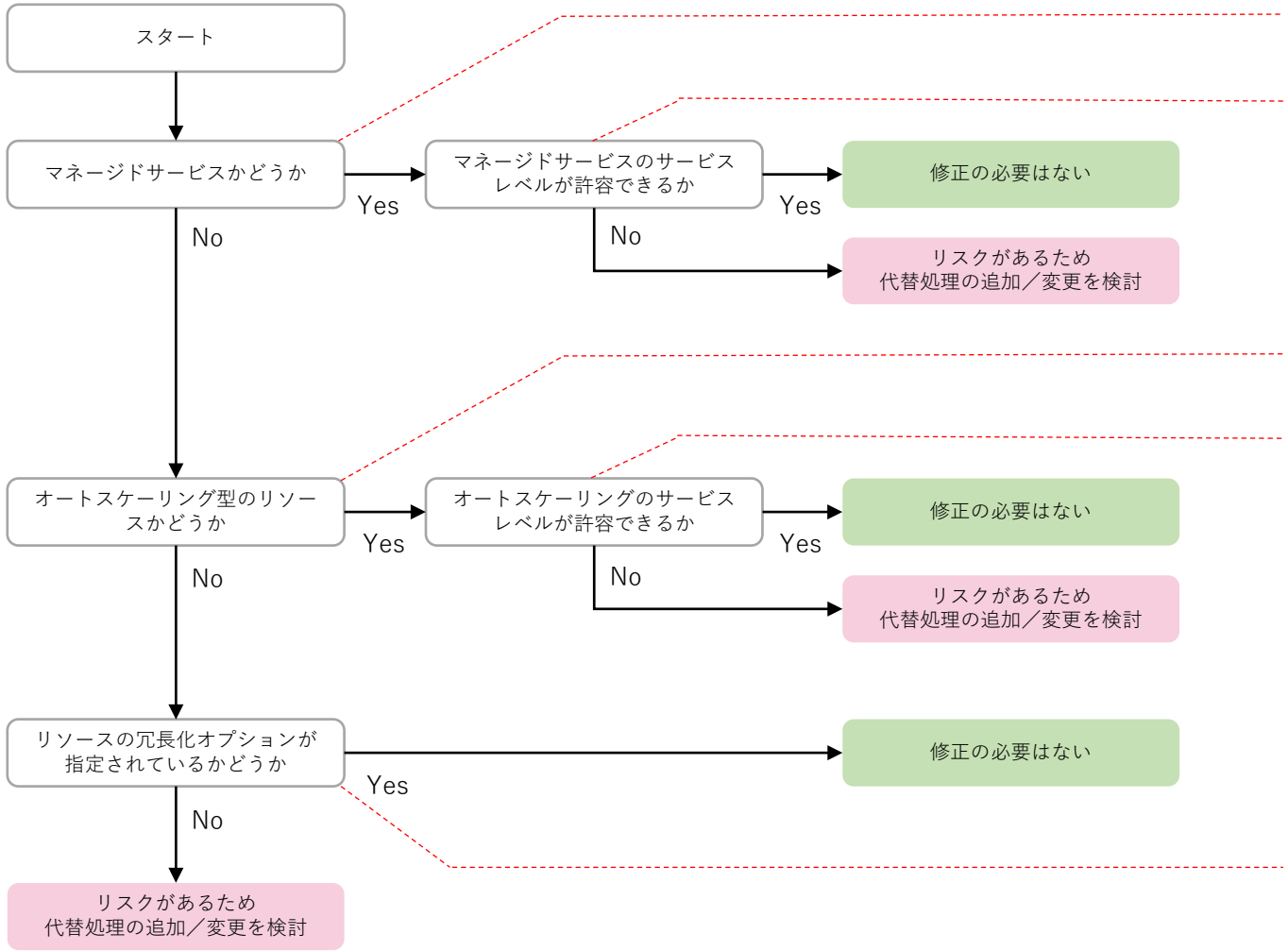
コンテキストタブで以下の情報を確認し、リスクの高い単一障害点を発見します。

１－１）障害からの復旧スピードが求められるコンテキストを探す
コンテキスト名をヒントに、障害からの復旧スピードが求められるコンテキストを探し、選択します。名称からの判定が難しい場合、レポートの「アクター」タブから重要なアクター（カスタマーやクライアントなど）を選択してください。同アクターの「関連コンテキストを選択」ボタンを押して選択されるコンテキストが、対象になると考えられます。

１－２）単一障害点となるリソースを探す
画面右の図には、各アクター／リソースに処理の順番を示す番号が割り当てられています。番号の昇順にリソースを見ていきましょう。同種の複数リソースが同じ番号で利用される場合には冗長化されており、単一障害点ではありません。そうでない場合は各リソースをクリックしてリソースの詳細を確認し、次頁のチャートにもとづいてリスクを評価してください。



3-4. レポート活用（障害リスクを評価する）



例えばリソース種別がAWS::ApiGateway::RestApiであれば、サーバ台数を意識せずAWSに維持管理任せられるマネージドサービス型のリソースです。一般的に同種のリソースは冗長化への配慮があり、単一障害点でないと判断できます。

マネージドサービスであっても、過去の傾向から障害リスクがあると考えられる場合には、追加のリスク低減策を講じる必要があります。（例：CloudFront障害時にDNSフェールオーバーでAPI Gatewayを利用するなど）

例えばリソース種別がAWS::AutoScaling::AutoScalingGroupであれば、障害リスクが高まった時に自動的にサーバ台数が調整されるオートスケーリング型のリソースです。同種のリソースは冗長化の調整が可能なため、概ね単一障害点ではないと判断できます。

オートスケーリング型のリソースであっても、スケーリング時の遅延時間が要件にそぐわない場合には、追加のリスク低減策を講じる必要があります。（例：AutoScalingに頼らずインスタンスを常設するなど）

リソースによってはオプションの設定で冗長化が行われている場合があります（例：RDSのMultiAZ）。オプションが有効かは「ソース確認」をクリックしてプロビジョニングコードを確認するか、設計者にお問い合わせください。