

CTC Engineers Voyage #1

CTC金融エンジニアのAWS LTチャレンジ

カオスエンジニアリングやってますか？

伊藤忠テクノソリューションズ株式会社

金融技術第4部

大友 勝明

無限の未来と、幾千のテクノロジーをつなぐ。

CTC Financial Services Group

P03

はじめに

はなすひとの紹介、はなすこと、はなさないこと

P07

カオスエンジニアリングのはなし

カオスエンジニアリングについての簡単な説明

P12

AWS Fault Injection Serviceのはなし

FISをさわってみて感じたこと、考えたこと

P25

さいごに

今日おぼえて帰って欲しいこと

無限の未来と、幾千のテクノロジーをつなぐ。

CTC Financial Services Group

はじめに

はなすひとの紹介、はなすこと、はなさないこと

無限の未来と、幾千のテクノロジーをつなぐ。

CTC Financial Services Group

はなすひとの紹介

- 名前：大友 勝明
- 所属：伊藤忠テクノソリューションズ株式会社 金融技術第4部
- 生年月日：1971年10月23日
- 好きなひと：妻と子供（高2男子）
- 好きなSF作家：テッド・チャン
- 好きな犬種：シーズー
- 好きな空手の型：スーパーリンペイ



無限の未来と、幾千のテクノロジーをつなぐ。

CTC Financial Services Group

はなすこと

- カオスエンジニアリングの簡単な説明
- FISをさわってみて感じたこと
- カオスエンジニアリングについてのちょっとした考察

カオスエンジニアリングをやってみたい！と思います

無限の未来と、幾千のテクノロジーをつなぐ。

CTC Financial Services Group

はなさないこと

- FISを利用するための具体的なノウハウ
- FISのベストプラクティス

FISがすぐに活用できるようにはなりません

無限の未来と、幾千のテクノロジーをつなぐ。

CTC Financial Services Group

カオスエンジニアリングのはなし

カオスエンジニアリングについての簡単な説明

無限の未来と、幾千のテクノロジーをつなぐ。

CTC Financial Services Group

本番環境でわざと障害起こすの！？本気！？

- サービス影響出たらどーすんの！？
- 障害テストとなにが違うの！？



無限の未来と、幾千のテクノロジーをつなぐ。

CTC Financial Services Group

カオスエンジニアリングとは？

Chaos Engineering is the discipline of experimenting on a system in order to build confidence in the system's capability to withstand turbulent conditions in production.

カオスエンジニアリングは、システムが本番環境における不安定な状態に耐える能力へ自信を持つためにシステム上で実験を行う訓練方法です。

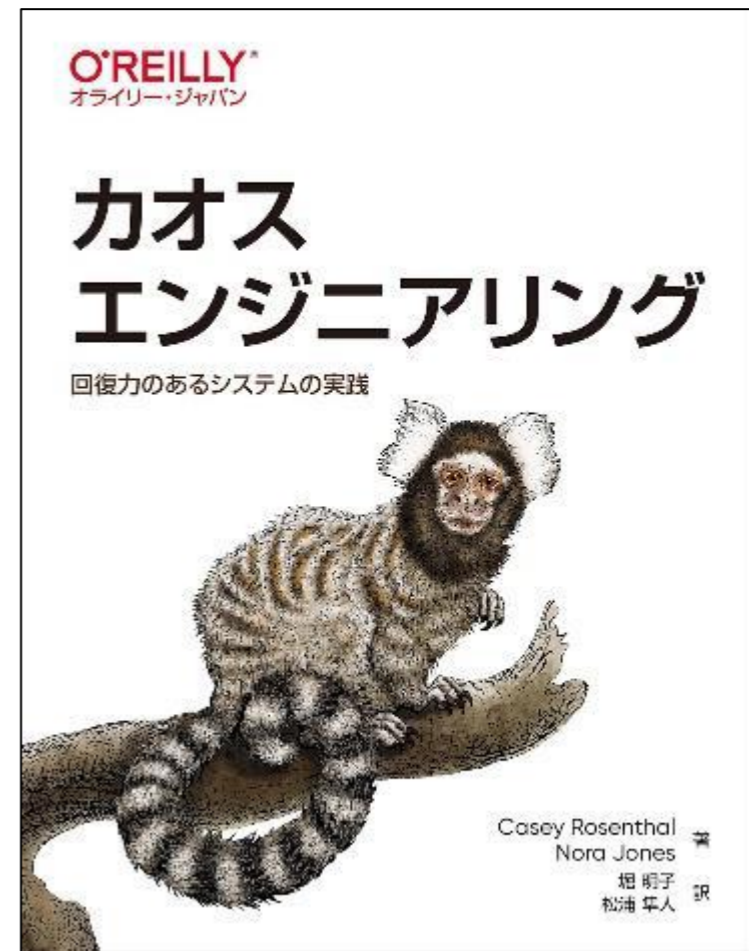
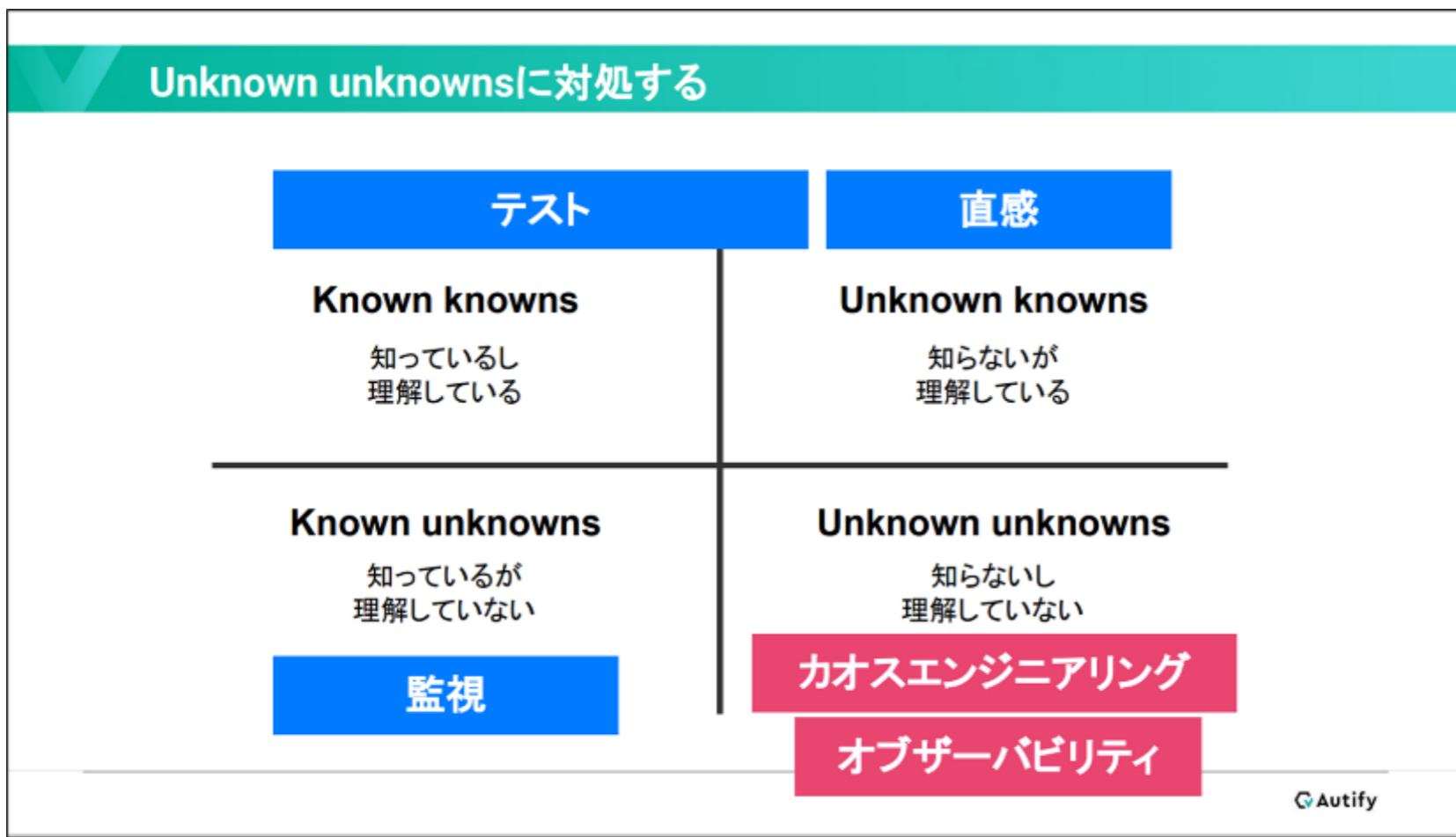
『[カオスエンジニアリングの原則](#)』（[日本語](#)）より

無限の未来と、幾千のテクノロジーをつなぐ。

CTC Financial Services Group

なぜカオスエンジニアリング？

システムが複雑化したことにより、**全体としてどのような振る舞いをするか**把握できない。



Cf. [カオスエンジニアリングの裏話](#)

カオスエンジニアリングの原則

- 定常状態における振る舞いの仮説を立てる
- 実世界の事象は多様である
- 本番環境で検証を実行する
- 継続的に実行する検証の自動化
- 影響範囲を局所化する

『[カオスエンジニアリングの原則](#)』（[日本語](#)）より

無限の未来と、幾千のテクノロジーをつなぐ。

CTC Financial Services Group

AWS Fault Injection Serviceのはなし

FISをさわってみて感じたこと、考えたこと

無限の未来と、幾千のテクノロジーをつなぐ。

CTC Financial Services Group

Workshopをやってみた

Chaos Engineering Workshop V2

Chaos Engineering Workshop V2

Welcome to our **chaos engineering workshop focusing on AWS Fault Injection Service**. In today's rapidly evolving digital landscape, system reliability isn't just a technical requirement—it's a business imperative. chaos engineering emerges as a proactive approach to building resilient systems by deliberately introducing controlled failures to understand and improve system behavior under stress.

Throughout this workshop, we'll explore how **chaos engineering can transform** your organization's approach to system reliability. You'll learn how this practice helps teams gain deeper insights into their systems' behavior during failures, reduce unexpected incidents, and validate system hypotheses in a controlled environment.

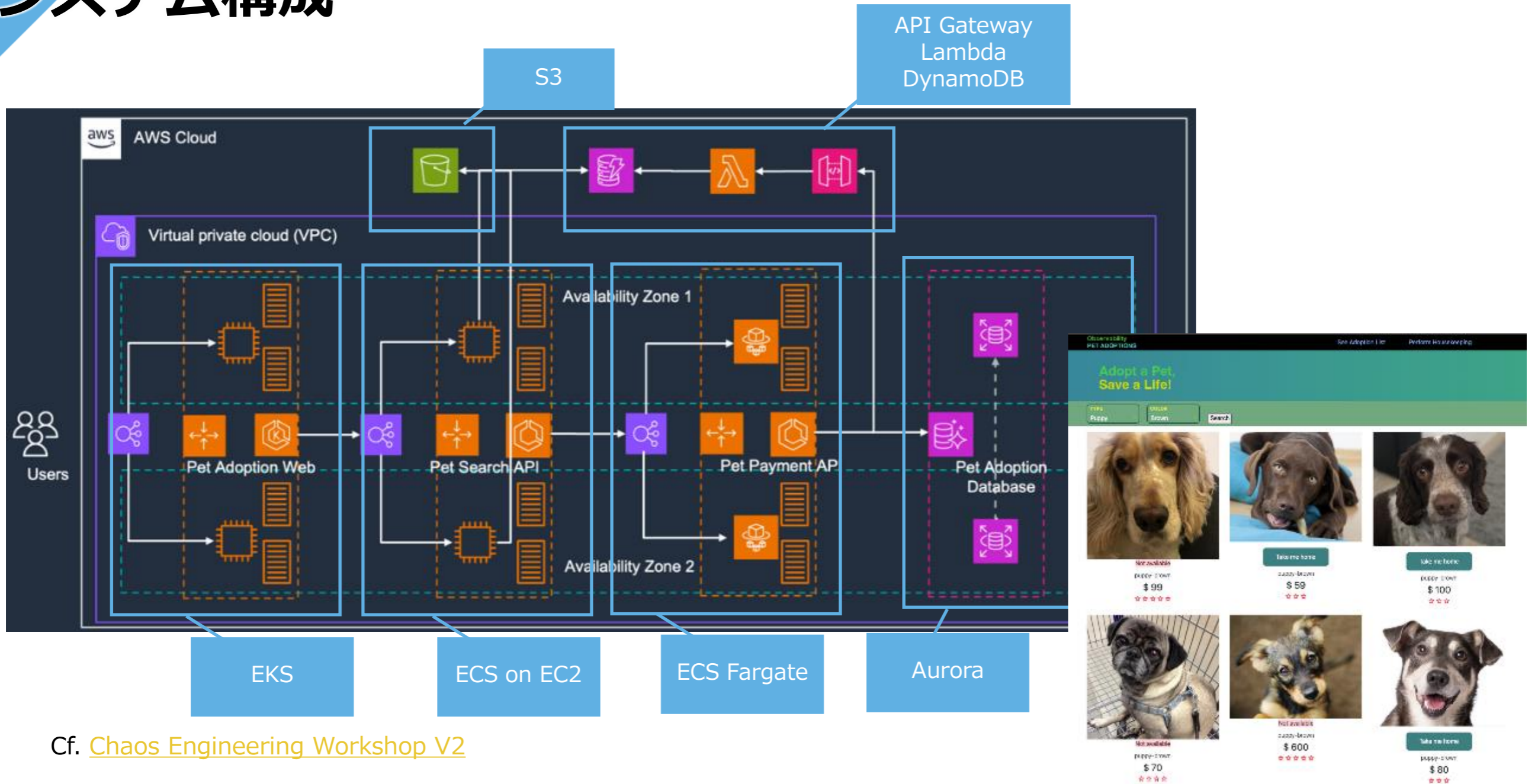
Using **AWS Fault Injection Service**, we'll walk through practical examples of implementing chaos engineering principles in your cloud infrastructure. Whether you're new to **chaos engineering** or looking to enhance your existing practices, this workshop will equip you with the knowledge and hands-on experience. You'll learn to build your own chaos experiments that help your applications withstand disruptions with minimal to no customer impact.

Let's begin our journey into the world of controlled chaos engineering and discover how it can lead to stronger, more reliable systems.



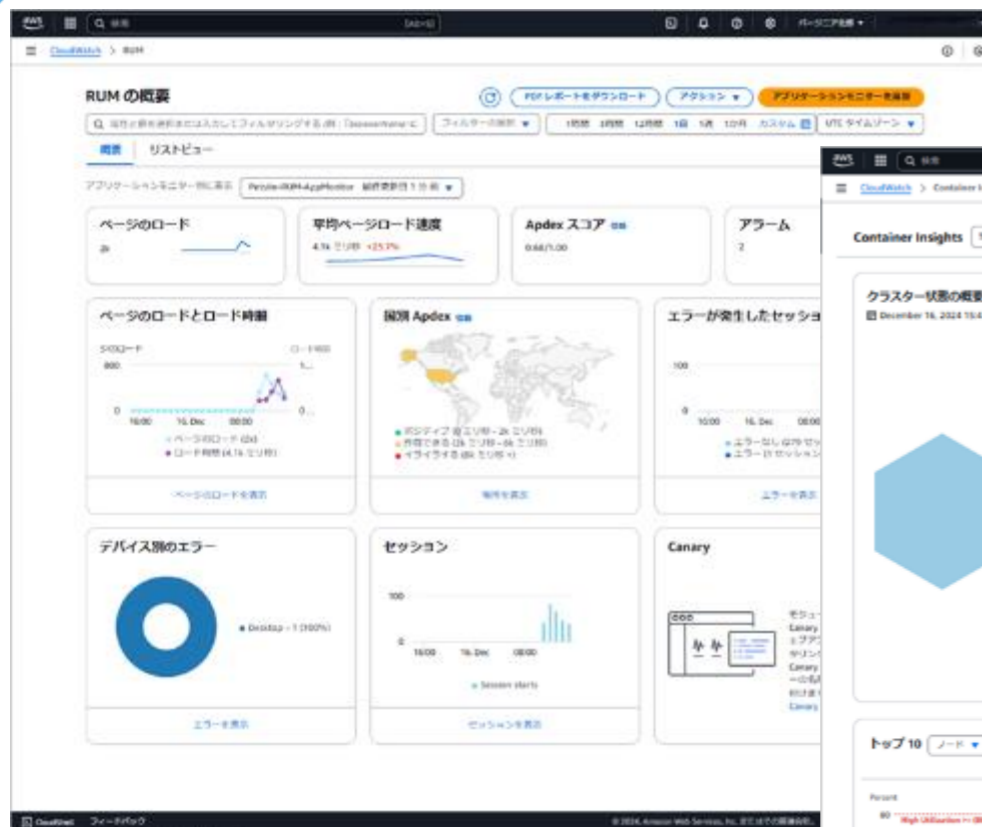
Cf. [Chaos Engineering Workshop V2](#)

システム構成

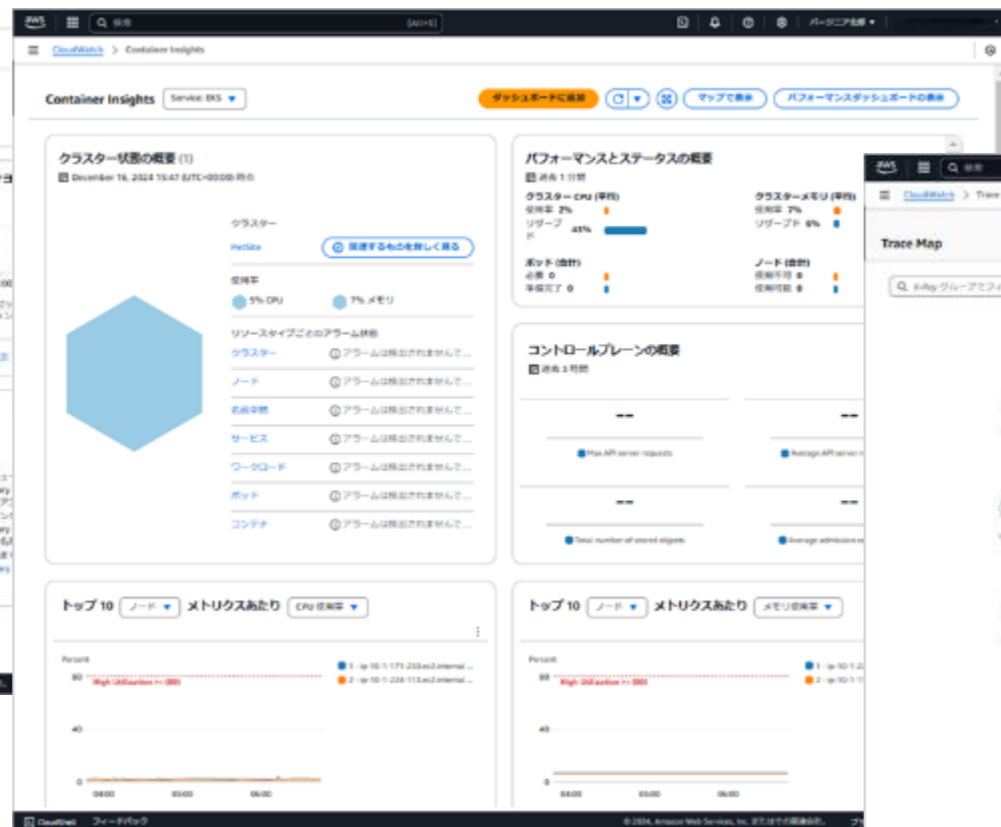


Cf. [Chaos Engineering Workshop V2](#)

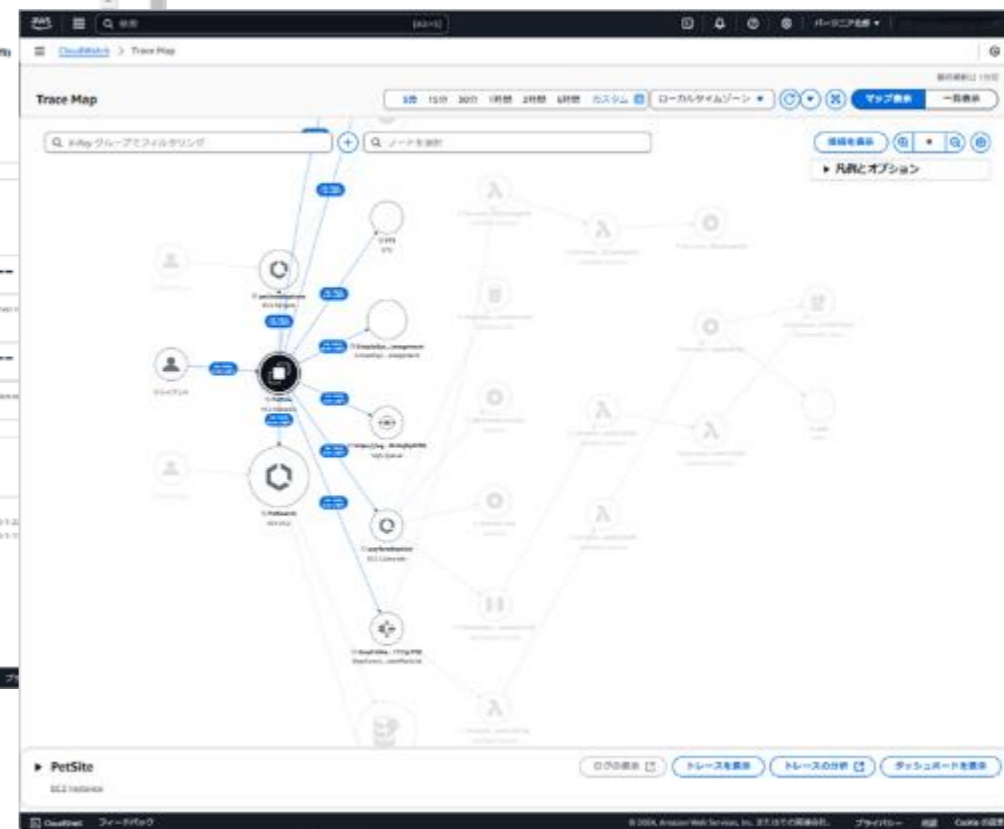
可観測性



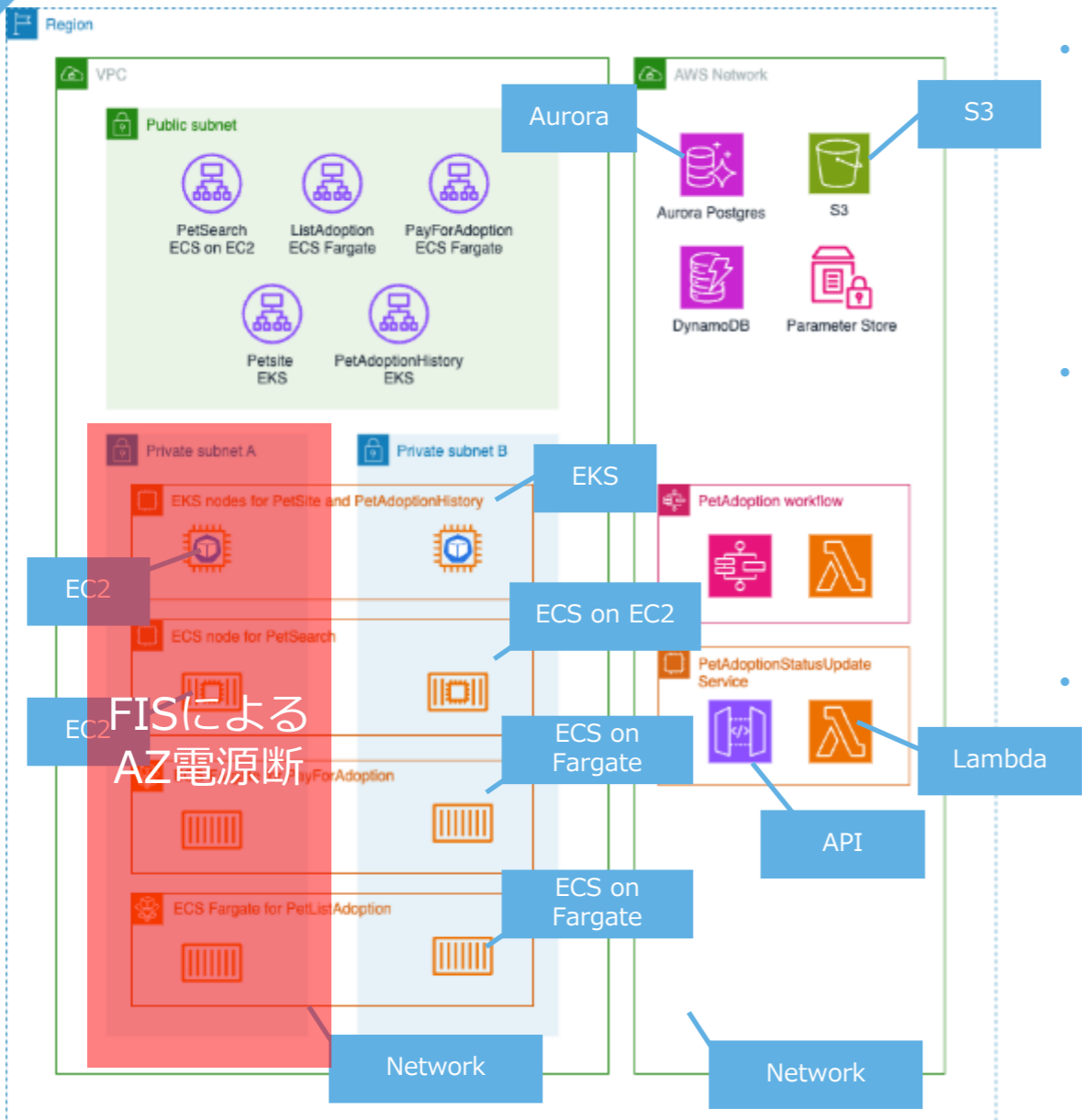
CloudWatch RUM (Real User Monitoring)



Container Insights



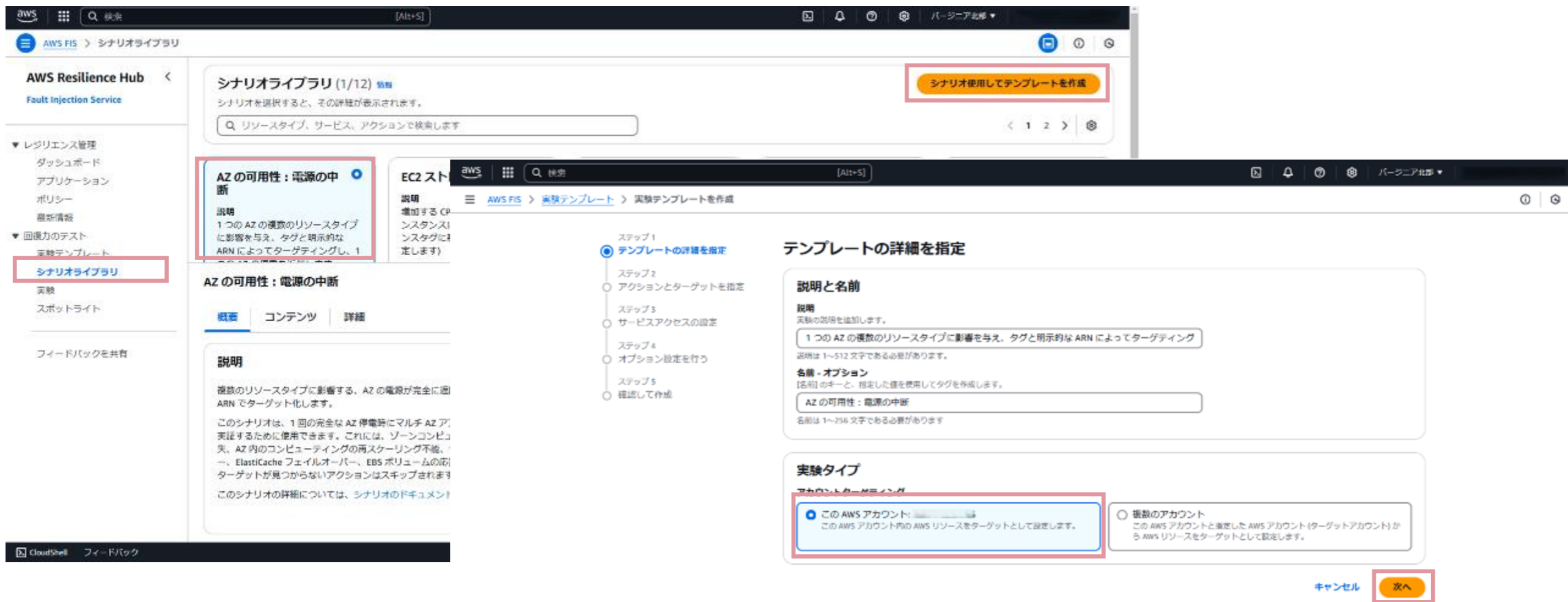
X-Ray Trace Map



- EC2
 - EC2 Disk Full
 - EC2 Latency injection
 - EBS I/O
 - EC2 stop instances
- Lambda
 - Invocation Add Delay
 - Invocation Error
 - Invocation HTTP Integration Response
- EKS
 - CPU Stress
 - Network Latency
 - Memory Stress
 - IO Stress
 - Pod Delete
- ECS on EC2
 - Task CPU Stress
 - Task I/O Stress
 - Network latency to DynamoDB
 - ECS terminate node
- ECS Fargate
 - Task CPU Stress
 - Task I/O Stress
- Databases
 - Aurora Instance reboot
 - Aurora Cluster Failover
- S3
 - S3 impairment in AZ1
 - S3 impairment in AZ2
- Network
 - AZ Disruption
 - Network Disruption
 - DynamoDB Network Disruption
- API
 - API Throttling
 - API Unavailable

Cf. [Chaos Engineering Workshop V2](#)

実験 (AZ電源断)



The screenshot displays the AWS Resilience Hub console for configuring a scenario. The main area is titled 'シナリオライブラリ (1/12) 詳細' (Scenario Library (1/12) Details). A search bar is present with the text 'リソースタイプ、サービス、アクションで検索します'. A button 'シナリオを使用してテンプレートを生成' (Generate template using scenario) is highlighted in orange. The left sidebar shows the navigation menu with 'シナリオライブラリ' (Scenario Library) highlighted in red. The main content area shows the details for the scenario 'AZの可用性: 電源の中断' (AZ Availability: Power Outage). The '説明' (Description) section states: '複数のリソースタイプに影響する、AZの電源が完全に落ち、ARNでターゲット化します。このシナリオは、1回の完全なAZ停電時にマルチAZアプライアンスのために使用できます。これには、ゾーンコンピューティング失敗、AZ内のコンピューティングの再スケール不能、ElastiCache フェイルオーバー、EBS ボリュームのターゲットが見つからないアクションはスキップされます。このシナリオの詳細については、シナリオのドキュメントを参照してください。' (Affects multiple resource types, AZ power is completely out, target by ARN. This scenario can be used for multi-AZ availability during a full AZ outage. This includes zone computing failure, re-scaling failure in AZ, ElastiCache failover, and actions whose targets are not found are skipped. For details on this scenario, see the scenario documentation.) The 'テンプレートの詳細を指定' (Specify template details) section shows a list of steps: 'ステップ1: テンプレートの詳細を指定' (Step 1: Specify template details), 'ステップ2: アクションとターゲットを指定' (Step 2: Specify actions and targets), 'ステップ3: サービスアクセスの設定' (Step 3: Configure service access), 'ステップ4: オプション設定を行う' (Step 4: Configure options), and 'ステップ5: 確認して作成' (Step 5: Review and create). The '説明と名前' (Description and name) section has a description field containing '1つのAZの複数のリソースタイプに影響を与え、タグと明示的なARNによってターゲットリング' (Affects multiple resource types in one AZ, target by tags and explicit ARN) and a name field containing 'AZの可用性: 電源の中断'. The '実験タイプ' (Experiment type) section has two radio buttons: 'このAWSアカウント' (This AWS account) which is selected and highlighted in red, and '複数のアカウント' (Multiple accounts). At the bottom right, there are 'キャンセル' (Cancel) and '次へ' (Next) buttons, with '次へ' highlighted in orange.

無限の未来と、幾千のテクノロジーをつなぐ。

CTC Financial Services Group

実験 (AZ電源断)



The screenshot shows the AWS FIS console interface for creating an experiment template. The main area is titled "アクションとターゲットを指定" (Specify Actions and Targets). It displays a list of actions and their associated targets:

- Pause-Instance-Launches (30分)**: Action: `aws:ec2:api-insufficient-instance-capacity-error`; Target: `IAM-role` (aws:iam:role). Note: "1つ、または複数のリソースIDが必要です。" (One or more resource IDs are required).
- Pause-EBS-IO (30分)**: Action: `aws:ebs:pause-volume-io`; Target: `EBS-Volumes` (aws:ec2:ebs-volume).
- Stop-Instances (30分)**: Target: `EC2-Instances`.

A modal window titled "シナリオパラメータを選択" (Select Scenario Parameters) is open on the right. It contains the following information:

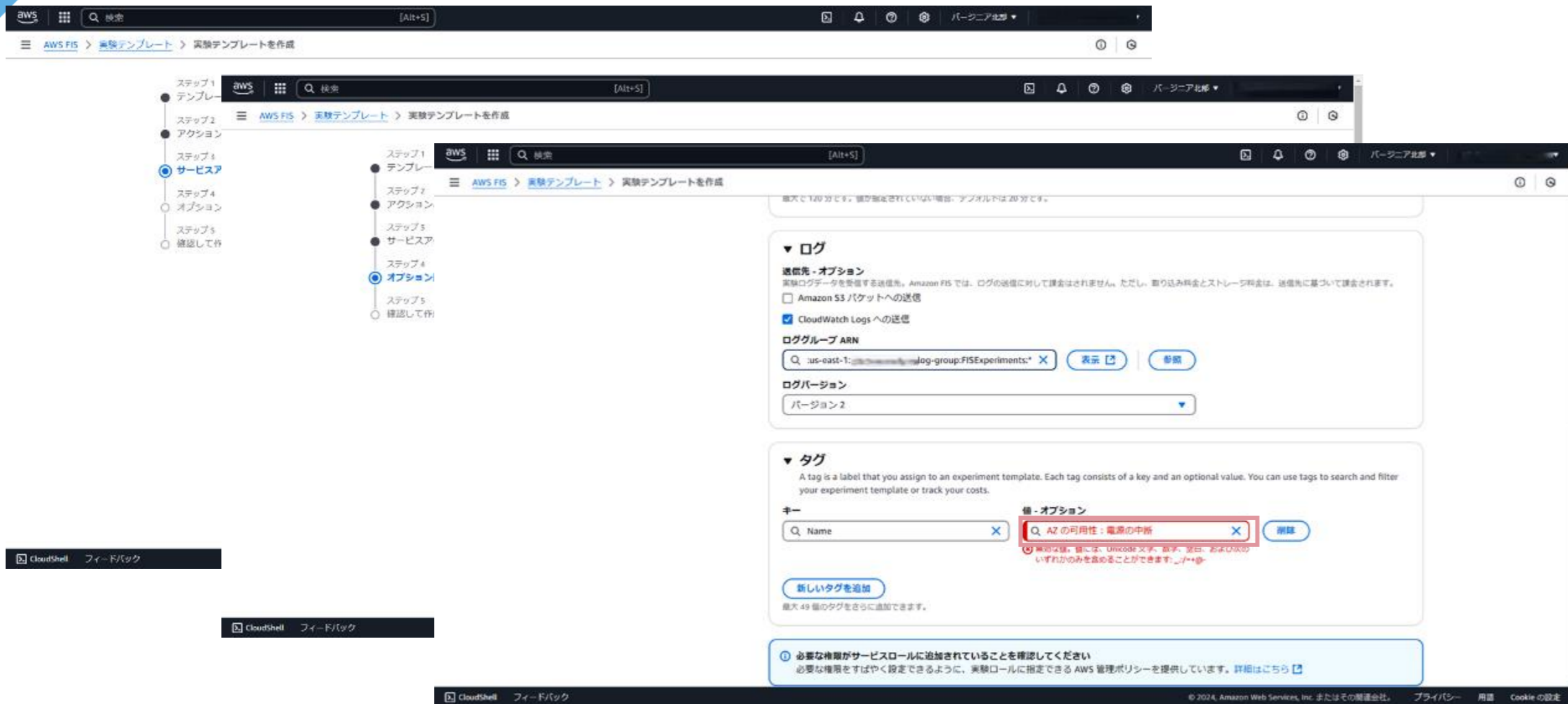
- シナリオの共有パラメータ** (Scenario Shared Parameters):
 - `affectedAz`: us-east-1a (highlighted with a red box).
 - `affectedRolesForInsufficientCapacityException`: FisServerless-FISDummyRoleForASG3A1FA08C-MMRk1xFJ16q (highlighted with a blue box).
- 詳細パラメータ - ターゲティングタグ** (Detailed Parameters - Targeting Tags):
 - `apiPauseVolumeTargetingTag`: Key: `AzImpairmentPower`, Value: `Ready`.
 - `disruptSubnetTargetingTag`: Key: `AzImpairmentPower`, Value: `Ready`.

Buttons for "共有パラメータを編集" (Edit Shared Parameters) and "共有パラメータを保存" (Save Shared Parameters) are visible.

無限の未来と、幾千のテクノロジーをつなぐ。

CTC Financial Services Group

実験 (AZ電源断)



The screenshot displays the AWS FIS console interface for creating an experiment template. The breadcrumb navigation shows 'AWS FIS > 実験テンプレート > 実験テンプレートを作成'. The left sidebar lists steps: 'ステップ1 テンプレート', 'ステップ2 アクション', 'ステップ3 サービス', 'ステップ4 オプション', and 'ステップ5 確認して作成'. The main content area is titled 'AWS FIS > 実験テンプレート > 実験テンプレートを作成' and contains the following sections:

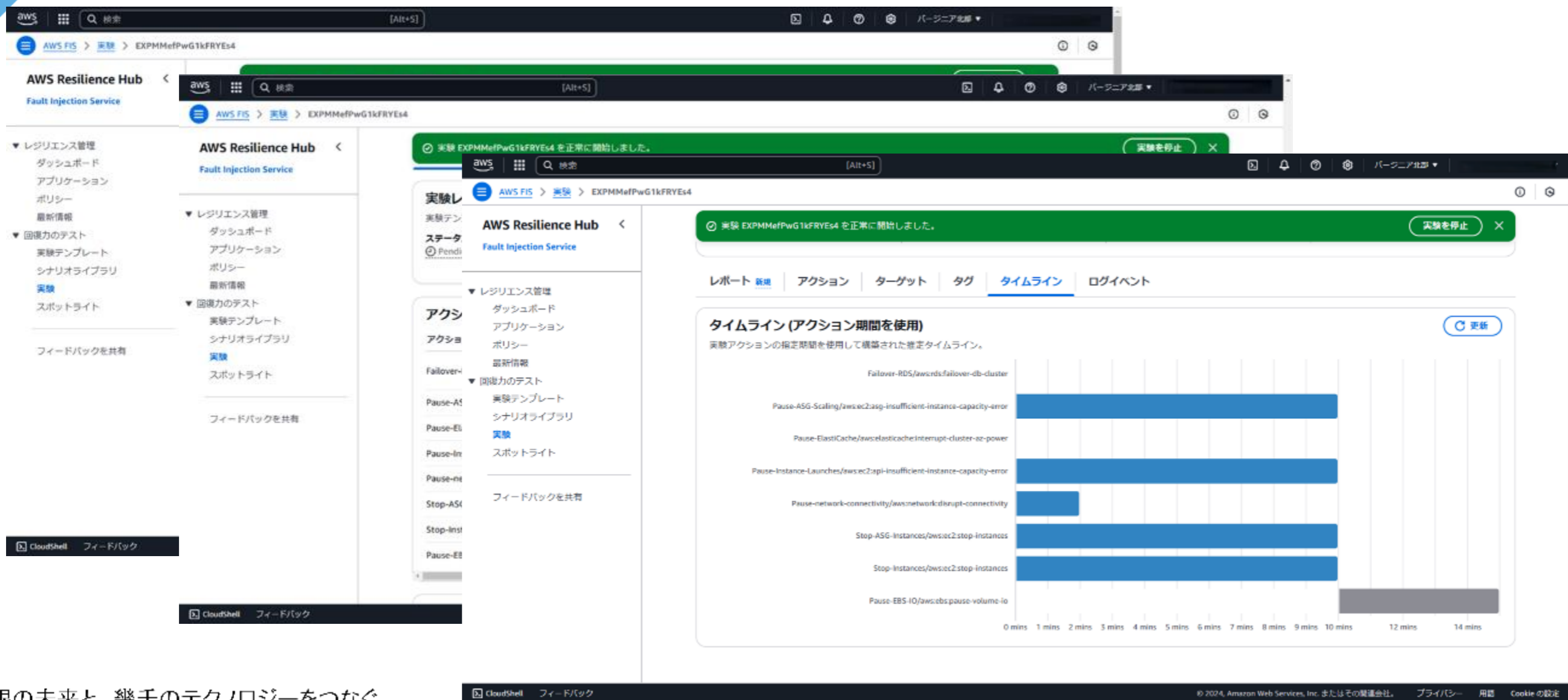
- ログ (Log):** Includes a '送信先 - オプション' (Destination - Option) section with a checkbox for 'Amazon S3 バケットへの送信' (Send to Amazon S3 bucket) and a checked checkbox for 'CloudWatch Logs への送信' (Send to CloudWatch Logs). Below this is a 'ロググループ ARN' (Log group ARN) field with the value 'us-east-1:aws-logs-9-1-log-group:FISExperiments*' and a 'バージョン' (Version) dropdown set to 'バージョン 2'.
- タグ (Tag):** Includes a 'タグ - オプション' (Tag - Option) section with a 'キー' (Key) field containing 'Name' and a '値 - オプション' (Value - Option) field containing 'AZの可用性: 電源の中断' (AZ availability: power outage). A '新しいタグを追加' (Add new tag) button is visible below.

At the bottom of the console, there is a notice: '必要な権限がサービスロールに追加されていることを確認してください。必要な権限をすばやく設定できるように、実験ロールに指定できる AWS 管理ポリシーを提供しています。詳細はこちら' (Verify that the necessary permissions are added to the service role. To help you quickly set the necessary permissions, we provide AWS managed policies that you can specify for the experiment role. See here for details).

無限の未来と、幾千のテクノロジーをつなぐ。

CTC Financial Services Group

実験 (AZ電源断)



実験 EXPMMefPwG1kFRYEs4 を正常に開始しました。

AWS Resilience Hub
Fault Injection Service

レポート 新規 | アクション | ターゲット | タグ | **タイムライン** | ログイベント

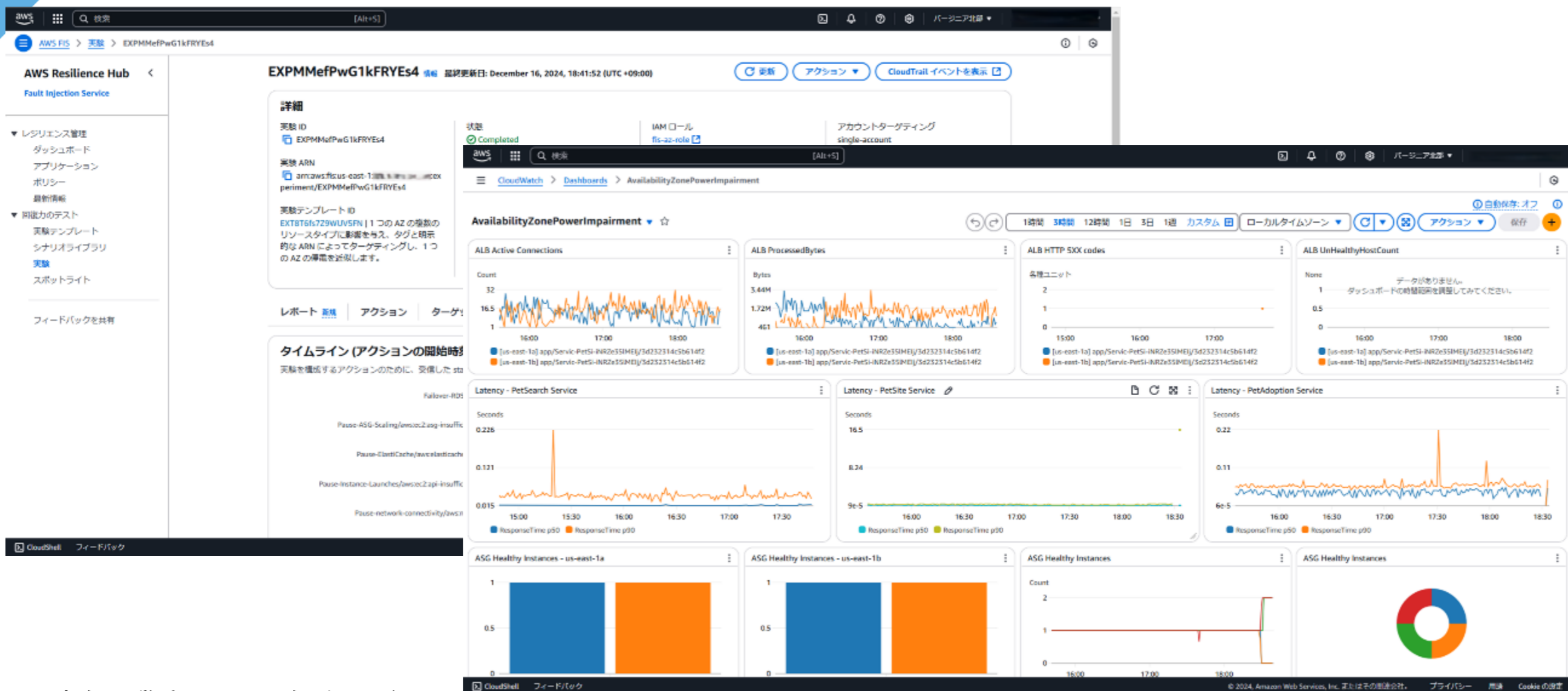
タイムライン (アクション期間を使用)
実験アクションの指定期間を使用して構築された予定タイムライン。

アクション	期間 (mins)
Failover-RDS/aws:rds:failover-db-cluster	0
Pause-ASG-Scaling/aws:ec2:asg-insufficient-instance-capacity-error	10
Pause-ElastiCache/aws:elasticache:interrupt-cluster-az-power	10
Pause-Instance-Launches/aws:ec2:api-insufficient-instance-capacity-error	10
Pause-network-connectivity/aws:network:dnstcpt-connectivity	2
Stop-ASG-Instances/aws:ec2:stop-instances	10
Stop-Instances/aws:ec2:stop-instances	10
Pause-EBS IO/aws:ebs:pause-volume-io	14

無限の未来と、幾千のテクノロジーをつなぐ。

CTC Financial Services Group

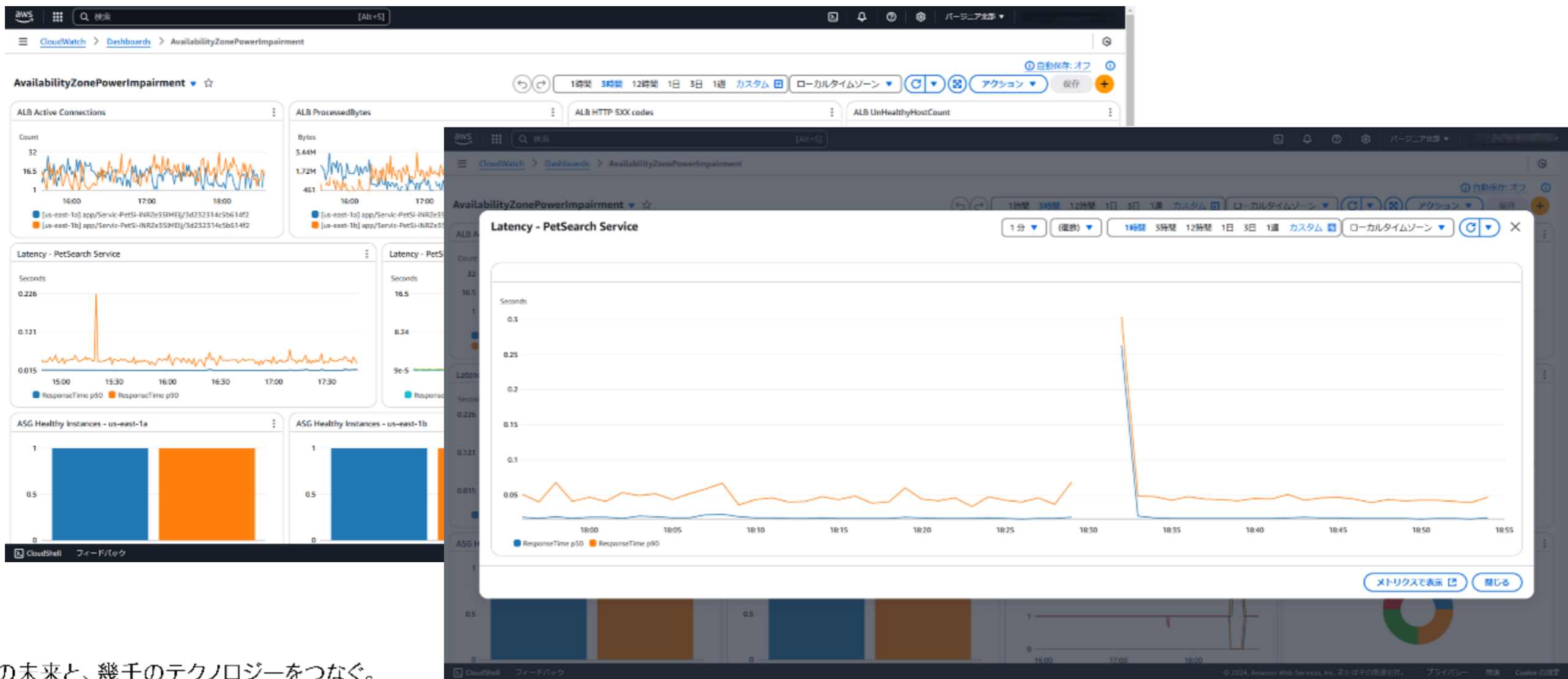
実験結果



無限の未来と、幾千のテクノロジーをつなぐ。

CTC Financial Services Group

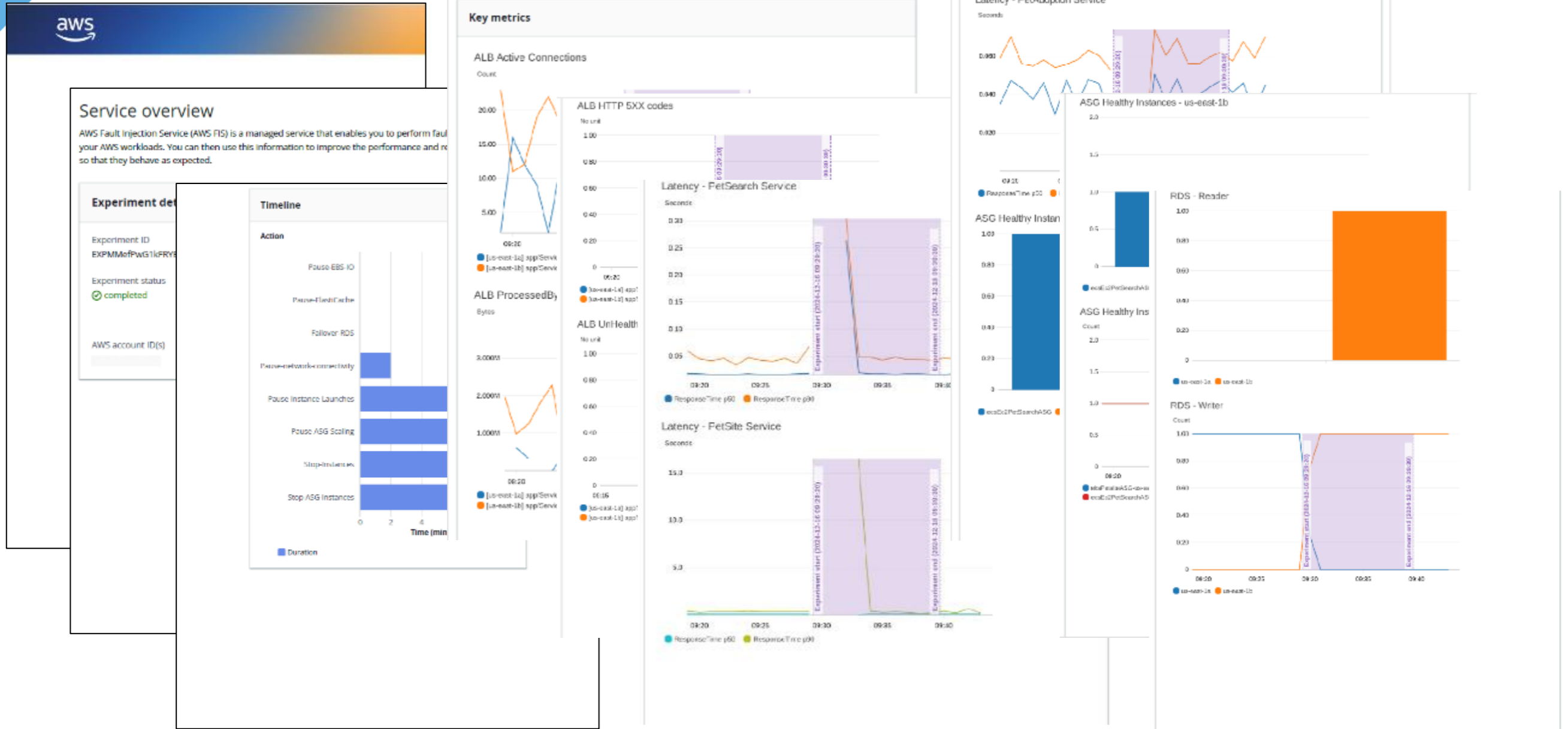
実験結果



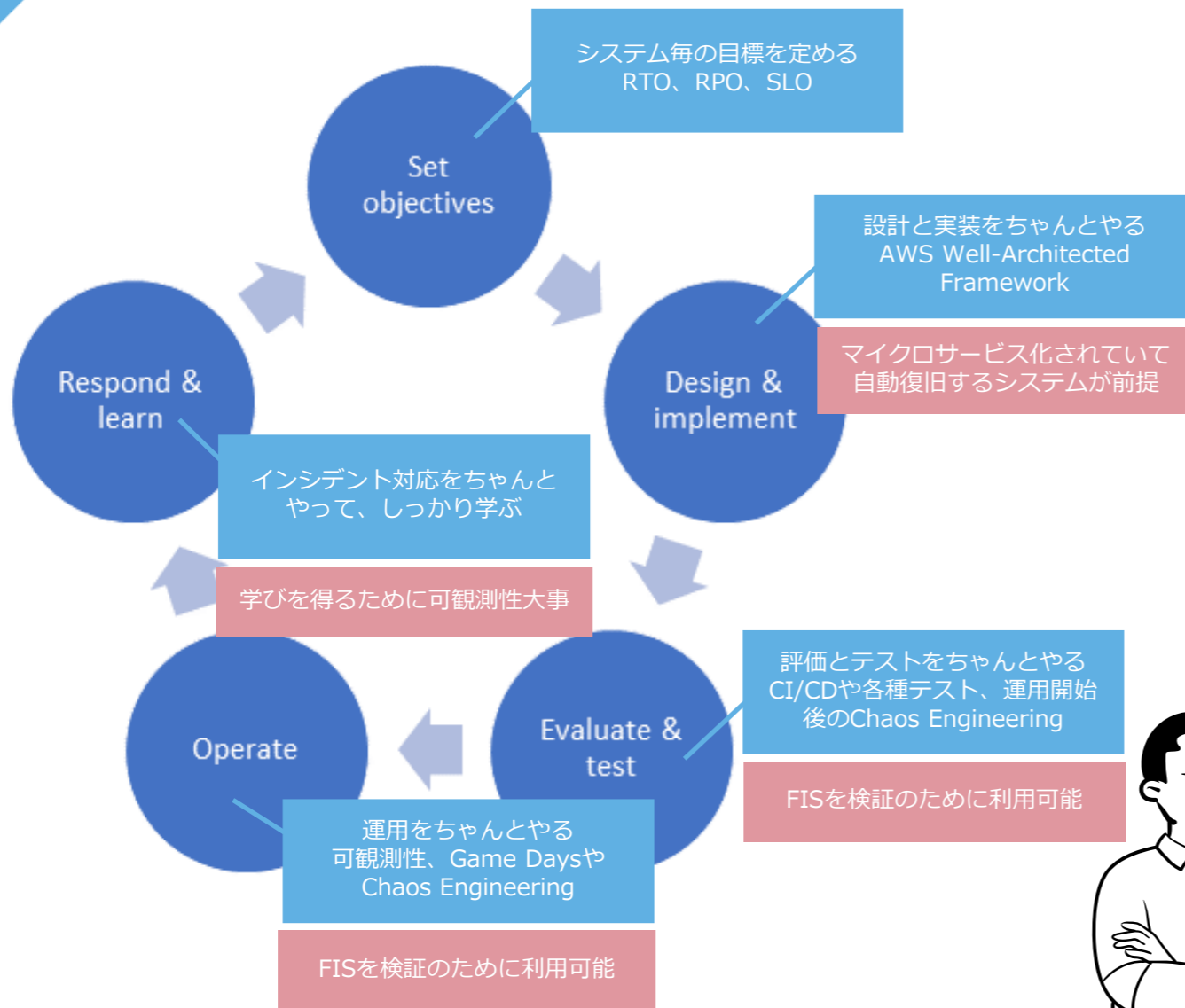
無限の未来と、幾千のテクノロジーをつなぐ。

CTC Financial Services Group

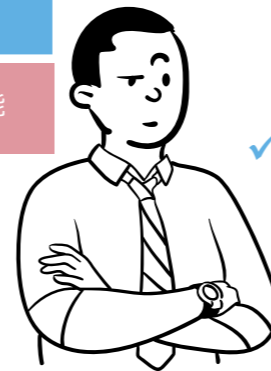
実験結果



感じたこと、考えたこと



- サービス影響出たらどーすんの！？
- ✓ 停止条件やセーフティレバーで影響範囲を局所化する
- ✓ 実験しないとわからない（仮説と反証）
- ✓ 想定外の事象が起きたときのリスクの方がビジネスとして大きな問題
- 障害テストとなにが違うの！？
- ✓ 個々の動作が想定通りであることの**確認**はやって当たり前
- ✓ 想定外の事象が起きたときの振る舞いを理解するためには**検証**が必要



Cf. [Resilience lifecycle framework: A continuous approach to resilience improvement](#)

さいごに

今日おぼえて帰って欲しいこと

無限の未来と、幾千のテクノロジーをつなぐ。

CTC Financial Services Group

このはなしの教訓

- 設計って大事
- テストって大事
- 可観測性って大事

でも、それだけでは「知らないし理解していない」ことは対処できない

実験を通じて、システムの理解を深めて自信をつけよう！

無限の未来と、幾千のテクノロジーをつなぐ。

CTC Financial Services Group

無限の未来と、
幾千のテクノロジーをつなぐ。

CTC Financial Services Group

