



Amazon CloudWatch Synthetics

AWS Black Belt Online Seminar

高野 翔史

Solutions Architect
2023/03

AWS Black Belt Online Seminarとは

- ・ 「サービス別」「ソリューション別」「業種別」などのテーマに分け、
アマゾン ウェブ サービス ジャパン合同会社が提供するオンラインセミナー
シリーズです
- ・ AWSの技術担当者が、AWSの各サービスやソリューションについてテーマご
とに動画を公開します
- ・ 動画を一時停止・スキップすることで、興味がある分野・項目だけの聴講も
可能、スキマ時間の学習にもお役立ていただけます
- ・ 以下のURLより、過去のセミナー含めた資料などをダウンロードするこ
とができます
 - ・ <https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-service-cut/>
 - ・ <https://www.youtube.com/playlist?list=PLzWGOASvSx6FIwIC2X1nObr1KcMCBBlqY>

内容についての注意点

- ・ 本資料では 2023 年 03 月時点のサービス内容および価格についてご説明しています。最新の情報は AWS 公式ウェブサイト (<https://aws.amazon.com/>) にてご確認ください
- ・ 資料作成には十分注意しておりますが、資料内の価格と AWS 公式ウェブサイト記載の価格に相違があった場合、AWS 公式ウェブサイトの価格を優先とさせていただきます
- ・ 価格は税抜表記となっています。日本居住者のお客様には別途消費税をご請求させていただきます

自己紹介

名前：高野 翔史

所属：エンタープライズ技術本部
ソリューションアーキテクト

経歴：国内SIerにてシステム開発に従事



好きなAWSサービス：Amazon CloudWatch、Amazon Timestream

本セミナーの対象者

- これからAWSを利用した運用設計を担当される方
- AWSでの監視設計を担当される方
- AWSを利用した外型監視に興味のある方

アジェンダ

Amazon CloudWatch Synthetics とは

Amazon CloudWatch Synthetics Canaryとは

Amazon CloudWatch Synthetics による Synthetics Monitoring

AWS モニタリングサービスとの連携

Amazon CloudWatch Synthetics 料金

まとめ

アジェンダ

Amazon CloudWatch Synthetics とは

Amazon CloudWatch Synthetics Canaryとは

Amazon CloudWatch Synthetics による Synthetics Monitoring

AWS モニタリングサービスとの連携

Amazon CloudWatch Synthetics 料金

まとめ

モニタリング (Monitoring) とは？

体系的な見直しを続けるために、一定期間にわたって、(何かの) 経過状況や品質を観察し確認する

observe and check the progress or quality of (something) over a period of time; keep under systematic review

引用 : Oxford English Dictionary

ユーザ体験の監視の重要性



サービスやアプリの健全性をチェックする



アプリケーションを元の状態に回復する



トラブルの原因を調査する



ユーザ行動を分析する



キヤパシティを分析する

ユーザ体験の監視の重要性



サービスやアプリの
健全性をチェックする



アプリケーションを
元の性能に回復する



トラブルの原因を
調査する

監視は ユーザ体験を損なわない



ために行う



ユーザ行動を
分析する

キャパシティを
分析する

ユーザ体験の監視手法

Synthetic Monitoring

※ Active monitoring の一種

- 定常に一定間隔で計測プログラムがアクセスしてデータを取得することを通じてパフォーマンスを計測する手法(主に製造業で利用される統計的品質管理手法と同じ思想に基づく計測手法)
- 短期のパフォーマンス計測用途に好適
- 一般に「外形監視」と訳される

Real User Monitoring

※ Passive monitoring の一種

- 実ユーザーアクセスの情報を取得して計測する手法
(実ユーザーによる実ブラウザでの実ロケーションからのアクセステータを取得・計測サーバーに送信)
- 中長期のユーザー傾向把握用途に好適

ユーザ体験の監視手法

Synthetic Monitoring

※ Active monitoring の一種

- 定常に一定間隔で計測プログラムがアクセスしてデータを取得することを通じてパフォーマンスを計測する手法(主に製造業で利用される統計的品質管理手法と同じ思想に基づく計測手法)
- 短期のパフォーマンス計測用途に好適
- 一般に「外形監視」と訳される



Amazon CloudWatch Synthetics

Real User Monitoring

※ Passive monitoring の一種

- 実ユーザーアクセスの情報を取得して計測する手法
(実ユーザーによる実ブラウザでの実ロケーションからのアクセステータを取得・計測サーバーに送信)
- 中長期のユーザー傾向把握用途に好適



Amazon CloudWatch RUM

ユーザ体験の監視手法

Synthetic Monitoring

※ Active monitoring の一種

- 定常的に一定間隔で計測プログラムがアクセスしてデータを取得することを通じてパフォーマンスを計測する手法(主に製造業で利用される統計的品質管理手法と同じ思想に基づく計測手法)
- 短期のパフォーマンス計測用途に好適
- 一般に「外形監視」と訳される

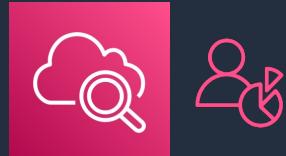


Amazon CloudWatch Synthetics

Real User Monitoring

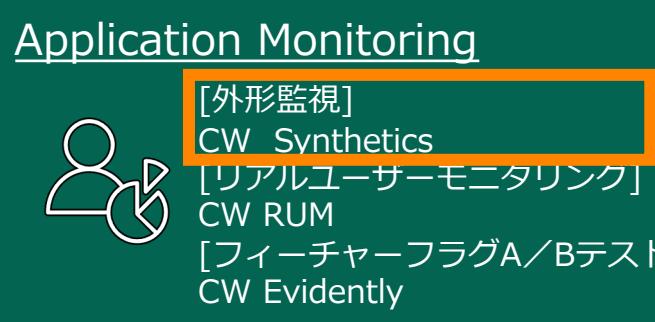
※ Passive monitoring の一種

- 実ユーザーアクセスの情報を取得して計測する手法
(実ユーザーによる実ブラウザでの実ロケーションからのアクセステータを取得・計測サーバーに送信)
- 中長期のユーザー傾向把握用途に好適



Amazon CloudWatch RUM

Amazon CloudWatchの全体像



[インターネット監視] CW Internet Monitor
[トレース] CW ServiceLens

[ダッシュボードに統合] CW Dashboard



Infrastructure



メトリクスを送信

ログを送信

イベントに応じた
ターゲットによる処理

[メトリクス] CW Metrics

[ログ] CW Logs

[アラーム] CW Alarms

メトリクスに応じた
アクション

CW Metrics Insight
[リアルタイムメトリクス分析]
CW Metrics Explorer
[タグベースの視覚化]



パートナー
サービス、
S3/Redshift

Amazon Kinesis
Data Firehose

[メトリクストリーム] CW Metrics Stream



[ログ分析] CW Logs Insights



[イベント]
Amazon EventBridge/
Amazon EventBridge
Scheduler

Insights



[構造化ログによるメトリクス]
CW Container Insights / Contributor Insights

[Lambda拡張機能によるメトリクス]

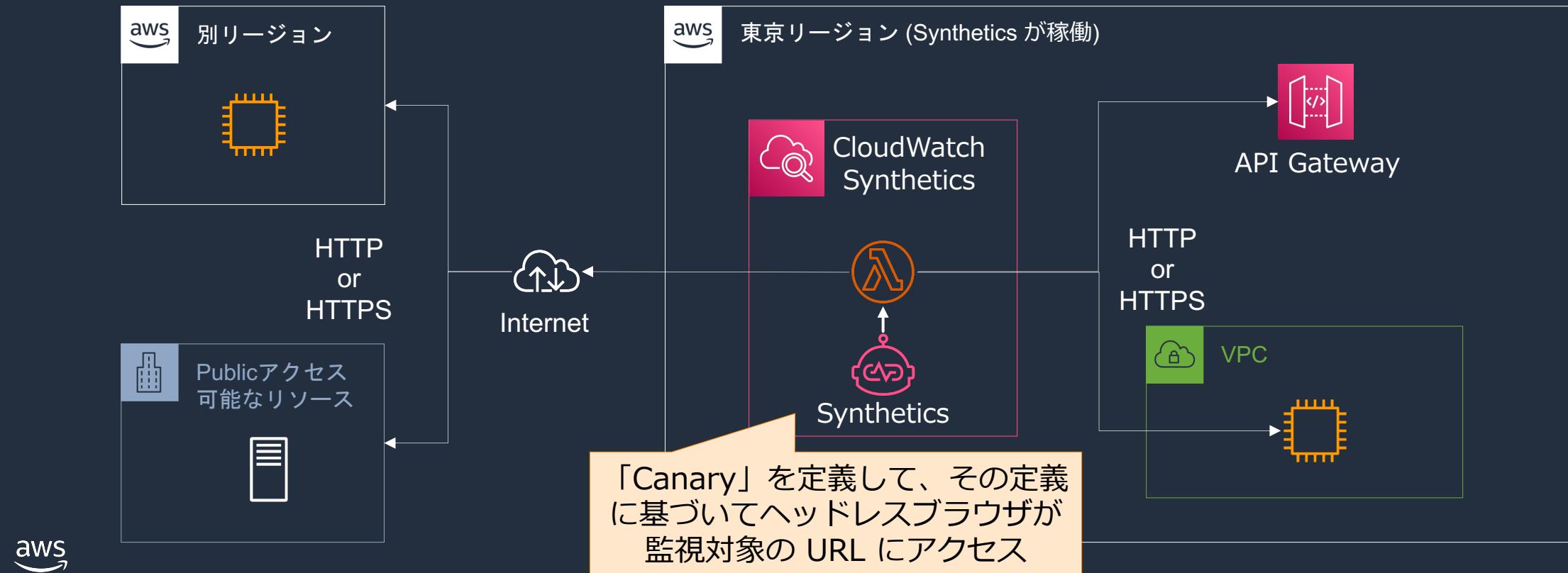
Lambda Insights

[アプリケーションコンポーネントのメトリクス]
Application Insights

※CW = CloudWatch

Amazon CloudWatch Synthetics による Monitoring

- CloudWatch Synthetics は、Web アプリケーションと API を簡単に監視できるようにするサービスです。
- 実際のユーザー トラフィックがなくてもエンドポイントを継続的にテストすることができます。
- 設定したしきい値に基づいてエラーがあればアラートを上げることができます。



Amazon CloudWatch Synthetics Canary 作成画面

設計図

- hardt-beat のモニタリング (選択済み)
- API Canary
- リンク切れチェッカー
- Canary レコーダー
- GUI ワークフロービルダー
- ビジュアルモニタリング

Canary ビルダー

名前

synthetics-demo

名前は、21 文字までの小文字、数字、ハイフン、またはアンダースコアで構成され、スペースを含めることはできません。

アプリケーションまたはエンドポイント URL 情報

http://[REDACTED] us-east-1.elb.amazonaws.com

削除

エンドポイントを追加

最大でさらに 4 個のエンドポイントを追加できます。スクリプトを変更することで、さらにエンドポイントを追加できます。

スクリーンショット

スクリーンショットを撮る
スクリーンショットは、Canary 実行ごとに Canary の詳細画面に表示されます

aws

Amazon Web Services, Inc. or its affiliates.

Blueprint が用意されており、
基本的なユースケースにおいては、
ノーコードで Canary を
作成できます (詳細は後程詳しく
紹介)。

Amazon CloudWatch Synthetics Canary 実行結果概要

Canary Canary のステータス一覧

3 時間 ▾

ステータス
現在実行中の Canary のステータス配分



● 成功 (5) ● 失敗 (3)

グループ
最終実行による

合計	失敗	アラーム
1	1 ⚠️	1 ⚠️

最も遅いパフォーマンス
最後の 3 時間

最も遅いグループ	最も遅いリージョン
18.2s	1. アジアパシフィック (東京) 2. 米国東部 (バージニア北部)

Canary (8)

リソースを検索 グループを表示 アクション リストを作成 Canary を作成

名前	前回の実行ステータス	成功 %	アラーム	平均時間	状態	ランタイムバージョン	リージョン
DemoGroup	④ 1/2 失敗	74%	⚠️ 1	18.2s	-	-	-
petsite	⑤ 成功	77%	⚠️ 1	17.9s	平均実行時間が表示される	米国東部 (バージニア北部)	
petsite-tokyo	⑥ 失敗	71%	-	18.6s	実行中	syn-nodejs-puppeteer-3.9	アジアパシフィック (東京)
petsite-heartbeat	⑤ 成功	74%	-	18.1s	実行中	syn-nodejs-puppeteer-3.9	米国東部 (バージニア北部)
petsite-python	⑤ 成功	-	-	-	実行中	syn-python-selenium-1.3	米国東部 (バージニア北部)
petsite-visu	⑤ 成功	63%	-	6.7s	実行中	syn-nodejs-puppeteer-3.9	米国東部 (バージニア北部)
petsite-visual	⑥ 失敗	0%	⚠️ 1	0ms	停止	syn-nodejs-puppeteer-3.9	米国東部 (バージニア北部)
test	⑤ 成功	0%	-	-	CloudWatch アラームを設定しており、条件にマッチしている場合は表示される	peteer-3.9	米国東部 (バージニア北部)
test-2-screenshotoff	⑥ 失敗	0%	-	-	-	peteer-3.9	米国東部 (バージニア北部)

aws

Amazon CloudWatch Synthetics のメリット

- 必要な時に素早く Synthetics Monitoring を実現
- モニタリング動作の振る舞いをプログラムで制御可能
 - シンプルな用途であれば、組み込み提供のテンプレート (Blueprint) を用いるだけでプログラミングは不要
 - カスタマイズにより複雑な操作も表現可能
- AWS サービスとの連携による高い柔軟性
 - S3, EventBridge, X-Ray などの AWS サービスと連携

アジェンダ

Amazon CloudWatch Synthetics とは

Amazon CloudWatch Synthetics Canaryとは

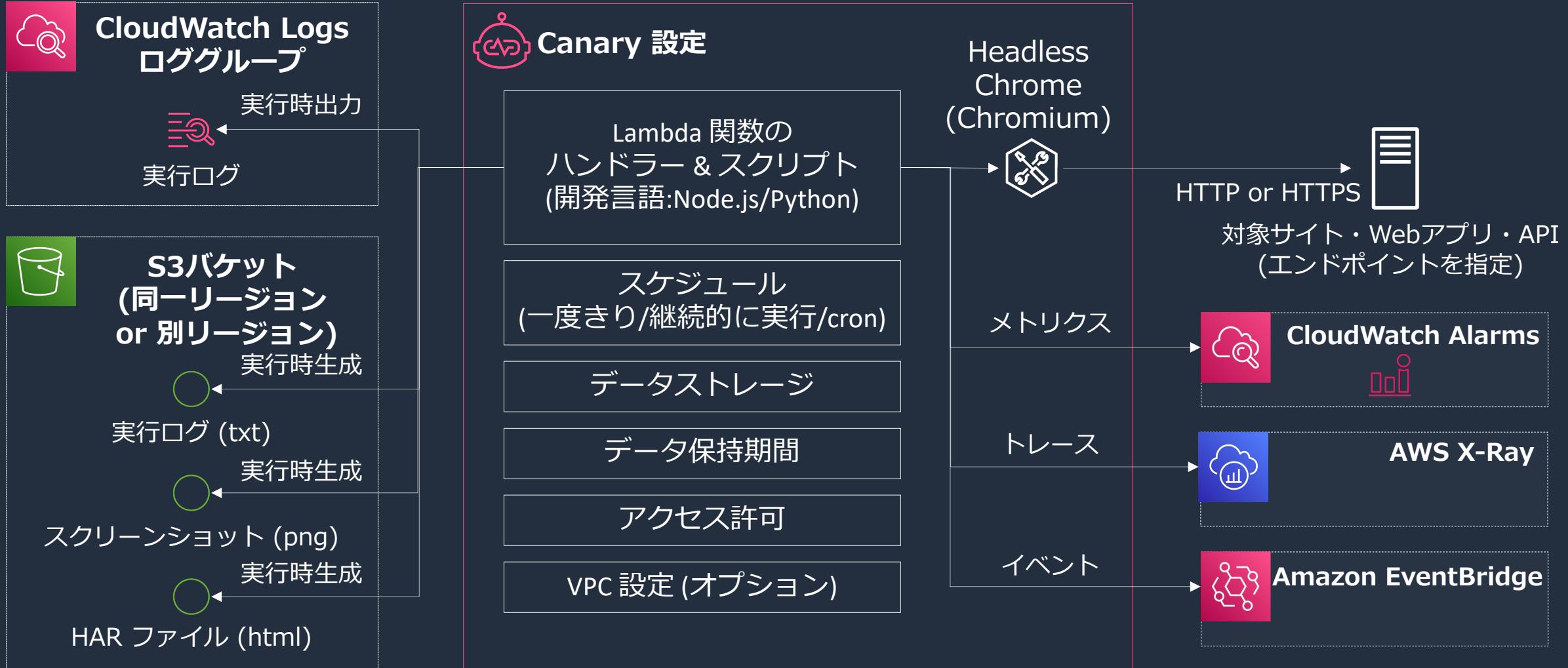
Amazon CloudWatch Synthetics による Synthetics Monitoring

AWS モニタリングサービスとの連携

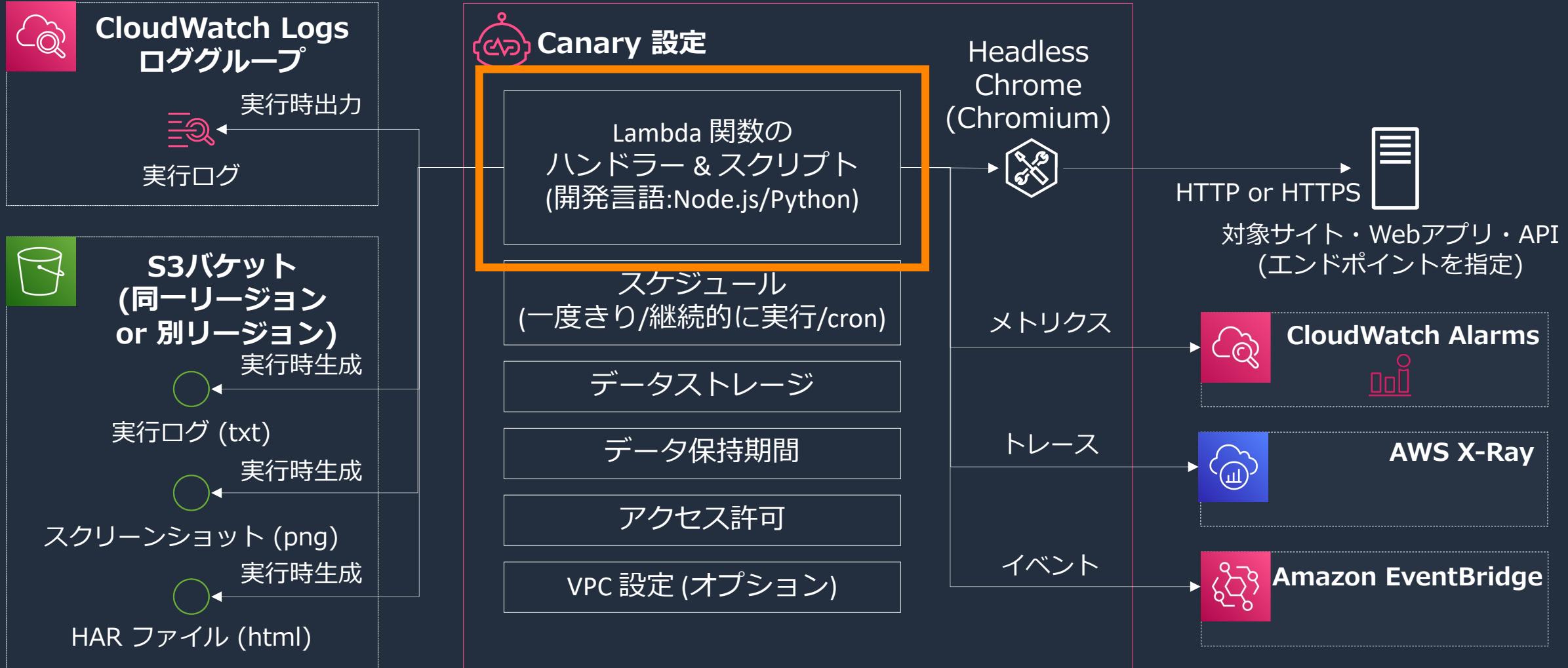
Amazon CloudWatch Synthetics 料金

まとめ

Amazon CloudWatch Synthetics Canary 全体構成イメージ



Amazon CloudWatch Synthetics Canary 全体構成イメージ



Canary 設定（スクリプト）

Lambda Layer として提供される Synthetics ランタイムを利用して、Node.js 又は Python コードでモニタリングしたい操作を記述します。



```
const { URL } = require('url');
const synthetics = require('Synthetics');
const log = require('SyntheticsLogger');
const syntheticsConfiguration = synthetics.getConfiguration();
const syntheticsLogHelper = require('SyntheticsLogHelper');

const loadBlueprint = async function () {
    const urls = [""];

    // Set screenshot option
    const takeScreenshot = true;

    :
}
```

Lambda Layer として AWS が提供するライブラリ

- ・ 設計図 (Blueprint) を利用するか、インラインエディタを使ってコーディングするか、S3 にスクリプトを配置してインポートするか 3 つのやり方を選択可能です。
- ・ Blueprint はユースケースに応じて用意されており、基本的な用途はコーディング不要で利用可能です（後程詳しく紹介）。
- ・ 環境変数を設定可能のため、コードを更新しなくても動作を調整可能です。
※環境変数は暗号化されないため、機密情報を利用する場合は AWS Secrets Manager と連携して下さい

※利用可能なAPIや詳細なコーディングについては

https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch_Synthetics_Canaries_Function_Library.html



CloudWatch Synthetics ランタイムについて

- 利用するプログラミング言語ごとに、ブラウザ操作ライブラリが異なります。

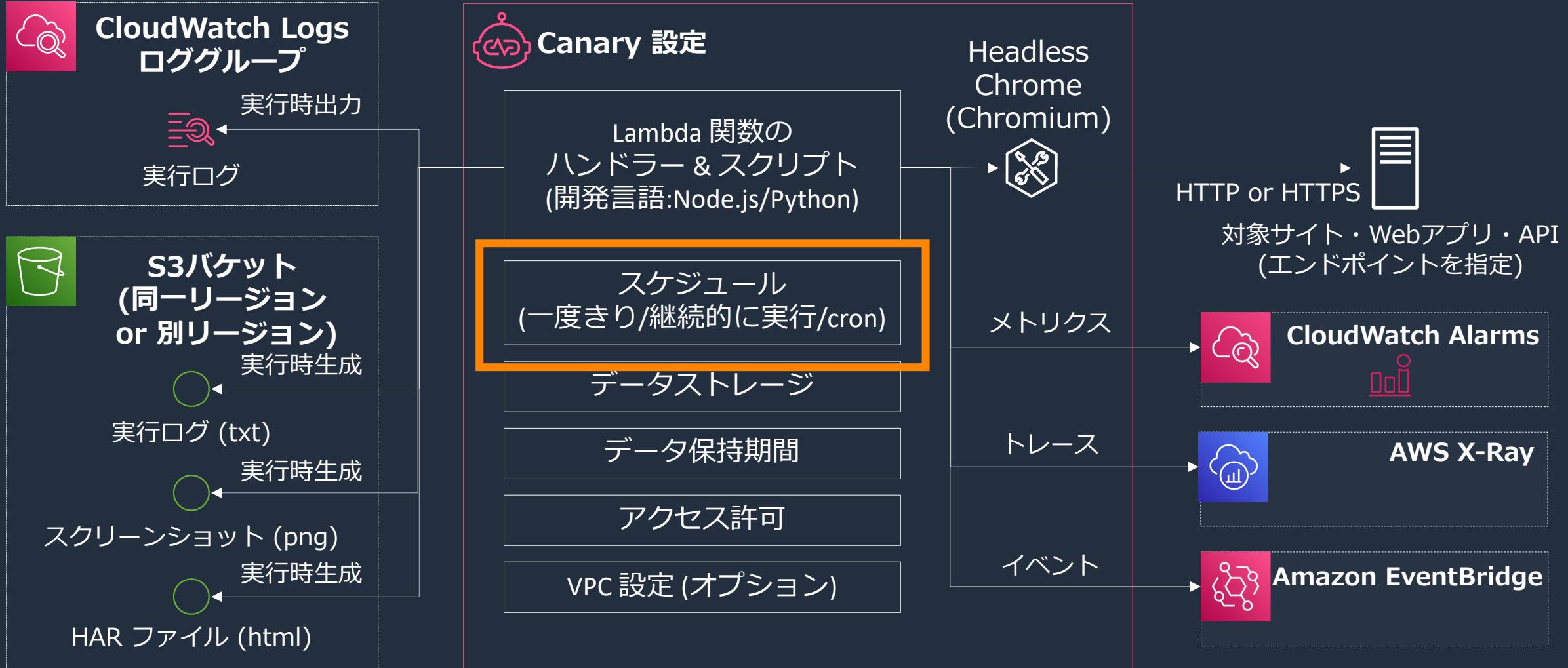
Lambda ランタイム	ブラウザ操作ライブラリ	ヘッドレスブラウザ
Node.js	Puppeteer	Chromium
Python	Selenium	Chromium

- Synthetics ランタイムのバージョンによって、Lambda ランタイム (Node.js/Python)、ブラウザ操作ライブラリ (Puppeteer/Selenium)、ヘッドレスブラウザ (Chromium) のバージョンが異なります。ランタイムのバージョンごとの各コンポーネントバージョンは下記 URL をご確認下さい。
 - Node.js と Puppeteer を利用するランタイムのバージョン
https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/CloudWatch_Synthetics_Library_nodejs_puppeteer.html
 - Python と Selenium Webdriver を使用するランタイムバージョン
https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/CloudWatch_Synthetics_Library_python.selenium.html

CloudWatch Synthetics ランタイム サポートポリシー

- Synthetics ランタイムに含まれているコンポーネントがサポートされなくなると、その Synthetics ランタイムのバージョンは**非推奨**になります。非推奨のランタイムバージョンを使用して新規に Canary を作成することはできません。既存で非推奨のランタイムバージョンを利用している Canary は利用可能ですが、**サポートされているランタイムバージョンへの更新を推奨**します。**60日以内**に非推奨となる予定のランタイムを使用している Canary がある場合に CloudWatch Synthetics は**通知メールを利用者に送信**します。サポートポリシー、ランタイムのバージョンアップ方法については、下記 URL をご確認ください。
- CloudWatch Synthetics ランタイムサポートポリシー
https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/CloudWatch_Synthetics_Canaries_Library.html#CloudWatch_Synthetics_Canaries_runtime_support

Amazon CloudWatch Synthetics Canary 全体構成イメージ



Canary 設定（スケジュール）

スケジュール 情報
この Canary を編集し、いつでも実行スケジュールを変更できます

「継続的に実行」、「CRON 式」、「1 回実行」の 3 つの中から 1 つを選択

継続的に実行
Canary 実行の頻度をスケジュールする

CRON 式
CRON 式で Canary をスケジュールする

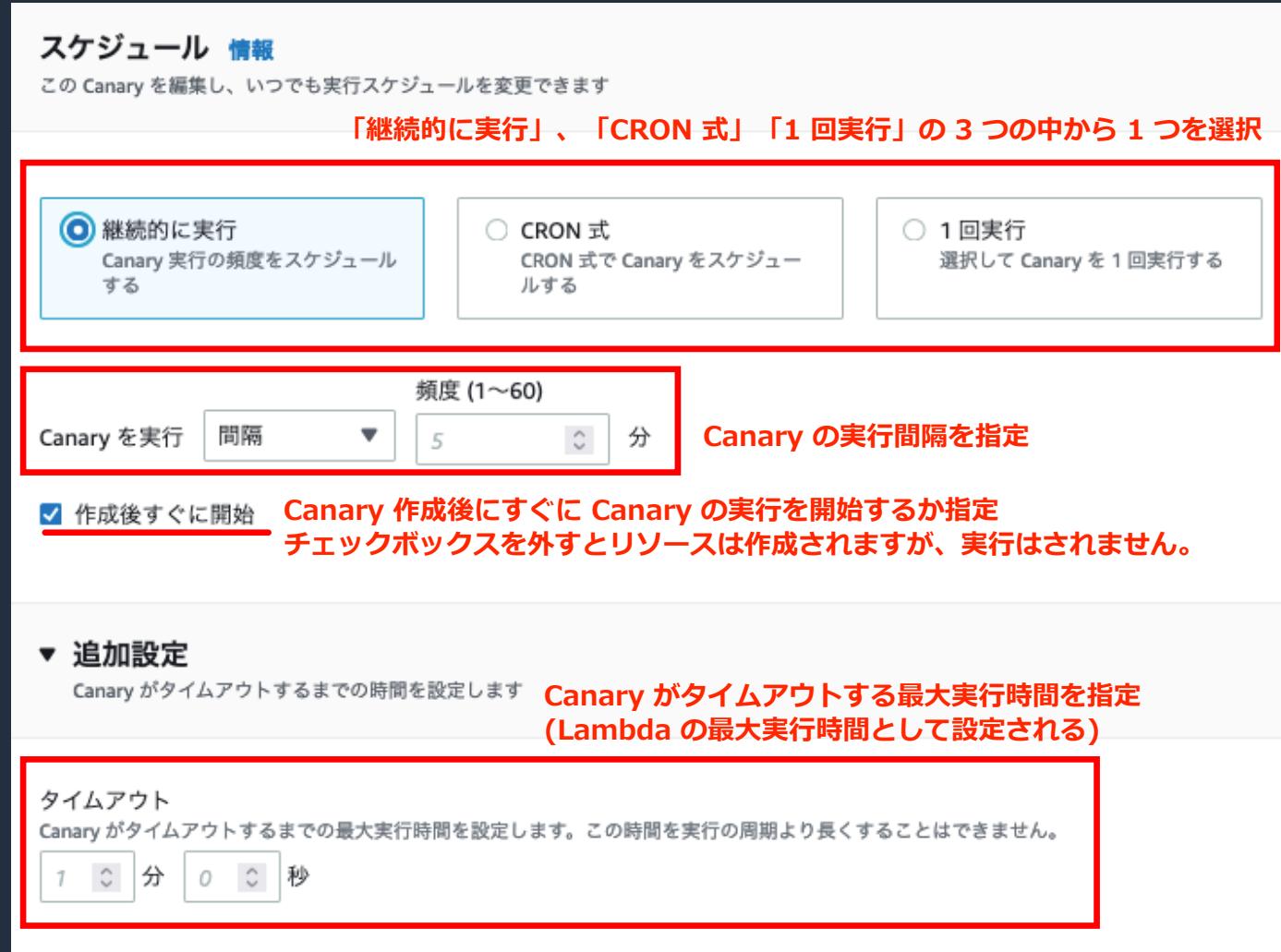
1 回実行
選択して Canary を 1 回実行する

Canary を実行 間隔 ▾ 5 分 Canary の実行間隔を指定

作成後すぐに開始 Canary 作成後にすぐに Canary の実行を開始するか指定
チェックボックスを外すとリソースは作成されますが、実行はされません。

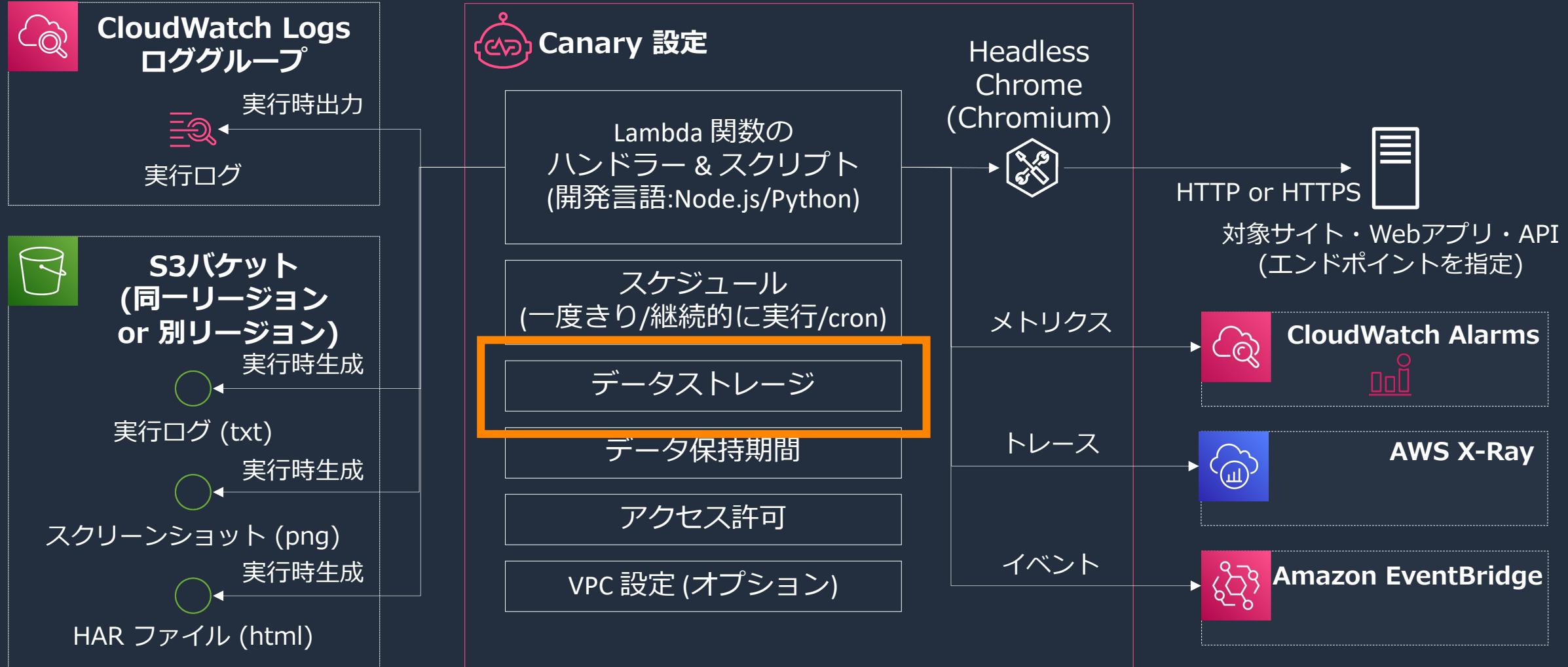
▼ 追加設定
Canary がタイムアウトするまでの時間を設定します Canary がタイムアウトする最大実行時間を指定 (Lambda の最大実行時間として設定される)

タイムアウト
Canary がタイムアウトするまでの最大実行時間を設定します。この時間を実行の周期より長くすることはできません。
1 分 0 秒



- Canary の実行スケジュールを指定します。以下の設定が可能です。
 - 継続的に実行
 - 間隔：1~60 分の間で指定
 - 毎日：タイムゾーン UTC で時刻を指定
 - CRON 式
 - CRON 式を使って柔軟に実行するタイミングを指定して継続的に実行する
 - 1 回実行
 - 1 回のみ実行する
- Canary がタイムアウトするまでの最大実行時間を設定可能です。
実行スケジュール間隔より長くは設定できません。
Lambda コールドスタート等の時間を許容するために 15 秒以上は長く設定して下さい。

Amazon CloudWatch Synthetics Canary 全体構成イメージ

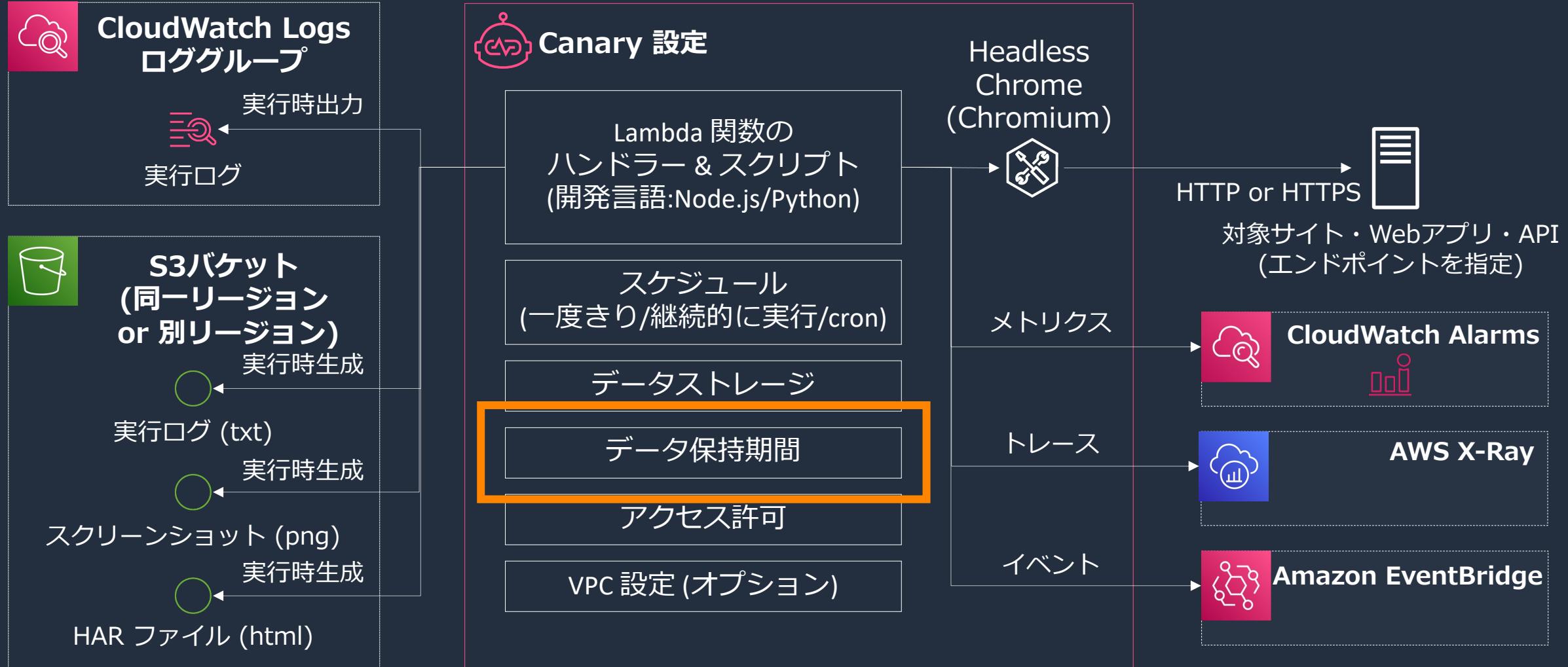


Canary 設定（データストレージ）

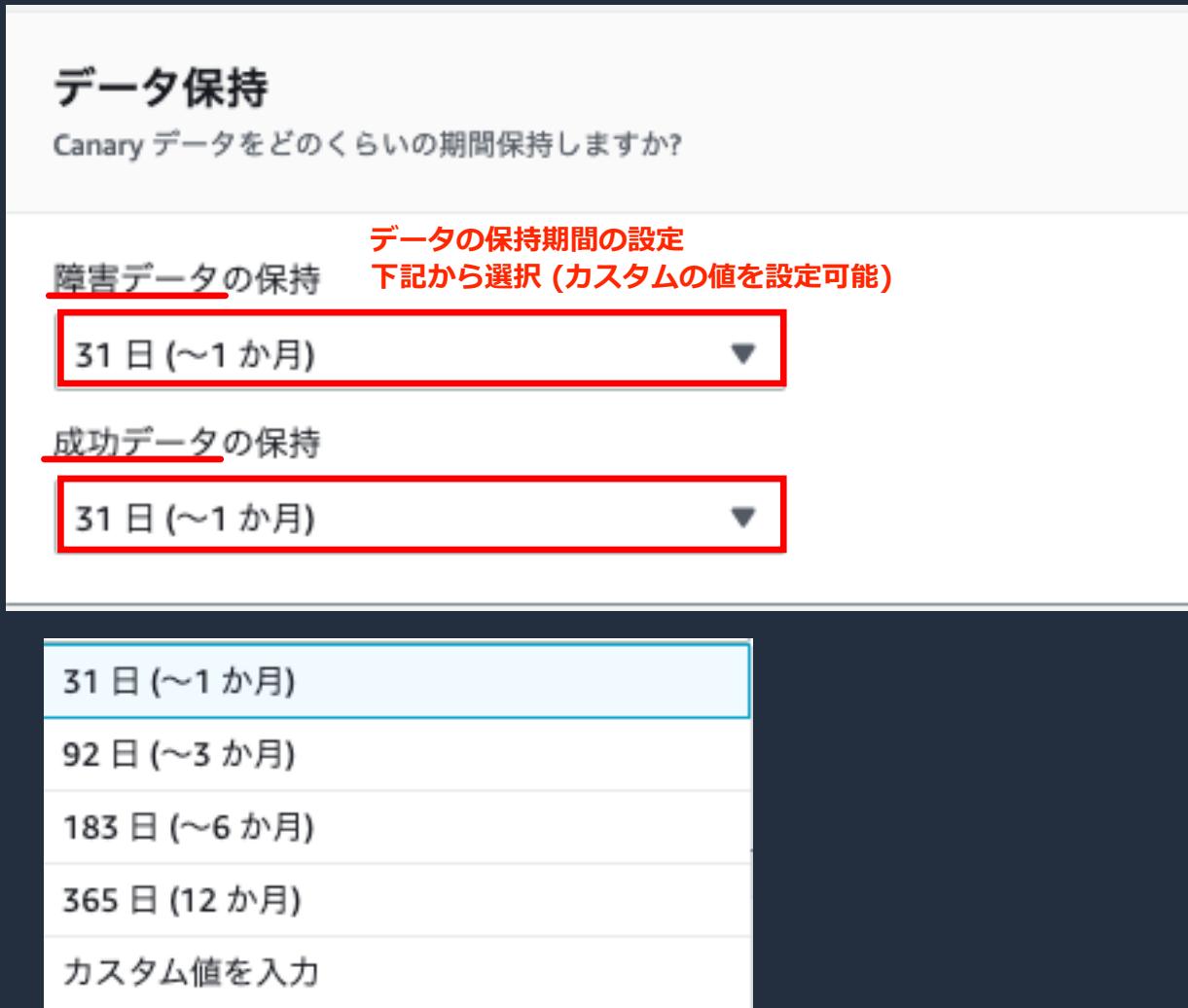
The screenshot shows the 'Data Storage' section of the CloudWatch Synthetics Canary configuration. It includes a note about saving artifacts to an S3 bucket and a link to S3 details. A red box highlights the 'S3 location' input field containing 's3://bucket/path/to/canary/artifacts/folder'. Below it, the 'Artifact output S3 bucket path selection' button is also highlighted in red. The 'Advanced Settings' section is expanded, showing the 'Encryption' tab with a note about using SSE-S3 or AWS KMS. A red box highlights the 'Encryption method selection' section, which includes three options: 'Select encryption method' (checked), 'SSE-S3 encryption' (radio button), and 'AWS Key Management Service (AWS KMS) usage' (radio button selected). The 'AWS KMS key selection or ARN input' field and 'AWS KMS key creation' button are also highlighted in red.

- Canary によって作成されるアーティファクト（ログ/スクリーンショット/HARファイル）の出力場所を指定します。
- デフォルトでは一番最初に Canary を作成する際は、同一リージョンに CloudWatch Synthetics 用に S3 バケットを作成して保存し、それ以降は作成された S3 バケットに保存するように設定されます。既存 S3 バケット（別リージョンでも可能）を指定することも可能です。
- 「syn-nodejs-puppeteer-3.3」以降のランタイムバージョンを使用している場合、アーティファクトの暗号化方法を選択でき、SSE-S3 暗号化又は SSE-KMS（カスタマーマネージドキー）を選択可能です。
※ 「syn-python-selenium」を利用する場合は本機能を利用することはできないことにご注意下さい。

Amazon CloudWatch Synthetics Canary 全体構成イメージ



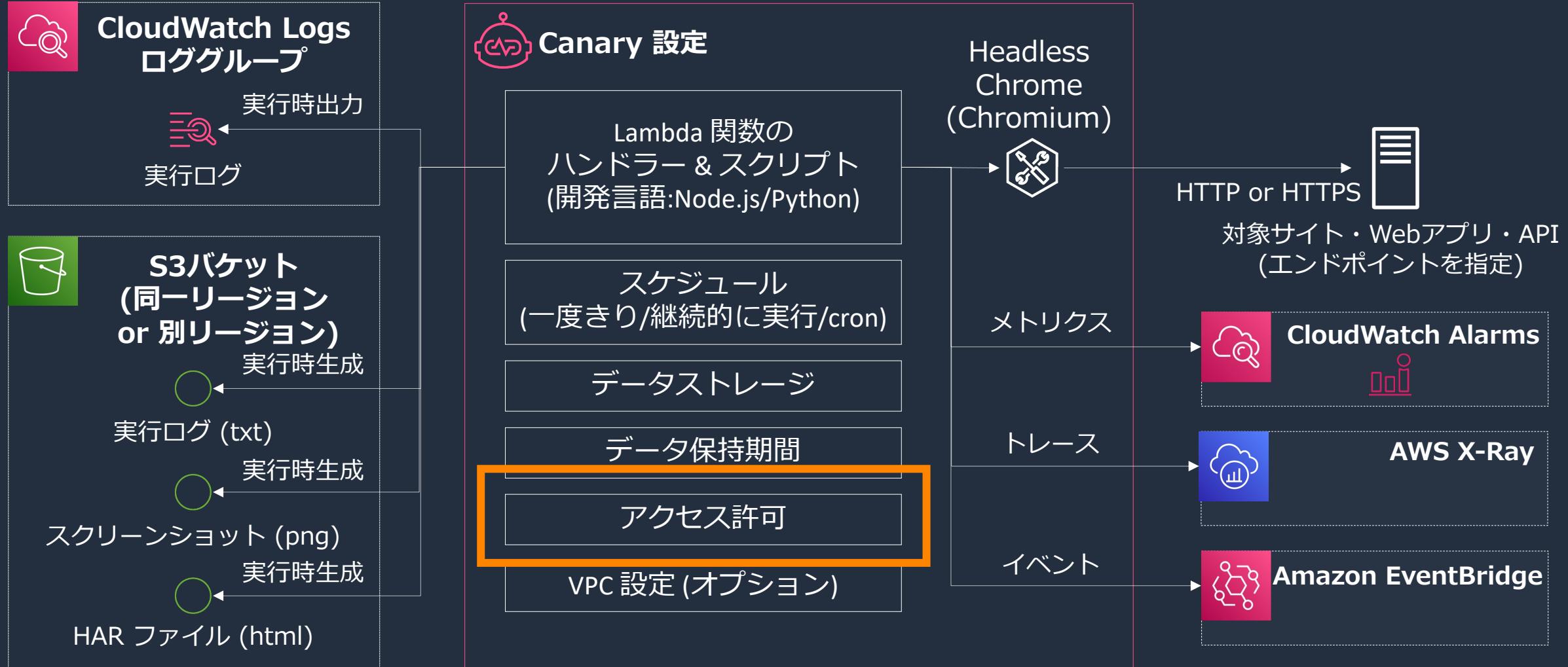
Canary 設定（データ保持期間）



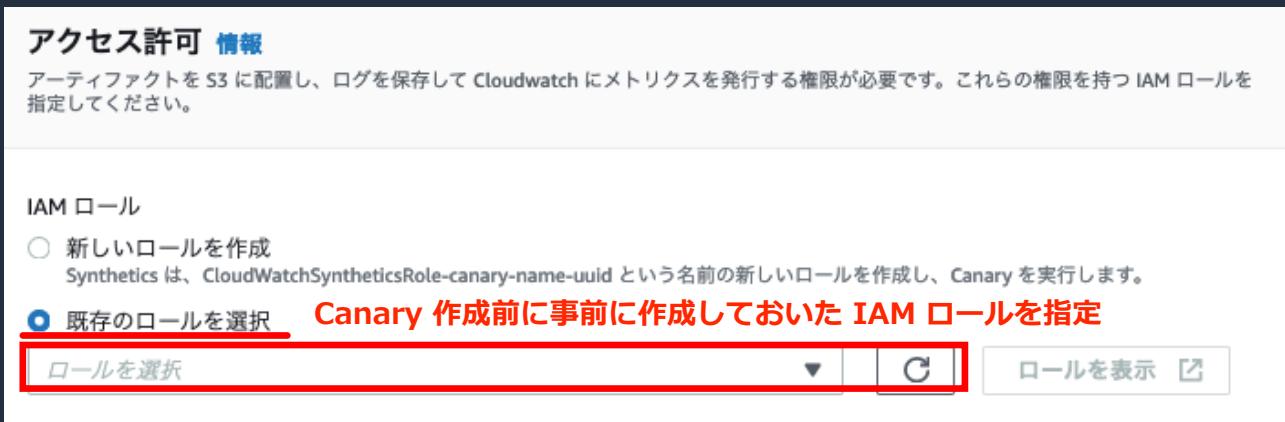
- 成功した Canary 実行結果と失敗した Canary 実行結果のそれぞれに対して、どれくらいの期間情報を保持するのか指定します。指定できる範囲は、1 - 455 日までです。

※あくまで AWS コンソールに保存および表示するデータに関する設定になります。S3 バケットに保存されるアーティファクト等には影響しない点はご注意下さい。

Amazon CloudWatch Synthetics Canary 全体構成イメージ

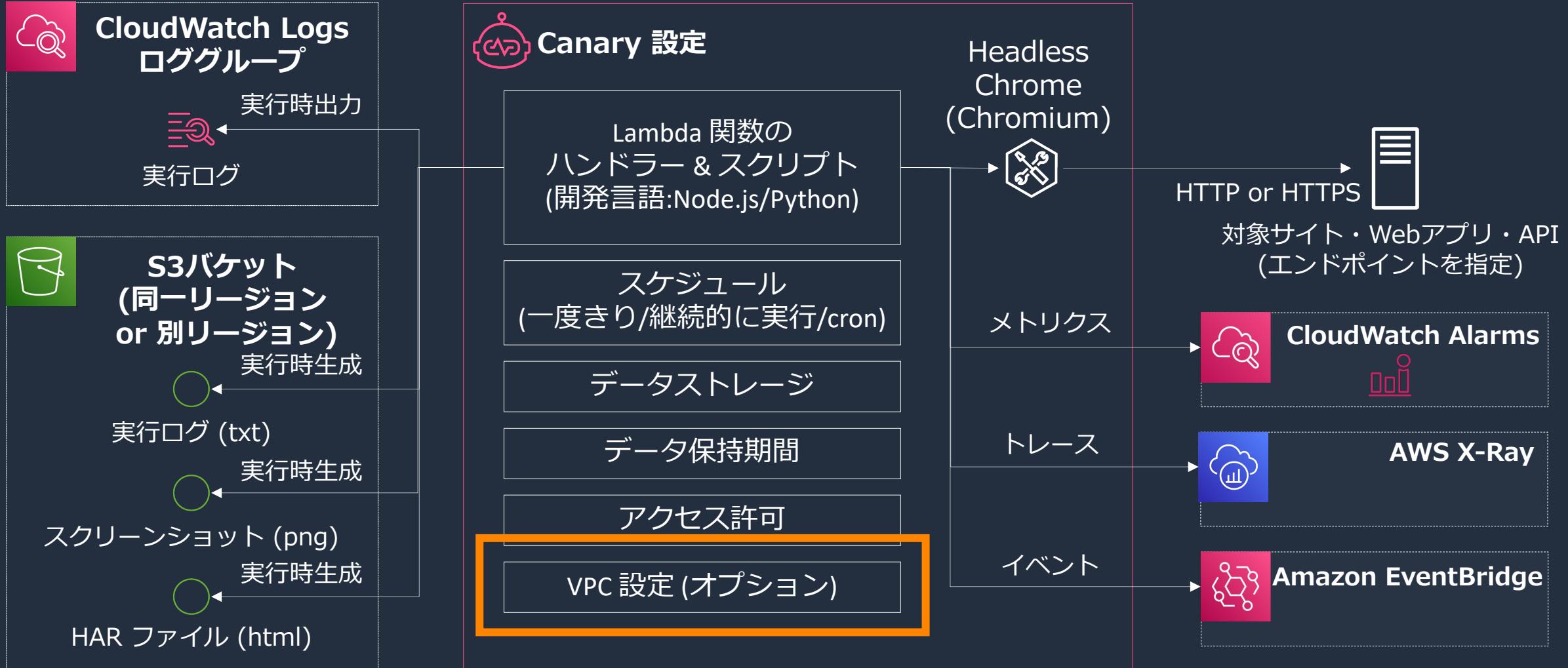


Canary 設定（アクセス許可）



- Canary によって作成されるアーティファクトを S3 バケットに保存したり、CloudWatch にメトリクスを発行するための IAM ロール (Lambda に設定される) を設定します。
- Canary 新規作成時に新しく作成するか、既存の IAM ロールを選択して設定することが可能です。
- 事前に IAM ロールを作成する場合は、構成によって必要な権限が異なります。下記を参照しながら IAM ロールを作成して下さい。
- Canaryに必要なロールとアクセス許可
https://docs.aws.amazon.com/ja_jp/AzonCloudWatch/latest/monitoring/CloudWatch_Synthetics_Canaries_CanaryPermissions.html

Amazon CloudWatch Synthetics Canary 全体構成イメージ



Canary 設定 (VPC 設定 - オプション)

▼ VPC 設定 - オプション 情報
エンドポイントがネットワーク下にある場合に使用します

Virtual Private Cloud (VPC) Canary を起動する VPC を指定
vpc- [REDACTED] (10.0.0.0/16)

① Synthetics は CloudWatch でメトリクスを作成し、スクリーンショット、HAR ファイル、ログなどの Canary 実行データを S3 に保存します。パブリックインターネットにアクセスせずに VPC を使用している場合は、特定のリージョンの CloudWatch メトリクスと S3 に VPC エンドポイントを追加してください。詳細はこちら [?]

サブネット Canary を起動するサブネットを指定
VPC 設定のセットアップに使用する Lambda の VPC サブネットを選択します。形式:「subnet-id (cidr-block) | az-name-tag」
オプションの選択

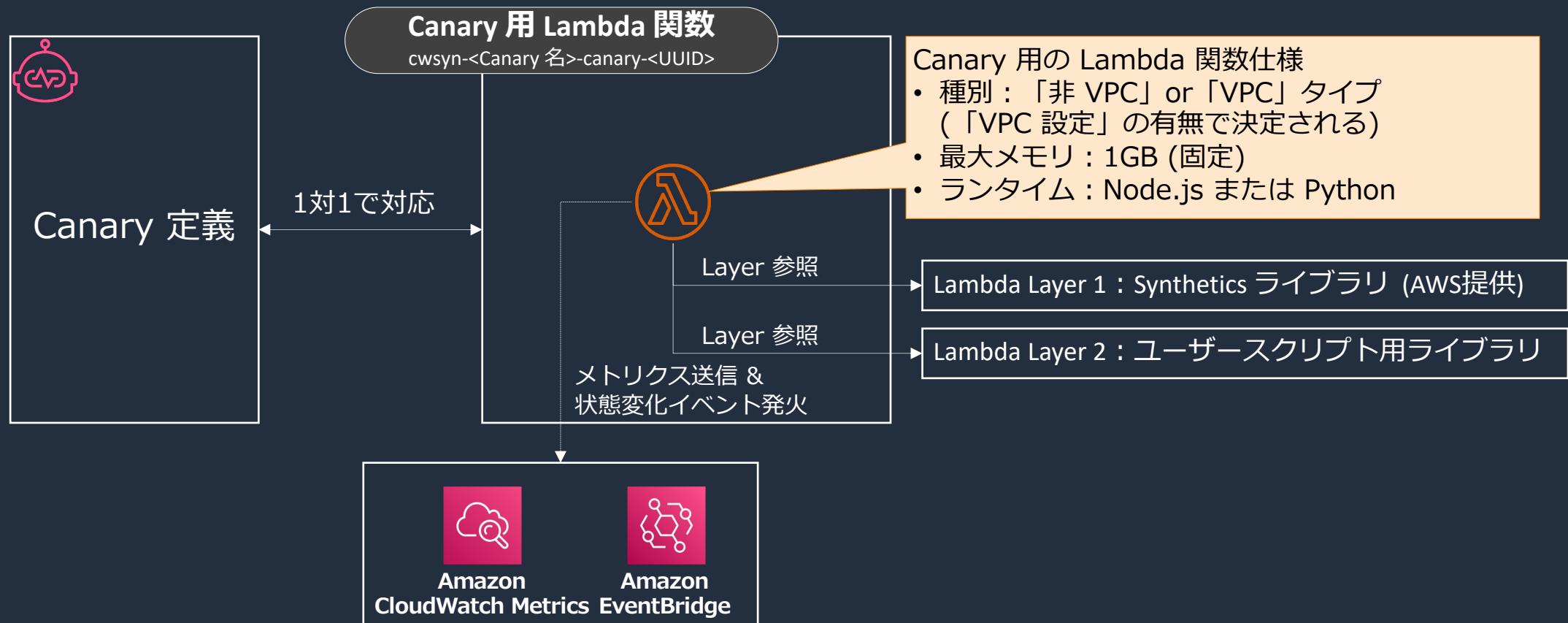
セキュリティグループ Canary に設定するセキュリティグループを指定
VPC 設定のセットアップに使用する Lambda の VPC セキュリティグループを選択します。形式:「sg-id (sg-name) | name-tag」。次の表に、選択したセキュリティグループのインバウンドルールとアウトバウンドルールを示します。
オプションの選択



- Canary は特定の VPC 内のリソースに対してもモニタリングすることが可能です。
- VPC 内の特定のサブネットから Canary (実体はLambda) を起動し、アーティファクトを S3 に保存したり、CloudWatch にメトリクスを送信できます。通信要件に応じてセキュリティグループを事前に用意して設定して下さい。
- VPC 内からインターネットにアクセスできず S3 等のパブリックエンドポイントにアクセスできない場合は、別途 VPC 内に VPC エンドポイントを作成して下さい。

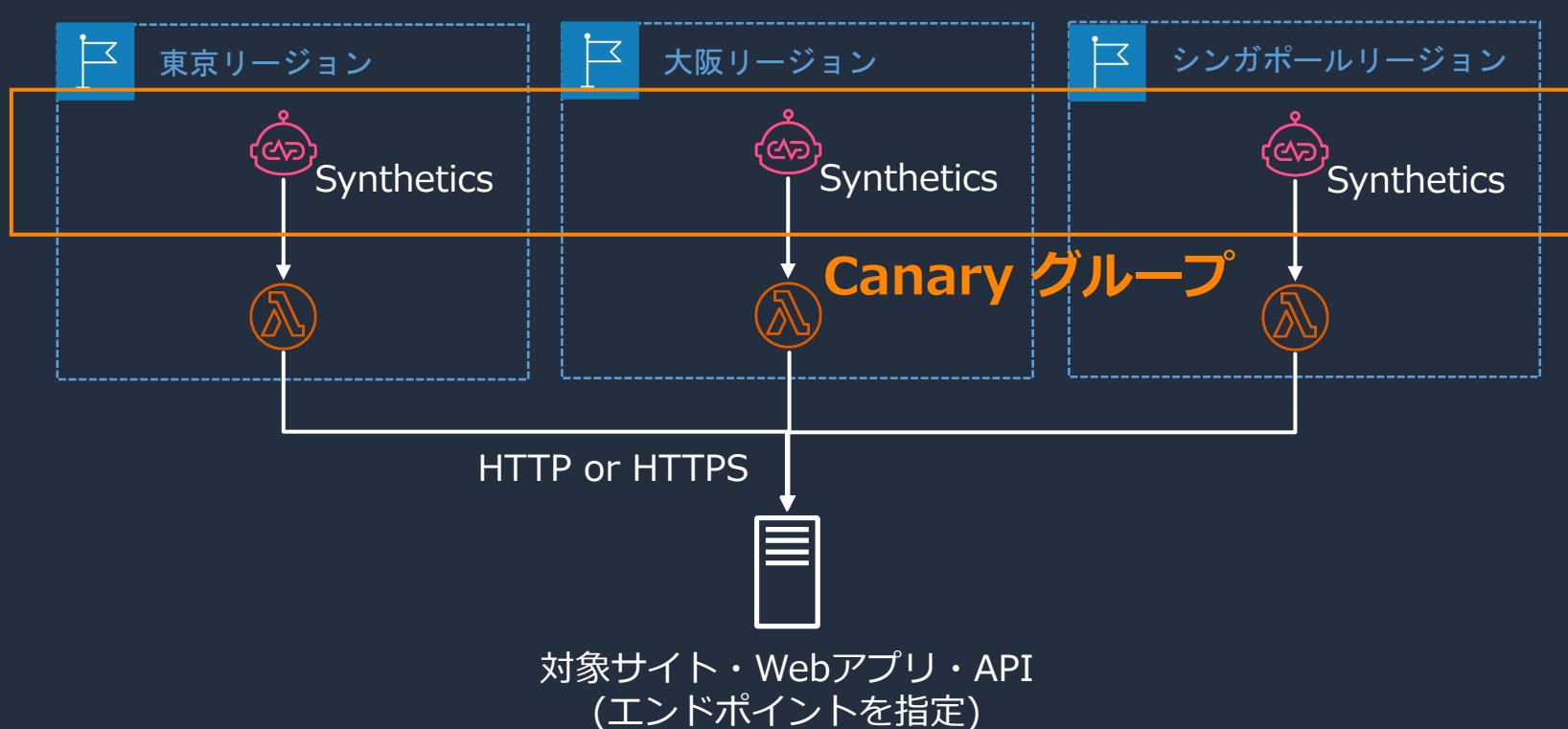
CloudWatch Synthetics Lambda 関数

Canary で定義したコードは、Lambda Layer として登録されて、CloudWatch Synthetics が作成する Lambda 関数コード内で読み込まれて実行されます。



Amazon CloudWatch Synthetics Canary グループ

リージョンを跨って、複数の Canary をグルーピングして
(1 グループ当たり最大 10) 、 Canary の結果をグループとして表示することが可能



- Canary グループ はグローバルリソースとして登録されるため、複数のリージョンから表示されます。
- 1 つの AWS アカウント当たり最大 20 個のグループを作成できます。
- 複数リージョンに跨ってグルーピングすることが可能なため、各リージョン固有の問題かどうかを簡単に切り分けることができるようになります。

Amazon CloudWatch Synthetics Canary グループ 作成

グループ名
グループの名前を入力します。

名前
DemoGroup
名前には最大 64 文字を含めることができます。任意の Unicode 文字を使用できます。

選択
グループ化する最大 10 個の Canary を選択します。

正確な Canary 名 グルーピングしたい Canary を完全一致で入力して検索する
 検索

petsite-tokyo アジアパシフィック (東京) petsite 米国東部 (バージニア北部)

名前 リージョン 検索の結果出力された Canary にチェックを入れると登録される

petsite-tokyo アジアパシフィック (東京)



- ・ グループ名を入力し、作成済みの Canary を検索して、選択することで、グループ内に登録することが可能です。
- ・ グループの検索は、Canary 名を完全一致で入力する必要があることに注意して下さい。
※途中まで入力して、検索してもヒットしません。

アジェンダ

Amazon CloudWatch Synthetics とは

Amazon CloudWatch Synthetics Canaryとは

Amazon CloudWatch Synthetics による Synthetics Monitoring

AWS モニタリングサービスとの連携

Amazon CloudWatch Synthetics 料金

まとめ

Canary Blueprint

Canary の実体は Node.js/Python で実装される Lambda 関数で、6 種類の Blueprint が用意されており、基本的な用途であればコーディングが不要です。

No.	Blueprint	特徴	Node.js	Python
1	ハートビートのモニタリング	指定した URL にアクセスして、ページのスクリーンショットと HTTP アーカイブファイル (HAR ファイル) を保存	○	○
2	API Canary	REST API に対してリクエストを送信して、応答をテスト	○	○
3	リンク切れチェッカー	テスト対象のURL内のすべてのリンクを収集し、リンク切れがないかテスト	○	-
4	Canary レコーダー	Google Chrome の拡張機能である CloudWatch Synthetics Recorder を利用して、ユーザ操作を記録し、テスト	○	-
5	GUI ワークフロービルダー	Web サイト上のユーザ操作ができるかを GUI ベースで作成してテスト	○	○
6	ビジュアルモニタリング	Web サイトの表示が変化していないかをベースラインと比較し、テスト	○	-

Canary 設計図の使用

https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/CloudWatch_Synthetics_Canaries_Blueprints.html

Canary Blueprint

Canary の実体は Node.js/Python で実装される Lambda 関数で、6 種類の Blueprint が用意されており、基本的な用途であればコーディングが不要です。

No.	Blueprint	特徴	Node.js	Python
1	ハートビートのモニタリング	指定した URL にアクセスして、ページのスクリーンショットと HTTP アーカイブファイル (HAR ファイル) を保存	○	○
2	API Canary	REST API に対してリクエストを送信して、応答をテスト	○	○
3	リンク切れチェッカー	テスト対象のURL内のすべてのリンクを収集し、リンク切れがないかテスト	○	-
4	Canary レコーダー	Google Chrome の拡張機能である CloudWatch Synthetics Recorder を利用して、ユーザ操作を記録し、テスト	○	-
5	GUI ワークフロービルダー	Web サイト上のユーザ操作ができるかを GUI ベースで作成してテスト	○	○
6	ビジュアルモニタリング	Web サイトの表示が変化していないかをベースラインと比較し、テスト	○	-

Canary 設計図の使用

https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/CloudWatch_Synthetics_Canaries_Blueprints.html



Canary Blueprint - ハートビートのモニタリング

設計図

- ハートビートのモニタリング
- API Canary
- リンク切れチェッカー
- Canary レコーダー
- GUI ワークフロービルダー
- ビジュアルモニタリング

Canary ビルダー

監視対象のURLを入力

名前
synthetics-demo

名前は、21 文字までの小文字、数字、ハイフン、またはアンダースコアで構成され、スペースを含めることはできません。

アプリケーションまたはエンドポイント URL 情報

http:// st-1.elb.a [削除]

エンドポイントを追加

syn-nodejs-puppeteer-3.1以上の場合、エンドポイントを複数追加可能

最大でさらに 4 個のエンドポイントを追加できます。スクリプトを変更することで、さらにエンドポイントを追加できます。

スクリーンショット

スクリーンショットを撮る **スクリーンショットの保存はチェックボックスで指定**

スクリーンショットは、Canary 実行ごとに Canary の詳細画面に表示されます

- 指定した URL にアクセスして、Web ページのスクリーンショット、HTTP アーカイブファイル (HAR ファイル)、実行ログを保存します。スクリーンショットについては保存するかチェックボックスで指定可能です。
- レスポンスのステータスコードをもとに正常/異常を判定します (デフォルトでは、ステータスコード 200 番台なら正常と判定するようにコード内で設定されています)。
- 「syn-nodejs-puppeteer-3.1」以降のランタイムバージョンを使用している場合、複数の URL をモニタリングすることが可能になります。
※ 「syn-python-selenium」を利用する場合は、複数の URL をモニタリングすることはできないため、ご注意下さい。

Canary 実行結果概要

Canary Canary のステータス一覧

ステータス
現在実行中の Canary のステータス配分



● 成功 (5) ● 失敗 (3)

Canary グループのステータス一覧

グループ
最終実行による

合計	失敗	アラーム
1	1 ⚠️	1 ⚠️

最も遅いパフォーマンス
最後の 3 時間

最も遅いグループ
18.2s

最も遅いリージョン
1. アジアパシフィック (東京)
2. 米国東部 (バージニア北部)

3 時間 ▾

Canary (8)

リソースを検索

すべて表示

すべてのリージョン

グループを表示

アクション ▾

グループを作成

Canary を作成

Canary グループはまとめて表示される

名前	前回の実行ステータス	成功 %	アラーム	平均時間	状態	ランタイムバージョン	リージョン
DemoGroup	② 1/2 失敗	74%	⚠️ 1	18.2s	-	-	-
petsite	③ 成功	77%	⚠️ 1	17.9s	実行中	syn-nodejs-puppeteer-3.9	米国東部 (バージニア北部)
petsite-tokyo	④ 失敗	71%	-	18.6s	実行中	syn-nodejs-puppeteer-3.9	アジアパシフィック (東京)
petsite-heartbeat	③ 成功	74%	-	18.1s	実行中	syn-nodejs-puppeteer-3.9	米国東部 (バージニア北部)
petsite-python	③ 成功	-	-	-	実行中	syn-python-selenium-1.3	米国東部 (バージニア北部)
petsite-visu	③ 成功	63%	-	6.7s	実行中	syn-nodejs-puppeteer-3.9	米国東部 (バージニア北部)
petsite-visual	④ 失敗	0%	⚠️ 1	0ms	停止	syn-nodejs-puppeteer-3.9	米国東部 (バージニア北部)
test	③ 成功	0%	-	-	-	peteer-3.9	米国東部 (バージニア北部)
test-2-screenshotoff	④ 失敗	0%	-	-	-	peteer-3.9	米国東部 (バージニア北部)

前回のステータスが表示される

CloudWatch アラームを設定しており、条件にマッチしている場合は表示される

Canary 実行結果詳細 (可用性 - ステップ)

失敗している場合は、選択することで失敗にフォーカスしたログを閲覧可能

問題 (334)
過去 24 時間

Error: Failed to load url: http://servi-petsi-
March 27, 2023 15:41

Error: Failed to load url: http://servi-petsi-
March 27, 2023 15:39

Error: Failed to load url: http://servi-petsi-
March 27, 2023 15:26

Error: Failed to load url: http://servi-petsi-
March 27, 2023 15:20

Error: Failed to load url: http://servi-petsi-
March 27, 2023 15:17

Error: Failed to load url: http://servi-petsi-
March 27, 2023 15:13

Canary 実行

詳細については、Canary のトラブルシューティングドキュメントを参照してください。 詳細は[こちら](#)

Canary の実行履歴が時系列で表示される

各ポイントは Canary 実行を表します。詳細については、各データポイントをクリックしてください。

3 時間 ▾

● 成功 ● 失敗

ステップ | スクリーンショット | ログ | HAR ファイル | トレース

実行されたステップ (1)

失敗したステップのみ < 1 > ⏪

ステップ	ステップ名	ステータス	説明	送信先 URL	期間	スクリーンショット
1	east-1.elb.amazonaws.com	成功	OK	http://[REDACTED]	798 ミリ秒	1

各ステップごとに実行結果が表示される

▶ Canary アーティファクトと S3 の場所

Canary 実行結果詳細 (可用性 - スクリーンショット)

ステップ | **スクリーンショット** | ログ | HAR ファイル | トレス

スクリーンショット (1) ヘッダレスブラウザで取得した Web ページ のスクリーンショットを表示

失敗したステップのみ X

The screenshot shows a web page titled "Observability PET ADOPTIONS". At the top, there are images of various pets: two rabbits, a grey rabbit, a yellow puppy, a black puppy, a calico kitten, and an orange kitten. Below this is a banner with the text "CUTE LITTLE PETS SEEKING A HOME". The main content area displays three dogs for adoption:

- A black and tan dog (Dachshund) with a "Take me home" button below it.
- A black and white puppy (Siberian Husky) with a "Take me home" button below it.
- A brown dog (Cocker Spaniel) with a "Take me home" button below it.

Each dog listing includes the name, price (\$ 89 or \$ 99), and a small image of the dog's face below the main image.

At the bottom of the screenshot, there are three smaller images of dog faces: a black and white puppy, a brown puppy, and another brown puppy.

On the right side of the screenshot, there is a status message: "☑ ステップに合格しました" and a link "http://[REDACTED].elb.am...".

Canary 実行結果詳細 (可用性 - ログ)

ログ

Canary の実行結果が出力される
CloudWatch Logs や S3に同じものが表示される

2023-03-14T05-21-34-476Z-log.txt

検索ログ

0/0

1 Start Canary

2 INFO: Event: {"canaryName": "petsite-heartbeat", "s3BaseFilePath": "cw-syn-results-[REDACTED]/canary/us-east-1/petsite-heartbe[REDACTED]"}
3 INFO: Context: {"callbackWaitsForEmptyEventLoop": true, "functionVersion": "1", "functionName": "cwsyn-petsite-heartbeat-[REDACTED]"}
4 INFO: Recording configuration:
5 INFO: Canary Name: petsite-heartbeat
6 INFO: Canary Arn: arn:aws:synthetics:[REDACTED]:canary:petsite-heartbeat
7 INFO: Canary lambda invoked at: Tue Mar 14 2023 05:21:34 GMT+0000 (Coordinated Universal Time)
8 INFO: AWS account Id: [REDACTED] and region us-east-1
9 INFO: S3 Artifact base location: cw-syn-results-[REDACTED]/canary/us-east-1/petsite-heartbeat-[REDACTED]
10 INFO: Artifacts will be encrypted using default KMS key for s3
11 INFO: Configuring tracing: canaryName: petsite-heartbeat canaryArn: arn:aws:synthetics:[REDACTED]:canary:petsite-heartbeat-[REDACTED]
12 INFO: Setting ActiveTracing to: true
13 INFO: memoryLimitInMB: 1000
14 INFO: awsRequestId: [REDACTED]
15 INFO: timeRemainingInMillis: 59998

Canary 実行結果詳細 (可用性 - HAR ファイル)

ステップ | スクリーンショット | ログ | **HAR ファイル** | トレス

Canary の実行した際に取得される HAR ファイルの内容が表示される
S3に同じものが表示される

Requests	Status code	Response size	Duration
http://servi-petsi-[REDACTED]-east-1.elb.amazonaws.com/	200 OK	118.9 KB	125.3ms
GET servi-petsi-[REDACTED]-east-1.elb.amazonaws.com/bootstrap.min.css	200 OK	152.1 KB	19.4ms
GET site.css	200 OK	1.3 KB	3.4ms
GET petstyles.css	200 OK	2.9 KB	24.1ms
GET brand.png	200 OK	3.9 KB	40.7ms
GET main_banner_text.png	200 OK	7.9 KB	78.7ms
GET p9.jpg?X-Amz-Security-Token=[REDACTED]	200 OK	88.2 KB	499.1ms
GET p10.jpg?X-Amz-Security-Token=[REDACTED]	200 OK	99 KB	537.3ms
GET p1.jpg?X-Amz-Security-Token=[REDACTED]	200 OK	78.3 KB	537.9ms
GET p11.jpg?X-Amz-Security-Token=[REDACTED]	200 OK	80 KB	537.5ms
GET p3.jpg?X-Amz-Security-Token=[REDACTED]	200 OK	78.3 KB	538.1ms
GET p6.jpg?X-Amz-Security-Token=[REDACTED]	200 OK	64.5 KB	538.3ms
GET jquery.min.js	200 OK	87.4 KB	62.4ms

Canary Blueprint

Canary の実体は Node.js/Python で実装される Lambda 関数で、6 種類の Blueprint が用意されており、基本的な用途であればコーディングが不要です。

No.	Blueprint	特徴	Node.js	Python
1	ハートビートのモニタリング	指定した URL にアクセスして、ページのスクリーンショットと HTTP アーカイブファイル (HAR ファイル) を保存	○	○
2	API Canary	REST API に対してリクエストを送信して、応答をテスト	○	○
3	リンク切れチェッカー	テスト対象のURL内のすべてのリンクを収集し、リンク切れがないかテスト	○	-
4	Canary レコーダー	Google Chrome の拡張機能である CloudWatch Synthetics Recorder を利用して、ユーザ操作を記録し、テスト	○	-
5	GUI ワークフロービルダー	Web サイト上のユーザ操作ができるかを GUI ベースで作成してテスト	○	○
6	ビジュアルモニタリング	Web サイトの表示が変化していないかをベースラインと比較し、テスト	○	-

Canary 設計図の使用

https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/CloudWatch_Synthetics_Canaries_Blueprints.html



Canary Blueprint - API Canary

The screenshot shows the configuration interface for a CloudFormation Canary Blueprint. It includes sections for API selection and an HTTP request step.

API 選択 情報
Amazon API Gateway API を使用中 [情報](#)
» Amazon API Gateway API を使用している場合は、このチェックボックスをオンにします。Canary には、Swagger テンプレートのアップロードなど、API Gateway に固有の追加オプションがあります。

Amazon API Gateway API を使用しています **チェックを入れることで、API Gateway の API を選択可能**

API を選択 情報
API とステージを選択するか、Swagger からテンプレートをアップロードできます。

API Gateway から API とステージを選択
 API Gateway Swagger テンプレートを使用

API
テストする API Gateway API
translate-api

同一 AWS アカウント、同一リージョンの API Gateway の API を選択可能

ステージ
API がデプロイされるステージ
dev

API Gateway で設定しているステージを選択する

HTTP リクエスト (1)
HTTP リクエストをステップとして追加します。順序を変更するには、ドラッグアンドドロップします。

HTTP リクエストを追加

呼び出しの順序	ステップ名	リソース	方法
1	検証 https://[REDACTED].execute-api.[REDACTED].amazonaws.com/translate	/translate	GET

- REST API に対して、リクエストを送信して、レスポンスが意図通りかどうかをテストできます。
- Amazon API Gateway と統合されており、API Gateway の API や ステージを Canary と同じ AWS アカウント、同じリージョンから選択したり、API Gateway から Swagger テンプレートをアップロードすることで、クロスアカウントやクロスリージョン API のモニタリングの設定が可能です。
- REST API の URL を直接指定することも可能です。

Canary Blueprint - API Canary HTTP リクエスト設定 1

HTTP リクエストの詳細
HTTP リクエストの詳細を入力します。この Canary に複数のリクエストを追加できます。

リソース
API Gateway 内に設定しているリソースパスを選択する
/translate

方法
API Gateway の対象リソースパスに設定されている HTTP リクエスト
メソッドを選択する
GET

URL クエリ文字列
クエリパラメータを入力
名前 値
ヘッダー 値
新しい文字列を追加 オプションのクエリ文字列を表示

ヘッダー
API リクエストとレスポンスに関連付けられたメタデータ
名前 値
ヘッダー 値
ヘッダーを追加 オプションのヘッダーを表示

データをリクエスト - オプション
必要に応じて、リクエストボディを指定する
1

HTTP リクエストとして下記を設定してテストが可能です。

- メソッド (GET/POST のみサポート)
- リソース
- URL クエリ文字列
- HTTP ヘッダー
- リクエストボディ

Canary Blueprint - API Canary HTTP リクエスト設定 2

レポート設定

ヘッダーとレスポンス本文をキャプチャ。 **Canary のヘッダーとレスポンスをキャプチャするか選択する**
ヘッダーとリクエスト/レスポンス本文に機密データが含まれている場合があります。これらの詳細は取得または保存したり、Canary 実行レポートに表示されたりしません。このデータをキャプチャすることを選択した場合、サービスはこの情報をログに記録または保存しないことに注意してください。データは S3 バケットにのみ保存されます。

ステップ名
後で Canary ステップを追跡できるように、Canary ステップに名前を付けます。

**複数ステップを定義する場合に、追跡しやすいように
Canary ステップに名前をつける**

ステップ失敗時に Canary 実行を続行する
このステップが失敗した場合は、他の Canary スクリプトの実行を続行します。

**複数ステップを定義する場合に、本ステップの後に設定されている
ステップを実行するか選択する**

別の呼び出しを保存して追加
追加の呼び出しにより、Canary の実行時間が長くなる可能性があります。

注意: レポート設定とステップは、ランタイムバージョン syn-nodejs-2.2 以降でのみサポートされます。

「syn-nodejs-2.2」以降のランタイムバージョンを利用してい る場合は、オプションとして、以下の機能が利用可能です。

- ヘッダーとレスポンス本文をキャプチャする
※機密情報が記録される可能性がある場合は取り扱いに注意して下さい。
- API を複数モニタリングする際に、複数ステップに分けて実行結果を表示する Canary の作成

※ 「syn-python-selenium」を利用することはできないため、ご注意下さい。

Canary 実行結果詳細 (可用性 - HTTP リクエスト)

HTTP ステップ | **HTTP リクエスト** | ログ | トレース

HTTP リクエスト (2) 情報
リクエスト/レスポンスのヘッダーと本文はスクリプト設定に基づいて利用できます

ステータスコードやリクエストにかかる時間が表示される

失敗したリクエストのみを表示 < 1 > ○

リクエスト	ステータスコード	説明	期間
https://east-1.amazonaws.com...	200	OK	1863 ミリ秒

GET https://[REDACTED]

リクエストの詳細

リクエストサイズ - 98 Bytes

リクエストヘッダー 98 Bytes

リクエスト本文 0 Bytes

レスポンスサイズ - 489 Bytes

レスポンスヘッダー 466 Bytes

レスポンス本文 23 Bytes

リクエスト詳細 (展開)

リクエストヘッダー

User-Agent : [REDACTED]
X-Amzn-Trace-Id : [REDACTED]

リクエスト本文

レスポンス詳細 (展開)

レスポンスヘッダー

content-type : application/json
content-length : 23
connection : close
date : Tue, 28 Mar 2023 14:10:46 GMT
x-amzn-requestid : [REDACTED]
x-amz-apigw-id : [REDACTED]
x-amzn-trace-id : [REDACTED] 25
x-cache : Miss from cloudfront
via : 1.1 [REDACTED] Front.net (CloudFront)
x-amz-cf-pop : IAD55-P4
x-amz-cf-id : [REDACTED] fMIIaWAWbn6_bQ==

レスポンス本文

ヘッダーとレスポンス本文をキャプチャすると
どのようなリクエストをした結果
どのようなレスポンスが返ってきたか
確認できる

aws

Canary Blueprint

Canary の実体は Node.js/Python で実装される Lambda 関数で、6 種類の Blueprint が用意されており、基本的な用途であればコーディングが不要です。

No.	Blueprint	特徴	Node.js	Python
1	ハートビートのモニタリング	指定した URL にアクセスして、ページのスクリーンショットと HTTP アーカイブファイル (HAR ファイル) を保存	○	○
2	API Canary	REST API に対してリクエストを送信して、応答をテスト	○	○
3	リンク切れチェッカー	テスト対象のURL内のすべてのリンクを収集し、リンク切れがないかテスト	○	-
4	Canary レコーダー	Google Chrome の拡張機能である CloudWatch Synthetics Recorder を利用して、ユーザ操作を記録し、テスト	○	-
5	GUI ワークフロービルダー	Web サイト上のユーザ操作ができるかを GUI ベースで作成してテスト	○	○
6	ビジュアルモニタリング	Web サイトの表示が変化していないかをベースラインと比較し、テスト	○	-

Canary 設計図の使用

https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/CloudWatch_Synthetics_Canaries_Blueprints.html

Canary Blueprint - リンク切れチェッカー

設計図

- hardt-beat のモニタリング
- API Canary API を HTTP ステップとしてモニタリングします。
- リンク切れチェッカー** 指定された URL で基本的なウェブクローラーを実行します。 (選択)
- Canary レコーダー AWS Canary Recorder ブラグインを使用します。
- GUI ワークフロービルダー 実行するアクションを含む GUI ワークフローを作成します。
- ビジュアルモニタリング 実行ごとに視覚的な変更をモニタリングする

Canary ビルダー

名前: synthetics-link-demo

名前は、21 文字までの小文字、数字、ハイフン、またはアンダースコアで構成され、スペースを含めることはできません。

アプリケーションまたはエンドポイント URL 情報
http://[REDACTED].st-1.elb.amazonaws.com

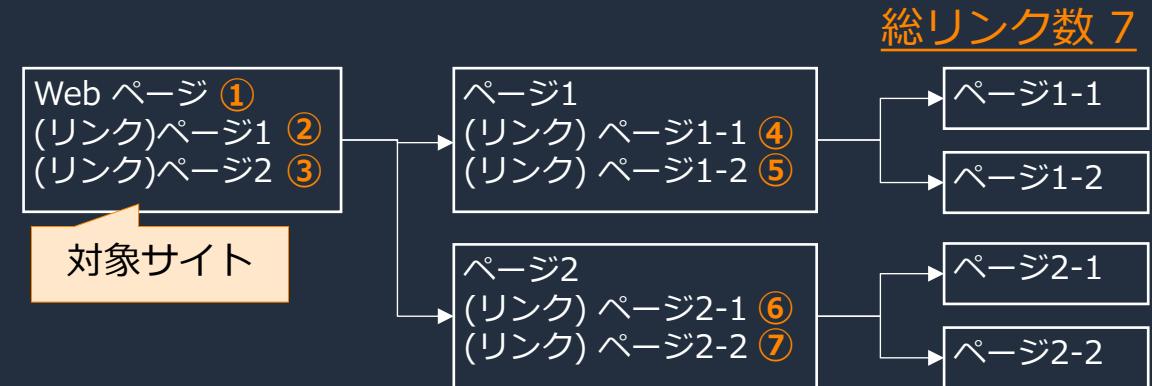
テストするエンドポイント、API、または URL を入力します。

フォローされるリンクの最大数: 5

モニタリングするリンクの数を指定する

スクリーンショット 情報
 スクリーンショットを撮る
スクリーンショットは、Canary 実行ごとに Canary の詳細画面に表示されます

- 指定した URL にアクセスして、Web ページに設定されているすべてのリンクを収集し、指定したリンク数までテストを実行します。リンク数のカウントは、指定した URL 自体が最初のリンクとしてカウントされることに注意してください。



- リンクに異常があれば、エラーと判定されます。リンク切れチェッカーで検出できるエラーの種類は下記を参照して下さい。
 - リンク切れチェッカー
https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/CloudWatch_Synthetics_Canaries_Blueprints.html#CloudWatch_Synthetics_Canaries_Blueprints_Broken_Links

Canary 実行結果詳細 (可用性 - チェックしたリンク)

チェックしたリンク | スクリーンショット | ログ | HAR ファイル

リンク (4) 指定した URL 内にあるリンクをチェックした結果がリンク毎に表示される

壊れたリンクのみ < 1 > ⏪

送信先 URL	ステータスコード	説明	アンカーテキスト	スクリーンショット
http://[REDACTED] [REDACTED]	200	OK	-	2
http://[REDACTED] [REDACTED]	200	OK	See Adoption List	2
http://[REDACTED] [REDACTED]	200	OK	Perform Housekeeping	2

Canary Blueprint

Canary の実体は Node.js/Python で実装される Lambda 関数で、6 種類の Blueprint が用意されており、基本的な用途であればコーディングが不要です。

No.	Blueprint	特徴	Node.js	Python
1	ハートビートのモニタリング	指定した URL にアクセスして、ページのスクリーンショットと HTTP アーカイブファイル (HAR ファイル) を保存	○	○
2	API Canary	REST API に対してリクエストを送信して、応答をテスト	○	○
3	リンク切れチェッカー	テスト対象のURL内のすべてのリンクを収集し、リンク切れがないかテスト	○	-
4	Canary レコーダー	Google Chrome の拡張機能である CloudWatch Synthetics Recorder を利用して、ユーザ操作を記録し、テスト	○	-
5	GUI ワークフロービルダー	Web サイト上のユーザ操作ができるかを GUI ベースで作成してテスト	○	○
6	ビジュアルモニタリング	Web サイトの表示が変化していないかをベースラインと比較し、テスト	○	-

Canary 設計図の使用

https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/CloudWatch_Synthetics_Canaries_Blueprints.html



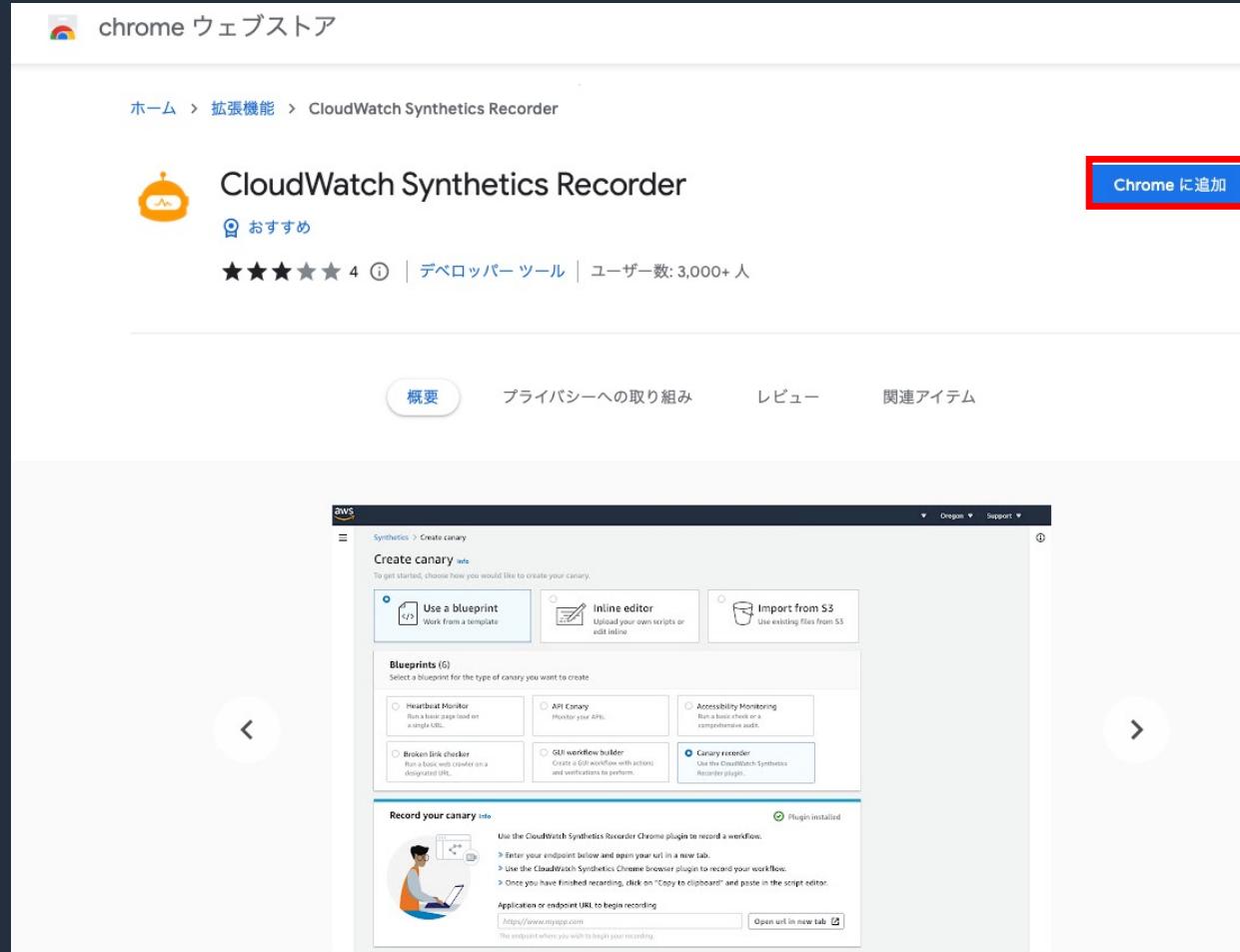
Canary Blueprint - Canary レコーダー

以下流れで記録したユーザ操作を Canary で再現し、モニタリングすることが可能。

- ① Google Chrome の拡張機能である CloudWatch Synthetics Recorder をChrome に追加
- ② CloudWatch Synthetics Recorder を利用して、Web サイトでの操作を記録して、Node.js のコードとして出力
- ③ 出力したコードをコピーし Canary 設定画面にペーストして、登録

Canary Blueprint - Canary レコーダー

① CloudWatch Synthetics Recorder を Chromeに追加

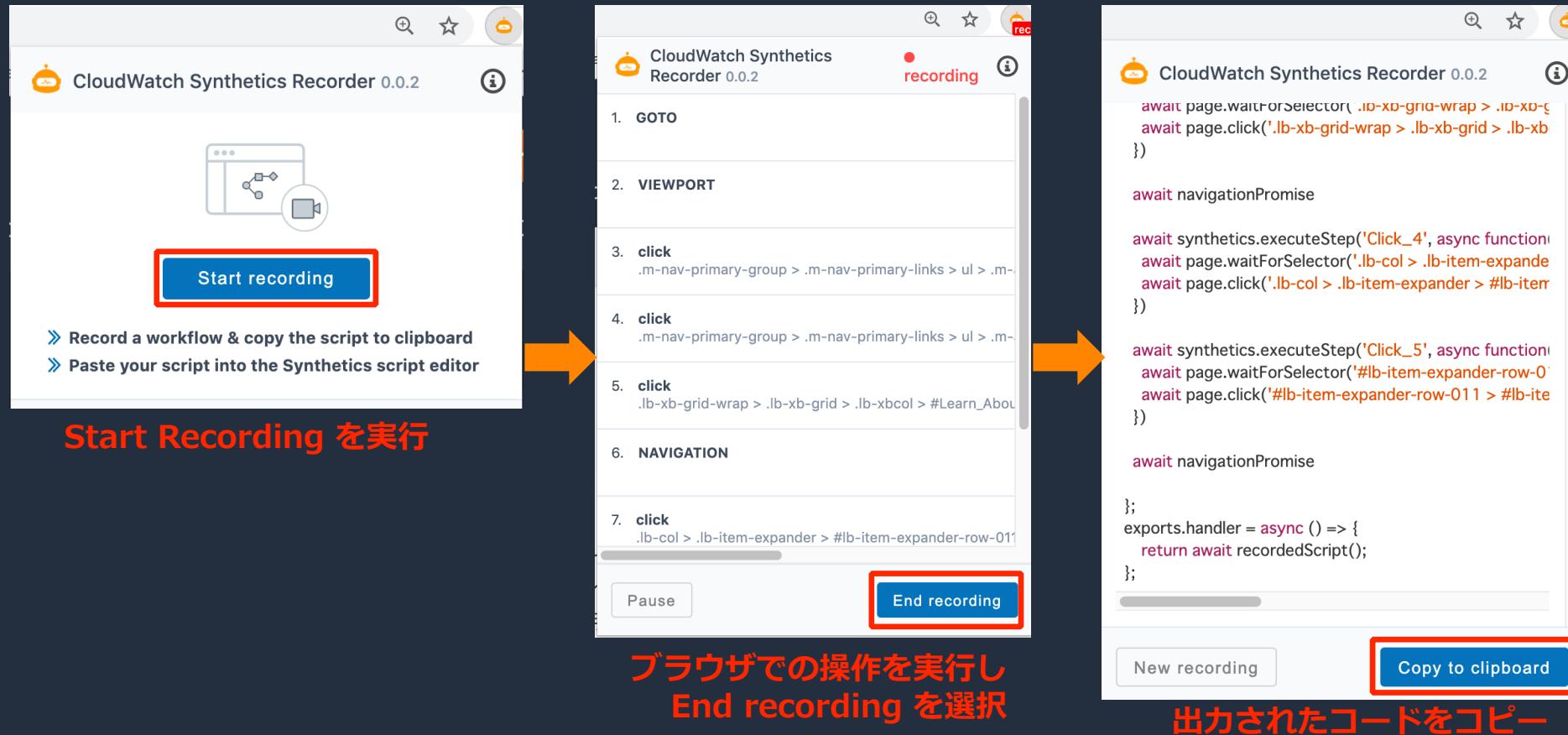


Google Chrome の拡張機能である CloudWatch Synthetics Recorder を以下のリンクより Google Chrome に追加する。

<https://chrome.google.com/webstore/detail/cloudwatch-synthetics-rec/bhdnlmmgiplmbcdmkkdfplenecepfno>

Canary Blueprint - Canary レコーダー

② CloudWatch Synthetics Recorder でユーザ操作を記録



Canary Blueprint - Canary レコーダー

③ 出力したコードをコピーし Canary 作成

設計図

Canary レコーダー AWS Canary Recorder プラグインを使用します。

Canary を記録 情報

AWS Synthetics Canary Recorder Chrome プラグインを使用してワークフローを記録します。

以下にエンドポイントを入力し、新しいタブで URL を開きます

CloudWatch Synthetics Chrome ブラウザプラグインを使用してワークフローを記録します

記録が完了したら、[クリップボードにコピー] をクリックし、以下のスクリプトエディタにスクリプトを貼り付けます。

レコーダーをスタートした URL を入力

http://

URL を新しいタブで開く

記録を開始するエンドポイント。

Canary ビルダー

名前

synthetics-recorddemo

名前は、21 文字までの小文字、数字、ハイphen、またはアンダースコアで構成され、スペースを含めることはできません。

スクリプトエディタ

スクリプトをアップロード 元に戻す エディタをクリア

ランタイムバージョン 情報

syn-nodejs-puppeteer-3.9

この Canary を実行する Synthetics ランタイムバージョンを選択します。

先ほどコピーしたコードを貼り付けて Canary 作成

```
1 var synthetics = require('Synthetics');
2 const log = require('SyntheticsLogger');
3
4 const recordedScript = async function () {
5   let page = await synthetics.getPage();
6
7   const navigationPromise = page.waitForNavigation();
8
9   await synthetics.executeStep('Goto_0', async function() {
10     await page.goto("http://servi-petsi-t462mqtv152v-607366110.us-east-1.elb.amazonaws.com/?select");
11   });
12
13   await page.setViewport({ width: 1920, height: 944 });
14 }
```

Canary 実行結果詳細 (可用性 - ステップ) Canary レコーダー

ステップ | スクリーンショット | ログ | HAR ファイル | トレス

実行されたステップ (5) Canary レコーダーで記録した操作のステップ毎に結果が表示される

失敗したステップのみ < 1 > ⚙️

ステップ	ステップ名	ステータス	説明	送信先 URL	期間	スクリーンショット
1	Goto_0	✓ 成功	OK	http://[REDACTED]	792 ミリ秒	2
2	Click_1	✓ 成功	OK	http://[REDACTED]	61 ミリ秒	2
3	Select_2	✓ 成功	OK	http://[REDACTED]	12 ミリ秒	2
4	Click_3	✓ 成功	OK	http://[REDACTED]	38 ミリ秒	2
5	Click_4	✗ 失敗	TimeoutError: waiting for...	http://[REDACTED]	30001 ミリ秒	2

Canary Blueprint

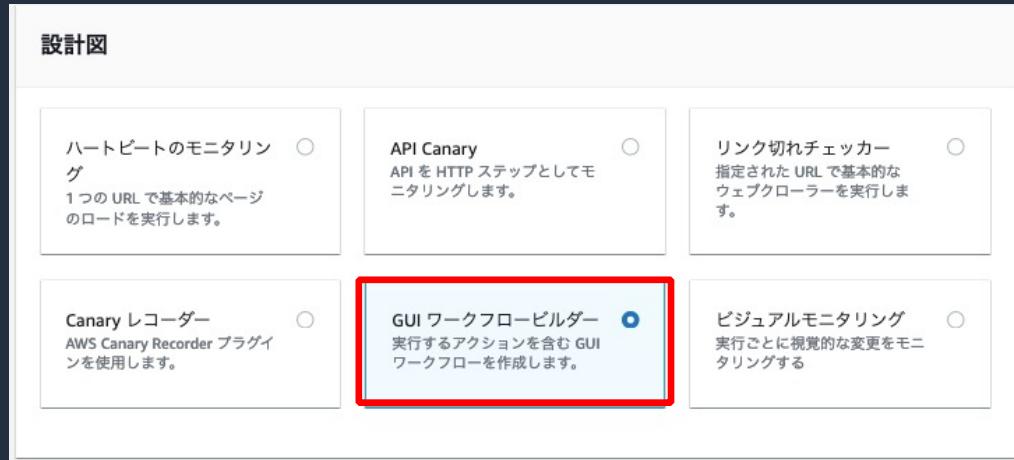
Canary の実体は Node.js/Python で実装される Lambda 関数で、6 種類の Blueprint が用意されており、基本的な用途であればコーディングが不要です。

No.	Blueprint	特徴	Node.js	Python
1	ハートビートのモニタリング	指定した URL にアクセスして、ページのスクリーンショットと HTTP アーカイブファイル (HAR ファイル) を保存	○	○
2	API Canary	REST API に対してリクエストを送信して、応答をテスト	○	○
3	リンク切れチェッカー	テスト対象のURL内のすべてのリンクを収集し、リンク切れがないかテスト	○	-
4	Canary レコーダー	Google Chrome の拡張機能である CloudWatch Synthetics Recorder を利用して、ユーザ操作を記録し、テスト	○	-
5	GUI ワークフロービルダー	Web サイト上のユーザ操作ができるかを GUI ベースで作成してテスト	○	○
6	ビジュアルモニタリング	Web サイトの表示が変化していないかをベースラインと比較し、テスト	○	-

Canary 設計図の使用

https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/CloudWatch_Synthetics_Canaries_Blueprints.html

Canary Blueprint - GUI ワークフロービルダー



The screenshot shows the 'Design' tab of the 'Canary Blueprint' interface. It lists several actions:

- ハートビートのモニタリング
- API Canary
- リンク切れチェッカー
- Canary レコーダー
- GUI ワークフロービルダー (This option is highlighted with a red box)
- ビジュアルモニタリング

The 'GUI ワークフロービルダー' option is selected, indicated by a blue dot next to it.



The screenshot shows the 'Canary Builder' configuration page for a blueprint named 'synthetics-gui-demo'. The configuration includes:

- 名前:** synthetics-gui-demo
- アプリケーションまたはエンドポイント URL 情報:** http://servi-... .elb.amazonaws.com/
- ワークフロービルダー:** Canary が実行するアクションを選択します。

A red box highlights the 'アクション' dropdown menu, which is set to 'クリック' (Click), and the 'セレクター' input field containing '[id='searchpets']'.

- Web ページに対して、ユーザの一連の操作を定義して、意図通りに操作できるかをテストすることができます。
- Web ページ上のアクションを GUI で定義していくことで、ユーザ操作がコードに反映されます。

Canary Blueprint - GUI ワークフロービルダー

定義できるアクション



- 定義できるアクションは、以下5つになります。

No.	アクション	アクション説明
1	クリック	ページ内の要素を指定して、ユーザによる要素のクリックをシミュレーション
2	セレクターを確認	指定した要素が Web ページ上に存在するか検証
3	テキストを確認	指定した文字列が指定した要素に含まれているかを検証
4	テキストを入力	指定したテキストを指定した要素に書き込む
5	ナビゲーションでクリック	指定した要素を選択した後で、ページ全体が読み込まれるまで待つ

- セレクターの定義は選択する言語 (Node.js/Python) で異なります。詳細は下記をご確認ください。
 - GUIワークフロービルダー
https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/CloudWatch_Synthetics_Canaries_Blueprints.html#CloudWatch_Synthetics_Canaries_Blueprint_GUI_Workflow

Canary Blueprint

Canary の実体は Node.js/Python で実装される Lambda 関数で、6 種類の Blueprint が用意されており、基本的な用途であればコーディングが不要です。

No.	Blueprint	特徴	Node.js	Python
1	ハートビートのモニタリング	指定した URL にアクセスして、ページのスクリーンショットと HTTP アーカイブファイル (HAR ファイル) を保存	○	○
2	API Canary	REST API に対してリクエストを送信して、応答をテスト	○	○
3	リンク切れチェッカー	テスト対象のURL内のすべてのリンクを収集し、リンク切れがないかテスト	○	-
4	Canary レコーダー	Google Chrome の拡張機能である CloudWatch Synthetics Recorder を利用して、ユーザ操作を記録し、テスト	○	-
5	GUI ワークフロービルダー	Web サイト上のユーザ操作ができるかを GUI ベースで作成してテスト	○	○
6	ビジュアルモニタリング	Web サイトの表示が変化していないかをベースラインと比較し、テスト	○	-

Canary 設計図の使用

https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/CloudWatch_Synthetics_Canaries_Blueprints.html

Canary Blueprint - ビジュアルモニタリング

設計図

- ハートビートのモニタリング
- API Canary
- リンク切れチェッカー
- Canary レコーダー
- GUI ワークフロービルダー
- ビジュアルモニタリング

① ベースラインのスクリーンショットからエリアを除外したり、視覚的な差異のしきい値を設定したりするなど、
視覚的な Canary を設定する方法についての詳細をご覧ください。 詳細は[こちら](#)

Canary ビルダー

名前
synthetics-visu-demo

名前は、21 文字までの小文字、数字、ハイフン、またはアンダースコアで構成され、スペースを含めることはできません。

アプリケーションまたはエンドポイント URL 情報
.amazonaws.com 削除

エンドポイントを追加

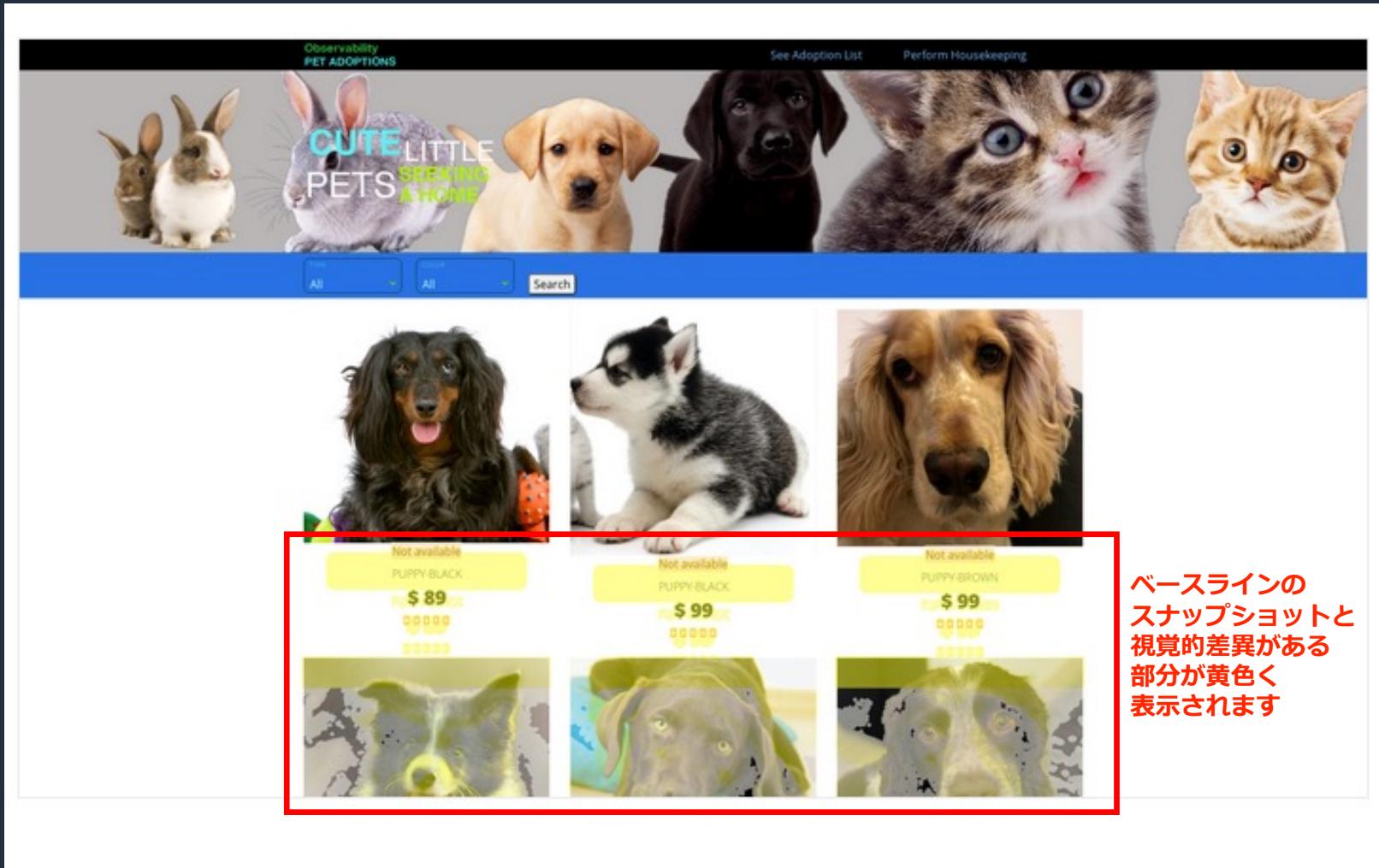
最大でさらに 4 個のエンドポイントを追加できます。スクリプトを変更することで、さらにエンドポイントを追加できます。

しきい値 情報
視覚的な差異が以下より大きい場合 0 % が Canary で失敗

aws Web Services, Inc. or its affiliates.

- 以下流れで、**指定した URL の Web ページの表示に異常な変化がないか**をチェックします。
 - 最初に正常実行した Canary のスクリーンショットをベースラインとして使用
 - その後の Canary 実行で取得したスクリーンショットとベースラインのスクリーンショットを比較して、**定めたしきい値を超えて**いる場合は、**エラー**として検知
- 「syn-nodejs-puppeteer-3.2」以降のランタイムバージョンを使用している場合に利用が可能な機能になります。
※ 「syn-python-selenium」では利用することはできないため、ご注意下さい。

Canary Blueprint - ビジュアルモニタリング 結果の確認



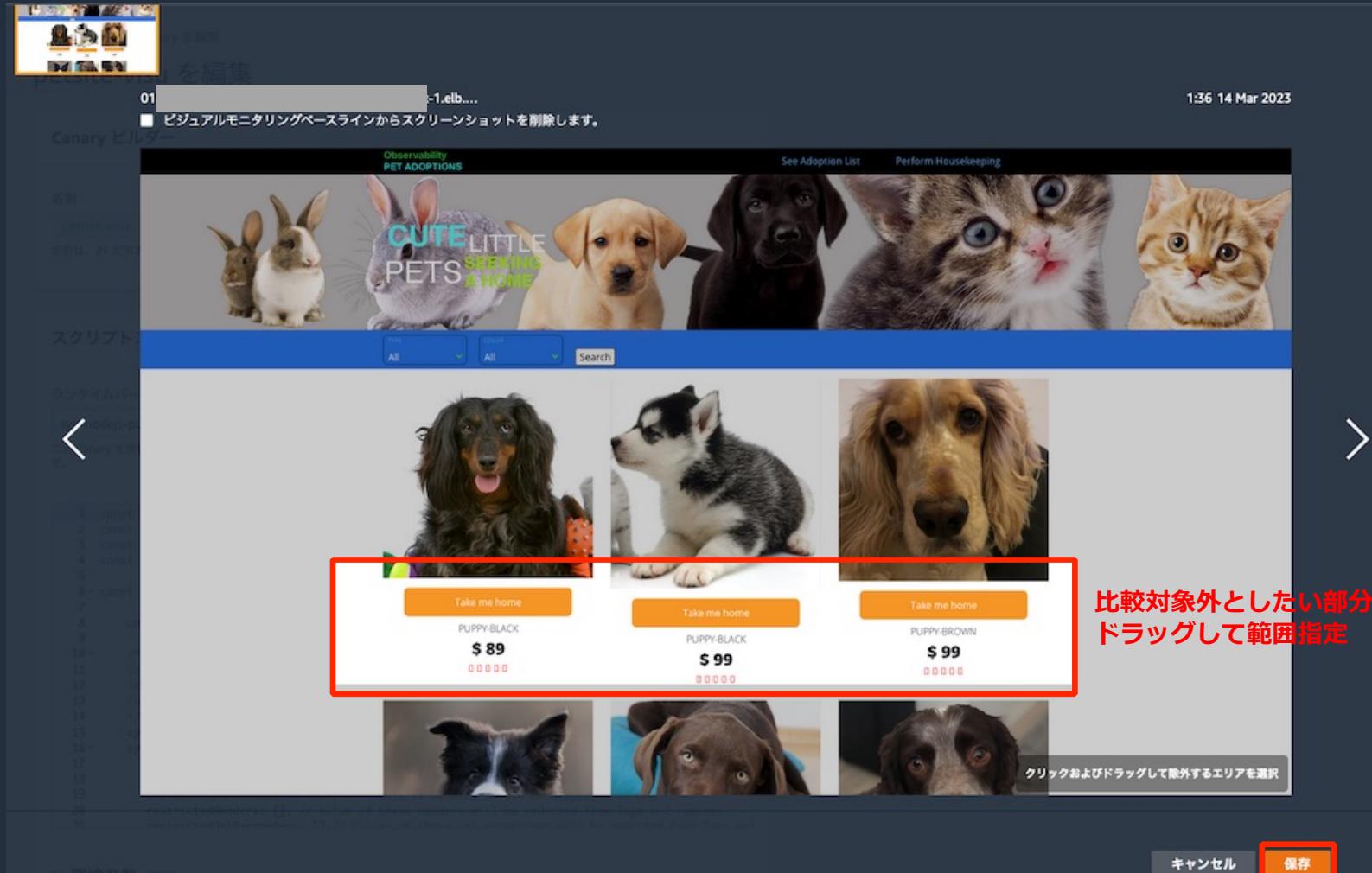
Canary の [可用性] タブの [スクリーンショット] タブを選択することで、視覚的差異が Web ページのどの部分で発生したかが、黄色く表示され、確認できます。

Canary Blueprint - ビジュアルモニタリング ベースライン再設定



- Web ページの変更等でベースラインを再設定したい場合、変更したい Canary の編集画面のビジュアルモニタリングの設定項目から新しくベースラインの設定をすることができます。
- ベースラインの比較対象から、一部を比較対象外としたい場合は、「ベースラインを編集」をクリックして除外する部分を設定することができます。

Canary Blueprint - ビジュアルモニタリング スクリーンショット比較対象領域除外設定



ベースラインのスクリーンショットから、比較対象外としたい箇所をドラッグして、範囲指定することで、指定した領域は比較対象外とすることが可能です。

アジェンダ

Amazon CloudWatch Synthetics とは

Amazon CloudWatch Synthetics Canaryとは

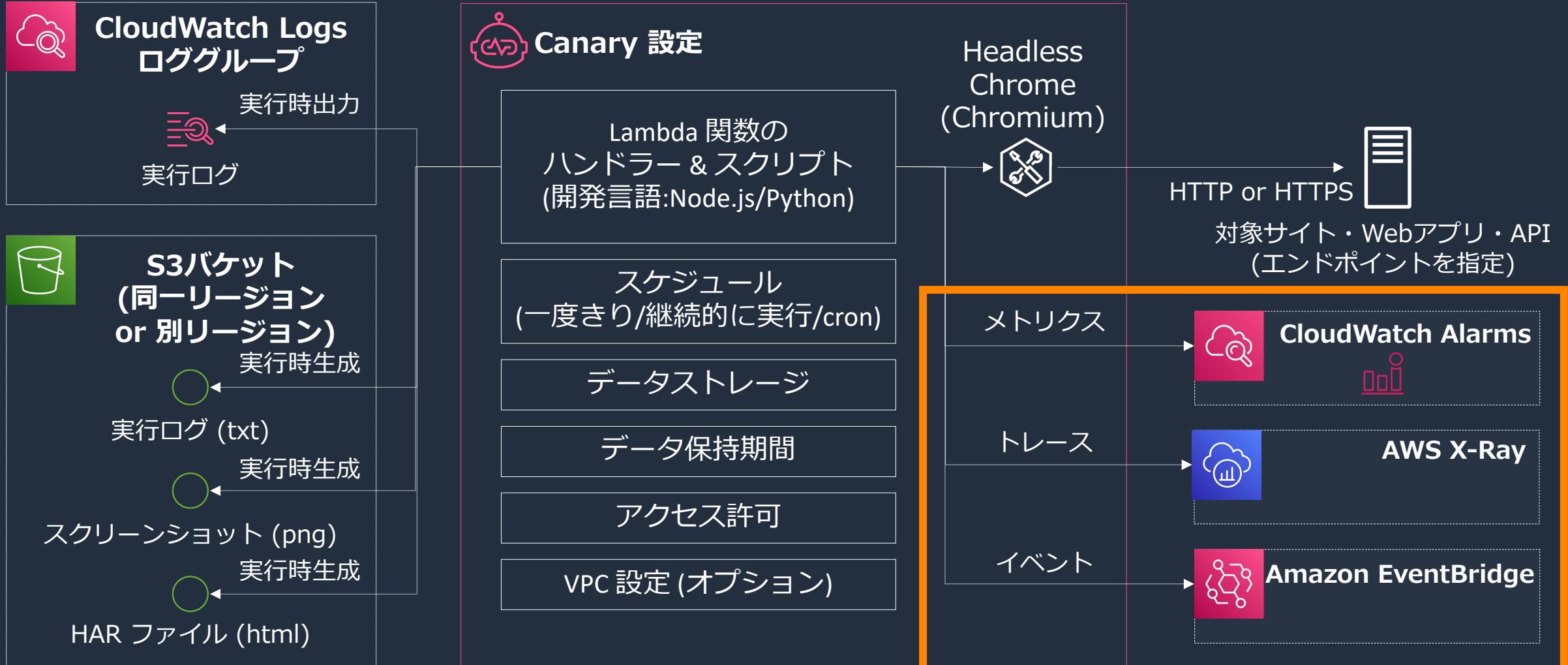
Amazon CloudWatch Synthetics による Synthetics Monitoring

AWS モニタリングサービスとの連携

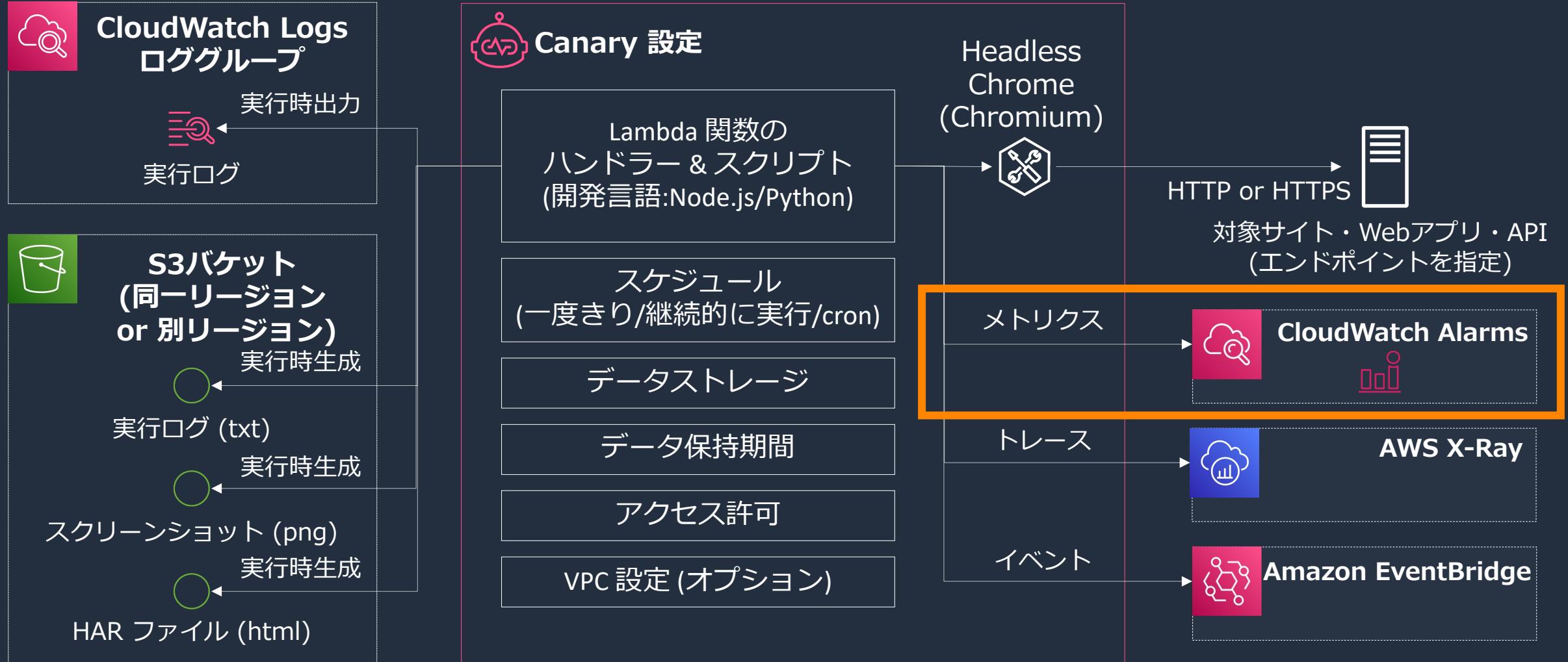
Amazon CloudWatch Synthetics 料金

まとめ

Amazon CloudWatch Synthetics Canary 全体構成イメージ



Amazon CloudWatch Synthetics Canary 全体構成イメージ



Canary モニタリングサービスの連携 – CloudWatch Alarms

The screenshot shows the 'CloudWatch アラーム - オプション' (CloudWatch Alarms - Options) section of the AWS CloudWatch Metrics console. It displays a table of alarms and a configuration panel for setting up notifications.

CloudWatch アラーム - オプション 情報
Synthetics で Canary のアラームを自動的に作成し、後でカスタマイズできます。

メトリクス名	アラームの状態	しきい値	期間	操作
SuccessPer...	より低い	90 %	5 分	削除

新しいアラームを追加

この Canary の通知を設定する
この Canary が定義したレベルに達したときに通知を受け取る場所を選択する

次の SNS トピックに通知を送信します。 [情報](#)
通知を受け取る SNS (Simple Notification Service) トピックを定義する

既存の SNS トピックの選択
 新しいトピックの作成

新しいトピックを作成...
トピック名は一意である必要があります。新しいトピックには Synthetics がプレフィックスとして付けられます。

Synthetics-petsit

SNS トピック名に使用できるのは、英数字、ハイフン (-)、アンダースコアのみです。

通知を受け取る E メールエンドポイント...
E メールアドレスのカンマ区切りのリストを追加します。各アドレスは、上記のトピックへのサブスクリプションとして追加されます。

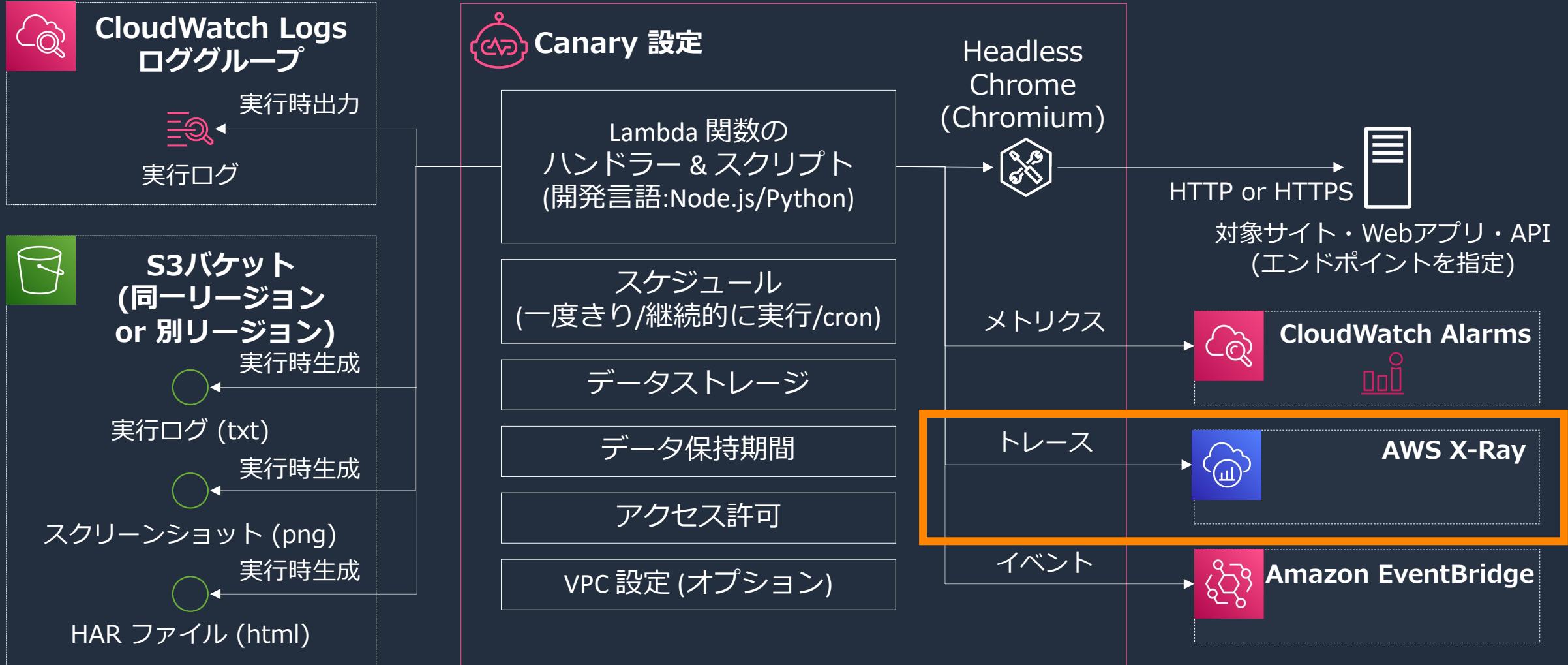
xxxxxx@amazon.co.jp

user1@example.com, user2@example.com

トピックの作成

- Canary 作成時、Canary 設定変更時、CloudWatch アラームの設定画面から Synthetics で取得しているメトリクスに対して、アラームを作成することができます。
- CloudWatch Synthetics 設定画面からは通知先としてSNS トピックを設定することも可能です。通知以外の処理を行いたい場合は、CloudWatch アラームの設定画面から設定をして下さい。
- CloudWatch Synthetics で取得している各種メトリクスについては、下記を参照して下さい。
 - Canaryによって発行されるCloudWatchメトリクス https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/CloudWatch_Synthetics_Canaries_metrics.html

Amazon CloudWatch Synthetics Canary 全体構成イメージ



Canary モニタリングサービスの連携 – AWS X-Ray

▼ アクティブトレース - オプション 情報

AWS X-Ray でアクティブなトレースを有効にして、トラブルシューティングを行い、解決までの平均時間を短縮します。

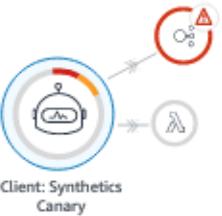
AWS X-Ray でサービスを追跡します。

AWS X-Ray と Synthetics は、分析とアバッテグを行って、進行中の障害の根本原因を見つけ、パフォーマンスのボトルネックや傾向を特定し、レイテンシー率を比較して、API と URL に十分な Canary カバレッジがあるかどうかを確認するのに役立ちます。

[詳細ははこちら](#)

その他の利点

- ✓ AWS X-Ray および CloudWatch ServiceLens サービスマップで Canary を表示します。
- ✓ 各 Canary 実行のトレースとセグメントを表示します。
- ✓ AWS X-Ray 分析を使用して傾向を表示します。


Client: Synthetics Canary

- 「syn-nodejs-2.0」以降のランタイムを使用するCanaryでは、Canary 新規作成時、または、Canary 設定変更時に、アクティブトレース設定画面で、「AWS X-Ray でサービスを追跡します」のチェックボックスにチェックを入れることで、Canary 実行におけるトレース情報を取得することができます。

※ 「syn-python-selenium」を利用する場合は本機能を利用することができないため、ご注意下さい。

Canary 実行結果詳細 (トレース)

ステップ | スクリーンショット | ログ | HAR ファイル | トレース

調査したいトレース ID を選択して、クリックすると該当のトレースマップが表示される

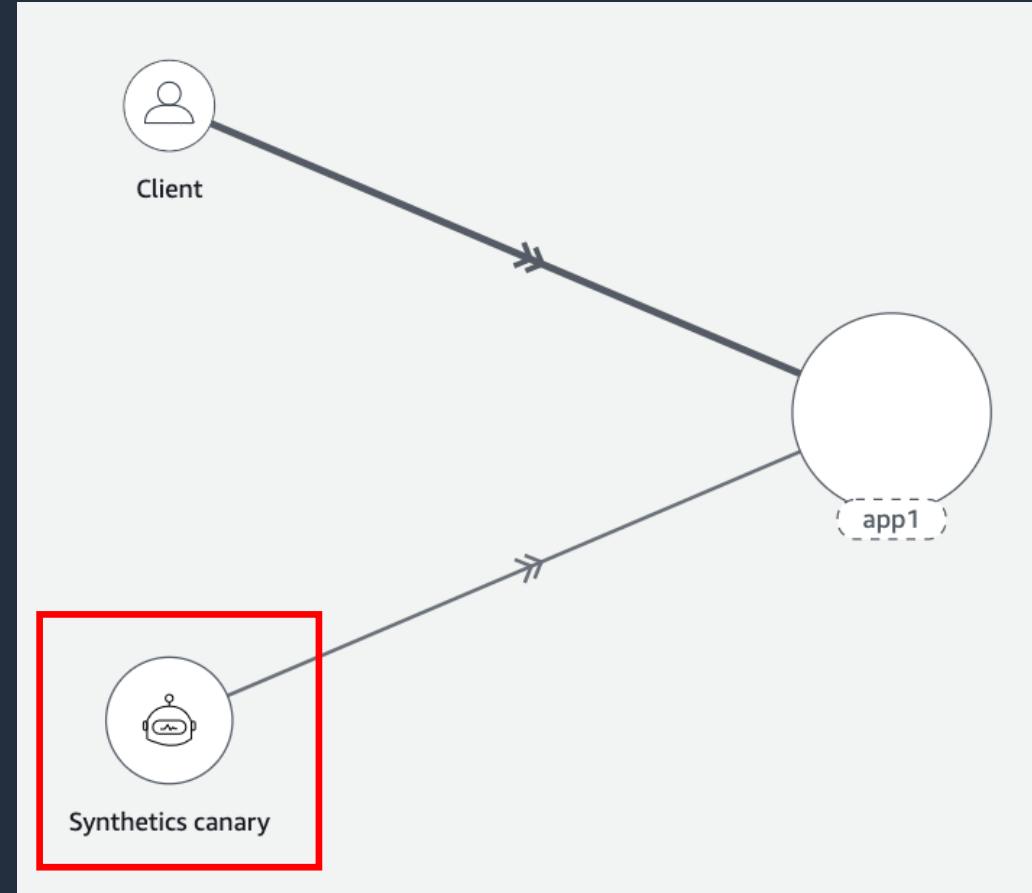
トレース (45) エラーのみを表示 トレースマップに移動

1 2 3 4 5 6 7 8

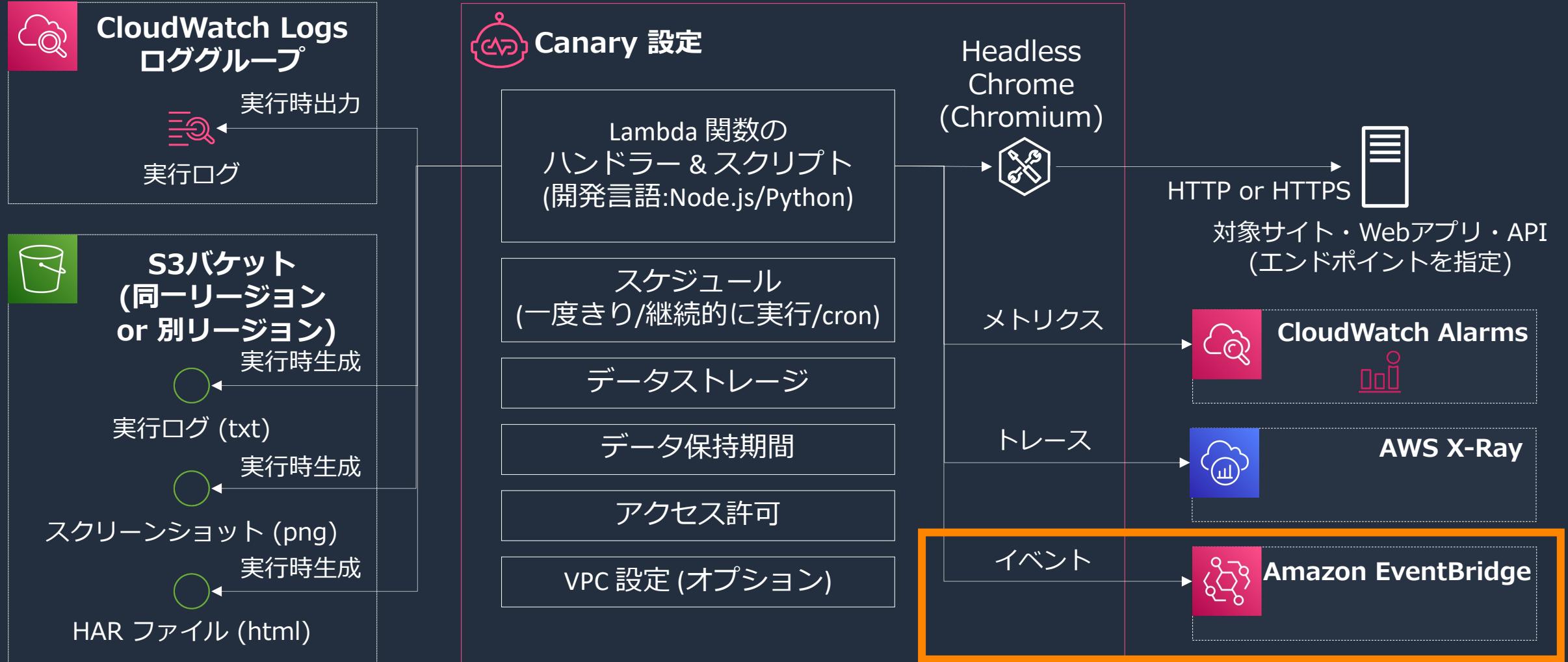
トレース ID	トレースステータス	レスポンスコード	応答時間	URL アドレス	HTTP メソッド
1-6410085b-375a02b4a69f36ef6d72c054	OK	-	0.96秒	-	-
1-6410085b-39287ee394447d9c10f65f2a	OK	200	0.13秒	http://servi-petsi-east-1.elb.amazonaws.com/	GET
1-6410085b-45fd38ae0c23e1c31a17a58	OK	200	0.12秒	http://servi-petsi-east-1.elb.amazonaws.com/css/site.css	GET
1-6410085b-4e113255d272fc138c451625	OK	-	0.90秒	-	-
1-6410085b-596def403747c177ce9cf64c	OK	-	1.12秒	-	-
1-6410085b-5cf9df818aa2210a9093457f	OK	-	0.06秒	-	-

Canary モニタリングサービスの連携 – X-Ray トレースマップ

Canary のトレースは通常のクライアントとは別のアイコンでトレースマップに表示されます。



Amazon CloudWatch Synthetics Canary 全体構成イメージ



Canary モニタリングサービスの連携 – Amazon EventBridge



Event Bridge

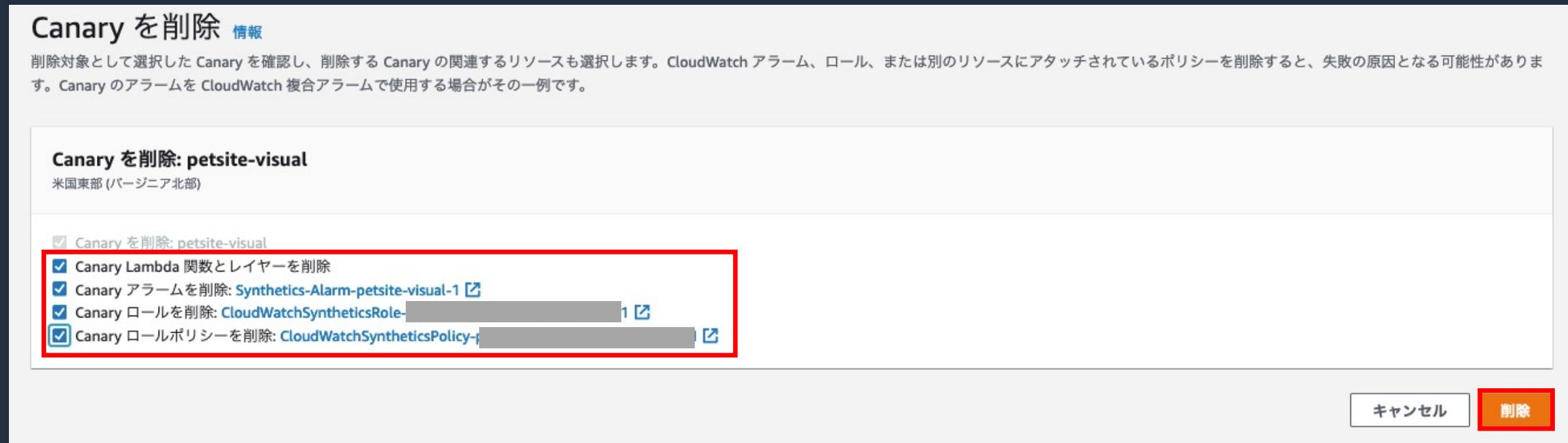
- Canary State Changes
- Canary Testrun Success
- Canary Testrun Failure



CloudWatch Synthetics は Canary の設定を変更したり、Canary の実行が完了した場合にイベントを出力します。Amazon EventBridge のイベントルールを設定することで、特定のイベントパターンをトリガーに通知を行ったり、是正処置を行うことができます。例えば以下のようなことに利用可能です。

- Canary のライフサイクルの追跡 (Canary の設定を変更した際に通知を上げたい)
→ Canary State Changes のイベント発行を機に SNS で通知する
- Canary 実行が失敗した場合の調査 (ワークフローの一部として利用)
→ Canary Testrun Failure のイベント発行を機に ワークフローを動かす

Canary の削除について



- Canary を削除する場合は、Canary を一度停止してから行います。
- Canary を作成時に作成されたリソース (Lambda 関数/IAM ロール・ポリシー /CloudWatch アラーム) も併せて削除することが可能です。
削除したいリソースを選択の上、削除して下さい。

アジェンダ

Amazon CloudWatch Synthetics とは

Amazon CloudWatch Synthetics Canaryとは

Amazon CloudWatch Synthetics による Synthetics Monitoring

AWS モニタリングサービスとの連携

Amazon CloudWatch Synthetics 料金

まとめ

Amazon CloudWatch Synthetics 料金 (2023/3現在)

① CloudWatch Synthetics 利用料

1回の Canary 実行あたり 0.0019 USD (※東京リージョン)

毎月 100 回 までの無料利用枠あり

月あたりに実行される Canary 実行回数に対しての利用料

<https://aws.amazon.com/jp/cloudwatch/pricing/>

② 関連サービス利用料

1. Canary としての Lambda 関数の実行にかかる利用料
2. S3 ストレージ利用料 (結果データやスクリプトの保存に伴う利用料)
3. CloudWatch Alarms, CloudWatch Logs, X-Ray 等の利用料

アジェンダ

Amazon CloudWatch Synthetics とは

Amazon CloudWatch Synthetics Canaryとは

Amazon CloudWatch Synthetics で実現できる外形監視

AWS モニタリングサービスとの連携

料金・注意点

まとめ

まとめ

Amazon CloudWatch Synthetics を利用すれば

- 必要な時に素早く Synthetics Monitoring を実現
- モニタリング動作の振る舞いをプログラムで制御可能
 - シンプルな用途であれば、組み込み提供のテンプレート (Blueprint) を用いるだけでプログラミングは不要
 - カスタマイズにより複雑な操作も表現可能
- AWS サービスとの連携による高い柔軟性
 - S3, EventBridge, X-Ray などの AWS サービスと連携

本資料に関するお問い合わせ・ご感想

技術的な内容に関しましては、有料のAWSサポート窓口へ
お問い合わせください

<https://aws.amazon.com/jp/premiumsupport/>

料金面でのお問い合わせに関しましては、カスタマーサポート窓口へ
お問い合わせください（マネジメントコンソールへのログインが必要です）

<https://console.aws.amazon.com/support/home#/case/create?issueType=customer-service>

具体的な案件に対する構成相談は、後述する個別相談会をご活用ください



ご感想はTwitterへ！ハッシュタグは以下をご利用ください
#awsblackbelt



その他コンテンツのご紹介

ウェビナーなど、AWSのイベントスケジュールをご参照いただけます

<https://aws.amazon.com/jp/events/>

ハンズオンコンテンツ

<https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-hands-on/>

AWS 個別相談会

AWSのソリューションアーキテクトと直接会話いただけます

<https://pages.awscloud.com/JAPAN-event-SP-Weekly-Sales-Consulting-Seminar-2021-reg-event.html>



Thank you!