

[検証環境の準備]

1. CloudFormation の画面をひらきます



2. [スタックの作成]をおします

スタックの作成

前提条件 - テンプレートの準備

テンプレートの準備
各スタックはテンプレートに基づきます。テンプレートとは、スタックに含む AWS リソースに関する設定情報を含む JSON または YAML ファイルです。

☒ テンプレートの準備完了 ☐ サンプルテンプレートを使用 ☐ デザイナーでテンプレートを作成

テンプレートの指定

テンプレートは、スタックのリソースおよびプロパティを表す JSON または YAML ファイルです。

テンプレートソース
テンプレートを選択すると、保存先となる Amazon S3 URL が生成されます。

☐ Amazon S3 URL ☒ テンプレートファイルのアップロード

3. 用意されている[sample.yaml]をアップロードし、[次へ]をおします
4. 適当な名前を入力し、[次へ]をおします。次の画面ではそのまま再度[次へ]をおします

スタックの詳細を指定

スタックの名前

スタックの名前

devopsguru

スタック名では、大文字および小文字 (A-Z~a-z)、数字 (0-9)、ダッシュ (-) を使用することができます。

5. IAM 関連の注意喚起にチェックをつけて[スタックの作成]をおします

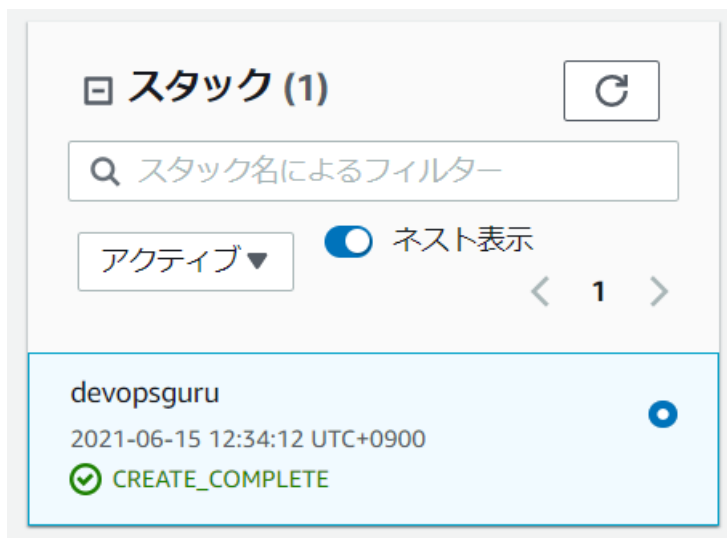
① The following resource(s) require capabilities: [AWS::IAM::Role]

このテンプレートには、ご利用の AWS アカウントに変更を加えるエンティティにアクセスを与える可能性を持つ Identity and Access Management (IAM) リソースが含まれています。これらのリソースを個別に作成し、それぞれに最小限必要な権限を与えるかどうか確認してください。 [詳細はこちら](#)

☒ AWS CloudFormation によって IAM リソースが作成される場合があることを承認します。

キャンセル 戻る 変更セットの作成 **スタックの作成**

- 5 分程度まちます。左のペインが以下のように、緑文字で[CREATE_COMPLETE]と表示されれば作成完了です。



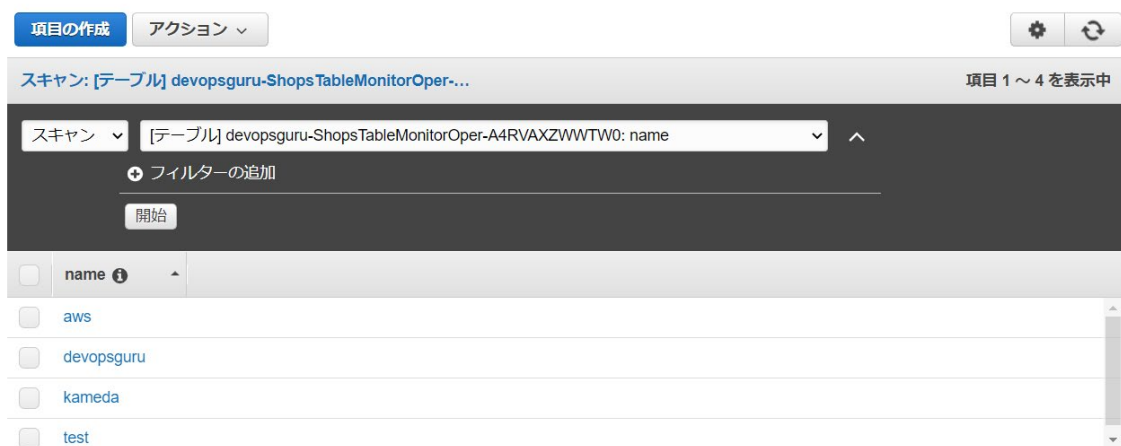
- [出力]タブから[ListRestApiEndpointMonitorOper]の値をコピーしてブラウザでアクセスします。[] と表示されれば OK です。
- DynamoDB の画面に移動します
- 左ペインの[テーブル]をクリックするとテーブルが 1 個作成されていることがわかります



- テーブルをクリックし、[項目]のタブを開きます
- [項目の作成]をおして、何かデータを書き込み[保存]をおします



12. 同じ手順を 2～3 回、異なる値を書き込むために行います。値はなんでも OK です
以下のような画面になりますので、再度ブラウザから同じ URL にアクセスして値が
出力されるかを確認しておきます。



13. 続いて[キャパシティ]のタブをクリックします

devopsguru-ShopsTableMonitorOper-A4RVAXZWWTW0 [閉じる](#)

概要 項目 **メトリックス** アラーム **キャパシティー** インデックス グローバルテーブル バックアップ [さらに](#) v

▶ スケーリングアクティビティ


読み込み/書き込みキャパシティーモード

容量計画を適用せずに、実行した読み取り/書き込みに対してのみ料金を支払う場合は、オンデマンドを選択します。アプリケーションのスループット要件を確実に見積もることができる場合は、スループットにかかる費用を節約するため、プロビジョンドを選択します。詳細については、[DynamoDB 料金表ページ](#) および [DynamoDB デベロッパーガイド](#) を参照してください。

読み込み/書き込みキャパシティーモードは後で変更できます。

☒ プロビジョンド (無料利用枠の対象)
☐ オンデマンド

オンデマンドモードへの最後の変更: 読み込み/書き込みキャパシティーモードの変更は行われていません。
 オンデマンドモードへの次に利用可能な変更: いつでも、オンデマンドモードに更新できます。

プロビジョニングされたキャパシティー 

14. 読み込みキャパシティーユニット、を 1 に設定して[保存]をおします。後ほど意図的にプロビジョニングされたキャパシティーを超える読み込みを発生させ DynamoDB でエラーを発生させるための手順です。

プロビジョニングされたキャパシティー 

	読み込みキャパシティーユニット	書き込みキャパシティーユニット
テーブル	5	5
推定コスト	\$2.91/月 (キャパシティー計算ツール)	

Auto Scaling

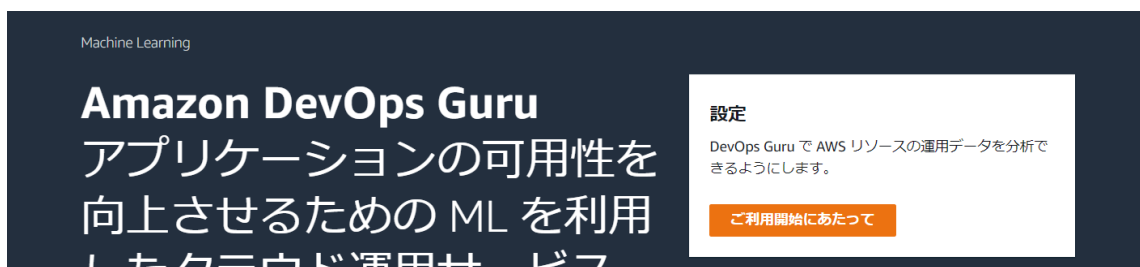
☐ 読み込みキャパシティー ☐ 書き込みキャパシティー

保存 キャンセル

[DevOps Guru の有効化]

ここから DevOps Guru を有効化し監視をかいしさせます

15. DevOps Guru の画面に移動します



16. [ご利用開始にあたって]をおします
17. [Amazon DevOps Guru 分析カバレッジ]で[後で選択]を選び、[有効化]をおします

Amazon DevOps Guru 分析カバレッジ

DevOps Guru は、選択した内容に基づいて AWS リソースの運用データを分析します。アクティブなリソースごとに、分析された AWS リソース時間数に対して料金が発生します。リソースは、1 時間以内にメトリクス、イベント、またはログエントリを生成する場合にのみアクティブになります。詳細については、[料金ページ](#) をご覧ください。

カバレッジ境界を指定して、分析する AWS リソースを選択します。

☐ このリージョンの現在の AWS アカウントのすべての AWS リソースを分析
☒ 後で選択

AWS CloudFormation スタックをカバレッジの境界として使用して、分析する特定の AWS リソースを指定できます。

18. 画面左ペインから[設定]をクリックします

Amazon DevOps Guru > 設定

設定

DevOps Guru 分析カバレッジ

DevOps Guru カバレッジは、AWS アカウントとリージョンで分析されるリソースを決定します。

管理

AWS リソースカバレッジの境界

このリージョンのリソースを分析しない

19. [管理]をおします

AWS リソースの選択

DevOps Guru リソースカバレッジを選択

DevOps Guru カバレッジは、AWS アカウントとリージョンで分析されるリソースを決定します。

☐ このリージョンの現在の AWS アカウントのすべての AWS リソースを分析
☐ このリージョンの指定された CloudFormation スタックのすべての AWS リソースを分析
☒ このリージョンのリソースを分析しない

アカウントのリソースは分析されません

Amazon DevOps Guru で AWS リソースを分析しないように設定した場合は、インサイトを生成できません。これを変更するには、後でナビゲーションペインの [設定] を選択し、Amazon DevOps Guru で分析する AWS リソースを指定します。

20. [このリージョンの指定された CloudFormation スタックのすべての AWS リソースを分析]を選び、今日作成した CloudFormation スタックを選び、[保存]をおします

CloudFormation スタック (1/2)

最大 500 個の CloudFormation スタックを選択できます。

CloudFormation スタックを検索

スタック名	説明	ステータス
<input type="checkbox"/> aws-cloud9-devopsguru-a9b09bfa5b7d4478b14635258f800d27	-	Not enabled
<input checked="" type="checkbox"/> devopsguru	Create a typical serverless infrastructure, comprising of API-Gateway, Lambda and DynamoDB services, exposing create and list API's.	Not enabled

キャンセル 保存

21. ダイアログがでてくるので[確認]をおします。以下のようにスタックが DevOps Guru の監視対象となりました。

DevOps Guru 分析カバレッジ

管理

DevOps Guru カバレッジは、AWS アカウントとリージョンで分析されるリソースを決定します。

AWS リソースカバレッジの境界
このリージョンの指定された CloudFormation スタックのすべての AWS リソースを分析します。

CloudFormation スタック (1)

DevOps Guru は現在、これらの CloudFormation スタックで指定された AWS リソースを分析しています。

< 1 > ⚙️

スタック名	説明
devopsguru	Create a typical serverless infrastructure, comprising of API-Gateway, Lambda and DynamoDB services, exposing create and list API's.

[ベースラインの構築]

DevOps Guru は機械学習により異常を検出する仕組みとなっており、監視が有効化されてから、ベースライン（正常状態のメトリクス）の構築まで2時間待ちます。
つぎの Step は必ず2時間待ってから行ってください。ハンズオンイベントの場合は講師がデモをしますのでそちらをご覧ください

- Cloud9 に画面に遷移します。本ハンズオンでは異なるリージョンでも特別な設定不要で動作しますので、すでに Cloud9 環境をお持ちの方はそちらを使ってください

AWS Cloud9 > Your environments

Your environments (1)

Open IDE

View details

Edit

Delete

Create environment

< 1 > ⚙️

- [Create environment]をおします
- 適当な名前をつけて、[Next Step]をおします

Environment name and description

Name
The name needs to be unique per user. You can update it at any time in your environment settings.

Name

Limit: 60 characters

Description - *Optional*
This will appear on your environment's card in your dashboard. You can update it at any time in your environment settings.

Write a short description for your environment

Limit: 200 characters

Cancel

Next step

25. 次の画面では、全てデフォルトで[Next Step]をおします。作業中のリージョンにデフォルト VPC が存在しない場合は、[Network Settings]から VPC/PublicSubnet を指定してください

▼ Network settings (advanced)

Network (VPC)
Launch your EC2 instance into an existing Amazon Virtual Private Cloud (VPC) or create a new one. To allow the AWS Cloud9 environment to connect to its EC2 instance, attach an internet gateway (IGW) to your new VPC.

vpc-7653bb1f (default) ▼

↻

Create new VPC

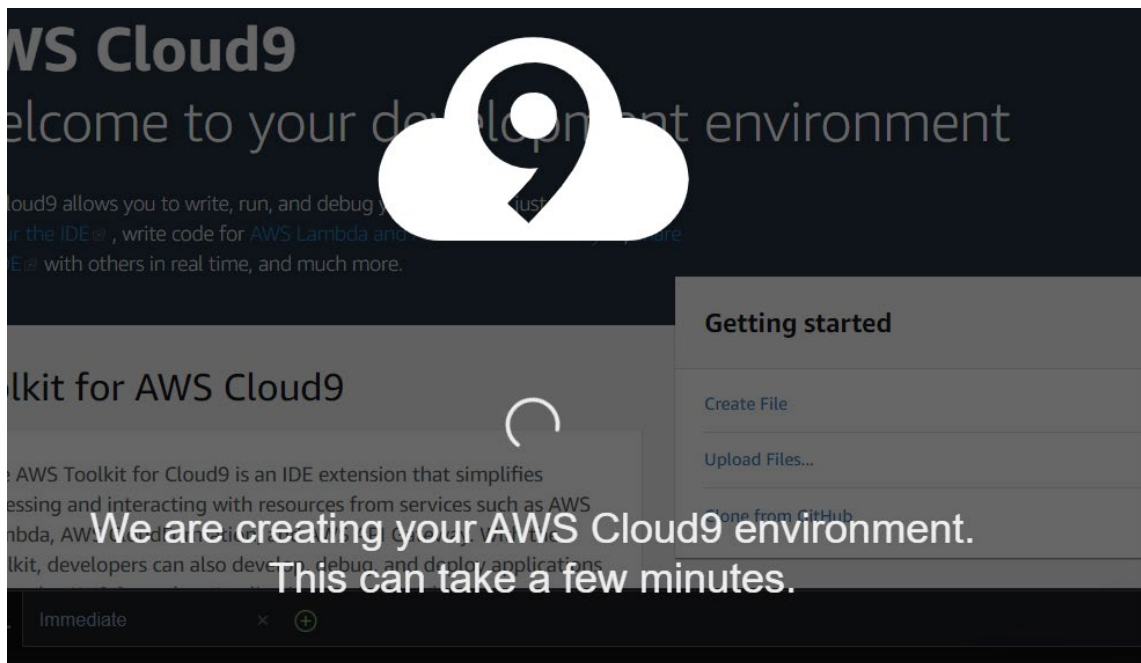
Subnet
Select a public subnet in which the EC2 instance is created. (For a private subnet, you must create an environment that connects to its instance via Systems Manager.)

No preference (default subnet in any Availability Zone) ▼

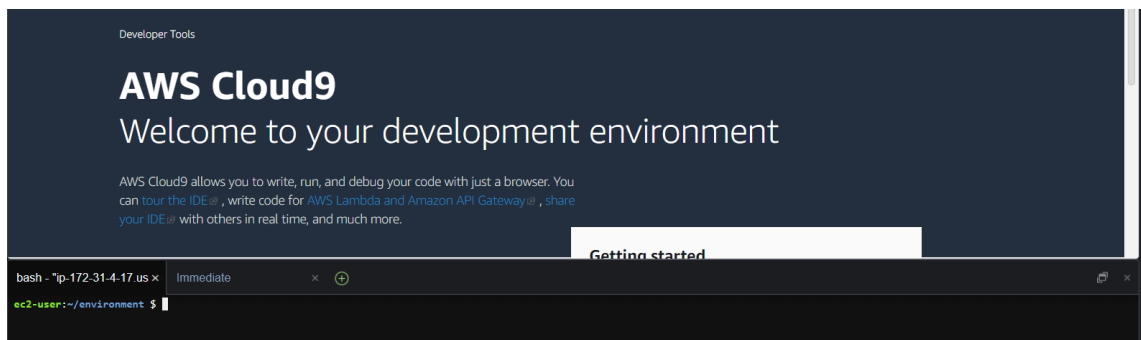
↻

Create new subnet

26. [Create environment]をおします
27. 起動の間数分間まちます



28. 起動するとターミナルがつかえるようになります



29. File → New File で新しいファイルを開き、[sendAPIRequest.py]の中身をコピーします
- <https://<API-ID>.execute-api.<REGION>.amazonaws.com/prod/>
を先程コピーした URL（ブラウザでアクセスした URL）に置換します。
忘れた方は CloudFormation の出力タブから確認できます
30. test.py でファイルを保存します。（タブの右上の丸い部分をクリック）
31. 画面下部のターミナルで以下を実行します
- ```
pip install requests
python test.py
```



```

 Iterating sending requests...+++++
<Response [200]> [{"name":"aws"}, {"name":"kameda"}, {"name":"devopsguru"}, {"name":"test"}] 200

 Iterating sending requests...+++++
<Response [200]> [{"name":"aws"}, {"name":"kameda"}, {"name":"devopsguru"}, {"name":"test"}] 200

 Iterating sending requests...+++++
<Response [200]> [{"name":"aws"}, {"name":"kameda"}, {"name":"devopsguru"}, {"name":"test"}] 200

 Iterating sending requests...+++++
<Response [200]> [{"name":"aws"}, {"name":"kameda"}, {"name":"devopsguru"}, {"name":"test"}] 200

```

DynamoDB へのデータ読み込みが行われています。このスクリプトでは停止するまで読み込みが行われ続けます

32. より明確にエラーを発生させるため、Terminal をもう 5 つ開いて同ジスクリプトを 6 重で同時実行させてください
33. 10 分程度まつと DevOpsGuru が異常を検知しはじめます

# インサイト

事後的

予測的

## 事後的インサイト (1) 情報

事後的インサイトにより、今すぐアプリケーションのパフォーマンスを向上させるための推奨事項を知ることができます。

Q

ステータス、重大度、または影響を受けるリソースでイ...

12h

1d

1w

1M

カスタム

🗲

<

1

>

⚙

| 名前                                                 | ステータス  | 重要度 | 影響を受けるリソース | 作成時刻                   |
|----------------------------------------------------|--------|-----|------------|------------------------|
| Lambda Errors <small>スタック内の devops guru 異常</small> | 🟢 クローズ | 高   | 2          | Jun 15, 2021 02:16 UTC |

インサイトには2種類のインサイトが存在しています

## 事後的インサイト：実際に発生した問題

(5xx エラーなど。ベースライン構築期間に 5xx エラーが定期的に発生する場合、それは正常とみなされます)

予測的インサイト：これから発生が予測される問題

(ストレージのキャパシティ、ユーザーリクエスト増等)

おつかれさまでした！

以下を削除してください

CFn スタック 2 つ：Serverless 環境、Cloud9 環境

DevOpsGuru の[設定]→[DevOps Guru 分析カバレッジ]から

[このリージョンのリソースを分析しない]を選択