aws

開発者ガイド

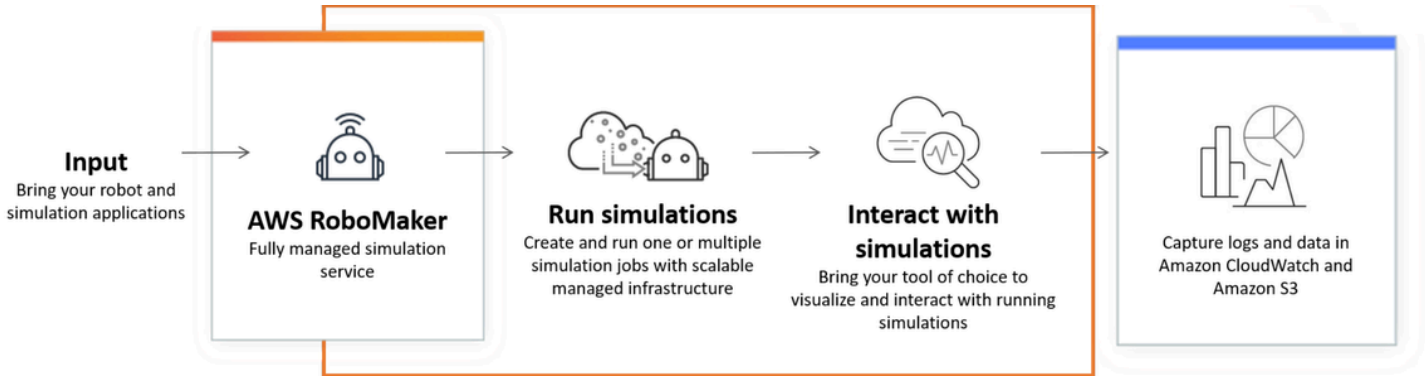
AWS RoboMaker

AWS RoboMaker: 開発者ガイド

Table of Contents

AWS RoboMaker とは

AWS RoboMaker は、ロボティクス開発者がインフラストラクチャを管理せずにシミュレーションを実行、スケーリング、自動化するために使用するクラウドベースのシミュレーションサービスです。AWS RoboMaker を使用すると、ロボット開発者はシミュレーションワークロードを費用対効果の高い方法でスケーリングおよび自動化したり、1 回の API 呼び出しで大規模かつ並列のシミュレーションを実行したり、ユーザー定義のランダム化された 3D 仮想環境を作成したりすることができます。シミュレーションサービスを使用すると、アプリケーションのテストを加速し、定義したテンプレートから数百の新しいワールドを作成できます。



AWS RoboMaker は、継続的インテグレーションと継続的デリバリー (CI/CD) パイプライン内での自動テスト、大量の反復試行による強化モデルのトレーニング、複数の同時シミュレーションをフリート管理ソフトウェアに接続してのテストを行うことができます。AWS の機械学習、監視、分析のサービスと組み合わせると、ロボットはデータのストリーミング、ナビゲート、通信、理解、学習を行うことができます。

[AWS RoboMaker リソース] ページには、シミュレーション教育リソース、シミュレーションワールドアセット、サンプルアプリケーション、ワークショップとチュートリアルのライブラリ、ハードウェア開発キットへのリンクが含まれています。

AWS RoboMaker の機能

AWS RoboMaker には、以下の特徴があります。

- AWS RoboMaker によるシミュレーション は、インフラストラクチャのプロビジョニングや管理を行わずにシミュレーションジョブを実行できる、フルマネージド型のシミュレーションサービスです。このサービスは、大規模な並列シミュレーションをサポートし、テストするシナリオの複雑度に基づいて自動的にスケーリングします。AWS RoboMaker シミュレーションは、ROS、カス

タムロボットアプリケーション、Gazebo、Unity、Unreal、NVIDIA Isaac ベースのシミュレーションなど、任意のロボットソフトウェアとシミュレータの実行に使用できます。

- Simulation WorldForge でのワールドの作成 は、投資の設計やワールド生成インフラの管理を行うことなく、現実世界の条件を模倣する、事前に定義、ランダム化された数百ものシミュレーションワールドを自動作成することができます。現在、Simulation WorldForgeは、設定可能な間取りや家具を備えた、屋内家庭環境向けのワールドを提供しています。

AWS RoboMaker の料金

他の AWS 製品と同様、AWS RoboMaker を使用するための契約や最低契約金は必要ありません。AWS RoboMaker の使用料金に関しては、「AWS RoboMaker の料金」を参照してください。

AWS RoboMaker を開始してサービスの詳細を知るには、AWS RoboMaker の開始方法 に進んでください。

AWS RoboMaker の開始方法

AWS RoboMaker はクラウドでロボットシミュレーションを実行します。はじめに、IAM ロールを持つ AWS アカウントを作成して、シミュレートされたロボットと環境をコンソールに表示できるようにします。次に、環境とロボットアプリケーションの両方に使用するコンテナを構築して、シミュレーションジョブを実行します。次に、シミュレーションジョブからログとデータを取得します。

トピック

- AWS RoboMaker の概念
- セットアップ AWS RoboMaker
- 最初のシミュレーションの実行

AWS RoboMaker の概念

このセクションでは、AWS RoboMaker を効果的に使用するために理解しておく必要のある重要な概念と用語について説明します。詳細については、「AWS RoboMaker に関するよくある質問」を参照してください。

概念

- コンテナ
- シミュレーションジョブ
- Simulation WorldForge
- 環境
- アプリケーション
- アプリケーションの設定

コンテナ

コンテナイメージは Amazon ECR に保存されます。コンテナは、サービスによって実行されるときにイメージから作成されます。一般的なシミュレーションではロボットオペレーティングシステム (ROS) を使用し、1 つのコンテナは Gazebo の環境をシミュレートし、もう 1 つのコンテナはロボットをシミュレートします。詳細については、Amazon ECR ユーザーガイドの「Amazon ECR とは」を参照してください。

シミュレーションジョブ

1 つのシミュレーションジョブで 1 つまたは 2 つのアプリケーションが実行されます。一般的なシミュレーションジョブには、ロボットアプリケーション (環境データに応答するカスタムロジック) と環境 (ロボットが活動する世界のモデル) を組み合わせたものが含まれます。シミュレーションジョブは結果とメトリクスを提供します。詳細については、「AWS RoboMaker によるシミュレーション」を参照してください。

Simulation WorldForge

Simulation WorldForge を使用すると、定義したテンプレートからシミュレーションワールドをより簡単かつ迅速に生成することができます。さらに、Simulation WorldForge は、ドメインのランダム化を含む多数のシミュレーションワールドが必要となるシミュレーションワークロードを管理するのに役立ちます。詳細については、「Simulation WorldForge でのワールドの作成」を参照してください。

環境

アプリケーションで、環境の設定と環境内で実行するツールを指定します。環境内で実行されるツールは、同じファイルシステム、環境変数、ネットワーキングを共有します。環境内で実行されるアプリケーションとツールには、環境内のファイルへの変更が反映され、環境は使用可能なツールを提供する必要があります。環境にコンテナイメージを提供する必要があります。詳細については、「イメージを使用した AWS RoboMaker アプリケーションの開発」を参照してください。

アプリケーション

シミュレーションジョブを作成する前に、AWS RoboMaker でロボットアプリケーションまたはシミュレーションアプリケーションを作成する必要があります。ロボットアプリケーションには、ナビゲーションと認識のためのロボットコードが含まれています。シミュレーションアプリケーションには、環境をシミュレートするのに必要なすべてのアセットとロジックが含まれています。AWS RoboMaker では、ロボットアプリケーションとシミュレーションアプリケーションの複数のバージョンの作成をサポートしています。詳細については、「バージョニングアプリケーション」を参照してください。

アプリケーションは次の 2 つの (主要な) コンポーネントで構成されています。

- コンテナは、コードとその依存関係すべてをパッケージ化するソフトウェアのスタンダード単位で、アプリケーションが 1 つのコンピューティング環境から別のコンピューティング環境に迅速かつ確実に実行します。

- ソフトウェアスイートは、バンドルの内容を抽出、調達、検証、実行できる環境のことです。現在、サポートされているソフトウェアスイートは一般的（ロボットアプリケーション用）とシミュレーションランタイム（シミュレーションアプリケーション用）です。

アプリケーションの設定

CreateSimulationJob でシミュレーションやロボットアプリケーションを提供する場合、実際には RobotApplicationConfig と SimulationApplicationConfig を指定します。つまり、実際のアプリケーションの ARN とバージョン、さらには次の起動構成、アップロード構成、ツールを指定するということです。

- LaunchConfig：環境内でアプリケーションコードをどのように実行したいかをシミュレーションサービスに伝えます。
- UploadConfiguration：アプリケーションごとに最大 10 個のアップロード構成を渡すことができます。AWS RoboMaker はアップロード構成パスに記載されたファイルを出力バケットへアップロードします。
- Tool：アプリケーションコンテナで実行するカスタマイズプロセスのリストです。

詳細については、「AWS RoboMaker によるシミュレーション」を参照してください。

セットアップ AWS RoboMaker

セットアップするには AWS RoboMaker、 AWS まずアカウントと IAM 管理ユーザーを作成する必要があります。

にサインアップしてください AWS アカウント

をお持ちでない場合は AWS アカウント、次の手順を実行して作成してください。

にサインアップするには AWS アカウント

1. https://portal.aws.amazon.com/billing/signup を開きます。
2. オンラインの手順に従います。

   サインアップ手順の一環として、通話呼び出しを受け取り、電話キーパッドで検証コードを入力するように求められます。

にサインアップすると AWS アカウント、AWS アカウントのルートユーザーが作成されます。ルートユーザーには、アカウントのすべての AWS のサービス とリソースへのアクセス権があります。セキュリティ上のベストプラクティスとして、ユーザーに管理アクセスを割り当て、root ユーザーアクセスを必要とするタスクを実行するときは root ユーザーのみを使用してください。

AWS サインアッププロセスが完了すると、確認メールが送信されます。https://aws.amazon.com/ の [マイアカウント] を選んで、いつでもアカウントの現在のアクティビティを表示し、アカウントを管理できます。

管理者権限を持つユーザーを作成します。

にサインアップしたら AWS アカウント、日常のタスクに root ユーザーを使用しないように AWS IAM Identity Center、管理ユーザーを保護し、有効にしてから作成してください。 AWS アカウントのルートユーザー

セキュリティを確保してください。 AWS アカウントのルートユーザー

1. [Root user] を選択し、 AWS アカウント メールアドレスを入力して、AWS Management Consoleアカウント所有者としてにログインします。次のページでパスワードを入力します。

   ルートユーザーを使用してサインインする方法については、AWS サインイン ユーザーガイドの「ルートユーザーとしてサインインする」を参照してください。
2. ルートユーザーの多要素認証 (MFA) を有効にします。

   手順については、IAM ユーザーガイドの「AWS アカウント root ユーザー (コンソール) の仮想 MFA デバイスを有効にする」を参照してください。

管理者権限を持つユーザーを作成します。

1. IAM アイデンティティセンターを有効にします。

   手順については、「AWS IAM Identity Center ユーザーガイド」の「AWS IAM Identity Centerの有効化」を参照してください。
2. IAM Identity Center で、ユーザーに管理アクセスを付与します。

IAM アイデンティティセンターディレクトリ をアイデンティティソースとして使用するチュートリアルについては、『ユーザーガイド』の「 IAM アイデンティティセンターディレクトリデフォルトでのユーザーアクセスの設定」を参照してください。AWS IAM Identity Center

管理者権限を持つユーザーとしてサインインします。

- IAM アイデンティティセンターのユーザーとしてサインインするには、IAM アイデンティティセンターのユーザーの作成時に E メールアドレスに送信されたサインイン URL を使用します。

  IAM Identity Center ユーザーを使用してサインインする方法については、AWS サインイン ユーザーガイドの「 AWS アクセスポータルへのサインイン」を参照してください。

追加のユーザーにアクセス権を割り当てます。

1. IAM Identity Center で、最小権限の権限を適用するというベストプラクティスに従った権限セットを作成します。

   手順については、『ユーザーガイド』の「権限セットの作成」を参照してください。AWS IAM Identity Center
2. ユーザーをグループに割り当て、そのグループにシングルサインオンアクセスを割り当てます。

   手順については、『AWS IAM Identity Center ユーザーガイド』の「グループの追加」を参照してください。

最初のシミュレーションの実行

このガイドの以下のセクションでは、最初のシミュレーションを実行する方法を説明します。順番に実行してください。

最初のシミュレーションを実行するには

1. コンテナ化されたアプリケーションの構築
2. Amazon ECR に発行する
3. シミュレーションの実行

AWS RoboMaker での開発

このセクションでは、AWS RoboMaker での開発環境のセットアップをサポートします。イメージをビルドして Amazon ECR に公開する方法と、イメージを使用してアプリケーションを開発する方法を説明します。

トピック

アプリケーションコンテナの構築

AWS RoboMaker でシミュレーションジョブを送信するには、アプリケーションコンテナの構築、コンテナの AWS RoboMaker アプリケーションへのリンク、コンテナを使用したシミュレーションジョブの送信、という 3 つのステップがあります。このセクションでは、Docker for AWS RoboMaker を使用してアプリケーションコンテナを構築する方法について説明します。hello-world のサンプルアプリケーションを使用して、ROS ベースのサンプルロボットとシミュレーションアプリケーションコンテナの構築に必要な手順を紹介します。このページでは、コンテナをローカルでテストする方法についても説明します。

ROS を使用していない場合は、AWS RoboMaker での GPU とコンテナサポートを使用した高忠実度シミュレーションの実行方法を説明したブログ記事をご覧ください。

セクション

前提条件

始める前に、開発環境に必要な依存関係があることを確認してください。Docker、AWS CLI、VCS インポートツールがマシンにインストールされている必要があります。

- AWS CLI のインストール先：https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html
- Dockerのインストール先：https://docs.docker.com/get-docker/
- （ワークフローで必要な場合）VCS インポートツールのインストール先：

```
sudo pip3 install vcstool
```

以下の権限を含む IAM ロールの AWS アカウントも必要です。

- IAM ロールの作成
- AWS RoboMaker リソース（シミュレーションジョブ、ロボット、シミュレーションアプリケーション）の作成
- Amazon ECR リポジトリの作成とアップロード

最後に、アカウント番号を控え、シミュレーションを実行するリージョンを選択します。AWS RoboMaker は、AWS RoboMaker エンドポイントとクォータ に記載されているリージョンでサポートされています。

ROS ワークスペースからのアプリケーションコンテナの構築

AWS RoboMaker シミュレーションは、シミュレーションアプリケーションとオプションのロボットアプリケーションで構成されます。これらのアプリケーションはそれぞれ名前とコンテナイメージで定義されます。このセクションでは、シミュレーションアプリケーションとロボットアプリケーションの両方のコンテナイメージを構築する方法を紹介します。次の例では、両方のアプリケーションが 1 つのワークスペース内で構築されています。以下のアプローチは、どの ROS プロジェクトにも簡単に一般化できます。

まず、`hello world` リポジトリをクローンしてソースをインポートします。

```
git clone https://github.com/aws-robotics/aws-robomaker-sample-application-
helloworld.git helloworld
cd helloworld
vcs import robot_ws < robot_ws/.rosinstall
```

```
vcs import simulation_ws < simulation_ws/.rosinstall
```

次に、helloworld ディレクトリで Dockerfile という名前の新しいファイルを作成します。以下の内容をコピーして貼り付けます。

```
# ======== ROS/Colcon Dockerfile ========
# This sample Dockerfile will build a Docker image for AWS RoboMaker
# in any ROS workspace where all of the dependencies are managed by rosdep.
#
# Adapt the file below to include your additional dependencies/configuration
# outside of rosdep.
# ======================================

# ==== Arguments ====
# Override the below arguments to match your application configuration.
# ===================

# ROS Distribution (ex: melodic, foxy, etc.)
ARG ROS_DISTRO=melodic
# Application Name (ex: helloworld)
ARG APP_NAME=robomaker_app
# Path to workspace directory on the host (ex: ./robot_ws)
ARG LOCAL_WS_DIR=workspace
# User to create and use (default: robomaker)
ARG USERNAME=robomaker
# The gazebo version to use if applicable (ex: gazebo-9, gazebo-11)
ARG GAZEBO_VERSION=gazebo-9
# Where to store the built application in the runtime image.
ARG IMAGE_WS_DIR=/home/$USERNAME/workspace

# ======== ROS Build Stages ========
# ${ROS_DISTRO}-ros-base
#   -> ros-robomaker-base
#      -> ros-robomaker-application-base
#         -> ros-robomaker-build-stage
#         -> ros-robomaker-app-runtime-image
# ==================================

# ==== ROS Base Image ============
# If running in production, you may choose to build the ROS base image
# from the source instruction-set to prevent impact from upstream changes.
# ARG UBUNTU_DISTRO=focal
# FROM public.ecr.aws/lts/ubuntu:${UBUNTU_DISTRO} as ros-base
```

```
# Instruction for each ROS release maintained by OSRF can be found here:
# https://github.com/osrf/docker_images
# ================================

# ==== Build Stage with AWS RoboMaker Dependencies ====
# This stage creates the robomaker user and installs dependencies required
# to run applications in RoboMaker.
# ================================

FROM public.ecr.aws/docker/library/ros:${ROS_DISTRO}-ros-base AS ros-robomaker-base
ARG USERNAME
ARG IMAGE_WS_DIR

RUN apt-get clean
RUN apt-get update && apt-get install -y \
   lsb  \
   unzip \
   wget \
   curl \
   xterm \
   python3-colcon-common-extensions \
   devilspie \
   xfce4-terminal

RUN groupadd $USERNAME && \
   useradd -ms /bin/bash -g $USERNAME $USERNAME && \
   sh -c 'echo "$USERNAME ALL=(root) NOPASSWD:ALL" >> /etc/sudoers'

USER $USERNAME
WORKDIR /home/$USERNAME

RUN mkdir -p $IMAGE_WS_DIR

# ==== ROS Application Base ====
# This section installs exec dependencies for your ROS application.
# Note: Make sure you have defined 'exec' and 'build' dependencies correctly
# in your package.xml files.
# ======================================
FROM ros-robomaker-base as ros-robomaker-application-base
ARG LOCAL_WS_DIR
ARG IMAGE_WS_DIR
ARG ROS_DISTRO
ARG USERNAME
```

```
WORKDIR $IMAGE_WS_DIR
COPY --chown=$USERNAME:$USERNAME $LOCAL_WS_DIR/src $IMAGE_WS_DIR/src

RUN sudo apt update && \
   rosdep update && \
   rosdep fix-permissions

# Note: This will install all dependencies.
# You could further optimize this by only defining the exec dependencies.
# Then, install the build dependencies in the build image.
RUN rosdep install --from-paths src --ignore-src -r -y

# ==== ROS Workspace Build Stage ====
# In this stage, we will install copy source files, install build dependencies
# and run a build.
# ===================================
FROM ros-robomaker-application-base AS ros-robomaker-build-stage
LABEL build_step="${APP_NAME}Workspace_Build"
ARG APP_NAME
ARG LOCAL_WS_DIR
ARG IMAGE_WS_DIR

RUN  . /opt/ros/$ROS_DISTRO/setup.sh && \
   colcon build \
    --install-base $IMAGE_WS_DIR/$APP_NAME

# ==== ROS Robot Runtime Image ====
# In the final stage, we will copy the staged install directory to the runtime
# image.
# ================================
FROM ros-robomaker-application-base AS ros-robomaker-app-runtime-image
ARG APP_NAME
ARG USERNAME
ARG GAZEBO_VERSION

ENV USERNAME=$USERNAME
ENV APP_NAME=$APP_NAME
ENV GAZEBO_VERSION=$GAZEBO_VERSION

RUN rm -rf $IMAGE_WS_DIR/src

COPY --from=ros-robomaker-build-stage $IMAGE_WS_DIR/$APP_NAME $IMAGE_WS_DIR/$APP_NAME

# Add the application source file to the entrypoint.
```

```
WORKDIR /
COPY entrypoint.sh /entrypoint.sh
RUN sudo chmod +x /entrypoint.sh && \
   sudo chown -R $USERNAME /entrypoint.sh && \
   sudo chown -R $USERNAME $IMAGE_WS_DIR/$APP_NAME

ENTRYPOINT ["/entrypoint.sh"]
```

先ほど作成した Dockerfile は、Docker イメージを構築するために使用される命令セットです。Dockerfile のコメントを読んで何が構築されるのかを理解し、必要に応じて変更してください。開発を容易にするため、Dockerfile は Open Source Robotics Foundation (OSRF) が管理する ROS の公式 Docker イメージに基づいています。ただし、本番環境で実行する場合は、アップストリームの変更による影響を防ぐために、GitHub の OSRF ソース命令セットを使用して ROS のベースイメージをビルドすることもできます。

次に、entrypoint.sh という名前の新しいファイルを作成します。

```
#!/bin/bash
set -e
source "/home/$USERNAME/workspace/$APP_NAME/setup.bash"
if [[ -f "/usr/share/$GAZEBO_VERSION/setup.sh" ]]
then
   source /usr/share/$GAZEBO_VERSION/setup.sh
fi
printenv
exec "${@:1}"
```

ENTRYPOINT ファイルは Docker コンテナの生成時に実行される実行ファイルです。ROS ワークスペースのソースにエントリポイントを使用しているため、AWS RoboMaker で簡単に roslaunch のコマンドを実行できます。この ENTRYPOINT ファイルに、独自の環境設定ステップを追加することもできます。

Dockerfile では、マルチステージビルドと Docker BuildKit との統合キャッシュを使用します。マルチステージビルドでは個別のビルドステップによるワークフローが可能なため、ビルドの依存関係やソースコードはランタイムイメージにコピーされません。このため Docker イメージのサイズが小さくなり、パフォーマンスが向上します。キャッシュ操作は、以前にビルドされたファイルを保存することで、その後のビルドを加速します。

次のコマンドを使用してロボットアプリケーションをビルドします。

```
DOCKER_BUILDKIT=1 docker build . \
```

```
 --build-arg ROS_DISTRO=melodic \
 --build-arg LOCAL_WS_DIR=./robot_ws \
 --build-arg APP_NAME=helloworld-robot-app \
 -t robomaker-helloworld-robot-app
```

ロボットアプリケーションをビルドしたら、次のようにシミュレーションアプリケーションをビルドできます。

```
DOCKER_BUILDKIT=1 docker build . \
 --build-arg GAZEBO_VERSION=gazebo-9 \
 --build-arg ROS_DISTRO=melodic \
 --build-arg LOCAL_WS_DIR=./simulation_ws \
 --build-arg APP_NAME=helloworld-sim-app \
 -t robomaker-helloworld-sim-app
```

docker images コマンドを実行して Docker イメージが正常にビルドされたことを確認します。出力は次のようになります。

```
Administrator:~/environment/helloworld (ros1) $ docker images
REPOSITORY                        TAG          IMAGE ID       CREATED          SIZE
robomaker-helloworld-sim-app      latest       5cb08816b6b3   6  minutes ago  2.8GB
robomaker-helloworld-robot-app    latest       b5f6f755feec   10 minutes ago  2.79GB
```

これで、Docker イメージが正常にビルドされました。これらをアップロードしてで使用する前に、AWS RoboMaker を使ってローカルでテストすることをおすすめします。次のセクションは、これを行う方法を示しています。

コンテナのテスト

次のコマンドを使用すると、ローカル開発環境でアプリケーションを実行できます。

ロボットアプリケーションを起動します。

```
docker run -it -v /tmp/.X11-unix/:/tmp/.X11-unix/ \
-u robomaker -e ROBOMAKER_GAZEBO_MASTER_URI=http://localhost:5555 \
-e ROBOMAKER_ROS_MASTER_URI=http://localhost:11311 \
robomaker-helloworld-robot-app:latest roslaunch hello_world_robot rotate.launch
```

シミュレーションアプリケーションを起動します。

```
docker run -it -v /tmp/.X11-unix/:/tmp/.X11-unix/ \
```

```
-u robomaker -e ROBOMAKER_GAZEBO_MASTER_URI=http://localhost:5555 \
-e ROBOMAKER_ROS_MASTER_URI=http://localhost:11311 \
robomaker-helloworld-sim-app:latest roslaunch hello_world_simulation empty_world.launch
```

コンテナが正常に機能していることを確認したら、アプリケーションコンテナを Amazon ECR に公開して、シミュレーションジョブを送信することができます。

Amazon ECR へのアプリケーションコンテナの公開

シミュレーションジョブで AWS RoboMaker が使用するコンテナは、フルマネージド型のコンテナレジストリである Amazon Elastic Container Registry (ECR) に保存する必要があります。正常にアプリケーションコンテナをビルドしたら、これを Amazon ECR にプッシュします。このセクションでは、その方法を説明します。

はじめに、以下のコマンドで再利用する環境変数をいくつか設定することで、繰り返し入力する手間を省くことができます。

```
export robotapp=robomaker-helloworld-robot-app
export simapp=robomaker-helloworld-sim-app
export account=<YOUR AWS ACCOUNT NUMBER>
export region=<YOUR AWS REGION>
export ecruri=$account.dkr.ecr.$region.amazonaws.com
```

次に、サインインして 2 つの新しいリポジトリを作成します。

```
aws ecr get-login-password --region $region | docker login --username AWS --password-
stdin $ecruri
aws ecr create-repository --repository-name $robotapp
aws ecr create-repository --repository-name $simapp
```

Docker イメージには Amazon ECR リポジトリの URI をタグ付けできます。

```
docker tag $robotapp $ecruri/$robotapp:latest
docker tag $simapp $ecruri/$simapp:latest
```

次に、Docker イメージを Amazon ECR にプッシュします。

```
docker push $ecruri/$robotapp
```

```
docker push $ecruri/$simapp
```

最後に、以下のコマンドを実行して Amazon ECR にアップロードされたイメージを確認できます。

```
aws ecr list-images --repository-name $simapp
aws ecr list-images --repository-name $robotapp
```

以下のコードスニペットは、期待される出力を示します。

```
Administrator:~/environment/helloworld (ros1) $ aws ecr list-images --repository-name
 $simapp
{
   "imageIds": [
       {
           "imageDigest": "sha256:28cad40230402343024kf303f30fk20f2f2fa0a8148",
           "imageTag": "latest"
       }
   ]
}
Administrator:~/environment/helloworld (ros1) $ aws ecr list-images --repository-name
 $robotapp
{
   "imageIds": [
       {
           "imageDigest": "sha256:28cad40230402343024kf303f30fk20f2f2fa0a8148",
           "imageTag": "latest"
       }
   ]
}
```

これで、ロボットとシミュレーションの Docker イメージが Amazon ECR 内にホストされました。シミュレーションジョブの送信に進む前に、ロボットアプリケーションまたはシミュレーションアプリケーションとこれらのイメージを関連付ける必要があります。

アプリケーションのバージョニング

AWS RoboMaker は、ロボットアプリケーションとシミュレーションアプリケーションの複数のバージョンの作成をサポートしています。これにより、ロボットとシミュレーションで使用するコードを制御できます。バージョンは、アプリケーションの $LATEST バージョンの番号付きスナップショットです。バージョンは、開発ワークフローの段階別に作成できます。例えば、開発、ベータデプロイ、本番稼働用に別のバージョンを使用します。

AWS RoboMaker のロボットアプリケーションまたはシミュレーションアプリケーションのバージョニングを行う場合は、アプリケーションのスナップショットを作成します。Amazon ECR では、イメージダイジェストを使用してアプリケーションのバージョンが表示されます。AWS RoboMaker は、各バージョンのイメージダイジェストを記憶します。

イメージが Amazon ECR にアップロードされており、イメージダイジェストを変更していない場合は、そのバージョンのアプリケーションにアクセスして使用することができます。アプリケーションごとに作成できるバージョンは最大 40 個です。

イメージの作成時にイメージにタグを適用することもできます。$LATEST バージョンでは、タグフィールドの値を `latest` として指定することができます。これらの値は相互に区別されます。

イメージに `latest` タグを付ける方法は 2 つあります。

- `latest` の値を含むタグを指定した。
- タグが付いていないイメージをプッシュすると、Amazon ECR により `latest` タグが付いてイメージが更新されます。

AWS RoboMaker では、イメージに対してタグを指定すると、そのイメージは常に $LATEST バージョンとして選別されます。例えば、イメージ名が myImage、タグが xyz、イメージダイジェストが 123 である口ボットアプリケーションを作成すると、$LATEST バージョンはダイジェストが 123 である `myImage:xyz` になります。

以下は、タグを追加する際のシナリオです。

- 新しいタグを使用するために、$LATEST バージョンに更新します。たとえば、myImage というイメージがある場合は、abc タグを使用してイメージを更新できます。$LATEST バージョンのイメージは `myImage:abc` を指しています。
- イメージを更新して再びタグを付けします。例えば、タグ abc が付いているイメージに変更を加えることができます。。タグ xyz は更新後に使用できます。$LATEST バージョンは `myImage:xyz` を指しています。

詳細については、「バージョニングアプリケーション」を参照してください。

ロボットアプリケーションの使用

AWS RoboMaker ロボットアプリケーションは、ロボットのアプリケーションスタックを実行するコンテナイメージです。ロボットアプリケーションのイメージは Amazon ECR でホストする必要があ

ります。多くの場合、ロボットアプリケーションをシミュレーションアプリケーションと組み合わせてシミュレーションジョブを作成します。

セクション

ロボットアプリケーションの作成

Using the console

1. https://console.aws.amazon.com/robomaker/ で AWS RoboMaker コンソールにサインインします。
2. 左側のペインで、[Development] (開発)、[Robot applications] (ロボットアプリケーション) の順に選択します。
3. [Create robot application] (ロボットアプリケーションの作成) を選択します。
4. [Create robot application] (ロボットアプリケーションの作成) ページで、ロボットアプリケーションの名前を入力します。ロボットを識別しやすい名前を選択します。
5. Amazon ECR コンテナイメージを提供します。Amazon ECR にプッシュしたイメージを使用できます。詳細については、「Amazon Elastic Container Registry とは?」を参照してください。
6. タグ付けの詳細については、AWS RoboMaker リソースのタグ付け を参照してください。
7. [Create] (作成) を選択します。

Using the AWS CLI

```
aws robomaker create-robot-application \
--name my-robot-app \
```

```
--robot-software-suite name=General \
--environment uri=:<ACCOUNT>.dkr.ecr.<REGION>.amazonaws.com/my-robot-app:latest
```

アプリケーションバージョンの作成

Using the console

1. https://console.aws.amazon.com/robomaker/ で AWS RoboMaker コンソールにサインインします。
2. 左側のナビゲーションペインで、[Development] (開発)、[Robot applications] (ロボットアプリケーション) の順に選択します。
3. ロボットアプリケーションの名前を選択します。
4. [Robot applications details] (ロボットアプリケーションの詳細) ページで、[Create new version] (新しいバージョンの作成)、[Create] (作成) の順に選択します。

Using the AWS CLI

```
aws robomaker create-robot-application-version --name my-robot-app-arn
```

ロボットアプリケーションの表示

Using the console

1. https://console.aws.amazon.com/robomaker/ で AWS RoboMaker コンソールにサインインします。
2. 左側のナビゲーションペインで、[Development] (開発)、[Robot applications] (ロボットアプリケーション) の順に選択します。
3. ロボットアプリケーションの名前を選択します。

Using the AWS CLI

```
aws robomaker describe-robot-application --application my-robot-application-arn
```

ロボットアプリケーションの更新

Using the console

1. https://console.aws.amazon.com/robomaker/ で AWS RoboMaker コンソールにサインインします。
2. 左側のナビゲーションペインで、[Development] (開発)、[Robot applications] (ロボットアプリケーション) の順に選択します。
3. 更新するロボットアプリケーションの横にあるチェックボックスをオンにします。
4. [Actions] (アクション)、[Update] (更新) の順に選択します。
5. ソースは追加または削除できますが、少なくとも 1 つのソースロボットアプリケーションファイルが必要です。
6. [Update] (更新) を選択して、ロボットアプリケーションを更新します。

Using the AWS CLI

```
aws robomaker update-robot-application \
--application my-robot-application-arn \
--robot-software-suite name=General \
--environment uri=:<ACCOUNT>.dkr.ecr.<REGION>.amazonaws.com/my-robot-app:latest
```

ロボットアプリケーションの削除

Using the console

1. AWS RoboMaker コンソール (https://console.aws.amazon.com/robomaker/) にサインインします。
2. 左側のナビゲーションペインで、[Development] (開発)、[Robot applications] (ロボットアプリケーション) の順に選択します。

3. ロボットアプリケーションの名前を選択し、ロボットの作成日時や最終更新日などの詳細を表示します。
4. ロボットアプリケーションバージョンの詳細ページで、[Delete] (削除) を選択し、さらに [Delete] (削除) を選択して確定します。

Using the AWS CLI

```
aws robomaker delete-robot-application --application my-robot-application-arn
```

ロボットアプリケーションバージョンの削除

Using the console

1. AWS RoboMaker コンソール (https://console.aws.amazon.com/robomaker/) にサインインします。
2. 左側のナビゲーションペインで、[Development] (開発)、[Robot applications] (ロボットアプリケーション) の順に選択します。
3. ロボットアプリケーションの名前を選択してそのバージョンを表示します。
4. ロボットの詳細ページで [Version] (バージョン) を選択してバージョンの詳細を表示します。
5. ロボットアプリケーションバージョンの詳細ページで、[Delete] (削除) を選択し、さらに [Delete] (削除) を選択して確定します。

Using the AWS CLI

```
aws robomaker delete-robot-application-version \
--application my-robot-application-arn \
--version 2
```

シミュレーションアプリケーションの使用

AWS RoboMaker シミュレーションアプリケーションは、ロボットのシミュレーションスタックを実行するコンテナイメージです。シミュレーションアプリケーションのイメージは Amazon ECR でホストする必要があります。多くの場合、シミュレーションジョブを作成するために、シミュレーションアプリケーションをロボットアプリケーションと組み合わせます。

セクション

シミュレーションアプリケーションの作成

Using the console

1. AWS RoboMaker コンソール (https://console.aws.amazon.com/robomaker/) にサインインします。
2. 左側のナビゲーションペインで、[Development] (開発)、[Simulation applications] (シミュレーションアプリケーション) の順に選択します。
3. [シミュレーションアプリケーションの作成] を選択します。
4. [シミュレーションアプリケーションの作成] ページで、シミュレーションアプリケーションの[Name] (名前) を入力します。シミュレーションを識別しやすい名前を選択します。
5. Amazon ECR コンテナイメージを提供します。Amazon ECR にプッシュしたイメージを使用できます。詳細については、「Amazon VPC とは？」を参照してください。
6. タグ付けの詳細については、AWS RoboMaker リソースのタグ付け を参照してください。
7. [Create] (作成) を選択します。

Using the AWS CLI

```
aws robomaker create-simulation-application \
--name my-sim-app \
--simulation-software-suite name=SimulationRuntime \
 --robot-software-suite name=General \
--environment uri=:<ACCOUNT>.dkr.ecr.<REGION>.amazonaws.com/my-sim-app:latest
```

シミュレーションアプリケーションバージョンの作成

Using the console

1. AWS RoboMaker コンソール (https://console.aws.amazon.com/robomaker/) にサインインします。
2. 左側のナビゲーションペインで、[Development] (開発)、[Simulation applications] (シミュレーションアプリケーション) の順に選択します。
3. シミュレーションアプリケーションの[名前] を選択します。
4. [シミュレーションアプリケーションの詳細] ページで、[新しいバージョンの作成] 、[作成] の順に選択します。

Using the AWS CLI

```
aws robomaker create-simulation-application-version --name my-simulation-
application-arn
```

シミュレーションアプリケーションの表示

Using the console

1. AWS RoboMaker コンソール (https://console.aws.amazon.com/robomaker/) にサインインします。
2. 左側のナビゲーションペインで、[Development (開発)、[Simulation applications] (シミュレーションアプリケーション) の順に選択します。

3. シミュレーションアプリケーションの名前を選択し、作成日時や最終更新日などの詳細を表示します。

Using the AWS CLI

```
aws robomaker describe-simulation-application --job my-simulation-job-arn
```

シミュレーションアプリケーションの更新

Using the console

1. AWS RoboMaker コンソール (https://console.aws.amazon.com/robomaker/) にサインインします。
2. 左側のナビゲーションペインで、[Development (開発)、[Simulation applications] (シミュレーションアプリケーション) の順に選択します。
3. 更新するシミュレーションアプリケーションの横にあるチェックボックスをオンにします。
4. [Actions] (アクション)、[Update] (更新) の順に選択します。
5. ソースは追加または削除できますが、少なくとも 1 つのソースシミュレーションアプリケーションファイルが必要です。
6. [Update] (更新) を選択して、シミュレーションアプリケーションを更新します。

Using the AWS CLI

```
aws robomaker update-simulation-application \
--application my-simulation-application-arn \
--robot-software-suite name=General \
--simulation-software-suite name=SimulationRuntime \
--environment uri=:<ACCOUNT>.dkr.ecr.<REGION>.amazonaws.com/my-simulation-app:latest
```

シミュレーションアプリケーションの削除

Using the console

1. AWS RoboMaker コンソール (https://console.aws.amazon.com/robomaker/) にサインインします。
2. 左側のナビゲーションペインで、[Development (開発)、[Simulation applications] (シミュレーションアプリケーション) の順に選択します。
3. シミュレーションアプリケーションの [Name] (名前) を選択します。これには、作成日時や最終更新日時などの詳細が表示されます。
4. シミュレーションアプリケーションの詳細ページで、[Delete] (削除)、[Delete] (削除) の順に選択し、削除を確定します。

Using the AWS CLI

```
aws robomaker delete-simulation-application --application my-simulation-application-arn
```

シミュレーションアプリケーションバージョンの削除

Using the console

1. AWS RoboMaker コンソール (https://console.aws.amazon.com/robomaker/) にサインインします。
2. 左側のナビゲーションペインで、[Development (開発)、[Simulation applications] (シミュレーションアプリケーション) の順に選択します。
3. シミュレーションアプリケーションの名前を選択してそのバージョンを表示します。
4. シミュレーションの詳細ページで、[バージョン] を選択して詳細を表示します。
5. その詳細ページで、[削除] を選択し、[削除] を選択して確定します。

Using the AWS CLI

```
aws robomaker delete-simulation-application-version \
--application my-simulation-application-arn \
--version 2
```

バージョニングアプリケーション

AWS RoboMaker は、ロボットアプリケーションとシミュレーションアプリケーションの複数のバージョンの作成をサポートしています。これにより、ロボットとシミュレーションで使用するコードを制御できます。バージョンは、アプリケーションの $LATEST バージョンの番号付きスナップショットです。バージョンは、開発ワークフローの段階別に作成できます、たとえば開発、ベータデプロイ、本稼働。

AWS RoboMaker のロボットアプリケーションまたはシミュレーションアプリケーションのバージョニングを行う場合は、アプリケーションのスナップショットを作成します。

アプリケーションのビルドに `colcon` を使用している場合、AWS RoboMaker では、バージョンごとにファイルの Amazon S3 パスと ETag が記憶されます。アプリケーションのバージョンは、Amazon S3 パスにまだ存在し、変更されていない (ETag が変更されていない) 限り、作成時に存在していたとおりに使用できます。

アプリケーションにコンテナイメージを使用している場合は、イメージを Amazon ECR にアップロードします。Amazon ECR では、イメージダイジェストを使用してアプリケーションのバージョンが表示されます。AWSRoboMaker では、各バージョンのイメージダイジェストが記憶されます。

イメージが Amazon ECR にアップロードされており、イメージダイジェストを変更していない場合は、そのバージョンのアプリケーションにアクセスして使用することができます。

アプリケーションごとに作成できるバージョンは最大 40 個です。

トピック

- 画像によるアプリケーションのバージョニング
- $LATEST バージョン
- アプリケーションバージョンを更新する
- アプリケーションバージョンを削除する

画像によるアプリケーションのバージョニング

アプリケーションの開発時に $LATEST バージョンのコンテナイメージを更新できます。$LATEST バージョンを選択すると、指定した Amazon ECR の場所からそのバージョンを取得できます。

イメージの作成時にイメージにタグを適用することもできます。タグフィールドの値を $LATEST バージョンの "latest" として指定することができます。これらの値は相互に区別されます。

イメージに "latest" タグを付ける方法は 2 つあります。

- "latest" の値を含むタグを指定した。
- タグが付いていないイメージをプッシュすると、Amazon ECR により "latest" タグが付いてイメージが更新されます。

AWS RoboMaker では、イメージに対してタグを指定すると、そのイメージは常に $LATEST バージョンとして選別されます。例えば、イメージ名が "myImage"、タグが "xyz"、イメージダイジェストが "123" であるロボットアプリケーションを作成すると、$LATEST バージョンはダイジェストが "123" である myImage:xyz になります。

以下は、タグを追加する際のシナリオです。

- 新しいタグを使用するために $LATEST バージョンを更新する場合。たとえば、"myImage" というイメージがある場合は、"abc" タグを使用してイメージを更新できます。$LATEST バージョンのイメージは myImage:abc を指しています。
- イメージを更新して再びタグを付ける場合。例えば、タグ "abc" が付いているイメージに変更を加えることができます。。タグ "xyz" は更新後に使用できます。$LATEST バージョンは myImage:xyz を指しています。

$LATEST バージョン

バージョンを作成すると、AWS RoboMaker は $LATEST バージョンのスナップショットを取得し、バージョン番号を 1 ずつ増やします。AWSRoboMaker により、ファイルの Amazon S3 パスと ETag が記憶されます。パスは、ファイルを取得するために使用されます。ETag は、変更されていないことを確認するために使用されます。バージョン番号が再利用されることはありません。例えば、最新バージョンが 10 で、それを削除してから新しいバージョンを作成した場合、新しいバージョンはバージョン 11 になります。

アプリケーションの開発時に $LATEST バージョンを更新できます。$LATEST バージョンを選択すると、指定した Amazon S3 の場所から取得されます。例えば、ロボットアプリケーションとシミュレーションアプリケーションの最新バージョンを使用してシミュレーションジョブを開始し、その Amazon S3 パスでロボットアプリケーションを変更してから、シミュレーションジョブを再起動すると、更新されたロボットアプリケーションが使用されます。

ロボットアプリケーションをデプロイする場合は、デプロイする特定の番号付きバージョンを選択する必要があります。ロボットアプリケーションバージョンを作成する方法の詳細については、「アプリケーションバージョンの作成」を参照してください。

シミュレーションアプリケーションバージョンを作成する方法の詳細については、「シミュレーションアプリケーションバージョンの作成」を参照してください。ETags の詳細については、「一般的なレスポンスヘッダー」を参照してください。

アプリケーションバージョンを更新する

更新できるのは $LATEST バージョンの AWS RoboMaker アプリケーションのみです。更新したバージョンは、AWS RoboMaker で使用できるようになります。例えば、シミュレーションジョブを再起動すると、最新バージョンのアプリケーションがシミュレーションで使用されます。

詳細については、ロボットアプリケーションの更新 および シミュレーションアプリケーションの更新 を参照してください。

アプリケーションバージョンを削除する

不要になったアプリケーションバージョンは削除できます。詳細については、ロボットアプリケーションバージョンの削除 および シミュレーションアプリケーションバージョンの削除 を参照してください。

イメージを使用した AWS RoboMaker アプリケーションの開発

> **Important**
>
> 2022 年 3 月 15 日より AWS RoboMaker シミュレーションに変更が加えられ、これにより既存のシミュレーションジョブに影響が及んだ可能性があります。これらの変更点と、ロボットアプリケーション、シミュレーションアプリケーション、シミュレーションジョブに適用できる移行手順の詳細については、「ROS アプリケーションのコンテナへの移行」を参照してください。

1 つまたは複数のコンテナイメージを使用した、シミュレーションおよびロボットアプリケーションの開発と実行が可能です。イメージの詳細については、「Amazon ECS の Docker の基本」を参照してください。使用するイメージは AWS RoboMaker 互換コンテナ要件 に記載されている要件を満たしている必要があります。

サポートする開発環境のいずれかを使用している場合は、AWS RoboMaker で独自のイメージを使用できます。

コンテナイメージを使用したアプリケーション開発には複数の方法があります。アプリケーションの開発方法の例については、「イメージを作成して Hello World サンプルアプリケーションを実行する」を参照してください。

イメージを使用してアプリケーションを開発した後、それらをテストすることができます。アプリケーションの動作をテストする場合、ローカルの Linux マシンでアプリケーションを視覚化できます。

シミュレーションの動作をテストした後、イメージを Amazon ECR にプッシュし、シミュレーションジョブを実行して仮想環境における」ロボットのインタラクションを確認できます。

トピック

- ROS アプリケーションのコンテナへの移行
- ROS コンテナに関するよくある質問
- AWS RoboMaker 互換コンテナ要件
- GPU アプリケーションを実行するためのイメージの作成
- イメージを作成して Hello World サンプルアプリケーションを実行する

ROS アプリケーションのコンテナへの移行

2021 年 10 月より、AWS RoboMaker はあらゆるロボットとシミュレーションソフトウェアの組み合わせにも対応できるようサポートを拡大しました。これまで、AWS RoboMaker での実行がサポートされているロボットとシミュレーションソフトウェアの構成は、ロボットオペレーティングシステム (ROS) と Gazebo だけでした。この変更により、AWS RoboMaker でのシミュレーションの実行中に、任意のロボットとシミュレーションソフトウェアを設定できるようになりました。

ROS と Gazebo を引き続き使用したいと考えている場合、これはどういう意味になるのでしょうか?

AWS RoboMaker で使用する独自のアプリケーションコンテナをビルドするには、Docker ベースのワークフローに移行する必要があるということです。Docker は、開発者がアプリケーションの依存関係をバンドルし、ソフトウェアをバンドルパッケージ (コンテナ) として出荷できるようにする業界標準ツールです。詳細については、「Amazon ECS の Docker の基本」を参照してください。使用するイメージは AWS RoboMaker 互換コンテナ要件 に記載されている要件を満たしている必要があります。

ROS ベースのコンテナをすでに使用している場合はどうなりますか?

この場合、ほとんど作業は必要ありません。AWS コンソールまたは CLI 経由で、ロボットとシミュレーションアプリケーションのソフトウェアスイートを、ROS 関連のソフトウェアスイートから一般的およびシミュレーションランタイムのソフトウェアスイートへとアップデートする必要があります。次に、シミュレーションの実行 の手順に従います。

Docker ベースのワークフローに移行する方法

1. お好きな ROS のバージョンに応じて以下のチュートリアルのいずれかを選択し、その手順に従ってください。
   - ROS Melodic と Gazebo 9 でサンプルアプリケーションを実行する
   - ROS 2 Foxy と Gazebo 11 を使用したサンプルアプリケーションの実行
2. コンテナを作成したら、引き続きシミュレーションジョブの送信ができます。
   - シミュレーションの実行

ROS コンテナに関するよくある質問

このページでは、ROS ベースのロボットおよびシミュレーションアプリケーションを AWS RoboMaker の実行に適した Docker コンテナに移行することに関してよく寄せられる質問と回答をまとめています。

このワークフローでは、`colcon` をバンドルしたロボットとシミュレーションアプリケーションを使用してシミュレーションジョブを送信します。移行する必要がありますか?

はい、移行は必要です。移行の手順は「ROS アプリケーションのコンテナへの移行」に記載されています。

ロボットとシミュレーションアプリケーションを移行する必要があるかどうかがわかりません。どうすればわかりますか?

AWS コンソールまたは AWS CLI を使用して確認できます。手順については、次の該当するタブを選択してください。

Using the console

1. AWS RoboMaker コンソールにサインインします。
2. 左側のナビゲーションペインで、[Development] (開発)、[Simulation applications] (シミュレーションアプリケーション) の順に選択します。
3. シミュレーションアプリケーションの [名前] を選択すると、その詳細が表示されます。

   [全般] と [シミュレーションランタイム] と表示されている場合、移行は必要ありません。ROS または Gazebo 固有の値が表示される場合は、移行が必要です。

Using the AWS CLI

Example

コンソールを使用した手順と同じ処理を AWS CLI コマンドで実行する例は次のとおりです。

```
aws robomaker describe-simulation-application --application YOUR-SIM-APP-ARN
```

このコマンドは、simulationSoftwareSuite、robotSoftwareSuite (該当する場合)、environment の URI を示す出力を返します。simulationSoftwareSuite に [シミュレーションランタイム]、robotSoftwareSuite に [全般] が表示され、environment の URI が設定されている場合、シミュレーションアプリケーションを移行する必要はありません。

ロボットとシミュレーションアプリケーションのコンテナはどのようにして相互に通信しますか?

これは、ROS ベースのアプリケーションが ROS のミドルウェアを使用して一般的に相互通信する方法と変わりません。ただし、シミュレーションジョブリクエストの起動設定オブジェクト内に ROS 固有の環境変数を設定する必要があります。

以下は、ロボットアプリケーション launchConfig に使用する必要がある設定のスニペット例です。

```
"robotApplications": [
    {
```

```
        "application": "YOUR-ROBOT-APP-ARN",
        "applicationVersion": "$LATEST",
        "launchConfig": {
            "environmentVariables": {
                "ROS_IP": "ROBOMAKER_ROBOT_APP_IP",
                "ROS_MASTER_URI": "http://ROBOMAKER_ROBOT_APP_IP:11311",
                "GAZEBO_MASTER_URI": "http://ROBOMAKER_SIM_APP_IP:11345"
        },
 ... # Removed extra data for clarity
    }
]
```

以下は、シミュレーションアプリケーション launchConfig に使用する必要がある設定のスニペット例です。

```
"simulationApplications": [
    {
        "application": "YOUR-SIM-APP-ARN",
        "applicationVersion": "$LATEST",
        "launchConfig": {
            "environmentVariables": {
                "ROS_IP": "ROBOMAKER_SIM_APP_IP",
                "ROS_MASTER_URI": "http://ROBOMAKER_ROBOT_APP_IP:11311",
                "GAZEBO_MASTER_URI": "http://ROBOMAKER_SIM_APP_IP:11345"
        },
 ... # Removed extra data for clarity
    }
]
```

ROS_IP、ROS_MASTER_URI、GAZEBO_MASTER_URI の設定に提供された ROBOMAKER_* 文字列とポート番号を使用すれば、コンテナは想定通りに相互通信を行います。

詳細については、「シミュレーションの実行」を参照してください。

リアルタイム係数 (RTF) メトリクスはどこに行きましたか? 復元するにはどうすればいいですか?

AWS RoboMaker では、このメトリクスが自動で公開されなくなりました。このメトリクスを CloudWatch に公開する場合は、AWS RoboMaker CloudWatch Publisher をシミュレーションアプリケーションにインポートし、README.md ファイルに記載された指示に従ってシミュレーションの起動ファイルを修正する必要があります。

シミュレーションジョブをキャンセルしてタグ付けする方法を教えてください。

VPC 設定を使用すると、汎用 AWS API を使用した AWS RoboMaker のシミュレーションジョブを独立してタグ付け、またはキャンセルできます。次の方法を使用するには、NAT または IGW 経由の AWS API へのパブリックルートがある VPC でコンテナが実行されている必要があります。最も簡単な方法は、デフォルト VPC のパブリックサブネットを使用して AWS API に接続することです。プライベートサブネットでシミュレーションを実行したい場合は、代わりに NAT を設定するか、インターフェイス VPC エンドポイントを設定することもできます。詳細については、「AWS RoboMaker とインターフェース VPC エンドポイント (AWS PrivateLink)」を参照してください。

> **Note**
>
> IGW を使用している場合は、以下のドキュメントの説明に従って `assignPublicIp=True` を設定してください。パブリック IP を使用している場合は、セキュリティグループが十分にロックダウンされていることを確認してください。

リクエストパラメータに次のブロックを追加してください。

```
vpcConfig={
   'subnets': [
      'string',
    ],
    'securityGroups': [
       'string',
    ],
    'assignPublicIp': True|False
},
```

さらに、AWS RoboMaker のシミュレーションジョブには、シミュレーションジョブにタグを付けたりキャンセルしたりする権限を持つ IAM ロールが必要です。

シミュレーションジョブでは、AWS CLI または boto3 Python ライブラリを使用してパブリック AWS RoboMaker API を呼び出すことができます。AWS CLI と boto3 ライブラリは AWS RoboMaker シミュレーションジョブで使用する前に、コンテナにプリインストールしておく必要があります。次の Python サンプルコードは、シミュレーションジョブをキャンセルする方法を示しています。

```
class RoboMakerUtils:

    def __init__(self):
        self.job_arn = os.getenv('AWS_ROBOMAKER_SIMULATION_JOB_ARN')
```

```
        self.client = boto3.client('robomaker',
region_name=os.getenv('AWS_ROBOMAKER_REGION', 'us-east-1'))

    def tag_robomaker_sim_job(self, key, value):
        self.client.tag_resource(
            resourceArn=self.job_arn,
            tags={
               key: str(value)
            }
        )

    def cancel_robomaker_sim_job(self):
        self.tag_robomaker_sim_job("END_TIME", time.time())
        response = self.client.cancel_simulation_job(
            job=self.job_arn
        )
```

Simulation WorldForge のワールドをシミュレーションジョブにインポートする方法を教えてください。

Simulation WorldForge のアセットをシミュレーションジョブにインポートする必要がある場合は、`DataSource` API を使用してください。こうすることで、ワールドのエクスポートジョブの Amazon S3 の出力ディレクトリにあるワールドアセットを、シミュレーションジョブコンテナ内の好きなインポート先へインポートすることができます。

詳細については、「シミュレーションでのエクスポートしたワールドの使用」を参照してください。

アプリケーションのログファイルが作成されていません。何が起きているのでしょうか?

関連するアーティファクトのデバッグに使用するすべての出力ディレクトリを Dockerfile に作成していることを確認してください。たとえば、Dockerfile に次の行を追加します。

```
RUN mkdir -p $YOUR_LOG_DIR
```

詳細については、「カスタムアップロード設定の追加」を参照してください。

シミュレーションアプリケーションが「パラメータサーバーの run_id が宣言された run_id と一致しない」という理由で失敗しました。どうすればよいですか?

ロボットアプリケーションとシミュレーションアプリケーションの両方で ROS のシミュレーションジョブを起動する場合は、roslaunch コマンドに `--wait` を追加する必要があります.

AWS RoboMaker 互換コンテナ要件

AWS RoboMaker 互換コンテナ (コンテナイメージ)の実行とシミュレーションの正常な開始に関して、一連の要件を満たす必要があります。これらの要件を満たしていても、シミュレーションの実行に問題がある場合は、「シミュレーションジョブ」および「Simulation WorldForge」を参照してください。

シミュレーションランタイム要件

コンテナイメージでは Dockerfile にある VOLUME は使用できません。VOLUME が Dockerfileに入っている場合、4XX エラーコードが出てシミュレーションが失敗します。

コンテナイメージでは Dockerfile にある EXPOSE は使用できません。EXPOSE が Dockerfile に入っている場合、AWS RoboMaker は 4XX エラーコードが出てシミュレーションが失敗します。

コンテナイメージのサイズは、圧縮後で 20 GB 以下でなければなりません。コンテナイメージの未圧縮サイズが 20 GB より大きい場合、AWS RoboMaker は 4XX エラーコードが出てシミュレーションが失敗します。

Dockerfile 内にある CMD は指定できません。指定すると、AWS RoboMaker によってパッケージ名と起動ファイルでそれがオーバーライドされます。代わりに、各シミュレーションアプリケーションまたはロボットアプリケーションの launchConfig の command パラメータを CreateSimulationJob リクエスト内で使用して、起動コマンドのリストを提供することができます。これは、シミュレーションジョブで CMD として設定されます。例: command は ["/bin/bash", "-c", "sleep 365d"] です。

シミュレーションジョブにツールを追加する場合は、コンテナイメージに bash をインストールする必要があります。ツールは ["/bin/bash", "-c", "<command>"] で起動します。

コンテナが ROS を実行していて、ロボットアプリケーションとシミュレーションアプリケーション間の通信が必要な場合は、以下のロボットフレームワークを設定する必要があります。

- ROS Master
- Gazebo Master
- ROS IP

コンテナ内の /etc/resolv.conf ファイルはカスタマイズできません。AWS RoboMaker は独自のファイルでファイルを上書きします。

AWS で Dockerfile を実行している場合、イメージをマウントすることはできません。Dockerfile 内の `Mount` を指定した場合、AWS RoboMaker は `4XX` エラーコードが出てシミュレーションが失敗します。

デフォルトの Docker `seccomp` プロファイルによってブロックされているシステムコールは、コンテナイメージによって使用されません。ブロックされたシステムコールについては、「Seccomp セキュリティプロファイル」を参照してください。

イメージを実行するユーザーを指定するために、Dockerfile の `USER` キーワードを指定することができます。ユーザーを指定しない場合、AWS RoboMaker によってコンテナ内のルートユーザーが使用されます。

コンテナイメージにおいて、`USER` を名前または `UID:GID` のいずれかとして指定できます。コンテナイメージに UID がない場合、デフォルト値 `1000` が使用されます。

コンテナイメージによって `/opt/amazon/robomaker` やそのサブフォルダにデータを保存することはできません。AWS RoboMaker のみがそのディレクトリを使用できます。そのディレクトリを使用すると、シミュレーションが正しく動作しない可能性があります。

以下のランタイム設定はサポートされていません。

	Docker の実行引数	Description
1	`-\-add-host`	カスタムのホスト IP 間マッピング (host: ip) を追加する
2	`-\-attach , -a`	STDIN、STDOUT または STDERR にアタッチする
3	`-\-blkio-weight`	IO (相対重み) を 10 ~ 1000 でブロックするか、または 0 で無効にする (デフォルトは 0)
4	`-\-blkio-weight-device`	IO 重量 (相対デバイス重量) をブロックする
5	`-\-cap-add`	Linux 機能を追加する
6	`-\-cap-drop`	Linux 機能をドロップする

	Docker の実行引数	Description
7	`-\-cgroup-parent`	コンテナのオプションの親 cgroup
8	`-\-cgroupns`	API 1.41+ <https://docs.docker.com/engine/api/ v1.41/>__Cgroup 名前空間。(ホスト|プライベート) 'host' を使用する: Docker ホストの cgroup 名前空間 'private' でコンテナを実行する: コンテナを独自のプライベート cgroup 名前空間で実行する: デーモンでデフォルトの cgroupns モードオプションにより設定された cgroup 名前空間を使用する (デフォルト)
9	`-\-cidfile`	コンテナ ID をファイルに書き込む
10	`-\-cpu-count`	CPU カウント (Windows のみ)
11	`-\-cpu-percent`	CPU パーセント (Windows のみ)
12	`-\-cpu-period`	CPU CFS (完全公平スケジューラ) 期間を制限する
13	`-\-cpu-quota`	CPU CFS (完全公平スケジューラ) クォータを制限する
14	`-\-cpu-rt-period`	API 1.25+ <https://docs.docker.com/engine/api/ v1.25/>__CPU リアルタイム期間 (マイクロ秒) を制限する

	Docker の実行引数	Description
15	`-\-cpu-rt-runtime`	API 1.25+ <https://docs.docker.com/engine/api/ v1.25/>__CPU リアルタイムランタイム (マイクロ秒) を制限する
16	`-\-cpu-shares , -c`	CPU シェア (相対重み)
17	`-\-cpus`	API 1.25+ <https://docs.docker.com/engine/api/ v1.25/>__CPU の数
18	`-\-cpuset-cpus`	実行を許可する CPU (0-3, 0,1)
19	`-\-cpuset-mems`	実行を許可する MEM (0-3, 0,1)
20	`-\-detach , -d`	コンテナをバックグラウンドで実行し、コンテナ ID を印刷する
21	`-\-detach-keys`	コンテナをデタッチするためのキーシーケンスをオーバーライドする
22	`-\-device`	コンテナにホストデバイスを追加する
23	`-\-device-cgroup-rule`	cgroup 許可デバイスリストにルールを追加する
24	`-\-device-read-bps`	デバイスからの読み取りレート (バイト/秒) を制限する
25	`-\-device-read-iops`	デバイスからの読み取りレート (IO/秒) を制限する

	Docker の実行引数	Description
26	`-\-device-write-bps`	デバイスへの書き込みレート (バイト/秒) を制限する
27	`-\-device-write-iops`	デバイスへの書き込みレート (IO /秒) を制限する
28	`-\-disable-content-trust`	イメージの検証をスキップする
29	`-\-dns`	カスタム DNS サーバーを設定する
30	`-\-dns-opt`	DNS オプションを設定する
31	`-\-dns-option`	DNS オプションを設定する
32	`-\-dns-search`	カスタム DNS 検索ドメインを設定する
33	`-\-domainname`	コンテナ NIS ドメイン名
34	`-\-gpus`	API 1.40+ <https://docs.docker.com/engine/api/ v1.40/>__GPU デバイス。コンテナに追加する ('all' はすべての GPU を渡す)
35	`-\-group-add`	参加させるグループを追加する
36	`-\-health-cmd`	ヘルスチェックに対する実行コマンド
37	`-\-health-interval`	チェックの実行間隔 (msm|h) (デフォルトは 0)
38	`-\-health-retries`	不健全性報告を要する連続失敗

	Docker の実行引数	Description
39	-\-health-start-period	API 1.29+ <https://docs.docker.com/engine/api/ v1.29/>__ヘルスチェック再試行のカウントダウンを開始する前にコンテナを初期化するための開始期間 (デフォルトは 0)
40	-\-health-timeout	1 回のチェックを実行できる最大時間 (msm|h) (デフォルトは 0)
41	-\-help	使用状況を印刷する
42	-\-hostname , -h	コンテナホスト名
43	-\-init	API 1.25+ <https://docs.docker.com/engine/api/ v1.25/>__コンテナ内で初期化を実行して信号を転送しプロセスを取得する
44	-\-interactive , -i	STDIN をアタッチしていない場合も開いたままにしておく
45	-\-io-maxbandwidth	システムドライブの IO 帯域幅上限 (Windows のみ)
46	-\-io-maxiops	システムドライブの上限 IOps 制限 (Windows のみ)
47	-\-ip	IPv4 アドレス (172.30.100.104 など)
48	-\-ip6	IPv6 アドレス (2001:db8::33 など)
49	-\-ipc	使用する IPC モード

	Docker の実行引数	Description
50	-\-isolation	コンテナ分離技術
51	-\-kernel-memory	カーネルメモリ限界
52	-\-label , -l	コンテナにメタデータを設定する
53	-\-label-file	ラベルの行区切りファイルを読み込む
54	-\-link	別のコンテナにリンクを追加する
55	-\-link-local-ip	コンテナ IPv4/IPv6 リンクローカルアドレス
56	-\-log-driver	コンテナのロギングドライバー
57	-\-log-opt	ログドライバーのオプション
58	-\-mac-address	コンテナ MAC アドレス (92:d0:c6:0a:29:33 など)
59	-\-memory , -m	メモリ制限
60	-\-memory-reservation	メモリソフト制限
61	-\-memory-swap	メモリとスワップ: '-1' の和に等しいスワップ制限で、無制限スワップを有効にする
62	-\-memory-swappiness	コンテナメモリのスワップを調整する (0 ~ 100)
63	-\-name	コンテナに名前を割り当てる

	Docker の実行引数	Description
64	`-\-net`	コンテナをネットワークに接続する
65	`-\-net-alias`	コンテナのネットワークスコープエイリアスを追加する
66	`-\-network`	コンテナをネットワークに接続する
67	`-\-network-alias`	コンテナのネットワークスコープエイリアスを追加する
68	`-\-no-healthcheck`	コンテナ指定のヘルスチェックを無効にする
69	`-\-oom-kill-disable`	OOM Killer を無効にする
70	`-\-oom-score-adj`	ホストの OOM プリファレンスを調整する (-1000 ~ 1000)
71	`-\-pid`	使用する PID 名前空間
72	`-\-pids-limit`	コンテナ PID 制限を調整する (無制限の場合は -1 に設定)
73	`-\-platform`	API 1.32+ <https://docs.docker.com/engine/api/ v1.32/>__サーバーがマルチプラットフォームに対応している場合にプラットフォームを設定する
74	`-\-privileged`	このコンテナに拡張権限を付与する
75	`-\-publish , -p`	コンテナのポートをホストに向けて発行する

	Docker の実行引数	Description
76	`-\-publish-all , -P`	公開されたすべてのポートをランダムポートに向けて発行する
77	`-\-pull`	実行前にイメージをプルする (「always」(常時行う)「never」(決して行わない))
78	`-\-read-only`	コンテナのルートファイルシステムを読み取り専用としてマウントする
79	`-\-restart`	コンテナの終了時に適用するポリシーを再開始する
80	`-\-rm`	コンテナの終了時にコンテナを自動的に削除する
81	`-\-runtime`	このコンテナに使用するランタイム
82	`-\-security-opt`	セキュリティオプション
83	`-\-shm-size`	/dev/shm のサイズ
84	`-\-sig-proxy`	プロセスへの信号を受信した
85	`-\-stop-timeout`	API 1.25+ <https://docs.docker.com/engine/api/ v1.25/>__コンテナを停止させるタイムアウト (秒)
86	`-\-storage-opt`	コンテナのストレージドライバーオプション
87	`-\-sysctl`	Sysctl オプション

	Docker の実行引数	Description
88	`-\-tmpfs`	tmpfs ディレクトリをマウントする
89	`-\-tty , -t`	疑似 TTY を割り当てる
90	`-\-ulimit`	Ulimit オプション
91	`-\-userns`	使用するユーザー名前空間
92	`-\-uts`	使用する UTS 名前空間
93	`-\-volume , -v`	ボリュームをバインドマウントする
94	`-\-volume-driver`	オプションのコンテナ用ボリュームドライバー
95	`-\-volumes-from`	指定したコンテナからボリュームをマウントする

前述のランタイム設定でシミュレーションジョブを実行すると、AWS RoboMaker で 4XX エラーコードが出てシミュレーションが失敗します。

メタデータ要件

コンテナイメージ:

- Open Container Initiative (OCI) に対応している必要があります。
- X86_64 アーキテクチャ向けに構築されている必要があります。別のアーキテクチャ用に構築されている場合、AWS RoboMaker で 4XX エラーコードが出てシミュレーションが失敗します。
- 未圧縮サイズが 40 GB 以下である必要があります。コンテナイメージの未圧縮サイズが 40 GB より大きい場合、AWS RoboMaker で 4XX エラーコードが出てシミュレーションが失敗します。
- スキーマバージョン 2 互換の V2 イメージマニフェストが必要です。
- Linux をベースにしたベースイメージを使用する必要があります。Linux をベースにしたベースイメージを使用しない場合、AWS RoboMaker で 4XX エラーコードが出てシミュレーションが失敗します。

- 相互に互換性のある開発環境とオペレーティングシステムを使用する必要があります。以下は、相互に互換性がある開発環境とオペレーティングシステムの組み合わせの例です。
  - ロボットオペレーティングシステム (ROS) Melodic — ubuntu: bionic
  - ロボットオペレーティングシステム (ROS) 2 Foxy — ubuntu: focal

  相互に互換性があるロボットフレームワークとオペレーティングシステムを組み合わせて使用しない場合、シミュレーションにおいて予期しない動作が発生する可能性があります。

バイナリ要件

コンテナイメージのバイナリ要件は次のとおりです。

GUI ストリーミングをサポートするには、次のバイナリをインストールしてソースとすることをお勧めします。

- `devilspie`

コンテナイメージでは、実行可能ファイルに絶対パスを使用することをお勧めします。また、コンテナ内の実行可能ファイルの正常な実行も推奨されます。実行可能ファイルへのパスが見つからない場合は、シミュレーションが失敗します。

GPU 要件

コンテナイメージ:

- アプリケーションで OpenGL を使用している場合は、glvnd をインストールする必要があります。
- アプリケーションで CUDA を使用している場合は、NVIDIA CUDA 11.2 以下が必要です。
- アプリケーションで OpenGL を使用している場合は、OpenGL バージョン 4.6 以下が必要です。
- アプリケーションで Vulkan API を使用している場合は、Vulkan バージョン 1.2 以下が必要です。
- アプリケーションで OpenGL を使用している場合は、OpenGL バージョン 1.2 以下が必要です。

> **注意**
>
> AWS RoboMaker はオフスクリーンレンダリングのみ Vulkan をサポートしており、GUI ディスプレイでは動作しません。したがって、Vulkan を使用している場合は streamUI を `false` に設定すべきです。

GPU イメージの作成方法の詳細については、「GPU アプリケーションを実行するためのイメージの作成」を参照してください。

Dockerfile および環境変数の要件

コンテナイメージにはソーシング用のエントリポイントスクリプトが必要です。AWS RoboMaker がエントリポイントスクリプトを実行できるように、エントリポイントスクリプトの最終行として `exec "${@:1}"` が必要です。エントリポイントスクリプトを実行すると、`roslaunch` *`package-name`* コマンド、*`######`*コマンドを使用してコンテナを実行できるようになります。

コンテナイメージでは Dockerfile にある `VOLUME` は使用できません。`VOLUME` が Dockerfileに入っている場合、`4XX` エラーコードが出てシミュレーションが失敗します。

Dockerfile 内の `EXPOSE` キーワードは AWS RoboMaker によって無視されます。`EXPOSE` キーワードによって公開されているポートは、システムによって自動的に公開されません。シミュレーションでポートを公開したい場合は、AWS RoboMaker ポート転送設定を使用できます。

AWS RoboMaker は次の環境変数を使用します。AWS でシミュレーションを実行した場合、AWS RoboMaker により、次の環境変数に対して指定した値がすべて上書きされます。

- `ROBOMAKER*`
- `DCV_VIRTUAL_SESSION`
- `XDG_SESSION_ID`
- `DCV_SESSION_ID`
- `XDG_SESSION_TYPE`
- `XDG_RUNTIME_DIR`
- `SHLVL`
- `XAUTHORITY`

Dockerfile 内にある `CMD` は指定できません。これを実行した場合、AWS RoboMaker はシミュレーション `launchConfig` のコマンドで上書きします。

ネットワーク、マウント、セキュリティ、およびユーザーの要件

コンテナが ROS を実行していて、ロボットアプリケーションとシミュレーションアプリケーション間の通信が必要な場合は、以下のロボットフレームワークを設定する必要があります。

- ROS Master
- Gazebo Master
- ROS IP

コンテナ内の /etc/resolv.conf ファイルはカスタマイズできません。AWS RoboMaker は独自のファイルでファイルを上書きします。

AWS で Dockerfile を実行している場合、イメージをマウントすることはできません。Dockerfile 内の Mount を指定した場合、AWS RoboMaker は 4XX エラーコードが出てシミュレーションが失敗します。

デフォルトの Docker seccomp プロファイルによってブロックされているシステムコールは、コンテナイメージによって使用されません。ブロックされたシステムコールについては、「Seccomp セキュリティプロファイル」を参照してください。

イメージを実行するユーザーを指定するために、Dockerfile の USER キーワードを指定することができます。ユーザーを指定しない場合、AWS RoboMaker によってコンテナ内のルートユーザーが使用されます。

コンテナイメージにおいて、USER を名前または UID:GID のいずれかとして指定できます。コンテナイメージに UID がない場合、デフォルト値 1000 が使用されます。

その他の要件

コンテナイメージによって /opt/amazon/robomaker やそのサブフォルダにデータを保存することはできません。AWS RoboMaker のみがそのディレクトリを使用できます。そのディレクトリを使用すると、シミュレーションが正しく動作しない可能性があります。

以下のランタイム設定はサポートされていません。

	Docker の実行引数	Description
1	--add-host	カスタムのホスト IP 間マッピング (host: ip) を追加する
2	--attach , -a	STDIN、STDOUT または STDERR にアタッチする
3	--blkio-weight	IO (相対重み) を 10 ~ 1000 でブロックするか、または 0 で無効にする (デフォルトは 0)
4	--blkio-weight-device	IO 重量 (相対デバイス重量) をブロックする

	Docker の実行引数	Description
5	`--cap-add`	Linux 機能を追加する
6	`--cap-drop`	Linux 機能をドロップする
7	`--cgroup-parent`	コンテナのオプションの親 cgroup
8	`--cgroupns`	API 1.41+ <https://docs.docker.com/engine/api/ v1.41/>__Cgroup 名前空間。(ホスト|プライベート) 'host' を使用する: Docker ホストの cgroup 名前空間 'private' でコンテナを実行する: コンテナを独自のプライベート cgroup 名前空間で実行する: デーモンでデフォルトの cgroupns モードオプションにより設定された cgroup 名前空間を使用する (デフォルト)
9	`--cidfile`	コンテナ ID をファイルに書き込む
10	`--cpu-count`	CPU カウント (Windows のみ)
11	`--cpu-percent`	CPU パーセント (Windows のみ)
12	`--cpu-period`	CPU CFS (完全公平スケジューラ) 期間を制限する
13	`--cpu-quota`	CPU CFS (完全公平スケジューラ) クォータを制限する

	Docker の実行引数	Description
14	`--cpu-rt-period`	API 1.25+ <https://docs.docker.com/engine/api/ v1.25/>__CPU リアルタイム期間 (マイクロ秒) を制限する
15	`--cpu-rt-runtime`	API 1.25+ <https://docs.docker.com/engine/api/ v1.25/>__CPU リアルタイムランタイム (マイクロ秒) を制限する
16	`--cpu-shares , -c`	CPU シェア (相対重み)
17	`--cpus`	API 1.25+ <https://docs.docker.com/engine/api/ v1.25/>__CPU の数
18	`--cpuset-cpus`	実行を許可する CPU (0-3, 0,1)
19	`--cpuset-mems`	実行を許可する MEM (0-3, 0,1)
20	`--detach , -d`	コンテナをバックグラウンドで実行し、コンテナ ID を印刷する
21	`--detach-keys`	コンテナをデタッチするためのキーシーケンスをオーバーライドする
22	`--device`	コンテナにホストデバイスを追加する
23	`--device-cgroup-rule`	cgroup 許可デバイスリストにルールを追加する

	Docker の実行引数	Description
24	`--device-read-bps`	デバイスからの読み取りレート (バイト/秒) を制限する
25	`--device-read-iops`	デバイスからの読み取りレート (IO/秒) を制限する
26	`--device-write-bps`	デバイスへの書き込みレート (バイト/秒) を制限する
27	`--device-write-iops`	デバイスへの書き込みレート (IO /秒) を制限する
28	`--disable-content-trust`	イメージの検証をスキップする
29	`--dns`	カスタム DNS サーバーを設定する
30	`--dns-opt`	DNS オプションを設定する
31	`--dns-option`	DNS オプションを設定する
32	`--dns-search`	カスタム DNS 検索ドメインを設定する
33	`--domainname`	コンテナ NIS ドメイン名
34	`--gpus`	API 1.40+ <https://docs.docker.com/engine/api/ v1.40/>__GPU デバイス。コンテナに追加する ('all' はすべての GPU を渡す)
35	`--group-add`	参加させるグループを追加する
36	`--health-cmd`	ヘルスチェックのために実行する

	Docker の実行引数	Description
37	`--health-interval`	チェックの実行間隔 (msm|h) (デフォルトは 0)
38	`--health-retries`	不健全性報告を要する連続失敗
39	`--health-start-period`	API 1.29+ <https://docs.docker.com/engine/api/ v1.29/>__ヘルスチェック再試行のカウントダウンを開始する前にコンテナを初期化するための開始期間 (デフォルトは 0)
40	`--health-timeout`	1 回のチェックを実行できる最大時間 (msm|h) (デフォルトは 0)
41	`--help`	使用状況を印刷する
42	`--hostname , -h`	コンテナホスト名
43	`--init`	API 1.25+ <https://docs.docker.com/engine/api/ v1.25/>__コンテナ内で初期化を実行して信号を転送しプロセスを取得する
44	`--interactive , -i`	STDIN をアタッチしていない場合も開いたままにしておく
45	`--io-maxbandwidth`	システムドライブの IO 帯域幅上限 (Windows のみ)
46	`--io-maxiops`	システムドライブの上限 IOps 制限 (Windows のみ)

	Docker の実行引数	Description
47	`--ip`	IPv4 アドレス (172.30.100.104 など)
48	`--ip6`	IPv6 アドレス (2001:db8::33 など)
49	`--ipc`	使用する IPC モード
50	`--isolation`	コンテナ分離技術
51	`--kernel-memory`	カーネルメモリ限界
52	`--label , -l`	コンテナにメタデータを設定する
53	`--label-file`	ラベルの行区切りファイルを読み込む
54	`--link`	別のコンテナにリンクを追加する
55	`--link-local-ip`	コンテナ IPv4/IPv6 リンクローカルアドレス
56	`--log-driver`	コンテナのロギングドライバー
57	`--log-opt`	ログドライバーのオプション
58	`--mac-address`	コンテナ MAC アドレス (92:d0:c6:0a:29:33 など)
59	`--memory , -m`	メモリ制限
60	`--memory-reservation`	メモリソフト制限

	Docker の実行引数	Description
61	`--memory-swap`	メモリとスワップ: '-1' の和に等しいスワップ制限で、無制限スワップを有効にする
62	`--memory-swappiness`	コンテナメモリのスワップを調整する (0 ~ 100)
63	`--name`	コンテナに名前を割り当てる
64	`--net`	コンテナをネットワークに接続する
65	`--net-alias`	コンテナのネットワークスコープエイリアスを追加する
66	`--network`	コンテナをネットワークに接続する
67	`--network-alias`	コンテナのネットワークスコープエイリアスを追加する
68	`--no-healthcheck`	コンテナ指定のヘルスチェックを無効にする
69	`--oom-kill-disable`	OOM Killer を無効にする
70	`--oom-score-adj`	ホストの OOM プリファレンスを調整する (-1000 ~ 1000)
71	`--pid`	使用する PID 名前空間
72	`--pids-limit`	コンテナ PID 制限を調整する (無制限の場合は -1 に設定)

	Docker の実行引数	Description
73	`--platform`	API 1.32+ <https://docs.docker.com/engine/api/ v1.32/>__サーバーがマルチプラットフォームに対応している場合にプラットフォームを設定する
74	`--privileged`	このコンテナに拡張権限を付与する
75	`--publish , -p`	コンテナのポートをホストに向けて発行する
76	`--publish-all , -P`	公開されたすべてのポートをランダムポートに向けて発行する
77	`--pull`	実行前にイメージをプルする (「always」(常時行う)「never」(決して行わない))
78	`--read-only`	コンテナのルートファイルシステムを読み取り専用としてマウントする
79	`--restart`	コンテナの終了時に適用するポリシーを再開始する
80	`--rm`	コンテナの終了時にコンテナを自動的に削除する
81	`--runtime`	このコンテナに使用するランタイム
82	`--security-opt`	セキュリティオプション
83	`--shm-size`	/dev/shm のサイズ

	Docker の実行引数	Description
84	`--sig-proxy`	プロセスへの信号を受信した
85	`--stop-timeout`	API 1.25+ <https://docs.docker.com/engine/api/ v1.25/>__コンテナを停止させるタイムアウト (秒)
86	`--storage-opt`	コンテナのストレージドライバーオプション
87	`--sysctl`	Sysctl オプション
88	`--tmpfs`	tmpfs ディレクトリをマウントする
89	`--tty , -t`	疑似 TTY を割り当てる
90	`--ulimit`	Ulimit オプション
91	`--userns`	使用するユーザー名前空間
92	`--uts`	使用する UTS 名前空間
93	`--volume , -v`	ボリュームをバインドマウントする
94	`--volume-driver`	オプションのコンテナ用ボリュームドライバー
95	`--volumes-from`	指定したコンテナからボリュームをマウントする

前述のランタイム設定でシミュレーションジョブを実行すると、AWS RoboMaker で 4XX エラーコードが出てシミュレーションが失敗します。

GPU アプリケーションを実行するためのイメージの作成

AWS RoboMaker GPU シミュレーションジョブは CUDA、OpenGL、OpenCL、Vulkan API アクセスに対応しています。したがって、これらの API を使用するアプリケーションでは、対応するドライバーがイメージにインストールされている必要があります。

> **Note**
>
> OpenGL API の取得には、Nvidia のベースイメージを使用することをお勧めします。チュートリアルで使用されている Dockerfile の例は、OpenGL をサポートしている `nvidia/opengl:1.0-glvnd-runtime-ubuntu20.04` のみを対象としています。CUDA、Vulkan、OpenCL をサポートするコンテナイメージを見つけるには、Nvidia のドキュメントを参照してください。

GPU レンダリングで DCV ディスプレイを使用するには、`nice-dcv-gl` をインストールする必要があります。X0 は GPU と通信するシステムの Xorg プロセスであることに注意してください。また、X1 と X2 は XDCV プロセスです。X1 または X2 で OpenGL アプリケーションを起動すると、`nice-dcv-gl` により、X0 (GPU を使用できる) で呼び出しがリダイレクトされ、レンダリングが実行されます。

`nice-dcv-gl` をインストールするには、アーカイブをダウンロードして解凍し、DCV の公開ドキュメントに続く `nice-dcv-gl` パッケージをインストールします。「Linux に RLM サーバーをインストールする」を参照してください。

次の例は、ubuntu18.04 ベースイメージに nice-dcv-gl_2021.2 をインストールする Dockerfile を示しています。

```
FROM nvidia/opengl:1.0-glvnd-runtime-ubuntu20.04

ENV DEBIAN_FRONTEND="noninteractive"

RUN apt-get update && apt-get install -y --no-install-recommends \
        ca-certificates \
        gnupg2 \
        wget

RUN wget https://d1uj6qtbmh3dt5.cloudfront.net/NICE-GPG-KEY && gpg --import NICE-GPG-
KEY && \
```

```
        wget https://d1uj6qtbmh3dt5.cloudfront.net/2021.2/Servers/nice-
dcv-2021.2-11048-ubuntu1804-x86_64.tgz && \
        tar xvzf nice-dcv-2021.2-11048-ubuntu1804-x86_64.tgz && \
        cd nice-dcv-2021.2-11048-ubuntu1804-x86_64 && \
        apt install -y ./nice-dcv-gl_2021.2.944-1_amd64.ubuntu1804.deb
```

GPU アプリケーションのビルド手順の詳細については、「ROS2 Foxy と Gazebo 11 を使用した GPU サンプルアプリケーションの実行」を参照してください。

イメージを作成して Hello World サンプルアプリケーションを実行する

弊社が提供する Hello World サンプルアプリケーションを使用すれば、シミュレーションアプリケーションとロボットアプリケーションの作成方法および実行方法が分かります。次のセクションでは、以下の開発環境に対するイメージの作成方法と実行方法について説明します。

- ROS Melodic と Gazebo 9
- ROS 2 Foxy と Gazebo 11

ROS は、ロボットアプリケーションに使用されるロボットオペレーティングシステムです。Gazebo は、シミュレーションアプリケーション用のオペレーティングシステムです。 AWS RoboMaker は、コンテナイメージの使用と検証チェックに両方のソフトウェアスイートを使用します。

チュートリアルでは、AWS RoboMaker コンテナイメージを使用して Hello World のロボットアプリケーションとシミュレーションアプリケーションをセットアップする方法について説明します。Hello World アプリケーションは、AWS RoboMaker の操作方法の理解に役立つサンプルアプリケーションです。

各チュートリアルで、ロボットアプリケーションとシミュレーションアプリケーションのイメージを作成します。イメージをローカルで実行してその動作をテストすることができます。シミュレーションが正常に動作すれば、それらを Amazon ECR にプッシュし、クラウドでシミュレーションジョブを実行できます。シミュレーションジョブの詳細については、「AWS RoboMaker によるシミュレーション」を参照してください。

ROS 2 Foxy と Gazebo 11 を使用したサンプルアプリケーションの実行

以下のチュートリアルでは、Hello World ロボットアプリケーションおよびシミュレーションアプリケーションを作成して実行することにより、コンテナイメージを使用して ROS 2 Foxy および Gazebo 11 で開発する方法について説明します。本書で説明されているコマンドを実行することで、サンプルアプリケーションを動作させることができます。

このチュートリアルでは、3 つのコンテナイメージを作成して使用します。次に、このサンプルアプリケーションで使用するディレクトリ構造を示します。

```
### HelloWorldSampleAppROS2FoxyGazebo11 // Base Image
#   ### Dockerfile
### HelloWorldSampleAppROS2FoxyGazebo11RobotApp // Image for Robot App
#   ### Dockerfile
#   ### robot-entrypoint.sh
### HelloWorldSampleAppROS2FoxyGazebo11SimApp // Image for Simulation App
#   ### Dockerfile
#   ### simulation-entrypoint.sh
```

各 Dockerfile には、各イメージのビルドに必要な指示があります。

- ベースイメージの Dockerfile には、ROS と Gazebo を設定するためのコマンドがあります。
- ロボットアプリケーションの Dockerfile には、Hello World ロボットアプリケーションを設定するためのコマンドがあります。
- シミュレーションアプリケーションの Dockerfile には、Hello World シミュレーションアプリケーションを設定するためのコマンドがあります。

ロボットアプリケーションとシミュレーションアプリケーションの両方に、エントリポイントスクリプトがあります。これらのスクリプトは、それぞれのアプリケーションの環境をソースにします。これらのスクリプトによりパスが設定され、そのパスを使って、コマンドを実行してロボットアプリケーションとシミュレーションアプリケーションを起動します。

ベースイメージの作成

ベースイメージを作成するには、Dockerfile に環境作成用コマンドを保存します。次に Dockerfile を作成します。

- 次のコマンドを Dockerfile に保存します。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
FROM ros:foxy

ENV DEBIAN_FRONTEND noninteractive

RUN apt-get clean
RUN apt-get update && apt-get install -y \
```

```
    lsb  \
    unzip \
    wget \
    curl \
    sudo \
    python3-vcstool \
    python3-rosinstall \
    python3-colcon-common-extensions \
    ros-foxy-rviz2 \
    ros-foxy-rqt \
    ros-foxy-rqt-common-plugins \
    devilspie \
    xfce4-terminal

RUN wget https://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add -; \
    sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu-stable
 `lsb_release -cs` main" > /etc/apt/sources.list.d/gazebo-stable.list'
RUN apt-get update && apt-get install -y gazebo11

ENV QT_X11_NO_MITSHM=1

ARG USERNAME=robomaker
RUN groupadd $USERNAME
RUN useradd -ms /bin/bash -g $USERNAME $USERNAME
RUN sh -c 'echo "$USERNAME ALL=(root) NOPASSWD:ALL" >> /etc/sudoers'
USER $USERNAME

RUN sh -c 'cd /home/$USERNAME'

# Download and build our Robot and Simulation application
RUN sh -c 'mkdir -p /home/robomaker/workspace'
RUN sh -c 'cd /home/robomaker/workspace && wget https://github.com/aws-
robotics/aws-robomaker-sample-application-helloworld/archive/3527834.zip
 && unzip 3527834.zip && mv aws-robomaker-sample-application-
helloworld-3527834771373beff0ed3630c13479567db4149e aws-robomaker-sample-
application-helloworld-ros2'
RUN sh -c 'cd /home/robomaker/workspace/aws-robomaker-sample-application-
helloworld-ros2'

RUN sudo rosdep fix-permissions
RUN rosdep update
```

Dockerfile を作成したら、ターミナルで以下のコマンドを使用してビルドします。

```
cd ../HelloWorldSampleAppROS2FoxyGazebo11
docker build -t helloworldsampleappros2foxygazebo11:latest .
```

ベースイメージをビルドすると ROS 2 Foxy と Gazebo 11 がインストールされます。アプリケーションを正常に実行するには、両方のライブラリをインストールする必要があります。

ロボットアプリケーション用イメージの作成

ベースイメージを作成したら、ロボットアプリケーションのイメージを作成できます。以下のスクリプトを Dockerfile に保存してビルドします。このスクリプトにより、Hello World ロボットアプリケーションがダウンロードされて設定されます。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
FROM helloworldsampleappros2foxygazebo11:latest

# Build the Robot application
RUN cd /home/robomaker/workspace/aws-robomaker-sample-application-helloworld-ros2/
robot_ws && \
 /bin/bash -c "source /opt/ros/foxy/setup.bash && vcs import < .rosinstall && rosdep
 install --rosdistro foxy --from-paths src --ignore-src -r -y && colcon build"


COPY robot-entrypoint.sh /home/robomaker/robot-entrypoint.sh
RUN sh -c 'sudo chmod +x /home/robomaker/robot-entrypoint.sh'
RUN sh -c 'sudo chown robomaker:robomaker /home/robomaker/robot-entrypoint.sh'

CMD ros2 launch hello_world_robot rotate.launch.py
ENTRYPOINT [ "/home/robomaker/robot-entrypoint.sh" ]
```

以下のコマンドにより、Dockerfile からロボットアプリケーションのイメージが作成されます。

```
cd HelloWorldSampleAppROS2FoxyGazebo11RobotApp/
HelloWorldSampleAppROS2FoxyGazebo11RobotApp
docker build -t helloworldsampleappros2foxygazebo11robotapp:latest .
```

`robot-entrypoint.sh` として保存できるスクリプトの内容を以下に示します。このスクリプトはロボットアプリケーションの環境をソースにします。

```
#!/bin/bash

if [ ! -z $GAZEBO_MASTER_URI ]; then
        tmp_GAZEBO_MASTER_URI=$GAZEBO_MASTER_URI
fi

cd /home/robomaker/workspace/aws-robomaker-sample-application-helloworld-ros2/robot_ws
source /opt/ros/foxy/setup.bash
source /usr/share/gazebo-11/setup.sh
source ./install/setup.sh

if [ ! -z $tmp_GAZEBO_MASTER_URI ]; then
        export GAZEBO_MASTER_URI=$tmp_GAZEBO_MASTER_URI
        unset tmp_GAZEBO_MASTER_URI
fi

printenv

exec "${@:1}"
```

シミュレーションアプリケーション用イメージの作成

ベースイメージとロボットアプリケーション用イメージを作成すれば、シミュレーションアプリケーションのイメージを作成できます。以下のスクリプトを Dockerfile に保存してビルドします。このスクリプトにより、Hello World ロボットアプリケーションがダウンロードされて設定されます。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
FROM helloworldsampleappros2foxygazebo11:latest

# Build the Simulation application
RUN cd /home/robomaker/workspace/aws-robomaker-sample-application-helloworld-ros2/
simulation_ws && \
 /bin/bash -c "source /opt/ros/foxy/setup.bash && vcs import < .rosinstall && rosdep
 install --rosdistro foxy --from-paths src --ignore-src -r -y && colcon build"


COPY simulation-entrypoint.sh /home/robomaker/simulation-entrypoint.sh

RUN sh -c 'sudo chmod +x /home/robomaker/simulation-entrypoint.sh'
RUN sh -c 'sudo chown robomaker:robomaker /home/robomaker/simulation-entrypoint.sh'

CMD ros2 launch hello_world_simulation empty_world.launch.py
```

```
ENTRYPOINT [ "/home/robomaker/simulation-entrypoint.sh" ]
```

以下のコマンドを実行してイメージを作成します。

```
cd HelloWorldSampleAppROS2FoxyGazebo11SimApp/HelloWorldSampleAppROS2FoxyGazebo11SimApp
docker build -t helloworldsampleappros2foxygazebo11simapp:latest .
```

simulation-entrypoint.sh として保存できるスクリプトの内容を以下に示します。このスクリプトはシミュレーションアプリケーションの環境をソースにします。

```
#!/bin/bash

if [ ! -z $GAZEBO_MASTER_URI ]; then
        tmp_GAZEBO_MASTER_URI=$GAZEBO_MASTER_URI
fi

cd /home/robomaker/workspace/aws-robomaker-sample-application-helloworld-ros2/
simulation_ws
source /opt/ros/foxy/setup.bash
source /usr/share/gazebo-11/setup.sh
source ./install/setup.sh

if [ ! -z $tmp_GAZEBO_MASTER_URI ]; then
        export GAZEBO_MASTER_URI=$tmp_GAZEBO_MASTER_URI
        unset tmp_GAZEBO_MASTER_URI
fi

printenv

exec "${@:1}"
```

アプリケーションを実行して Amazon ECR にプッシュする

イメージを作成したら、ローカルの Linux 環境で適切に動作していることを確認します。イメージが実行されていることを確認した後、Docker イメージを Amazon ECR にプッシュしてシミュレーションジョブを作成できます。

次のコマンドを使用すると、ローカル Linux 環境で Hello World アプリケーションを実行できます。

```
docker run -it -e DISPLAY -v /tmp/.X11-unix/:/tmp/.X11-unix/ --name robot_app \
```

```
-u robomaker -e ROBOMAKER_GAZEBO_MASTER_URI=http://localhost:5555 \
-e ROBOMAKER_ROS_MASTER_URI=http://localhost:11311 \
helloworldsampleappros2foxygazebo11robotapp:latest
```

```
docker run -it -e DISPLAY -v /tmp/.X11-unix/:/tmp/.X11-unix/ --name sim_app \
-u robomaker -e ROBOMAKER_GAZEBO_MASTER_URI=http://localhost:5555 \
-e ROBOMAKER_ROS_MASTER_URI=http://localhost:11311 \
helloworldsampleappros2foxygazebo11simapp:latest
```

ロボットアプリケーションコンテナとシミュレーションアプリケーションコンテナを実行すると、Gazebo GUI ツールを使用してシミュレーションを視覚化できます。以下のコマンドを使用して次の作業を行います。

1. シミュレーションアプリケーションを実行しているコンテナに接続します。
2. Gazebo グラフィカルユーザーインターフェイス (GUI) を実行することでアプリケーションを視覚化します。

```
# Enable access to X server to launch Gazebo from docker container
$ xhost +

# Check that the robot_app and sim_app containers are running. The command should list
 both containers
$ docker container ls

# Connect to the sim app container
$ docker exec -it sim_app bash

# Launch Gazebo from within the container
$ /home/robomaker/simulation-entrypoint.sh ros2 launch gazebo_ros gzclient.launch.py
```

イメージにタグを追加できます。以下のコマンドによりイメージにタグを付けることができます。

```
docker tag helloworldsampleappros2foxygazebo11robotapp:latest accountID.dkr.ecr.us-
west-2.amazonaws.com/helloworldsampleappros2foxygazebo11robotapp:latest
```

```
docker tag helloworldsampleappros2foxygazebo11simapp:latest accountID.dkr.ecr.us-
west-2.amazonaws.com/helloworldsampleappros2foxygazebo11simapp:latest
```

アプリケーションが正常に動作していることを確認したら、以下のコマンドを使用して Amazon ECR にプッシュできます。

```
aws ecr get-login-password --region us-west-2 | docker login --username AWS --password-
stdin accountID.dkr.ecr.us-west-2.amazonaws.com
docker push accountID.dkr.ecr.us-west-2.amazonaws.com/
helloworldsampleappros2foxygazebo11robotapp:latest
docker push accountID.dkr.ecr.us-west-2.amazonaws.com/
helloworldsampleappros2foxygazebo11simapp:latest
```

その後、イメージに対してシミュレーションジョブを実行できます。シミュレーションジョブの詳細については、「AWS RoboMaker によるシミュレーション」を参照してください。

ROS Melodic と Gazebo 9 でサンプルアプリケーションを実行する

以下のチュートリアルでは、Hello World ロボットアプリケーションおよびシミュレーションアプリケーションを作成して実行することにより、コンテナイメージを使用して ROS および Gazebo 9 で開発する方法について説明します。本書で説明されているコマンドを実行することで、サンプルアプリケーションを動作させることができます。

このチュートリアルでは、3 つのコンテナイメージを作成して使用します。次に、このサンプルアプリケーションで使用するディレクトリ構造を示します。

```
### HelloWorldSampleAppROSMelodicGazebo9 // Base Image
#   ### Dockerfile
### HelloWorldSampleAppROSMelodicGazebo9RobotApp // Image for Robot App
#   ### Dockerfile
#   ### robot-entrypoint.sh
### HelloWorldSampleAppROSMelodicGazebo9SimApp // Image for Simulation App
#   ### Dockerfile
#   ### simulation-entrypoint.sh
```

各 Dockerfile には、各イメージのビルドに必要な指示があります。

- ベースイメージの Dockerfile には、ROS と Gazebo を設定するためのコマンドがあります。

- ロボットアプリケーションの Dockerfile には、Hello World ロボットアプリケーションを設定するためのコマンドがあります。
- シミュレーションアプリケーションの Dockerfile には、Hello World シミュレーションアプリケーションを設定するためのコマンドがあります。

ロボットアプリケーションとシミュレーションアプリケーションの両方に、エントリポイントスクリプトがあります。これらのスクリプトは、それぞれのアプリケーションの環境をソースにします。これらのスクリプトによりパスが設定され、そのパスを使って、コマンドを実行してロボットアプリケーションとシミュレーションアプリケーションを起動します。

ベースイメージの作成

ベースイメージを作成するには、Dockerfile に環境作成用の例のコマンドを保存します。次に Dockerfile を作成します。

1. 次のコマンドを Dockerfile に保存します。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
FROM ros:melodic

ENV DEBIAN_FRONTEND noninteractive

RUN apt-get clean
RUN apt-get update && apt-get install -y \
    lsb  \
    unzip \
    wget \
    curl \
    sudo \
    python-vcstool \
    python-rosinstall \
    python3-colcon-common-extensions \
    ros-melodic-rviz \
    ros-melodic-rqt \
    ros-melodic-rqt-common-plugins \
    devilspie \
    xfce4-terminal \
    ros-melodic-gazebo-ros-pkgs \
    ros-melodic-gazebo-ros-control \
    ros-melodic-turtlebot3
```

```
ENV QT_X11_NO_MITSHM=1

ARG USERNAME=robomaker
RUN groupadd $USERNAME
RUN useradd -ms /bin/bash -g $USERNAME $USERNAME
RUN sh -c 'echo "$USERNAME ALL=(root) NOPASSWD:ALL" >> /etc/sudoers'
USER $USERNAME

RUN sh -c 'cd /home/$USERNAME'

# Download and build our Robot and Simulation application
RUN sh -c 'mkdir -p /home/robomaker/workspace'
RUN sh -c 'cd /home/robomaker/workspace && wget https://github.com/aws-robotics/
aws-robomaker-sample-application-helloworld/archive/ros1.zip && unzip ros1.zip'
RUN sh -c 'cd /home/robomaker/workspace/aws-robomaker-sample-application-
helloworld-ros1'

RUN sudo rosdep fix-permissions
RUN rosdep update
```

2. Dockerfile を作成したら、ターミナルで以下のコマンドを使用してビルドします。

```
cd ../HelloWorldSampleAppROSMelodicGazebo9
docker build -t helloworldsampleapprosmelodicgazebo9:latest .
```

ベースイメージをビルドすると ROS Melodic と Gazebo 9 がインストールされます。アプリケーションを正常に実行するには、両方のライブラリをインストールする必要があります。

ロボットアプリケーション用イメージの作成

ベースイメージを作成したら、ロボットアプリケーションのイメージを作成できます。

1. 以下のスクリプトを Dockerfile に保存してビルドします。このスクリプトにより、Hello World ロボットアプリケーションがダウンロードされて設定されます。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
FROM helloworldsampleapprosmelodicgazebo9:latest

# Build the Robot application
```

```
RUN cd /home/robomaker/workspace/aws-robomaker-sample-application-helloworld-ros1/
robot_ws && \
 /bin/bash -c "source /opt/ros/melodic/setup.bash && vcs import < .rosinstall &&
 rosdep install --rosdistro melodic --from-paths src --ignore-src -r -y && colcon
 build"

COPY robot-entrypoint.sh /home/robomaker/robot-entrypoint.sh

RUN sh -c 'sudo chmod +x /home/robomaker/robot-entrypoint.sh'
RUN sh -c 'sudo chown robomaker:robomaker /home/robomaker/robot-entrypoint.sh'

CMD roslaunch hello_world_robot rotate.launch
ENTRYPOINT [ "/home/robomaker/robot-entrypoint.sh" ]
```

2. 以下のコマンドを使用して、Dockerfile から ロボットアプリケーションのイメージが作成されます。

```
cd HelloWorldSampleAppROSMelodicGazebo9RobotApp/
HelloWorldSampleAppROSMelodicGazebo9RobotApp
docker build -t helloworldsampleapprosmelodicgazebo9robotapp:latest image/.
```

3. robot-entrypoint.sh として保存できるスクリプトの内容を以下に示します。このスクリプトは ロボットアプリケーションの環境をソースにします。

```
#!/bin/bash

if [ ! -z $GAZEBO_MASTER_URI ]; then
        tmp_GAZEBO_MASTER_URI=$GAZEBO_MASTER_URI
fi

cd /home/robomaker/workspace/aws-robomaker-sample-application-helloworld-ros1/
robot_ws
source /opt/ros/melodic/setup.bash
source /usr/share/gazebo-9/setup.sh
source ./install/setup.sh

if [ ! -z $tmp_GAZEBO_MASTER_URI ]; then
        export GAZEBO_MASTER_URI=$tmp_GAZEBO_MASTER_URI
        unset tmp_GAZEBO_MASTER_URI
fi

printenv
```

```
exec "${@:1}"
```

シミュレーションアプリケーション用イメージの作成

ベースイメージとロボットアプリケーション用イメージを作成すれば、シミュレーションアプリケーションのイメージを作成できます。

1. 以下のスクリプトを Dockerfile に保存してビルドします。このスクリプトにより、Hello World ロボットアプリケーションがダウンロードされて設定されます。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
FROM helloworldsampleapprosmelodicgazebo9:latest

# Build the Simulation application
RUN cd /home/robomaker/workspace/aws-robomaker-sample-application-helloworld-ros1/
simulation_ws && \
    /bin/bash -c "source /opt/ros/melodic/setup.bash && vcs import < .rosinstall &&
 rosdep install --rosdistro melodic --from-paths src --ignore-src -r -y && colcon
 build"

COPY simulation-entrypoint.sh /home/robomaker/simulation-entrypoint.sh

RUN sh -c 'sudo chmod +x /home/robomaker/simulation-entrypoint.sh'
RUN sh -c 'sudo chown robomaker:robomaker /home/robomaker/simulation-entrypoint.sh'

CMD roslaunch hello_world_simulation empty_world.launch
ENTRYPOINT [ "/home/robomaker/simulation-entrypoint.sh" ]
```

2. 次の simulation-entrypoint.sh スクリプトを保存してください。このスクリプトはシミュレーションアプリケーションの環境をソースにします。

```
#!/bin/bash

if [ ! -z $GAZEBO_MASTER_URI ]; then
        tmp_GAZEBO_MASTER_URI=$GAZEBO_MASTER_URI
fi

cd /home/robomaker/workspace/aws-robomaker-sample-application-helloworld-ros1/
simulation_ws
source /opt/ros/melodic/setup.bash
source /usr/share/gazebo-9/setup.sh
```

```
source ./install/setup.sh

if [ ! -z $tmp_GAZEBO_MASTER_URI ]; then
        export GAZEBO_MASTER_URI=$tmp_GAZEBO_MASTER_URI
        unset tmp_GAZEBO_MASTER_URI
fi

printenv

exec "${@:1}"
```

アプリケーションを実行して ECR にプッシュする

イメージを作成したら、ローカルの Linux 環境で適切に動作していることを確認します。Docker イメージが実行されていることを確認した後、そのイメージを Amazon ECR にプッシュしてシミュレーションジョブを作成することができます。

1. 次のコマンドを使用して、ローカル Linux 環境で Hello World アプリケーションを実行します。

```
docker run -it -e DISPLAY -v /tmp/.X11-unix/:/tmp/.X11-unix/ \
-u robomaker -e ROBOMAKER_GAZEBO_MASTER_URI=http://localhost:5555 \
-e ROBOMAKER_ROS_MASTER_URI=http://localhost:11311 \
helloworldsampleapprosmelodicgazebo9robotapp:latest
```

```
docker run -it -e DISPLAY -v /tmp/.X11-unix/:/tmp/.X11-unix/ \
-u robomaker -e ROBOMAKER_GAZEBO_MASTER_URI=http://localhost:5555 \
-e ROBOMAKER_ROS_MASTER_URI=http://localhost:11311 \
helloworldsampleapprosmelodicgazebo9simapp:latest
```

2. ロボットアプリケーションコンテナとシミュレーションアプリケーションコンテナを実行し、Gazebo GUI ツールを使用してシミュレーションを視覚化できます。以下のコマンドを使用して次の作業を行います。

   1. シミュレーションアプリケーションを実行しているコンテナに接続します。
   2. Gazebo グラフィカルユーザーインターフェイス (GUI) を実行することでアプリケーションを視覚化します。

```
# Enable access to X server to launch Gazebo from docker container
$ xhost +
```

```
# Check that the robot_app and sim_app containers are running. The command should
 list both containers
$ docker container ls

# Connect to the sim app container
$ docker exec -it sim_app bash

# Launch Gazebo from within the container
$ rosrun gazebo_ros gzclient
```

3. イメージにタグを追加して整理します。これらのイメージをタグするには、次のコマンドを使用します。

```
docker tag
 helloworldsampleapprosmelodicgazebo9robotapp:latest accountID.dkr.ecr.us-
west-2.amazonaws.com/helloworldsampleapprosmelodicgazebo9robotapp:latest
```

```
docker tag helloworldsampleapprosmelodicgazebo9simapp:latest accountID.dkr.ecr.us-
west-2.amazonaws.com/helloworldsampleapprosmelodicgazebo9simapp:latest
```

4. アプリケーションが正常に動作していることを確認したら、以下のコマンドを使用して Amazon ECR にプッシュできます。

```
aws ecr get-login-password --region us-west-2 | docker login --username AWS --
password-stdin accountID.dkr.ecr.us-west-2.amazonaws.com
docker push accountID.dkr.ecr.us-west-2.amazonaws.com/
helloworldsampleapprosmelodicgazebo9robotapp:latest
docker push accountID.dkr.ecr.us-west-2.amazonaws.com/
helloworldsampleapprosmelodicgazebo9simapp:latest
```

その後、イメージに対してシミュレーションジョブを実行できます。シミュレーションジョブの詳細については、「AWS RoboMaker によるシミュレーション」を参照してください。

ROS2 Foxy と Gazebo 11 を使用した GPU サンプルアプリケーションの実行

このチュートリアルでは、以下の例に概要が示されている 3 つのコンテナイメージを使用して Hello World のロボットアプリケーションとシミュレーションアプリケーションを作成し、実行することに

より、コンテナイメージ内で GPU ドライバーを使用して ROS 2 Foxy と Gazebo 11 で開発する方法について説明します。

```
### SampleGPUBaseApp // Base Image
#   ### Dockerfile
### SampleGPURobotApp // Image for Robot App
#   ### Dockerfile
#   ### robot-entrypoint.sh
### SampleGPUSimulationApp // Image for Simulation App
#   ### Dockerfile
#   ### simulation-entrypoint.sh
```

各 Dockerfile には、各イメージのビルドに必要な指示があります。

- ベースイメージの Dockerfile には、ROS、Gazebo、GPU ドライバーを設定するためのコマンドがあります。
- ロボットアプリケーションの Dockerfile には、Hello World ロボットアプリケーションを設定するためのコマンドがあります。
- シミュレーションアプリケーションの Dockerfile には、Hello World シミュレーションアプリケーションを設定するためのコマンドがあります。

ロボットアプリケーションとシミュレーションアプリケーションの両方に、エントリポイントスクリプトがあります。これらのスクリプトは、それぞれのアプリケーションの環境をソースにして、コマンドを実行してロボットアプリケーションとシミュレーションアプリケーションを起動するためのパスを設定します。

ベース GPU イメージの作成

次の Dockerfile には、NVIDIA OpenGL からベースイメージを作成して DCV をインストールするためのコマンドが含まれています。

- 以下のコマンドを SampleGPUBaseApp ディレクトリの Dockerfile に保存します。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
FROM nvidia/opengl:1.0-glvnd-runtime-ubuntu20.04

ENV DEBIAN_FRONTEND="noninteractive"
ENV QT_X11_NO_MITSHM=1
```

```
RUN apt-get clean
RUN apt-get update && apt-get install -y --no-install-recommends \
        ca-certificates \
        devilspie \
        gnupg2 \
        mesa-utils \
        sudo \
        unzip \
        wget \
        xfce4-terminal

RUN wget https://d1uj6qtbmh3dt5.cloudfront.net/NICE-GPG-KEY && gpg --import NICE-GPG-
KEY && \
        wget https://d1uj6qtbmh3dt5.cloudfront.net/2021.2/Servers/nice-
dcv-2021.2-11048-ubuntu1804-x86_64.tgz && \
        tar xvzf nice-dcv-2021.2-11048-ubuntu1804-x86_64.tgz && \
        cd nice-dcv-2021.2-11048-ubuntu1804-x86_64 && \
        apt install -y ./nice-dcv-gl_2021.2.944-1_amd64.ubuntu1804.deb

RUN apt update && apt -y install locales && \
        locale-gen en_US en_US.UTF-8 && \
        update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8

ENV LANG=en_US.UTF-8

RUN apt-get update && apt-get install -y --no-install-recommends curl lsb-release

RUN curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key  -o /usr/
share/keyrings/ros-archive-keyring.gpg && \
        curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | apt-
key add - && \
        echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-
archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(lsb_release -cs) main" |
 tee /etc/apt/sources.list.d/ros2.list > /dev/null && \
        apt update && \
        apt install -y ros-foxy-desktop && \
        /bin/bash -c "source /opt/ros/foxy/setup.bash"

RUN apt -y install ros-foxy-gazebo-ros-pkgs

RUN apt-key adv --fetch-keys 'http://packages.osrfoundation.org/gazebo.key' && \
        apt update && \
        apt install -y python3-rosdep git
```

```
RUN if [ ! -f "/etc/ros/rosdep/sources.list.d/20-default.list" ]; then \
        rosdep init; \
    fi

RUN rosdep update

RUN apt-get install -y python3-apt python3-pip python3-vcstool python3-testresources

RUN pip3 install -U pytest setuptools colcon-ros-bundle

RUN useradd --create-home robomaker && \
        sh -c 'echo "robomaker ALL=(root) NOPASSWD:ALL" >> /etc/sudoers'

RUN sh -c 'mkdir -p /home/robomaker/workspace' && \
        sh -c 'cd /home/robomaker/workspace && wget https://github.com/aws-robotics/
aws-robomaker-sample-application-helloworld/archive/ros2.zip && unzip ros2.zip'
```

Dockerfile を作成したら、ターミナルで以下のコマンドを使用してビルドします。

```
cd SampleGPUBaseApp
docker build -t samplegpubaseapp:latest .
```

ベースイメージをビルドすると ROS 2 Foxy、Gazebo 11、NVIDIA OpenGL、NICE-DCV がインストールされます。

ロボットアプリケーション用イメージの作成

ベースイメージを作成したら、ロボットアプリケーションのイメージを作成できます。以下のスクリプトを SampleGPURobotApp ディレクトリの Dockerfile に保存してビルドします。このスクリプトにより、Hello World ロボットアプリケーションがダウンロードされて設定されます。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
FROM samplegpubaseapp:latest

# Build the Robot application
RUN cd /home/robomaker/workspace/aws-robomaker-sample-application-helloworld-ros2/
robot_ws && \
 /bin/bash -c "source /opt/ros/foxy/setup.bash && vcs import < .rosinstall && rosdep
 install --rosdistro foxy --from-paths src --ignore-src -r -y && colcon build"

COPY robot-entrypoint.sh /home/robomaker/robot-entrypoint.sh
RUN sh -c 'sudo chmod +x /home/robomaker/robot-entrypoint.sh'
```

```
RUN sh -c 'sudo chown robomaker:robomaker /home/robomaker/robot-entrypoint.sh'

CMD ros2 launch hello_world_robot rotate.launch.py
ENTRYPOINT [ "/home/robomaker/robot-entrypoint.sh" ]
```

robot-entrypoint.sh として保存できるスクリプトの内容を以下に示します。このスクリプトはロボットアプリケーションの環境をソースにします。

```
#!/bin/bash
cd /home/robomaker/workspace/aws-robomaker-sample-application-helloworld-ros2/robot_ws
source /opt/ros/foxy/setup.bash
source /usr/share/gazebo-11/setup.sh
source ./install/setup.sh
printenv

exec "${@:1}"
```

以下のコマンドにより、Dockerfile からロボットアプリケーションのイメージが作成されます。

```
cd SampleGPURobotApp
docker build -t samplegpurobotapp:latest .
```

シミュレーションアプリケーション用イメージの作成

シミュレーションアプリケーション用イメージの作成

ベースイメージとロボットアプリケーション用イメージを作成すれば、シミュレーションアプリケーションのイメージを作成できます。以下のスクリプトを SampleGPUSimulationApp ディレクトリの Dockerfile に保存してビルドします。このスクリプトにより、Hello World シミュレーションアプリケーションがダウンロードされて設定されます。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
FROM samplegpubaseapp:latest

# Build the Simulation application
RUN cd /home/robomaker/workspace/aws-robomaker-sample-application-helloworld-ros2/
simulation_ws && \
 /bin/bash -c "source /opt/ros/foxy/setup.bash && vcs import < .rosinstall && rosdep
 install --rosdistro foxy --from-paths src --ignore-src -r -y && colcon build"

COPY simulation-entrypoint.sh /home/robomaker/simulation-entrypoint.sh
```

```
RUN sh -c 'sudo chmod +x /home/robomaker/simulation-entrypoint.sh'
RUN sh -c 'sudo chown robomaker:robomaker /home/robomaker/simulation-entrypoint.sh'

CMD ros2 launch hello_world_simulation empty_world.launch.py
ENTRYPOINT [ "/home/robomaker/simulation-entrypoint.sh" ]
```

simulation-entrypoint.sh として保存できるスクリプトの内容を以下に示します。このスクリプトはシミュレーションアプリケーションの環境をソースにします。

```
#!/bin/bash
if [ ! -z $GAZEBO_MASTER_URI ]; then
   tmp_GAZEBO_MASTER_URI=$GAZEBO_MASTER_URI
fi

cd /home/robomaker/workspace/aws-robomaker-sample-application-helloworld-ros2/
simulation_ws
source /opt/ros/foxy/setup.bash
source /usr/share/gazebo-11/setup.sh

 if [ ! -z $tmp_GAZEBO_MASTER_URI ]; then
    export GAZEBO_MASTER_URI=$tmp_GAZEBO_MASTER_URI
    unset tmp_GAZEBO_MASTER_URI
fi

source ./install/setup.sh
printenv

exec "${@:1}"
```

以下のコマンドを実行してイメージを作成します。

```
cd SampleGPUSimulationApp
docker build -t samplegpusimulationapp:latest .
```

アプリケーションを実行して Amazon ECR にプッシュする

イメージを作成したら、ローカルの Linux 環境で適切に動作していることを確認します。イメージが実行されていることを確認した後、Docker イメージを Amazon ECR にプッシュしてシミュレーションジョブを作成できます。

次のコマンドを使用すると、ローカル Linux 環境で Hello World アプリケーションを実行できます。

```
docker run -it -e DISPLAY -v /tmp/.X11-unix/:/tmp/.X11-unix/ --name gpu_robot_app \
-u robomaker -e ROBOMAKER_GAZEBO_MASTER_URI=http://localhost:5555 \
-e ROBOMAKER_ROS_MASTER_URI=http://localhost:11311 \
samplegpurobotapp:latest

docker run -it -e DISPLAY -v /tmp/.X11-unix/:/tmp/.X11-unix/ --name gpu_sim_app \
-u robomaker -e ROBOMAKER_GAZEBO_MASTER_URI=http://localhost:5555 \
-e ROBOMAKER_ROS_MASTER_URI=http://localhost:11311 \
samplegpusimulationapp:latest
```

ロボットアプリケーションコンテナとシミュレーションアプリケーションコンテナを実行すると、Gazebo GUI ツールを使用してシミュレーションを視覚化できます。以下のコマンドを使用して次の作業を行います。

- シミュレーションアプリケーションを実行しているコンテナに接続します。
- Gazebo グラフィカルユーザーインターフェイス (GUI) を実行することでアプリケーションを視覚化します。

```
# Enable access to X server to launch Gazebo from docker container
$ xhost +

# Check that the robot_app and sim_app containers are running. The command should list
 both containers
$ docker container ls

# Connect to the sim app container
$ docker exec -it gpu_sim_app bash

# Launch Gazebo from within the container
$ /home/robomaker/simulation-entrypoint.sh ros2 launch gazebo_ros gzclient.launch.py
```

イメージにタグを追加できます。以下のコマンドによりイメージにタグを付けることができます。

```
docker tag samplegpurobotapp:latest accountID.dkr.ecr.us-west-2.amazonaws.com/
samplegpurobotapp:latest

docker tag samplegpusimulationapp:latest accountID.dkr.ecr.us-west-2.amazonaws.com/
samplegpusimulationapp:latest
```

アプリケーションが正常に動作していることを確認したら、以下のコマンドを使用して Amazon ECR にプッシュできます。

```
aws ecr get-login-password --region us-west-2 | docker login --username AWS --password-
stdin accountID.dkr.ecr.us-west-2.amazonaws.com
docker push accountID.dkr.ecr.us-west-2.amazonaws.com/samplegpurobotapp:latest
docker push accountID.dkr.ecr.us-west-2.amazonaws.com/samplegpusimulationapp:latest
```

これで、これらのイメージを使用して GPU コンピューティングでシミュレーションジョブを実行できるようになりました。シミュレーションジョブの詳細については、「AWS RoboMaker によるシミュレーション」を参照してください。

AWS RoboMaker によるシミュレーション

AWS RoboMaker シミュレーションジョブは、クラウド内で実行されるシミュレーションアプリケーションとロボットアプリケーションのペアリングです。実行中のシミュレーションジョブを、グラフィカルツールおよびターミナルを使用して操作し、センサーデータを視覚化したり、ロボットのコンポーネントを制御したりできます。以下のトピックでは、AWS RoboMaker のシミュレーションジョブを実行、設定、管理、ログ、バッチ処理する方法について説明します。

トピック

シミュレーションの実行

シミュレーションの実行を開始するには、以下の AWS CLI コマンドを使用してアプリケーションを記述します。これらのコマンドには出力があるので、シミュレーションジョブの作成に進む準備ができているかどうかを確認できます。

以下のコマンドは、ロボットアプリケーションに関連するデータを取得します。

```
aws robomaker describe-robot-application --application YOUR-ROBOT-APP-ARN
```

`describe-robot-application` の出力には、以下のデータが含まれています。

```
{
   "arn": "YOUR-ROBOT-APP-ARN",
   "name": "YOUR-ROBOT-APP-NAME",

 ... # Removed extra data for clarity

   "robotSoftwareSuite": {
       "name": "General"
   },
```

```
 ... # Removed extra data for clarity

   "environment": {
       "uri": "YOUR-ROBOT-APP-ECR-URI"
   }
}
```

以下のコマンドは、シミュレーションアプリケーションに関連するデータを取得します。

```
aws robomaker describe-simulation-application --application YOUR-SIM-APP-ARN
```

describe-simulation-application の出力には以下のデータが含まれます。

```
{
   "arn": "YOUR-SIM-APP-ARN",
   "name": "YOUR-SIM-APP-NAME",

 ... # Removed extra data for clarity

   "simulationSoftwareSuite": {
       "name": "SimulationRuntime"
   },
   "robotSoftwareSuite": {
       "name": "General"
   },

 ... # Removed extra data for clarity

   "environment": {
       "uri": "YOUR-SIM-APP-ECR-URI"
   }
}
```

YOUR-ROBOT-APP-ARN および YOUR-SIM-APP-ARN の戻り値を保存します。シミュレーションジョブの送信には、この両方の値が必要です。WorldForge のアセットをシミュレーションジョブにインポートする必要がある場合は、DataSource API を使用してください。こうすることで、ワールドのエクスポートジョブのAmazon S3 の出力ディレクトリにあるワールドアセットを、シミュレーションジョブコンテナ内の好きなインポート先へインポートすることができます。詳細については、「シミュレーションでのエクスポートしたワールドの使用」を参照してください。

シミュレーションジョブを送信するには、create_simulation_job.json という名前の作業ディレクトリに JSON ファイルを作成します。YOUR-IAM-ROLE-ARN、YOUR-ROBOT-APP-ARN、YOUR-SIM-APP-ARN を含む赤色の斜体テキストで表示されている文字列をコピーして貼り付け、編集します。以下の roslaunch コマンド、TurtleBot 環境変数、ツール設定は、hello world サンプルアプリケーション固有のものです。シミュレーションジョブのニーズに応じて、これらの設定を独自のカスタム値に更新する必要があります。詳細については、CreateSimulationJob API を参照してください。

```
{
   "maxJobDurationInSeconds": 3600,
   "iamRole": "IAM-ROLE-ARN",
   "robotApplications": [
       {
           "application": "YOUR-ROBOT-APP-ARN",
           "applicationVersion": "$LATEST",
           "launchConfig": {
               "environmentVariables": {
                   "ROS_IP": "ROBOMAKER_ROBOT_APP_IP",
                   "ROS_MASTER_URI": "http://ROBOMAKER_ROBOT_APP_IP:11311",
                   "GAZEBO_MASTER_URI": "http://ROBOMAKER_SIM_APP_IP:11345"
               },
               "streamUI": false,
               "command": [
                   "/bin/bash", "-c", "roslaunch hello_world_robot rotate.launch"
               ]
           },
           "tools": [
               {
                   "streamUI": true,
                   "name": "robot-terminal",
                   "command": "/entrypoint.sh && xfce4-terminal",
                   "streamOutputToCloudWatch": true,
                   "exitBehavior": "RESTART"
               }
           ]
       }
   ],
   "simulationApplications": [
       {
           "application": "YOUR-SIM-APP-ARN",
           "launchConfig": {
               "environmentVariables": {
```

```
                "ROS_IP": "ROBOMAKER_SIM_APP_IP",
                "ROS_MASTER_URI": "http://ROBOMAKER_ROBOT_APP_IP:11311",
                "GAZEBO_MASTER_URI": "http://ROBOMAKER_SIM_APP_IP:11345",
                "TURTLEBOT3_MODEL":"waffle_pi"
            },
            "streamUI": true,
            "command": [
                "/bin/bash", "-c", "roslaunch hello_world_simulation
 empty_world.launch --wait"
            ]
        },
        "tools": [
            {
                "streamUI": true,
                "name": "gzclient",
                "command": "/entrypoint.sh && gzclient",
                "streamOutputToCloudWatch": true,
                "exitBehavior": "RESTART"
            }
        ]
    }
  ]
}
```

> **Note**
>
> ロボットとシミュレーションアプリケーションの launchConfig オブジェクトにある ROS_ と GAZEBO_ に固有の環境変数設定については、特に注意してください。ROBOMAKER_* 文字列値とポート番号は、ロボットアプリケーションコンテナがシミュレーションアプリケーションコンテナと通信できるようにするために必要なものです。

ジョブの設定を確認したら、以下のコマンドを使用してジョブを送信できます。

```
aws robomaker create-simulation-job --cli-input-json file://create_simulation_job.json
```

AWS RoboMaker でシミュレーションが実行されていることを確認するには、AWS RoboMaker コンソールの [シミュレーションジョブ] ページにアクセスしてください。実行中のジョブを探して選択すると、詳細が表示され、関連するツールが起動します。おめでとうございます。AWS RoboMaker でシミュレーションジョブが実行されました。

シミュレーションの設定

次のセクションでは、シミュレーションジョブの設定方法について説明します。詳細については、アプリケーションの設定 で説明されているコンセプトを参照してください。

セクション

- Amazon VPC アクセスに関するシミュレーションジョブの設定
- シミュレーションジョブ用のインターネットアクセス
- SimulationJob コンピューティングの設定
- カスタムシミュレーションツールの設定
- ルートアクセスとシステム機能

Amazon VPC アクセスに関するシミュレーションジョブの設定

Amazon Virtual Private Cloud (Amazon VPC) でリソースを作成した場合、そのリソースをパブリックインターネット経由で読み取ることはできません。リソースの例としては、Amazon Redshift データウェアハウスや Amazon ElastiCache クラスターなどがあります。また、Amazon Elastic Compute Cloud インスタンス上のサービスである場合もあります。デフォルトでは、Amazon VPC 内のリソースは AWS RoboMaker シミュレーションジョブにアクセスできません。

> **Note**
>
> AWS RoboMaker は、外部接続のない独立したネットワーク上でシミュレーションジョブを実行します。ただし、ジョブが Amazon VPC 内のリソースにアクセスできるようにするには、Amazon VPC サブネット ID とセキュリティグループ ID を含む VPC 固有のデータを指定する必要があります。AWS RoboMaker では、このデータを使用して、Elastic Network Interface (ENI) を設定します。ENI は、ジョブをプライベート Amazon VPC 内の他のリソースに安全に接続するのに役立ちます。

AWS RoboMaker は、専有テナント VPC 内のリソースに接続しません。詳細については、専用 VPCを参照してください

ジョブを作成するときに `VpcConfig` パラメータを使用して、Amazon VPC データを AWS RoboMaker シミュレーションジョブに追加します (「CreateSimulationJob」を参照) 。以下は、AWS CLI にパブリック IP が割り当てられた場合の例です。

```
aws robomaker create-simulation-job \
--output-location s3Bucket=my-bucket,s3Prefix=my-output-folder \
--max-job-duration-in-seconds 3600 \
--iam-role my-role-arn \
--failure-behavior Continue \
--robot-applications application='my-robot-application-
arn,launchConfig={command=["roslaunch", "hello_world_robot", "rotate.launch"]}' \
--simulation-applications application='my-simulation-application-
arn,launchConfig={command=["roslaunch", "hello_world_simulation",
 "empty_world.launch"]}' \
--vpc-config assignPublicIp=true,subnets=comma-separated-vpc-subnet-
ids,securityGroups=comma-separated-security-group-ids
```

> **Note**
>
> シミュレーションジョブを VPC 内で実行するよう設定すると、ENI ペナルティが発生します。ネットワークリソースに接続する際にアドレス解決が遅延する場合があります。

シミュレーションジョブ用のインターネットアクセス

AWS RoboMaker では、指定された VPC データを使用して ENI を設定します。ENI により、ジョブが VPC リソースにアクセスできるようになります。各 ENI には、指定されたサブネットの範囲からプライベート IP アドレスが割り当てられます。デフォルトでは、ENI にパブリック IP アドレスは割り当てられません。

ジョブでインターネットアクセスが必要で (VPC エンドポイントを持たない AWS サービスを検索する場合など) 、プライベートサブネットを使用している場合、VPC 内に NAT を設定できます。Amazon VPC NAT ゲートウェイを使用して、AWS RoboMaker をパブリック IP に割り当てるためのリクエストができます。詳細については、Amazon VPC ユーザーガイド の NAT ゲートウェイ を参照してください。

> **Note**
>
> VPC に添付されたインターネットゲートウェイを使用することはできません。インターネット接続には ENI にパブリック IP アドレスがある必要があります。デフォルトでは、ENI にはプライベート IP アドレスがあります。

パブリックサブネットの使用時にインターネットアクセスを設定するには、assignPublicIp=true を設定して ENI にパブリック IP を割り当てます。

シミュレーションジョブのみがパブリック AWS API へのアクセスが必要で、よりプライバシーを確保したい場合、「AWS RoboMaker とインターフェース VPC エンドポイント (AWS PrivateLink)」を参照してください。この情報を使用すると、インターフェイス VPC エンドポイントを作成し、CreateSimulationJob API を使用して VPC を追加できます。

SimulationJob コンピューティングの設定

SimulationJobs で GPU を使用するには、SimulationJob の ComputeType を設定すれば GPU コンピューティングを使用できます。AWS RoboMaker でグラフィックス処理ユニット (GPU) ベースのシミュレーションジョブを使用すると、次のようなメリットが得られます。

- GPU ベースのシミュレーションジョブは、OpenGL、CUDA、OpenCL、Vulkan を使用することで、GPU 対応のセンサープラグインと、高忠実度のレンダリングとパフォーマンスを必要とするアプリケーションの実行を可能にします。
- GPU ベースのシミュレーションジョブは、AWS RoboMaker GUI ツールの HD 解像度の質が高く、オブジェクトをより詳細に確認できます。GPU では 1 秒あたりのフレームレートが高くなるため、GUI ツール体験は理想的です。
- GPU ベースのシミュレーションにより、シミュレーションジョブの完了時間が短縮されます。GPU を使用すれば、リアルタイム係数と 1 秒あたりのフレーム数のパフォーマンスヒットを引き起こすことなく、複雑なシミュレーションシーンを実行できます。
- GPU ベースのシミュレーションジョブによって強化学習モデルのトレーニングが強化されます。

コンピューティング

CreateSimulationJob リクエストの Compute パラメータを使えば、SimulationJob に必要なコンピューティングの種類を設定できます。

ComputeType

ComputeType によりジョブに必要なコンピューティングのタイプを指定します。有効な値は、CPU および GPU_AND_CPU です。デフォルト値は、「CPU」です。GPU_AND_CPU が指定されている場合、作成されたジョブでは CPU とともに GPU を使用できます。

GPUUnitLimit

GpuUnitLimit パラメータを使用すれば、ジョブに割り当てる必要がある GPU ユニットの数を指定できます。GPU_AND_CPU ComputeType の場合、これは 1 である必要があります。CPU ComputeType の場合、これは 0 である必要があります。

GPU を利用するコンテナを構築する方法については、「GPU アプリケーションを実行するためのイメージの作成」を参照してください。

カスタムシミュレーションツールの設定

AWS RoboMaker では、シミュレーションジョブでアプリケーションのカスタムツールを設定できます。カスタムツールを使用してシミュレーションを操作したり、診断ユーティリティとして使用したり、その他の目的で操作したりすることができます。AWS RoboMaker が提供する rqt や rviz などのデフォルトツールを設定することもできます。シミュレーションジョブが自動パイプラインの一部である場合は、デフォルトのツールを無効にして、使用するリソースを減らすことができます。

最大 10 個のカスタムツールを設定できます。カスタムツールは、メインの ROS 起動プロセスの開始後にのみ起動されます。

カスタムツール設定には以下の要素が含まれています。

- Tool name (ツール名) — ツールの名前。
- コマンド — bash シェルでツールを起動するコマンド。実行可能ツール名を含める必要があります。引数ではカスタム変数を含む環境変数を使用できます。例えば、現在のシミュレーションジョブ ID を使用するには AWS_ROBOMAKER_SIMULATION_JOB_ID を参照できます。
- Exit behavior (終了動作) — カスタムツールが終了した場合に実行されるアクションを決定します。fail を指定するとシミュレーションジョブは失敗します。restart を指定すると、ツールが再起動されます。デフォルトは restart です。
- UI ストリーミング：ツールに対してストリーミングセッションを設定するかどうかを指定します。[True] の場合、AWS RoboMaker により、シミュレーションでの実行中にツールを操作できるように接続が設定されます。そのためにはグラフィカルユーザーインターフェイスが必要です。デフォルトは false です。
- ログ動作：ツール stdout と stderr を CloudWatch Logs にストリーミングするかどうかを指定します。デフォルトは false です。

ルートアクセスとシステム機能

AWS RoboMaker は、シミュレーションジョブで実行されているアプリケーションへの制限されたルート (sudo) アクセスを提供します。以下は、ブロックされる重要な syscall の一覧です (ただし、すべてではありません)。

- acct
- add_key
- bpf
- clock_adjtime
- clock_settime
- clone
- create_module
- delete_module
- finit_module
- get_kernel_syms
- get_mempolicy
- init_module
- ioperm
- iopl
- kcmp
- kexec_file_load
- kexec_load
- keyctl
- lookup_dcookie
- mbind
- mount
- move_pages
- name_to_handle_at
- nfsservctl
- open_by_handle_at
- perf_event_open
- personality
- pivot_root
- process_vm_readv
- process_vm_writev
- ptrace
- query_module
- quotactl
- reboot
- request_key
- set_mempolicy

- setns
- settimeofday
- stime
- swapon
- swapoff
- sysfs
- _sysctl
- umount
- umount2
- unshare
- uselib
- userfaultfd
- ustat
- vm86
- vm86old

シミュレーションの管理

以下のセクションでは、シミュレーションジョブの作成、表示、キャンセル、複製、再起動を行う方法を説明します。

セクション

シミュレーションジョブの作成

任意のシミュレーションプラットフォームを使用して仮想世界でロボットアプリケーションを実行する場合は、シミュレーションジョブを作成します。シミュレーションアプリケーションを指定するときに、ソフトウェアスイート名を選択します。現在、[全般] と [シミュレーションランタイム] のソフトウェアスイートがサポートされています。

シミュレーションジョブを作成するには

以下のいずれかのタブのステップに従ってください。

Using the console

1. AWS RoboMaker コンソール (https://console.aws.amazon.com/robomaker/) にサインインします。
2. 左側のナビゲーションペインで、[Simulation run] (シミュレーション実行)、[Simulation jobs] (シミュレーションジョブ) の順に選択します。
3. [Create simulation job] (シミュレーションジョブの作成) を選択します。
4. [Simulation configuration] (シミュレーション設定) ページで、[simulation job duration] (シミュレーションジョブの期間) を選択します。5 分から 14 日までの値を選択します。

   > **Important**
   >
   > AWS RoboMaker の料金の詳細については、「AWS RoboMaker 料金表」を参照してください。

5. [Failure behavior] (失敗の動作) を選択します。シミュレーションジョブが失敗した場合にホストインスタンスを終了するには、[失敗] を選択します。ホストインスタンスを保持し、接続および調査できるようにするには、[続行] を選択します。

   以下でオプションの S3 フォルダを指定した場合は、このフォルダにシミュレーションデータが格納されます。このフォルダは、選択した失敗の動作に関係なしに利用できます。
6. [IAM ロール] でロールを選択するか、[ロールの新規作成] を選択してロールを作成します。AWS RoboMaker では、このロールを使用して自動的にリソースにアクセスします。このロールは、アプリケーションで Amazon Rekognition や Amazon Lex などの AWS リソースにアクセスする場合にも使用します。
7. オプション: [Compute] (コンピューティング) で、シミュレーション単位制限を選択します。シミュレーションには、指定されたシミュレーション単位制限に比例して CPU とメモリが割り当てられます。シミュレーション単位は 1 vcpu と 2 GB のメモリです。デフォルト値は 15 です。
8. オプション : [出力先] に、シミュレーションジョブ出力の保存先となる Amazon S3 フォルダの名前を入力します。必要に応じて、[Create new S3 folder] (S3 フォルダの新規作成) を選択して新しい Amazon S3 フォルダを作成します。
9. オプション: ロボットアプリケーションまたはシミュレーションアプリケーションが Amazon VPC でリソースにアクセスする場合は、[Networking] (ネットワーク) で、VPC、サブネッ

ト、セキュリティグループを選択します。使用可能なすべてのサブネットを選択して、すべてのリソース制限が使用可能であることを確認します。詳細については、「VPC とサブネット」を参照してください。

VPC 外からシミュレーションジョブにアクセスする場合は、[Assign public IP] (パブリック IP の割り当て) を選択します。

10. 必要に応じて、[Tags] (タグ) で、シミュレーションジョブ用に 1 つ以上のタグを指定します。タグとは、AWS リソースを識別および整理するためのメタデータとして使用される単語やフレーズのことです。各タグは、キーと値から構成されます。シミュレーションジョブのタグは、[Simulation Job details] (シミュレーションジョブの詳細) ページで管理できます。

    タグ付けの詳細については、「AWS Billing and Cost Management」の「コスト配分タグの使用」を参照してください。

11. [Next] (次へ) を選択します。
12. [Specify robot application] (ロボットアプリケーションの指定) ページで、[Robot application] (ロボットアプリケーション) の [Create new application] (アプリケーションの新規作成) を選択します。必要に応じて、[Choose existing application] (既存のアプリケーションの選択) を選択し、作成済みのロボットアプリケーションを使用します。
13. ロボットアプリケーションの名前を入力します。
14. [コンテナイメージ] で、ロボットアプリケーションコンテナの Amazon ECR リポジトリの場所を指定します。詳細については、「AWS RoboMaker 互換コンテナ要件」を参照してください。

    > **Note**
    >
    > $LATEST を使用しても Amazon ECR の変更からユーザーは保護されません。AWS RoboMaker リポジトリにアクセスすると、読み取り専用に設定されます。

    バージョニングの詳細については、バージョニングアプリケーション を参照してください。

15. [ロボットアプリケーション設定] で、ロボットアプリケーションの [起動コマンド] を指定します。
16. オプション: ロボットアプリケーションツールを設定するには、[Robot application tools] (ロボットアプリケーションツール) を展開します。[Use default tools] (デフォルトツールを使用する) を選択して既定のツールを使用します。[ツールをカスタマイズする] を選択して、アプリケーションで使用するためのカスタムツールの追加、削除または編集を行います。

新しいカスタムツールを追加する方法

a. [Add tool] (ツールの追加) を選択します。

b. [Add application tool] (アプリケーションツールの追加) で、[Tool name] (ツール名) を指定します。

c. ツールのコマンドライン引数を指定します。実行可能ツール名を含める必要があります。

d. [Exit behavior] (終了動作) を選択します。[失敗] を選択すると、ツール終了時にシミュレーションジョブが失敗します。[Restart] (再起動) を選択してツールを再起動します。デフォルトは [Restart] (再起動) です。

e. UI ストリーミングの有効化または無効化を選択します。UI ストリーミングはデフォルトで無効になっています。

f. ツールのログを記録する場合は [Send output to CloudWatch] (CloudWatch に出力を送信) を選択します。CloudWatch でそのログを利用できるようになります。デフォルトでは、出力は CloudWatch に送信されません。カスタムツールは、メインの ROS 起動プロセスの開始後にのみ起動されます。

17. オプション: アプリケーションにグラフィカルユーザーインターフェイスが含まれている場合は、[ストリーミングセッションで実行] を選択します。AWS RoboMaker は、シミュレーションの実行中にアプリケーションを操作できるように接続を設定します。[simulation job detail] (シミュレーションジョブの詳細) ページの [Simulation tools] (シミュレーションツール) で [Robot Application] (ロボットアプリケーション) を選択すると接続できます。

18. オプション: ロボットアプリケーションで環境変数を使用する場合は、[Name] (名前) と [Value] (値) のペアを指定します。環境変数名は、A ~ Z またはアンダースコアで始まり、A ~ Z、0 ~ 9、アンダースコアで構成される必要があります。AWS で始まる名前は予約されています。

    他の変数を追加するには、[Add environment variable] (環境変数の追加) を選択します。

    起動ファイル内の環境変数は、roslaunch の 代入引数を使用して読み取ることができます。

19. オプション: シミュレーションジョブポートからアプリケーションポートへのトラフィック転送を選択します。ロボットアプリケーションとシミュレーションアプリケーションのポートマッピングを指定するには、シミュレーションジョブネットワークを設定する必要があります。

20. オプション: [Robot application upload configurations] (ロボットアプリケーションのアップロード設定) を 1 つ以上指定します。アップロード設定を指定するには、シミュレーション

ジョブの出力先を設定する必要があります。各設定で、アップロード動作、Unix glob ファイル一致ルール、および一致するファイルの配置場所を指定します。カスタムアップロードの詳細については、「カスタムアップロード設定の追加」を参照してください。

21. [Next] (次へ) をクリックします。
22. [Specify simulation application] (シミュレーションアプリケーションの指定) ページで、[Create new application] (アプリケーションの新規作成) を選択します。必要に応じて、[既存のアプリケーションの選択] を選択し、作成済みのシミュレーションアプリケーションを使用します。
23. シミュレーションアプリケーションの名前を入力します。
24. [コンテナイメージ] で、ロボットアプリケーションコンテナの Amazon ECR リポジトリの場所を指定します。詳細については、「???」を参照してください。$LATEST を使用しても Amazon ECR の変更からユーザーは保護されません。AWS RoboMaker がリポジトリにアクセスすると、リポジトリは読み取り専用に設定されます。

    バージョニングの詳細については、バージョニングアプリケーション を参照してください。
25. [シミュレーションアプリケーションの構成] で、ロボットアプリケーションの [起動コマンド] を指定します。
26. オプション: ロボットアプリケーションツールを設定するには、[Simulation application tools] (シミュレーションアプリケーションツール) を展開します。[Use default tools] (デフォルトツールを使用する) を選択して既定のツールを使用します。[ツールをカスタマイズする] を選択して、アプリケーションで使用するためのカスタムツールの追加、削除または編集を行います。

    新しいカスタムツールを追加する方法

    a. [Add tool] (ツールの追加) を選択します。
    b. [Add application tool] (アプリケーションツールの追加) で、[Tool name] (ツール名) を指定します。
    c. ツールのコマンドライン引数を指定します。実行可能ツール名を含める必要があります。
    d. [Exit behavior] (終了動作) を選択します。[失敗] を選択すると、ツール終了時にシミュレーションジョブが失敗します。[Restart] (再起動) を選択してツールを再起動します。デフォルトは [Restart] (再起動) です。
    e. UI ストリーミングの有効化または無効化を選択します。UI ストリーミングはデフォルトで無効になっています。

f. ツールのログを記録する場合は [Send output to CloudWatch] (CloudWatch に出力を送信) を選択します。CloudWatch でそのログを利用できるようになります。デフォルトでは、出力は CloudWatch に送信されません。

カスタムツールは、メインの起動プロセスの開始後にのみ起動されます。

27. オプション: アプリケーションにグラフィカルユーザーインターフェイスが含まれている場合は、[ストリーミングセッションで実行] を選択します。AWS RoboMaker は、シミュレーションの実行中にアプリケーションを操作できるように接続を設定します。[simulation job detail] (シミュレーションジョブの詳細) ページの [Simulation tools] (シミュレーションツール) で [Simulation Application] (シミュレーションアプリケーション) を選択すると接続できます。
28. オプション: シミュレーションアプリケーションで環境変数を使用する場合は、[Name] (名前) と [Value] (値) のペアを指定します。他の変数を追加するには、[Add environment variable] (環境変数の追加) を選択します。
29. オプション: シミュレーションジョブポートからアプリケーションポートへのトラフィック転送を選択します。ロボットアプリケーションとシミュレーションアプリケーションのポートマッピングを指定するには、シミュレーションジョブネットワークを設定する必要があります。
30. オプション: [Simulation application upload configurations] (シミュレーションアプリケーションのアップロード設定) を 1 つ以上指定します。アップロード設定を指定するには、シミュレーションジョブの出力先を設定する必要があります。各設定で、アップロード動作、Unix glob ファイル一致ルール、および一致するファイルの配置場所を指定します。

    デフォルトのアップロード設定では、過去のシミュレーションジョブの出力設定との下位互換性が維持されます。デフォルト設定は、作成した追加のアップロード設定に追加されます。カスタムアップロードの詳細については、「カスタムアップロード設定の追加」を参照してください。
31. [Next] (次へ) をクリックします。
32. [Create] (作成) を選択してシミュレーションジョブを作成します。

Using the AWS CLI

Example

別のタブの、コンソールを使用した「シミュレーションジョブの作成」と同じ処理を AWS CLI コマンドで実行する例は次のとおりです。

```
aws robomaker create-simulation-job --max-job-duration-in-seconds 3600
 --iam-role arn:aws:iam::111111111111:role/MyRole --robot-applications
 application=arn:aws:robomaker:us-west-2:111111111111:robot-application/
MyRobotApplication/1551203485821,launchConfig="{command=["roslaunch",
 "hello_world_robot", "rotate.launch"]}" --simulation-applications
 application=arn:aws:robomaker:us-west-2:111111111111:simulation-application/
MySimulationApplication/1551203427605,launchConfig="{command=["roslaunch",
 "hello_world_simulation", "empty_world.launch"]}" --tags Region=North
```

シミュレーションジョブの表示

シミュレーションジョブに関する情報を表示できます。ジョブが実行中である場合は、グラフィカルツールまたはターミナルを起動してシミュレーションを操作できます。シミュレーションジョブの詳細を表示し、タグを管理することもできます。

シミュレーションジョブの表示

以下のいずれかのタブのステップに従ってください。

Using the console

1. AWS RoboMaker コンソール (https://console.aws.amazon.com/robomaker/) にサインインします。
2. 左側のナビゲーションペインで、[Simulations] (シミュレーション)、[Simulation jobs] (シミュレーションジョブ) の順に選択します。
3. シミュレーションジョブの [Id] を選択し、ジョブの作成日時やロボットアプリケーション/シミュレーションアプリケーションの起動コマンドなどの詳細を表示します。

Using the AWS CLI

Example

別のタブの、コンソールを使用した「シミュレーションジョブの表示」と同じ処理を AWS CLI コマンドで実行する例は次のとおりです。

```
aws robomaker list-simulation-jobs
aws robomaker describe-simulation-job --job my-simulation-job-arn
```

シミュレーションジョブのキャンセル

実行中のシミュレーションジョブが不要になった場合は、キャンセルできます。

シミュレーションジョブをキャンセルするには

以下のいずれかのタブのステップに従ってください。

Using the console

1. AWS RoboMaker コンソール (https://console.aws.amazon.com/robomaker/) にサインインします。
2. 左側のナビゲーションペインで、[Simulations] (シミュレーション)、[Simulation jobs] (シミュレーションジョブ) の順に選択します。
3. キャンセルするシミュレーションジョブの [Id] を選択します。
4. [Simulation job detail] (シミュレーションジョブの詳細) ページで、[Actions] (アクション) の [Cancel] (キャンセル) を選択します。
5. [Cancel simulation job] (シミュレーションジョブのキャンセル) ページで、[Yes, cancel] (はい、キャンセルする) を選択します。

Using the AWS CLI

Example

AWS CLI コマンドを使用して、コンソールのタブの「シミュレーションジョブのキャンセル」と同じ処理を実行する例は次のとおりです。

```
aws robomaker list-simulation-jobs
```

```
aws robomaker cancel-simulation-job --job my-simulation-job-arn
```

シミュレーションジョブ内からのキャンセルについては、「ROS コンテナに関するよくある質問」を参照してください。

シミュレーションジョブのクローン

AWS Management Console で [シミュレーションジョブの詳細] ページから既存のシミュレーションジョブをクローン化すれば、既存のシミュレーションジョブから新しいシミュレーションジョブを作成できます。

> **Note**
>
> ROS と Gazebo ソフトウェアスイートを使用したシミュレーションジョブはクローンできません。詳細については、「サポートポリシー」を参照してください。

1. AWS RoboMaker コンソール (https://console.aws.amazon.com/robomaker/) にサインインします。
2. 左側のナビゲーションペインで、[Simulations] (シミュレーション)、[Simulation jobs] (シミュレーションジョブ) の順に選択します。
3. 再起動する実行中のシミュレーションジョブの [Id] を選択します。
4. [Simulation job detail] (シミュレーションジョブの詳細) ページで、[Actions] (アクション) の [Clone] (クローン) を選択します。
5. [Review and create simulation job] (シミュレーションジョブの確認と作成) で、[Edit] (編集) を選択して変更を行います。
6. [Create] (作成) を選択してシミュレーションジョブを作成します。

シミュレーションジョブの再起動

実行中のシミュレーションジョブは再起動できます。再起動したシミュレーションジョブでは、Amazon S3 の場所に格納されているロボットアプリケーションおよびシミュレーションアプリケーションのソースファイルと、シミュレーションジョブの作成時に指定した他のすべての構成設定を使用します。

シミュレーションジョブを再起動するには

以下のいずれかのタブのステップに従ってください。

Using the console

1. AWS RoboMaker コンソール (https://console.aws.amazon.com/robomaker/) にサインインします。
2. 左側のナビゲーションペインで、[Simulations] (シミュレーション)、[Simulation jobs] (シミュレーションジョブ) の順に選択します。
3. 再起動する実行中のシミュレーションジョブの [Id] を選択します。
4. [Simulation job detail] (シミュレーションジョブの詳細) ページで、[Actions] (アクション) の [Restart] (再起動) を選択します。
5. [Restart simulation job] (シミュレーションジョブの再起動) ページで、[Yes, restart] (はい、再起動する) を選択します。

Using the AWS CLI

Example

別のタブの、コンソールを使用した「シミュレーションジョブの再起動」と同じ処理を AWS CLI コマンドで実行する例は次のとおりです。シミュレーションジョブは実行中であることが必要です。

```
aws robomaker restart-simulation-job --job my-simulation-job-arn
```

シミュレーションのロギング

シミュレーションジョブから出力ファイルやその他のアーティファクトをキャプチャする場合は、カスタムアップロードを設定できます。ロボットアプリケーションとシミュレーションアプリケーションのカスタムアップロードを設定できます。カスタムアップロードを設定すると、指定したファイルが、シミュレーションジョブから、指定した Amazon S3 シミュレーション出力場所にアップロードされます。これは、シミュレーションの実行中に生成されたアプリケーション出力のレビューや分析を行う場合や、アーティファクトを再利用する場合に便利です。

カスタムアップロードを設定する前に、シミュレーションジョブの Amazon S3 出力先を指定する必要があります。AWS RoboMaker は、一致するファイルを指定した名前のフォルダにアップロードし

ます。一致するファイルは、すべてのシミュレーションジョブツールが終了したときにアップロードされるか、または、生成されたときにアップロードされ、その後に削除されます。

デフォルトのアップロード設定は、オフにしない限り、カスタムアップロード設定に自動的に追加されます。デフォルトのアップロード設定では、ROS と Gazebo のデフォルトのログ出力がアップロードされます。これにより、ROS と Gazebo のデフォルトのログ出力がアップロードされた過去のシミュレーションジョブ出力設定との互換性が維持されます。コンソールでシミュレーションジョブを設定するときに、デフォルトのアップロード設定をオフにすることができます。CreateSimulationJob API で `useDefaultUploadConfigurations` を `false` に設定することでデフォルトのアップロード設定をオフにすることもできます。

シミュレーションアプリケーションは、単一の 128 GB パーティションに拡張され、そのパーティションに書き込みアクセスできるようになります。

セクション

- カスタムアップロード設定の追加
- AWS RoboMaker によって作成された環境変数

カスタムアップロード設定の追加

カスタムアップロード設定を作成するには、Amazon S3 でのファイルのアップロード先を指定する名前プレフィックス、アップロードするファイルを指定する Unix グロブパス、およびファイルがアップロードされるタイミングを指定するアップロード動作を指定する必要があります。

名前

名前とは、Amazon S3 でのファイルのアップロード方法を指定するプレフィックスです。最終パスを決定するために、シミュレーション出力場所に追加されます。

例えば、シミュレーション出力場所が `s3://my-bucket` であり、アップロード設定の名前が `robot-test` である場合、ファイルは `s3://my-bucket/<simid>/<runid>/robot-test` にアップロードされます。

[Path] (パス)

パスは、アップロードされるファイルを指定するものです。スタンダード Unix glob 一致ルールは、次の条件に従って受け入れられます。

- パスは `/home/robomaker/` または `/var/log` から始めてください。

- パスには逆パス式 (/..) を含めないでください。
- シンボリックリンクは追跡されません。
- パスの中で ** をスーパーアスタリスクとして使用できます。例えば、/var/log/**.log と指定すると .log ディレクトリツリー内のすべての /var/log ファイルが収集されます。

  また、標準のアスタリスクを標準のワイルドカードとして使用することもできます。例えば、/var/log/system.log* は、system.log_1111 で system.log_2222、/var/log などのファイルに一致します。

アップロード動作

次のいずれかのアップロード動作を選択できます。

- 終了時にアップロード (UPLOAD_ON_TERMINATE) は、シミュレーションジョブが終了状態になると、パスと一致するすべてのファイルをアップロードします。AWS RoboMaker は最大60分間ログのアップロードを試みます。

  シミュレーションで実行されているすべてのツールが停止するまで、AWS RoboMaker によってファイルのアップロードが開始されることはありません。
- 自動削除でローリングをアップロード(UPLOAD_ROLLING_AUTO_REMOVE) は、パスに一致するすべてのファイルを生成時にアップロードします。パスは 5 秒ごとにチェックされます。ファイルがアップロードされると、ソースファイルは削除されます。ファイルが削除されると、同じ名前で新規ファイルが作成された場合はこれが以前アップロードされたファイルに置き換わります。AWS RoboMaker は、シミュレーションで実行されているすべてのアプリケーションが停止した後で、最終チェックを行います。

  自動削除によるアップロードローリングは、ローリングログをアップロードする際に役立ちます。パス glob でカバーされない「アクティブ」ファイルに出力を書き込むか、ストリームします。アクティブファイルへの書き込みが終わったら、パス glob がカバーする場所にファイルをロールします。その後、そのファイルはアップロードされて削除されます。

  この設定は、シミュレーションジョブのスペースの節約に役立ちます。また、シミュレーションジョブが終了する前にファイルにアクセスする場合にも役立ちます。

シミュレーションジョブのパーティションサイズは 128 GB です。何らかの理由でシミュレーションジョブが終了した場合、AWS RoboMaker により、カスタムアップロード設定で指定されたすべてのファイルのアップロードが試行されます。

AWS RoboMaker によって作成された環境変数

AWS RoboMaker は、以下のシミュレーションジョブの環境変数を定義します。

- `AWS_ROBOMAKER_SIMULATION_JOB_ID`
- `AWS_ROBOMAKER_SIMULATION_JOB_ARN`
- `AWS_ROBOMAKER_SIMULATION_RUN_ID`

これらの変数は、アプリケーションまたはコマンドラインから取得できます。例えば、Python で現在のシミュレーションジョブ Amazon リソースネーム (ARN) を取得するには、`os.environ.get("AWS_ROBOMAKER_SIMULATION_JOB_ARN")` を使用します。

シミュレーションジョブの Amazon Simple Storage Service 出力バケットを指定した場合、環境変数を使用して出力パスを見つけることができます。AWS RoboMaker は出力を `s3://bucket-name/AWS_ROBOMAKER_SIMULATION_JOB_ID/AWS_ROBOMAKER_SIMULATION_RUN_ID` に書き込みます。この出力を使用して、コードまたはコマンドラインから Amazon S3 でオブジェクトを管理できます。

AWS RoboMaker も、`CreateSimulationJobRequest` にセットアップされた特定の環境変数を扱い、ロボットとシミュレーションアプリケーションコンテナの相互通信を実現します。詳細については、「ROS コンテナに関するよくある質問」を参照してください。

バッチシミュレーション

このセクションでは、シミュレーションジョブバッチを開始および管理する方法について説明します。シミュレーションジョブバッチを使用すると、1 回の API 呼び出しで多数のシミュレーションを起動して実行し、回帰テスト、パラメーター最適化、機械学習モデルトレーニング、合成データ生成を行うことができます。

> **Note**
>
> シミュレーションジョブバッチは、AWS RoboMaker SDK または AWS CLI からのみ開始できます。AWS RoboMaker コンソールを使用して、シミュレーションバッチの表示、クローン作成、およびキャンセルを行うことができます。

セクション

- シミュレーションジョブバッチの開始

- シミュレーションジョブバッチの表示
- シミュレーションジョブバッチのキャンセル
- シミュレーションジョブバッチのクローン作成

シミュレーションジョブバッチの開始

シミュレーションジョブバッチは AWS SDK または AWS CLI から開始します。シミュレーションジョブバッチには、1 つ以上のシミュレーションジョブリクエストが含まれています。各シミュレーションジョブリクエストは、各シミュレーションで使用するアプリケーション、ジョブの最大継続時間、およびその他の情報を識別します。シミュレーションジョブバッチおよび各シミュレーションジョブリクエストにはタグを適用できます。

シミュレーションジョブバッチを開始するには、次の操作を実行する必要があります。

1. AWS Command Line Interface のインストール。AWS CLI のインストールについての詳細は、の「AWS CLI のインストール」を参照してください。
2. 次の JSON を `startsimjobbatch.json` という名前のファイルにコピーします。必要な設定に合わせてファイルを変更し、保存します。

```
{
    "batchPolicy": {
        "timeoutInSeconds": 400,
        "maxConcurrency": 2
    },
    "createSimulationJobRequests": [
        {
            "maxJobDurationInSeconds": 300,
            "iamRole": "arn:aws:iam::111111111111:role/MyRole",
            "failureBehavior": "Fail",
            "robotApplications": [
                {
                    "application": "arn:aws:robomaker:us-east-1:111111111111:robot-
application/MyRobotApplicationArn",
                    "launchConfig": {
                        "packageName": "hello_world_robot",
                        "launchFile": "rotate.launch"
                    }
                }
            ],
            "simulationApplications": [
```

```
                {
                    "application": "arn:aws:robomaker:us-
east-1:111111111111:simulation-applicationMySimulationApplicationArn",
                    "launchConfig": {
                        "command": [
                            "roslaunch", "hello_world_robot", "rotate.launch"
                        ]
                    }
                }
            ],
            "tags": {
                "myRequestTagKey" : "myRequestTagValue"
            }
        },
        {
            "maxJobDurationInSeconds": 300,
            "iamRole": "arn:aws:iam::111111111111:role/MyRole",
            "failureBehavior": "Fail",
            "simulationApplications": [
                {
                    "application": "arn:aws:robomaker:us-
east-1:111111111111:simulation-applicationMySimulationApplicationArn",
                    "launchConfig": {
                        "command": [
                            "roslaunch", "hello_world_simulation",
 "empty_world.launch"
                        ]
                    }
                }
            ]
        }
    ],
    "tags": {
         "myBatchTagKey" : "myBatchTagValue"
     }
}
```

3. コマンドプロンプトを開き、次の AWS CLI コマンドを実行します。

```
$ aws robomaker start-simulation-job-batch --cli-input-json
 file://startsimjobbatch.json
```

シミュレーションジョブバッチを表示するには、「シミュレーションジョブバッチの表示」を参照してください。

シミュレーションジョブバッチの表示

バッチ内のシミュレーションジョブリクエストの詳細を含む、シミュレーションジョブバッチに関する情報を表示できます。

シミュレーションジョブバッチの詳細を表示するには

以下のいずれかのタブのステップに従ってください。

Using the console

1. AWS RoboMaker コンソール (https://console.aws.amazon.com/robomaker/) にサインインします。
2. 左側のナビゲーションペインで、[Simulations] (シミュレーション)、[Simulation job batches] (シミュレーションジョブバッチ) の順に選択します。
3. 詳細を表示するシミュレーションジョブバッチの Id を選択します。

Using the AWS CLI

Example

別のタブの、コンソールを使用した「シミュレーションジョブの表示」と同じ処理を AWS CLI コマンドで実行する例は次のとおりです。

```
aws robomaker list-simulation-job-batches
aws robomaker describe-simulation-job-batch --job my-simulation-job-batch-arn
```

シミュレーションジョブバッチのキャンセル

実行中のシミュレーションジョブが不要になった場合は、キャンセルできます。

シミュレーションジョブをキャンセルするには

以下のいずれかのタブのステップに従ってください。

Using the console

1. AWS RoboMaker コンソール (https://console.aws.amazon.com/robomaker/) にサインインします。
2. 左側のナビゲーションペインで、[Simulations] (シミュレーション)、[Simulation job batches] (シミュレーションジョブバッチ) の順に選択します。
3. キャンセルするシミュレーションジョブバッチの Id を選択します。
4. [Simulation job batch detail] (シミュレーションジョブバッチの詳細) ページで、[Batch actions] (バッチアクション) から [Cancel batch] (バッチのキャンセル) を選択します。
5. [Cancel simulation job batch] (シミュレーションジョブバッチのキャンセル) ページの [Cancel] (キャンセル) を選択します。

Using the AWS CLI

Example

AWS CLI コマンドを使用して、コンソールのタブの「シミュレーションジョブバッチのキャンセル」と同じ処理を実行する例は次のとおりです。

```
$ aws robomaker list-simulation-job-batches
$ aws robomaker cancel-simulation-job-batch --job my-simulation-job-batch-arn
```

シミュレーションジョブバッチのクローン作成

既存のシミュレーションジョブバッチのクローンを作成することで、新しいバッチを開始できます。クローンを作成する場合、すべてのシミュレーションジョブリクエストを含めるか、リクエストのサブセットを選択できます。

> **Note**
>
> ROS と Gazebo のソフトウェアスイートを使用したシミュレーションジョブバッチは複製できません。詳細については、「サポートポリシー」を参照してください。

シミュレーションジョブバッチのクローンを作成するには:

1. AWS RoboMaker コンソール (https://console.aws.amazon.com/robomaker/) にサインインします。
2. 左側のナビゲーションペインで、[Simulations] (シミュレーション)、[Simulation job batches] (シミュレーションジョブバッチ) の順に選択します。
3. クローンを作成するシミュレーションジョブバッチの Id を選択します。
4. バッチ全体のクローンを作成するには、[Simulation job batch detail] (シミュレーションジョブバッチの詳細) ページで、[Batch actions] (バッチアクション) から [Clone batch] (バッチのクローン作成) を選択します。

   バッチの特定のシミュレーションジョブリクエストからクローンを作成するには、[シミュレーションジョブリクエスト] でクローンを作成するシミュレーションジョブリクエストをオンにし、[リクエストアクション]、[リクエストのクローン作成] の順に選択します。
5. [Clone simulation job batch] (シミュレーションジョブバッチのクローン作成) ページの [Submit] (送信) を選択します。

Simulation WorldForge でのワールドの作成

Simulation WorldForgeでは、定義したシミュレーションワールドテンプレートからワールドが生成されます。シミュレーションワールドテンプレートでは、ワールドレイアウト、部屋の寸法、家具、部屋の接続方法、その他の詳細を指定します。壁と床、その他の部屋機能に、素材特性を加えることができます。部屋には、部屋タイプに合わせて自動的に家具を配置するか、または使用可能な家具を選択することもできます。生成されたワールドは、シミュレーションジョブで使用でき、エクスポートしてデベロッパーマシンで使用することができます。

Simulation WorldForge は、ドメインのランダム化を含む多数のシミュレーションワールドが必要となるシミュレーションワークロードを管理するのに役立ちます。一般的な Simulation WorldForge シナリオは以下のとおりです。

- 回帰テスト — 何百ものワールドでロボットアプリケーションをテストして、動作が正常かどうかを検証します。
- 合成画像データ生成 — 生成されたワールドから画像をキャプチャして、それを他のロボットアプリケーションで使用することができます。例えば、家具のレイアウトや素材構成が異なる部屋の画像をキャプチャできます。
- 強化学習 — 数百のユニークなワールドを作成して、インテリア構造をロボットアプリケーションで調べることができます。ワールドの構成は自分で操作します。
- アルゴリズムの開発 — ロボットナビゲーションエンジニアは、家具の配置が異なる既知のレイアウトにおけるナビゲーションアルゴリズムの成功を確認できます。ロボットローカリゼーションエンジニアは、レイアウトアルゴリズムにより、さまざまな間取り図内のさまざまな構造要素が検出されるようにすることができます。

ワールド生成アルゴリズムやインフラストラクチャの作成・管理方法を把握しておく必要はありません。Simulation WorldForge と AWS RoboMaker はフルマネージドサービスです。

WorldForge のシミュレーションの概念

Simulation WorldForge では、パラメータのコレクション (シミュレーションワールドテンプレート) を使用して新しいワールドの生成方法を決定します。1 つのシミュレーションワールドテンプレートを使用して、数百のワールドを生成できます。各ワールドには建物が含まれています。建物はワンフロアです。フロアには、部屋のサイズと形状が説明されている間取りテンプレートがあります。また、部屋をどのように連結できるかについても提示されています。フロアには、間取り図に記載され

ている壁や床などの構造要素の仕上げ方法を示すインテリアテンプレートもあります。インテリアテンプレートには、テーブルやソファなどの家具、衣服や台所用品などの備品を各部屋に追加する方法を説明するパラメータもあります。

https://console.aws.amazon.com/robomaker/ でコンソールを使用すれば、シミュレーションワールドテンプレートをサンプルテンプレートから作成したり、既存のテンプレートをクローン化して作成したり、一から作成したりすることができます。例えば、ベッドルーム 1 つを含むワールドを生成する場合は、1 ベッドルームアパートメントサンプルテンプレートから開始できます。これは、ベッドルームが 1 つ、バスルームが 1 つ、キッチンが 1 つ、リビングルームが 1 つある開放的な間取りです。各部屋タイプに適した一般的な物品と家具、備品を使用します。この間取りが保存されたら、ワールドジェネレータージョブを開始してワールドを生成できます。1 つのワールド生成ジョブでワールドを 50 個まで生成できます。

SDK または AWS Command Line Interface を使用してシミュレーションワールドテンプレートを作成することもできます。例えば、AWS CLI からテンプレートを作成するには、まず、テンプレートボディを使用して ワールドテンプレート JSON ドキュメントテンプレートを作成します。建物、間取り、インテリア、およびその他の詳細に関するパラメータを指定します。テンプレートを保存した後、`create-world-template` を呼び出して JSON ファイルを指定すれば、シミュレーションワールドを作成できます。

```
aws robomaker create-world-template --name "my-template" --templateBody file://
my_template_body.json
```

シミュレーションワールドテンプレートを設定して保存した後、ワールド生成ジョブを作成してワールドを生成できます。1 つのシミュレーションワールドテンプレートから数百のワールドを生成できます。1 つのワールド生成ジョブでワールドを 100 個まで生成できます。ワールドは AWS RoboMaker のシミュレーションで使用できます。ワールドをエクスポートして、独自の ROS 環境で修正して使用することもできます。

シミュレーションワールドテンプレートについて

このセクションでは、シミュレーションワールドテンプレートのコンポーネントについて説明します。コンポーネントには、間取り図、インテリア素材のプリファレンス、家具のプリファレンスが含まれます。Simulation WorldForge には、素材、家具の選定、部屋の接続など、多くのコンポーネントのデフォルト設定があります。デフォルト設定は、独自のプリファレンスでオーバーライドできます。Simulation WorldForge は、ワールド生成時に可能な限りプリファレンスに従います。

間取り図

間取り図は平屋住宅の屋内の間取りを指定するものです。間取り図には、ワールドディメンション、部屋の数とタイプ、部屋の接続方法に影響を与えるパラメータが含まれます。

各ワールドには必ず、座標 (0,0,0) (デフォルトのロボット開始位置) を中心とする 1 メートルのクリアな円筒が設けられます。Simulation WorldForge により部屋が決定されます。

ワールドディメンション

建物の縦横比と天井の高さを設定できます。有効な縦横比は 1:4 ~ 4:1 です。有効な天井高は 2.4 ~ 4.0 メートルです。すべての測定値の単位はメートルと平方メートルです。コンソールはアメリカ慣用単位とメートル法間の変換に対応しています。

ルーム

部屋数、部屋タイプ、部屋名、希望する面積、希望する縦横比、インテリア特性を指定できます。以下の部屋タイプがサポートされています。

- ベッドルーム
- バスルーム
- リビング
- ダイニング
- キッチン
- 廊下
- クローゼット

家具、壁材、床材は、部屋タイプに適したタイプから選択されます。例えば、バスルームには、タイル壁とリノリウムの床が割り当てられ、トイレとシャワーが配置される可能性があります。

接続

Simulation WorldForge では、デフォルト設定によりすべての部屋が自動的に接続されます。開口部または出入口で部屋を接続できます。部屋が開口部でつながっている場合、これらの部屋は開放間取りになります。壁はありません。出入口でつながっている部屋にはドアのない狭い開口部があります。出入口となる開口部は、隣接する壁に沿ってランダムに配置されます。

デフォルトの接続を好みの接続でオーバーライドできます。例えば、キッチン、ダイニングルーム、ベッドルームがある場合、キッチンとベッドルーム間のドア接続をリクエストすることができます。Simulation WorldForge は要求された接続を可能な限り実現しますが、保証はできません。

インテリア

さまざまなインテリア素材と家具タイプから選択できます。Simulation WorldForge は、床材、壁、家具を部屋タイプに合わせてランダムに部屋に割り当てます。例えば、キッチンにはオーブンが、ダイニングルームにはテーブルと椅子が割り当てられます。

床と壁の素材タイプをカスタムセットとして選択できます。カスタムセットを作成すれば、部屋タイプ別または部屋名別にカスタムの割り当てを適用できます。複数のカスタムセットを使用できます。競合がある場合は常に、部屋タイプの割り当てよりも部屋のカスタム割り当てが優先されます。

例えば、すべてのベッドルームにカスタムセット「モダンフローリング」が割り当てられ、部屋「マスターベッドルーム」にカスタムセット「シックフローリング」が割り当てられているとします。Simulation WorldForge により床材が割り当てられると、「マスターベッドルーム」には「シックフローリング」セットの床材が割り当てられます。他のベッドルームには、「モダンフローリング」セットから床材が選択されます。

このルールはカスタム家具セットにも適用されます。

床材タイプ

サポートされている床材タイプには次のものがあります。

- カーペット
- コンクリート
- フロアボード
- リノリウム
- 寄せ木張り
- タイル

床材は、選択したすべての床材タイプからランダムに選択されます。例えば、`Carpet`、`Concrete`、`linoleum`、`parquetry` を指定すると、部屋の床がコンクリートになる可能性があります。

壁材タイプ

サポートされている壁材タイプには次のものがあります。

- れんが
- コンクリート
- 石
- タイル
- 木製パネル
- 壁塗装
- 壁紙

壁材は、選択したすべての壁材タイプからランダムに選択されます。例えば、`Brick`、`Tiles`、`Wallpaper` を指定すると、部屋の壁がタイルと壁紙を使用した壁になる可能性があります。Simulation WorldForge では、選択したすべての壁材タイプから壁材が割り当てられるとは限りません。

家具タイプ

Simulation WorldForge では次の家具タイプがサポートされています。

- 浴槽
- バーキャビネット
- ベッド
- 本棚
- コーヒーテーブル
- コンソールテーブル
- コーナーキャビネット
- デスクチェア
- デスク
- ダイニングチェア
- ダイニングテーブル
- 食洗機
- ドレッサー

- エンドテーブル/サイドテーブル
- フロアランプ
- 冷蔵庫
- リビングルームチェア
- アイランドキッチン/カート
- メディア用収納家具
- ナイトスタンド
- オットマン
- オーブン
- キッチン用カート
- シャワー
- サイドボード/食器棚
- ソファ
- 収納家具
- 収納付きベンチ
- トイレ
- 洗面台
- 洗濯機/乾燥機

家具は、選択したすべての家具タイプからランダムに選択されます。例えば、`Sideboards and buffets`、`Sofas`、`Console tables` を指定すると、部屋にはソファが 1 つと コンソールテーブルが 2 つあるのに、サイドボードや食器棚がない可能性があります。Simulation WorldForge では、選択したすべての家具タイプから素材タイプが割り当てられるとは限りません。

一般的なタスク

このセクションでは、シミュレーションワールドテンプレートを作成するための一般的なタスクについて説明します。タスクの多くは、目的の接続または目的の形状を指定するものです。Simulation WorldForge は、シミュレーションワールドテンプレートパラメータに沿ったワールド生成に最適です。生成されたワールドには、必要なプロパティがすべて含まれるとは限りません。

トピック

- フロアの部屋リストの指定

- 長い廊下をリクエストする
- 部屋間の出入口をリクエストする
- すべての部屋への設定の適用
- 出入口のドアありをリクエストする
- 出入口のドアなしをリクエストする
- 横長の間取りのフットプリントをリクエストする
- カスタム天井高をリクエストする
- 異なる部屋の床に同じ素材タイプを指定する
- 同じタイプの部屋間の床に異なる素材タイプを指定する
- 部屋の家具の密集度を指定する
- すべてのベッドルームと 1 つの共用リビング/ダイニングルームに特定の家具タイプを追加する
- 家具のない部屋を指定する

フロアの部屋リストの指定

部屋タイプは、どの部屋を隣合わせにするかによって、間取りに影響します。部屋タイプは、デフォルトでランダムに配置される床や壁の素材タイプおよび家具タイプを決める際にも使用されます。デフォルトの床や壁の素材タイプおよび家具タイプは、部屋タイプまたは部屋名でオーバーライドすることができます。

部屋タイプは、ベッドルーム、バスルーム、リビングルーム、ダイニングルーム、キッチン、廊下、クローゼットの中から選ぶことができます。

以下の例では、3 ベッドルームハウスを指定します。部屋のサイズと形状はデフォルトで決定されます。

Using the console

1. [Simulation world template edit] (シミュレーションワールドテンプレートの編集) 画面の [Floor plan] (間取り) で、[Rooms] (部屋) を選択します。
2. [Rooms] (部屋) ペインで、[Add room] (部屋の追加) を選択します。
3. 部屋の詳細を追加します。部屋の [Name] (名前)、[Room type] (部屋タイプ)、[Desired area] (希望する面積)、[Desired aspect ratio] (希望する縦横比) を指定できます。
4. [Save] (保存) を選択して新しい部屋を保存します。希望の部屋ができるまでこの操作を繰り返します。部屋を追加しすぎた場合は、[Rooms] (部屋) ペインから削除できます。

Using the AWS CLI

Example

templateBody の次の JSON を create-world-template へのコールの一部として使用できます。

```
"Rooms": [
  {
   "Type": "Bedroom",
   "Name": "My Master Bedroom",
  },
  {
   "Type": "Bathroom",
   "Name": "My Ensuite",
  },
  {
   "Type": "Kitchen",
   "Name": "My Kitchen",
  }
]
```

長い廊下をリクエストする

DesiredShape プロパティを使用して希望する部屋の形状をリクエストできます。Type は部屋の形状には影響しません。以下の例では Hallway 縦横比は低です。十分な大きさの Area と組み合わせると、細く長い廊下が適していることが示されます。Simulation WorldForge により、目的の形状に似た部屋の生成が試行されます。

Using the console

1. [シミュレーションワールドテンプレートの編集] 画面の [間取り] で、[部屋] を選択します。
2. [Rooms] (部屋) ペインで、[Add room] (部屋の追加) を選択します。
3. 部屋の [Name] (名前) を指定して、[Room type] (部屋タイプ) に合わせて [Hallway] (廊下) を選択します。
4. [Desired area] (希望する面積) を 20、[Desired aspect ratio] (希望する縦横比) を 4:1 に指定します。
5. [Save] (保存) を選択して廊下を保存します。

Using the AWS CLI

Example

templateBody の次の JSON を create-world-template へのコールの一部として使用できます。

```
"Rooms": [
  {
    "Type": "Hallway",
    "Name": "My Hallway",
    "DesiredShape": {
      "Area": 20.0,
      "AspectRatio": {
        "x": 4, "y": 1
      }
    }
  }
]
```

部屋面積の有効範囲は 10 ~ 300 メートルです。部屋の縦横比の有効範囲は 1:4 ~ 4:1 です。

部屋間の出入口をリクエストする

2 つの部屋があり、それらが少なくとも 1 つの壁を共有している場合は、2 つの部屋の間の DesiredConnections をリクエストすることができます。Simulation WorldForge は、部屋を隣り合わせで配置しようとします。ConnectionType に応じて、隣接する壁に沿ってランダムな場所に Doorway が配置されるか、または隣接する壁が完全に取り除かれて Opening が作成されます。

次の例では、リビングルームとキッチンのオープン接続をリクエストします。また、ベッドルームとバスルームの接続には出入口をリクエストします。

Using the console

1. [シミュレーションワールドテンプレートの編集] 画面の [間取り] で、[接続] を選択します。
2. [Connections] (接続) ペインで、[Add connection] (接続の追加) を選択します。
3. [希望する接続] ペインで、[接続タイプ] の [開口部] を選択し、次に [場所 1] と [場所 2] の部屋を選択します。例えば、「私のリビングルーム」や「私のキッチン」などです。
4. [Save] (保存) を選択して希望する接続を保存します。

5. この操作を繰り返して、他の 2 つの場所の接続に必要な [ドア] を追加します。例えば、「私のベッドルーム」や「私のバスルーム」などです。

Using the AWS CLI

Example

templateBody の次の JSON を create-world-template へのコールの一部として使用できます。

```
"DesiredConnections": [
  {
    "Location": [ "My Living Room", "My Kitchen" ],
    "ConnectionType": "Opening"
  },
  {
    "Location": [ "My Bedroom", "My Bathroom" ],
    "ConnectionType": "Doorway"
  }
]
```

1 部屋あたりの有効な接続数は 4 で、部屋 1 組ごとに最大 1 つのオープン接続を設定できます。

すべての部屋への設定の適用

> **Note**
>
> バージョン 2 以降のテンプレートを使用している場合に限り、1 つの設定をすべての部屋に適用できます。詳細については、「すべての部屋への設定の適用」を参照してください。

Target.All キーワードを指定すると、すべての部屋に設定が適用できます。

次の例では、すべてのドアのドア状態を変更します。

Using the console

次の手順を実行すると、ワールド内のすべてのドアに設定を適用できます。また、フロア、物品セット、壁、家具のすべてに 1 つの設定を適用することもできます。

1. [シミュレーションワールドテンプレートの編集] 画面の [インテリア] で、[ドア] を選択します。
2. [Doors] (ドア) ペインで [Add custom doors] (カスタムドアの追加) を選択します。
3. [Set name] (セット名) でカスタムドアのセットの名前を指定します。
4. [Rooms affected] (影響を受ける部屋) で [All rooms] (すべての部屋) を指定します。
5. [Door state] (ドア状態) でドアの開放状態を選択します。
6. [Save] (保存) を選択してドア設定を保存します。

Using the AWS CLI

Example

`templateBody` の次の JSON を `create-world-template` へのコールの一部として使用できます。次の例では、出入口セット内のすべてのドアを対象としています。

```
"Interior": {
  "Doorways": {
    "DoorwaySets": [
      {
        "Name": "your-doorway-set",
        "TargetSet": "Target.All",
        "Door": {
          "InitialState": {
            "OpenPosition": {
              "Percent": "percentage-that-you-specify"
            }
          }
        }
      }
    ]
  }
}
```

出入口のドアありをリクエストする

> **Note**
>
> バージョン 2 以降のワールドテンプレートを使用している場合のみ、ドアがある出入口を設定できます。

テンプレートを使用すれば、AWS RoboMaker Simulation WorldForge ワールドの出入口のドアを指定できます。

指定できるドアのタイプは次の通りです。

- ヒンジ付きドア

これらのドアの開放率を設定できます。例として、指定できるいくつかの開放状態を以下に示します。

- 0% 開放 — 閉鎖
- 50% 開放 — 半開
- 70% 開放 — ほぼ全開
- 100% 開放 — 全開

また、AWS RoboMaker によって各ドアにランダムに開放率が割り当てられる設定を選択することもできます。

出入口にドアを追加するには、次の手順を実行します。

Using the console

1. [シミュレーションワールドテンプレートの編集] 画面の [インテリア] で、[ドア] を選択します。
2. [Doors] (ドア) ペインで [Add custom doors] (カスタムドアの追加) を選択します。
3. [Set name] (セット名) でカスタムドアセットに名前を付けます。
4. [Rooms affected] (影響を受ける部屋) の [Location] (場所) で、ドアを付ける部屋を選択します。

5. [Door type] (ドアタイプ) の [Customizations] (カスタマイズ) で追加するドアのタイプを選択します。
6. [ドアの状態] で、ドアが開いている状態、締まっている状態、部分的に空いている状態、ランダム化された状態のいずれかを選択します。
7. [Save] (保存) を選択して設定を保存します。

Using the AWS CLI

Example

`templateBody` の次の JSON を `create-world-template` へのコールの一部として使用できます。

```
"Interior": {
  "Doorways": {
    "DoorwaySets": [
      {
        "Name": "your-doorway-set",
        "TargetSet": "the-doorways-that-you-want-to-target",
        "Door": {
          "InitialState": {
          "OpenPosition": {
            "Percent": "the-open-percentage-that-you-specify-for-the-doors-that-
you're-targeting"
          }
        }
      }
    ]
  }
}
```

出入口のドアなしをリクエストする

> **Note**
>
> バージョン 2 以降のワールドテンプレートを使用している場合のみ、ドアがない出入口を明確に指定できます。

テンプレートを使用すれば、AWS RoboMaker Simulation WorldForge ワールドの出入口にドアがないことを明確に指定できます。

次の例では、部屋間の出入口にドアなしをリクエストします。

Using the console

1. [シミュレーションワールドテンプレートの編集] 画面の [インテリア] で、[ドア] を選択します。
2. [Doors] (ドア) ペインで [Add custom doors] (カスタムドアの追加) を選択します。
3. [Rooms affected] (影響を受ける部屋) ペインの [Location] (場所) で、[All rooms] (すべての部屋) を選択します。
4. [Door type] (ドアタイプ) の [Customizations] (カスタマイズ) で [No door in doorway] (出入口にドアなし) を選択します。
5. [Save] (保存) を選択します。

Using the AWS CLI

Example

`templateBody` の次の JSON を `create-world-template` へのコールの一部として使用できます。

```
"Interior": {
"Doorways": {
  "DoorwaySets": [
    {
      "Name": "doorway-set-name",
      "TargetSet": "Target.All",
      "Door": null
    }
  ]
}
```

横長の間取りのフットプリントをリクエストする

すべての部屋に影響を与える縦長または横長の間取りを希望する場合、`Footprint` の `DesiredAspectRatio` をリクエストできます。Simulation WorldForge では、リクエストしたフッ

トプリントの縦横比に間取りがしっかりと適合するように、部屋の全体的な形状と位置に影響を与えるこの優先設定を使用します。縦横比の設定はオプションで、デフォルトは正方形です。

次の例では、デフォルトの平方比 (1:1) を希望する広めのレイアウトにオーバーライドします。このレイアウトでは、正方形以外のフットプリントを作成するために、すべての部屋が引き伸ばされて配置されます。

Using the console

1. [シミュレーションワールドテンプレートの編集] 画面の [間取り] で、[ワールドディメンション] を選択します。
2. [World dimensions] (ワールドディメンション) ペインの [Desired aspect ratio] (希望する縦横比) で、[Width] (横) を 1、[Length] (縦) を 4 に指定します。
3. [Save] (保存) を選択して新しい部屋を保存します。

Using the AWS CLI

Example

`templateBody` の次の JSON を `create-world-template` へのコールの一部として使用できます。

```
"Footprint": {
  "DesiredAspectRatio": {
    "x": 1, "y": 4
  }
}
```

`DesiredAspectRatio` の有効範囲は 1:4 ~ 4:1 です。

カスタム天井高をリクエストする

間取りの天井高によってすべての部屋の壁の高さが決まります。デフォルトの天井高は 2.4 メートルです。この例では、デフォルトを 3.2 メートルにオーバーライドします。

Using the console

1. [シミュレーションワールドテンプレートの編集] 画面の [間取り] で、[ワールドディメンション] を選択します。

2. [World dimensions] (ワールドディメンション) ペインで、[Ceiling height] (天井高) を `3.2` に指定します。
3. [Save] (保存) を選択して新しい部屋を保存します。

Using the AWS CLI

Example

`templateBody` の次の JSON を `create-world-template` へのコールの一部として使用できます。

```
"Ceiling": {
  "Height": 3.2
}
```

異なる部屋の床に同じ素材タイプを指定する

部屋タイプまたは部屋名を使用して、インテリア床セクションの複数の部屋を一覧表示します。次の例では、ベッドルーム、リビングルーム、ダイニングルームのすべてに床板材がランダムに割り当てられます。

Using the console

1. [シミュレーションワールドテンプレートの編集] 画面の [インテリア] で、[床] を選択します。
2. [Flooring] (床) ペインで [Add flooring] (床の追加) を選択します。
3. [カスタム床] ペインで、「床材セット 1」などといった床の [セット名] を指定します。
4. [Filter type] (フィルタータイプ) で [By room type] (部屋タイプ) を選択します。
5. [Room types] (部屋タイプ) で、[Bedrooms] (ベッドルーム)、[Living rooms] (リビングルーム)、[Dining rooms] (ダイニングルーム) を選択します。
6. [Custom flooring] (カスタム床) で、[Add material] (素材の追加) を選択し、次に [Floorboard] (床板) を選択します。
7. [Save] (保存) を選択して床セットを保存します。

Using the AWS CLI

Example

templateBody の次の JSON を create-world-template へのコールの一部として使用できます。

```
"Flooring": {
  "MaterialSets": [
    {
      "Name": "Flooring Material Set 1",
      "TargetSet": {
        "RoomTypes": [ "Bedroom", "Living", "Dining" ]
      },
      "SampleSet": {
        "MaterialTypes": [ "Floorboards" ]
      }
    }
  ]
}
```

同じタイプの部屋間の床に異なる素材タイプを指定する

次の例では、Bedroom 3 以外のベッドルーム、リビングルーム、ダイニングルームのすべてに床板材がランダムに割り当てられます。カーペット素材がランダムに割り当てられます。

Using the console

1. [シミュレーションワールドテンプレートの編集] 画面の [インテリア] で、[床] を選択します。
2. [Flooring] (床) ペインで [Add flooring] (床の追加) を選択します。
3. [カスタム床] ペインで床の [セット名] (Flooring Material Set 1 など) を指定します。
4. [Filter type] (フィルタータイプ) で [By room type] (部屋タイプ) を選択します。
5. [Room types] (部屋タイプ) で、[Bedrooms] (ベッドルーム)、[Living rooms] (リビングルーム)、[Dining rooms] (ダイニングルーム) を選択します。
6. [Custom flooring] (カスタム床) で、[Add material] (素材の追加) を選択し、次に [Floorboard] (床板) を選択します。

7. [Save] (保存) を選択して床セットを保存します。
8. [Flooring] (床) ペインで [Add flooring] (床の追加) を選択します。
9. [カスタム床] ペインで床の [セット名] (`Flooring Material Set for Bedroom 3` など) を指定します。
10. [Filter type] (フィルタータイプ) で [By room name] (部屋名) を選択します。
11. [部屋名] (`Bedroom 3` など) で部屋名を選択します。
12. [Custom flooring] (カスタム床) で、[Add material] (素材の追加) を選択し、次に [Carpet] (カーペット) を選択します。
13. [Save] (保存) を選択して床セットを保存します。

Using the AWS CLI

Example

`templateBody` の次の JSON を `create-world-template` へのコールの一部として使用できます。

```
"Flooring": {
  "MaterialSets": [
    {
      "Name": "Flooring Material Set 1",
      "TargetSet": {
        "RoomTypes": [ "Bedroom", "Living", "Dining" ]
      },
      "SampleSet": {
        "MaterialTypes": [ "Floorboards" ]
      }
    },
    {
      "Name": "Flooring Material Set for Bedroom 3",
      "TargetSet": {
        "RoomNames": [ "Bedroom 3" ]
      },
      "SampleSet": {
        "MaterialTypes": [ "Carpet" ]
      }
    }
  ]
}
```

部屋の家具の密集度を指定する

家具の密集度を部屋名または部屋タイプで指定できます。デフォルトでは、部屋には家具が中程度の間隔でランダムに配置されます。次の例では、すべてのベッドルームの家具が密集してランダムに配置されています。リビングルームとダイニングルームの家具はまばらに配置されています。他のすべての部屋にはデフォルトで家具が配置されます。

Using the console

1. [シミュレーションワールドテンプレートの編集] 画面の [インテリア] で、[家具] を選択します。
2. [Furniture] (家具) ペインで [Add custom furniture] (カスタム家具の追加) を選択します。
3. [カスタム家具] ペインで、カスタム家具の [セット名] (`Dense Furniture Arrangement` など) を指定します。
4. [Filter type] (フィルタータイプ) で [By room type] (部屋タイプ) を選択します。
5. [Room types] (部屋タイプ) で [Bedrooms] (ベッドルーム) を選択します。
6. デフォルトの家具を使用する場合は [Override furniture] (家具のオーバーライド) をオンにします。
7. [Furniture density] (家具の密集度) で [Dense] (密集) を選択します。
8. [Save] (保存) を選択して家具セットを保存します。
9. [Furniture] (家具) ペインで [Add custom furniture] (カスタム家具の追加) を選択します。
10. [カスタム家具] ペインで、カスタム家具の [セット名] (`Sparse Furniture Arrangement` など) を指定します。
11. [Filter type] (フィルタータイプ) で [By room name] (部屋名) を選択します。
12. [部屋名] で、家具をまばらに配置にしたい部屋を選択します。例えば、`My Living Room` や `My Dining Room` などです。
13. デフォルトの家具を使用する場合は [Override furniture] (家具のオーバーライド) をオンにします。
14. [Furniture density] (家具の密集度) で [Sparse] (散在) を選択します。
15. [Save] (保存) を選択して家具セットを保存します。

Using the AWS CLI

Example

templateBody の次の JSON を create-world-template へのコールの一部として使用できます。

```
"Furniture": {
  "FurnitureArrangements": [
    {
      "Name": "Dense Furniture Arrangement",
      "TargetSet": {
        "RoomTypes": [ "Bedroom" ]
      },
      "DesiredSpatialDensity": "Dense"
    },
    {
      "Name": "Sparse Furniture Arrangement",
      "TargetSet": {
        "RoomNames": [ "My Living Room", "My Dining Room" ]
      },
      "DesiredSpatialDensity": "Sparse"
    }
  ]
}
```

すべてのベッドルームと 1 つの共用リビング/ダイニングルームに特定の家具タイプを追加する

部屋の家具タイプは、部屋名または部屋タイプで指定できます。次の例では、すべてのベッドルームにランダムなベッド、デスク、ドレッサー、フロアランプが適度に揃えられています。部屋「私のリビング/ダイニングルーム」には、ランダムなダイニングテーブル、ダイニングチェア、フロアランプ、ソファ、コーヒーテーブルが密集して配置されています。他のすべての部屋にはデフォルトで家具が配置されます。

Using the console

1. [シミュレーションワールドテンプレートの編集] 画面の [インテリア] で、[家具] を選択します。
2. [Furniture] (家具) ペインで [Add custom furniture] (カスタム家具の追加) を選択します。

3. [カスタム家具] ペインで、カスタム家具の [セット名] (`Bedroom Furniture` など) を指定します。
4. [Filter type] (フィルタータイプ) で [By room type] (部屋タイプ) を選択します。
5. [Room types] (部屋タイプ) で [Bedrooms] (ベッドルーム) を選択します。
6. [Override furniture] (家具のオーバーライド) が選択されていることを確認します。選択されていない場合、Simulation WorldForge によりデフォルトの家具が使用されます。
7. [Furniture types] (家具のタイプ) で、[Add furniture] (家具の追加) を選択し、次に [Beds] (ベッド)、[Desks] (デスク)、[Dressers] (ドレッサー)、[Floorlamp] (フロアランプ) を選択します。
8. [Save] (保存) を選択して家具セットを保存します。
9. [Furniture] (家具) ペインで [Add custom furniture] (カスタム家具の追加) を選択します。
10. [カスタム家具] ペインで、カスタム家具の [セット名] (`Living and Dining Furniture` など) を指定します。
11. [Filter type] (フィルタータイプ) で [By room name] (部屋名) を選択します。
12. [部屋名] で、`My living and dining room` などの部屋を選択します。
13. [Override furniture] (家具のオーバーライド) が選択されていることを確認します。選択されていない場合、Simulation WorldForge によりデフォルトの家具が使用されます。
14. [Furniture types] (家具のタイプ) で、[DiningTables] (ダイニングテーブル)、[DiningChairs] (ダイニングチェア)、[FloorLamps] (フロアランプ)、[Sofas] (ソファ)、[CoffeeTables] (コーヒーテーブル) を選択します。
15. [Furniture density] (家具の密集度) で [Dense] (密集) を選択します。
16. [Save] (保存) を選択して家具セットを保存します。

Using the AWS CLI

Example

`templateBody` の次の JSON を `create-world-template` へのコールの一部として使用できます。

```
"Furniture": {
  "FurnitureArrangements": [
    {
      "Name": "Bedroom Furniture",
      "TargetSet": {
        "RoomTypes": [ "Bedroom" ]
```

```
      },
      "SampleSet": {
        "ModelTypes": [
          "Beds",
          "Desks",
          "Dressers",
          "FloorLamps"
        ]
      }
    }
    {
      "Name": "Living and Dining Furniture",
      "TargetSet": {
        "RoomNames": [ "My living and dining room" ]
      },
      "SampleSet": {
        "ModelTypes": [
          "DiningTables",
          "DiningChairs",
          "FloorLamps",
          "Sofas",
          "CoffeeTables"
        ],
        "DesiredSpatialDensity": "Dense"
      }
    }
  ]
}
```

家具のない部屋を指定する

家具配置のモデルセットとして空のリストを指定します。他のすべての部屋にはデフォルトで家具が配置されます。

Using the console

1. [シミュレーションワールドテンプレートの編集] 画面の [インテリア] で、[家具] を選択します。
2. [Furniture] (家具) ペインで [Add custom furniture] (カスタム家具の追加) を選択します。
3. [カスタム家具] ペインで、カスタム家具の [セット名] (`No furniture` など) を指定します。

4. [Filter type] (フィルタータイプ) で [By room name] (部屋名) を選択します。
5. [部屋名] で、`My Spare Room` などの家具がない状態にしたい部屋を選択します。
6. [Override furniture] (家具のオーバーライド) が選択されていることを確認します。選択されていない場合、Simulation WorldForge によりデフォルトの家具が使用されます。
7. [Furniture types] (家具のタイプ) で、どのタイプも選択されていないことを確認してください。
8. [Save] (保存) を選択して家具セットを保存します。

Using the AWS CLI

Example

`templateBody` の次の JSON を `create-world-template` へのコールの一部として使用できます。

```
"Furniture": {
  "FurnitureArrangements": [
    {
      "Name": "No Furniture",
      "TargetSet": {
        "RoomNames": [ "My Spare Room" ]
      },
      "SampleSet": {
        "ModelTypes": []
      }
    }
  ]
}
```

シミュレーションワールドテンプレート本文の JSON スキーマ

`templateBody` (シミュレーションワールドテンプレート本文) は CreateWorldTemplate オペレーションの入力パラメータです。このパラメータは JSON 形式の文字列です。JSON はシミュレーションワールドテンプレートを指定するもので、Simulation WorldForge でワールドの生成に使用されるパラメータが含まれています。

以下はワールドテンプレートの各バージョンのスキーマです。

バージョン 2

以下はバージョン 2 スキーマのテンプレートです。

```
{
  "title": "WorldTemplate",
  "description": "The top-level template for parameterizing a randomly generated world.
 By default, a single\nresidential building with one floor and one room is generated.",
  "type": "object",
  "properties": {
    "Version": {
      "title": "Version",
      "type": "string"
    },
    "Buildings": {
      "title": "Buildings",
      "default": [
        {
          "Floors": [
            {
              "Floorplan": {
                "Footprint": {
                  "DesiredAspectRatio": {
                    "x": 1.0,
                    "y": 1.0
                  }
                },
                "Ceiling": {
                  "Height": 3.0
                },
                "Rooms": [
                  {
                    "Type": "Living",
                    "Name": "My_Living_Room",
                    "OriginalName": "My Living Room",
                    "DesiredShape": {
                      "Area": 20.0,
                      "AspectRatio": {
                        "x": 1.0,
                        "y": 1.0
                      }
                    }
                  }
                ],
```

```
              "DesiredConnections": []
            },
            "Interior": {
              "Doorways": {
                "DoorwaySets": []
              },
              "Flooring": {
                "MaterialSets": []
              },
              "Walls": {
                "MaterialSets": []
              },
              "Furniture": {
                "FurnitureArrangements": []
              }
            }
          }
        ]
      }
    ],
    "type": "array",
    "items": {
      "$ref": "#/definitions/BuildingTemplate"
    },
    "minItems": 1,
    "maxItems": 1
  }
},
"required": [
  "Version"
],
"additionalProperties": false,
"definitions": {
  "AspectRatio": {
    "title": "AspectRatio",
    "type": "object",
    "properties": {
      "x": {
        "title": "X",
        "default": 1,
        "minimum": 1,
        "maximum": 4,
        "type": "number"
      },
```

```
        "y": {
          "title": "Y",
          "default": 1,
          "minimum": 1,
          "maximum": 4,
          "type": "number"
        }
      },
      "additionalProperties": false
    },
    "FloorplanFootprint": {
      "title": "FloorplanFootprint",
      "description": "The desired footprint of this floorplan.",
      "type": "object",
      "properties": {
        "DesiredAspectRatio": {
          "title": "Desiredaspectratio",
          "default": {
            "x": 1.0,
            "y": 1.0
          },
          "allOf": [
            {
              "$ref": "#/definitions/AspectRatio"
            }
          ]
        }
      },
      "additionalProperties": false
    },
    "FloorplanCeiling": {
      "title": "FloorplanCeiling",
      "description": "The height of the ceiling for this floorplan in metres.",
      "type": "object",
      "properties": {
        "Height": {
          "title": "Height",
          "default": 3.0,
          "type": "number",
          "minimum": 2.4,
          "maximum": 4.0
        }
      },
      "additionalProperties": false
```

```
    },
    "Rectangle": {
      "title": "Rectangle",
      "description": "A rectangle defined by area in square metres and aspect ratio.",
      "type": "object",
      "properties": {
        "Area": {
          "title": "Area",
          "type": "number"
        },
        "AspectRatio": {
          "$ref": "#/definitions/AspectRatio"
        }
      },
      "required": [
        "Area",
        "AspectRatio"
      ],
      "additionalProperties": false
    },
    "FloorplanRoom": {
      "title": "FloorplanRoom",
      "description": "A description for single room for this floorplan.",
      "type": "object",
      "properties": {
        "Type": {
          "title": "Type",
          "enum": [
            "Bedroom",
            "Bathroom",
            "Living",
            "Dining",
            "Kitchen",
            "Hallway",
            "Closet"
          ],
          "type": "string"
        },
        "Name": {
          "title": "Name",
          "maxLength": 255,
          "minLength": 1,
          "pattern": "^[a-zA-Z0-9_\\- ]*$",
          "type": "string"
```

```
        },
        "OriginalName": {
          "title": "Originalname",
          "type": "string"
        },
        "DesiredShape": {
          "title": "Desiredshape",
          "default": {
            "Area": 20.0,
            "AspectRatio": {
              "x": 1.0,
              "y": 1.0
            }
          },
          "allOf": [
            {
              "$ref": "#/definitions/Rectangle"
            }
          ]
        }
      },
      "required": [
        "Type",
        "Name"
      ],
      "additionalProperties": false
    },
    "FloorplanConnection": {
      "title": "FloorplanConnection",
      "description": "Descibes the desired layout of the rooms and their adjacent
rooms. A connection can be either a doorway or\nan open space without any walls. Two
rooms cannot both share an interior doorway and an opening.\nThe same two rooms can
have multiple doorways, up to a limit.",
      "type": "object",
      "properties": {
        "Location": {
          "title": "Location",
          "type": "array",
          "items": {
            "type": "string"
          },
          "minItems": 2,
          "maxItems": 2
        },
```

```
        "ConnectionType": {
          "title": "Connectiontype",
          "enum": [
            "Doorway",
            "Opening"
          ],
          "type": "string"
        }
      },
      "required": [
        "Location",
        "ConnectionType"
      ],
      "additionalProperties": false
    },
    "FloorplanTemplate": {
      "title": "FloorplanTemplate",
      "description": "The top-level floorplan template that parameterizes the randomly
 generated\narchitectural layout. By default, a residential floorplan with bedroom
 and\nliving room are generated with a random doorway or opening connection.\n\nThe
 footprint contributes to the overall shape of the floor layout along\nwith rooms. The
 footprint shape is desired as it is a preference and not\nguaranteed.\n\nThe ceiling
 determines the height of the walls. There are minimum and maximum ceiling heights.
 The ceiling height is guaranteed.\n\nRooms are required. Each room has a desired
 shape. Together, the room\nshapes and footprint determine floor layout. The room
 types contribute to\nthe layout and are used when randomly selecting furniture and
 materials for\nthe walls and floors.\n\nDesiredConnections are optional. Two rooms are
 connected if they share a\nwall and doorway or adjacent without any wall aka \"opening
\". All rooms are\nguaranteed to be connected randomly if they are not specified in the
\nconnections list. Connections that are specified are _not_ guaranteed but\nwill be
 attempted as best-effort.",
      "type": "object",
      "properties": {
        "Footprint": {
          "title": "Footprint",
          "default": {
            "DesiredAspectRatio": {
              "x": 1.0,
              "y": 1.0
            }
          },
          "allOf": [
            {
              "$ref": "#/definitions/FloorplanFootprint"
```

```
            }
          ]
        },
        "Ceiling": {
          "title": "Ceiling",
          "default": {
            "Height": 3.0
          },
          "allOf": [
            {
              "$ref": "#/definitions/FloorplanCeiling"
            }
          ]
        },
        "Rooms": {
          "title": "Rooms",
          "default": [
            {
              "Type": "Living",
              "Name": "My_Living_Room",
              "OriginalName": "My Living Room",
              "DesiredShape": {
                "Area": 20.0,
                "AspectRatio": {
                  "x": 1.0,
                  "y": 1.0
                }
              }
            }
          ],
          "type": "array",
          "items": {
            "$ref": "#/definitions/FloorplanRoom"
          },
          "minItems": 1,
          "maxItems": 6
        },
        "DesiredConnections": {
          "title": "Desiredconnections",
          "default": [],
          "type": "array",
          "items": {
            "$ref": "#/definitions/FloorplanConnection"
          },
```

```
          "minItems": 0,
          "maxItems": 12
        }
      },
      "additionalProperties": false
    },
    "RoomNameList": {
      "title": "RoomNameList",
      "description": "The set of all rooms matching any of the listed room names.",
      "type": "object",
      "properties": {
        "RoomNames": {
          "title": "Roomnames",
          "type": "array",
          "items": {
            "type": "string"
          },
          "minItems": 1,
          "maxItems": 6
        }
      },
      "required": [
        "RoomNames"
      ],
      "additionalProperties": false
    },
    "RoomTypeList": {
      "title": "RoomTypeList",
      "description": "The set of all rooms matching any of the listed room types.",
      "type": "object",
      "properties": {
        "RoomTypes": {
          "title": "Roomtypes",
          "type": "array",
          "items": {
            "enum": [
              "Bedroom",
              "Bathroom",
              "Living",
              "Dining",
              "Kitchen",
              "Hallway",
              "Closet"
            ],
```

```
            "type": "string"
          },
          "minItems": 1,
          "maxItems": 7
        }
      },
      "required": [
        "RoomTypes"
      ],
      "additionalProperties": false
    },
    "RoomPairTargetFilter": {
      "title": "RoomPairTargetFilter",
      "description": "Defines a target set as a pair of rooms. The pairs are defined as
 the cross product of two lists\nFrom and To.",
      "type": "object",
      "properties": {
        "From": {
          "title": "From",
          "anyOf": [
            {
              "$ref": "#/definitions/RoomNameList"
            },
            {
              "$ref": "#/definitions/RoomTypeList"
            }
          ]
        },
        "To": {
          "title": "To",
          "anyOf": [
            {
              "$ref": "#/definitions/RoomNameList"
            },
            {
              "$ref": "#/definitions/RoomTypeList"
            }
          ]
        }
      },
      "required": [
        "From",
        "To"
      ],
```

```
      "additionalProperties": false
    },
    "DoorOpenPosition": {
      "title": "DoorOpenPosition",
      "description": "Defines the amount of openness of an InteriorDoor.\n\nThe range
 for Percent is [0., 100.]",
      "type": "object",
      "properties": {
        "Percent": {
          "title": "Percent",
          "default": 100.0,
          "anyOf": [
            {
              "type": "number",
              "minimum": 0.0,
              "maximum": 100.0
            },
            {
              "const": "Random",
              "type": "string"
            }
          ]
        }
      },
      "additionalProperties": false
    },
    "DoorInitialState": {
      "title": "DoorInitialState",
      "description": "Defines the initial state for an InteriorDoor object\n
\nOpenPosition specifies how much the door should be open.",
      "type": "object",
      "properties": {
        "OpenPosition": {
          "title": "Openposition",
          "default": {
            "Percent": 100.0
          },
          "allOf": [
            {
              "$ref": "#/definitions/DoorOpenPosition"
            }
          ]
        }
      },
```

```
      "additionalProperties": false
    },
    "InteriorDoor": {
      "title": "InteriorDoor",
      "description": "Custom configuration for each Doorway Set.\n\nInitial State of
doors includes the ability to configure how much the door should be open in\npercent
[0., 100.]",
      "type": "object",
      "properties": {
        "InitialState": {
          "title": "Initialstate",
          "default": {
            "OpenPosition": {
              "Percent": 100.0
            }
          },
          "allOf": [
            {
              "$ref": "#/definitions/DoorInitialState"
            }
          ]
        }
      },
      "additionalProperties": false
    },
    "InteriorDoorwaySet": {
      "title": "InteriorDoorwaySet",
      "description": "A set of doors to randomly assign to a set of interior target
elements.\n\nThe target set determines *what room pairs* are receive the doors as
specified in `Door`.\nRooms may be targeted by room type or room name.\n\nThe Door
customizes the configuration for doors added in the specified target set.",
      "type": "object",
      "properties": {
        "Name": {
          "title": "Name",
          "maxLength": 255,
          "minLength": 1,
          "pattern": "^[a-zA-Z0-9_\\- ]*$",
          "type": "string"
        },
        "TargetSet": {
          "title": "Targetset",
          "anyOf": [
            {
```

```
              "const": "Target.All",
              "type": "string"
            },
            {
              "$ref": "#/definitions/RoomPairTargetFilter"
            }
          ]
        },
        "Door": {
          "title": "Door",
          "anyOf": [
            {
              "$ref": "#/definitions/InteriorDoor"
            },
            {
              "const": null
            }
          ]
        }
      },
      "required": [
        "Name",
        "TargetSet"
      ],
      "additionalProperties": false
    },
    "InteriorDoorways": {
      "title": "InteriorDoorways",
      "description": "Describes the interior template parameters for all doorways for
this floorplan.\nAll doorways not explicitly targeted will have a random door assigned
fully opened.",
      "type": "object",
      "properties": {
        "DoorwaySets": {
          "title": "Doorwaysets",
          "default": [],
          "type": "array",
          "items": {
            "$ref": "#/definitions/InteriorDoorwaySet"
          },
          "minItems": 0,
          "maxItems": 13
        }
      },
```

```
      "additionalProperties": false
    },
    "MaterialSetByMaterialType": {
      "title": "MaterialSetByMaterialType",
      "description": "The set of materials that match any of the material types listed.
 An empty\nset is invalid since all targets require materials.",
      "type": "object",
      "properties": {
        "MaterialTypes": {
          "title": "Materialtypes",
          "type": "array",
          "items": {
            "type": "string"
          },
          "minItems": 1
        }
      },
      "required": [
        "MaterialTypes"
      ],
      "additionalProperties": false
    },
    "InteriorMaterialSet": {
      "title": "InteriorMaterialSet",
      "description": "A set of sample materials to randomly assign to a set of interior
target elements.\n\nThe target set determines *what rooms* receive the materials
in the sample\nset. The targets in a room are the walls and flooring. Rooms may be
targeted\nby room type or room name.\n\nThe sample set determines *what materials* to
randomly select for the\ntarget rooms' walls and floors.\n\nThe sample set is optional
and when not specified (null) materials are\nrandomly selected according to the room
type for each room in the target\nset.\n\nA sample set with an empty material set is
invalid since all wall\nand flooring targets require materials.",
      "type": "object",
      "properties": {
        "Name": {
          "title": "Name",
          "maxLength": 255,
          "minLength": 1,
          "pattern": "^[a-zA-Z0-9_\\- ]*$",
          "type": "string"
        },
        "TargetSet": {
          "title": "Targetset",
          "anyOf": [
```

```
            {
              "const": "Target.All",
              "type": "string"
            },
            {
              "anyOf": [
                {
                  "$ref": "#/definitions/RoomNameList"
                },
                {
                  "$ref": "#/definitions/RoomTypeList"
                }
              ]
            }
          ]
        },
        "SampleSet": {
          "$ref": "#/definitions/MaterialSetByMaterialType"
        }
      },
      "required": [
        "Name",
        "TargetSet"
      ],
      "additionalProperties": false
    },
    "InteriorFlooring": {
      "title": "InteriorFlooring",
      "description": "Describes the interior template parameters for all floors for
this floorplan.\nAll floors not explicitly targeted will have a random floor material
assigned by room type.",
      "type": "object",
      "properties": {
        "MaterialSets": {
          "title": "Materialsets",
          "default": [],
          "type": "array",
          "items": {
            "$ref": "#/definitions/InteriorMaterialSet"
          },
          "minItems": 0,
          "maxItems": 6
        }
      },
```

```
      "additionalProperties": false
    },
    "InteriorWalls": {
      "title": "InteriorWalls",
      "description": "Describes the interior template parameters for all walls for
this floorplan.\nAll walls not explicitly targeted will have a random wall material
assigned by room type.",
      "type": "object",
      "properties": {
        "MaterialSets": {
          "title": "Materialsets",
          "default": [],
          "type": "array",
          "items": {
            "$ref": "#/definitions/InteriorMaterialSet"
          },
          "minItems": 0,
          "maxItems": 6
        }
      },
      "additionalProperties": false
    },
    "ModelTypeList": {
      "title": "ModelTypeList",
      "description": "The set of all models matching any of the listed model types.\nAn
empty set means zero models to sample/select.",
      "type": "object",
      "properties": {
        "ModelTypes": {
          "title": "Modeltypes",
          "type": "array",
          "items": {
            "enum": [
              "Baths",
              "BarCabinets",
              "Beds",
              "Bookcases",
              "CoffeeTables",
              "ConsoleTables",
              "CornerCabinets",
              "DeskChairs",
              "Desks",
              "DiningChairs",
              "DiningTables",
```

```
              "DishWashers",
              "Dressers",
              "EndAndSideTables",
              "FloorLamps",
              "Fridges",
              "LivingRoomChairs",
              "KitchenIslandsAndCarts",
              "MediaStorage",
              "Nightstands",
              "Ottomans",
              "Ovens",
              "ServingCarts",
              "Showers",
              "SideboardsAndBuffets",
              "Sofas",
              "Storage",
              "StorageBenches",
              "Toilets",
              "VanityCounters",
              "WashingMachinesAndDryers"
            ],
            "type": "string"
          },
          "minItems": 0
        }
      },
      "required": [
        "ModelTypes"
      ],
      "additionalProperties": false
    },
    "FurnitureArrangementSet": {
      "title": "FurnitureArrangementSet",
      "description": "Describes the interior template for placing furniture in one
or more rooms.\n\n- TargetSet is the set of rooms to furnish, filter by room name or
room\n  type.\n- SampleSet is a set of all furnishing models to randomly choose and\n
 place.\n- DesiredSpatialDensity is the desired level of free space after placing\n
furniture.",
      "type": "object",
      "properties": {
        "Name": {
          "title": "Name",
          "maxLength": 255,
          "minLength": 1,
```

```
          "pattern": "^[a-zA-Z0-9_\\- ]*$",
          "type": "string"
        },
        "TargetSet": {
          "title": "Targetset",
          "anyOf": [
            {
              "const": "Target.All",
              "type": "string"
            },
            {
              "anyOf": [
                {
                  "$ref": "#/definitions/RoomNameList"
                },
                {
                  "$ref": "#/definitions/RoomTypeList"
                }
              ]
            }
          ]
        },
        "SampleSet": {
          "$ref": "#/definitions/ModelTypeList"
        },
        "DesiredSpatialDensity": {
          "title": "Desiredspatialdensity",
          "default": "Moderate",
          "enum": [
            "Sparse",
            "Moderate",
            "Dense"
          ],
          "type": "string"
        }
      },
      "required": [
        "Name",
        "TargetSet"
      ],
      "additionalProperties": false
    },
    "InteriorFurnishings": {
      "title": "InteriorFurnishings",
```

```
      "description": "Describes the types of furniture models for randomly placing into
 each room\nin the world.  Rooms are targeted by room type or room name. Rooms that
 are\nnot targeted are furnished at random by their room type with moderate density.
\ndensity. For an empty room, specify an empty sample set.",
      "type": "object",
      "properties": {
        "FurnitureArrangements": {
          "title": "Furniturearrangements",
          "default": [],
          "type": "array",
          "items": {
            "$ref": "#/definitions/FurnitureArrangementSet"
          },
          "minItems": 0,
          "maxItems": 6
        }
      },
      "additionalProperties": false
    },
    "InteriorTemplate": {
      "title": "InteriorTemplate",
      "description": "Top-level template for parameterizing the interior finishes and
 furnishings for\nthis floorplan.",
      "type": "object",
      "properties": {
        "Doorways": {
          "title": "Doorways",
          "default": {
            "DoorwaySets": []
          },
          "allOf": [
            {
              "$ref": "#/definitions/InteriorDoorways"
            }
          ]
        },
        "Flooring": {
          "title": "Flooring",
          "default": {
            "MaterialSets": []
          },
          "allOf": [
            {
              "$ref": "#/definitions/InteriorFlooring"
```

```
            }
          ]
        },
        "Walls": {
          "title": "Walls",
          "default": {
            "MaterialSets": []
          },
          "allOf": [
            {
              "$ref": "#/definitions/InteriorWalls"
            }
          ]
        },
        "Furniture": {
          "title": "Furniture",
          "default": {
            "FurnitureArrangements": []
          },
          "allOf": [
            {
              "$ref": "#/definitions/InteriorFurnishings"
            }
          ]
        }
      },
      "additionalProperties": false
    },
    "FloorTemplate": {
      "title": "FloorTemplate",
      "description": "Describes a single foor within a building. Defaults to a
single residential room\nof a randomy type and size, and the interior is randomly
furnished.",
      "type": "object",
      "properties": {
        "Floorplan": {
          "title": "Floorplan",
          "default": {
            "Footprint": {
              "DesiredAspectRatio": {
                "x": 1.0,
                "y": 1.0
              }
            },
```

```
            "Ceiling": {
              "Height": 3.0
            },
            "Rooms": [
              {
                "Type": "Living",
                "Name": "My_Living_Room",
                "OriginalName": "My Living Room",
                "DesiredShape": {
                  "Area": 20.0,
                  "AspectRatio": {
                    "x": 1.0,
                    "y": 1.0
                  }
                }
              }
            ],
            "DesiredConnections": []
          },
          "allOf": [
            {
              "$ref": "#/definitions/FloorplanTemplate"
            }
          ]
        },
        "Interior": {
          "title": "Interior",
          "default": {
            "Doorways": {
              "DoorwaySets": []
            },
            "Flooring": {
              "MaterialSets": []
            },
            "Walls": {
              "MaterialSets": []
            },
            "Furniture": {
              "FurnitureArrangements": []
            }
          },
          "allOf": [
            {
              "$ref": "#/definitions/InteriorTemplate"
```

```
            }
          ]
        }
      },
      "additionalProperties": false
    },
    "BuildingTemplate": {
      "title": "BuildingTemplate",
      "description": "Describes a building to be randomly generated. Defaults to one
 residential floor.",
      "type": "object",
      "properties": {
        "Floors": {
          "title": "Floors",
          "default": [
            {
              "Floorplan": {
                "Footprint": {
                  "DesiredAspectRatio": {
                    "x": 1.0,
                    "y": 1.0
                  }
                },
                "Ceiling": {
                  "Height": 3.0
                },
                "Rooms": [
                  {
                    "Type": "Living",
                    "Name": "My_Living_Room",
                    "OriginalName": "My Living Room",
                    "DesiredShape": {
                      "Area": 20.0,
                      "AspectRatio": {
                        "x": 1.0,
                        "y": 1.0
                      }
                    }
                  }
                ],
                "DesiredConnections": []
              },
              "Interior": {
                "Doorways": {
```

```
                  "DoorwaySets": []
                },
                "Flooring": {
                  "MaterialSets": []
                },
                "Walls": {
                  "MaterialSets": []
                },
                "Furniture": {
                  "FurnitureArrangements": []
                }
              }
            }
          ],
          "type": "array",
          "items": {
            "$ref": "#/definitions/FloorTemplate"
          },
          "minItems": 1,
          "maxItems": 1
        }
      },
      "additionalProperties": false
    }
  }
}
```

バージョン 1

以下はバージョン 1 スキーマのテンプレートです。

```
{
  "title": "WorldTemplate",
  "description": "The top-level template for parameterizing a randomly generated world.
 By default, a single\nresidential building with one floor and one room is generated.",
  "type": "object",
  "properties": {
    "Version": {
      "title": "Version",
      "default": "1",
      "type": "string"
    },
    "Buildings": {
```

```
      "title": "Buildings",
      "default": [
        {
          "Floors": [
            {
              "Floorplan": {
                "Footprint": {
                  "DesiredAspectRatio": {
                    "x": 1.0,
                    "y": 1.0
                  }
                },
                "Ceiling": {
                  "Height": 3.0
                },
                "Rooms": [
                  {
                    "Type": "Living",
                    "Name": "My Living Room",
                    "DesiredShape": {
                      "Area": 20.0,
                      "AspectRatio": {
                        "x": 1.0,
                        "y": 1.0
                      }
                    }
                  },
                  {
                    "Type": "Bedroom",
                    "Name": "My Bedroom",
                    "DesiredShape": {
                      "Area": 20.0,
                      "AspectRatio": {
                        "x": 1.0,
                        "y": 1.0
                      }
                    }
                  }
                ],
                "DesiredConnections": []
              },
              "Interior": {
                "Flooring": {
                  "MaterialSets": []
```

```
                },
                "Walls": {
                  "MaterialSets": []
                },
                "Furniture": {
                  "FurnitureArrangements": []
                }
              }
            }
          ]
        }
      ],
      "type": "array",
      "items": {
        "$ref": "#/definitions/BuildingTemplate"
      },
      "minItems": 1,
      "maxItems": 1
    }
  },
  "additionalProperties": false,
  "definitions": {
    "AspectRatio": {
      "title": "AspectRatio",
      "type": "object",
      "properties": {
        "x": {
          "title": "X",
          "default": 1,
          "minimum": 1,
          "maximum": 4,
          "type": "number"
        },
        "y": {
          "title": "Y",
          "default": 1,
          "minimum": 1,
          "maximum": 4,
          "type": "number"
        }
      },
      "additionalProperties": false
    },
    "FloorplanFootprint": {
```

```
      "title": "FloorplanFootprint",
      "description": "The desired footprint of this floorplan.",
      "type": "object",
      "properties": {
        "DesiredAspectRatio": {
          "title": "Desiredaspectratio",
          "default": {
            "x": 1.0,
            "y": 1.0
          },
          "allOf": [
            {
              "$ref": "#/definitions/AspectRatio"
            }
          ]
        }
      },
      "additionalProperties": false
    },
    "FloorplanCeiling": {
      "title": "FloorplanCeiling",
      "description": "The height of the ceiling for this floorplan in metres.",
      "type": "object",
      "properties": {
        "Height": {
          "title": "Height",
          "default": 3.0,
          "type": "number",
          "minimum": 2.4,
          "maximum": 4.0
        }
      },
      "additionalProperties": false
    },
    "Rectangle": {
      "title": "Rectangle",
      "description": "A rectangle defined by area in square metres and aspect ratio.",
      "type": "object",
      "properties": {
        "Area": {
          "title": "Area",
          "type": "number"
        },
        "AspectRatio": {
```

```
          "$ref": "#/definitions/AspectRatio"
        }
      },
      "required": [
        "Area",
        "AspectRatio"
      ],
      "additionalProperties": false
    },
    "FloorplanRoom": {
      "title": "FloorplanRoom",
      "description": "A description for single room for this floorplan.",
      "type": "object",
      "properties": {
        "Type": {
          "title": "Type",
          "enum": [
            "Bedroom",
            "Bathroom",
            "Living",
            "Dining",
            "Kitchen",
            "Hallway",
            "Closet"
          ],
          "type": "string"
        },
        "Name": {
          "title": "Name",
          "type": "string"
        },
        "DesiredShape": {
          "title": "Desiredshape",
          "default": {
            "Area": 20.0,
            "AspectRatio": {
              "x": 1.0,
              "y": 1.0
            }
          },
          "allOf": [
            {
              "$ref": "#/definitions/Rectangle"
            }
```

```
          ]
        }
      },
      "required": [
        "Type",
        "Name"
      ],
      "additionalProperties": false
    },
    "FloorplanConnection": {
      "title": "FloorplanConnection",
      "description": "Descibes the desired layout of the rooms and their adjacent
rooms. A connection can be either a doorway or \nan open space without any walls. Two
rooms cannot both share an interior doorway and an opening. \nThe same two rooms can
have multiple doorways, up to a limit.",
      "type": "object",
      "properties": {
        "Location": {
          "title": "Location",
          "type": "array",
          "items": {
            "type": "string"
          },
          "minItems": 2,
          "maxItems": 2
        },
        "ConnectionType": {
          "title": "Connectiontype",
          "enum": [
            "Doorway",
            "Opening"
          ],
          "type": "string"
        }
      },
      "required": [
        "Location",
        "ConnectionType"
      ],
      "additionalProperties": false
    },
    "FloorplanTemplate": {
      "title": "FloorplanTemplate",
```

```
      "description": "The top-level floorplan template that parameterizes the randomly
 generated \narchitectural layout. By default, a residential floorplan with bedroom
 and \nliving room are generated with a random doorway or opening connection. \n\nThe
 footprint contributes to the overall shape of the floor layout along\nwith rooms. The
 footprint shape is desired as it is a preference and not\nguaranteed.\n\nThe ceiling
 determines the height of the walls. There are minimum and\nmaximum ceiling heights.
 The ceiling height is guaranteed.\n\nRooms are required. Each room has a desired
 shape. Together, the room\nshapes and footprint determine floor layout. The room
 types contribute to\nthe layout and are used when randomly selecting furniture and
 materials for\nthe walls and floors.\n\nDesiredConnections are optional. Two rooms are
 connected if they share a\nwall and doorway or adjacent without any wall aka \"opening
\". All rooms are\nguaranteed to be connected randomly if they are not specified in the
\nconnections list. Connections that are specified are _not_ guaranteed but\nwill be
 attempted as best-effort.",
      "type": "object",
      "properties": {
        "Footprint": {
          "title": "Footprint",
          "default": {
            "DesiredAspectRatio": {
              "x": 1.0,
              "y": 1.0
            }
          },
          "allOf": [
            {
              "$ref": "#/definitions/FloorplanFootprint"
            }
          ]
        },
        "Ceiling": {
          "title": "Ceiling",
          "default": {
            "Height": 3.0
          },
          "allOf": [
            {
              "$ref": "#/definitions/FloorplanCeiling"
            }
          ]
        },
        "Rooms": {
          "title": "Rooms",
          "default": [
```

```
            {
              "Type": "Living",
              "Name": "My Living Room",
              "DesiredShape": {
                "Area": 20.0,
                "AspectRatio": {
                  "x": 1.0,
                  "y": 1.0
                }
              }
            },
            {
              "Type": "Bedroom",
              "Name": "My Bedroom",
              "DesiredShape": {
                "Area": 20.0,
                "AspectRatio": {
                  "x": 1.0,
                  "y": 1.0
                }
              }
            }
          ],
          "type": "array",
          "items": {
            "$ref": "#/definitions/FloorplanRoom"
          },
          "minItems": 1,
          "maxItems": 6
        },
        "DesiredConnections": {
          "title": "Desiredconnections",
          "default": [],
          "type": "array",
          "items": {
            "$ref": "#/definitions/FloorplanConnection"
          },
          "minItems": 0,
          "maxItems": 12
        }
      },
      "additionalProperties": false
    },
    "RoomNameList": {
```

```
      "title": "RoomNameList",
      "description": "The set of all rooms matching any of the listed room names.",
      "type": "object",
      "properties": {
        "RoomNames": {
          "title": "Roomnames",
          "type": "array",
          "items": {
            "type": "string"
          }
        }
      },
      "required": [
        "RoomNames"
      ],
      "additionalProperties": false
    },
    "RoomTypeList": {
      "title": "RoomTypeList",
      "description": "The set of all rooms matching any of the listed room types.",
      "type": "object",
      "properties": {
        "RoomTypes": {
          "title": "Roomtypes",
          "type": "array",
          "items": {
            "enum": [
              "Bedroom",
              "Bathroom",
              "Living",
              "Dining",
              "Kitchen",
              "Hallway",
              "Closet"
            ],
            "type": "string"
          }
        }
      },
      "required": [
        "RoomTypes"
      ],
      "additionalProperties": false
    },
```

```
    "MaterialSetByMaterialType": {
      "title": "MaterialSetByMaterialType",
      "description": "The set of materials that match any of the material types listed.
 An empty\nset is invalid since all targets require materials.",
      "type": "object",
      "properties": {
        "MaterialTypes": {
          "title": "Materialtypes",
          "type": "array",
          "items": {
            "type": "string"
          },
          "minItems": 1
        }
      },
      "required": [
        "MaterialTypes"
      ],
      "additionalProperties": false
    },
    "InteriorMaterialSet": {
      "title": "InteriorMaterialSet",
      "description": "A set of sample materials to randomly assign to a set of interior
target elements.\n\nThe target set determines *what rooms* receive the materials
in the sample\nset. The targets in a room are the walls and flooring. Rooms may be
targeted \nby room type or room name. \n\nThe sample set determines *what materials*
to randomly select for the\ntarget rooms' walls and floors. \n\nThe sample set is
optional and when not specified (null) materials are\nrandomly selected according to
the room type for each room in the target\nset.\n\nA sample set with an empty material
set is invalid since all wall \nand flooring targets require materials.",
      "type": "object",
      "properties": {
        "Name": {
          "title": "Name",
          "type": "string"
        },
        "TargetSet": {
          "title": "Targetset",
          "anyOf": [
            {
              "$ref": "#/definitions/RoomNameList"
            },
            {
              "$ref": "#/definitions/RoomTypeList"
```

```
            }
          ]
        },
        "SampleSet": {
          "$ref": "#/definitions/MaterialSetByMaterialType"
        }
      },
      "required": [
        "Name",
        "TargetSet"
      ],
      "additionalProperties": false
    },
    "InteriorFlooring": {
      "title": "InteriorFlooring",
      "description": "Describes the interior template parameters for all floors for
this floorplan.\nAll floors not explicitly targeted will have a random floor material
assigned by room type.",
      "type": "object",
      "properties": {
        "MaterialSets": {
          "title": "Materialsets",
          "default": [],
          "type": "array",
          "items": {
            "$ref": "#/definitions/InteriorMaterialSet"
          },
          "minItems": 0,
          "maxItems": 6
        }
      },
      "additionalProperties": false
    },
    "InteriorWalls": {
      "title": "InteriorWalls",
      "description": "Describes the interior template parameters for all walls for
this floorplan.\nAll walls not explicitly targeted will have a random wall material
assigned by room type.",
      "type": "object",
      "properties": {
        "MaterialSets": {
          "title": "Materialsets",
          "default": [],
          "type": "array",
```

```
          "items": {
            "$ref": "#/definitions/InteriorMaterialSet"
          },
          "minItems": 0,
          "maxItems": 6
        }
      },
      "additionalProperties": false
    },
    "ModelTypeList": {
      "title": "ModelTypeList",
      "description": "The set of all models matching any of the listed model types.\nAn
empty set means zero models to sample/select.",
      "type": "object",
      "properties": {
        "ModelTypes": {
          "title": "Modeltypes",
          "type": "array",
          "items": {
            "type": "string"
          },
          "minItems": 0
        }
      },
      "required": [
        "ModelTypes"
      ],
      "additionalProperties": false
    },
    "FurnitureArrangementSet": {
      "title": "FurnitureArrangementSet",
      "description": "Describes the interior template for placing furniture in one
or more rooms.\n\n- TargetSet is the set of rooms to furnish, filter by room name or
room\n  type.\n- SampleSet is a set of all furnishing models to randomly choose and\n
 place. \n- DesiredSpatialDensity is the desired level of free space after placing\n
furniture.",
      "type": "object",
      "properties": {
        "Name": {
          "title": "Name",
          "type": "string"
        },
        "TargetSet": {
          "title": "Targetset",
```

```
          "anyOf": [
            {
              "$ref": "#/definitions/RoomNameList"
            },
            {
              "$ref": "#/definitions/RoomTypeList"
            }
          ]
        },
        "SampleSet": {
          "$ref": "#/definitions/ModelTypeList"
        },
        "DesiredSpatialDensity": {
          "title": "Desiredspatialdensity",
          "default": "Moderate",
          "enum": [
            "Sparse",
            "Moderate",
            "Dense"
          ],
          "type": "string"
        }
      },
      "required": [
        "Name",
        "TargetSet"
      ],
      "additionalProperties": false
    },
    "InteriorFurnishings": {
      "title": "InteriorFurnishings",
      "description": "Describes the types of furniture models for randomly placing into
 each room\nin the world.  Rooms are targeted by room type or room name. Rooms that
 are\nnot targeted are furnished at random by their room type with moderate density.
\ndensity. For an empty room, specify an empty sample set.",
      "type": "object",
      "properties": {
        "FurnitureArrangements": {
          "title": "Furniturearrangements",
          "default": [],
          "type": "array",
          "items": {
            "$ref": "#/definitions/FurnitureArrangementSet"
          },
```

```
          "minItems": 0,
          "maxItems": 6
        }
      },
      "additionalProperties": false
    },
    "InteriorTemplate": {
      "title": "InteriorTemplate",
      "description": "Top-level template for parameterizing the interior finishes and
 furnishings for\nthis floorplan.",
      "type": "object",
      "properties": {
        "Flooring": {
          "title": "Flooring",
          "default": {
            "MaterialSets": []
          },
          "allOf": [
            {
              "$ref": "#/definitions/InteriorFlooring"
            }
          ]
        },
        "Walls": {
          "title": "Walls",
          "default": {
            "MaterialSets": []
          },
          "allOf": [
            {
              "$ref": "#/definitions/InteriorWalls"
            }
          ]
        },
        "Furniture": {
          "title": "Furniture",
          "default": {
            "FurnitureArrangements": []
          },
          "allOf": [
            {
              "$ref": "#/definitions/InteriorFurnishings"
            }
          ]
```

```
        }
      },
      "additionalProperties": false
    },
    "FloorTemplate": {
      "title": "FloorTemplate",
      "description": "Describes a single foor within a building. Defaults to a
single residential room\nof a randomy type and size, and the interior is randomly
furnished.",
      "type": "object",
      "properties": {
        "Floorplan": {
          "title": "Floorplan",
          "default": {
            "Footprint": {
              "DesiredAspectRatio": {
                "x": 1.0,
                "y": 1.0
              }
            },
            "Ceiling": {
              "Height": 3.0
            },
            "Rooms": [
              {
                "Type": "Living",
                "Name": "My Living Room",
                "DesiredShape": {
                  "Area": 20.0,
                  "AspectRatio": {
                    "x": 1.0,
                    "y": 1.0
                  }
                }
              },
              {
                "Type": "Bedroom",
                "Name": "My Bedroom",
                "DesiredShape": {
                  "Area": 20.0,
                  "AspectRatio": {
                    "x": 1.0,
                    "y": 1.0
                  }
```

```
                }
              }
            ],
            "DesiredConnections": []
          },
          "allOf": [
            {
              "$ref": "#/definitions/FloorplanTemplate"
            }
          ]
        },
        "Interior": {
          "title": "Interior",
          "default": {
            "Flooring": {
              "MaterialSets": []
            },
            "Walls": {
              "MaterialSets": []
            },
            "Furniture": {
              "FurnitureArrangements": []
            }
          },
          "allOf": [
            {
              "$ref": "#/definitions/InteriorTemplate"
            }
          ]
        }
      },
      "additionalProperties": false
    },
    "BuildingTemplate": {
      "title": "BuildingTemplate",
      "description": "Describes a building to be randomly generated. Defaults to one
residential floor.",
      "type": "object",
      "properties": {
        "Floors": {
          "title": "Floors",
          "default": [
            {
              "Floorplan": {
```

```
                "Footprint": {
                  "DesiredAspectRatio": {
                    "x": 1.0,
                    "y": 1.0
                  }
                },
                "Ceiling": {
                  "Height": 3.0
                },
                "Rooms": [
                  {
                    "Type": "Living",
                    "Name": "My Living Room",
                    "DesiredShape": {
                      "Area": 20.0,
                      "AspectRatio": {
                        "x": 1.0,
                        "y": 1.0
                      }
                    }
                  },
                  {
                    "Type": "Bedroom",
                    "Name": "My Bedroom",
                    "DesiredShape": {
                      "Area": 20.0,
                      "AspectRatio": {
                        "x": 1.0,
                        "y": 1.0
                      }
                    }
                  }
                ],
                "DesiredConnections": []
              },
              "Interior": {
                "Flooring": {
                  "MaterialSets": []
                },
                "Walls": {
                  "MaterialSets": []
                },
                "Furniture": {
                  "FurnitureArrangements": []
```

```
                }
              }
            }
          ],
          "type": "array",
          "items": {
            "$ref": "#/definitions/FloorTemplate"
          },
          "minItems": 1,
          "maxItems": 1
        }
      },
      "additionalProperties": false
    }
  }
}
```

JSON のサンプルワールドテンプレート

templateBody (シミュレーションワールドテンプレート本文) は CreateWorldTemplate API の入力パラメータです。このパラメータは JSON 形式の文字列です。JSON はシミュレーションワールドテンプレートを指定するもので、Simulation WorldForge でワールドの生成に使用されるパラメータが含まれています。

このセクションでは、サンプルのシミュレーションワールドテンプレートボディについて説明します。

トピック

1 ベッドルームハウス

以下の例では、1 ベッドルームハウスを指定します。インテリア用品と家具を指定します。

```
{
  "name": "OneBedroomHouse",
  "templateBody": {
    "Version": "2",
```

```
    "Buildings": [
      {
        "Floors": [
          {
            "Floorplan": {
              "Footprint": {
                "DesiredAspectRatio": {
                  "x": 1,
                  "y": 1
                }
              },
              "Ceiling": {
                "Height": 3
              },
              "Rooms": [
                {
                  "Type": "Bedroom",
                  "Name": "Bedroom",
                  "DesiredShape": {
                    "Area": 25,
                    "AspectRatio": {
                      "x": 1,
                      "y": 1.2
                    }
                  }
                },
                {
                  "Type": "Living",
                  "Name": "Living room",
                  "DesiredShape": {
                    "Area": 30,
                    "AspectRatio": {
                      "x": 1,
                      "y": 1.5
                    }
                  }
                },
                {
                  "Type": "Bathroom",
                  "Name": "Bathroom",
                  "DesiredShape": {
                    "Area": 10,
                    "AspectRatio": {
                      "x": 1,
```

```
                    "y": 1.5
                  }
                }
              },
              {
                "Type": "Kitchen",
                "Name": "Kitchen",
                "DesiredShape": {
                  "Area": 15,
                  "AspectRatio": {
                    "x": 1.5,
                    "y": 1
                  }
                }
              }
            ],
            "DesiredConnections": [
              {
                "Location": [
                  "Bathroom",
                  "Living room"
                ],
                "ConnectionType": "Doorway"
              },
              {
                "Location": [
                  "Living room",
                  "Kitchen"
                ],
                "ConnectionType": "Opening"
              },
              {
                "Location": [
                  "Bedroom",
                  "Living room"
                ],
                "ConnectionType": "Doorway"
              }
            ]
          },
          "Interior": {
            "Flooring": {
              "MaterialSets": [
                {
```

```
                "Name": "Floorboard room types",
                "TargetSet": {
                  "RoomTypes": [
                    "Kitchen"
                  ]
                },
                "SampleSet": {
                  "MaterialTypes": [
                    "Floorboards"
                  ]
                }
              },
              {
                "Name": "Carpet room types",
                "TargetSet": {
                  "RoomTypes": [
                    "Living",
                    "Bedroom"
                  ]
                },
                "SampleSet": {
                  "MaterialTypes": [
                    "Carpet"
                  ]
                }
              },
              {
                "Name": "Bathroom",
                "TargetSet": {
                  "RoomNames": [
                    "Bathroom"
                  ]
                },
                "SampleSet": {
                  "MaterialTypes": [
                    "Parquetry"
                  ]
                }
              }
            ]
          },
          "Walls": {
            "MaterialSets": [
              {
```

```
                "Name": "Brick room types",
                "TargetSet": {
                  "RoomTypes": [
                    "Living"
                  ]
                },
                "SampleSet": {
                  "MaterialTypes": [
                    "Brick"
                  ]
                }
              },
              {
                "Name": "Tiles room types",
                "TargetSet": {
                  "RoomTypes": [
                    "Bathroom"
                  ]
                },
                "SampleSet": {
                  "MaterialTypes": [
                    "Tiles"
                  ]
                }
              }
            ]
          },
          "Furniture": {
            "FurnitureArrangements": [
              {
                "Name": "Dense furniture room types",
                "TargetSet": {
                  "RoomTypes": [
                    "Living",
                    "Bedroom",
                    "Kitchen",
                    "Bathroom"
                  ]
                },
                "DesiredSpatialDensity": "Dense"
              }
            ]
          }
        }
```

```
          }
        ]
      }
    ]
  }
}
```

1 ルームのみ

以下の例では、1 ベッドルームハウスを指定します。インテリア家具を指定します。

```
{
  "Version": "2",
  "Buildings": [
    {
      "Floors": [
        {
          "Floorplan": {
            "Footprint": {
              "DesiredAspectRatio": {
                "x": 1,
                "y": 1
              }
            },
            "Ceiling": {
              "Height": 3
            },
            "Rooms": [
              {
                "Type": "Bedroom",
                "Name": "Bedroom",
                "DesiredShape": {
                  "Area": 40,
                  "AspectRatio": {
                    "x": 1,
                    "y": 1.61
                  }
                }
              }
            ],
            "DesiredConnections": []
          },
          "Interior": {
```

```
            "Furniture": {
              "FurnitureArrangements": [
                {
                  "Name": "Bedroom furniture",
                  "TargetSet": {
                    "RoomNames": [
                      "Bedroom"
                    ]
                  },
                  "DesiredSpatialDensity": "Dense"
                }
              ]
            }
          }
        }
      ]
    }
  ]
}
```

2 ルーム

以下の例では、1 ベッドルームハウスを指定します。Simulation WorldForge により、床材、壁材、家具の配置、接続性などの詳細が決定されます。

```
{
  "name": "TwoRooms",
  "templateBody": {
    "Version": "2",
    "Buildings": [
      {
        "Floors": [
          {
            "Floorplan": {
              "Footprint": {
                "DesiredAspectRatio": {
                  "x": 1,
                  "y": 1
                }
              },
              "Ceiling": {
                "Height": 3
              },
```

```
              "Rooms": [
                {
                  "Type": "Living",
                  "Name": "Living room",
                  "DesiredShape": {
                    "Area": 30,
                    "AspectRatio": {
                      "x": 1,
                      "y": 1.5
                    }
                  }
                },
                {
                  "Type": "Dining",
                  "Name": "Dining room",
                  "DesiredShape": {
                    "Area": 30,
                    "AspectRatio": {
                      "x": 1,
                      "y": 1.5
                    }
                  }
                }
              ],
              "DesiredConnections": []
            },
            "Interior": {}
          }
        ]
      }
    ]
  }
}
```

シミュレーションワールドテンプレートの管理

このセクションでは、シミュレーションワールドテンプレートの作成方法と管理方法について説明します。シミュレーションワールドテンプレートを使用して、Simulation WorldForgeによるワールドの生成方法を指定します。部屋の数、部屋の接続方法、家具、インテリア要素を指定できます。

シミュレーションワールドテンプレートの詳細について、まず シミュレーションワールドテンプレートについて から説明します。JSON `templateBody` でもシミュレーションワールドテンプレー

トの説明を確認できます。詳細については、「シミュレーションワールドテンプレート本文の JSON スキーマ」を参照してください。

トピック

- シミュレーションワールドテンプレートの作成
- シミュレーションワールドテンプレートの表示
- シミュレーションワールドテンプレートの修正
- シミュレーションワールドテンプレートの削除
- シミュレーションワールドテンプレートのバージョン、機能、および変更

シミュレーションワールドテンプレートの作成

シミュレーションワールドテンプレートを作成して、Simulation WorldForge によるワールドの生成方法を指定します。シミュレーションワールドテンプレートが完成したら、ワールド生成ジョブを作成して、さまざまな部屋設定とインテリア設定があるワールドを生成します。

シミュレーションワールドテンプレートは、サンプルテンプレートや保存されたテンプレートから作成するか、または一から作成することもできます。テンプレートを作成した後でも、間取り図、インテリア、およびその他の詳細を修正できます。シミュレーションワールドテンプレートの変更の詳細については、「シミュレーションワールドテンプレートの修正」を参照してください。

シミュレーションワールドテンプレートを作成する方法

以下のいずれかのタブのステップに従ってください。

Using the console

シミュレーションワールドテンプレートを作成する方法

1. https://console.aws.amazon.com/robomaker/ で AWS RoboMaker コンソールにサインインします。
2. AWS RoboMaker コンソールで、左にある [Simulation WorldForge] を展開して [ワールドテンプレート] を選択します。
3. [World templates] (ワールドテンプレート) ページで [Create template] (テンプレートの作成) を選択します。

4. [Create a world template] (ワールドテンプレートの作成) ページで。テンプレートオプションのうちの 1 つを選択します。あらかじめ設定されている [サンプルテンプレート] から 1 つ選択したり、[保存されたテンプレート] をクローン化して修正したり、デフォルトのワールドを使って [一から作成] したりすることができます。
5. [Template detail] (テンプレートの詳細) ページの左上にある [Rename] (名前の変更) を選択して、テンプレートの名前を指定します。
6. (オプション) 間取り図とインテリアの詳細をカスタマイズします。詳細については、「シミュレーションワールドテンプレートについて」を参照してください。
7. [Template detail] (テンプレートの詳細) ページで [Save and exit] (保存して終了) を選択します。

Using the AWS CLI

Example

AWS CLI を使用してシミュレーションワールドテンプレートを更新することもできます。まず、Simulation WorldForge により生成されるワールドを指定する JSON ドキュメントを作成します。次に、`create-world-template` を使用してシミュレーションワールドテンプレートを作成します。

例えば、次の JSON ドキュメントでは 1 ベッドルームハウスを指定します。

```
{
  "title": "WorldTemplate",
  "description": "The top-level template for parameterizing a randomly generated
 world. By default, a single\nresidential building with one floor and one room is
 generated.",
  "type": "object",
  "properties": {
    "Version": {
      "title": "Version",
      "default": "1",
      "type": "string"
    },
    "Buildings": {
      "title": "Buildings",
      "default": [
        {
          "Floors": [
```

```
            {
              "Floorplan": {
                "Footprint": {
                  "DesiredAspectRatio": {
                    "x": 1.0,
                    "y": 1.0
                  }
                },
                "Ceiling": {
                  "Height": 3.0
                },
                "Rooms": [
                  {
                    "Type": "Living",
                    "Name": "My Living Room",
                    "DesiredShape": {
                      "Area": 20.0,
                      "AspectRatio": {
                        "x": 1.0,
                        "y": 1.0
                      }
                    }
                  }
                ],
                "DesiredConnections": []
              },
              "Interior": {
                "Flooring": {
                  "MaterialSets": []
                },
                "Walls": {
                  "MaterialSets": []
                },
                "Furniture": {
                  "FurnitureArrangements": []
                }
              }
            }
          ]
        }
      ],
      "type": "array",
      "items": {
        "$ref": "#/definitions/BuildingTemplate"
```

```
      },
      "minItems": 1,
      "maxItems": 1
    }
  },
  "additionalProperties": false,
  "definitions": {
    "AspectRatio": {
      "title": "AspectRatio",
      "type": "object",
      "properties": {
        "x": {
          "title": "X",
          "default": 1,
          "minimum": 1,
          "maximum": 4,
          "type": "number"
        },
        "y": {
          "title": "Y",
          "default": 1,
          "minimum": 1,
          "maximum": 4,
          "type": "number"
        }
      },
      "additionalProperties": false
    },
    "FloorplanFootprint": {
      "title": "FloorplanFootprint",
      "description": "The desired footprint of this floorplan.",
      "type": "object",
      "properties": {
        "DesiredAspectRatio": {
          "title": "Desiredaspectratio",
          "default": {
            "x": 1.0,
            "y": 1.0
          },
          "allOf": [
            {
              "$ref": "#/definitions/AspectRatio"
            }
          ]
```

```
        }
      },
      "additionalProperties": false
    },
    "FloorplanCeiling": {
      "title": "FloorplanCeiling",
      "description": "The height of the ceiling for this floorplan in metres.",
      "type": "object",
      "properties": {
        "Height": {
          "title": "Height",
          "default": 3.0,
          "type": "number",
          "minimum": 2.4,
          "maximum": 4.0
        }
      },
      "additionalProperties": false
    },
    "Rectangle": {
      "title": "Rectangle",
      "description": "A rectangle defined by area in square metres and aspect
ratio.",
      "type": "object",
      "properties": {
        "Area": {
          "title": "Area",
          "type": "number"
        },
        "AspectRatio": {
          "$ref": "#/definitions/AspectRatio"
        }
      },
      "required": [
        "Area",
        "AspectRatio"
      ],
      "additionalProperties": false
    },
    "FloorplanRoom": {
      "title": "FloorplanRoom",
      "description": "A description for single room for this floorplan.",
      "type": "object",
      "properties": {
```

```
        "Type": {
          "title": "Type",
          "enum": [
            "Bedroom",
            "Bathroom",
            "Living",
            "Dining",
            "Kitchen",
            "Hallway",
            "Closet"
          ],
          "type": "string"
        },
        "Name": {
          "title": "Name",
          "maxLength": 255,
          "minLength": 1,
          "pattern": "^[a-zA-Z0-9_\\- ]*$",
          "type": "string"
        },
        "DesiredShape": {
          "title": "Desiredshape",
          "default": {
            "Area": 20.0,
            "AspectRatio": {
              "x": 1.0,
              "y": 1.0
            }
          },
          "allOf": [
            {
              "$ref": "#/definitions/Rectangle"
            }
          ]
        }
      },
      "required": [
        "Type",
        "Name"
      ],
      "additionalProperties": false
    },
    "FloorplanConnection": {
      "title": "FloorplanConnection",
```

```
      "description": "Descibes the desired layout of the rooms and their adjacent
 rooms. A connection can be either a doorway or \nan open space without any walls.
 Two rooms cannot both share an interior doorway and an opening. \nThe same two
 rooms can have multiple doorways, up to a limit.",
      "type": "object",
      "properties": {
        "Location": {
          "title": "Location",
          "type": "array",
          "items": {
            "type": "string"
          },
          "minItems": 2,
          "maxItems": 2
        },
        "ConnectionType": {
          "title": "Connectiontype",
          "enum": [
            "Doorway",
            "Opening"
          ],
          "type": "string"
        }
      },
      "required": [
        "Location",
        "ConnectionType"
      ],
      "additionalProperties": false
    },
    "FloorplanTemplate": {
      "title": "FloorplanTemplate",
      "description": "The top-level floorplan template that parameterizes the
 randomly generated \narchitectural layout. By default, a residential floorplan
 with bedroom and \nliving room are generated with a random doorway or opening
 connection. \n\nThe footprint contributes to the overall shape of the floor layout
 along\nwith rooms. The footprint shape is desired as it is a preference and not
\nguaranteed.\n\nThe ceiling determines the height of the walls. There are minimum
 and\nmaximum ceiling heights. The ceiling height is guaranteed.\n\nRooms are
 required. Each room has a desired shape. Together, the room\nshapes and footprint
 determine floor layout. The room types contribute to\nthe layout and are used
 when randomly selecting furniture and materials for\nthe walls and floors.\n
\nDesiredConnections are optional. Two rooms are connected if they share a\nwall
 and doorway or adjacent without any wall aka \"opening\". All rooms are\nguaranteed
```

```
 to be connected randomly if they are not specified in the\nconnections list.
 Connections that are specified are _not_ guaranteed but\nwill be attempted as best-
effort.",
      "type": "object",
      "properties": {
        "Footprint": {
          "title": "Footprint",
          "default": {
            "DesiredAspectRatio": {
              "x": 1.0,
              "y": 1.0
            }
          },
          "allOf": [
            {
              "$ref": "#/definitions/FloorplanFootprint"
            }
          ]
        },
        "Ceiling": {
          "title": "Ceiling",
          "default": {
            "Height": 3.0
          },
          "allOf": [
            {
              "$ref": "#/definitions/FloorplanCeiling"
            }
          ]
        },
        "Rooms": {
          "title": "Rooms",
          "default": [
            {
              "Type": "Living",
              "Name": "My Living Room",
              "DesiredShape": {
                "Area": 20.0,
                "AspectRatio": {
                  "x": 1.0,
                  "y": 1.0
                }
              }
            }
```

```
          ],
          "type": "array",
          "items": {
            "$ref": "#/definitions/FloorplanRoom"
          },
          "minItems": 1,
          "maxItems": 6
        },
        "DesiredConnections": {
          "title": "Desiredconnections",
          "default": [],
          "type": "array",
          "items": {
            "$ref": "#/definitions/FloorplanConnection"
          },
          "minItems": 0,
          "maxItems": 12
        }
      },
      "additionalProperties": false
    },
    "RoomNameList": {
      "title": "RoomNameList",
      "description": "The set of all rooms matching any of the listed room names.",
      "type": "object",
      "properties": {
        "RoomNames": {
          "title": "Roomnames",
          "type": "array",
          "items": {
            "type": "string"
          },
          "minItems": 1,
          "maxItems": 6
        }
      },
      "required": [
        "RoomNames"
      ],
      "additionalProperties": false
    },
    "RoomTypeList": {
      "title": "RoomTypeList",
      "description": "The set of all rooms matching any of the listed room types.",
```

```
      "type": "object",
      "properties": {
        "RoomTypes": {
          "title": "Roomtypes",
          "type": "array",
          "items": {
            "enum": [
              "Bedroom",
              "Bathroom",
              "Living",
              "Dining",
              "Kitchen",
              "Hallway",
              "Closet"
            ],
            "type": "string"
          },
          "minItems": 1,
          "maxItems": 7
        }
      },
      "required": [
        "RoomTypes"
      ],
      "additionalProperties": false
    },
    "MaterialSetByMaterialType": {
      "title": "MaterialSetByMaterialType",
      "description": "The set of materials that match any of the material types
listed.  An empty\nset is invalid since all targets require materials.",
      "type": "object",
      "properties": {
        "MaterialTypes": {
          "title": "Materialtypes",
          "type": "array",
          "items": {
            "type": "string"
          },
          "minItems": 1
        }
      },
      "required": [
        "MaterialTypes"
      ],
```

```
      "additionalProperties": false
    },
    "InteriorMaterialSet": {
      "title": "InteriorMaterialSet",
      "description": "A set of sample materials to randomly assign to a set of
 interior target elements.\n\nThe target set determines *what rooms* receive the
 materials in the sample\nset. The targets in a room are the walls and flooring.
 Rooms may be targeted\nby room type or room name.\n\nThe sample set determines
 *what materials* to randomly select for the\ntarget rooms' walls and floors.\n
\nThe sample set is optional and when not specified (null) materials are\nrandomly
 selected according to the room type for each room in the target\nset.\n\nA sample
 set with an empty material set is invalid since all wall\nand flooring targets
 require materials.",
      "type": "object",
      "properties": {
        "Name": {
          "title": "Name",
          "maxLength": 255,
          "minLength": 1,
          "pattern": "^[a-zA-Z0-9_\\- ]*$",
          "type": "string"
        },
        "TargetSet": {
          "title": "Targetset",
          "anyOf": [
            {
              "$ref": "#/definitions/RoomNameList"
            },
            {
              "$ref": "#/definitions/RoomTypeList"
            }
          ]
        },
        "SampleSet": {
          "$ref": "#/definitions/MaterialSetByMaterialType"
        }
      },
      "required": [
        "Name",
        "TargetSet"
      ],
      "additionalProperties": false
    },
    "InteriorFlooring": {
```

```
      "title": "InteriorFlooring",
      "description": "Describes the interior template parameters for all floors
 for this floorplan.\nAll floors not explicitly targeted will have a random floor
 material assigned by room type.",
      "type": "object",
      "properties": {
        "MaterialSets": {
          "title": "Materialsets",
          "default": [],
          "type": "array",
          "items": {
            "$ref": "#/definitions/InteriorMaterialSet"
          },
          "minItems": 0,
          "maxItems": 6
        }
      },
      "additionalProperties": false
    },
    "InteriorWalls": {
      "title": "InteriorWalls",
      "description": "Describes the interior template parameters for all walls for
 this floorplan.\nAll walls not explicitly targeted will have a random wall material
 assigned by room type.",
      "type": "object",
      "properties": {
        "MaterialSets": {
          "title": "Materialsets",
          "default": [],
          "type": "array",
          "items": {
            "$ref": "#/definitions/InteriorMaterialSet"
          },
          "minItems": 0,
          "maxItems": 6
        }
      },
      "additionalProperties": false
    },
    "ModelTypeList": {
      "title": "ModelTypeList",
      "description": "The set of all models matching any of the listed model types.
\nAn empty set means zero models to sample/select.",
      "type": "object",
```

```
      "properties": {
        "ModelTypes": {
          "title": "Modeltypes",
          "type": "array",
          "items": {
            "enum": [
              "Baths",
              "BarCabinets",
              "Beds",
              "Bookcases",
              "CoffeeTables",
              "ConsoleTables",
              "CornerCabinets",
              "DeskChairs",
              "Desks",
              "DiningChairs",
              "DiningTables",
              "DishWashers",
              "Dressers",
              "EndAndSideTables",
              "FloorLamps",
              "Fridges",
              "LivingRoomChairs",
              "KitchenIslandsAndCarts",
              "MediaStorage",
              "Nightstands",
              "Ottomans",
              "Ovens",
              "ServingCarts",
              "Showers",
              "SideboardsAndBuffets",
              "Sofas",
              "Storage",
              "StorageBenches",
              "Toilets",
              "VanityCounters",
              "WashingMachinesAndDryers"
            ],
            "type": "string"
          },
          "minItems": 0
        }
      },
      "required": [
```

```
        "ModelTypes"
      ],
      "additionalProperties": false
    },
    "FurnitureArrangementSet": {
      "title": "FurnitureArrangementSet",
      "description": "Describes the interior template for placing furniture in one
or more rooms.\n\n- TargetSet is the set of rooms to furnish, filter by room name
or room\n  type.\n- SampleSet is a set of all furnishing models to randomly choose
and\n  place.\n- DesiredSpatialDensity is the desired level of free space after
placing\n  furniture.",
      "type": "object",
      "properties": {
        "Name": {
          "title": "Name",
          "maxLength": 255,
          "minLength": 1,
          "pattern": "^[a-zA-Z0-9_\\- ]*$",
          "type": "string"
        },
        "TargetSet": {
          "title": "Targetset",
          "anyOf": [
            {
              "$ref": "#/definitions/RoomNameList"
            },
            {
              "$ref": "#/definitions/RoomTypeList"
            }
          ]
        },
        "SampleSet": {
          "$ref": "#/definitions/ModelTypeList"
        },
        "DesiredSpatialDensity": {
          "title": "Desiredspatialdensity",
          "default": "Moderate",
          "enum": [
            "Sparse",
            "Moderate",
            "Dense"
          ],
          "type": "string"
        }
```

```
      },
      "required": [
        "Name",
        "TargetSet"
      ],
      "additionalProperties": false
    },
    "InteriorFurnishings": {
      "title": "InteriorFurnishings",
      "description": "Describes the types of furniture models for randomly placing
into each room\nin the world.  Rooms are targeted by room type or room name. Rooms
that are\nnot targeted are furnished at random by their room type with moderate
density.\ndensity. For an empty room, specify an empty sample set.",
      "type": "object",
      "properties": {
        "FurnitureArrangements": {
          "title": "Furniturearrangements",
          "default": [],
          "type": "array",
          "items": {
            "$ref": "#/definitions/FurnitureArrangementSet"
          },
          "minItems": 0,
          "maxItems": 6
        }
      },
      "additionalProperties": false
    },
    "InteriorTemplate": {
      "title": "InteriorTemplate",
      "description": "Top-level template for parameterizing the interior finishes
and furnishings for\nthis floorplan.",
      "type": "object",
      "properties": {
        "Flooring": {
          "title": "Flooring",
          "default": {
            "MaterialSets": []
          },
          "allOf": [
            {
              "$ref": "#/definitions/InteriorFlooring"
            }
          ]
```

```
        },
        "Walls": {
          "title": "Walls",
          "default": {
            "MaterialSets": []
          },
          "allOf": [
            {
              "$ref": "#/definitions/InteriorWalls"
            }
          ]
        },
        "Furniture": {
          "title": "Furniture",
          "default": {
            "FurnitureArrangements": []
          },
          "allOf": [
            {
              "$ref": "#/definitions/InteriorFurnishings"
            }
          ]
        }
      },
      "additionalProperties": false
    },
    "FloorTemplate": {
      "title": "FloorTemplate",
      "description": "Describes a single foor within a building. Defaults to a
single residential room\nof a randomy type and size, and the interior is randomly
furnished.",
      "type": "object",
      "properties": {
        "Floorplan": {
          "title": "Floorplan",
          "default": {
            "Footprint": {
              "DesiredAspectRatio": {
                "x": 1.0,
                "y": 1.0
              }
            },
            "Ceiling": {
              "Height": 3.0
```

```
            },
            "Rooms": [
              {
                "Type": "Living",
                "Name": "My Living Room",
                "DesiredShape": {
                  "Area": 20.0,
                  "AspectRatio": {
                    "x": 1.0,
                    "y": 1.0
                  }
                }
              }
            ],
            "DesiredConnections": []
          },
          "allOf": [
            {
              "$ref": "#/definitions/FloorplanTemplate"
            }
          ]
        },
        "Interior": {
          "title": "Interior",
          "default": {
            "Flooring": {
              "MaterialSets": []
            },
            "Walls": {
              "MaterialSets": []
            },
            "Furniture": {
              "FurnitureArrangements": []
            }
          },
          "allOf": [
            {
              "$ref": "#/definitions/InteriorTemplate"
            }
          ]
        }
      },
      "additionalProperties": false
    },
```

```
   "BuildingTemplate": {
     "title": "BuildingTemplate",
     "description": "Describes a building to be randomly generated. Defaults to one
residential floor.",
     "type": "object",
     "properties": {
       "Floors": {
         "title": "Floors",
         "default": [
           {
             "Floorplan": {
               "Footprint": {
                 "DesiredAspectRatio": {
                   "x": 1.0,
                   "y": 1.0
                 }
               },
               "Ceiling": {
                 "Height": 3.0
               },
               "Rooms": [
                 {
                   "Type": "Living",
                   "Name": "My Living Room",
                   "DesiredShape": {
                     "Area": 20.0,
                     "AspectRatio": {
                       "x": 1.0,
                       "y": 1.0
                     }
                   }
                 }
               ],
               "DesiredConnections": []
             },
             "Interior": {
               "Flooring": {
                 "MaterialSets": []
               },
               "Walls": {
                 "MaterialSets": []
               },
               "Furniture": {
                 "FurnitureArrangements": []
```

```
                }
              }
            }
          ],
          "type": "array",
          "items": {
            "$ref": "#/definitions/FloorTemplate"
          },
          "minItems": 1,
          "maxItems": 1
        }
      },
      "additionalProperties": false
    }
  }
}
```

JSON を one-bedroom-house.json という名前のファイルに保存した場合、それを AWS CLI とともに使用すればシミュレーションワールドテンプレートを作成できます。

```
$ aws robomaker create-world-template --template my-simulation-world-template-arn --
template-body file://one-bedroom-house.json
```

シミュレーションワールドテンプレートの表示

シミュレーションワールドテンプレートの詳細を表示します。

シミュレーションワールドテンプレートの詳細を表示する方法

以下のいずれかのタブのステップに従ってください。

Using the console

1. https://console.aws.amazon.com/robomaker/ で AWS RoboMaker コンソールにサインインします。
2. 左側のナビゲーションペインで、[Simulation WorldForge]、[World templates] (ワールドテンプレート) の順に選択します。
3. 間取り図とインテリアを含む詳細を表示するシミュレーションワールドテンプレートの ID を選択します。詳細ビューからワールドを生成することもできます。

Using the AWS CLI

Example

以下の AWS CLI の例では、`list-world-templates` を使用して既存のテンプレートを一覧表示し、次に `describe-world-template` と `get-world-template-body` を使用してシミュレーションワールドテンプレートの詳細を表示します。

```
$ aws robomaker list-world-templates
$ aws robomaker describe-world-template --template my-simulation-world-template-arn
$ aws robomaker get-world-template-body --template my-simulation-world-template-arn
```

シミュレーションワールドテンプレートの修正

間取り図を選択して、部屋の数とタイプや、間取り図内の部屋間の接続をカスタマイズします。インテリアを選択して、床、壁、家具をカスタマイズします。

シミュレーションワールドテンプレートを修正する方法

以下のいずれかのタブのステップに従ってください。

Using the console

シミュレーションワールドテンプレートを修正する方法

1. https://console.aws.amazon.com/robomaker/ で AWS RoboMaker コンソールにサインインします。
2. AWS RoboMaker コンソールで、左のナビゲーションペインにある [Simulation WorldForge] を展開して [ワールドテンプレート] を選択します。
3. [World templates] (ワールドテンプレート) ページで、修正するシミュレーションワールドテンプレートを選択します。
4. 修正する各要素の横にある [Edit] (編集) または [Override] (オーバーライド) を選択します。シミュレーションワールドテンプレートのコンポーネントの詳細については、「シミュレーションワールドテンプレートについて」を参照してください。

Using the AWS CLI

Example

以下の AWS CLI の例では、`list-world-templates` を使用して既存のテンプレートを一覧表示し、次に `describe-world-template` を使用してシミュレーションワールドテンプレートの詳細を表示し、`get-world-template-body` を使用してテンプレート本文 JSON を取得してそれをファイルに書き込みます。

```
$ aws robomaker list-world-templates
$ aws robomaker describe-world-template --template my-simulation-world-template-arn
$ aws robomaker get-world-template-body --template my-simulation-world-template-arn
 --output json > myTemplateBody.json
$ aws robomaker update-world-template-body --template my-simulation-world-template-
arn --template-body file://myTemplateBody.json
```

シミュレーションワールドテンプレートの削除

不要になったシミュレーションワールドテンプレートは削除できます。

Using the console

1. https://console.aws.amazon.com/robomaker/ で AWS RoboMaker コンソールにサインインします。
2. 左側のナビゲーションペインで、[Simulation WorldForge]、[World templates] (ワールドテンプレート) の順に選択します。
3. シミュレーションワールドテンプレートの [Id] を選択し、[Template actions] (テンプレートアクション) を選択し、[Delete] (削除) を選択して、ダイアログボックスの [Delete] (削除) を選択することで削除を確定します。

Using the AWS CLI

Example

以下の AWS CLI の例では、`list-world-templates` を使用して既存のテンプレートを一覧表示し、次に `delete-world-template` を使用してシミュレーションワールドテンプレートを削除します。

```
$ aws robomaker list-world-templates
$ aws robomaker delete-world-template --template my-simulation-world-template-arn
```

シミュレーションワールドテンプレートのバージョン、機能、および変更

AWS RoboMaker Simulation WorldForge は、ワールドテンプレートの新しいバージョンをリリースします。これらのテンプレートの新機能と改善点を利用すれば、ユース ケースに適したワールドを作成できます。

ワールドテンプレートのすべての機能を使用するには、ワールドテンプレートを最新バージョンにアップグレードします。ワールドテンプレートの最新バージョンには、過去のバージョンに存在するすべての機能が備わっています。

AWS RoboMaker コンソールまたは AWS CLI のいずれかを使ってワールドテンプレートを更新することができます。AWS RoboMaker コンソールを使用している場合は、テンプレートのアップグレードに使用できるプロンプトが表示されます。

API を使用してワールドテンプレートを最新バージョンにアップグレードするには、ワールドテンプレートを定義する JSON の `Version` フィールドを最新バージョンの数値に設定します。例えば、バージョン 2 が最新バージョンである場合は、ワールドテンプレートの本文の `"Version": "2"` を指定します。最新のスキーマの表示方法については、「シミュレーションワールドテンプレート本文の JSON スキーマ」を参照してください。

ここからは、ワールドテンプレートの機能と更新に関して説明します。まず、最新バージョンの更新が表示されます。

シミュレーションワールドテンプレートのバージョン 2 のリリース

バージョン 2 の更新には以下が含まれます。

- ワールドにヒンジ付きドアを追加する機能
- すべての部屋に設定を適用する機能
- ワールドを説明する新しいフィールド

- 床摩擦値に対する変更
- バージョンに依存しない更新

ドア

AWS RoboMaker Simulation WorldForge テンプレートのバージョン 2 を使用すれば、ヒンジ付きのドアがあるワールドを作成できます。

これらのドアの開放度を設定できます。例として、指定できるいくつかの開放状態を以下に示します。

- 0% 開放 — 閉鎖
- 50% 開放 — 半開
- 70% 開放 — ほぼ全開
- 100% 開放 — 全開

また、開放度をランダムな状態に設定することで、Simulation WorldForge でドアの開放度がランダムに設定されるように指定することもできます。

ワールドテンプレートの `Interior` セクションで、ワールドで表示するドアを設定できます。ワールドテンプレートを使用してドア付きの部屋を作成する方法については、「出入口のドアありをリクエストする」を参照してください。

すべての部屋への設定の適用

ワールドテンプレートの `Target.All` キーワードを使用すれば設定変更をすべての部屋に適用できます。全ての部屋には変更できる事柄がいくつかあります。

- 床材
- 壁材
- 出入口
- 家具の配置

例えば、ワールドテンプレートですべてのドアの閉鎖を指定する場合は、ドアの開放度を 0 % に指定し、`Target.All` キーワードを使用してその条件をすべてのドアに適用することができます。詳細については、「すべての部屋への設定の適用」を参照してください。

ワールドを説明する新しいフィールド

バージョン 2 テンプレートとともに作成されるワールドには world_description.json ファイルがあります。このファイルは、Gazebo WorldForge .world ファイルと同じディレクトリに表示されます。

world_description.json ファイルには、Simulation WorldForge ワールド内のすべてのドアが一覧表示されます。DescribeWorld オペレーションを使用すればワールドの説明を確認できます。その説明は worldDescriptionBody フィールドの値です。バージョン 1 テンプレートを使用してワールドを作成した場合、フィールド の値は空です。

床摩擦値に対するバージョン 2 の変更

バージョン 2 では、床の床摩擦値は Gazebo 基面と同じです。バージョン 1 の床摩擦値は変更されません。

バージョンに依存しない更新

すべてのワールドテンプレートについて、Gazebo モデル名では部屋名中のスペースが下線に置き換えられます。この変更により、すべての Simulation WorldForge Gazebo モデルに ROS トピックを使用できるようになりました。ROS トピックを使用すれば、モデルに関する情報の取得やモデルに対する変更が可能です。

ワールド生成ジョブの管理

ワールド生成ジョブを使用して、シミュレーションワールドテンプレートからワールドを生成します。ワールド生成ジョブを作成するときに、異なる間取り図とインテリア設定の数を指定します。1 つのワールド生成ジョブでワールドを 50 個まで生成できます。

トピック

- ワールド生成ジョブの作成
- ワールド生成ジョブの表示
- ワールド生成ジョブのキャンセル

ワールド生成ジョブの作成

ワールド生成ジョブを作成して、さまざまな部屋設定とインテリア設定があるワールドを生成します。1 つのワールド生成ジョブでワールドを 50 個まで生成できます。

ワールド生成ジョブの作成方法

以下のいずれかのタブのステップに従ってください。

Using the console

シミュレーションワールドテンプレートを作成する方法

1. https://console.aws.amazon.com/robomaker/ で AWS RoboMaker コンソールにサインインします。
2. AWS RoboMaker コンソールで、左にある [Simulation WorldForge] を展開して [ワールドテンプレート] を選択します。
3. [World templates] (ワールドテンプレート) ページで、ワールドの生成に使用するシミュレーションワールドテンプレートを選択し、次に [Generate worlds] (ワールドの生成) を選択します。
4. [Generate worlds] (ワールドの生成) ページで [Number of floor plans] (間取り図の数) を指定します。間取り図の数に間取り図 1 つ当たりのインテリアバリエーションの数を掛けた数が 50 を超えないようにしてください。
5. 間取り図 1 つ当たりのインテリアバリエーションの数を指定します。間取り図の数に間取り図 1 つ当たりのインテリアバリエーションの数を掛けた数が 50 を超えないようにしてください。
6. オプション: 生成するすべてのワールドに割り当てられるワールドタグを追加します。
7. オプション: 生成ジョブに割り当てられる生成ジョブタグを追加します。これらのタグは、生成するワールドには適用されません。
8. [Generate] (生成) を選択します。

   [World generation job detail] (ワールド生成ジョブの詳細) ページでワールド生成ジョブの進捗状況を追跡できます。ワールドの生成に必要な時間は、シミュレーションワールドテンプレートの複雑さと生成するワールドの数によって異なります。

Using the AWS CLI

Example

AWS CLI を使えば、シミュレーションワールドテンプレートからワールドを生成できます。`create-world-generation-job` を使用してワールド生成ジョブを作成します。

次の AWS CLI の例は 2 つの異なるインテリア間取り図を含む 2 つの間取り図があるワールドを 4 つ生成する方法を示しています。

```
$ aws robomaker list-world-templates
$ aws robomaker create-world-generation-job --template my-simulation-world-template-arn --worldCount floorplanCount=2,interiorCountPerFloorplan=2
$ aws robomaker list-world-generation-jobs
$ aws robomaker describe-world-generation-job --job my-world-generation-job-arn
```

ワールド生成ジョブの表示

ワールド生成の進捗状況、概要情報、およびワールド生成ジョブに関するその他の詳細を表示することができます。

ワールド生成ジョブの詳細を表示する方法

以下のいずれかのタブのステップに従ってください。

Using the console

1. https://console.aws.amazon.com/robomaker/ で AWS RoboMaker コンソールにサインインします。
2. 左側のナビゲーションペインで、[Simulation WorldForge]、[World templates] (ワールドテンプレート) の順に選択します。
3. 詳細を表示するワールド生成ジョブの ID を選択します。検索バーを使用すれば生成ジョブを検索できます。

Using the AWS CLI

Example

次の AWS CLI の例では、`list-world-generation-jobs` を使用して既存のワールド生成ジョブを一覧表示し、次に `describe-world-generation-job` を使用して特定のワールド生成ジョブの詳細を表示しています。

```
$ aws robomaker list-world-generation-jobs
$ aws robomaker describe-world-generation-job --job my-world-generation-job-arn
```

ワールド生成ジョブのキャンセル

進行中のワールド生成ジョブをキャンセルすることができます。

ワールド生成ジョブをキャンセルする方法

以下のいずれかのタブのステップに従ってください。

Using the console

1. https://console.aws.amazon.com/robomaker/ で AWS RoboMaker コンソールにサインインします。
2. 左側のナビゲーションペインで、[Simulations] (シミュレーション)、[Generation jobs] (生成ジョブ) の順に選択します。
3. [Generation jobs] (生成ジョブ) ページで、キャンセルするワールド生成ジョブを選択します。
4. [Cancel] (キャンセル) を選択します。[Cancel generation job] (生成ジョブのキャンセル) ページで、[Cancel job] (ジョブのキャンセル) を選択してジョブをキャンセルします。

Using the AWS CLI

Example

次の AWS CLI の例では、`list-world-generation-jobs` を使用して既存のワールド生成ジョブを一覧表示し、次に `cancel-world-generation-job` を使用して特定のワールド生成ジョブをキャンセルします。

```
$ aws robomaker list-world-generation-jobs
$ aws robomaker cancel-world-generation-job --job my-world-generation-job-arn
```

ワールドエクスポートジョブの管理

Simulation WorldForge により生成されたワールドをエクスポートして、独自の環境で使用することができます。ワールドは、.zip ファイルで Amazon S3 バケットにエクスポートされます。.zip ファイルには、Gazebo アセットとワールド用の ROS ワークスペースが含まれています。

トピック

ワールドエクスポートジョブの作成

ワールドを選択して Amazon S3 バケットにエクスポートすることができます。エクスポート用に選択したすべてのワールドが 1 つの .zip ファイルに入ります。

ワールドエクスポートジョブを作成する方法

以下のいずれかのタブのステップに従ってください。

Using the console

1 つのエクスポートジョブでワールドを 1 つエクスポートできます。

シミュレーションワールドテンプレートを作成する方法

1. https://console.aws.amazon.com/robomaker/ で AWS RoboMaker コンソールにサインインします。
2. AWS RoboMaker コンソールで、左のナビゲーションペインにある [Simulation WorldForge] を展開して [ワールド] を選択します。
3. [World] (ワールド) ページで [Create export job] (エクスポートジョブの作成) を選択します。
4. [Create export job] (エクスポートジョブの作成) ページでエクスポートするワールドを選択します。

5. PutObject を含む IAM ロール、GetObject、Amazon S3 バケットへの AbortMultipartUpload アクセス許可を選択します。[Create] (作成) を選択し、自分のために作成された適切なアクセス許可が含まれているロールを持ちます。
6. ワールド出力の S3 送信先を選択します。ページの下部付近にある [Create new S3 bucket] (新しい S3 バケットの作成) を選択して新しい Amazon S3 バケットを作成することもできます。
7. オプション: [エクスポートジョブの作成] ページで、エクスポートされたワールドに割り当てられるタグを追加します。
8. [Create] (作成) を選択してワールドエクスポートジョブを作成します。

   エクスポートジョブの進行状況は、ワールドエクスポートジョブ詳細ページで追跡できます。ジョブの作成後に自動的にそのページに移動します。

Using the AWS CLI

Example

AWS CLI を使用すればワールドをエクスポートすることができます。create-world-export-job を使用してワールドエクスポートジョブを作成します。1 つのエクスポートジョブでワールドを 1 つエクスポートできます。

次の AWS CLI の例はワールドのエクスポート方法を示しています。まず、list-worlds を使用してワールドを一覧表示し、次にワールド Amazon リソースネーム (ARN) を指定する create-world-export-job を呼び出します。list-world-export-jobs と describe-world-export-job を呼び出せばステータスを確認できます。

```
aws robomaker list-worlds
aws robomaker create-world-export-job --worlds my-simulation-world-arn --iam-role
 my-iam-role-arn --outputLocation s3Bucket=my-bucket,s3prefix=prefix
aws robomaker list-world-export-jobs
aws robomaker describe-world-export-job --job my-world-export-job-arn
```

ワールドエクスポートジョブの表示

ワールドエクスポートジョブのステータスおよびその他の詳細を表示します。

ワールドエクスポートジョブの詳細を表示する方法

以下のいずれかのタブのステップに従ってください。

Using the console

1. https://console.aws.amazon.com/robomaker/ で AWS RoboMaker コンソールにサインインします。
2. 左側のナビゲーションペインで、[Simulation WorldForge]、[Export jobs] (ジョブのエクスポート) の順に選択します。
3. 詳細を表示するワールドエクスポートジョブの ID を選択します。ワールドのエクスポートジョブの検索やキャンセルも可能です。

Using the AWS CLI

Example

次の AWS CLI の例では、`list-world-export-jobs` を使用して既存のワールドエクスポートジョブを一覧表示し、次に `describe-world-export-job` を使用して特定のワールドエクスポートジョブの詳細を表示します。

```
aws robomaker list-world-export-jobs
aws robomaker describe-world-export-job --job my-world-export-job-arn
```

シミュレーションでのエクスポートしたワールドの使用

Simulation WorldForge を使用して、AWS RoboMaker で使用するワールドを作成できます。シミュレーションで使用するには、ワールドの作成後にエクスポートする必要があります。ワールドをアップロードしてシミュレーションに使用することもできます。

ワールドをエクスポートすると、以下を利用できるようになります。

- デフォルトの SDF 物理演算とは異なる物理演算
- 特殊照明
- カスタムモデル

このセクションでは、生成されたワールドをシミュレーションで使用する方法について詳しく説明します。

> **Important**
>
> AWS RoboMaker の課金の詳細については、「AWS RoboMaker 料金表」を参照してください。

セクション

エクスポートされたワールドをデータソースとして使用する

Simulation WorldForge を使用すると、ROS 環境で使用できるワールドを簡単にエクスポートできます。エクスポートするワールドを選ぶと、Amazon S3 バケット内の 1 つの .zip ファイルにコピーされます。このセクションでは、Amazon S3 バケットにエクスポートされたワールドをシミュレーションジョブで使用する方法について説明します。起動ファイルの調整方法を説明した後、AWS Management Console または CLI のいずれかを使用してシミュレーションジョブを作成します。

AWS Management Console または AWS CLI を使用してデータソースを追加する前に、まずはシミュレーションアプリケーションの起動ファイルを更新する必要があります。

シミュレーションの起動ファイルを更新するには

1. 以下のコマンドを実行します。

```
<launch>
  <!-- Always set GUI to false for AWS RoboMaker Simulation
  Use gui:=true on roslaunch command-line to run with gzclient.
  -->
  <arg name="gui" default="false"/>

  <include file="$(find aws_robomaker_worldforge_worlds)/launch/
launch_world.launch">
  <arg name="gui" value="$(arg gui)"/>
  </include>
```

```
  <!-- Your other launch commands go here. -->
</launch>
```

(0, 0, 0) でロボットをスポーンできます。Simulation WorldForge により生成されたワールドは、(0, 0, 0) で 1 メートルシリンダーがクリアになることが保証されています。

2. イメージを再ビルドして通常どおりにプッシュします。詳細については、「イメージを使用した AWS RoboMaker アプリケーションの開発」を参照してください。

データソースを追加するには

以下のいずれかのタブのステップに従ってください。

Using the console

1. 「ワールドエクスポートジョブの作成」の手順に従ってワールドをエクスポートします。
2. シミュレーションジョブの作成中に、新しいデータソースを追加します。データソースには、例で使用している `WorldForge` など、わかりやすい名前を使用します。
3. オプションで、ワールドを配置する宛先ディレクトリを指定します。

> **Note**
>
> AWS RoboMaker は宛先を dataSource ファイルで上書きするため、ワークスペースディレクトリを宛先として使用しないでください。代わりに、`your_workspace/src/aws_exported_world` など、ワークスペースの下の別ディレクトリを指定できます。

4. タイプに [アーカイブ] を選択します。AWSRoboMaker はワールドを目的のディレクトリに解凍します。
5. [S3 を参照] を選択し、ワールドを生成した正しいエクスポートを探します。
6. シミュレーションジョブの作成を通常どおり続けます。

Using the AWS CLI

Example

「ワールドエクスポートジョブの作成」の手順に従ってワールドをエクスポートします。

ワークスペースが /home/simulation_ws のコンテナにある場合、以下のコマンドでワールドをデフォルトの保存先ディレクトリに抽出します。

```
aws robomaker create-simulation-job \
  --max-job-duration-in-seconds <time> \
  --iam-role <IAM role ARN> \
  --data-sources '[{
    "name": "WorldForge",
    "type": "Archive",
    "destination": "/home/simulation_ws/src/aws_exported_world",
    "s3Bucket": "worldforge-test",
    "s3Keys": ["aws-robomaker-worldforge-export-q376mqk4z7gm.zip"]
  }]' \
  --robot-applications <config> \
  --simulation-applications <config>
```

エクスポートされたワールドを ROS と Gazebo で使用する

前のセクションで説明したように、Simulation WorldForge は選択したワールドを単一の .zip ファイルにエクスポートします。.zip ファイルには、ROS と Gazebo を使用してワールドを変更および視覚化するために必要なすべてのアセットが含まれています。このファイルは次の重要なフォルダが含まれています。

- ルートフォルダ workspace_src は ROS ワークスペースです。これには、共有モデル、ワールドデータ、ワールドのその他の情報が含まれています。ROS 1 および ROS 2 との互換性があります。
- 共有モデルは workspace_src/src/aws_robomaker_worldforge_shared_models/models にコピーされます。例えば、複数のワールドで同一の椅子が使用されている場合、その椅子は共有モデルフォルダに配置されます。
- ワールドデータは workspace_src/src/aws_robomaker_worldforge_worlds/worlds/ にコピーされます。

シミュレーションの起動ファイルを更新するには

1. 「ワールドエクスポートジョブの作成」の手順に従ってワールドをエクスポートします。

2. ワールドを ROS ワークスペースに Unzip します。

```
cd MyApplication/simulation_ws
unzip MyExportedWorld.zip
```

3. ワールドを構築します。

```
rosdep install --from-paths src --ignore-src -r -y$ colcon build
```

4. ワールドを起動します。

```
source install/setup.sh
roslaunch aws_robomaker_worldforge_worlds launch_world.launch gui:=true
```

ワールドを構築して起動する方法は次の通りです。

1. 「ワールドエクスポートジョブの作成」の手順に従ってワールドをエクスポートします。
2. dataSource を使用して、エクスポートしたワールドをワークスペースのソースパッケージディレクトリ /home/simulation_ws/src/aws_exported_world にインポートします。
3. シミュレーションアプリケーションの LaunchConfig を変更します。

```
"launchConfig": {
    "environmentVariables": {
        "ROS_IP": "ROBOMAKER_SIM_APP_IP",
        "ROS_MASTER_URI": "http://ROBOMAKER_ROBOT_APP_IP:11311",
        "GAZEBO_MASTER_URI": "http://ROBOMAKER_SIM_APP_IP:11345",
        "GAZEBO_MODEL_PATH":"@GAZEBO_MODEL_PATH:/home/
simulation_ws/src/aws_exported_world/aws_robomaker_worldforge_pkgs/
aws_robomaker_worldforge_shared_models/models"
    },
    "streamUI": true,
    "command": [
        "/bin/bash", "-c", "cd /home/simulation_ws && colcon build && source
 install/setup.sh && roslaunch hello_world_simulation worldforge_world.launch"
    ]
},
```

エクスポートしたワールドをカスタム物理演算、照明、モデルで使用する

シミュレーションシナリオのカスタマイズが必要な場合は、ワールドをエクスポートして変更を加えることができます。例えば、カスタム物理演算の適用、さまざまな照明効果の適用、カスタムモデルの追加、その他の変更を行うことができます。

ワールドをエクスポートしたら、.world ファイルに変更を加えてエクスポートしたワールドモデルを含める必要があります。.world ファイルでは SDF が使用されます。SDF の詳細については、SDFormat を参照してください。

.world ファイルに変更を加えてエクスポートしたワールドモデルを含める方法

1. 「ワールドエクスポートジョブの作成」の手順に従ってワールドをエクスポートします。
2. 以下のコードを .world ファイルにコピーします。ワールド名がエクスポートされたモデル名と一致していることを確認してください。

```
<sdf version="1.6">
  <world name="generation_82856b0yq33y_world_16">
    <model name="WorldForge World">
      <include>
         <uri>model://generation_82856b0yq33y_world_16</uri>
      </include>
    </model>
    <!-- Your other <world> elements go here -->
  </world>
</sdf>
```

3. 起動ファイルに、変更した .world ファイルが含まれていることを確認します。更新した起動ファイルを使用して、シミュレーションを起動します。

セキュリティ

このセクションでは、AWS RoboMaker のさまざまな側面を保護するためのガイドラインを示します。

トピック

AWS RoboMaker でのデータ保護

AWS 責任共有モデルは、AWS RoboMaker でのデータ保護に適用されます。このモデルで説明されているように、AWS は、AWS クラウド のすべてを実行するグローバルインフラストラクチャを保護するがあります。お客様は、このインフラストラクチャでホストされているコンテンツに対する管理を維持する責任があります。また、使用する AWS のサービスのセキュリティ設定と管理タスクもユーザーの責任となります。データプライバシーの詳細については、「データプライバシーのよくある質問」を参照してください。欧州でのデータ保護の詳細については、「AWS セキュリティブログ」に投稿された「AWS 責任共有モデルおよび GDPR」のブログ記事を参照してください。

データを保護するため、AWS アカウント の認証情報を保護し、AWS IAM Identity Center または AWS Identity and Access Management (IAM) を使用して個々のユーザーをセットアップすることをお勧めします。この方法により、それぞれのジョブを遂行するために必要な権限のみを各ユーザーに付与できます。また、次の方法でデータを保護することをおすすめします。

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。TLS 1.2 が必須です。TLS 1.3 が推奨されます。
- AWS CloudTrail で API とユーザーアクティビティロギングをセットアップします。

- AWS のサービス内でデフォルトである、すべてのセキュリティ管理に加え、AWS の暗号化ソリューションを使用します。
- Amazon Macie などの高度なマネージドセキュリティサービスを使用します。これらは、Amazon S3 に保存されている機密データの検出と保護を支援します。
- コマンドラインインターフェイスまたは API により AWS にアクセスするときに FIPS 140-2 検証済み暗号化モジュールが必要な場合は、FIPS エンドポイントを使用します。利用可能な FIPS エンドポイントの詳細については、「連邦情報処理規格 (FIPS) 140-2」を参照してください。

顧客の E メールアドレスなどの機密情報や重要情報は、タグや Name フィールドなどの自由形式のフィールドに入力しないことを強くお勧めします。これは、AWS RoboMakerを使用する場合や、コンソール、API、AWS CLI、または AWS SDK を採用しているその他の AWS のサービス を使用する場合も同様です。名前に使用する自由記述のテキストフィールドやタグに入力したデータは、課金や診断ログに使用される場合があります。外部サーバーへの URL を提供する場合は、そのサーバーへのリクエストを検証するための認証情報を URL に含めないように強くお勧めします。

AWS RoboMaker の認証とアクセスコントロール

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全にコントロールするために役立つ AWS RoboMaker のサービスです。管理者は IAM を使用して、AWS RoboMaker リソースの使用について認証される (サインイン) 者と認可される (アクセス許可を持つ) 者を管理します。IAM は追加料金なしで提供される AWS アカウントの機能です。

> **Important**
>
> 迅速に使用を開始するには、このページの入門情報を確認し、次に「IAM の使用開始」および「ポリシーとは」を参照してください。

トピック

- 認証とアクセスコントロールの概要
- 必要な許可
- IAM で AWS RoboMaker を使用する方法について
- 認証とアクセスコントロールのトラブルシューティング

認証とアクセスコントロールの概要

AWS RoboMaker は、幅広い機能を提供する AWS Identity and Access Management (IAM) と統合されます。

- AWS アカウント にユーザーとグループを作成します。
- AWS アカウント 内のユーザー間で AWS リソースを簡単に共有できます。
- 各ユーザーに一意のセキュリティ認証情報を割り当てます。
- サービスとリソースへの各ユーザーのアクセス権限を制御します。
- AWS アカウント 内のすべてのユーザーに対する単一の請求書を受け取ります。

IAM の詳細については、以下を参照してください。

- AWS Identity and Access Management (IAM)
- ご利用開始にあたって
- IAM ユーザーガイド

必要な許可

AWS RoboMaker を使用したり、自分や他のユーザーの認証とアクセスコントロールを管理したりするには、適切なアクセス許可を持っている必要があります。

AWS RoboMaker コンソールを使用するために必要なアクセス権限

AWS RoboMaker コンソールにアクセスするには、AWS アカウントの AWS RoboMaker リソースに関する詳細を表示して確認するための最小限のアクセス許可が必要です。最小限必要なアクセス許可よりも制限されたアイデンティティベースのアクセス許可ポリシーを作成すると、そのポリシーをアタッチしたエンティティに対してはコンソールが意図したとおりに機能しません。

AWS RoboMaker コンソールへの読み取り専用アクセスには、AWSRoboMakerReadOnlyAccess ポリシーを使用します。

IAM ユーザーがシミュレーションジョブを作成する場合は、そのユーザーに対して `iam:PassRole` アクセス許可を付与する必要があります。ロールの受け渡しの詳細については、「AWS サービスにロールを渡すアクセス許可をユーザーに許可する」を参照してください。

例えば、ユーザーに次のポリシーをアタッチできます。このポリシーは、シミュレーションジョブを作成するアクセス許可を提供します。

```
 {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::123456789012:role/S3AndCloudWatchAccess"
        }
    ]
}
```

AWS CLI または AWS API のみを呼び出すユーザーには、最小限のコンソール許可を付与する必要はありません。代わりに、実行する API オペレーションに対応するアクセス許可のみが必要です。

コンソール の AWS RoboMaker でワールドを表示するために必要な許可

次のポリシーをユーザーにアタッチすれば、AWS RoboMaker コンソールで AWS RoboMaker ワールドを表示するために必要なアクセス権限を付与できます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "robomaker: DescribeWorld"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

AWS RoboMaker シミュレーションツールを使用するために必要なアクセス権限

シミュレーションを作成するために使用される IAM ユーザーまたはロールには、シミュレーションツールへのアクセス許可が自動的に与えられます。他のユーザーまたはロールの場合は、`robomaker:CreateSimulationJob` 権限を付与する必要があります。

認証の管理に必要なアクセス許可

自分の認証情報 (パスワード、アクセスキー、多要素認証 (MFA) デバイスなど) を管理するには、管理者から必要なアクセス許可を取得する必要があります。これらのアクセス許可が含まれているポリシーを表示するには、「認証情報の自己管理をユーザーに許可する」を参照してください。

AWS 管理者は、 IAMでユーザー、グループ、ロール、ポリシーを作成して管理するために IAM へのフルアクセスが必要です。AWS のすべてに対するフルアクセスを含む AdministratorAccess AWS マネージドポリシーを使用すべきです。このポリシーは、AWS Billing and Cost Management コンソールへのアクセスを提供しません。また、ルートユーザー認証情報を必要とするタスクを許可しません。詳細については、AWS 全般のリファレンス の「AWS アカウント ルートユーザー認証情報が必要な AWS タスク」を参照してください。

> **Warning**
>
> AWS へのフルアクセスを持つのは、管理者ユーザーに限ります。このポリシーをアタッチされたユーザーは、だれでも、認証とアクセスコントロールを完全に管理するアクセス許可と、AWS のすべてのリソースを変更するアクセス許可を付与されます。このユーザーを作成する方法については、「IAM 管理者ユーザーを作成する」を参照してください。

アクセスコントロールに必要なアクセス許可

管理者から IAM ユーザー認証情報が提供されている場合は、アクセスできるリソースをコントロールするためのポリシーが IAM ユーザーにアタッチされています。ユーザーにアタッチされているポリシーを AWS Management Console で表示するには、以下のアクセス許可が必要です。

```
{
 "Version": "2012-10-17",
 "Statement": [
     {
         "Sid": "ViewOwnUserInfo",
         "Effect": "Allow",
         "Action": [
             "iam:GetUserPolicy",
             "iam:ListGroupsForUser",
             "iam:ListAttachedUserPolicies",
             "iam:ListUserPolicies",
             "iam:GetUser"
         ],
```

```
            "Resource": [
                "arn:aws:iam::*:user/${aws:username}"
            ]
        },
        {
            "Sid": "ListUsersViewGroupsAndPolicies",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

追加のアクセス許可が必要な場合は、管理者に依頼し、ポリシーを更新して必要なアクションへのアクセスを許可してもらいます。

シミュレーションジョブに必要なアクセス許可

シミュレーションジョブは、作成時に、以下のアクセス許可のある IAM ロールを引き受ける必要があります。

- ロボットおよびシミュレーションアプリケーションバンドルが含まれるバケットの名前で `my-input-bucket` を置き換えます。
- AWS RoboMaker が出力ファイルを書き込むバケットを参照するように `my-output-bucket` を置き換えます。
- `account#` をアカウント番号に置き換えます。

パブリック ECR ジョブには、`ecr-public:GetAuthorizationToken`、`sts:GetServiceBearerToken`、その他最終的な実装に必要な権限などの個別の権限が必要です。詳細については、「Amazon ECR ユーザーガイド」の「Public リポジトリポリシー」を参照してください。

Jobs with Private ECR images

```
{
  "Version": "2012-10-17",
  "Statement": [
      {
          "Action": "s3:ListBucket",
          "Resource": [
              "arn:aws:s3:::my-input-bucket"
          ],
          "Effect": "Allow"
      },
      {
          "Action": [
              "s3:Get*",
              "s3:List*"
          ],
          "Resource": [
              "arn:aws:s3:::my-input-bucket/*"
          ],
          "Effect": "Allow"
      },
      {
          "Action": "s3:Put*",
          "Resource": [
              "arn:aws:s3:::my-output-bucket/*"
          ],
          "Effect": "Allow"
      },
      {
          "Action": [
              "logs:CreateLogGroup",
              "logs:CreateLogStream",
              "logs:PutLogEvents",
              "logs:DescribeLogStreams"
          ],
          "Resource": [
              "arn:aws:logs:*:account#:log-group:/aws/robomaker/SimulationJobs*"
          ],
          "Effect": "Allow"
      },
      {
          "Action": [
```

```
                "ecr:BatchGetImage",
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer"
            ],
            "Resource":
 "arn:partition:ecr:region:account#:repository/repository_name",
            "Effect": "Allow"
        }
    ]
}
```

Jobs with Public ECR images

```
 {
   "Version": "2012-10-17",
   "Statement": [
       {
           "Action": "s3:ListBucket",
           "Resource": [
               "arn:aws:s3:::my-input-bucket"
           ],
           "Effect": "Allow"
       },
       {
           "Action": [
               "s3:Get*",
               "s3:List*"
           ],
           "Resource": [
               "arn:aws:s3:::my-input-bucket/*"
           ],
           "Effect": "Allow"
       },
       {
           "Action": "s3:Put*",
           "Resource": [
               "arn:aws:s3:::my-output-bucket/*"
           ],
           "Effect": "Allow"
       },
       {
```

```
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams"
            ],
            "Resource": [
                "arn:aws:logs:*:account#:log-group:/aws/robomaker/SimulationJobs*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "ecr-public:GetAuthorizationToken",
                "sts:GetServiceBearerToken"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

このポリシーは、以下の信頼ポリシーを使用してロールにアタッチする必要があります。

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Principal": { "Service": "robomaker.amazonaws.com" },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": "account#" // Account where the simulation job
 resource is created
            },
            "StringEquals": {
                "aws:SourceArn": "arn:aws:robomaker:region:account#:simulation-job/*"
            }
        }
    }
}
```

条件キーは、サービス間のトランザクション中に AWS サービスが混乱した代理として使用されるのを防ぐことができます。条件キーに関する詳しい情報については、「SourceAccount」と「SourceArn」を参照してください。

ROS アプリケーションまたは ROS コマンドラインからタグを使用するために必要なアクセス許可

シミュレーションジョブのタグ付け、タグ解除、一覧表示は、ROS コマンドラインまたは ROS アプリケーションの実行中に行うことができます。以下のアクセス許可を持つ IAM ロールが必要です。account# をアカウント番号に置き換えます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "robomaker:TagResource",
                "robomaker:UntagResource",
                "robomaker:ListTagsForResource",
            ],
            "Resource": [
                "arn:aws:robomaker:*:account#:simulation-job*"
            ],
            "Effect": "Allow"
        }
    ]
}
```

このポリシーは、以下の信頼ポリシーを使用してロールにアタッチする必要があります。

```

{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Principal": { "Service": "robomaker.amazonaws.com" },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": "account#" // Account where the simulation job
 resource is created
            },
```

```
            "StringEquals": {
                "aws:SourceArn": "arn:aws:robomaker:region:account#:simulation-job/*"
            }
        }
    }
}
```

条件キーは、サービス間のトランザクション中に AWS サービスが混乱した代理として使用されるのを防ぐことができます。条件キーに関する詳しい情報については、「SourceAccount」と「SourceArn」を参照してください。

IAM で AWS RoboMaker を使用する方法について

サービスではいくつかの方法で IAM を使用できます。

- アクション：ポリシーでのアクションの使用をサポートしています。これにより、管理者は、AWS RoboMaker でオペレーションを実行することをエンティティに許可するかどうかをコントロールできます。例えば、ポリシーを表示するために `GetPolicy` AWS API オペレーションを実行することをエンティティに許可する場合、管理者は `iam:GetPolicy` アクションを許可するポリシーをアタッチする必要があります。
- リソースレベルのアクセス許可：リソースレベルのアクセス許可はサポートされていません。リソースレベルの許可では、ARN を使用してポリシー内で個々のリソースを指定できます。AWS RoboMaker ではこの機能がサポートされていないため、ポリシービジュアルエディタで [すべてのリソース] を選択する必要があります。JSON ポリシードキュメントでは、* 要素に `Resource` を使用する必要があります。
- タグベースの承認：AWS RoboMaker はタグベースの承認をサポートされていません。この機能により、ポリシーの条件でリソースタグを使用できます。
- 一時的認証情報：AWS RoboMaker は一時的な認証情報をサポートします。この機能により、フェデレーションを使用してサインインし、IAM ロールまたはクロスアカウントロールを引き受けることができます。一時的なセキュリティ認証情報を取得するには、AssumeRole または GetFederationToken などの AWS STS API オペレーションを呼び出します。
- サービスリンクロール：AWS RoboMaker はサービスロールをサポートします。この機能では、サービスリンクロールをユーザーに代わって引き受けることをサービスに許可します。このロールにより、サービスがユーザーに代わって他のサービスのリソースにアクセスし、アクションを完了することが許可されます。サービスリンクロールは、IAM アカウント内に表示され、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールの許可を表示できますが、編集することはできません。

- サービスロール：AWS RoboMaker はサービスロールをサポートします。この機能により、ユーザーに代わってサービスがサービスロールを引き受けることが許可されます。このロールにより、サービスがユーザーに代わって他のサービスのリソースにアクセスし、アクションを完了することが許可されます。サービスロールは、IAM アカウントに表示され、サービスによって所有されます。つまり、IAM 管理者が、このロールの許可を変更することができます。ただし、これにより、サービスの機能が損なわれる場合があります。

認証とアクセスコントロールのトラブルシューティング

次の情報は、IAM の使用に伴って発生する可能性がある一般的な問題の診断や修復に役立ちます。

トピック

- AWS RoboMaker でアクションを実行する権限がない
- 管理者として AWS RoboMaker へのアクセスを他のユーザーに許可したい

AWS RoboMaker でアクションを実行する権限がない

アクションを実行する権限がないというエラーが AWS Management Console に表示された場合は、ユーザー名とパスワードの提供元の管理者に問い合わせる必要があります。

以下の例のエラーは、 ユーザー (my-user-name) がコンソールを使用して CreateRobotApplication アクションを実行するときに、そのためのアクセス許可がない場合に発生します。

```
User: arn:aws:iam::123456789012:user/my-user-name is not authorized to perform: aws-robomaker:CreateRobotApplication on resource: my-example-robot-application
```

この例の場合は、aws-robomaker:CreateRobotApplication アクションを使用して my-example-robot-application リソースへのアクセスを許可するように、管理者にポリシーを更新してもらいます。

管理者として AWS RoboMaker へのアクセスを他のユーザーに許可したい

AWS RoboMaker へのアクセスを他のユーザーに許可するには、アクセスを必要とする人またはアプリケーションの IAM エンティティ (ユーザーまたはロール) を作成する必要があります。ユーザーは、このエンティティの認証情報を使用して AWS にアクセスします。次に、AWS RoboMaker の適切なアクセス許可を付与するポリシーを、そのエンティティにアタッチする必要があります。

すぐに開始するには、「IAM の使用開始」を参照してください。

ポリシーとは

AWS でのアクセスは、ポリシーを作成し、それらを IAM アイデンティティまたは AWS リソースに添付することで制御できます。

> **Note**
>
> 迅速に使用を開始するには、AWS RoboMaker の認証とアクセスコントロール の入門情報を確認し、次に「IAM の使用開始」を参照してください。

ポリシー は AWS のオブジェクトであり、エンティティやリソースに関連付けて、これらのアクセス許可を定義します。AWS は、ユーザーなどのプリンシパルがリクエストを行ったときに、それらのポリシーを評価します。ポリシーでの許可により、リクエストが許可されるか拒否されるかが決まります。大半のポリシーは JSON ドキュメントとして AWS に保存されます。

IAM ポリシーは、オペレーションの実行方法を問わず、アクションの許可を定義します。例えば、GetUser アクションを許可するポリシーを適用されたユーザーは、AWS Management Console、AWS CLI、または AWS API からユーザー情報を取得できます。IAM ユーザーを作成したら、コンソールまたはプログラムによるアクセスを許可するようにユーザーを設定できます。IAM は、ユーザー名とパスワードを使用してコンソールにサインインできます。または、アクセスキーを使用して CLI または API を操作できます。

アクセスを提供するには、ユーザー、グループ、またはロールにアクセス許可を追加します。

- AWS IAM Identity Center のユーザーとグループ:

  アクセス許可セットを作成します。「AWS IAM Identity Center ユーザーガイド」の「シークレットの作成と管理」の手順に従ってください。
- ID プロバイダーを通じて IAM で管理されているユーザー:

  ID フェデレーションのロールを作成する。詳細については、「IAM ユーザーガイド」の「サードパーティー ID プロバイダー (フェデレーション) 用のロールの作成」を参照してください。
- IAM ユーザー:
  - ユーザーが設定できるロールを作成します。手順については、「IAM ユーザーガイド」の「IAM ユーザー用ロールの作成」を参照してください。

- (非推奨) ポリシーをユーザーに直接アタッチするか、ユーザーをユーザーグループに追加します。「IAM ユーザーガイド」の「ユーザー (コンソール) へのアクセス許可の追加」の指示に従います。

> **AWS RoboMaker でサポートされていないポリシー**
>
> AWS RoboMaker では、リソースベースのポリシーやアクセスコントロールリスト (ACL) はサポートされていません。詳細については、「IAM ユーザーガイド」の「ポリシータイプ」を参照してください。

トピック

アイデンティティベースのポリシー

ポリシーを IAM アイデンティティにアタッチできます。例えば、次の操作を実行できます。

- アカウントのユーザーまたはグループにアクセス権限ポリシーをアタッチする：ロボットアプリケーションなどの AWS RoboMaker リソースを作成するためのアクセス許可をユーザーに付与するために、ユーザーまたはユーザーが所属するグループにアクセス許可のポリシーをアタッチできます。
- 許可ポリシーをロールにアタッチする (クロスアカウントの許可を付与) - ID ベースのアクセス許可ポリシーを IAM ロールにアタッチして、クロスアカウントの許可を付与することができます。例えば、アカウント A の管理者は、次のように別の AWS アカウント (例えば、アカウント B) または AWS サービスにクロスアカウントアクセス許可を付与するロールを作成できます。
  1. アカウント A の管理者は、IAM ロールを作成して、アカウント A のリソースに許可を付与するロールに許可ポリシーをアタッチします。
  2. アカウント A の管理者は、アカウント B をそのロールを引き受けるプリンシパルとして識別するロールに、信頼ポリシーをアタッチします。
  3. アカウント B の管理者は、アカウント B のユーザーにロールを引き受ける権限を委任できるようになります。これにより、アカウント B のユーザーはアカウント A のリソースの作成とアクセスができます。ロールを引き受ける権限を AWS のサービスに付与すると、信頼ポリシー内のプリンシパルも AWS のサービスのプリンシパルとなることができます。

IAM を使用した許可の委任の詳細については、「IAM ユーザーガイド」の「アクセス管理」を参照してください。

ユーザー、グループ、ロール、許可の詳細については、「IAM ユーザーガイド」の「アイデンティティ (ユーザー、グループ、ロール)」を参照してください。

ポリシーのアクセスレベルの分類

IAM コンソールでは、アクションが以下のアクセスレベルの分類に従ってグループ分けされます。

- リスト – サービス内のリソースを一覧表示し、オブジェクトの存否を判断するアクセス許可を提供します。このレベルのアクセス権を持つアクションはオブジェクトをリストできますが、リソースのコンテンツは表示されません。リストアクセスレベルの大半のアクションは、特定のリソースに対しては実行できません。これらのアクションを使用してポリシーステートメントを作成する場合は、[All resources] (すべてのリソース) ("*") を指定する必要があります。
- 読み取り– サービス内のリソースの内容と属性を読み取るアクセス許可を提供します。ただし、編集することはできません。例えば、Amazon S3 アクション `GetObject` および `GetBucketLocation` には、読み取りアクセスレベルがあります。
- 書き込み – サービス内のリソースを作成、削除、または変更するアクセス許可を提供します。例えば、Amazon S3 アクション `CreateBucket`、`DeleteBucket` および `PutObject` には、書き込みアクセスレベルがあります。
- アクセス許可管理 – サービス内のリソースに対するアクセス許可を付与または変更するアクセス許可を提供します。例えば、大半の IAM と AWS Organizations のポリシーアクションには、アクセス許可管理アクセスレベルがあります。

  > **ヒント**
  >
  > AWS アカウントのセキュリティを強化するには、Permissions management アクセスレベル分類を含むポリシーを制限したり定期的にモニタリングしたりします。

- タグ付け – サービス内のリソースにアタッチされているタグを作成、削除、または変更するアクセス許可を提供します。例えば、 Amazon EC2 の `CreateTags` アクションおよび `DeleteTags` アクションには、タグ付けアクセスレベルがあります。

AWS RoboMaker の AWS マネージドポリシー

ユーザー、グループ、ロールに許可を追加するには、自分でポリシーを作成するよりも、AWS マネージドポリシーを使用する方が簡単です。チームに必要な許可のみを提供する IAM カスタマーマネージドポリシーを作成するには、時間と専門知識が必要です。すぐに使用を開始するために、AWS マネージドポリシーを使用できます。これらのポリシーは、一般的なユースケースをターゲット範囲に含めており、AWS アカウント で利用できます。AWS マネージドポリシーの詳細については、「IAM ユーザーガイド」の「AWS マネージドポリシー」を参照してください。

AWS のサービスは、AWS マネージドポリシーを維持および更新します。AWS マネージドポリシーの許可を変更することはできません。サービスでは、新しい機能を利用できるようにするために、AWS マネージドポリシーに許可が追加されることがあります。この種類の更新は、ポリシーがアタッチされている、すべてのアイデンティティ (ユーザー、グループおよびロール) に影響を与えます。新しい機能が立ち上げられた場合や、新しいオペレーションが使用可能になった場合に、各サービスが AWS マネージドポリシーを更新する可能性が最も高くなります。サービスは、AWS マネージドポリシーから許可を削除しないため、ポリシーの更新によって既存の許可が破棄されることはありません。

さらに、AWS は、複数のサービスにまたがるジョブ機能の特徴に対するマネージドポリシーもサポートしています。例えば、ReadOnlyAccess AWS マネージドポリシーでは、すべての AWS のサービスおよびリソースへの読み取り専用アクセスを許可します。サービスが新しい機能を起動する場合、AWS は、新たなオペレーションとリソース用に、読み取り専用の許可を追加します。ジョブ関数ポリシーのリストと説明については、IAM ユーザーガイドの「ジョブ関数の AWS マネージドポリシー」を参照してください。

AWS マネージドポリシー: AWSRoboMaker_FullAccess

このポリシーは、アプリケーションの作成に使用できるイメージまたはバンドルを AWS RoboMaker で読み取ることができる寄稿者許可を付与するものです。さらに、このポリシーにより、すべての AWS RoboMaker リソースとオペレーションにアクセスできるようになります。また、アカウント内の Amazon EC2 リソースを管理する IAM ロールがアカウント内で作成されます。

許可の詳細

このポリシーには、以下の許可が含まれています。

- `s3:GetObject` : ロボットアプリケーションまたはシミュレーションアプリケーションのいずれかにバンドルを使用している場合、AWS RoboMaker で Amazon S3 バケットから zip ファイルを取得できます。

- `ecr:BatchGetImage`：ロボットアプリケーションまたはシミュレーションアプリケーションのいずれかにイメージを使用している場合、AWS RoboMaker で Amazon ECR リポジトリからそのイメージを取得できます。
- `ecr-public:DescribeImages`：ロボットアプリケーションまたはシミュレーションアプリケーションのいずれかに公開イメージを使用している場合、AWS RoboMaker で Amazon ECR リポジトリからそのイメージを取得できます。
- `iam:CreateServiceLinkedRole`：AWS RoboMaker を正常に動作させるために必要な Amazon EC2 リソースへのアクセスが提供されます。詳細については、「AWS RoboMaker のサービスにリンクされたロールの使用」を参照してください。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "robomaker:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "robomaker.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ecr:BatchGetImage",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "robomaker.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
```

```
            "Action": "ecr-public:DescribeImages",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "robomaker.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "robomaker.amazonaws.com"
                }
            }
        }
    ]
}
```

AWS が管理するポリシー：AWSRoboMakerReadOnlyAccess

このマネージドポリシーは、AWS Management Console と SDK を通じて AWS RoboMaker への読み取り専用アクセス権を提供します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "robomaker:List*",
                "robomaker:BatchDescribe*",
                "robomaker:Describe*",
                "robomaker:Get*"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

AWS マネージドポリシーに対する AWS RoboMaker 更新

AWS RoboMaker の AWS マネージドポリシーの更新で、このサービスによりこれらの変更の追跡が開始された後の分の詳細を表示します。このページの変更に関する自動通知については、「AWS RoboMaker ドキュメントの履歴」ページの RSS フィードを購読してください。

変更	説明	日付
AWSRoboMaker_FullAccess – 新しいポリシー	AWS RoboMaker では、正常な動作に必要なリソースへのアクセスを許可するための新しいポリシーが追加されました。 このポリシーにより、AWS RoboMaker では、Amazon S3 に保存した Amazon ECR イメージまたは zip ファイルにアクセスして、ロボットアプリケーションとシミュレーションアプリケーションを作成できるようになりました。また、正常な動作に必要な Amazon EC2 にアクセスする機能も AWS RoboMaker に備わりました。	2021 年 7 月 27 日
「AWSRoboMakerReadOnlyAccess」：新たなポリシー	AWS RoboMaker に、AWS RoboMaker リソースへの読み取り専用アクセスを許可する新しいポリシーが追加されました。	2022 年 1 月 11 日

変更	説明	日付
AWS RoboMaker で変更の追跡が開始されました	AWS RoboMaker で AWS マネージドポリシーの変更の追跡が開始されました。	2021 年 7 月 27 日

AWS RoboMaker のサービスにリンクされたロールの使用

AWS RoboMaker は AWS Identity and Access Management (IAM) サービスにリンクされたロールを使用します。サービスにリンクされたロールは、AWS RoboMaker に直接リンクされた一意のタイプの IAM ロールです。サービスにリンクされたロールは、AWS RoboMaker による事前定義済みのロールであり、ユーザーに代わってサービスから他の AWS のサービスを呼び出すために必要なすべての許可を備えています。

サービスにリンクされたロールを使用すると、必要なアクセス許可を手動で追加する必要がなくなるため、AWS RoboMaker の設定が簡単になります。AWS RoboMaker はサービスにリンクされたロールのアクセス許可を定義し、別途定義されている場合を除き、AWS RoboMaker のみがそのロールを引き受けることができます。定義される許可には、信頼ポリシーと許可ポリシーが含まれており、その許可ポリシーを他のIAM エンティティにアタッチすることはできません。

サービスリンクロールを削除するには、まずその関連リソースを削除します。これにより、リソースへの意図しないアクセスによるアクセス許可の削除が防止され、AWS RoboMaker リソースは保護されます。

サービスリンクロールをサポートする他のサービスについては、「IAM と連携する AWS のサービス」でサービスリンクロール列がはいになっているサービスを検索してください。そのサービスに関するサービスにリンクされたロールのドキュメントを表示するには、リンクが設定されている [Yes] (はい) を選択します。

AWS RoboMaker のサービスにリンクされたロールの許可

AWS RoboMaker では AWSServiceRoleForRoboMaker という名前のサービスリンクロールを使用します。このロールにより、RoboMaker が EC2 および Lambda リソースにアクセスできるようになります。

AWSServiceRoleForRoboMaker サービスリンクロールは、以下のサービスを信頼してロールを引き受けます。

- `robomaker.amazonaws.com`

ロールのアクセス許可ポリシーは、指定したリソースに対して以下のアクションを実行することを AWS RoboMaker に許可します。

- シミュレーションジョブバッチの一部として作成されたシミュレーションジョブを作成および取り消す
- Amazon EC2 ネットワークリソースの管理
- AWS Lambda 関数の作成と取得

サービスにリンクされたロールの作成、編集、削除をIAM エンティティ (ユーザー、グループ、ロールなど) に許可するには、許可を設定する必要があります。詳細については、「IAM ユーザーガイド」の「サービスにリンクされたロールの許可」を参照してください。

サービスにリンクされたロールの作成

サービスにリンクされたロールを手動で作成する必要はありません。SimulationJob または DeploymentJob を AWS Management Console、AWS CLI、AWS API のいずれかで実行すると、AWS RoboMaker でサービスにリンクされたロールが自動的に作成されます。

このサービスにリンクされたロールを削除した後で再度作成する必要が生じた場合は、同じ方法でアカウントにロールを再作成できます。SimulationJob、SimulationJobBatch、または DeploymentJob を作成すると、AWS RoboMaker でサービスにリンクされたロールが自動的に再作成されます。

IAM コンソールを使用して、RoboMaker ユースケースでサービスリンクロールを作成することもできます。AWS CLI または AWS API で、`robomaker.amazonaws.com` サービス名を使用してサービスリンクロールを作成します。詳細については、IAM ユーザーガイドの「サービスリンクロールの作成」を参照してください。このサービスリンクロールを削除する場合、この同じプロセスを使用して、もう一度ロールを作成できます。

サービスにリンクされたロールの編集

AWS RoboMaker では、AWSServiceRoleForRoboMaker のサービスにリンクされたロールを編集することはできません。サービスにリンクされたロールを作成すると、多くのエンティティによってロールが参照される可能性があるため、ロール名を変更することはできません。ただし、IAM を使用したロールの説明の編集はできます。詳細については、「IAM ユーザーガイド」の「サービスにリンクされたロールの編集」を参照してください。

サービスにリンクされたロールの削除

サービスにリンクされたロールが必要な機能またはサービスが不要になった場合には、そのロールを削除することをお勧めします。そうすることで、使用していないエンティティがアクティブにモニタリングされたり、メンテナンスされたりすることがなくなります。ただし、手動で削除する前に、サービスにリンクされたロールのリソースをクリーンアップする必要があります。

> **Note**
>
> リソースを削除する際に、AWS RoboMaker のサービスでロールが使用されている場合、削除は失敗することがあります。その場合は、数分待ってからオペレーションを再試行してください。

IAM を使用してサービスリンクロールを手動で削除するには

IAM コンソール、AWS CLI、または AWS API を使用して、サービスリンクロール AWSServiceRoleForRoboMaker を削除します。詳細については、「IAM ユーザーガイド」の「サービスにリンクされたロールの削除」を参照してください。

AWS RoboMaker のサービスにリンクされたロールをサポートするリージョン

AWS RoboMaker は、サービスを利用できるすべてのリージョンで、サービスにリンクされたロールの使用をサポートします。詳細については、「AWS リージョンとエンドポイント」を参照してください。

AWS RoboMaker は、サービスを利用できるすべてのリージョンで、サービスにリンクされたロールの使用をサポートしているわけではありません。AWSServiceRoleForRoboMaker ロールは、以下のリージョンで使用できます。

リージョン名	リージョン識別子	AWS RoboMaker でのサポート
米国東部 (バージニア北部)	us-east-1	はい
米国東部 (オハイオ)	us-east-2	はい
米国西部 (北カリフォルニア)	us-west-1	はい

リージョン名	リージョン識別子	AWS RoboMaker でのサポート
米国西部 (オレゴン)	us-west-2	はい
アジアパシフィック (ムンバイ)	ap-south-1	はい
アジアパシフィック (大阪)	ap-northeast-3	はい
アジアパシフィック (ソウル)	ap-northeast-2	はい
アジアパシフィック (シンガポール)	ap-southeast-1	はい
アジアパシフィック (シドニー)	ap-southeast-2	はい
アジアパシフィック (東京)	ap-northeast-1	はい
カナダ (中部)	ca-central-1	はい
欧州 (フランクフルト)	eu-central-1	はい
欧州 (アイルランド)	eu-west-1	はい
欧州 (ロンドン)	eu-west-2	はい
欧州 (パリ)	eu-west-3	はい
南米 (サンパウロ)	sa-east-1	はい
AWS GovCloud (US)	us-gov-west-1	いいえ

IAM の使用開始

AWS Identity and Access Management (IAM) はサービスおよびリソースへのアクセスを安全に管理できる AWS サービスです。IAM は追加料金なしで提供される AWS アカウントの機能です。

> **Note**
>
> IAM を開始する前に、AWS RoboMaker の認証とアクセスコントロール の基本情報に目を通してください。

AWS アカウント を作成する場合は、このアカウントのすべての AWS のサービス とリソースに対して完全なアクセス権を持つ 1 つのサインインアイデンティティから始めます。この ID は AWS アカウント ルートユーザーと呼ばれ、アカウントの作成に使用した E メールアドレスとパスワードでサインインすることによってアクセスできます。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザーの認証情報を保護し、それらを使用してルートユーザーのみが実行できるタスクを実行します。ルートユーザーとしてサインインする必要があるタスクの完全なリストについては、「IAM ユーザーガイド」の「ルートユーザー認証情報が必要なタスク」を参照してください。

IAM 管理者ユーザーを作成する

管理者ユーザーを作成するには、以下のいずれかのオプションを選択します。

管理者を管理する方法を 1 つ選択します	To	By	以下の操作も可能
IAM Identity Center 内 (推奨)	短期の認証情報を使用して AWS にアクセスします。 これはセキュリティのベストプラクティスと一致しています。ベストプラクティスの詳細については、IAM ユーザーガイドの「IAM でのセキュリティのベストプラクティス」を参照してください。	AWS IAM Identity Center ユーザーガイドの「開始方法」の手順に従います。	AWS Command Line Interface ユーザーガイドの「AWS IAM Identity Center を使用するための AWS CLI の設定」に従って、プログラムによるアクセスを設定します。
IAM 内 (非推奨)	長期認証情報を使用して AWS にアクセスする。	IAM ユーザーガイドの「最初の IAM 管理者のユー	IAM ユーザーガイドの「IAM ユーザーのアクセスキーの管

管理者を管理する方法を 1 つ選択します	To	By	以下の操作も可能
		ザーおよびグループの作成」の手順に従います。	理」に従って、プログラムによるアクセスを設定します。

AWS RoboMaker の委任ユーザーを作成する

AWS アカウントの複数のユーザーをサポートするには、許可するアクションのみ他のユーザーが実行できるようにアクセス許可を委任する必要があります。そのためには、そのようなユーザーが必要なアクセス許可を持つ IAM グループを作成し、IAM ユーザーの作成時に必要なグループに追加します。このプロセスを使用して、AWS アカウント全体のグループ、ユーザー、およびアクセス許可を設定できます。このソリューションは、AWS 管理者が手動でユーザーやグループを管理する中小企業で最もよく使用されます。大規模な組織では、カスタムの IAM ロール、フェデレーション、またはシングルサインオンを使用できます。

委任されたユーザーに関する例と詳しい情報は、「IAM ユーザーガイド」の「IAM ユーザーに権限を委任するロールの作成」を参照してください。

認証情報の自己管理をユーザーに許可する

MFA を設定するには、ユーザーの仮想 MFA デバイスをホストするハードウェアに物理的にアクセスできる必要があります。例えば、スマートフォンで実行される仮想 MFA デバイスを使用するユーザー用に MFA を設定するとします。その場合、ウィザードを完了するには、そのスマートフォンを利用できる必要があります。このため、ユーザーが自分の仮想 MFA デバイスを設定して管理できるようにすることをお勧めします。この場合、必要な IAM アクションを実行する権限をユーザーに付与する必要があります。

必要な権限の付与に関するポリシーの例については、「IAM ユーザーガイド」の「IAM: IAMユーザーが MFA デバイスを自己管理できるようにする」を参照してください。

IAM ユーザーの MFA を有効にする

セキュリティを強化するために、すべての IAM ユーザーが多要素認証 (MFA) を設定して AWS RoboMaker リソースを保護することをお勧めします。MFA では、さらなるセキュリティが追加され

ます。ユーザーは、通常のサインイン認証情報に加えて、AWS でサポートされている MFA デバイスから一意の認証情報を提供することを求められるためです。セットアップ手順とMFA オプションに関する詳しい情報は、「IAM ユーザーガイド」の「AWS におけるユーザーの MFA デバイスの有効化」を参照してください。

> **Note**
>
> IAM ユーザーの MFA を設定するには、ユーザーの仮想 MFA デバイスをホストするモバイルデバイスに物理的にアクセスできる必要があります。

AWS RoboMaker でのログ記録とモニタリング

モニタリングは、AWS RoboMaker と AWS ソリューションの信頼性、可用性、パフォーマンスを維持する上で重要な部分です。マルチポイント障害が発生した場合は、その障害をより簡単にデバッグできるように、AWS ソリューションのすべての部分からモニタリングデータを収集する必要があります。

トピック

- Amazon CloudWatch による AWS RoboMaker のモニターリング
- AWS CloudTrail を使用したコールのログ記録

Amazon CloudWatch による AWS RoboMaker のモニターリング

AWS RoboMaker は、メトリクスを Amazon CloudWatch に送信します。AWS Management Console、AWS CLI、または API を使用して、AWS RoboMaker により CloudWatch に送信されるメトリクスのリストを表示することができます。

メトリクスは作成されたリージョンにのみ存在します。メトリクスは削除できませんが、それらに対して新しいデータが発行されない場合、15 か月後に自動的に有効期限切れになります。

Amazon CloudWatch の詳細については、Amazon CloudWatch ユーザーガイドを参照してください。

トピック

- AWS RoboMaker シミュレーションメトリクス
- AWS RoboMaker 使用状況メトリクス

AWS RoboMaker シミュレーションメトリクス

Amazon CloudWatch を使用して AWS RoboMaker シミュレーションジョブをモニタリングすることで、シミュレーションジョブから情報を収集し、リアルタイムに近い読み取り可能なメトリクスに加工することができます。情報は 1 分間隔で提供されます。

`SimulationJobId` ディメンションで表示されるメトリクスは、以下のとおりです。

メトリクス	説明
`RealTimeFactor`	シミュレートされた時間と実時間の比率。例えば、30 分をシミュレートするために 1 時間を要した場合、係数は 0.5 です。 シミュレーションが複雑になるほど、実時間係数が低くなります。
`vCPU*`	シミュレーションジョブで使用される仮想 CPU コアの数 単位: 個
`Memory*`	シミュレーションジョブで使用される GB 単位のメモリの量 単位: GB
`SimulationUnit*`	`SimulationUnit` は、シミュレーションジョブの vCPU およびメモリの消費量に基づいて計算されます。 単位: 個

Important

* が付いているメトリクスは予測を目的としています。AWSシミュレーションジョブの実行準備中に RoboMaker によりメトリクスが発行されます。シミュレーションジョブが `Running` 状態になるまで料金は発生しません。

AWS RoboMaker 使用状況メトリクス

CloudWatch 使用状況メトリクスを使用して、アカウントのリソースの使用状況を把握できます。これらのメトリクスを使用して、CloudWatch グラフやダッシュボードで現在のサービスの使用状況を可視化できます。

AWS RoboMaker 使用状況メトリクスは、AWS Service Quotas に対応しています。使用量がサービスクォータに近づいたときに警告するアラームを設定することもできます。Service Quotas と CloudWatch の統合の詳細については、「Service Quotas の統合と使用状況のメトリクス」を参照してください。

AWS/Usage ディメンションで表示されるメトリクスは、以下のとおりです。

メトリクス	説明
ResourceCount	アカウントで実行されている指定されたリソースの数。リソースは、メトリクスに関連付けられたディメンションによって定義されます。 このメトリクスで最も役に立つ統計は MAXIMUM です。これは、1 分間の期間中に使用されるリソースの最大数を表します。

以下のディメンションは、AWS RoboMaker によって発行される使用状況メトリクスを絞り込むために使用されます。

ディメンション	説明
Service	リソースを含む AWS のサービスの名前。AWS RoboMaker 使用状況メトリクスの場合、このディメンションの値は RoboMaker　です。
Type	報告されるエンティティタイプ。現在、AWS RoboMaker 使用状況メトリクスの有効な値は Resource のみです。
Resource	実行中のリソースタイプ。現在、AWS RoboMaker 使用状況メトリクスの有効な

ディメンション	説明
	値は RobotApplication 、SimulationApplication 、ActiveSimulationJob 、および ActiveSimulationJobBatch です。
Class	追跡されているリソースのクラス。Resource ディメンションの値として ActiveSimulationJob を使用する AWS RoboMaker 使用状況メトリクスの場合、有効な値は CPU | GPU_AND_CPU です。このディメンションの値により、そのメトリクスによって報告されるシミュレーションジョブで使用されるコンピューティングリソースの種類が定義されます。その他の場合、クラス値は None です。

これらのメトリクスは、毎分発行されます。これらのメトリクスを使用して使用量をモニタリングしてから、必要に応じて対応する制限の引き上げをリクエストします。使用量のモニタリングの詳細については、「Service Quotas の視覚化とアラームの設定」を参照してください。

AWS CloudTrail を使用したコールのログ記録

AWS RoboMaker は、これはユーザー、ロール、または AWS RoboMaker 内にある AWS サービスによって実行されたアクションを記録するためのサービスと AWS CloudTrail を統合されています。CloudTrail は、AWS RoboMaker のすべての API コールをイベントとしてキャプチャします。キャプチャされたコールには、AWS RoboMaker コンソールからのコールと、AWS RoboMaker API オペレーションへのコードコールが含まれます。証跡を作成する場合は、AWS RoboMaker のイベントなど、Amazon S3 バケットへの CloudTrail イベントの継続的な配信を有効にすることができます。追跡を設定しない場合でも、CloudTrail コンソールの [Event history] (イベント履歴) で最新のイベントを表示できます。CloudTrail で収集された情報を使用して、AWS RoboMaker に対するリクエスト、リクエスト元の IP アドレス、リクエスト者、リクエスト日時などの詳細を確認できます。

CloudTrail の詳細については、AWS CloudTrail ユーザーガイドを参照してください。

CloudTrail における AWS RoboMaker 情報

AWS アカウントを作成すると、そのアカウントに対して CloudTrail が有効になります。AWS RoboMaker でアクティビティが発生すると、そのアクティビティはイベント履歴の他の AWS サービスイベントとともに CloudTrail イベントに記録されます。AWS アカウントで最近のイベントを表示、検索、ダウンロードできます。詳細については、「Viewing Events with CloudTrail Event History」(CloudTrail イベント履歴でのイベントの表示) を参照してください。

AWS RoboMaker のイベントなど、AWS アカウントのイベントを継続的に記録するには、証跡を作成します。追跡により、CloudTrail はログファイルを Amazon S3 バケットに配信できます。デフォルトでは、コンソールで追跡を作成するときに、追跡がすべての AWS リージョンに適用されます。追跡は、AWSパーティションのすべてのリージョンからのイベントをログに記録し、指定した Amazon S3 バケットにログファイルを配信します。さらに、CloudTrail ログで収集したイベントデータをより詳細に分析し、それに基づく対応するためにその他の AWS のサービスを設定できます。詳細については、次を参照してください。

- 追跡を作成するための概要
- CloudTrail のサポート対象サービスと統合
- Amazon SNS の CloudTrail の通知の設定
- 「複数のリージョンから CloudTrail ログファイルを受け取る」および「複数のアカウントから CloudTrail ログファイルを受け取る」

すべての AWS RoboMaker アクションは CloudTrail が記録します。これらのアクションは「AWS RoboMaker API リファレンス」で説明されています。例えば、`CreateSimulationJob`、`RegisterRobot`、`UpdateRobotApplication`の各アクションを呼び出すと、CloudTrail ログファイルにエントリが生成されます。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます。

- リクエストが、ルート認証情報と AWS Identity and Access Management (IAM) ユーザー認証情報のどちらを使用して送信されたか。
- リクエストがロールまたはフェデレーティッドユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが、別の AWS のサービスによって送信されたかどうか。

詳細については、「CloudTrail userIdentity エレメント」を参照してください。

AWS RoboMaker ログファイルエントリの概要

「トレイル」は、指定した Simple Storage Service (Amazon S3) バケットにイベントをログファイルとして配信するように設定できます。CloudTrail のログファイルには、単一か複数のログエントリがあります。イベントはあらゆるソースからの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストのパラメータなどの情報が含まれます。CloudTrail ログファイルは、パブリック API コールの順序付けられたスタックトレースではないため、特定の順序では表示されません。

次の例は、DescribeRobotアクションを示す CloudTrail ログエントリです。


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "my-principal-id",
        "arn": "my-arn",
        "accountId": "my-account-id",
        "accessKeyId": "my-access-key",
        "userName": "my-user-name"
    },
    "eventTime": "2018-12-07T00:28:03Z",
    "eventSource": "robomaker.amazonaws.com",
    "eventName": "DescribeRobot",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "my-ip-address",
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.455
 Linux/4.4.83-0.1.fm.327.54.326.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.192-b12
 java/1.8.0_192,",
    "requestParameters": {
        "robot": "my-robot-arn"
    },
    "responseElements": null,
    "requestID": "f54cdf8b-f9b6-11e8-8883-c3f04579eca3",
    "eventID": "affb0303-ff48-4f65-af8e-d7d19710bac3",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "recipientAccountId": "my-recipient-account-id"
}
```

AWS RoboMaker リソースのタグ付け

フリート、ロボット、ロボットアプリケーション、シミュレーションアプリケーション、およびシミュレーションジョブを管理および整理しやすいように、必要に応じて、タグの形式で特定のリソースに独自のメタデータを割り当てることができます。このセクションでは、タグとその作成方法について説明します。

ベーシックタグ

タグを使用すると、AWS RoboMaker リソースを目的、所有者、環境などさまざまな方法で分類することができます。これは、同じタイプのリソースが多数ある場合に役立ちます。割り当てたタグに基づいて特定のリソースをすばやく識別できます。タグはそれぞれ、1 つのキーとオプションの値で設定され、どちらもユーザーが定義します。例えば、機能ごとにデバイスを追跡するのに役立つ、ロボット用の一連のタグを定義できます。リソースの種類ごとのニーズを合わせて一連のタグキーを作成することをお勧めします。一貫性のあるタグキーセットを使用することで、リソースの管理が容易になります。

追加または適用したタグに基づいて、リソースを検索およびフィルター処理できます。また、「IAM ポリシーでのタグの使用」で説明しているように、タグを使用してリソースへのアクセスを制御することもできます。

使いやすさのために、のタグエディタは、タグを作成および管理するための AWS Management Console の中央の統一された方法を提供します。詳細については、 の操作 AWS Management Consoleのタグエディタの操作を参照してください。

また、AWS CLI および AWS RoboMaker API を使用してタグを操作することもできます。モノのグループ、モノのタイプ、トピックルール、ジョブ、セキュリティプロファイル、および請求グループの作成時に、以下のコマンドで [タグ] フィールドを使用してタグを関連付けることができます。

- CreateRobotApplication
- CreateSimulationApplication
- CreateSimulationJob
- CreateWorldExportJob
- CreateWorldGenerationJob
- CreateWorldTemplate
- StartSimulationJobBatch

以下のコマンドを使用して、タグ付けがサポートされている既存のリソースに対してタグを追加、変更、または削除できます。

- TagResource
- ListTagsForResource
- UntagResource

タグのキーと値は編集でき、タグはリソースからいつでも削除できます。タグの値を空の文字列に設定することはできますが、タグの値を null に設定することはできません。特定のリソースについて既存のタグと同じキーを持つタグを追加した場合、古い値は新しい値によってオーバーライドされます。リソースを削除すると、リソースに関連付けられているすべてのタグも削除されます。

タグの制約と制限

次のベーシックな制限がタグに適用されます。

- リソースあたりのタグの最大数: 50
- キーの最大長: 127 文字 (Unicode) (UTF-8)
- 値の最大長: 255 文字 (Unicode) (UTF-8)
- タグのキーと値は大文字と小文字が区別されます。
- タグの名前または値に `aws:` プレフィックスを使用しないでください。これは、AWS での使用のために予約されているためです。このプレフィックスが含まれるタグの名前または値は編集または削除できません。このプレフィックスを持つタグは、リソースあたりのタグ数の制限にはカウントされません。
- 複数のサービス間およびリソース間でタグ付けスキーマを使用する場合、他のサービスでも許可される文字に制限が適用されることがあるのでご注意ください。一般に、許可される文字は、UTF-8 で表現可能な文字、スペース、数字、および次の特殊文字です: + - = . _ : / @。

IAM ポリシーでのタグの使用

AWS RoboMaker API アクションに対して使用する IAM ポリシーで、タグベースのリソースレベルアクセス許可を適用することができます。これにより、ユーザーがどのリソースを作成、変更、または使用できるかを制御しやすくなります。IAM ポリシーの以下の条件コンテキストのキーと値とともに `Condition` 要素 (`Condition` ブロックとも呼ばれる) を使用して、リソースのタグに基づいてユーザーアクセス (アクセス許可) を制御できます。

- 特定のタグを持つリソースに対してユーザーアクションを許可または拒否するには、`aws:ResourceTag/tag-key: tag-value` を使用します。
- タグが許可されているリソースを作成または変更する API リクエストを作成する場合に、特定のタグが使用されている (または、使用されていない) ことを要求するには、`aws:RequestTag/tag-key: tag-value` を使用します。
- タグが許可されているリソースを作成または変更する API リクエストを作成する場合に、特定の一連のタグが使用されている (または、使用されていない) ことを要求するには、`aws:TagKeys: [tag-key, ...]` を使用します。

> **Note**
>
> IAM ポリシーの条件コンテキストのキーと値は、タグ付け可能なリソースの ID が必須パラメータである AWS RoboMaker アクションにのみ適用されます。例えば、このリクエストではタグ付け可能なリソース (フリート、ロボット、ロボットアプリケーション、シミュレーションアプリケーション、シミュレーションジョブ、デプロイジョブ) が参照されないため、ListFleets の使用は条件コンテキストキーおよび値に基づいて許可または拒否されることはありません。

詳細については、「AWS Identity and Access Management ユーザーガイド」の「AWS のリソースへのアクセスの制御」をご参照ください。そのガイドの [IAM JSON policy reference] (IAM JSON ポリシーリファレンス) セクションには、IAM での JSON ポリシーの要素、可変、および評価ロジックの詳細な構文、説明、および例が記載されています。

次のポリシー例では、タグベースの 2 つの制約が適用されています。このポリシーによって制限されている IAM ユーザーは、次のように制限されます。

- "env=prod" タグが付いている口ボットを作成することはできません (この例の "aws:RequestTag/env" : "prod" の行を参照)。
- 既存のタグ "env=prod" の付いた口ボットを削除することはできません (この例の "aws:ResourceTag/env" : "prod" の行を参照)。

```
{
    "Version" : "2012-10-17",
    "Statement" : [
        {
```

```
            "Effect" : "Deny",
            "Action" : "robomaker:CreateRobot",
            "Resource" : "*",
            "Condition" : {
              "StringEquals" : {
                "aws:RequestTag/env" : "prod"
              }
            }
          },
          {
            "Effect" : "Deny",
            "Action" : "robomaker:DeleteRobot",
            "Resource" : "*",
            "Condition" : {
              "StringEquals" : {
                "aws:ResourceTag/env" : "prod"
              }
            }
          },
          {
            "Effect": "Allow",
            "Action": "robomaker:*",
            "Resource": "*"
          }
    ]
}
```

次のように、タグ値を1 つのリストとして指定して、1 つのタグキーに対して複数のタグ値を指定することもできます。

```
            "StringEquals" : {
              "aws:ResourceTag/env" : ["dev", "test"]
}
```

> **Note**
>
> タグに基づいてリソースへのユーザーのアクセスを許可または拒否する場合は、ユーザーが同じリソースに対してそれらのタグを追加または削除することを明示的に拒否することを検討する必要があります。そうしないと、ユーザーはそのリソースのタグを変更することで、制限を回避してリソースにアクセスできてしまいます。

セキュリティコンプライアンス

AWS の HIPAA コンプライアンスプログラムには、HIPAA 適格サービスとして AWS RoboMaker が含まれています。AWS PCI DSS コンプライアンスプログラムには、PCI 準拠サービスとして AWS RoboMaker が含まれています。

AWS クラウドおよび HIPAA コンプライアンスの概要については、以下を参照してください。

- HIPAA への準拠
- Architecting for HIPAA Security and Compliance on Amazon Web Services

AWS RoboMaker の耐障害性

AWS のグローバルインフラストラクチャは AWS リージョンとアベイラビリティーゾーンを中心に構築されます。AWSリージョンには、低レイテンシー、高いスループット、そして高度の冗長ネットワークで Connect されている複数の物理的に独立· 隔離されたアベイラビリティーゾーンがあります。アベイラビリティーゾーンでは、ゾーン間で中断することなく自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティーゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性、耐障害性、および拡張性が優れています。

AWS リージョンとアベイラビリティーゾーンの詳細については、「AWS グローバルインフラストラクチャ」を参照してください。

AWS RoboMaker には、AWS グローバルインフラストラクチャに加えて、データの耐障害性とバックアップのニーズに対応できるように複数の機能が備わっています。

AWS RoboMaker のインフラストラクチャセキュリティ

マネージドサービスである AWS RoboMaker は AWS グローバルネットワークセキュリティで保護されています。AWSセキュリティサービスと AWS がインフラストラクチャを保護する方法については、「AWS クラウドセキュリティ」を参照してください。インフラストラクチャセキュリティのベストプラクティスを使用して AWS 環境を設計するには、「セキュリティの柱 - AWS Well-Architected Framework」の「インフラストラクチャ保護」を参照してください。

AWS が公開した API コールを使用して、ネットワーク経由で AWS RoboMaker にアクセスします。クライアントは以下をサポートする必要があります。

- Transport Layer Security (TLS) TLS 1.2 および TLS 1.3 をお勧めします。
- DHE (Ephemeral Diffie-Hellman) や ECDHE (Elliptic Curve Ephemeral Diffie-Hellman) などの Perfect Forward Secrecy (PFS) を使用した暗号スイートです。これらのモードは、Java 7 以降など、最近のほとんどのシステムでサポートされています。

また、リクエストは、アクセスキー ID と、IAM プリンシパルに関連付けられているシークレットアクセスキーを使用して署名する必要があります。または、AWS Security Token Service (AWS STS) を使用して、一時的なセキュリティ認証情報を生成し、リクエストに署名することもできます。

AWS RoboMaker とインターフェース VPC エンドポイント (AWS PrivateLink)

VPC と AWS RoboMaker とのプライベート接続を確立するには、インターフェース VPC エンドポイントを作成します。インターフェイスエンドポイントは、インターネットゲートウェイ、NAT デバイス、VPN 接続、AWS Direct Connect 接続のいずれも必要とせずに AWS RoboMaker API にプライベートにアクセスできるテクノロジーである AWS PrivateLink を利用しています。VPC のインスタンスは、パブリック IP アドレスがなくても AWS RoboMaker API と通信できます。VPC と AWS RoboMaker 間のトラフィックは、Amazon ネットワークを離れません。

各インターフェースエンドポイントは、サブネット内の 1 つ以上の Elastic Network Interface によって表されます。

詳細については、「AWS PrivateLink ガイド」の「インターフェイス VPC エンドポイント (AWS PrivateLink)」を参照してください。

AWS RoboMaker VPC エンドポイントに関する考慮事項

AWS RoboMaker 用の VPC エンドポイントを設定する前に、「AWS PrivateLink ガイド」の「インターフェイスエンドポイントのプロパティと制限」を確認してください。

AWS RoboMaker は、VPC からのすべての API アクションの呼び出しをサポートしています。

AWS RoboMaker 用のインターフェイス VPC エンドポイントの作成

AWS RoboMaker サービス用の VPC エンドポイントは、Amazon VPC コンソールまたは AWS Command Line Interface (AWS CLI) を使用して作成できます。詳細については、「AWS PrivateLink ガイド」の「インターフェイスエンドポイントの作成」を参照してください。

AWS RoboMaker 用の VPC エンドポイントは、以下のサービス名を使用して作成します。

- com.amazonaws.*region*.robomaker

エンドポイントのプライベート DNS を有効にすると、リージョンのデフォルト DNS 名（`robomaker.us-east-1.amazonaws.com` など）を使用して、AWS RoboMaker への API リクエストを実行できます。

詳細については、「AWS PrivateLink ガイド」の「インターフェイスエンドポイントを介したサービスへアクセスする」を参照してください。

AWS RoboMaker 用の VPC エンドポイントポリシーの作成

VPC エンドポイントには、AWS RoboMaker へのアクセスを制御するエンドポイントポリシーをアタッチできます。このポリシーでは、以下の情報を指定します。

- アクションを実行できるプリンシパル。
- 実行可能なアクション。
- このアクションを実行できるリソース。

詳細については、「AWS PrivateLink ガイド」の「VPC エンドポイントによるサービスのアクセスコントロール」を参照してください。

例：AWS RoboMaker アクション用の VPC エンドポイントポリシー

以下は、AWS RoboMaker 用のエンドポイントポリシーの例です。エンドポイントにアタッチされると、このポリシーは、すべてのリソースですべてのプリンシパルに、リストされている AWS RoboMaker アクションへのアクセス権を付与します。

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "robomaker:ListSimulationJobs",
            "robomaker:ListSimulationJobBatches"
         ],
         "Resource":"*"
      }
```

```
    ]
}
```

API リファレンス

この章には、の API リファレンスドキュメントが含まれています。 AWS RoboMakerこれには、次のセクションがあります。

セクション

- アクション
- データ型
- 共通エラー
- 共通パラメータ

アクション

以下のアクションがサポートされています:

- BatchDeleteWorlds
- BatchDescribeSimulationJob
- CancelDeploymentJob
- CancelSimulationJob
- CancelSimulationJobBatch
- CancelWorldExportJob
- CancelWorldGenerationJob
- CreateDeploymentJob
- CreateFleet
- CreateRobot
- CreateRobotApplication
- CreateRobotApplicationVersion
- CreateSimulationApplication
- CreateSimulationApplicationVersion
- CreateSimulationJob
- CreateWorldExportJob
- CreateWorldGenerationJob

- CreateWorldTemplate
- DeleteFleet
- DeleteRobot
- DeleteRobotApplication
- DeleteSimulationApplication
- DeleteWorldTemplate
- DeregisterRobot
- DescribeDeploymentJob
- DescribeFleet
- DescribeRobot
- DescribeRobotApplication
- DescribeSimulationApplication
- DescribeSimulationJob
- DescribeSimulationJobBatch
- DescribeWorld
- DescribeWorldExportJob
- DescribeWorldGenerationJob
- DescribeWorldTemplate
- GetWorldTemplateBody
- ListDeploymentJobs
- ListFleets
- ListRobotApplications
- ListRobots
- ListSimulationApplications
- ListSimulationJobBatches
- ListSimulationJobs
- ListTagsForResource
- ListWorldExportJobs
- ListWorldGenerationJobs
- ListWorlds

- ListWorldTemplates
- RegisterRobot
- RestartSimulationJob
- StartSimulationJobBatch
- SyncDeploymentJob
- TagResource
- UntagResource
- UpdateRobotApplication
- UpdateSimulationApplication
- UpdateWorldTemplate

BatchDeleteWorlds

バッチオペレーションで 1 つまたは複数のワールドを削除します。

リクエストの構文

```
POST /batchDeleteWorlds HTTP/1.1
Content-type: application/json

{
   "worlds": [ "string" ]
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

worlds

削除するワールドに対応する Amazon リソースネーム (ARN) の一覧。

タイプ：文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 100 項目です。

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
```

```
   "unprocessedWorlds": [ "string" ]
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

unprocessedWorlds

コールに関連付けられている未処理のワールドのリスト。これらのワールドは削除されませんでした。

タイプ：文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 100 項目です。

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン：`arn:.*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

BatchDescribeSimulationJob

1 つまたは複数のシミュレーションジョブを記述します。

リクエストの構文

```
POST /batchDescribeSimulationJob HTTP/1.1
Content-type: application/json

{
   "jobs": [ "string" ]
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

jobs

記述するシミュレーションジョブの Amazon リソースネーム (ARN) のリスト。

タイプ：文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 100 項目です。

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "jobs": [
      {
```

```
        "arn": "string",
        "clientRequestToken": "string",
        "compute": {
           "computeType": "string",
           "gpuUnitLimit": number,
           "simulationUnitLimit": number
        },
        "dataSources": [
           {
              "destination": "string",
              "name": "string",
              "s3Bucket": "string",
              "s3Keys": [
                 {
                    "etag": "string",
                    "s3Key": "string"
                 }
              ],
              "type": "string"
           }
        ],
        "failureBehavior": "string",
        "failureCode": "string",
        "failureReason": "string",
        "iamRole": "string",
        "lastStartedAt": number,
        "lastUpdatedAt": number,
        "loggingConfig": {
           "recordAllRosTopics": boolean
        },
        "maxJobDurationInSeconds": number,
        "name": "string",
        "networkInterface": {
           "networkInterfaceId": "string",
           "privateIpAddress": "string",
           "publicIpAddress": "string"
        },
        "outputLocation": {
           "s3Bucket": "string",
           "s3Prefix": "string"
        },
        "robotApplications": [
           {
              "application": "string",
```

```
            "applicationVersion": "string",
            "launchConfig": {
               "command": [ "string" ],
               "environmentVariables": {
                  "string" : "string"
               },
               "launchFile": "string",
               "packageName": "string",
               "portForwardingConfig": {
                  "portMappings": [
                     {
                        "applicationPort": number,
                        "enableOnPublicIp": boolean,
                        "jobPort": number
                     }
                  ]
               },
               "streamUI": boolean
            },
            "tools": [
               {
                  "command": "string",
                  "exitBehavior": "string",
                  "name": "string",
                  "streamOutputToCloudWatch": boolean,
                  "streamUI": boolean
               }
            ],
            "uploadConfigurations": [
               {
                  "name": "string",
                  "path": "string",
                  "uploadBehavior": "string"
               }
            ],
            "useDefaultTools": boolean,
            "useDefaultUploadConfigurations": boolean
         }
      ],
      "simulationApplications": [
         {
            "application": "string",
            "applicationVersion": "string",
            "launchConfig": {
```

```
            "command": [ "string" ],
            "environmentVariables": {
               "string" : "string"
            },
            "launchFile": "string",
            "packageName": "string",
            "portForwardingConfig": {
               "portMappings": [
                  {
                     "applicationPort": number,
                     "enableOnPublicIp": boolean,
                     "jobPort": number
                  }
               ]
            },
            "streamUI": boolean
         },
         "tools": [
            {
               "command": "string",
               "exitBehavior": "string",
               "name": "string",
               "streamOutputToCloudWatch": boolean,
               "streamUI": boolean
            }
         ],
         "uploadConfigurations": [
            {
               "name": "string",
               "path": "string",
               "uploadBehavior": "string"
            }
         ],
         "useDefaultTools": boolean,
         "useDefaultUploadConfigurations": boolean,
         "worldConfigs": [
            {
               "world": "string"
            }
         ]
      }
   ],
   "simulationTimeMillis": number,
   "status": "string",
```

```
            "tags": {
               "string" : "string"
            },
            "vpcConfig": {
               "assignPublicIp": boolean,
               "securityGroups": [ "string" ],
               "subnets": [ "string" ],
               "vpcId": "string"
            }
         }
      ],
      "unprocessedJobs": [ "string" ]
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

jobs

シミュレーションジョブのリスト。

型: SimulationJob オブジェクトの配列

unprocessedJobs

未処理のシミュレーションジョブの Amazon リソースネーム (ARN) のリスト。

タイプ：文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 100 項目です。

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン：`arn:.*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

CancelDeploymentJob

このアクションは非推奨になりました。

> **Important**
>
> この API はサポートされなくなりました。詳細については、「サポートポリシー」ページの 2022 年 5 月 2 日の更新をご覧ください。

指定されたデプロイジョブをキャンセルします。

リクエストの構文

```
POST /cancelDeploymentJob HTTP/1.1
Content-type: application/json

{
   "job": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

job

キャンセルするデプロイジョブ ARN。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
```

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス

- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

CancelSimulationJob

指定されたシミュレーションジョブをキャンセルします。

リクエストの構文

```
POST /cancelSimulationJob HTTP/1.1
Content-type: application/json

{
   "job": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

job

キャンセルするシミュレーションジョブ ARN。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
```

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK

- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

CancelSimulationJobBatch

シミュレーションジョブバッチをキャンセルします。シミュレーションジョブバッチをキャンセルすると、そのバッチの一部として作成されたアクティブなシミュレーションジョブもすべてキャンセルされます。

リクエストの構文

```
POST /cancelSimulationJobBatch HTTP/1.1
Content-type: application/json

{
   "batch": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

batch

キャンセルするバッチの ID。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
```

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK

- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

CancelWorldExportJob

指定されたエクスポートジョブをキャンセルします。

リクエストの構文

```
POST /cancelWorldExportJob HTTP/1.1
Content-type: application/json

{
   "job": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

job

キャンセルするワールドエクスポートジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
```

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK

- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

CancelWorldGenerationJob

指定されたワールドジェネレータージョブをキャンセルします。

リクエストの構文

```
POST /cancelWorldGenerationJob HTTP/1.1
Content-type: application/json

{
   "job": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

job

キャンセルするワールドジェネレータージョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
```

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK

- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

CreateDeploymentJob

このアクションは非推奨になりました。

> **Important**
>
> この API はサポートされなくなったため、使用した場合はエラーがスローされます。詳細については、「サポートポリシー」ページの 2022 年 1 月 31 日の更新情報を参照してください。

特定のバージョンのロボットアプリケーションをフリート内のロボットにデプロイします。

ロボットアプリケーションには、整合性のために番号付きの applicationVersion が必要です。新しいバージョンを作成するには、CreateRobotApplicationVersion を使用するか、または「ロボットアプリケーションバージョンの作成」を参照してください。

> **Note**
>
> 90 日後、デプロイメントジョブは期限切れになり、削除されます。これらのジョブにはアクセスできなくなります。

リクエストの構文

```
POST /createDeploymentJob HTTP/1.1
Content-type: application/json

{
   "clientRequestToken": "string",
   "deploymentApplicationConfigs": [
      {
         "application": "string",
         "applicationVersion": "string",
         "launchConfig": {
            "environmentVariables": {
               "string" : "string"
            },
            "launchFile": "string",
            "packageName": "string",
            "postLaunchFile": "string",
```

```
            "preLaunchFile": "string"
         }
      }
   ],
   "deploymentConfig": {
      "concurrentDeploymentPercentage": number,
      "downloadConditionFile": {
         "bucket": "string",
         "etag": "string",
         "key": "string"
      },
      "failureThresholdPercentage": number,
      "robotDeploymentTimeoutInSeconds": number
   },
   "fleet": "string",
   "tags": {
      "string" : "string"
   }
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

clientRequestToken

リクエストのべき等のために割り当てる一意の識別子 (大文字と小文字を区別)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 64 文字です。

パターン: `[a-zA-Z0-9_\-=]*`

必須：はい

deploymentApplicationConfigs

デプロイアプリケーションの設定。

型: DeploymentApplicationConfig オブジェクトの配列

配列メンバー: 定数は 1 項目です。

必須: はい

deploymentConfig

リクエストされたデプロイ設定。

タイプ：DeploymentConfig オブジェクト

必須: いいえ

fleet

デプロイするフリートの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

tags

デプロイジョブにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
```

```
Content-type: application/json

{
   "arn": "string",
   "createdAt": number,
   "deploymentApplicationConfigs": [
      {
         "application": "string",
         "applicationVersion": "string",
         "launchConfig": {
            "environmentVariables": {
               "string" : "string"
            },
            "launchFile": "string",
            "packageName": "string",
            "postLaunchFile": "string",
            "preLaunchFile": "string"
         }
      }
   ],
   "deploymentConfig": {
      "concurrentDeploymentPercentage": number,
      "downloadConditionFile": {
         "bucket": "string",
         "etag": "string",
         "key": "string"
      },
      "failureThresholdPercentage": number,
      "robotDeploymentTimeoutInSeconds": number
   },
   "failureCode": "string",
   "failureReason": "string",
   "fleet": "string",
   "status": "string",
   "tags": {
      "string" : "string"
   }
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

デプロイジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

createdAt

フリートが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

deploymentApplicationConfigs

デプロイアプリケーションの設定。

型: DeploymentApplicationConfig オブジェクトの配列

配列メンバー: 定数は 1 項目です。

deploymentConfig

デプロイ設定。

タイプ：DeploymentConfig オブジェクト

failureCode

シミュレーションジョブが失敗した場合の失敗コード:

BadPermissionError

AWS Greengrass では、他のサービスにアクセスする場合にサービスレベルロールの許可が必要です。このロールには AWSGreengrassResourceAccessRolePolicy マネージドポリシーが含まれている必要があります。

ExtractingBundleFailure

ロボットアプリケーションをバンドルから抽出できませんでした。

FailureThresholdBreached

更新できなかったロボットの割合が、デプロイのために設定されている割合を超えました。

GreengrassDeploymentFailed

ロボットアプリケーションをロボットにデプロイできませんでした。

GreengrassGroupVersionDoesNotExist

ロボットに関連付けられている AWS Greengrass グループまたはバージョンが見つかりません。

InternalServerError

内部エラーが発生しました。リクエストを再試行してください。それでも問題が解決しない場合は、詳細をお知らせください。

MissingRobotApplicationArchitecture

ロボットアプリケーションには、ロボットのアーキテクチャに一致するソースがありません。

MissingRobotDeploymentResource

ロボットアプリケーションに対して指定された 1 つ以上のリソースが見つかりません。例えば、ロボットアプリケーションには正しい起動パッケージと起動ファイルがありますか?

PostLaunchFileFailure

起動後スクリプトが失敗しました。

PreLaunchFileFailure

起動前スクリプトが失敗しました。

ResourceNotFound

1 つ以上のデプロイリソースが見つかりません。例えば、ロボットアプリケーションのソースバンドルはまだ存在していますか?

RobotDeploymentNoResponse

ロボットからの反応がありません。電源が入っていないか、インターネットに接続されていない可能性があります。

型: 文字列

有効な値 : `ResourceNotFound | EnvironmentSetupError | EtagMismatch | FailureThresholdBreached | RobotDeploymentAborted | RobotDeploymentNoResponse | RobotAgentConnectionTimeout | GreengrassDeploymentFailed | InvalidGreengrassGroup |`

MissingRobotArchitecture | MissingRobotApplicationArchitecture | MissingRobotDeploymentResource | GreengrassGroupVersionDoesNotExist | LambdaDeleted | ExtractingBundleFailure | PreLaunchFileFailure | PostLaunchFileFailure | BadPermissionError | DownloadConditionFailed | BadLambdaAssociated | InternalServerError | RobotApplicationDoesNotExist | DeploymentFleetDoesNotExist | FleetDeploymentTimeout

failureReason

デプロイジョブが失敗した場合の失敗の理由。

型: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: .*

fleet

デプロイジョブのターゲットフリート。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: arn:.*

status

デプロイジョブのステータス。

型: 文字列

有効な値：Pending | Preparing | InProgress | Failed | Succeeded | Canceled

tags

デプロイジョブに追加されたすべてのタグのリスト。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: [a-zA-Z0-9 _.\-\/+=:]*

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

ConcurrentDeploymentException

障害割合のしきい値割合が満たされました。

HTTP ステータスコード：400

IdempotentParameterMismatchException

リクエストは、前の、しかし同一ではないリクエストと同じクライアントトークンを使用します。リクエストが同一でない限り、異なるリクエストでクライアントトークンを再利用しないでください。

HTTP ステータスコード：400

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

LimitExceededException

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

CreateFleet

このアクションは非推奨になりました。

> **Important**
>
> この API はサポートされなくなったため、使用した場合はエラーがスローされます。詳細については、「サポートポリシー」ページの 2022 年 1 月 31 日の更新情報を参照してください。

同じロボットアプリケーションを実行しているロボットの論理グループを表すフリートを作成します。

リクエストの構文

```
POST /createFleet HTTP/1.1
Content-type: application/json

{
   "name": "string",
   "tags": {
      "string" : "string"
   }
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

name

フリートの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: [a-zA-Z0-9_\-]*

必須：はい

tags

フリートにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: [a-zA-Z0-9 _.\-\/+=:]*

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: [a-zA-Z0-9 _.\-\/+=:]*

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "createdAt": number,
   "name": "string",
   "tags": {
      "string" : "string"
   }
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

フリートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

createdAt

フリートが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

name

フリートの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

tags

フリートに追加されたすべてのタグのリスト。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

LimitExceededException

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

CreateRobot

このアクションは非推奨になりました。

> **Important**
>
> この API はサポートされなくなったため、使用した場合はエラーがスローされます。詳細については、「サポートポリシー」ページの 2022 年 1 月 31 日の更新情報を参照してください。

ロボットを作成します。

リクエストの構文

```
POST /createRobot HTTP/1.1
Content-type: application/json

{
   "architecture": "string",
   "greengrassGroupId": "string",
   "name": "string",
   "tags": {
      "string" : "string"
   }
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

architecture

ロボットのターゲットアーキテクチャ。

型: 文字列

有効な値：X86_64 | ARM64 | ARMHF

必須: はい

greengrassGroupId

Greengrass グループの ID。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: .*

必須：はい

name

ロボットの名前。

型: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: [a-zA-Z0-9_\-]*

必須：はい

tags

ロボットにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: [a-zA-Z0-9 _.\-\/+=:]*

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: [a-zA-Z0-9 _.\-\/+=:]*

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "architecture": "string",
   "arn": "string",
   "createdAt": number,
   "greengrassGroupId": "string",
   "name": "string",
   "tags": {
      "string" : "string"
   }
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

architecture

ロボットのターゲットアーキテクチャ。

型: 文字列

有効な値：X86_64 | ARM64 | ARMHF

arn

ロボットの Amazon リソースネーム (ARN)

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: arn:.*

createdAt

ロボットが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

greengrassGroupId

ロボットに関連付けられている Greengrass グループの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `.*`

name

ロボットの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

tags

ロボットに追加されたすべてのタグのリスト。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

LimitExceededException

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

HTTP ステータスコード：400

ResourceAlreadyExistsException

指定したリソースはすでに存在しています。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

CreateRobotApplication

ロボットアプリケーションを作成します。

リクエストの構文

```
POST /createRobotApplication HTTP/1.1
Content-type: application/json

{
   "environment": {
      "uri": "string"
   },
   "name": "string",
   "robotSoftwareSuite": {
      "name": "string",
      "version": "string"
   },
   "sources": [
      {
         "architecture": "string",
         "s3Bucket": "string",
         "s3Key": "string"
      }
   ],
   "tags": {
      "string" : "string"
   }
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

environment

ロボットアプリケーションに使用する Docker イメージの URI を含むオブジェクト。

タイプ：Environment オブジェクト

必須: いいえ

name

ロボットアプリケーションの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: [a-zA-Z0-9_\-]*

必須：はい

robotSoftwareSuite

ロボットアプリケーションで使用するロボットソフトウェアスイート。

型: RobotSoftwareSuite オブジェクト

必須: はい

sources

ロボットアプリケーションのソース。

型: SourceConfig オブジェクトの配列

必須: いいえ

tags

ロボットアプリケーションにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: [a-zA-Z0-9 _.\-\/+=:]*

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: [a-zA-Z0-9 _.\-\/+=:]*

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "environment": {
      "uri": "string"
   },
   "lastUpdatedAt": number,
   "name": "string",
   "revisionId": "string",
   "robotSoftwareSuite": {
      "name": "string",
      "version": "string"
   },
   "sources": [
      {
         "architecture": "string",
         "etag": "string",
         "s3Bucket": "string",
         "s3Key": "string"
      }
   ],
   "tags": {
      "string" : "string"
   },
   "version": "string"
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

ロボットアプリケーションの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

environment

ロボットアプリケーションの作成に使用した Docker イメージ URI が含まれているオブジェクト。

タイプ：Environment オブジェクト

lastUpdatedAt

ロボットアプリケーションが最後に更新されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

name

ロボットアプリケーションの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

revisionId

ロボットアプリケーションのリビジョン ID。

型: 文字列

長さの制限：最小長は 1 です。最大長は 40 です。

Pattern: `[a-zA-Z0-9_.\-]*`

robotSoftwareSuite

ロボットアプリケーションで使用するロボットソフトウェアスイート。

タイプ：RobotSoftwareSuite オブジェクト

sources

ロボットアプリケーションのソース。

型: Source オブジェクトの配列

tags

ロボットアプリケーションに追加されたすべてのタグのリスト。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

version

ロボットアプリケーションのバージョン。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン：`(\$LATEST)|[0-9]*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

IdempotentParameterMismatchException

リクエストは、前の、しかし同一ではないリクエストと同じクライアントトークンを使用します。リクエストが同一でない限り、異なるリクエストでクライアントトークンを再利用しないでください。

HTTP ステータスコード：400

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

LimitExceededException

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

HTTP ステータスコード：400

ResourceAlreadyExistsException

指定したリソースはすでに存在しています。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

CreateRobotApplicationVersion

ロボットアプリケーションのバージョンを作成します。

リクエストの構文

```
POST /createRobotApplicationVersion HTTP/1.1
Content-type: application/json

{
   "application": "string",
   "currentRevisionId": "string",
   "imageDigest": "string",
   "s3Etags": [ "string" ]
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

application

ロボットアプリケーションのアプリケーション情報。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

currentRevisionId

ロボットアプリケーションの現在のリビジョン ID。値を指定して、その値が最新のリビジョン ID と一致する場合、新しいバージョンが作成されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 40 です。

Pattern: `[a-zA-Z0-9_.\-]*`

必須: いいえ

imageDigest

ロボットアプリケーションに使用する Docker イメージの SHA256 識別子。

型: 文字列

長さの制限: 最小長は 0 です。最大長は 72 です。

Pattern: `[Ss][Hh][Aa]256:[0-9a-fA-F]{64}`

必須: いいえ

s3Etags

ロボットアプリケーションに使用する zip ファイルバンドルの Amazon S3 識別子。

タイプ：文字列の配列

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "environment": {
      "uri": "string"
   },
   "lastUpdatedAt": number,
   "name": "string",
   "revisionId": "string",
   "robotSoftwareSuite": {
      "name": "string",
      "version": "string"
   },
   "sources": [
```

```
      {
         "architecture": "string",
         "etag": "string",
         "s3Bucket": "string",
         "s3Key": "string"
      }
   ],
   "version": "string"
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

ロボットアプリケーションの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

environment

ロボットアプリケーションの作成に使用した Docker イメージ URI が含まれているオブジェクト。

タイプ：Environment オブジェクト

lastUpdatedAt

ロボットアプリケーションが最後に更新されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

name

ロボットアプリケーションの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

revisionId

ロボットアプリケーションのリビジョン ID。

型: 文字列

長さの制限：最小長は 1 です。最大長は 40 です。

Pattern: `[a-zA-Z0-9_.\-]*`

robotSoftwareSuite

ロボットアプリケーションで使用するロボットソフトウェアスイート。

タイプ：RobotSoftwareSuite オブジェクト

sources

ロボットアプリケーションのソース。

型: Source オブジェクトの配列

version

ロボットアプリケーションのバージョン。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン：`(\$LATEST)|[0-9]*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

IdempotentParameterMismatchException

リクエストは、前の、しかし同一ではないリクエストと同じクライアントトークンを使用します。リクエストが同一でない限り、異なるリクエストでクライアントトークンを再利用しないでください。

HTTP ステータスコード：400

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

LimitExceededException

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

CreateSimulationApplication

シミュレーションアプリケーションを作成します。

リクエストの構文

```
POST /createSimulationApplication HTTP/1.1
Content-type: application/json

{
   "environment": {
      "uri": "string"
   },
   "name": "string",
   "renderingEngine": {
      "name": "string",
      "version": "string"
   },
   "robotSoftwareSuite": {
      "name": "string",
      "version": "string"
   },
   "simulationSoftwareSuite": {
      "name": "string",
      "version": "string"
   },
   "sources": [
      {
         "architecture": "string",
         "s3Bucket": "string",
         "s3Key": "string"
      }
   ],
   "tags": {
      "string" : "string"
   }
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

environment

シミュレーションアプリケーションの作成に使用した Docker イメージ URI が含まれているオブジェクト。

タイプ：Environment オブジェクト

必須: いいえ

name

シミュレーションアプリケーションの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

必須：はい

renderingEngine

シミュレーションアプリケーションのレンダリングエンジン。

タイプ：RenderingEngine オブジェクト

必須: いいえ

robotSoftwareSuite

シミュレーションアプリケーションによって使用されるロボットソフトウェアスイート。

型: RobotSoftwareSuite オブジェクト

必須: はい

simulationSoftwareSuite

シミュレーションアプリケーションで使用するシミュレーションソフトウェアスイート。

型: SimulationSoftwareSuite オブジェクト

必須: はい

sources

シミュレーションアプリケーションのソース。

型: SourceConfig オブジェクトの配列

必須: いいえ

tags

シミュレーションアプリケーションにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "environment": {
      "uri": "string"
   },
   "lastUpdatedAt": number,
   "name": "string",
   "renderingEngine": {
      "name": "string",
      "version": "string"
```

```
    },
    "revisionId": "string",
    "robotSoftwareSuite": {
       "name": "string",
       "version": "string"
    },
    "simulationSoftwareSuite": {
       "name": "string",
       "version": "string"
    },
    "sources": [
       {
          "architecture": "string",
          "etag": "string",
          "s3Bucket": "string",
          "s3Key": "string"
       }
    ],
    "tags": {
       "string" : "string"
    },
    "version": "string"
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

シミュレーションアプリケーションの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

environment

シミュレーションアプリケーションの作成に使用した Docker イメージ URI が含まれているオブジェクト。

タイプ：Environment オブジェクト

lastUpdatedAt

シミュレーションアプリケーションが最後に更新されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

name

シミュレーションアプリケーションの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: [a-zA-Z0-9_\-]*

renderingEngine

シミュレーションアプリケーションのレンダリングエンジン。

タイプ：RenderingEngine オブジェクト

revisionId

シミュレーションアプリケーションのリビジョン ID。

型: 文字列

長さの制限：最小長は 1 です。最大長は 40 です。

Pattern: [a-zA-Z0-9_.\-]*

robotSoftwareSuite

ロボットソフトウェアスイートに関する情報。

タイプ：RobotSoftwareSuite オブジェクト

simulationSoftwareSuite

シミュレーションアプリケーションで使用するシミュレーションソフトウェアスイート。

タイプ：SimulationSoftwareSuite オブジェクト

sources

シミュレーションアプリケーションのソース。

型: Source オブジェクトの配列

tags

シミュレーションアプリケーションに追加されたすべてのタグのリスト。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

version

シミュレーションアプリケーションのバージョン。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン：`(\$LATEST)|[0-9]*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

IdempotentParameterMismatchException

リクエストは、前の、しかし同一ではないリクエストと同じクライアントトークンを使用します。リクエストが同一でない限り、異なるリクエストでクライアントトークンを再利用しないでください。

HTTP ステータスコード：400

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

LimitExceededException

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

HTTP ステータスコード：400

ResourceAlreadyExistsException

指定したリソースはすでに存在しています。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK

- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

CreateSimulationApplicationVersion

特定のリビジョン ID を持つシミュレーションアプリケーションを作成します。

リクエストの構文

```
POST /createSimulationApplicationVersion HTTP/1.1
Content-type: application/json

{
   "application": "string",
   "currentRevisionId": "string",
   "imageDigest": "string",
   "s3Etags": [ "string" ]
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

application

シミュレーションアプリケーションのアプリケーション情報。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

currentRevisionId

シミュレーションアプリケーションの現在のリビジョン ID。値を指定して、その値が最新のリビジョン ID と一致する場合、新しいバージョンが作成されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 40 です。

Pattern: `[a-zA-Z0-9_.\-]*`

必須: いいえ

imageDigest

シミュレーションアプリケーションの作成に使用された Docker イメージ URI の識別に使用される SHA256 ダイジェスト。

型: 文字列

長さの制限: 最小長は 0 です。最大長は 72 です。

Pattern: `[Ss][Hh][Aa]256:[0-9a-fA-F]{64}`

必須: いいえ

s3Etags

シミュレーションアプリケーションに使用する zip ファイルバンドルの Amazon S3 eTag 識別子。

タイプ : 文字列の配列

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "environment": {
      "uri": "string"
   },
   "lastUpdatedAt": number,
   "name": "string",
   "renderingEngine": {
      "name": "string",
      "version": "string"
   },
   "revisionId": "string",
   "robotSoftwareSuite": {
```

```
      "name": "string",
      "version": "string"
   },
   "simulationSoftwareSuite": {
      "name": "string",
      "version": "string"
   },
   "sources": [
      {
         "architecture": "string",
         "etag": "string",
         "s3Bucket": "string",
         "s3Key": "string"
      }
   ],
   "version": "string"
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

シミュレーションアプリケーションの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

environment

シミュレーションアプリケーションの作成に使用した Docker イメージ URI が含まれているオブジェクト。

タイプ：Environment オブジェクト

lastUpdatedAt

シミュレーションアプリケーションが最後に更新されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

name

シミュレーションアプリケーションの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: [a-zA-Z0-9_\-]*

renderingEngine

シミュレーションアプリケーションのレンダリングエンジン。

タイプ：RenderingEngine オブジェクト

revisionId

シミュレーションアプリケーションのリビジョン ID。

型: 文字列

長さの制限：最小長は 1 です。最大長は 40 です。

Pattern: [a-zA-Z0-9_.\-]*

robotSoftwareSuite

ロボットソフトウェアスイートに関する情報。

タイプ：RobotSoftwareSuite オブジェクト

simulationSoftwareSuite

シミュレーションアプリケーションで使用するシミュレーションソフトウェアスイート。

タイプ：SimulationSoftwareSuite オブジェクト

sources

シミュレーションアプリケーションのソース。

型: Source オブジェクトの配列

version

シミュレーションアプリケーションのバージョン。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン：(\$LATEST)|[0-9]*

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

IdempotentParameterMismatchException

リクエストは、前の、しかし同一ではないリクエストと同じクライアントトークンを使用します。リクエストが同一でない限り、異なるリクエストでクライアントトークンを再利用しないでください。

HTTP ステータスコード：400

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

LimitExceededException

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

CreateSimulationJob

シミュレーションジョブを作成します。

> **Note**
>
> 90 日後、シミュレーションジョブは期限切れになり、削除されます。これらのジョブにはアクセスできなくなります。

リクエストの構文

```
POST /createSimulationJob HTTP/1.1
Content-type: application/json

{
   "clientRequestToken": "string",
   "compute": {
      "computeType": "string",
      "gpuUnitLimit": number,
      "simulationUnitLimit": number
   },
   "dataSources": [
      {
         "destination": "string",
         "name": "string",
         "s3Bucket": "string",
         "s3Keys": [ "string" ],
         "type": "string"
      }
   ],
   "failureBehavior": "string",
   "iamRole": "string",
   "loggingConfig": {
      "recordAllRosTopics": boolean
   },
   "maxJobDurationInSeconds": number,
   "outputLocation": {
      "s3Bucket": "string",
      "s3Prefix": "string"
   },
   "robotApplications": [
```

```
      {
         "application": "string",
         "applicationVersion": "string",
         "launchConfig": {
            "command": [ "string" ],
            "environmentVariables": {
               "string" : "string"
            },
            "launchFile": "string",
            "packageName": "string",
            "portForwardingConfig": {
               "portMappings": [
                  {
                     "applicationPort": number,
                     "enableOnPublicIp": boolean,
                     "jobPort": number
                  }
               ]
            },
            "streamUI": boolean
         },
         "tools": [
            {
               "command": "string",
               "exitBehavior": "string",
               "name": "string",
               "streamOutputToCloudWatch": boolean,
               "streamUI": boolean
            }
         ],
         "uploadConfigurations": [
            {
               "name": "string",
               "path": "string",
               "uploadBehavior": "string"
            }
         ],
         "useDefaultTools": boolean,
         "useDefaultUploadConfigurations": boolean
      }
   ],
   "simulationApplications": [
      {
         "application": "string",
```

```
         "applicationVersion": "string",
         "launchConfig": {
            "command": [ "string" ],
            "environmentVariables": {
               "string" : "string"
            },
            "launchFile": "string",
            "packageName": "string",
            "portForwardingConfig": {
               "portMappings": [
                  {
                     "applicationPort": number,
                     "enableOnPublicIp": boolean,
                     "jobPort": number
                  }
               ]
            },
            "streamUI": boolean
         },
         "tools": [
            {
               "command": "string",
               "exitBehavior": "string",
               "name": "string",
               "streamOutputToCloudWatch": boolean,
               "streamUI": boolean
            }
         ],
         "uploadConfigurations": [
            {
               "name": "string",
               "path": "string",
               "uploadBehavior": "string"
            }
         ],
         "useDefaultTools": boolean,
         "useDefaultUploadConfigurations": boolean,
         "worldConfigs": [
            {
               "world": "string"
            }
         ]
      }
   ],
```

```
    "tags": {
       "string" : "string"
    },
    "vpcConfig": {
       "assignPublicIp": boolean,
       "securityGroups": [ "string" ],
       "subnets": [ "string" ]
    }
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

clientRequestToken

リクエストのべき等のために割り当てる一意の識別子 (大文字と小文字を区別)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 64 文字です。

パターン: `[a-zA-Z0-9_\-=]*`

必須: いいえ

compute

シミュレーションジョブの情報のコンピューティングを行います。

タイプ：Compute オブジェクト

必須: いいえ

dataSources

データソースを指定して、S3 からシミュレーションに読み取り専用ファイルをマウントします。これらのファイルは `/opt/robomaker/datasources/data_source_name` で入手できます。

> **Note**
>
> ファイル数は 100 個、全 DataSourceConfig オブジェクトの合計サイズは 25GB に制限されます。

型: DataSourceConfig オブジェクトの配列

配列メンバー：最小数は 1 項目です。最大数は 6 項目です。

必須: いいえ

failureBehavior

シミュレーションジョブの失敗動作。

続行

4XX エラーコード後の最大タイムアウト期間中もインスタンスが実行されるようにします。

失敗

シミュレーションジョブを停止し、インスタンスを終了します。

型: 文字列

有効な値：Fail | Continue

必須：いいえ

iamRole

関連ポリシーで指定されたAWS APIがシミュレーションインスタンスによって呼び出されるようにする IAM ロール名。これは、シミュレーションジョブに認証情報が渡される方法になります。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: arn:aws:iam::\w+:role/.*

必須：はい

loggingConfig

ログ処理の設定。

タイプ：LoggingConfig オブジェクト

必須: いいえ

maxJobDurationInSeconds

シミュレーションジョブの最長期間 (秒) (最長 14 日または 1,209,600 秒)。maxJobDurationInSeconds に達すると、シミュレーションジョブのステータスが Completed に移行します。

タイプ: Long

必須: はい

outputLocation

シミュレーションジョブにより生成される出力ファイルの場所。

タイプ : OutputLocation オブジェクト

必須: いいえ

robotApplications

シミュレーションジョブで使用するロボットアプリケーション。

型: RobotApplicationConfig オブジェクトの配列

配列メンバー: 定数は 1 項目です。

必須: いいえ

simulationApplications

シミュレーションジョブで使用するシミュレーションアプリケーション。

型: SimulationApplicationConfig オブジェクトの配列

配列メンバー: 定数は 1 項目です。

必須: いいえ

tags

シミュレーションジョブにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: [a-zA-Z0-9 _.\-\/+=:]*

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: [a-zA-Z0-9 _.\-\/+=:]*

必須: いいえ

vpcConfig

シミュレーションジョブが VPC 内のリソースにアクセスする場合は、セキュリティグループ ID とサブネット ID のリストを識別するこのパラメータを指定します。これらは同じ VPC に属している必要があります。少なくとも 1 つのセキュリティグループと 1 つのサブネット ID を指定する必要があります。

タイプ : VPCConfig オブジェクト

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "clientRequestToken": "string",
   "compute": {
      "computeType": "string",
      "gpuUnitLimit": number,
      "simulationUnitLimit": number
   },
   "dataSources": [
      {
         "destination": "string",
         "name": "string",
         "s3Bucket": "string",
         "s3Keys": [
            {
               "etag": "string",
               "s3Key": "string"
            }
         ],
```

```
        "type": "string"
      }
   ],
   "failureBehavior": "string",
   "failureCode": "string",
   "iamRole": "string",
   "lastStartedAt": number,
   "lastUpdatedAt": number,
   "loggingConfig": {
      "recordAllRosTopics": boolean
   },
   "maxJobDurationInSeconds": number,
   "outputLocation": {
      "s3Bucket": "string",
      "s3Prefix": "string"
   },
   "robotApplications": [
      {
         "application": "string",
         "applicationVersion": "string",
         "launchConfig": {
            "command": [ "string" ],
            "environmentVariables": {
               "string" : "string"
            },
            "launchFile": "string",
            "packageName": "string",
            "portForwardingConfig": {
               "portMappings": [
                  {
                     "applicationPort": number,
                     "enableOnPublicIp": boolean,
                     "jobPort": number
                  }
               ]
            },
            "streamUI": boolean
         },
         "tools": [
            {
               "command": "string",
               "exitBehavior": "string",
               "name": "string",
               "streamOutputToCloudWatch": boolean,
```

```
            "streamUI": boolean
         }
      ],
      "uploadConfigurations": [
         {
            "name": "string",
            "path": "string",
            "uploadBehavior": "string"
         }
      ],
      "useDefaultTools": boolean,
      "useDefaultUploadConfigurations": boolean
   }
],
"simulationApplications": [
   {
      "application": "string",
      "applicationVersion": "string",
      "launchConfig": {
         "command": [ "string" ],
         "environmentVariables": {
            "string" : "string"
         },
         "launchFile": "string",
         "packageName": "string",
         "portForwardingConfig": {
            "portMappings": [
               {
                  "applicationPort": number,
                  "enableOnPublicIp": boolean,
                  "jobPort": number
               }
            ]
         },
         "streamUI": boolean
      },
      "tools": [
         {
            "command": "string",
            "exitBehavior": "string",
            "name": "string",
            "streamOutputToCloudWatch": boolean,
            "streamUI": boolean
         }
```

```
        ],
        "uploadConfigurations": [
           {
              "name": "string",
              "path": "string",
              "uploadBehavior": "string"
           }
        ],
        "useDefaultTools": boolean,
        "useDefaultUploadConfigurations": boolean,
        "worldConfigs": [
           {
              "world": "string"
           }
        ]
     }
  ],
  "simulationTimeMillis": number,
  "status": "string",
  "tags": {
     "string" : "string"
  },
  "vpcConfig": {
     "assignPublicIp": boolean,
     "securityGroups": [ "string" ],
     "subnets": [ "string" ],
     "vpcId": "string"
  }
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

シミュレーションジョブの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

clientRequestToken

リクエストのべき等のために割り当てる一意の識別子 (大文字と小文字を区別)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 64 文字です。

パターン: `[a-zA-Z0-9_\-=]*`

compute

シミュレーションジョブの情報のコンピューティングを行います。

タイプ：ComputeResponse オブジェクト

dataSources

シミュレーションジョブのデータソース。

型: DataSource オブジェクトの配列

failureBehavior

シミュレーションジョブの失敗動作。

型: 文字列

有効な値：`Fail | Continue`

failureCode

シミュレーションジョブが失敗した場合の失敗コード:

InternalServiceError

内部サービスエラー。

RobotApplicationCrash

ロボットアプリケーションが異常終了しました。

SimulationApplicationCrash

シミュレーションアプリケーションが異常終了しました。

BadPermissionsRobotApplication

ロボットアプリケーションバンドルをダウンロードできませんでした。

BadPermissionsSimulationApplication

シミュレーションアプリケーションバンドルをダウンロードできませんでした。

BadPermissionsS3 アウトプット

お客様が用意した S3 バケットに出力を発行できません。

BadPermissionsCloudwatchLogs

CloudWatch お客様が提供した Logs リソースにログを公開できません。

SubnetIpLimitExceeded

サブネット IP 限界を超えました

ENI LimitExceeded

ENI 限界を超えました。

BadPermissionsUserCredentials

提供されたロールを使用できません。

InvalidBundleRobotApplication

ロボットバンドルを抽出できません (無効な形式、バンドルエラー、またはその他の問題)。

InvalidBundleSimulationApplication

シミュレーションバンドルを抽出できません (無効な形式、バンドルエラー、またはその他の問題)。

RobotApplicationVersionMismatchedEtag

バージョン作成中に Etag RobotApplication の値が一致しない。

SimulationApplicationVersionMismatchedEtag

Etag for SimulationApplication はバージョン作成中の値と一致しません。

型: 文字列

有効な値 : `InternalServiceError | RobotApplicationCrash | SimulationApplicationCrash | RobotApplicationHealthCheckFailure |`

SimulationApplicationHealthCheckFailure | BadPermissionsRobotApplication | BadPermissionsSimulationApplication | BadPermissionsS3Object | BadPermissionsS3Output | BadPermissionsCloudwatchLogs | SubnetIpLimitExceeded | ENILimitExceeded | BadPermissionsUserCredentials | InvalidBundleRobotApplication | InvalidBundleSimulationApplication | InvalidS3Resource | ThrottlingError | LimitExceeded | MismatchedEtag | RobotApplicationVersionMismatchedEtag | SimulationApplicationVersionMismatchedEtag | ResourceNotFound | RequestThrottled | BatchTimedOut | BatchCanceled | InvalidInput | WrongRegionS3Bucket | WrongRegionS3Output | WrongRegionRobotApplication | WrongRegionSimulationApplication | UploadContentMismatchError

iamRole

関連ポリシーで指定されたAWS APIがシミュレーションジョブによって呼び出されるようにするIAM ロール。

型: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: arn:aws:iam::\w+:role/.*

lastStartedAt

シミュレーションジョブが最後に開始されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

lastUpdatedAt

シミュレーションジョブが最後に更新されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

loggingConfig

ログ処理の設定。

タイプ：LoggingConfig オブジェクト

maxJobDurationInSeconds

最大シミュレーションジョブ時間 (秒)。

型: 長整数

outputLocation

シミュレーションジョブの出力ファイルの場所。

タイプ：OutputLocation オブジェクト

robotApplications

シミュレーションジョブで使用されたロボットアプリケーション。

型: RobotApplicationConfig オブジェクトの配列

配列メンバー: 定数は 1 項目です。

simulationApplications

シミュレーションジョブにより使用されたシミュレーションアプリケーション。

型: SimulationApplicationConfig オブジェクトの配列

配列メンバー: 定数は 1 項目です。

simulationTimeMillis

シミュレーションジョブの実行時間 (ミリ秒)。

型: 長整数

status

シミュレーションジョブのステータス。

型: 文字列

有効な値：`Pending | Preparing | Running | Restarting | Completed | Failed | RunningFailed | Terminating | Terminated | Canceled`

tags

シミュレーションジョブに追加されたすべてのタグのリスト。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

vpcConfig

VPC 設定に関する情報。

型: VPCConfigResponse オブジェクト

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

IdempotentParameterMismatchException

リクエストは、前の、しかし同一ではないリクエストと同じクライアントトークンを使用します。リクエストが同一でない限り、異なるリクエストでクライアントトークンを再利用しないでください。

HTTP ステータスコード：400

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

LimitExceededException

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ServiceUnavailableException

サーバーの一時的な障害により、リクエストは失敗しました。

HTTP ステータスコード: 503

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

CreateWorldExportJob

ワールドのエクスポートジョブを作成します。

リクエストの構文

```
POST /createWorldExportJob HTTP/1.1
Content-type: application/json

{
   "clientRequestToken": "string",
   "iamRole": "string",
   "outputLocation": {
      "s3Bucket": "string",
      "s3Prefix": "string"
   },
   "tags": {
      "string" : "string"
   },
   "worlds": [ "string" ]
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

clientRequestToken

リクエストのべき等のために割り当てる一意の識別子 (大文字と小文字を区別)。

型: 文字列

長さの制限 : 最小長は 1 です。最大長は 64 文字です。

パターン: [a-zA-Z0-9_\-=]*

必須: いいえ

iamRole

ワールドエクスポートプロセスで Amazon S3 バケットへのアクセスとエクスポートの実行に使用される IAM ロール。

型: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `arn:aws:iam::\w+:role/.*`

必須：はい

outputLocation

出力場所。

型: OutputLocation オブジェクト

必須: はい

tags

ワールドエクスポートジョブにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

必須: いいえ

worlds

エクスポートするワールドに対応する Amazon リソースネーム (ARN) の一覧。

タイプ：文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 100 項目です。

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "clientRequestToken": "string",
   "createdAt": number,
   "failureCode": "string",
   "iamRole": "string",
   "outputLocation": {
      "s3Bucket": "string",
      "s3Prefix": "string"
   },
   "status": "string",
   "tags": {
      "string" : "string"
   }
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

ワールドエクスポートジョブの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

clientRequestToken

リクエストのべき等のために割り当てる一意の識別子 (大文字と小文字を区別)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 64 文字です。

パターン: [a-zA-Z0-9_\-=]*

createdAt

ワールドのエクスポートジョブが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

failureCode

ワールドエクスポートジョブが失敗した場合の失敗コード:

InternalServiceError

内部サービスエラー。

LimitExceeded

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

ResourceNotFound

指定したリソースが見つかりませんでした。

RequestThrottled

リクエストがスロットリングされました。

InvalidInput

リクエストの入力パラメータが無効です。

AllWorldGenerationFailed

ワールド生成ジョブのすべてのワールドが失敗しました。これは、`worldCount` が 50 より大きい場合、または 1 より小さい場合に発生する可能性があります。

トラブルシューティングの詳細については WorldForge、「トラブルシューティングシミュレーション」を参照してください WorldForge。

型: 文字列

有効な値：`InternalServiceError | LimitExceeded | ResourceNotFound | RequestThrottled | InvalidInput | AccessDenied`

iamRole

ワールドエクスポートプロセスで Amazon S3 バケットへのアクセスとエクスポートの実行に使用される IAM ロール。

型: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `arn:aws:iam::\w+:role/.*`

outputLocation

出力場所。

タイプ：OutputLocation オブジェクト

status

ワールドエクスポートジョブのステータス。

保留中

ワールドエクスポートジョブのリクエストが保留中です。

実行中

ワールドエクスポートジョブが実行中です。

完了

ワールドエクスポートジョブが完了しました。

[失敗]

ワールドエクスポートジョブが失敗しました。詳細については「`failureCode`」を参照してください。

キャンセル

ワールドエクスポートジョブがキャンセルされました。

キャンセル中

ワールドエクスポートジョブをキャンセルしています。

型: 文字列

有効な値 : `Pending | Running | Completed | Failed | Canceling | Canceled`

tags

ワールドエクスポートジョブにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

IdempotentParameterMismatchException

リクエストは、前の、しかし同一ではないリクエストと同じクライアントトークンを使用します。リクエストが同一でない限り、異なるリクエストでクライアントトークンを再利用しないでください。

HTTP ステータスコード : 400

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード : 500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード : 400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ServiceUnavailableException

サーバーの一時的な障害により、リクエストは失敗しました。

HTTP ステータスコード: 503

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

CreateWorldGenerationJob

指定したテンプレートを使用してワールドを作成します。

リクエストの構文

```
POST /createWorldGenerationJob HTTP/1.1
Content-type: application/json

{
   "clientRequestToken": "string",
   "tags": {
      "string" : "string"
   },
   "template": "string",
   "worldCount": {
      "floorplanCount": number,
      "interiorCountPerFloorplan": number
   },
   "worldTags": {
      "string" : "string"
   }
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

clientRequestToken

リクエストのべき等のために割り当てる一意の識別子 (大文字と小文字を区別)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 64 文字です。

パターン: `[a-zA-Z0-9_\-=]*`

必須: いいえ

tags

ワールドジェネレータージョブにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

必須: いいえ

template

作成するワールドを表すワールドテンプレートの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

worldCount

ワールドカウントに関する情報。

型: WorldCount オブジェクト

必須: はい

worldTags

生成したワールドにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: [a-zA-Z0-9 _.\-\/+=:]*

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: [a-zA-Z0-9 _.\-\/+=:]*

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "clientRequestToken": "string",
   "createdAt": number,
   "failureCode": "string",
   "status": "string",
   "tags": {
      "string" : "string"
   },
   "template": "string",
   "worldCount": {
      "floorplanCount": number,
      "interiorCountPerFloorplan": number
   },
   "worldTags": {
      "string" : "string"
   }
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

ワールドジェネレータージョブの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

clientRequestToken

リクエストのべき等のために割り当てる一意の識別子 (大文字と小文字を区別)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 64 文字です。

パターン: `[a-zA-Z0-9_\-=]*`

createdAt

ワールドのジェネレータージョブが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

failureCode

ワールドジェネレータージョブが失敗した場合の失敗コード:

InternalServiceError

内部サービスエラー。

LimitExceeded

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

ResourceNotFound

指定したリソースが見つかりませんでした。

RequestThrottled

リクエストがスロットリングされました。

InvalidInput

リクエストの入力パラメータが無効です。

型: 文字列

有効な値：`InternalServiceError | LimitExceeded | ResourceNotFound | RequestThrottled | InvalidInput | AllWorldGenerationFailed`

status

ワールドジェネレータージョブのステータス。

保留中

ワールドジェネレータージョブのリクエストが保留中です。

実行中

ワールドジェネレータージョブが実行中です。

完了

ワールドジェネレータージョブが完了しました。

[失敗]

ワールドジェネレータージョブが失敗しました。詳細については、「failureCode」を参照してください。

PartialFailed

一部のワールドが生成されませんでした。

キャンセル

ワールドジェネレータージョブがキャンセルされました。

キャンセル中

ワールドジェネレータージョブをキャンセルしています。

型: 文字列

有効な値 : `Pending | Running | Completed | Failed | PartialFailed | Canceling | Canceled`

tags

ワールドジェネレータージョブにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

template

ワールドテンプレートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

worldCount

ワールドカウントに関する情報。

タイプ：WorldCount オブジェクト

worldTags

生成したワールドにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

IdempotentParameterMismatchException

リクエストは、前の、しかし同一ではないリクエストと同じクライアントトークンを使用します。リクエストが同一でない限り、異なるリクエストでクライアントトークンを再利用しないでください。

HTTP ステータスコード：400

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

LimitExceededException

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ServiceUnavailableException

サーバーの一時的な障害により、リクエストは失敗しました。

HTTP ステータスコード: 503

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス

- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

CreateWorldTemplate

ワールドテンプレートを作成します。

リクエストの構文

```
POST /createWorldTemplate HTTP/1.1
Content-type: application/json

{
   "clientRequestToken": "string",
   "name": "string",
   "tags": {
      "string" : "string"
   },
   "templateBody": "string",
   "templateLocation": {
      "s3Bucket": "string",
      "s3Key": "string"
   }
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

clientRequestToken

リクエストのべき等のために割り当てる一意の識別子 (大文字と小文字を区別)。

型: 文字列

長さの制限 : 最小長は 1 です。最大長は 64 文字です。

パターン: [a-zA-Z0-9_\-=]*

必須: いいえ

name

ワールドテンプレートの名前。

型: 文字列

長さの制限: 最小長は 0 です。最大長は 255 です。

パターン: .*

必須: いいえ

tags

ワールドテンプレートにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: [a-zA-Z0-9 _.\-\/+=:]*

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: [a-zA-Z0-9 _.\-\/+=:]*

必須: いいえ

templateBody

ワールドテンプレートの本文。

型: 文字列

長さの制限：最小長は 1 です。最大長は 262,144 です。

Pattern: [\S\s]+

必須: いいえ

templateLocation

ワールドテンプレートの場所。

タイプ：TemplateLocation オブジェクト

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "clientRequestToken": "string",
   "createdAt": number,
   "name": "string",
   "tags": {
      "string" : "string"
   }
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

ワールドテンプレートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

clientRequestToken

リクエストのべき等のために割り当てる一意の識別子 (大文字と小文字を区別)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 64 文字です。

パターン: `[a-zA-Z0-9_\-=]*`

createdAt

ワールドテンプレートが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

name

ワールドテンプレートの名前。

型: 文字列

長さの制限: 最小長は 0 です。最大長は 255 です。

パターン: .*

tags

ワールドテンプレートにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: [a-zA-Z0-9 _.\-\/+=:]*

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: [a-zA-Z0-9 _.\-\/+=:]*

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

LimitExceededException

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

HTTP ステータスコード：400

ResourceAlreadyExistsException

指定したリソースはすでに存在しています。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

DeleteFleet

このアクションは非推奨になりました。

> **Important**
>
> この API はサポートされなくなりました。詳細については、「サポートポリシー」ページの 2022 年 5 月 2 日の更新をご覧ください。

フリートを削除します。

リクエストの構文

```
POST /deleteFleet HTTP/1.1
Content-type: application/json

{
   "fleet": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

fleet

フリートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
```

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK

- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

DeleteRobot

このアクションは非推奨になりました。

> **Important**
>
> この API はサポートされなくなりました。詳細については、「サポートポリシー」ページの 2022 年 5 月 2 日の更新をご覧ください。

ロボットを削除します。

リクエストの構文

```
POST /deleteRobot HTTP/1.1
Content-type: application/json

{
   "robot": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

robot

ロボットの Amazon リソースネーム (ARN)

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
```

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK

- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

DeleteRobotApplication

ロボットアプリケーションを削除します。

リクエストの構文

```
POST /deleteRobotApplication HTTP/1.1
Content-type: application/json

{
   "application": "string",
   "applicationVersion": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

application

ロボットアプリケーションの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

applicationVersion

削除するロボットアプリケーションのバージョン。

型: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `(\$LATEST)|[0-9]*`

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
```

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++

- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

DeleteSimulationApplication

シミュレーションアプリケーションを削除します。

リクエストの構文

```
POST /deleteSimulationApplication HTTP/1.1
Content-type: application/json

{
   "application": "string",
   "applicationVersion": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

application

削除するシミュレーションアプリケーションのアプリケーション情報。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

applicationVersion

削除するシミュレーションアプリケーションのバージョン。

型: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `(\$LATEST)|[0-9]*`

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
```

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++

- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

DeleteWorldTemplate

ワールドテンプレートを削除します。

リクエストの構文

```
POST /deleteWorldTemplate HTTP/1.1
Content-type: application/json

{
   "template": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

template

削除するワールドテンプレートの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
```

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK

- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

DeregisterRobot

このアクションは非推奨になりました。

> **Important**
>
> この API はサポートされなくなりました。詳細については、「サポートポリシー」ページの 2022 年 5 月 2 日の更新をご覧ください。

ロボットの登録を解除します。

リクエストの構文

```
POST /deregisterRobot HTTP/1.1
Content-type: application/json

{
   "fleet": "string",
   "robot": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

fleet

フリートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

robot

ロボットの Amazon リソースネーム (ARN)

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "fleet": "string",
   "robot": "string"
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

fleet

フリートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

robot

ロボットの Amazon リソースネーム (ARN)

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン：`arn:.*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK

- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

DescribeDeploymentJob

このアクションは非推奨になりました。

> **Important**
>
> この API はサポートされなくなりました。詳細については、「サポートポリシー」ページの 2022 年 5 月 2 日の更新をご覧ください。

デプロイジョブを記述します。

リクエストの構文

```
POST /describeDeploymentJob HTTP/1.1
Content-type: application/json

{
   "job": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

job

デプロイジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "createdAt": number,
   "deploymentApplicationConfigs": [
      {
         "application": "string",
         "applicationVersion": "string",
         "launchConfig": {
            "environmentVariables": {
               "string" : "string"
            },
            "launchFile": "string",
            "packageName": "string",
            "postLaunchFile": "string",
            "preLaunchFile": "string"
         }
      }
   ],
   "deploymentConfig": {
      "concurrentDeploymentPercentage": number,
      "downloadConditionFile": {
         "bucket": "string",
         "etag": "string",
         "key": "string"
      },
      "failureThresholdPercentage": number,
      "robotDeploymentTimeoutInSeconds": number
   },
   "failureCode": "string",
   "failureReason": "string",
   "fleet": "string",
   "robotDeploymentSummary": [
      {
         "arn": "string",
         "deploymentFinishTime": number,
         "deploymentStartTime": number,
         "failureCode": "string",
         "failureReason": "string",
```

```
            "progressDetail": {
               "currentProgress": "string",
               "estimatedTimeRemainingSeconds": number,
               "percentDone": number,
               "targetResource": "string"
            },
            "status": "string"
         }
      ],
      "status": "string",
      "tags": {
         "string" : "string"
      }
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

デプロイジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

createdAt

デプロイジョブが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

deploymentApplicationConfigs

デプロイアプリケーションの設定。

型: DeploymentApplicationConfig オブジェクトの配列

配列メンバー: 定数は 1 項目です。

deploymentConfig

デプロイ設定。

タイプ：DeploymentConfig オブジェクト

failureCode

デプロイジョブの失敗コード。

型: 文字列

有効な値：ResourceNotFound | EnvironmentSetupError | EtagMismatch | FailureThresholdBreached | RobotDeploymentAborted | RobotDeploymentNoResponse | RobotAgentConnectionTimeout | GreengrassDeploymentFailed | InvalidGreengrassGroup | MissingRobotArchitecture | MissingRobotApplicationArchitecture | MissingRobotDeploymentResource | GreengrassGroupVersionDoesNotExist | LambdaDeleted | ExtractingBundleFailure | PreLaunchFileFailure | PostLaunchFileFailure | BadPermissionError | DownloadConditionFailed | BadLambdaAssociated | InternalServerError | RobotApplicationDoesNotExist | DeploymentFleetDoesNotExist | FleetDeploymentTimeout

failureReason

デプロイジョブが失敗した理由の簡単な説明。

型: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: .*

fleet

フリートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: arn:.*

robotDeploymentSummary

ロボットデプロイ概要のリスト。

型: RobotDeployment オブジェクトの配列

status

デプロイジョブのステータス。

型: 文字列

有効な値 : Pending | Preparing | InProgress | Failed | Succeeded | Canceled

tags

指定のデプロイジョブに追加されたすべてのタグのリスト。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: [a-zA-Z0-9 _.\-\/+=:]*

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: [a-zA-Z0-9 _.\-\/+=:]*

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード : 500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード : 400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

DescribeFleet

このアクションは非推奨になりました。

> **Important**
>
> この API はサポートされなくなりました。詳細については、「サポートポリシー」ページの 2022 年 5 月 2 日の更新をご覧ください。

フリートを記述します。

リクエストの構文

```
POST /describeFleet HTTP/1.1
Content-type: application/json

{
   "fleet": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

fleet

フリートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "createdAt": number,
   "lastDeploymentJob": "string",
   "lastDeploymentStatus": "string",
   "lastDeploymentTime": number,
   "name": "string",
   "robots": [
      {
         "architecture": "string",
         "arn": "string",
         "createdAt": number,
         "fleetArn": "string",
         "greenGrassGroupId": "string",
         "lastDeploymentJob": "string",
         "lastDeploymentTime": number,
         "name": "string",
         "status": "string"
      }
   ],
   "tags": {
      "string" : "string"
   }
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

フリートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

createdAt

フリートが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

lastDeploymentJob

最後のデプロイジョブの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

lastDeploymentStatus

最終のデプロイのステータス。

型: 文字列

有効な値：`Pending | Preparing | InProgress | Failed | Succeeded | Canceled`

lastDeploymentTime

最後のデプロイの時間。

型: タイムスタンプ

name

フリートの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

robots

ロボットのリスト。

タイプ：Robot オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 1000 項目です。

tags

指定のフリートに追加されたすべてのタグのリスト。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

DescribeRobot

このアクションは非推奨になりました。

> **Important**
>
> この API はサポートされなくなりました。詳細については、「サポートポリシー」ページの 2022 年 5 月 2 日の更新をご覧ください。

ロボットを記述します。

リクエストの構文

```
POST /describeRobot HTTP/1.1
Content-type: application/json

{
   "robot": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

robot

記述されるロボットの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "architecture": "string",
   "arn": "string",
   "createdAt": number,
   "fleetArn": "string",
   "greengrassGroupId": "string",
   "lastDeploymentJob": "string",
   "lastDeploymentTime": number,
   "name": "string",
   "status": "string",
   "tags": {
      "string" : "string"
   }
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

architecture

ロボットのターゲットアーキテクチャ。

型: 文字列

有効な値：X86_64 | ARM64 | ARMHF

arn

ロボットの Amazon リソースネーム (ARN)

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: arn:.*

createdAt

ロボットが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

fleetArn

フリートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

greengrassGroupId

Greengrass グループの ID。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `.*`

lastDeploymentJob

最後のデプロイジョブの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

lastDeploymentTime

最後のデプロイジョブの時間。

型: タイムスタンプ

name

ロボットの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

status

フリートのステータス。

型: 文字列

有効な値：`Available | Registered | PendingNewDeployment | Deploying | Failed | InSync | NoResponse`

tags

指定の口ボットに追加されたすべてのタグのリスト。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

DescribeRobotApplication

ロボットアプリケーションを記述します。

リクエストの構文

```
POST /describeRobotApplication HTTP/1.1
Content-type: application/json

{
   "application": "string",
   "applicationVersion": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

application

ロボットアプリケーションの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

applicationVersion

記述するロボットアプリケーションのバージョン。

型: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `(\$LATEST)|[0-9]*`

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "environment": {
      "uri": "string"
   },
   "imageDigest": "string",
   "lastUpdatedAt": number,
   "name": "string",
   "revisionId": "string",
   "robotSoftwareSuite": {
      "name": "string",
      "version": "string"
   },
   "sources": [
      {
         "architecture": "string",
         "etag": "string",
         "s3Bucket": "string",
         "s3Key": "string"
      }
   ],
   "tags": {
      "string" : "string"
   },
   "version": "string"
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

ロボットアプリケーションの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

environment

ロボットアプリケーションの作成に使用した Docker イメージ URI が含まれているオブジェクト。

タイプ：Environment オブジェクト

imageDigest

ロボットアプリケーションに使用する Docker イメージの SHA256 識別子。

型: 文字列

長さの制限: 最小長は 0 です。最大長は 72 です。

Pattern: `[Ss][Hh][Aa]256:[0-9a-fA-F]{64}`

lastUpdatedAt

ロボットアプリケーションが最後に更新されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

name

ロボットアプリケーションの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

revisionId

ロボットアプリケーションのリビジョン ID。

型: 文字列

長さの制限：最小長は 1 です。最大長は 40 です。

Pattern: `[a-zA-Z0-9_.\-]*`

robotSoftwareSuite

ロボットアプリケーションで使用するロボットソフトウェアスイート。

タイプ：RobotSoftwareSuite オブジェクト

sources

ロボットアプリケーションのソース。

型: Source オブジェクトの配列

tags

指定のロボットアプリケーションに追加されたすべてのタグのリスト。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

version

ロボットアプリケーションのバージョン。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン：`(\$LATEST)|[0-9]*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

DescribeSimulationApplication

シミュレーションアプリケーションを記述します。

リクエストの構文

```
POST /describeSimulationApplication HTTP/1.1
Content-type: application/json

{
   "application": "string",
   "applicationVersion": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

application

シミュレーションアプリケーションのアプリケーション情報。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

applicationVersion

記述するシミュレーションアプリケーションのバージョン。

型: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `(\$LATEST)|[0-9]*`

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "environment": {
      "uri": "string"
   },
   "imageDigest": "string",
   "lastUpdatedAt": number,
   "name": "string",
   "renderingEngine": {
      "name": "string",
      "version": "string"
   },
   "revisionId": "string",
   "robotSoftwareSuite": {
      "name": "string",
      "version": "string"
   },
   "simulationSoftwareSuite": {
      "name": "string",
      "version": "string"
   },
   "sources": [
      {
         "architecture": "string",
         "etag": "string",
         "s3Bucket": "string",
         "s3Key": "string"
      }
   ],
   "tags": {
      "string" : "string"
   },
   "version": "string"
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

ロボットシミュレーションアプリケーションの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

environment

シミュレーションアプリケーションの作成に使用した Docker イメージ URI が含まれているオブジェクト。

タイプ：Environment オブジェクト

imageDigest

シミュレーションアプリケーションに使用する Docker イメージの SHA256 識別子。

型: 文字列

長さの制限: 最小長は 0 です。最大長は 72 です。

Pattern: `[Ss][Hh][Aa]256:[0-9a-fA-F]{64}`

lastUpdatedAt

シミュレーションアプリケーションが最後に更新されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

name

シミュレーションアプリケーションの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

renderingEngine

シミュレーションアプリケーションのレンダリングエンジン。

タイプ：RenderingEngine オブジェクト

revisionId

シミュレーションアプリケーションのリビジョン ID。

型: 文字列

長さの制限：最小長は 1 です。最大長は 40 です。

Pattern: `[a-zA-Z0-9_.\-]*`

robotSoftwareSuite

ロボットソフトウェアスイートに関する情報。

タイプ：RobotSoftwareSuite オブジェクト

simulationSoftwareSuite

シミュレーションアプリケーションで使用するシミュレーションソフトウェアスイート。

タイプ：SimulationSoftwareSuite オブジェクト

sources

シミュレーションアプリケーションのソース。

型: Source オブジェクトの配列

tags

指定のシミュレーションアプリケーションに追加されたすべてのタグのリスト。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

version

シミュレーションアプリケーションのバージョン。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン：`(\$LATEST)|[0-9]*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

DescribeSimulationJob

シミュレーションジョブを記述します。

リクエストの構文

```
POST /describeSimulationJob HTTP/1.1
Content-type: application/json

{
   "job": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

job

記述されるシミュレーションジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "clientRequestToken": "string",
   "compute": {
      "computeType": "string",
```

```
      "gpuUnitLimit": number,
      "simulationUnitLimit": number
   },
   "dataSources": [
      {
         "destination": "string",
         "name": "string",
         "s3Bucket": "string",
         "s3Keys": [
            {
               "etag": "string",
               "s3Key": "string"
            }
         ],
         "type": "string"
      }
   ],
   "failureBehavior": "string",
   "failureCode": "string",
   "failureReason": "string",
   "iamRole": "string",
   "lastStartedAt": number,
   "lastUpdatedAt": number,
   "loggingConfig": {
      "recordAllRosTopics": boolean
   },
   "maxJobDurationInSeconds": number,
   "name": "string",
   "networkInterface": {
      "networkInterfaceId": "string",
      "privateIpAddress": "string",
      "publicIpAddress": "string"
   },
   "outputLocation": {
      "s3Bucket": "string",
      "s3Prefix": "string"
   },
   "robotApplications": [
      {
         "application": "string",
         "applicationVersion": "string",
         "launchConfig": {
            "command": [ "string" ],
            "environmentVariables": {
```

```
            "string" : "string"
         },
         "launchFile": "string",
         "packageName": "string",
         "portForwardingConfig": {
            "portMappings": [
               {
                  "applicationPort": number,
                  "enableOnPublicIp": boolean,
                  "jobPort": number
               }
            ]
         },
         "streamUI": boolean
      },
      "tools": [
         {
            "command": "string",
            "exitBehavior": "string",
            "name": "string",
            "streamOutputToCloudWatch": boolean,
            "streamUI": boolean
         }
      ],
      "uploadConfigurations": [
         {
            "name": "string",
            "path": "string",
            "uploadBehavior": "string"
         }
      ],
      "useDefaultTools": boolean,
      "useDefaultUploadConfigurations": boolean
   }
],
"simulationApplications": [
   {
      "application": "string",
      "applicationVersion": "string",
      "launchConfig": {
         "command": [ "string" ],
         "environmentVariables": {
            "string" : "string"
         },
```

```
            "launchFile": "string",
            "packageName": "string",
            "portForwardingConfig": {
               "portMappings": [
                  {
                     "applicationPort": number,
                     "enableOnPublicIp": boolean,
                     "jobPort": number
                  }
               ]
            },
            "streamUI": boolean
         },
         "tools": [
            {
               "command": "string",
               "exitBehavior": "string",
               "name": "string",
               "streamOutputToCloudWatch": boolean,
               "streamUI": boolean
            }
         ],
         "uploadConfigurations": [
            {
               "name": "string",
               "path": "string",
               "uploadBehavior": "string"
            }
         ],
         "useDefaultTools": boolean,
         "useDefaultUploadConfigurations": boolean,
         "worldConfigs": [
            {
               "world": "string"
            }
         ]
      }
   ],
   "simulationTimeMillis": number,
   "status": "string",
   "tags": {
      "string" : "string"
   },
   "vpcConfig": {
```

```
        "assignPublicIp": boolean,
        "securityGroups": [ "string" ],
        "subnets": [ "string" ],
        "vpcId": "string"
    }
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

シミュレーションジョブの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

clientRequestToken

リクエストのべき等のために割り当てる一意の識別子 (大文字と小文字を区別)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 64 文字です。

パターン: `[a-zA-Z0-9_\-=]*`

compute

シミュレーションジョブの情報のコンピューティングを行います。

タイプ：ComputeResponse オブジェクト

dataSources

シミュレーションジョブのデータソース。

型: DataSource オブジェクトの配列

failureBehavior

シミュレーションジョブの失敗動作。

型: 文字列

有効な値：`Fail | Continue`

failureCode

シミュレーションジョブが失敗した場合の失敗コード:

InternalServiceError

内部サービスエラー。

RobotApplicationCrash

ロボットアプリケーションが異常終了しました。

SimulationApplicationCrash

シミュレーションアプリケーションが異常終了しました。

BadPermissionsRobotApplication

ロボットアプリケーションバンドルをダウンロードできませんでした。

BadPermissionsSimulationApplication

シミュレーションアプリケーションバンドルをダウンロードできませんでした。

BadPermissionsS3 アウトプット

お客様が用意した S3 バケットに出力を発行できません。

BadPermissionsCloudwatchLogs

CloudWatch お客様が提供した Logs リソースにログを公開できません。

SubnetIpLimitExceeded

サブネット IP 限界を超えました

ENI LimitExceeded

ENI 限界を超えました。

BadPermissionsUserCredentials

提供されたロールを使用できません。

InvalidBundleRobotApplication

ロボットバンドルを抽出できません (無効な形式、バンドルエラー、またはその他の問題)。

InvalidBundleSimulationApplication

シミュレーションバンドルを抽出できません (無効な形式、バンドルエラー、またはその他の問題)。

RobotApplicationVersionMismatchedEtag

バージョン作成中に Etag RobotApplication の値が一致しない。

SimulationApplicationVersionMismatchedEtag

Etag for SimulationApplication はバージョン作成中の値と一致しません。

型: 文字列

有効な値 : `InternalServiceError | RobotApplicationCrash | SimulationApplicationCrash | RobotApplicationHealthCheckFailure | SimulationApplicationHealthCheckFailure | BadPermissionsRobotApplication | BadPermissionsSimulationApplication | BadPermissionsS3Object | BadPermissionsS3Output | BadPermissionsCloudwatchLogs | SubnetIpLimitExceeded | ENILimitExceeded | BadPermissionsUserCredentials | InvalidBundleRobotApplication | InvalidBundleSimulationApplication | InvalidS3Resource | ThrottlingError | LimitExceeded | MismatchedEtag | RobotApplicationVersionMismatchedEtag | SimulationApplicationVersionMismatchedEtag | ResourceNotFound | RequestThrottled | BatchTimedOut | BatchCanceled | InvalidInput | WrongRegionS3Bucket | WrongRegionS3Output | WrongRegionRobotApplication | WrongRegionSimulationApplication | UploadContentMismatchError`

failureReason

シミュレーションジョブが失敗した理由の詳細。トラブルシューティングの詳細については、「トラブルシューティング」を参照してください。

型: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: `.*`

iamRole

関連ポリシーで指定されたAWS APIがシミュレーションインスタンスによって呼び出されるようにする IAM ロール。

型: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `arn:aws:iam::\w+:role/.*`

lastStartedAt

シミュレーションジョブが最後に開始されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

lastUpdatedAt

シミュレーションジョブが最後に更新されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

loggingConfig

ログ処理の設定。

タイプ：LoggingConfig オブジェクト

maxJobDurationInSeconds

最大ジョブ時間 (秒)。値は 8 日 (691,200 秒) 以下でなければなりません。

型: 長整数

name

シミュレーションジョブの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

networkInterface

シミュレーションジョブのネットワークインターフェイス情報。

タイプ：NetworkInterface オブジェクト

outputLocation

シミュレーションジョブにより生成される出力ファイルの場所。

タイプ：OutputLocation オブジェクト

robotApplications

ロボットアプリケーションのリスト。

型: RobotApplicationConfig オブジェクトの配列

配列メンバー: 定数は 1 項目です。

simulationApplications

シミュレーションアプリケーションのリスト。

型: SimulationApplicationConfig オブジェクトの配列

配列メンバー: 定数は 1 項目です。

simulationTimeMillis

シミュレーションジョブの実行時間 (ミリ秒)。

型: 長整数

status

シミュレーションジョブのステータス。

型: 文字列

有効な値：`Pending | Preparing | Running | Restarting | Completed | Failed | RunningFailed | Terminating | Terminated | Canceled`

tags

指定のシミュレーションジョブに追加されたすべてのタグのリスト。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

vpcConfig

VPC の設定。

型: VPCConfigResponse オブジェクト

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

DescribeSimulationJobBatch

シミュレーションジョブバッチを記述します。

リクエストの構文

```
POST /describeSimulationJobBatch HTTP/1.1
Content-type: application/json

{
   "batch": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

batch

記述するバッチの ID。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "batchPolicy": {
      "maxConcurrency": number,
      "timeoutInSeconds": number
```

```
    },
    "clientRequestToken": "string",
    "createdAt": number,
    "createdRequests": [
       {
          "arn": "string",
          "computeType": "string",
          "dataSourceNames": [ "string" ],
          "lastUpdatedAt": number,
          "name": "string",
          "robotApplicationNames": [ "string" ],
          "simulationApplicationNames": [ "string" ],
          "status": "string"
       }
    ],
    "failedRequests": [
       {
          "failedAt": number,
          "failureCode": "string",
          "failureReason": "string",
          "request": {
             "compute": {
                "computeType": "string",
                "gpuUnitLimit": number,
                "simulationUnitLimit": number
             },
             "dataSources": [
                {
                   "destination": "string",
                   "name": "string",
                   "s3Bucket": "string",
                   "s3Keys": [ "string" ],
                   "type": "string"
                }
             ],
             "failureBehavior": "string",
             "iamRole": "string",
             "loggingConfig": {
                "recordAllRosTopics": boolean
             },
             "maxJobDurationInSeconds": number,
             "outputLocation": {
                "s3Bucket": "string",
                "s3Prefix": "string"
```

```
        },
        "robotApplications": [
           {
              "application": "string",
              "applicationVersion": "string",
              "launchConfig": {
                 "command": [ "string" ],
                 "environmentVariables": {
                    "string" : "string"
                 },
                 "launchFile": "string",
                 "packageName": "string",
                 "portForwardingConfig": {
                    "portMappings": [
                       {
                          "applicationPort": number,
                          "enableOnPublicIp": boolean,
                          "jobPort": number
                       }
                    ]
                 },
                 "streamUI": boolean
              },
              "tools": [
                 {
                    "command": "string",
                    "exitBehavior": "string",
                    "name": "string",
                    "streamOutputToCloudWatch": boolean,
                    "streamUI": boolean
                 }
              ],
              "uploadConfigurations": [
                 {
                    "name": "string",
                    "path": "string",
                    "uploadBehavior": "string"
                 }
              ],
              "useDefaultTools": boolean,
              "useDefaultUploadConfigurations": boolean
           }
        ],
        "simulationApplications": [
```

```
{
   "application": "string",
   "applicationVersion": "string",
   "launchConfig": {
      "command": [ "string" ],
      "environmentVariables": {
         "string" : "string"
      },
      "launchFile": "string",
      "packageName": "string",
      "portForwardingConfig": {
         "portMappings": [
            {
               "applicationPort": number,
               "enableOnPublicIp": boolean,
               "jobPort": number
            }
         ]
      },
      "streamUI": boolean
   },
   "tools": [
      {
         "command": "string",
         "exitBehavior": "string",
         "name": "string",
         "streamOutputToCloudWatch": boolean,
         "streamUI": boolean
      }
   ],
   "uploadConfigurations": [
      {
         "name": "string",
         "path": "string",
         "uploadBehavior": "string"
      }
   ],
   "useDefaultTools": boolean,
   "useDefaultUploadConfigurations": boolean,
   "worldConfigs": [
      {
         "world": "string"
      }
   ]
```

```
            }
         ],
         "tags": {
            "string" : "string"
         },
         "useDefaultApplications": boolean,
         "vpcConfig": {
            "assignPublicIp": boolean,
            "securityGroups": [ "string" ],
            "subnets": [ "string" ]
         }
      }
   }
],
"failureCode": "string",
"failureReason": "string",
"lastUpdatedAt": number,
"pendingRequests": [
   {
      "compute": {
         "computeType": "string",
         "gpuUnitLimit": number,
         "simulationUnitLimit": number
      },
      "dataSources": [
         {
            "destination": "string",
            "name": "string",
            "s3Bucket": "string",
            "s3Keys": [ "string" ],
            "type": "string"
         }
      ],
      "failureBehavior": "string",
      "iamRole": "string",
      "loggingConfig": {
         "recordAllRosTopics": boolean
      },
      "maxJobDurationInSeconds": number,
      "outputLocation": {
         "s3Bucket": "string",
         "s3Prefix": "string"
      },
      "robotApplications": [
```

```
            {
               "application": "string",
               "applicationVersion": "string",
               "launchConfig": {
                  "command": [ "string" ],
                  "environmentVariables": {
                     "string" : "string"
                  },
                  "launchFile": "string",
                  "packageName": "string",
                  "portForwardingConfig": {
                     "portMappings": [
                        {
                           "applicationPort": number,
                           "enableOnPublicIp": boolean,
                           "jobPort": number
                        }
                     ]
                  },
                  "streamUI": boolean
               },
               "tools": [
                  {
                     "command": "string",
                     "exitBehavior": "string",
                     "name": "string",
                     "streamOutputToCloudWatch": boolean,
                     "streamUI": boolean
                  }
               ],
               "uploadConfigurations": [
                  {
                     "name": "string",
                     "path": "string",
                     "uploadBehavior": "string"
                  }
               ],
               "useDefaultTools": boolean,
               "useDefaultUploadConfigurations": boolean
            }
         ],
         "simulationApplications": [
            {
               "application": "string",
```

```
            "applicationVersion": "string",
            "launchConfig": {
               "command": [ "string" ],
               "environmentVariables": {
                  "string" : "string"
               },
               "launchFile": "string",
               "packageName": "string",
               "portForwardingConfig": {
                  "portMappings": [
                     {
                        "applicationPort": number,
                        "enableOnPublicIp": boolean,
                        "jobPort": number
                     }
                  ]
               },
               "streamUI": boolean
            },
            "tools": [
               {
                  "command": "string",
                  "exitBehavior": "string",
                  "name": "string",
                  "streamOutputToCloudWatch": boolean,
                  "streamUI": boolean
               }
            ],
            "uploadConfigurations": [
               {
                  "name": "string",
                  "path": "string",
                  "uploadBehavior": "string"
               }
            ],
            "useDefaultTools": boolean,
            "useDefaultUploadConfigurations": boolean,
            "worldConfigs": [
               {
                  "world": "string"
               }
            ]
         }
      ],
```

```
        "tags": {
           "string" : "string"
        },
        "useDefaultApplications": boolean,
        "vpcConfig": {
           "assignPublicIp": boolean,
           "securityGroups": [ "string" ],
           "subnets": [ "string" ]
        }
     }
  ],
  "status": "string",
  "tags": {
     "string" : "string"
  }
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

バッチの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

batchPolicy

バッチポリシー。

タイプ：BatchPolicy オブジェクト

clientRequestToken

リクエストのべき等のために割り当てる一意の識別子 (大文字と小文字を区別)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 64 文字です。

パターン: `[a-zA-Z0-9_\-=]*`

createdAt

シミュレーションジョブバッチが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

createdRequests

作成されたシミュレーションジョブの概要のリスト。

タイプ：SimulationJobSummary オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 100 項目です。

failedRequests

失敗したシミュレーションジョブ作成リクエストのリスト。シミュレーションジョブに対してリクエストを作成できませんでした。失敗したリクエストにはシミュレーションジョブ ID がありません。

型: FailedCreateSimulationJobRequest オブジェクトの配列

failureCode

シミュレーションジョブバッチの失敗コード。

型: 文字列

有効な値：`InternalServiceError`

failureReason

シミュレーションジョブバッチが失敗した理由。

型: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: `.*`

lastUpdatedAt

シミュレーションジョブバッチが最後に更新されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

pendingRequests

保留中のシミュレーションジョブリクエストのリスト。これらのリクエストはシミュレーションジョブに対してまだ作成されていません。

型: SimulationJobRequest オブジェクトの配列

配列メンバー: 最小数は 1 項目です。最大数は 1000 項目です。

status

バッチのステータス。

保留中

シミュレーションジョブバッチリクエストが保留中です。

InProgress

シミュレーションジョブバッチが進行中です。

[失敗]

シミュレーションジョブバッチが失敗しました。内部障害 (InternalServiceError など) により、1 つまたは複数のシミュレーションジョブリクエストを完了できませんでした 詳細については、「failureCode」と「 failureReason」 を参照してください。

完了

シミュレーションバッチジョブが完了しました。バッチは、(1) バッチ内に保留中のシミュレーションジョブリクエストがなく、失敗したシミュレーションジョブリクエストがいずれも InternalServiceError を原因としない場合、および (2) 作成されたすべてのシミュレーションジョブが終了状態に達したとき (例えば Completed または Failed) に完了となります。

キャンセル

シミュレーションバッチジョブがキャンセルされました。

キャンセル中

シミュレーションバッチジョブをキャンセルしています。

完了中

シミュレーションバッチジョブを完了しています。

TimingOut

シミュレーションバッチジョブがタイムアウトしています。

バッチがタイムアウトし、内部障害 (`InternalServiceError` など) のために失敗していた保留中のリクエストがある場合の場合、バッチステータスは `Failed` になります。このような失敗リクエストがない場合、バッチステータスは `TimedOut` になります。

TimedOut

シミュレーションバッチジョブがタイムアウトしました。

型: 文字列

有効な値 : `Pending | InProgress | Failed | Completed | Canceled | Canceling | Completing | TimingOut | TimedOut`

tags

シミュレーションジョブバッチにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード : 500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

DescribeWorld

ワールドを記述します。

リクエストの構文

```
POST /describeWorld HTTP/1.1
Content-type: application/json

{
   "world": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

world

記述するワールドの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "createdAt": number,
```

```
    "generationJob": "string",
    "tags": {
       "string" : "string"
    },
    "template": "string",
    "worldDescriptionBody": "string"
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

ワールドの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

createdAt

ワールドが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

generationJob

そのワールドを生成したワールド生成ジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

tags

ワールドにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

template

ワールドテンプレート。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

worldDescriptionBody

ワールドの内容を記述する JSON 形式の文字列を返します。

型: 文字列

長さの制限：最小長は 1 です。最大長は 262,144 です。

パターン：`[\S\s]+`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

DescribeWorldExportJob

ワールドのエクスポートジョブを記述します

リクエストの構文

```
POST /describeWorldExportJob HTTP/1.1
Content-type: application/json

{
   "job": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

job

記述するワールドエクスポートジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "clientRequestToken": "string",
```

```
    "createdAt": number,
    "failureCode": "string",
    "failureReason": "string",
    "iamRole": "string",
    "outputLocation": {
       "s3Bucket": "string",
       "s3Prefix": "string"
    },
    "status": "string",
    "tags": {
       "string" : "string"
    },
    "worlds": [ "string" ]
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

ワールドエクスポートジョブの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

clientRequestToken

リクエストのべき等のために割り当てる一意の識別子 (大文字と小文字を区別)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 64 文字です。

パターン: `[a-zA-Z0-9_\-=]*`

createdAt

ワールドのエクスポートジョブが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

failureCode

ワールドエクスポートジョブが失敗した場合の失敗コード:

InternalServiceError

内部サービスエラー。

LimitExceeded

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

ResourceNotFound

指定したリソースが見つかりませんでした。

RequestThrottled

リクエストがスロットリングされました。

InvalidInput

リクエストの入力パラメータが無効です。

型: 文字列

有効な値 : `InternalServiceError | LimitExceeded | ResourceNotFound | RequestThrottled | InvalidInput | AccessDenied`

failureReason

ワールドエクスポートジョブが失敗した理由。

型: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: `.*`

iamRole

ワールドエクスポートプロセスで Amazon S3 バケットへのアクセスとエクスポートの実行に使用される IAM ロール。

型: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `arn:aws:iam::\w+:role/.*`

outputLocation

出力場所。

タイプ：OutputLocation オブジェクト

status

ワールドエクスポートジョブのステータス。

保留中

ワールドエクスポートジョブのリクエストが保留中です。

実行中

ワールドエクスポートジョブが実行中です。

完了

ワールドエクスポートジョブが完了しました。

[失敗]

ワールドエクスポートジョブが失敗しました。詳細については、「`failureCode`」と「`failureReason`」を参照してください。

キャンセル

ワールドエクスポートジョブがキャンセルされました。

キャンセル中

ワールドエクスポートジョブをキャンセルしています。

型: 文字列

有効な値：`Pending | Running | Completed | Failed | Canceling | Canceled`

tags

ワールドエクスポートジョブにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

worlds

エクスポートするワールドに対応する Amazon リソースネーム (ARN) のリスト。

タイプ：文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 100 項目です。

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン：`arn:.*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

DescribeWorldGenerationJob

ワールド生成ジョブを記述します。

リクエストの構文

```
POST /describeWorldGenerationJob HTTP/1.1
Content-type: application/json

{
   "job": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

job

記述するワールド生成ジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "clientRequestToken": "string",
```

```
    "createdAt": number,
    "failureCode": "string",
    "failureReason": "string",
    "finishedWorldsSummary": {
       "failureSummary": {
          "failures": [
             {
                "failureCode": "string",
                "failureCount": number,
                "sampleFailureReason": "string"
             }
          ],
          "totalFailureCount": number
       },
       "finishedCount": number,
       "succeededWorlds": [ "string" ]
    },
    "status": "string",
    "tags": {
       "string" : "string"
    },
    "template": "string",
    "worldCount": {
       "floorplanCount": number,
       "interiorCountPerFloorplan": number
    },
    "worldTags": {
       "string" : "string"
    }
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

ワールド生成ジョブの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

clientRequestToken

リクエストのべき等のために割り当てる一意の識別子 (大文字と小文字を区別)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 64 文字です。

パターン: `[a-zA-Z0-9_\-=]*`

createdAt

ワールドの生成ジョブが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

failureCode

ワールド生成ジョブが失敗した場合の失敗コード:

InternalServiceError

内部サービスエラー。

LimitExceeded

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

ResourceNotFound

指定したリソースが見つかりませんでした。

RequestThrottled

リクエストがスロットリングされました。

InvalidInput

リクエストの入力パラメータが無効です。

型: 文字列

有効な値：`InternalServiceError | LimitExceeded | ResourceNotFound | RequestThrottled | InvalidInput | AllWorldGenerationFailed`

failureReason

ワールド生成ジョブが失敗した理由。

型: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: .*

finishedWorldsSummary

完了したワールドに関する概要情報。

タイプ：FinishedWorldsSummary オブジェクト

status

ワールド生成ジョブのステータス:

保留中

ワールド生成ジョブのリクエストが保留中です。

実行中

ワールド生成ジョブが実行中です。

完了

ワールド生成ジョブが完了しました。

[失敗]

ワールド生成ジョブが失敗しました。詳細については、「failureCode」を参照してください。

PartialFailed

一部のワールドが生成されませんでした。

キャンセル

ワールド生成ジョブがキャンセルされました。

キャンセル中

ワールド生成ジョブをキャンセルしています。

型: 文字列

有効な値 : `Pending | Running | Completed | Failed | PartialFailed | Canceling | Canceled`

tags

ワールド生成ジョブにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

template

ワールドテンプレートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限 : 最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

worldCount

ワールドカウントに関する情報。

タイプ : WorldCount オブジェクト

worldTags

生成したワールドにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++

- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

DescribeWorldTemplate

ワールドテンプレートを記述します。

リクエストの構文

```
POST /describeWorldTemplate HTTP/1.1
Content-type: application/json

{
   "template": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

template

記述するワールドテンプレートの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "clientRequestToken": "string",
```

```
    "createdAt": number,
    "lastUpdatedAt": number,
    "name": "string",
    "tags": {
       "string" : "string"
    },
    "version": "string"
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

ワールドテンプレートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

clientRequestToken

リクエストのべき等のために割り当てる一意の識別子 (大文字と小文字を区別)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 64 文字です。

パターン: `[a-zA-Z0-9_\-=]*`

createdAt

ワールドテンプレートが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

lastUpdatedAt

ワールドテンプレートが最後に更新されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

name

ワールドテンプレートの名前。

型: 文字列

長さの制限: 最小長は 0 です。最大長は 255 です。

パターン: .*

tags

ワールドテンプレートにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: [a-zA-Z0-9 _.\-\/+=:]*

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: [a-zA-Z0-9 _.\-\/+=:]*

version

使用しているワールドテンプレートのバージョン。

型: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン：.*

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

GetWorldTemplateBody

ワールドテンプレートの本文を取得します。

リクエストの構文

```
POST /getWorldTemplateBody HTTP/1.1
Content-type: application/json

{
   "generationJob": "string",
   "template": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

generationJob

ワールドジェネレータージョブの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

template

ワールドテンプレートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "templateBody": "string"
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

templateBody

ワールドテンプレートの本文。

型: 文字列

長さの制限：最小長は 1 です。最大長は 262,144 です。

パターン：[\S\s]+

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

ListDeploymentJobs

このアクションは非推奨になりました。

> **Important**
>
> この API はサポートされなくなりました。詳細については、「サポートポリシー」ページの 2022 年 5 月 2 日の更新をご覧ください。

フリートのデプロイジョブのリストを返します。フィルターを指定して特定のデプロイジョブを取得することもできます。

リクエストの構文

```
POST /listDeploymentJobs HTTP/1.1
Content-type: application/json

{
   "filters": [
      {
         "name": "string",
         "values": [ "string" ]
      }
   ],
   "maxResults": number,
   "nextToken": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

filters

結果を制限するためのフィルター (オプション)。

サポートされているフィルター名は status と fleetName です。フィルタリングするときは、フィルターされた項目の完全な値を使用する必要があります。最大 3 つのフィルターを使用できますが、それらのフィルターは同じ名前の項目に対して使用する必要があります。例えば、ステータスが InProgress または Pending である項目を探す場合などです。

型: Filter オブジェクトの配列

配列メンバー: 定数は 1 項目です。

必須: いいえ

maxResults

このパラメータを使用すると、ListDeploymentJobs により maxResults 件の結果と nextToken レスポンス要素が 1 ページにまとめられます。最初のリクエストの残りの結果は、返された nextToken 値を含む別の ListDeploymentJobs リクエストを送信することで確認できます。この値は 1 ~ 200 の範囲で指定できます。このパラメータを使用しない場合は、ListDeploymentJobs により最大 200 件の結果と、該当する場合は nextToken 値が返されます。

タイプ: 整数

必須: いいえ

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListDeploymentJobs を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、 NextToken 前のレスポンスオブジェクトのパラメータはnullに設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

パターン: [a-zA-Z0-9_.\-\/+=]*

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
```

```
Content-type: application/json

{
   "deploymentJobs": [
      {
         "arn": "string",
         "createdAt": number,
         "deploymentApplicationConfigs": [
            {
               "application": "string",
               "applicationVersion": "string",
               "launchConfig": {
                  "environmentVariables": {
                     "string" : "string"
                  },
                  "launchFile": "string",
                  "packageName": "string",
                  "postLaunchFile": "string",
                  "preLaunchFile": "string"
               }
            }
         ],
         "deploymentConfig": {
            "concurrentDeploymentPercentage": number,
            "downloadConditionFile": {
               "bucket": "string",
               "etag": "string",
               "key": "string"
            },
            "failureThresholdPercentage": number,
            "robotDeploymentTimeoutInSeconds": number
         },
         "failureCode": "string",
         "failureReason": "string",
         "fleet": "string",
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

deploymentJobs

リクエストの条件を満たすデプロイジョブのリスト。

タイプ：DeploymentJob オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 200 項目です。

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListDeploymentJobs を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、 NextToken 前のレスポンスオブジェクトのパラメーターは null に設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

Pattern: [a-zA-Z0-9_.\-\/+=]*

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

ListFleets

このアクションは非推奨になりました。

> **Important**
>
> この API はサポートされなくなりました。詳細については、「サポートポリシー」ページの 2022 年 5 月 2 日の更新をご覧ください。

フリートのリストを返します。フィルターを指定して特定のフリートを取得することもできます。

リクエストの構文

```
POST /listFleets HTTP/1.1
Content-type: application/json

{
   "filters": [
      {
         "name": "string",
         "values": [ "string" ]
      }
   ],
   "maxResults": number,
   "nextToken": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

filters

結果を制限するためのフィルター (オプション)。

フィルター名 name はサポートされています。フィルタリングするときは、フィルターされた項目の完全な値を使用する必要があります。フィルターは 3 つまで使用できます。

型: Filter オブジェクトの配列

配列メンバー: 定数は 1 項目です。

必須: いいえ

maxResults

このパラメータを使用すると、ListFleets により maxResults 件の結果と nextToken レスポンス要素が 1 ページにまとめられます。最初のリクエストの残りの結果は、返された nextToken 値を含む別の ListFleets リクエストを送信することで確認できます。この値は 1～200 の範囲で指定できます。このパラメータを使用しない場合は、ListFleets により最大 200 件の結果と、該当する場合は nextToken 値が返されます。

タイプ: 整数

必須: いいえ

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListFleets を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、 NextToken 前のレスポンスオブジェクトのパラメータはnullに設定されます。

> **Note**
>
> このトークンは、リスト内の次の項目を取得するためだけに使用される不透明な識別子として扱われ、他のプログラム目的では使用されません。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

パターン: [a-zA-Z0-9_.\-\/+=]*

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
```

```
Content-type: application/json

{
   "fleetDetails": [
      {
         "arn": "string",
         "createdAt": number,
         "lastDeploymentJob": "string",
         "lastDeploymentStatus": "string",
         "lastDeploymentTime": number,
         "name": "string"
      }
   ],
   "nextToken": "string"
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

fleetDetails

リクエスト条件を満たしているフリートの詳細のリスト。

タイプ：Fleet オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 200 項目です。

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListFleets を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、 NextToken 前のレスポンスオブジェクトのパラメーターは null に設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

Pattern: [a-zA-Z0-9_.\-\/+=]*

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK

- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

ListRobotApplications

ロボットアプリケーションのリストを返します。フィルターを指定して特定のロボットアプリケーションを取得することもできます。

リクエストの構文

```
POST /listRobotApplications HTTP/1.1
Content-type: application/json

{
   "filters": [
      {
         "name": "string",
         "values": [ "string" ]
      }
   ],
   "maxResults": number,
   "nextToken": "string",
   "versionQualifier": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

filters

結果を制限するためのフィルター (オプション)。

フィルター名 name はサポートされています。フィルタリングするときは、フィルターされた項目の完全な値を使用する必要があります。フィルターは 3 つまで使用できます。

型: Filter オブジェクトの配列

配列メンバー: 定数は 1 項目です。

必須: いいえ

maxResults

このパラメータを使用すると、ListRobotApplications により maxResults 件の結果と nextToken レスポンス要素が 1 ページにまとめられます。最初のリクエストの残りの結果は、返された nextToken 値を含む別の ListRobotApplications リクエストを送信することで確認できます。この値は 1 ~ 100 の範囲で指定できます。このパラメータを使用しない場合は、ListRobotApplications により最大 100 件の結果と、該当する場合は nextToken 値が返されます。

タイプ: 整数

必須: いいえ

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListRobotApplications を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、 NextToken 前のレスポンスオブジェクトのパラメータはnullに設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

パターン: [a-zA-Z0-9_.\-\/+=]*

必須: いいえ

versionQualifier

ロボットアプリケーションのバージョン修飾子。

型: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: ALL

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
```

```
Content-type: application/json

{
   "nextToken": "string",
   "robotApplicationSummaries": [
      {
         "arn": "string",
         "lastUpdatedAt": number,
         "name": "string",
         "robotSoftwareSuite": {
            "name": "string",
            "version": "string"
         },
         "version": "string"
      }
   ]
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListRobotApplications を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、 NextToken 前のレスポンスオブジェクトのパラメーターは null に設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

Pattern: `[a-zA-Z0-9_.\-\/+=]*`

robotApplicationSummaries

リクエストの条件を満たしているロボットアプリケーション概要のリスト。

タイプ：RobotApplicationSummary オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 100 項目です。

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

ListRobots

このアクションは非推奨になりました。

> **Important**
>
> この API はサポートされなくなりました。詳細については、「サポートポリシー」ページの 2022 年 5 月 2 日の更新をご覧ください。

ロボットのリストを返します。フィルターを指定して特定のロボットを取得することもできます。

リクエストの構文

```
POST /listRobots HTTP/1.1
Content-type: application/json

{
   "filters": [
      {
         "name": "string",
         "values": [ "string" ]
      }
   ],
   "maxResults": number,
   "nextToken": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

filters

結果を制限するためのフィルター (オプション)。

サポートされているフィルター名は `status` と `fleetName` です。フィルタリングするときは、フィルターされた項目の完全な値を使用する必要があります。最大 3 つのフィルターを使用で

きますが、それらのフィルターは同じ名前の項目に対して使用する必要があります。例えば、ステータスが Registered または Available である項目を探す場合などです。

型: Filter オブジェクトの配列

配列メンバー: 定数は 1 項目です。

必須: いいえ

maxResults

このパラメータを使用すると、ListRobots により maxResults 件の結果と nextToken レスポンス要素が 1 ページにまとめられます。最初のリクエストの残りの結果は、返された nextToken 値を含む別の ListRobots リクエストを送信することで確認できます。この値は 1～200 の範囲で指定できます。このパラメータを使用しない場合は、ListRobots により最大 200 件の結果と、該当する場合は nextToken 値が返されます。

タイプ: 整数

必須: いいえ

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListRobots を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、 NextToken 前のレスポンスオブジェクトのパラメータはnullに設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

パターン: [a-zA-Z0-9_.\-\/+=]*

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json
```

```
{
   "nextToken": "string",
   "robots": [
      {
         "architecture": "string",
         "arn": "string",
         "createdAt": number,
         "fleetArn": "string",
         "greenGrassGroupId": "string",
         "lastDeploymentJob": "string",
         "lastDeploymentTime": number,
         "name": "string",
         "status": "string"
      }
   ]
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListRobots を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、 NextToken 前のレスポンスオブジェクトのパラメーターは null に設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

Pattern: [a-zA-Z0-9_.\-\/+=]*

robots

リクエストの条件を満たすロボットのリスト。

タイプ：Robot オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 1000 項目です。

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK

- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

ListSimulationApplications

シミュレーションアプリケーションのリストを返します。フィルターを指定して特定のシミュレーションアプリケーションを取得することもできます。

リクエストの構文

```
POST /listSimulationApplications HTTP/1.1
Content-type: application/json

{
   "filters": [
      {
         "name": "string",
         "values": [ "string" ]
      }
   ],
   "maxResults": number,
   "nextToken": "string",
   "versionQualifier": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

filters

結果を制限するためのフィルターのリスト (オプション)。

フィルター名 name はサポートされています。フィルタリングするときは、フィルターされた項目の完全な値を使用する必要があります。フィルターは 3 つまで使用できます。

型: Filter オブジェクトの配列

配列メンバー: 定数は 1 項目です。

必須: いいえ

maxResults

このパラメータを使用すると、ListSimulationApplications により maxResults 件の結果と nextToken レスポンス要素が 1 ページにまとめられます。最初のリクエストの残りの結果は、返された nextToken 値を含む別の ListSimulationApplications リクエストを送信することで確認できます。この値は 1 ~ 100 の範囲で指定できます。このパラメータを使用しない場合は、ListSimulationApplications により最大 100 件の結果と、該当する場合は nextToken 値が返されます。

タイプ: 整数

必須: いいえ

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListSimulationApplications を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、NextToken 前のレスポンスオブジェクトのパラメータはnullに設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

パターン: [a-zA-Z0-9_.\-\/+=]*

必須: いいえ

versionQualifier

シミュレーションアプリケーションのバージョン修飾子。

型: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: ALL

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
```

```
Content-type: application/json

{
   "nextToken": "string",
   "simulationApplicationSummaries": [
      {
         "arn": "string",
         "lastUpdatedAt": number,
         "name": "string",
         "robotSoftwareSuite": {
            "name": "string",
            "version": "string"
         },
         "simulationSoftwareSuite": {
            "name": "string",
            "version": "string"
         },
         "version": "string"
      }
   ]
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListSimulationApplications を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、NextToken 前のレスポンスオブジェクトのパラメーターは null に設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

Pattern: `[a-zA-Z0-9_.\-\/+=]*`

simulationApplicationSummaries

リクエストの条件を満たしているシミュレーションアプリケーション概要のリスト。

タイプ：SimulationApplicationSummary オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 100 項目です。

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK

- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

ListSimulationJobBatches

シミュレーションジョブバッチのリストを返します。フィルターを指定して特定のシミュレーションバッチジョブを取得することもできます。

リクエストの構文

```
POST /listSimulationJobBatches HTTP/1.1
Content-type: application/json

{
   "filters": [
      {
         "name": "string",
         "values": [ "string" ]
      }
   ],
   "maxResults": number,
   "nextToken": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

filters

結果を制限するためのフィルター (オプション)。

型: Filter オブジェクトの配列

配列メンバー: 定数は 1 項目です。

必須: いいえ

maxResults

このパラメータを使用すると、ListSimulationJobBatches により maxResults 件の結果と nextToken レスポンス要素が 1 ページにまとめられます。最初のリクエストの残りの結果は、

返された nextToken 値を含む別の ListSimulationJobBatches リクエストを送信することで確認できます。

タイプ: 整数

必須: いいえ

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListSimulationJobBatches を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、NextToken 前のレスポンスオブジェクトのパラメータはnullに設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

パターン: [a-zA-Z0-9_.\-\/+=]*

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "nextToken": "string",
   "simulationJobBatchSummaries": [
      {
         "arn": "string",
         "createdAt": number,
         "createdRequestCount": number,
         "failedRequestCount": number,
         "lastUpdatedAt": number,
         "pendingRequestCount": number,
         "status": "string"
      }
   ]
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListSimulationJobBatches を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、NextToken 前のレスポンスオブジェクトのパラメーターは null に設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

Pattern: [a-zA-Z0-9_.\-\/+=]*

simulationJobBatchSummaries

シミュレーションジョブバッチの概要のリスト。

型: SimulationJobBatchSummary オブジェクトの配列

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

ListSimulationJobs

シミュレーションジョブのリストを返します。フィルターを指定して特定のシミュレーションジョブを取得することもできます。

リクエストの構文

```
POST /listSimulationJobs HTTP/1.1
Content-type: application/json

{
   "filters": [
      {
         "name": "string",
         "values": [ "string" ]
      }
   ],
   "maxResults": number,
   "nextToken": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

filters

結果を制限するためのフィルター (オプション)。

フィルタ名 `status` と `simulationApplicationName` と`robotApplicationName` がサポートされています。フィルタリングするときは、フィルターされた項目の完全な値を使用する必要があります。最大 3 つのフィルターを使用できますが、それらのフィルターは同じ名前の項目に対して使用する必要があります。例えば、ステータスが `Preparing` または `Running` である項目を探す場合などです。

型: Filter オブジェクトの配列

配列メンバー: 定数は 1 項目です。

必須: いいえ

maxResults

このパラメータを使用すると、ListSimulationJobs により maxResults 件の結果と nextToken レスポンス要素が 1 ページにまとめられます。最初のリクエストの残りの結果は、返された nextToken 値を含む別の ListSimulationJobs リクエストを送信することで確認できます。この値は 1 ~ 1000 の範囲で指定できます。このパラメータを使用しない場合は、ListSimulationJobs により最大 1000 件の結果と、該当する場合は nextToken 値が返されます。

タイプ: 整数

必須: いいえ

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListSimulationJobs を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、 NextToken 前のレスポンスオブジェクトのパラメータはnullに設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

パターン: [a-zA-Z0-9_.\-\/+=]*

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "nextToken": "string",
   "simulationJobSummaries": [
      {
         "arn": "string",
```

```
            "computeType": "string",
            "dataSourceNames": [ "string" ],
            "lastUpdatedAt": number,
            "name": "string",
            "robotApplicationNames": [ "string" ],
            "simulationApplicationNames": [ "string" ],
            "status": "string"
        }
    ]
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListSimulationJobs を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、 NextToken 前のレスポンスオブジェクトのパラメーターは null に設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

Pattern: [a-zA-Z0-9_.\-\/+=]*

simulationJobSummaries

リクエストの条件を満たしているシミュレーションジョブ概要のリスト。

タイプ：SimulationJobSummary オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 100 項目です。

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

ListTagsForResource

AWS RoboMaker リソースのすべてのタグを一覧表示します。

リクエストの構文

```
GET /tags/resourceArn HTTP/1.1
```

URI リクエストパラメータ

リクエストでは、次の URI パラメータを使用します。

resourceArn

リストされるタグ付きの AWS RoboMaker Amazon リソースネーム (ARN)。

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: はい

リクエストボディ

リクエストにリクエスト本文がありません。

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "tags": {
      "string" : "string"
   }
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

tags

指定のリソースに追加されたすべてのタグのリスト。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

ListWorldExportJobs

ワールドエクスポートジョブを一覧表示します。

リクエストの構文

```
POST /listWorldExportJobs HTTP/1.1
Content-type: application/json

{
   "filters": [
      {
         "name": "string",
         "values": [ "string" ]
      }
   ],
   "maxResults": number,
   "nextToken": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

filters

結果を制限するためのフィルター (オプション)。generationJobId と templateId を使用できます。

型: Filter オブジェクトの配列

配列メンバー: 定数は 1 項目です。

必須: いいえ

maxResults

このパラメータを使用すると、ListWorldExportJobs により maxResults 件の結果と nextToken レスポンス要素が 1 ページにまとめられます。最初のリクエストの残りの結果

は、返された nextToken 値を含む別の ListWorldExportJobs リクエストを送信することで確認できます。この値は 1 ~ 100 の範囲で指定できます。このパラメータを使用しない場合は、ListWorldExportJobs により最大 100 件の結果と、該当する場合は nextToken 値が返されます。

タイプ: 整数

必須: いいえ

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListWorldExportJobs を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、 NextToken 前のレスポンスオブジェクトのパラメータはnullに設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

パターン: [a-zA-Z0-9_.\-\/+=]*

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "nextToken": "string",
   "worldExportJobSummaries": [
      {
         "arn": "string",
         "createdAt": number,
         "outputLocation": {
            "s3Bucket": "string",
            "s3Prefix": "string"
         },
         "status": "string",
         "worlds": [ "string" ]
```

```
        }
    ]
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListWorldExportJobsRequest を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、NextToken 前のレスポンスオブジェクトのパラメーターは null に設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

Pattern: [a-zA-Z0-9_.\-\/+=]*

worldExportJobSummaries

ワールドエクスポートジョブの概要情報。

タイプ：WorldExportJobSummary オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 100 項目です。

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

ListWorldGenerationJobs

ワールドジェネレータージョブを一覧表示します。

リクエストの構文

```
POST /listWorldGenerationJobs HTTP/1.1
Content-type: application/json

{
   "filters": [
      {
         "name": "string",
         "values": [ "string" ]
      }
   ],
   "maxResults": number,
   "nextToken": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

filters

結果を制限するためのフィルター (オプション)。`status` と `templateId` を使用できます。

型: Filter オブジェクトの配列

配列メンバー: 定数は 1 項目です。

必須: いいえ

maxResults

このパラメータを使用すると、`ListWorldGeneratorJobs` により `maxResults` 件の結果と `nextToken` レスポンス要素が 1 ページにまとめられます。最初のリクエストの残りの結果は、返された `nextToken` 値を含む別の `ListWorldGeneratorJobs` リクエストを送信すること

で確認できます。この値は 1 ~ 100 の範囲で指定できます。このパラメータを使用しない場合は、ListWorldGeneratorJobs により最大 100 件の結果と、該当する場合は nextToken 値が返されます。

タイプ: 整数

必須: いいえ

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListWorldGenerationJobsRequest を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、NextToken 前のレスポンスオブジェクトのパラメータはnullに設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

パターン: [a-zA-Z0-9_.\-\/+=]*

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "nextToken": "string",
   "worldGenerationJobSummaries": [
      {
         "arn": "string",
         "createdAt": number,
         "failedWorldCount": number,
         "status": "string",
         "succeededWorldCount": number,
         "template": "string",
         "worldCount": {
            "floorplanCount": number,
            "interiorCountPerFloorplan": number
```

```
            }
        }
    ]
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListWorldGeneratorJobsRequest を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、NextToken 前のレスポンスオブジェクトのパラメーターは null に設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

Pattern: [a-zA-Z0-9_.\-\/+=]*

worldGenerationJobSummaries

ワールドジェネレータージョブの概要情報。

タイプ：WorldGenerationJobSummary オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 100 項目です。

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

ListWorlds

ワールドを一覧表示します。

リクエストの構文

```
POST /listWorlds HTTP/1.1
Content-type: application/json

{
   "filters": [
      {
         "name": "string",
         "values": [ "string" ]
      }
   ],
   "maxResults": number,
   "nextToken": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

filters

結果を制限するためのフィルター (オプション)。`status` を使用できます。

型: Filter オブジェクトの配列

配列メンバー: 定数は 1 項目です。

必須: いいえ

maxResults

このパラメータを使用すると、`ListWorlds` により `maxResults` 件の結果と `nextToken` レスポンス要素が 1 ページにまとめられます。最初のリクエストの残りの結果は、返された `nextToken` 値を含む別の `ListWorlds` リクエストを送信することで確認できます。この値は

1 ~ 100 の範囲で指定できます。このパラメータを使用しない場合は、ListWorlds により最大 100 件の結果と、該当する場合は nextToken 値が返されます。

タイプ: 整数

必須: いいえ

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListWorlds を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、 NextToken 前のレスポンスオブジェクトのパラメータはnullに設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

パターン: [a-zA-Z0-9_.\-\/+=]*

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "nextToken": "string",
   "worldSummaries": [
      {
         "arn": "string",
         "createdAt": number,
         "generationJob": "string",
         "template": "string"
      }
   ]
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListWorlds を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、 NextToken 前のレスポンスオブジェクトのパラメーターは null に設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

Pattern: [a-zA-Z0-9_.\-\/+=]*

worldSummaries

ワールドの概要情報。

型: WorldSummary オブジェクトの配列

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

ListWorldTemplates

ワールドテンプレートを一覧表示します。

リクエストの構文

```
POST /listWorldTemplates HTTP/1.1
Content-type: application/json

{
   "maxResults": number,
   "nextToken": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

maxResults

このパラメータを使用すると、ListWorldTemplates により maxResults 件の結果と nextToken レスポンス要素が 1 ページにまとめられます。最初のリクエストの残りの結果は、返された nextToken 値を含む別の ListWorldTemplates リクエストを送信することで確認できます。この値は 1 ~ 100 の範囲で指定できます。このパラメータを使用しない場合は、ListWorldTemplates により最大 100 件の結果と、該当する場合は nextToken 値が返されます。

タイプ: 整数

必須: いいえ

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListWorldTemplates を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、 NextToken 前のレスポンスオブジェクトのパラメータはnullに設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

パターン: [a-zA-Z0-9_.\-\/+=]*

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "nextToken": "string",
   "templateSummaries": [
      {
         "arn": "string",
         "createdAt": number,
         "lastUpdatedAt": number,
         "name": "string",
         "version": "string"
      }
   ]
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

nextToken

前のページ分割されたリクエストにより残りの結果のすべてが返されなかった場合、レスポンスオブジェクトの nextToken パラメータ値がトークンに設定されます。次の一連の結果を取得するには、もう一度 ListWorldTemplates を呼び出し、そのトークンをリクエストオブジェクトの nextToken パラメータに割り当ててください。結果が残っていない場合、 NextToken 前のレスポンスオブジェクトのパラメーターは null に設定されます。

型: 文字列

長さの制限：最小長は 1 です。最大長は 2,048 です。

Pattern: `[a-zA-Z0-9_.\-\/+=]*`

templateSummaries

テンプレートの概要情報。

型: TemplateSummary オブジェクトの配列

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK

- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

RegisterRobot

このアクションは非推奨になりました。

ロボットをフリートに登録します。

> **Important**
>
> この API はサポートされなくなったため、使用した場合はエラーがスローされます。詳細については、「サポートポリシー」ページの 2022 年 1 月 31 日の更新情報を参照してください。

リクエストの構文

```
POST /registerRobot HTTP/1.1
Content-type: application/json

{
   "fleet": "string",
   "robot": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

fleet

フリートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

robot

ロボットの Amazon リソースネーム (ARN)

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "fleet": "string",
   "robot": "string"
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

fleet

ロボットが加わるフリートの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

robot

ロボット登録に関する情報。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン：`arn:.*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

LimitExceededException

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

RestartSimulationJob

実行中のシミュレーションジョブを再開します。

リクエストの構文

```
POST /restartSimulationJob HTTP/1.1
Content-type: application/json

{
   "job": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

job

シミュレーションジョブの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
```

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

LimitExceededException

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++

- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

StartSimulationJobBatch

新しいシミュレーションジョブバッチを開始します。バッチは 1 つまたは複数の SimulationJobRequest オブジェクトを使用して定義されます。

リクエストの構文

```
POST /startSimulationJobBatch HTTP/1.1
Content-type: application/json

{
   "batchPolicy": {
      "maxConcurrency": number,
      "timeoutInSeconds": number
   },
   "clientRequestToken": "string",
   "createSimulationJobRequests": [
      {
         "compute": {
            "computeType": "string",
            "gpuUnitLimit": number,
            "simulationUnitLimit": number
         },
         "dataSources": [
            {
               "destination": "string",
               "name": "string",
               "s3Bucket": "string",
               "s3Keys": [ "string" ],
               "type": "string"
            }
         ],
         "failureBehavior": "string",
         "iamRole": "string",
         "loggingConfig": {
            "recordAllRosTopics": boolean
         },
         "maxJobDurationInSeconds": number,
         "outputLocation": {
            "s3Bucket": "string",
            "s3Prefix": "string"
         },
         "robotApplications": [
```

```
            {
               "application": "string",
               "applicationVersion": "string",
               "launchConfig": {
                  "command": [ "string" ],
                  "environmentVariables": {
                     "string" : "string"
                  },
                  "launchFile": "string",
                  "packageName": "string",
                  "portForwardingConfig": {
                     "portMappings": [
                        {
                           "applicationPort": number,
                           "enableOnPublicIp": boolean,
                           "jobPort": number
                        }
                     ]
                  },
                  "streamUI": boolean
               },
               "tools": [
                  {
                     "command": "string",
                     "exitBehavior": "string",
                     "name": "string",
                     "streamOutputToCloudWatch": boolean,
                     "streamUI": boolean
                  }
               ],
               "uploadConfigurations": [
                  {
                     "name": "string",
                     "path": "string",
                     "uploadBehavior": "string"
                  }
               ],
               "useDefaultTools": boolean,
               "useDefaultUploadConfigurations": boolean
            }
         ],
         "simulationApplications": [
            {
               "application": "string",
```

```
            "applicationVersion": "string",
            "launchConfig": {
               "command": [ "string" ],
               "environmentVariables": {
                  "string" : "string"
               },
               "launchFile": "string",
               "packageName": "string",
               "portForwardingConfig": {
                  "portMappings": [
                     {
                        "applicationPort": number,
                        "enableOnPublicIp": boolean,
                        "jobPort": number
                     }
                  ]
               },
               "streamUI": boolean
            },
            "tools": [
               {
                  "command": "string",
                  "exitBehavior": "string",
                  "name": "string",
                  "streamOutputToCloudWatch": boolean,
                  "streamUI": boolean
               }
            ],
            "uploadConfigurations": [
               {
                  "name": "string",
                  "path": "string",
                  "uploadBehavior": "string"
               }
            ],
            "useDefaultTools": boolean,
            "useDefaultUploadConfigurations": boolean,
            "worldConfigs": [
               {
                  "world": "string"
               }
            ]
         }
      ],
```

```
            "tags": {
               "string" : "string"
            },
            "useDefaultApplications": boolean,
            "vpcConfig": {
               "assignPublicIp": boolean,
               "securityGroups": [ "string" ],
               "subnets": [ "string" ]
            }
         }
      ],
      "tags": {
         "string" : "string"
      }
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

batchPolicy

バッチポリシー。

タイプ : BatchPolicy オブジェクト

必須: いいえ

clientRequestToken

リクエストのべき等のために割り当てる一意の識別子 (大文字と小文字を区別)。

型: 文字列

長さの制限 : 最小長は 1 です。最大長は 64 文字です。

パターン: `[a-zA-Z0-9_\-=]*`

必須: いいえ

createSimulationJobRequests

バッチで作成するシミュレーションジョブリクエストのリスト。

型: SimulationJobRequest オブジェクトの配列

配列メンバー: 最小数は 1 項目です。最大数は 1000 項目です。

必須: はい

tags

デプロイジョブバッチにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "batchPolicy": {
      "maxConcurrency": number,
      "timeoutInSeconds": number
   },
   "clientRequestToken": "string",
   "createdAt": number,
   "createdRequests": [
      {
         "arn": "string",
         "computeType": "string",
```

```
        "dataSourceNames": [ "string" ],
        "lastUpdatedAt": number,
        "name": "string",
        "robotApplicationNames": [ "string" ],
        "simulationApplicationNames": [ "string" ],
        "status": "string"
      }
   ],
   "failedRequests": [
      {
        "failedAt": number,
        "failureCode": "string",
        "failureReason": "string",
        "request": {
          "compute": {
            "computeType": "string",
            "gpuUnitLimit": number,
            "simulationUnitLimit": number
          },
          "dataSources": [
            {
              "destination": "string",
              "name": "string",
              "s3Bucket": "string",
              "s3Keys": [ "string" ],
              "type": "string"
            }
          ],
          "failureBehavior": "string",
          "iamRole": "string",
          "loggingConfig": {
            "recordAllRosTopics": boolean
          },
          "maxJobDurationInSeconds": number,
          "outputLocation": {
            "s3Bucket": "string",
            "s3Prefix": "string"
          },
          "robotApplications": [
            {
              "application": "string",
              "applicationVersion": "string",
              "launchConfig": {
                "command": [ "string" ],
```

```
                "environmentVariables": {
                   "string" : "string"
                },
                "launchFile": "string",
                "packageName": "string",
                "portForwardingConfig": {
                   "portMappings": [
                      {
                         "applicationPort": number,
                         "enableOnPublicIp": boolean,
                         "jobPort": number
                      }
                   ]
                },
                "streamUI": boolean
             },
             "tools": [
                {
                   "command": "string",
                   "exitBehavior": "string",
                   "name": "string",
                   "streamOutputToCloudWatch": boolean,
                   "streamUI": boolean
                }
             ],
             "uploadConfigurations": [
                {
                   "name": "string",
                   "path": "string",
                   "uploadBehavior": "string"
                }
             ],
             "useDefaultTools": boolean,
             "useDefaultUploadConfigurations": boolean
          }
       ],
       "simulationApplications": [
          {
             "application": "string",
             "applicationVersion": "string",
             "launchConfig": {
                "command": [ "string" ],
                "environmentVariables": {
                   "string" : "string"
```

```
            },
            "launchFile": "string",
            "packageName": "string",
            "portForwardingConfig": {
               "portMappings": [
                  {
                     "applicationPort": number,
                     "enableOnPublicIp": boolean,
                     "jobPort": number
                  }
               ]
            },
            "streamUI": boolean
         },
         "tools": [
            {
               "command": "string",
               "exitBehavior": "string",
               "name": "string",
               "streamOutputToCloudWatch": boolean,
               "streamUI": boolean
            }
         ],
         "uploadConfigurations": [
            {
               "name": "string",
               "path": "string",
               "uploadBehavior": "string"
            }
         ],
         "useDefaultTools": boolean,
         "useDefaultUploadConfigurations": boolean,
         "worldConfigs": [
            {
               "world": "string"
            }
         ]
      }
   ],
   "tags": {
      "string" : "string"
   },
   "useDefaultApplications": boolean,
   "vpcConfig": {
```

```
            "assignPublicIp": boolean,
            "securityGroups": [ "string" ],
            "subnets": [ "string" ]
         }
      }
   }
],
"failureCode": "string",
"failureReason": "string",
"pendingRequests": [
   {
      "compute": {
         "computeType": "string",
         "gpuUnitLimit": number,
         "simulationUnitLimit": number
      },
      "dataSources": [
         {
            "destination": "string",
            "name": "string",
            "s3Bucket": "string",
            "s3Keys": [ "string" ],
            "type": "string"
         }
      ],
      "failureBehavior": "string",
      "iamRole": "string",
      "loggingConfig": {
         "recordAllRosTopics": boolean
      },
      "maxJobDurationInSeconds": number,
      "outputLocation": {
         "s3Bucket": "string",
         "s3Prefix": "string"
      },
      "robotApplications": [
         {
            "application": "string",
            "applicationVersion": "string",
            "launchConfig": {
               "command": [ "string" ],
               "environmentVariables": {
                  "string" : "string"
               },
```

```
            "launchFile": "string",
            "packageName": "string",
            "portForwardingConfig": {
               "portMappings": [
                  {
                     "applicationPort": number,
                     "enableOnPublicIp": boolean,
                     "jobPort": number
                  }
               ]
            },
            "streamUI": boolean
         },
         "tools": [
            {
               "command": "string",
               "exitBehavior": "string",
               "name": "string",
               "streamOutputToCloudWatch": boolean,
               "streamUI": boolean
            }
         ],
         "uploadConfigurations": [
            {
               "name": "string",
               "path": "string",
               "uploadBehavior": "string"
            }
         ],
         "useDefaultTools": boolean,
         "useDefaultUploadConfigurations": boolean
      }
   ],
   "simulationApplications": [
      {
         "application": "string",
         "applicationVersion": "string",
         "launchConfig": {
            "command": [ "string" ],
            "environmentVariables": {
               "string" : "string"
            },
            "launchFile": "string",
            "packageName": "string",
```

```
            "portForwardingConfig": {
               "portMappings": [
                  {
                     "applicationPort": number,
                     "enableOnPublicIp": boolean,
                     "jobPort": number
                  }
               ]
            },
            "streamUI": boolean
         },
         "tools": [
            {
               "command": "string",
               "exitBehavior": "string",
               "name": "string",
               "streamOutputToCloudWatch": boolean,
               "streamUI": boolean
            }
         ],
         "uploadConfigurations": [
            {
               "name": "string",
               "path": "string",
               "uploadBehavior": "string"
            }
         ],
         "useDefaultTools": boolean,
         "useDefaultUploadConfigurations": boolean,
         "worldConfigs": [
            {
               "world": "string"
            }
         ]
      }
   ],
   "tags": {
      "string" : "string"
   },
   "useDefaultApplications": boolean,
   "vpcConfig": {
      "assignPublicIp": boolean,
      "securityGroups": [ "string" ],
      "subnets": [ "string" ]
```

```
        }
      }
   ],
   "status": "string",
   "tags": {
      "string" : "string"
   }
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

バッチの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

batchPolicy

バッチポリシー。

タイプ：BatchPolicy オブジェクト

clientRequestToken

リクエストのべき等のために割り当てる一意の識別子 (大文字と小文字を区別)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 64 文字です。

パターン: `[a-zA-Z0-9_\-=]*`

createdAt

シミュレーションジョブバッチが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

createdRequests

作成されたシミュレーションジョブリクエスト概要のリスト。

タイプ：SimulationJobSummary オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 100 項目です。

failedRequests

失敗したシミュレーションジョブリクエストのリスト。シミュレーションジョブに対してリクエストを作成できませんでした。失敗したリクエストにはシミュレーションジョブ ID がありません。

型: FailedCreateSimulationJobRequest オブジェクトの配列

failureCode

シミュレーションジョブバッチが失敗した場合の失敗コード。

型: 文字列

有効な値：`InternalServiceError`

failureReason

シミュレーションジョブバッチが失敗した理由。

型: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: `.*`

pendingRequests

保留中のシミュレーションジョブリクエストのリスト。これらのリクエストはシミュレーションジョブに対してまだ作成されていません。

型: SimulationJobRequest オブジェクトの配列

配列メンバー: 最小数は 1 項目です。最大数は 1000 項目です。

status

シミュレーションジョブバッチのステータス。

保留中

シミュレーションジョブバッチリクエストが保留中です。

InProgress

シミュレーションジョブバッチが進行中です。

[失敗]

シミュレーションジョブバッチが失敗しました。内部障害 (InternalServiceError など) により、1 つまたは複数のシミュレーションジョブリクエストを完了できませんでした 詳細については、「failureCode」と「 failureReason」 を参照してください。

完了

シミュレーションバッチジョブが完了しました。バッチは、(1) バッチ内に保留中のシミュレーションジョブリクエストがなく、失敗したシミュレーションジョブリクエストがいずれも InternalServiceError を原因としない場合、および (2) 作成されたすべてのシミュレーションジョブが終了状態に達したとき (例えば Completed または Failed) に完了となります。

キャンセル

シミュレーションバッチジョブがキャンセルされました。

キャンセル中

シミュレーションバッチジョブをキャンセルしています。

完了中

シミュレーションバッチジョブを完了しています。

TimingOut

シミュレーションバッチジョブがタイムアウトしています。

バッチがタイムアウトし、内部障害 (InternalServiceError など) のために失敗していた保留中のリクエストがある場合の場合、バッチステータスは Failed になります。このような失敗リクエストがない場合、バッチステータスは TimedOut になります。

TimedOut

シミュレーションバッチジョブがタイムアウトしました。

型: 文字列

有効な値 : `Pending | InProgress | Failed | Completed | Canceled | Canceling | Completing | TimingOut | TimedOut`

tags

デプロイジョブバッチにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

IdempotentParameterMismatchException

リクエストは、前の、しかし同一ではないリクエストと同じクライアントトークンを使用します。リクエストが同一でない限り、異なるリクエストでクライアントトークンを再利用しないでください。

HTTP ステータスコード : 400

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード : 500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード : 400

LimitExceededException

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

SyncDeploymentJob

このアクションは非推奨になりました。

> **Important**
>
> この API はサポートされなくなりました。詳細については、「サポートポリシー」ページの 2022 年 5 月 2 日の更新をご覧ください。

フリート内のロボットを最新のデプロイに同期させます。これは、デプロイ後にロボットが追加された場合に便利です。

リクエストの構文

```
POST /syncDeploymentJob HTTP/1.1
Content-type: application/json

{
   "clientRequestToken": "string",
   "fleet": "string"
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

clientRequestToken

リクエストのべき等のために割り当てる一意の識別子 (大文字と小文字を区別)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 64 文字です。

パターン: `[a-zA-Z0-9_\-=]*`

必須：はい

fleet

同期のターゲットフリート。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "createdAt": number,
   "deploymentApplicationConfigs": [
      {
         "application": "string",
         "applicationVersion": "string",
         "launchConfig": {
            "environmentVariables": {
               "string" : "string"
            },
            "launchFile": "string",
            "packageName": "string",
            "postLaunchFile": "string",
            "preLaunchFile": "string"
         }
      }
   ],
   "deploymentConfig": {
      "concurrentDeploymentPercentage": number,
      "downloadConditionFile": {
         "bucket": "string",
         "etag": "string",
         "key": "string"
      },
      "failureThresholdPercentage": number,
```

```
        "robotDeploymentTimeoutInSeconds": number
    },
    "failureCode": "string",
    "failureReason": "string",
    "fleet": "string",
    "status": "string"
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

同期リクエストの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

createdAt

フリートが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

deploymentApplicationConfigs

デプロイアプリケーション設定に関する情報。

型: DeploymentApplicationConfig オブジェクトの配列

配列メンバー: 定数は 1 項目です。

deploymentConfig

デプロイ設定に関する情報。

タイプ：DeploymentConfig オブジェクト

failureCode

ジョブが失敗した場合の失敗コード:

InternalServiceError

内部サービスエラー。

RobotApplicationCrash

ロボットアプリケーションが異常終了しました。

SimulationApplicationCrash

シミュレーションアプリケーションが異常終了しました。

BadPermissionsRobotApplication

ロボットアプリケーションバンドルをダウンロードできませんでした。

BadPermissionsSimulationApplication

シミュレーションアプリケーションバンドルをダウンロードできませんでした。

BadPermissionsS3 アウトプット

お客様が用意した S3 バケットに出力を発行できません。

BadPermissionsCloudwatchLogs

CloudWatch お客様が提供した Logs リソースにログを公開できません。

SubnetIpLimitExceeded

サブネット IP 限界を超えました

ENI LimitExceeded

ENI 限界を超えました。

BadPermissionsUserCredentials

提供されたロールを使用できません。

InvalidBundleRobotApplication

ロボットバンドルを抽出できません (無効な形式、バンドルエラー、またはその他の問題)。

InvalidBundleSimulationApplication

シミュレーションバンドルを抽出できません (無効な形式、バンドルエラー、またはその他の問題)。

RobotApplicationVersionMismatchedEtag

バージョン作成中に Etag RobotApplication の値が一致しない。

SimulationApplicationVersionMismatchedEtag

Etag for SimulationApplication はバージョン作成中の値と一致しません。

型: 文字列

有効な値 : `ResourceNotFound | EnvironmentSetupError | EtagMismatch | FailureThresholdBreached | RobotDeploymentAborted | RobotDeploymentNoResponse | RobotAgentConnectionTimeout | GreengrassDeploymentFailed | InvalidGreengrassGroup | MissingRobotArchitecture | MissingRobotApplicationArchitecture | MissingRobotDeploymentResource | GreengrassGroupVersionDoesNotExist | LambdaDeleted | ExtractingBundleFailure | PreLaunchFileFailure | PostLaunchFileFailure | BadPermissionError | DownloadConditionFailed | BadLambdaAssociated | InternalServerError | RobotApplicationDoesNotExist | DeploymentFleetDoesNotExist | FleetDeploymentTimeout`

failureReason

ジョブが失敗した場合の失敗の理由。

型: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: `.*`

fleet

フリートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限 : 最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

status

同期ジョブのステータス。

型: 文字列

有効な値：`Pending | Preparing | InProgress | Failed | Succeeded | Canceled`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

ConcurrentDeploymentException

障害割合のしきい値割合が満たされました。

HTTP ステータスコード：400

IdempotentParameterMismatchException

リクエストは、前の、しかし同一ではないリクエストと同じクライアントトークンを使用します。リクエストが同一でない限り、異なるリクエストでクライアントトークンを再利用しないでください。

HTTP ステータスコード：400

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

LimitExceededException

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

TagResource

AWS RoboMaker リソースのタグを追加または編集します。

各タグはタグキーとタグ値で構成されています。タグキーとタグ値の両方が必要ですが、タグ値は空の文字列にすることができます。

タグキーとタグ値に適用されるルールの詳細については、『 AWS 請求とコスト管理ユーザーガイド』の「ユーザー定義タグの制限」を参照してください。

リクエストの構文

```
POST /tags/resourceArn HTTP/1.1
Content-type: application/json

{
   "tags": {
      "string" : "string"
   }
}
```

URI リクエストパラメータ

リクエストでは、次の URI パラメータを使用します。

resourceArn

タグ付けしている AWS リソースの Amazon RoboMaker リソースネーム (ARN)。

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: はい

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

tags

リソースにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

必須: はい

レスポンスの構文

```
HTTP/1.1 200
```

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

UntagResource

指定された AWS RoboMaker リソースから指定されたタグを削除します。

タグを削除するには、タグキーを指定します。既存のタグキーのタグ値を変更するには、TagResource を使用します。

リクエストの構文

```
DELETE /tags/resourceArn?tagKeys=tagKeys HTTP/1.1
```

URI リクエストパラメータ

リクエストでは、次の URI パラメータを使用します。

resourceArn

タグを削除する AWS リソースの Amazon RoboMaker リソースネーム (ARN)。

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

tagKeys

リソースへのアタッチが解除されるタグキーとタグ値を含むマップ。

長さの制限：最小長は 1 です。最大長は 128 です。

パターン: `[a-zA-Z0-9 _.\-\/+=:]*`

必須: はい

リクエストボディ

リクエストにリクエスト本文がありません。

レスポンスの構文

```
HTTP/1.1 200
```

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK

- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

UpdateRobotApplication

ロボットアプリケーションを更新します。

リクエストの構文

```
POST /updateRobotApplication HTTP/1.1
Content-type: application/json

{
   "application": "string",
   "currentRevisionId": "string",
   "environment": {
      "uri": "string"
   },
   "robotSoftwareSuite": {
      "name": "string",
      "version": "string"
   },
   "sources": [
      {
         "architecture": "string",
         "s3Bucket": "string",
         "s3Key": "string"
      }
   ]
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

application

ロボットアプリケーションのアプリケーション情報。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

currentRevisionId

ロボットアプリケーションのリビジョン ID。

型: 文字列

長さの制限：最小長は 1 です。最大長は 40 です。

Pattern: `[a-zA-Z0-9_.\-]*`

必須: いいえ

environment

ロボットアプリケーションの Docker イメージ URI が含まれているオブジェクト。

タイプ：Environment オブジェクト

必須: いいえ

robotSoftwareSuite

ロボットアプリケーションで使用するロボットソフトウェアスイート。

型: RobotSoftwareSuite オブジェクト

必須: はい

sources

ロボットアプリケーションのソース。

型: SourceConfig オブジェクトの配列

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json
```

```
{
   "arn": "string",
   "environment": {
      "uri": "string"
   },
   "lastUpdatedAt": number,
   "name": "string",
   "revisionId": "string",
   "robotSoftwareSuite": {
      "name": "string",
      "version": "string"
   },
   "sources": [
      {
         "architecture": "string",
         "etag": "string",
         "s3Bucket": "string",
         "s3Key": "string"
      }
   ],
   "version": "string"
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

更新されたロボットアプリケーションの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

environment

ロボットアプリケーションの Docker イメージ URI が含まれているオブジェクト。

タイプ：Environment オブジェクト

lastUpdatedAt

ロボットアプリケーションが最後に更新されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

name

ロボットアプリケーションの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: [a-zA-Z0-9_\-]*

revisionId

ロボットアプリケーションのリビジョン ID。

型: 文字列

長さの制限：最小長は 1 です。最大長は 40 です。

Pattern: [a-zA-Z0-9_.\-]*

robotSoftwareSuite

ロボットアプリケーションで使用するロボットソフトウェアスイート。

タイプ：RobotSoftwareSuite オブジェクト

sources

ロボットアプリケーションのソース。

型: Source オブジェクトの配列

version

ロボットアプリケーションのバージョン。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン：(\$LATEST)|[0-9]*

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

LimitExceededException

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++

- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

UpdateSimulationApplication

シミュレーションアプリケーションを更新します。

リクエストの構文

```
POST /updateSimulationApplication HTTP/1.1
Content-type: application/json

{
   "application": "string",
   "currentRevisionId": "string",
   "environment": {
      "uri": "string"
   },
   "renderingEngine": {
      "name": "string",
      "version": "string"
   },
   "robotSoftwareSuite": {
      "name": "string",
      "version": "string"
   },
   "simulationSoftwareSuite": {
      "name": "string",
      "version": "string"
   },
   "sources": [
      {
         "architecture": "string",
         "s3Bucket": "string",
         "s3Key": "string"
      }
   ]
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

application

シミュレーションアプリケーションのアプリケーション情報。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

currentRevisionId

ロボットアプリケーションのリビジョン ID。

型: 文字列

長さの制限：最小長は 1 です。最大長は 40 です。

Pattern: `[a-zA-Z0-9_.\-]*`

必須: いいえ

environment

シミュレーションアプリケーションの Docker イメージ URI が含まれているオブジェクト。

タイプ：Environment オブジェクト

必須: いいえ

renderingEngine

シミュレーションアプリケーションのレンダリングエンジン。

タイプ：RenderingEngine オブジェクト

必須: いいえ

robotSoftwareSuite

ロボットソフトウェアスイートに関する情報。

型: RobotSoftwareSuite オブジェクト

必須: はい

simulationSoftwareSuite

シミュレーションアプリケーションで使用するシミュレーションソフトウェアスイート。

型: SimulationSoftwareSuite オブジェクト

必須: はい

sources

シミュレーションアプリケーションのソース。

型: SourceConfig オブジェクトの配列

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "environment": {
      "uri": "string"
   },
   "lastUpdatedAt": number,
   "name": "string",
   "renderingEngine": {
      "name": "string",
      "version": "string"
   },
   "revisionId": "string",
   "robotSoftwareSuite": {
      "name": "string",
      "version": "string"
   },
   "simulationSoftwareSuite": {
      "name": "string",
      "version": "string"
   },
   "sources": [
```

```
      {
         "architecture": "string",
         "etag": "string",
         "s3Bucket": "string",
         "s3Key": "string"
      }
   ],
   "version": "string"
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

更新されたシミュレーションアプリケーションの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

environment

シミュレーションアプリケーションに使用されている Docker イメージ URI が含まれているオブジェクト。

タイプ：Environment オブジェクト

lastUpdatedAt

シミュレーションアプリケーションが最後に更新されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

name

シミュレーションアプリケーションの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: [a-zA-Z0-9_\-]*

renderingEngine

シミュレーションアプリケーションのレンダリングエンジン。

タイプ：RenderingEngine オブジェクト

revisionId

シミュレーションアプリケーションのリビジョン ID。

型: 文字列

長さの制限：最小長は 1 です。最大長は 40 です。

Pattern: [a-zA-Z0-9_.\-]*

robotSoftwareSuite

ロボットソフトウェアスイートに関する情報。

タイプ：RobotSoftwareSuite オブジェクト

simulationSoftwareSuite

シミュレーションアプリケーションで使用するシミュレーションソフトウェアスイート。

タイプ：SimulationSoftwareSuite オブジェクト

sources

シミュレーションアプリケーションのソース。

型: Source オブジェクトの配列

version

ロボットアプリケーションのバージョン。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン：(\$LATEST)|[0-9]*

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

LimitExceededException

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++

- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

UpdateWorldTemplate

ワールドテンプレートを更新します。

リクエストの構文

```
POST /updateWorldTemplate HTTP/1.1
Content-type: application/json

{
   "name": "string",
   "template": "string",
   "templateBody": "string",
   "templateLocation": {
      "s3Bucket": "string",
      "s3Key": "string"
   }
}
```

URI リクエストパラメータ

リクエストでは URI パラメータを使用しません。

リクエストボディ

リクエストは以下の JSON 形式のデータを受け入れます。

name

テンプレートの名前。

タイプ: 文字列

長さの制限: 最小長は 0 です。最大長は 255 です。

パターン: .*

必須: いいえ

template

更新するワールドテンプレートの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

templateBody

ワールドテンプレートの本文。

型: 文字列

長さの制限：最小長は 1 です。最大長は 262,144 です。

Pattern: `[\S\s]+`

必須: いいえ

templateLocation

ワールドテンプレートの場所。

タイプ：TemplateLocation オブジェクト

必須: いいえ

レスポンスの構文

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "createdAt": number,
   "lastUpdatedAt": number,
   "name": "string"
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

arn

ワールドテンプレートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

パターン: `arn:.*`

createdAt

ワールドテンプレートが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

lastUpdatedAt

ワールドテンプレートが最後に更新されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

name

ワールドテンプレートの名前。

型: 文字列

長さの制限: 最小長は 0 です。最大長は 255 です。

パターン：`.*`

エラー

すべてのアクションに共通のエラーについては、「共通エラー」を参照してください。

InternalServerException

AWS RoboMaker にサービスの問題が発生しました。もう一度やり直してください。

HTTP ステータスコード：500

InvalidParameterException

リクエストで指定されたパラメータが無効であるか、サポートされていないか、または使用できません。返されたメッセージはエラー値の説明を提供します。

HTTP ステータスコード：400

ResourceNotFoundException

指定されたリソースは存在しません。

HTTP ステータスコード：400

ThrottlingException

AWS RoboMaker は一時的にリクエストを処理できません。もう一度やり直してください。

HTTP ステータスコード：400

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS コマンドラインインターフェイス
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go バージョン 2 用 SDK
- AWS Java V2 用 SDK
- AWS V3 用 JavaScript SDK
- AWS PHP V3 用SDK
- AWS Python 用 SDK
- AWS ルビー V3 用 SDK

データ型

以下のデータ型 (タイプ) がサポートされています。

- BatchPolicy
- Compute
- ComputeResponse
- DataSource

- DataSourceConfig
- DeploymentApplicationConfig
- DeploymentConfig
- DeploymentJob
- DeploymentLaunchConfig
- Environment
- FailedCreateSimulationJobRequest
- FailureSummary
- Filter
- FinishedWorldsSummary
- Fleet
- LaunchConfig
- LoggingConfig
- NetworkInterface
- OutputLocation
- PortForwardingConfig
- PortMapping
- ProgressDetail
- RenderingEngine
- Robot
- RobotApplicationConfig
- RobotApplicationSummary
- RobotDeployment
- RobotSoftwareSuite
- S3KeyOutput
- S3Object
- SimulationApplicationConfig
- SimulationApplicationSummary
- SimulationJob
- SimulationJobBatchSummary

- SimulationJobRequest
- SimulationJobSummary
- SimulationSoftwareSuite
- Source
- SourceConfig
- TemplateLocation
- TemplateSummary
- Tool
- UploadConfiguration
- VPCConfig
- VPCConfigResponse
- WorldConfig
- WorldCount
- WorldExportJobSummary
- WorldFailure
- WorldGenerationJobSummary
- WorldSummary

BatchPolicy

バッチポリシーに関する情報。

コンテンツ

maxConcurrency

バッチの一部として作成され、同時にアクティブ状態になることがあるアクティブなシミュレーションジョブの数。

アクティブ状態に
は、Pending、Preparing、Running、Restarting、RunningFailed、Terminating が含まれます。その他の状態はすべて終了状態です。

タイプ: 整数

必須: いいえ

timeoutInSeconds

バッチが完了するまでの待機時間 (秒単位)

バッチがタイムアウトし、内部障害 (InternalServiceError など) のために失敗していた保留中のリクエストがある場合の場合、それらのリクエストは失敗リストに移動され、バッチステータスは Failed になります。保留中のリクエストが別の理由で失敗した場合、失敗した保留中のリクエストは失敗リストに移動され、バッチステータスは TimedOut になります。

型: Long

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

Compute

シミュレーションジョブの情報のコンピューティングを行います。

コンテンツ

computeType

シミュレーションジョブのタイプ情報のコンピューティングを行います。

型: 文字列

有効な値：CPU | GPU_AND_CPU

必須：いいえ

gpuUnitLimit

シミュレーションジョブの GPU 単位制限のコンピューティングを行います。に割り当てられた GPU の数と同じです。 SimulationJob

タイプ: 整数

有効な範囲: 最小値 は 0 です。最大値は 1 です。

必須: いいえ

simulationUnitLimit

シミュレーション単位制限。シミュレーションには、指定されたシミュレーション単位制限に比例して CPU とメモリが割り当てられます。シミュレーション単位は 1 vcpu と 2 GB のメモリです。指定された最大値まで消費した SU 使用率に対してのみ請求されます。デフォルト値は 15 です。

タイプ：整数

有効範囲: 最小値 は 1 です。最大値は 15 です。

必須：いいえ

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

ComputeResponse

シミュレーションジョブの情報のコンピューティングを行います。

コンテンツ

computeType

シミュレーションジョブのタイプレスポンス情報のコンピューティングを行います。

型: 文字列

有効な値：CPU | GPU_AND_CPU

必須：いいえ

gpuUnitLimit

シミュレーションジョブの GPU 単位制限のコンピューティングを行います。に割り当てられた GPU の数と同じです。 SimulationJob

タイプ: 整数

有効な範囲: 最小値 は 0 です。最大値は 1 です。

必須: いいえ

simulationUnitLimit

シミュレーション単位制限。シミュレーションには、指定されたシミュレーション単位制限に比例して CPU とメモリが割り当てられます。シミュレーション単位は 1 vcpu と 2 GB のメモリです。指定された最大値まで消費した SU 使用率に対してのみ請求されます。デフォルト値は 15 です。

タイプ：整数

有効範囲: 最小値 は 1 です。最大値は 15 です。

必須：いいえ

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

DataSource

データソースに関する情報。

コンテンツ

destination

コンテナイメージにファイルがマウントされている場所。

データソースの `type` を `Archive` に指定した場合は、アーカイブに Amazon S3 オブジェクトキーを提供する必要があります。オブジェクトキーは `.zip` または `.tar.gz` ファイルのいずれかを指します。

データソースの `type` を `Prefix` に指定した場合は、データソースに使用しているファイルを指す Amazon S3 プレフィックスを提供します。

データソースの `type` を `File` に指定した場合は、データソースとして使用しているファイルを指す Amazon S3 パスを提供します。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 1,024 です。

パターン: `.*`

必須: いいえ

name

データソースの名前。

型: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

必須: いいえ

s3Bucket

データファイルが存在する S3 バケット。

型: 文字列

長さの制限: 最小長は 3 です。最大長は 63 です。

パターン: `[a-z0-9][a-z0-9.\-]*[a-z0-9]`

必須: いいえ

s3Keys

データソースファイルを識別する S3 キーのリスト。

型: S3KeyOutput オブジェクトの配列

必須: いいえ

type

コンテナイメージまたはシミュレーションジョブに使用しているデータソースのデータ型。このフィールドを使用して、データソースがアーカイブ、Amazon S3 プレフィックス、またはファイルのいずれであるかを指定できます。

フィールドを指定しなかった場合、デフォルト値は `File` です。

型: 文字列

有効な値 : `Prefix | Archive | File`

必須 : いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

DataSourceConfig

データソースに関する情報。

コンテンツ

name

データソースの名前。

型: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

必須：はい

s3Bucket

データファイルが存在する S3 バケット。

型: 文字列

長さの制限: 最小長は 3 です。最大長は 63 です。

Pattern: `[a-z0-9][a-z0-9.\-]*[a-z0-9]`

必須：はい

s3Keys

データソースファイルを識別する S3 キーのリスト。

タイプ：文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 100 項目です。

長さの制限: 最小長は 0 です。最大長は 1,024 です。

Pattern: `.*`

必須：はい

destination

コンテナイメージにファイルがマウントされている場所。

データソースの `type` を `Archive` に指定した場合は、アーカイブに Amazon S3 オブジェクトキーを提供する必要があります。オブジェクトキーは `.zip` または `.tar.gz` ファイルのいずれかを指します。

データソースの `type` を `Prefix` に指定した場合は、データソースに使用しているファイルを指す Amazon S3 プレフィックスを提供します。

データソースの `type` を `File` に指定した場合は、データソースとして使用しているファイルを指す Amazon S3 パスを提供します。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 1,024 です。

パターン: `.*`

必須: いいえ

type

コンテナイメージまたはシミュレーションジョブに使用しているデータソースのデータ型。このフィールドを使用して、データソースがアーカイブ、Amazon S3 プレフィックス、またはファイルのいずれであるかを指定できます。

フィールドを指定しなかった場合、デフォルト値は `File` です。

型: 文字列

有効な値 : `Prefix | Archive | File`

必須 : いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK

- AWS ルビー V3 用 SDK

DeploymentApplicationConfig

デプロイアプリケーション設定に関する情報。

コンテンツ

application

ロボットアプリケーションの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

applicationVersion

アプリケーションのバージョン。

型: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[0-9]*`

必須：はい

launchConfig

起動設定

型: DeploymentLaunchConfig オブジェクト

必須：はい

以下の資料も参照してください。

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++

- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

DeploymentConfig

デプロイ設定に関する情報。

コンテンツ

concurrentDeploymentPercentage

デプロイを同時に受けるロボットの割合。

タイプ : 整数

有効範囲: 最小値 は 1 です。最大値は 100 です。

必須: いいえ

downloadConditionFile

ダウンロード条件ファイル。

タイプ : S3Object オブジェクト

必須: いいえ

failureThresholdPercentage

デプロイを停止する前に必ず失敗するデプロイの割合。

タイプ : 整数

有効範囲: 最小値 は 1 です。最大値は 100 です。

必須: いいえ

robotDeploymentTimeoutInSeconds

1 つのロボットに対するデプロイが完了するまでの待機時間 (秒) 1 分 ~ 7 日の範囲で時間を選択してください。デフォルトは 5 時間です。

型: Long

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

DeploymentJob

デプロイジョブに関する情報。

コンテンツ

arn

デプロイジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

createdAt

デプロイジョブが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

必須: いいえ

deploymentApplicationConfigs

デプロイアプリケーションの設定。

型: DeploymentApplicationConfig オブジェクトの配列

配列メンバー: 定数は 1 項目です。

必須: いいえ

deploymentConfig

デプロイ設定。

タイプ：DeploymentConfig オブジェクト

必須: いいえ

failureCode

デプロイジョブの失敗コード。

型: 文字列

有効な値：ResourceNotFound | EnvironmentSetupError | EtagMismatch | FailureThresholdBreached | RobotDeploymentAborted | RobotDeploymentNoResponse | RobotAgentConnectionTimeout | GreengrassDeploymentFailed | InvalidGreengrassGroup | MissingRobotArchitecture | MissingRobotApplicationArchitecture | MissingRobotDeploymentResource | GreengrassGroupVersionDoesNotExist | LambdaDeleted | ExtractingBundleFailure | PreLaunchFileFailure | PostLaunchFileFailure | BadPermissionError | DownloadConditionFailed | BadLambdaAssociated | InternalServerError | RobotApplicationDoesNotExist | DeploymentFleetDoesNotExist | FleetDeploymentTimeout

必須：いいえ

failureReason

デプロイジョブが失敗した理由の簡単な説明。

型: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: `.*`

必須: いいえ

fleet

フリートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

status

デプロイジョブのステータス。

型: 文字列

有効な値：`Pending | Preparing | InProgress | Failed | Succeeded | Canceled`

必須：いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

DeploymentLaunchConfig

デプロイ起動の設定情報。

コンテンツ

launchFile

起動ファイル名。

タイプ: 文字列

長さの制限: 最小長は 1 です。最大長は 1,024 です。

Pattern: `[a-zA-Z0-9_.\-]*`

必須：はい

packageName

パッケージ名。

タイプ: 文字列

長さの制限: 最小長は 1 です。最大長は 1,024 です。

Pattern: `[a-zA-Z0-9_.\-]*`

必須：はい

environmentVariables

ロボットアプリケーションの環境変数を指定するキー/値ペアの配列

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 20 項目です。

キーの長さ制限: 最小長は 1 です。最大長は 1024 です。

キーパターン: `[A-Z_][A-Z0-9_]*`

値の長さの制限: 最小長は 1 です。最大長は 1,024 です。

値のパターン: `.*`

必須: いいえ

postLaunchFile

デプロイの起動後のファイル。このファイルは、起動ファイルの後に実行されます。

タイプ: 文字列

長さの制限: 最小長は 1 です。最大長は 1,024 です。

パターン: .*

必須: いいえ

preLaunchFile

デプロイの起動前のファイル。このファイルは、起動ファイルの前に実行されます。

タイプ: 文字列

長さの制限: 最小長は 1 です。最大長は 1,024 です。

パターン: .*

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

Environment

ロボットアプリケーションまたはシミュレーションアプリケーションの Docker イメージ URI が含まれているオブジェクト。

コンテンツ

uri

ロボットアプリケーションまたはシミュレーションアプリケーションのいずれかの Docker イメージ URI。

タイプ: 文字列

長さの制限: 最小長は 1 です。最大長は 1,024 です。

パターン: .+

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

FailedCreateSimulationJobRequest

失敗したシミュレーションジョブ作成リクエストに関する情報。

コンテンツ

failedAt

シミュレーションジョブバッチが失敗したときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

必須: いいえ

failureCode

失敗コード。

型: 文字列

有効な値 : `InternalServiceError | RobotApplicationCrash | SimulationApplicationCrash | RobotApplicationHealthCheckFailure | SimulationApplicationHealthCheckFailure | BadPermissionsRobotApplication | BadPermissionsSimulationApplication | BadPermissionsS3Object | BadPermissionsS3Output | BadPermissionsCloudwatchLogs | SubnetIpLimitExceeded | ENILimitExceeded | BadPermissionsUserCredentials | InvalidBundleRobotApplication | InvalidBundleSimulationApplication | InvalidS3Resource | ThrottlingError | LimitExceeded | MismatchedEtag | RobotApplicationVersionMismatchedEtag | SimulationApplicationVersionMismatchedEtag | ResourceNotFound | RequestThrottled | BatchTimedOut | BatchCanceled | InvalidInput | WrongRegionS3Bucket | WrongRegionS3Output | WrongRegionRobotApplication | WrongRegionSimulationApplication | UploadContentMismatchError`

必須 : いいえ

failureReason

シミュレーションジョブリクエストの失敗理由。

タイプ: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: .*

必須: いいえ

request

シミュレーションジョブのリクエスト。

タイプ：SimulationJobRequest オブジェクト

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

FailureSummary

失敗したワールドに関する情報。

コンテンツ

failures

失敗したワールド。

タイプ：WorldFailure オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 100 項目です。

必須: いいえ

totalFailureCount

失敗の総数。

タイプ: 整数

必須：いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

Filter

フィルターに関する情報。

コンテンツ

name

フィルターの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

必須: いいえ

values

値のリスト。

タイプ：文字列の配列

配列メンバー: 定数は 1 項目です。

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

FinishedWorldsSummary

完了したワールドに関する情報。

コンテンツ

failureSummary

失敗したワールドに関する情報。

タイプ：FailureSummary オブジェクト

必須: いいえ

finishedCount

完了したワールドの総数。

タイプ: 整数

必須: いいえ

succeededWorlds

成功した世界の一覧。

タイプ：文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 100 項目です。

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

Fleet

フリートに関する情報。

コンテンツ

arn

フリートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

createdAt

フリートが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

必須: いいえ

lastDeploymentJob

最後のデプロイジョブの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

lastDeploymentStatus

最後のフリートデプロイのステータス。

型: 文字列

有効な値：`Pending | Preparing | InProgress | Failed | Succeeded | Canceled`

必須：いいえ

lastDeploymentTime

最後のデプロイの時間。

型: タイムスタンプ

必須: いいえ

name

フリートの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

LaunchConfig

起動設定に関する情報。

コンテンツ

command

General を RobotSoftwareSuite の値として指定した場合は、このフィールドを使用してコンテナイメージのコマンドのリストを指定できます。

SimulationRuntime を SimulationSoftwareSuite の値として指定した場合は、このフィールドを使用してコンテナイメージのコマンドのリストを指定できます。

型: 文字列の配列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: .+

必須: いいえ

environmentVariables

アプリケーション起動の環境変数。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 20 項目です。

キーの長さ制限: 最小長は 1 です。最大長は 1024 です。

キーパターン: [A-Z_][A-Z0-9_]*

値の長さの制限: 最小長は 1 です。最大長は 1,024 です。

値のパターン: .*

必須: いいえ

launchFile

起動ファイル名。

タイプ: 文字列

長さの制限: 最小長は 1 です。最大長は 1,024 です。

パターン: `[a-zA-Z0-9_.\-]*`

必須: いいえ

packageName

パッケージ名。

タイプ: 文字列

長さの制限: 最小長は 1 です。最大長は 1,024 です。

パターン: `[a-zA-Z0-9_.\-]*`

必須: いいえ

portForwardingConfig

ポート転送の設定。

タイプ：PortForwardingConfig オブジェクト

必須: いいえ

streamUI

アプリケーションに対してストリーミングセッションが設定されるかどうかを示すブール値。の場合`True`、AWS RoboMaker は接続を設定して、シミュレーションで実行中のアプリケーションを操作できるようにします。コンポーネントを設定して起動する必要があります。そのためにはグラフィカルユーザーインターフェイスが必要です。

型: ブール値

必須：いいえ

その他の参照資料

この API を言語固有の AWS SDK で使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK

- AWS ルビー V3 用 SDK

LoggingConfig

ログ処理の設定。

コンテンツ

recordAllRosTopics

このメンバーは非推奨になりました。

すべての ROS トピックを記録するかどうかを示すブール値。

> **Important**
> この API はサポートされなくなったため、使用した場合はエラーがスローされます。

型: ブール値

必須：いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

NetworkInterface

ネットワークインターフェースを記述します。

コンテンツ

networkInterfaceId

ネットワークインターフェイスの ID。

タイプ: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: .*

必須: いいえ

privateIpAddress

サブネット内のネットワークインターフェイスの IPv4 アドレス。

タイプ: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: .*

必須: いいえ

publicIpAddress

ネットワークインターフェイスの IPv4 パブリックアドレス。

タイプ: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: .*

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

OutputLocation

出力場所。

コンテンツ

s3Bucket

出力用の S3 バケット。

タイプ: 文字列

長さの制限: 最小長は 3 です。最大長は 63 です。

パターン: `[a-z0-9][a-z0-9.\-]*[a-z0-9]`

必須: いいえ

s3Prefix

出力ファイルが配置される `s3Bucket` にある S3 フォルダ。

タイプ: 文字列

長さの制限: 最小長は 1 です。最大長は 1,024 です。

パターン: `.*`

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

PortForwardingConfig

ポート転送の設定情報。

コンテンツ

portMappings

設定のポートマッピング。

タイプ：PortMapping オブジェクトの配列

の配列メンバー: 最小数は 0 項目です。最大数は 10 項目です。

必須：いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

PortMapping

ポートマッピングを表すオブジェクト。

コンテンツ

applicationPort

アプリケーションのポート番号。

タイプ: 整数

値の範囲: 最小値 は 1,024 です。最大値は 65,535 です。

必須: はい

jobPort

リモート接続ポイントとして使用するシミュレーションジョブインスタンスのポート番号。

タイプ：整数

有効範囲：最小値 は 1 最大値は 65,535 です。

必須: はい

enableOnPublicIp

パブリック IP でこのポートマッピングを有効にするかどうかを示すブール値。

型: ブール値

必須：いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

ProgressDetail

デプロイジョブの進行状況に関する情報。

コンテンツ

currentProgress

現在の進行状況。

検証中

デプロイを検証しています。

DownloadingExtracting

ロボットでバンドルのダウンロードと抽出を実行しています。

ExecutingPreLaunch

起動前スクリプトが提供されている場合はそれを実行しています。

起動中

ロボットアプリケーションを起動しています。

ExecutingPostLaunch

起動後スクリプトが提供されている場合はそれを実行しています。

完了

デプロイが完了しました。

型: 文字列

有効な値 : `Validating | DownloadingExtracting | ExecutingDownloadCondition | ExecutingPreLaunch | Launching | ExecutingPostLaunch | Finished`

必須 : いいえ

estimatedTimeRemainingSeconds

ステップにおける推定の残り時間 (秒)。これは現在、デプロイの `Downloading/Extracting` ステップにのみ適用されます。他のステップでは空です。

タイプ: 整数

必須: いいえ

percentDone

ステップの実行済みの割合。これは現在、デプロイの `Downloading/Extracting` ステップにのみ適用されます。他のステップでは空です。

タイプ: 浮動小数点

値の範囲: 最小値 は 0.0 です。最大値は 100.0 です。

必須: いいえ

targetResource

デプロイジョブの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: `.*`

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

RenderingEngine

レンダリングエンジンに関する情報。

コンテンツ

name

レンダリングエンジンの名前。

型: 文字列

有効な値：OGRE

必須：いいえ

version

レンダリングエンジンのバージョン。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 4 です。

Pattern: 1.x

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

Robot

ロボットに関する情報。

コンテンツ

architecture

ロボットのアーキテクチャ。

型: 文字列

有効な値：X86_64 | ARM64 | ARMHF

必須：いいえ

arn

ロボットの Amazon リソースネーム (ARN)

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: arn:.*

必須: いいえ

createdAt

ロボットが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

必須: いいえ

fleetArn

フリートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: arn:.*

必須: いいえ

greenGrassGroupId

ロボットに関連付けられる Greengrass グループ。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `.*`

必須: いいえ

lastDeploymentJob

最後のデプロイジョブの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

lastDeploymentTime

最後のデプロイの時間。

型: タイムスタンプ

必須: いいえ

name

ロボットの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

必須: いいえ

status

ロボットのステータス。

型: 文字列

有効な値 : `Available | Registered | PendingNewDeployment | Deploying | Failed | InSync | NoResponse`

必須 : いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

RobotApplicationConfig

ロボットのアプリケーション設定情報。

コンテンツ

application

ロボットアプリケーションのアプリケーション情報。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

launchConfig

ロボットアプリケーションの起動設定。

型: LaunchConfig オブジェクト

必須: はい

applicationVersion

ロボットアプリケーションのバージョン。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `(\$LATEST)|[0-9]*`

必須: いいえ

tools

ロボットアプリケーションに対して設定されたツールに関する情報。

タイプ：Tool オブジェクトの配列

の配列メンバー: 最小数は 0 項目です。最大数は 10 項目です。

必須: いいえ

uploadConfigurations

ロボットアプリケーションのアップロード設定。

タイプ：UploadConfiguration オブジェクトの配列

の配列メンバー: 最小数は 0 項目です。最大数は 10 項目です。

必須: いいえ

useDefaultTools

このメンバーは非推奨になりました。

デフォルトのロボットアプリケーションツールを使用するかどうかを示すブール値。デフォルトのツールは、rviz、rqt、terminal、rosbag レコードです。デフォルトは `False` です。

> **Important**
> この API はサポートされなくなったため、使用した場合はエラーがスローされます。

型: ブール値

必須: いいえ

useDefaultUploadConfigurations

このメンバーは非推奨になりました。

デフォルトのアップロード設定を使用するかどうかを示すブール値。デフォルトでは、アプリケーションの終了時に `.ros` ファイルと `.gazebo` ファイルがアップロードされ、すべての ROS トピックが記録されます。

この値を設定する場合は、`outputLocation` を指定する必要があります。

> **Important**
> この API はサポートされなくなったため、使用した場合はエラーがスローされます。

型: ブール値

必須：いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

RobotApplicationSummary

ロボットアプリケーションの概要情報。

コンテンツ

arn

ロボットの Amazon リソースネーム (ARN)

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

lastUpdatedAt

ロボットアプリケーションが最後に更新されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

必須: いいえ

name

ロボットアプリケーションの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

必須: いいえ

robotSoftwareSuite

ロボットソフトウェアスイートに関する情報。

タイプ：RobotSoftwareSuite オブジェクト

必須: いいえ

version

ロボットアプリケーションのバージョン。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: (\$LATEST)|[0-9]*

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

RobotDeployment

ロボットデプロイに関する情報。

コンテンツ

arn

ロボットデプロイの Amazon リソースネーム (ARN)

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

deploymentFinishTime

デプロイが完了したときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

必須: いいえ

deploymentStartTime

デプロイが開始されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

必須: いいえ

failureCode

ロボットデプロイの失敗コード。

型: 文字列

有効な値：`ResourceNotFound | EnvironmentSetupError | EtagMismatch | FailureThresholdBreached | RobotDeploymentAborted | RobotDeploymentNoResponse | RobotAgentConnectionTimeout | GreengrassDeploymentFailed | InvalidGreengrassGroup |`

MissingRobotArchitecture | MissingRobotApplicationArchitecture | MissingRobotDeploymentResource | GreengrassGroupVersionDoesNotExist | LambdaDeleted | ExtractingBundleFailure | PreLaunchFileFailure | PostLaunchFileFailure | BadPermissionError | DownloadConditionFailed | BadLambdaAssociated | InternalServerError | RobotApplicationDoesNotExist | DeploymentFleetDoesNotExist | FleetDeploymentTimeout

必須：いいえ

failureReason

ロボットデプロイが失敗した理由の簡単な説明。

タイプ: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: .*

必須: いいえ

progressDetail

デプロイの進行方法に関する情報

タイプ：ProgressDetail オブジェクト

必須: いいえ

status

ロボットデプロイのステータス。

型: 文字列

有効な値：Available | Registered | PendingNewDeployment | Deploying | Failed | InSync | NoResponse

必須：いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

RobotSoftwareSuite

ロボットソフトウェアスイートに関する情報。

コンテンツ

name

ロボットソフトウェアスイートの名前。General は、サポートされている唯一の値です。

型: 文字列

有効な値 : ROS | ROS2 | General

必須 : いいえ

version

ロボットソフトウェアスイートのバージョン。一般的なソフトウェアスイートには適用されません。

型: 文字列

有効な値 : Kinetic | Melodic | Dashing | Foxy

必須 : いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

S3KeyOutput

S3 キーに関する情報。

コンテンツ

etag

オブジェクトの Etag。

タイプ: 文字列

必須: いいえ

s3Key

S3 キー。

タイプ: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: .*

必須: いいえ

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

S3Object

S3 オブジェクトに関する情報。

コンテンツ

bucket

オブジェクトを含むバケット。

タイプ: 文字列

長さの制限: 最小長は 3 です。最大長は 63 です。

Pattern: `[a-z0-9][a-z0-9.\-]*[a-z0-9]`

必須：はい

key

オブジェクトのキー。

タイプ: 文字列

長さの制限: 最小長は 1 です。最大長は 1,024 です。

Pattern: `.*`

必須：はい

etag

オブジェクトの Etag。

タイプ: 文字列

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++

- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

SimulationApplicationConfig

シミュレーションアプリケーション設定に関する情報。

コンテンツ

application

シミュレーションアプリケーションのアプリケーション情報。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須：はい

launchConfig

シミュレーションアプリケーションの起動設定。

型: LaunchConfig オブジェクト

必須: はい

applicationVersion

シミュレーションアプリケーションのバージョン。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `(\$LATEST)|[0-9]*`

必須: いいえ

tools

シミュレーションアプリケーションに対して設定されたツールに関する情報。

タイプ：Tool オブジェクトの配列

の配列メンバー: 最小数は 0 項目です。最大数は 10 項目です。

必須: いいえ

uploadConfigurations

シミュレーションアプリケーションのアップロード設定に関する情報。

タイプ：UploadConfiguration オブジェクトの配列

の配列メンバー: 最小数は 0 項目です。最大数は 10 項目です。

必須: いいえ

useDefaultTools

このメンバーは非推奨になりました。

デフォルトのシミュレーションアプリケーションツールを使用するかどうかを示すブール値。デフォルトのツールは、rviz、rqt、terminal、rosbag レコードです。デフォルトは `False` です。

> **Important**
> この API はサポートされなくなったため、使用した場合はエラーがスローされます。

型: ブール値

必須: いいえ

useDefaultUploadConfigurations

このメンバーは非推奨になりました。

デフォルトのアップロード設定を使用するかどうかを示すブール値。デフォルトでは、アプリケーションの終了時に `.ros` ファイルと `.gazebo` ファイルがアップロードされ、すべての ROS トピックが記録されます。

この値を設定する場合は、`outputLocation` を指定する必要があります。

> **Important**
> この API はサポートされなくなったため、使用した場合はエラーがスローされます。

型: ブール値

必須: いいえ

worldConfigs

ワールド設定のリスト。

> **Important**
> この API はサポートされなくなったため、使用した場合はエラーがスローされます。

タイプ：WorldConfig オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 1 項目です。

必須：いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

SimulationApplicationSummary

シミュレーションアプリケーションの概要情報。

コンテンツ

arn

シミュレーションアプリケーションの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

lastUpdatedAt

シミュレーションアプリケーションが最後に更新されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

必須: いいえ

name

シミュレーションアプリケーションの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

必須: いいえ

robotSoftwareSuite

ロボットソフトウェアスイートに関する情報。

タイプ：RobotSoftwareSuite オブジェクト

必須: いいえ

simulationSoftwareSuite

シミュレーションソフトウェアスイートに関する情報。

タイプ：SimulationSoftwareSuite オブジェクト

必須: いいえ

version

シミュレーションアプリケーションのバージョン。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `(\$LATEST)|[0-9]*`

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

SimulationJob

シミュレーションジョブに関する情報。

コンテンツ

arn

シミュレーションジョブの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

clientRequestToken

この `SimulationJob` リクエストの一意な識別子。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 64 文字です。

パターン: `[a-zA-Z0-9_\-=]*`

必須: いいえ

compute

シミュレーションジョブの情報のコンピューティングを行います。

タイプ：ComputeResponse オブジェクト

必須: いいえ

dataSources

シミュレーションジョブのデータソース。

型: DataSource オブジェクトの配列

必須: いいえ

failureBehavior

シミュレーションジョブの失敗動作。

続行

4XX エラーコード後の最大タイムアウト期間中もホストが実行されるようにします。

失敗

シミュレーションジョブを停止し、インスタンスを終了します。

型: 文字列

有効な値 : `Fail | Continue`

必須 : いいえ

failureCode

シミュレーションジョブが失敗した場合の失敗コード。

型: 文字列

有効な値 : `InternalServiceError | RobotApplicationCrash | SimulationApplicationCrash | RobotApplicationHealthCheckFailure | SimulationApplicationHealthCheckFailure | BadPermissionsRobotApplication | BadPermissionsSimulationApplication | BadPermissionsS3Object | BadPermissionsS3Output | BadPermissionsCloudwatchLogs | SubnetIpLimitExceeded | ENILimitExceeded | BadPermissionsUserCredentials | InvalidBundleRobotApplication | InvalidBundleSimulationApplication | InvalidS3Resource | ThrottlingError | LimitExceeded | MismatchedEtag | RobotApplicationVersionMismatchedEtag | SimulationApplicationVersionMismatchedEtag | ResourceNotFound | RequestThrottled | BatchTimedOut | BatchCanceled | InvalidInput | WrongRegionS3Bucket | WrongRegionS3Output | WrongRegionRobotApplication | WrongRegionSimulationApplication | UploadContentMismatchError`

必須 : いいえ

failureReason

シミュレーションジョブが失敗した理由。

タイプ: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: `.*`

必須: いいえ

iamRole

関連ポリシーで指定されたAWS APIがシミュレーションインスタンスによって呼び出されるようにする IAM ロール。これは、シミュレーションジョブに認証情報が渡される方法になります。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `arn:aws:iam::\w+:role/.*`

必須: いいえ

lastStartedAt

シミュレーションジョブが最後に開始されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

必須: いいえ

lastUpdatedAt

シミュレーションジョブが最後に更新されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

必須: いいえ

loggingConfig

ログ処理の設定。

タイプ：LoggingConfig オブジェクト

必須: いいえ

maxJobDurationInSeconds

最大シミュレーションジョブ時間 (秒)。値は 8 日 (691,200 秒) 以下でなければなりません。

タイプ: Long

必須: いいえ

name

シミュレーションジョブの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

必須: いいえ

networkInterface

ネットワークインターフェイスに関する情報の取得

タイプ：NetworkInterface オブジェクト

必須: いいえ

outputLocation

シミュレーションジョブにより生成される出力ファイルの場所。

タイプ：OutputLocation オブジェクト

必須: いいえ

robotApplications

ロボットアプリケーションのリスト。

型: RobotApplicationConfig オブジェクトの配列

配列メンバー: 定数は 1 項目です。

必須: いいえ

simulationApplications

シミュレーションアプリケーションのリスト。

型: SimulationApplicationConfig オブジェクトの配列

配列メンバー: 定数は 1 項目です。

必須: いいえ

simulationTimeMillis

シミュレーションジョブの実行時間 (ミリ秒)。

タイプ: Long

必須: いいえ

status

シミュレーションジョブのステータス。

型: 文字列

有効な値 : `Pending | Preparing | Running | Restarting | Completed | Failed | RunningFailed | Terminating | Terminated | Canceled`

必須 : いいえ

tags

シミュレーションジョブにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

必須: いいえ

vpcConfig

VPC 設定情報。

タイプ : VPCConfigResponse オブジェクト

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

SimulationJobBatchSummary

シミュレーションジョブバッチに関する情報。

コンテンツ

arn

バッチの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

createdAt

シミュレーションジョブバッチが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

必須: いいえ

createdRequestCount

作成されたシミュレーションジョブリクエストの数。

タイプ: 整数

必須: いいえ

failedRequestCount

失敗したシミュレーションジョブリクエストの数。

タイプ: 整数

必須: いいえ

lastUpdatedAt

シミュレーションジョブバッチが最後に更新されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

必須: いいえ

pendingRequestCount

保留中のシミュレーションジョブリクエストの数。

タイプ: 整数

必須: いいえ

status

シミュレーションジョブバッチのステータス。

保留中

シミュレーションジョブバッチリクエストが保留中です。

InProgress

シミュレーションジョブバッチが進行中です。

[失敗]

シミュレーションジョブバッチが失敗しました。内部障害 (InternalServiceError など) により、1 つまたは複数のシミュレーションジョブリクエストを完了できませんでした 詳細については、「failureCode」と「 failureReason」 を参照してください。

完了

シミュレーションバッチジョブが完了しました。バッチは、(1) バッチ内に保留中のシミュレーションジョブリクエストがなく、失敗したシミュレーションジョブリクエストがいずれも InternalServiceError を原因としない場合、および (2) 作成されたすべてのシミュレーションジョブが終了状態に達したとき (例えば Completed または Failed) に完了となります。

キャンセル

シミュレーションバッチジョブがキャンセルされました。

キャンセル中

シミュレーションバッチジョブをキャンセルしています。

完了中

シミュレーションバッチジョブを完了しています。

TimingOut

シミュレーションバッチジョブがタイムアウトしています。

バッチがタイムアウトし、内部障害 (`InternalServiceError` など) のために失敗していた保留中のリクエストがある場合の場合、バッチステータスは `Failed` になります。このような失敗リクエストがない場合、バッチステータスは `TimedOut` になります。

TimedOut

シミュレーションバッチジョブがタイムアウトしました。

型: 文字列

有効な値 : `Pending | InProgress | Failed | Completed | Canceled | Canceling | Completing | TimingOut | TimedOut`

必須 : いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

SimulationJobRequest

シミュレーションジョブリクエストに関する情報。

コンテンツ

maxJobDurationInSeconds

最大シミュレーションジョブ時間 (秒)。値は 8 日 (691,200 秒) 以下でなければなりません。

タイプ: Long

必須: はい

compute

シミュレーションジョブの情報のコンピューティングを行います。

タイプ : Compute オブジェクト

必須: いいえ

dataSources

データソースを指定して、S3 からシミュレーションに読み取り専用ファイルをマウントします。これらのファイルは `/opt/robomaker/datasources/data_source_name` で入手できます。

> **Note**
>
> ファイル数は 100 個、全 `DataSourceConfig` オブジェクトの合計サイズは 25GB に制限されます。

型: DataSourceConfig オブジェクトの配列

配列メンバー : 最小数は 1 項目です。最大数は 6 項目です。

必須: いいえ

failureBehavior

シミュレーションジョブの失敗動作。

続行

4XX エラーコード後の最大タイムアウト期間中もホストが実行されるようにします。

失敗

シミュレーションジョブを停止し、インスタンスを終了します。

型: 文字列

有効な値：`Fail | Continue`

必須：いいえ

iamRole

関連ポリシーで指定されたAWS APIがシミュレーションインスタンスによって呼び出されるようにする IAM ロール名。これは、シミュレーションジョブに認証情報が渡される方法になります。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `arn:aws:iam::\w+:role/.*`

必須: いいえ

loggingConfig

ログ処理の設定。

タイプ：LoggingConfig オブジェクト

必須: いいえ

outputLocation

出力場所。

タイプ：OutputLocation オブジェクト

必須: いいえ

robotApplications

シミュレーションジョブで使用するロボットアプリケーション。

型: RobotApplicationConfig オブジェクトの配列

配列メンバー: 定数は 1 項目です。

必須: いいえ

simulationApplications

シミュレーションジョブで使用するシミュレーションアプリケーション。

型: SimulationApplicationConfig オブジェクトの配列

配列メンバー: 定数は 1 項目です。

必須: いいえ

tags

シミュレーションジョブリクエストにアタッチされているタグキーとタグ値を含むマップ。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 50 項目です。

キーの長さ制限: 最小長さは 1 です。最大長は 128 です。

キーパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

値の長さの制限: 最小長は 0 です。最大長は 256 です。

値のパターン: `[a-zA-Z0-9 _.\-\/+=:]*`

必須: いいえ

useDefaultApplications

シミュレーションジョブでデフォルトのアプリケーションを使用するかどうかを示すブール値。デフォルトのアプリケーションには、Gazebo、rqt、rviz、ターミナルアクセスが含まれます。

型: ブール値

必須: いいえ

vpcConfig

シミュレーションジョブが VPC 内のリソースにアクセスする場合は、セキュリティグループ ID とサブネット ID のリストを識別するこのパラメータを指定します。これらは同じ VPC に属して

いる必要があります。少なくとも 1 つのセキュリティグループと 2 つのサブネット ID を指定する必要があります。

タイプ：VPCConfig オブジェクト

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

SimulationJobSummary

シミュレーションジョブの概要情報。

コンテンツ

arn

シミュレーションジョブの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

computeType

シミュレーションジョブの概要のコンピューティングタイプ。

型: 文字列

有効な値：`CPU | GPU_AND_CPU`

必須：いいえ

dataSourceNames

データソースの名前。

型: 文字列の配列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

必須: いいえ

lastUpdatedAt

シミュレーションジョブが最後に更新されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

必須: いいえ

name

シミュレーションジョブの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

必須: いいえ

robotApplicationNames

シミュレーションジョブのロボットアプリケーション名のリスト。

型: 文字列の配列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

必須: いいえ

simulationApplicationNames

シミュレーションジョブのシミュレーションアプリケーション名のリスト。

型: 文字列の配列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

必須: いいえ

status

シミュレーションジョブのステータス。

型: 文字列

有効な値：`Pending | Preparing | Running | Restarting | Completed | Failed | RunningFailed | Terminating | Terminated | Canceled`

必須：いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

SimulationSoftwareSuite

シミュレーションソフトウェアスイートに関する情報。

コンテンツ

name

シミュレーションソフトウェアスイートの名前。SimulationRuntime は、現在サポートされている唯一の値です。

型: 文字列

有効な値 : Gazebo | RosbagPlay | SimulationRuntime

必須 : いいえ

version

シミュレーションソフトウェアスイートのバージョン。SimulationRuntime には該当なし。

型: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: 7|9|11|Kinetic|Melodic|Dashing|Foxy

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

Source

ソースに関する情報。

コンテンツ

architecture

アプリケーションのターゲットプロセッサアーキテクチャ。

型: 文字列

有効な値：`X86_64 | ARM64 | ARMHF`

必須：いいえ

etag

`s3Bucket` と `s3Key` によって指定されたオブジェクトのハッシュ。

タイプ: 文字列

必須: いいえ

s3Bucket

s3 バケット名

タイプ: 文字列

長さの制限: 最小長は 3 です。最大長は 63 です。

パターン: `[a-z0-9][a-z0-9.\-]*[a-z0-9]`

必須: いいえ

s3Key

S3 オブジェクトキー。

タイプ: 文字列

長さの制限: 最小長は 1 です。最大長は 1,024 です。

パターン: `.*`

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

SourceConfig

ソース構成に関する情報。

コンテンツ

architecture

アプリケーションのターゲットプロセッサアーキテクチャ。

型: 文字列

有効な値：`X86_64 | ARM64 | ARMHF`

必須：いいえ

s3Bucket

Amazon S3 バケット名。

タイプ: 文字列

長さの制限: 最小長は 3 です。最大長は 63 です。

パターン: `[a-z0-9][a-z0-9.\-]*[a-z0-9]`

必須: いいえ

s3Key

S3 オブジェクトキー。

タイプ: 文字列

長さの制限: 最小長は 1 です。最大長は 1,024 です。

パターン: `.*`

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

TemplateLocation

テンプレートの場所に関する情報。

コンテンツ

s3Bucket

Amazon S3 バケット名。

タイプ: 文字列

長さの制限: 最小長は 3 です。最大長は 63 です。

Pattern: `[a-z0-9][a-z0-9.\-]*[a-z0-9]`

必須：はい

s3Key

データソースファイルを識別する S3 キーのリスト。

タイプ: 文字列

長さの制限: 最小長は 1 です。最大長は 1,024 です。

Pattern: `.*`

必須：はい

以下の資料も参照してください。

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

TemplateSummary

テンプレートの概要情報。

コンテンツ

arn

テンプレートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

createdAt

テンプレートが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

必須: いいえ

lastUpdatedAt

テンプレートが最後に更新されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

必須: いいえ

name

テンプレートの名前。

タイプ: 文字列

長さの制限: 最小長は 0 です。最大長は 255 です。

パターン: `.*`

必須: いいえ

version

使用しているテンプレートのバージョン。

タイプ: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: .*

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

Tool

ツールに関する情報。ツールはシミュレーションジョブで使用されます。

コンテンツ

command

ツールのコマンドライン引数。これには実行可能ツール名を含める必要があります。

タイプ: 文字列

長さの制限: 最小長は 1 です。最大長は 1,024 です。

Pattern: `.*`

必須：はい

name

ツールの名前。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

必須：はい

exitBehavior

終了動作によって、ツール実行の終了時に何が発生するかが決まります。`RESTART` の場合はツールが再起動されます。`FAIL` の場合はジョブが終了します。デフォルトは `RESTART` です。

型: 文字列

有効な値：`FAIL | RESTART`

必須：いいえ

streamOutputToCloudWatch

ツールのログを記録するかどうかを示すブール値。 CloudWatch デフォルトは `False` です。

型: ブール値

必須: いいえ

streamUI

ツールに対してストリーミングセッションが設定されるかどうかを示すブール値。の場合`True`、AWS RoboMaker は接続を設定して、シミュレーションで実行中のツールを操作できるようにします。そのためにはグラフィカルユーザーインターフェイスが必要です。デフォルトは `False` です。

型: ブール値

必須：いいえ

その他の参照資料

この API を言語固有の AWS SDK で使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

UploadConfiguration

アップロード設定情報を提供します。シミュレーションジョブから、指定した場所にファイルがアップロードされます。

コンテンツ

name

Amazon S3 でのファイルのアップロード先となる場所を指定するプレフィックス。最終パスを決定するために、シミュレーション出力場所に追加されます。

例えば、シミュレーション出力場所が `s3://my-bucket` であり、アップロード設定の名前が `robot-test` である場合、ファイルは `s3://my-bucket/<simid>/<runid>/robot-test` にアップロードされます。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: `[a-zA-Z0-9_\-]*`

必須：はい

path

アップロードするファイルのパスを指定します。スーパーアスタリスクとして ** を追加すると、標準 Unix glob マッチングルールが受け入れられます。例えば、`/var/log/**.log` を指定すると `/var/log` ディレクトリツリー内のすべての .log ファイルが収集されます。他の例については、「Glob Library」を参照してください。

タイプ: 文字列

長さの制限: 最小長は 1 です。最大長は 1,024 です。

Pattern: `.*`

必須：はい

uploadBehavior

ファイルをアップロードするタイミングを指定します。

UPLOAD_ON_TERMINATE

シミュレーションが TERMINATING 状態に入ると一致するファイルがアップロードされます。一致するファイルは、すべてのコード (ツールを含む) が停止するまではアップロードされません。

ファイルのアップロード時に問題が発生した場合は、アップロードが再試行されます。問題が解決しない場合、それ以上のアップロードは試行されません。

UPLOAD_ROLLING_AUTO_REMOVE

一致するファイルは作成時にアップロードされます。アップロード後に削除されます。指定したパスは 5 秒ごとにチェックされます。最終チェックは、すべてのコード (ツールを含む) が停止したときに行われます。

型: 文字列

有効な値 : UPLOAD_ON_TERMINATE | UPLOAD_ROLLING_AUTO_REMOVE

必須 : はい

以下の資料も参照してください。

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

VPCConfig

シミュレーションジョブが VPC 内のリソースにアクセスする場合は、セキュリティグループ ID とサブネット ID のリストを識別するこのパラメータを指定します。これらは同じ VPC に属している必要があります。少なくとも 1 つのセキュリティグループと 2 つのサブネット ID を指定する必要があります。

コンテンツ

subnets

VPC 内の 1 つ以上のサブネット ID のリスト。

タイプ：文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 16 項目です。

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: .+

必須：はい

assignPublicIp

パブリック IP アドレスを割り当てるかどうかを示すブール値。

型: ブール値

必須: いいえ

securityGroups

VPC 内の 1 つ以上のセキュリティグループ ID のリスト。

タイプ：文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 5 項目です。

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: .+

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

VPCConfigResponse

シミュレーションジョブに関連付けられた VPC 設定。

コンテンツ

assignPublicIp

パブリック IP が割り当てられたかどうかを示すブール値。

型: ブール値

必須: いいえ

securityGroups

シミュレーションジョブに関連付けられているセキュリティグループ ID のリスト。

タイプ：文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 5 項目です。

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: .+

必須: いいえ

subnets

シミュレーションジョブに関連付けられているサブネット ID のリスト。

タイプ：文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 16 項目です。

長さの制限：最小長は 1 です。最大長は 255 です。

パターン: .+

必須: いいえ

vpcId

シミュレーションジョブに関連付けられた VPC ID。

タイプ: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: .*

必須: いいえ

その他の参照資料

言語固有の AWS SDK の 1 つでこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

WorldConfig

ワールドの設定情報。

コンテンツ

world

シミュレーションによって生成された世界 WorldForge。

型: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

その他の参照資料

この API を言語固有の AWS SDK で使用する方法について詳しくは、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

WorldCount

作成されるワールドの数。一意の間取り図の数と、各間取り図の一意のインテリアの数を設定できます。例えば、一意のインテリアが 20 個あるワールドが 1 つ必要な場合は、`floorplanCount = 1` と `interiorCountPerFloorplan = 20` を設定します。これでワールドが 20 個になります (`floorplanCount * interiorCountPerFloorplan`)。

`floorplanCount = 4` と `interiorCountPerFloorplan = 5` を設定した場合、一意の間取り図が 5 つあるワールドが 20 個存在することになります。

コンテンツ

floorplanCount

一意の間取り図の数。

タイプ: 整数

必須: いいえ

interiorCountPerFloorplan

間取り図 1 つ当たりの一意のインテリアの数。

タイプ: 整数

必須：いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

WorldExportJobSummary

ワールドエクスポートジョブに関する情報。

コンテンツ

arn

ワールドエクスポートジョブの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

createdAt

ワールドのエクスポートジョブが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

必須: いいえ

outputLocation

出力場所。

タイプ：OutputLocation オブジェクト

必須: いいえ

status

ワールドエクスポートジョブのステータス。

保留中

ワールドエクスポートジョブのリクエストが保留中です。

実行中

ワールドエクスポートジョブが実行中です。

完了

ワールドエクスポートジョブが完了しました。

[失敗]

ワールドエクスポートジョブが失敗しました。詳細については「failureCode」を参照してください。

キャンセル

ワールドエクスポートジョブがキャンセルされました。

キャンセル中

ワールドエクスポートジョブをキャンセルしています。

型: 文字列

有効な値 : Pending | Running | Completed | Failed | Canceling | Canceled

必須 : いいえ

worlds

ワールドのリスト。

タイプ : 文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 100 項目です。

長さの制限 : 最小長は 1 です。最大長は 1224 です。

Pattern: arn:.*

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

WorldFailure

失敗したワールドに関する情報。

コンテンツ

failureCode

ワールドエクスポートジョブが失敗した場合の失敗コード:

InternalServiceError

内部サービスエラー。

LimitExceeded

リクエストされたリソースが最大許容数を超えているか、または同時ストリームリクエストの数が最大許容数を超えています。

ResourceNotFound

指定したリソースが見つかりませんでした。

RequestThrottled

リクエストがスロットリングされました。

InvalidInput

リクエストの入力パラメータが無効です。

型: 文字列

有効な値 : `InternalServiceError | LimitExceeded | ResourceNotFound | RequestThrottled | InvalidInput | AllWorldGenerationFailed`

必須 : いいえ

failureCount

失敗したワールドの数。

タイプ: 整数

必須: いいえ

sampleFailureReason

ワールドが失敗した理由のサンプル。ワールドエラーが集計されます。サンプルは `sampleFailureReason` として使用されます。

タイプ: 文字列

長さの制限: 最小長は 0 です。最大長は 1,024 です。

パターン: `.*`

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

WorldGenerationJobSummary

ワールドジェネレータージョブに関する情報。

コンテンツ

arn

ワールドジェネレータージョブの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

createdAt

ワールドのジェネレータージョブが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

必須: いいえ

failedWorldCount

失敗したワールドの数。

タイプ: 整数

必須: いいえ

status

ワールドジェネレータージョブのステータス:

保留中

ワールドジェネレータージョブのリクエストが保留中です。

実行中

ワールドジェネレータージョブが実行中です。

完了

ワールドジェネレータージョブが完了しました。

[失敗]

ワールドジェネレータージョブが失敗しました。詳細については、「failureCode」を参照してください。

PartialFailed

一部のワールドが生成されませんでした。

キャンセル

ワールドジェネレータージョブがキャンセルされました。

キャンセル中

ワールドジェネレータージョブをキャンセルしています。

型: 文字列

有効な値 : `Pending | Running | Completed | Failed | PartialFailed | Canceling | Canceled`

必須 : いいえ

succeededWorldCount

生成されたワールドの数。

タイプ: 整数

必須: いいえ

template

ワールドテンプレートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限 : 最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

worldCount

ワールドカウントに関する情報。

タイプ：WorldCount オブジェクト

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

WorldSummary

ワールドに関する情報。

コンテンツ

arn

ワールドの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

createdAt

ワールドが作成されたときの、エポックからのミリ秒単位の時間。

型: タイムスタンプ

必須: いいえ

generationJob

ワールド生成ジョブの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

template

ワールドテンプレートの Amazon リソースネーム (ARN)。

タイプ: 文字列

長さの制限：最小長は 1 です。最大長は 1224 です。

Pattern: `arn:.*`

必須: いいえ

その他の参照資料

言語固有の AWS SDK でこの API を使用する方法の詳細については、以下を参照してください。

- AWS SDK for C++
- AWS Java V2 用 SDK
- AWS ルビー V3 用 SDK

共通エラー

このセクションでは、AWS のすべてのサービスの API アクションに共通のエラーを一覧表示しています。このサービスの API アクションに固有のエラーについては、その API アクションのトピックを参照してください。

AccessDeniedException

このアクションを実行する十分なアクセス権限がありません。

HTTP ステータスコード: 400

IncompleteSignature

リクエストの署名が AWS 基準に適合しません。

HTTP ステータスコード: 400

InternalFailure

リクエストの処理が、不明なエラー、例外、または障害により実行できませんでした。

HTTP ステータスコード: 500

InvalidAction

リクエストされたアクション、またはオペレーションは無効です。アクションが正しく入力されていることを確認します。

HTTP ステータスコード: 400

InvalidClientTokenId

指定された x.509 証明書、または AWS アクセスキー ID が見つかりません。

HTTP ステータスコード: 403

NotAuthorized

このアクションを実行するにはアクセス許可が必要です。

HTTP ステータスコード: 400

OptInRequired

サービスを利用するためには、AWS アクセスキー ID を取得する必要があります。

HTTP ステータスコード: 403

RequestExpired

リクエストの日付スタンプの 15 分以上後またはリクエストの有効期限 (署名付き URL の場合など) の 15 分以上後に、リクエストが到着しました。または、リクエストの日付スタンプが現在より 15 分以上先です。

HTTP ステータスコード: 400

ServiceUnavailable

リクエストは、サーバーの一時的障害のために実行に失敗しました。

HTTP ステータスコード: 503

ThrottlingException

リクエストは、制限が必要なために実行が拒否されました。

HTTP ステータスコード: 400

ValidationError

入力が、AWS サービスで指定された制約を満たしていません。

HTTP ステータスコード: 400

共通パラメータ

次のリストには、すべてのアクションが署名バージョン 4 リクエストにクエリ文字列で署名するために使用するパラメータを示します。アクション固有のパラメータは、アクションのトピックに示されています。Signature Version 4 の詳細については、「IAM ユーザーガイド」の「AWS API リクエストの署名」を参照してください。

Action

実行するアクション。

型: 文字列

必須: はい

Version

リクエストが想定している API バージョンである、YYYY-MM-DD 形式で表示されます。

型: 文字列

必須: はい

X-Amz-Algorithm

リクエストの署名を作成するのに使用したハッシュアルゴリズム。

条件: HTTP 認証ヘッダーではなくクエリ文字列に認証情報を含める場合は、このパラメータを指定します。

型: 文字列

有効な値: `AWS4-HMAC-SHA256`

必須: 条件による

X-Amz-Credential

認証情報スコープの値で、アクセスキー、日付、対象とするリージョン、リクエストしているサービス、および終了文字列 ("aws4_request") を含む文字列です。値は次の形式で表現されます。[access_key/YYYYYYYYMMDD/リージョン/サービス/aws4_request]

詳細については、「IAM ユーザーガイド」の「署名付きAWS API リクエストの作成」を参照してください。

条件: HTTP 認証ヘッダーではなくクエリ文字列に認証情報を含める場合は、このパラメータを指定します。

型: 文字列

必須: 条件による

X-Amz-Date

署名を作成するときに使用する日付です。形式は ISO 8601 基本形式の YYYYMMDD'T'HHMMSS'Z' でなければなりません。例えば、日付 `20120325T120000Z` は、有効な X-Amz-Date の値です。

条件: X-Amz-Date はすべてのリクエストに対してオプションです。署名リクエストで使用する日付よりも優先される日付として使用できます。ISO 8601 ベーシック形式で日付ヘッダーが指定されている場合、X-Amz-Date は必要ありません。X-Amz-Date を使用すると、常に Date ヘッダーの値よりも優先されます。詳細については、「IAM ユーザーガイド」の「AWS API リクエスト署名の要素」を参照してください。

タイプ: 文字列

必須: 条件による

X-Amz-Security-Token

AWS Security Token Service (AWS STS) へのを呼び出しで取得された一時的なセキュリティトークン。AWS STS の一時的なセキュリティ認証情報をサポートするサービスのリストについては、「IAM ユーザーガイド」の「IAM と連携するAWS のサービス」を参照してください。

条件:AWS STS の一時的なセキュリティ認証情報を使用する場合、セキュリティトークンを含める必要があります。

タイプ: 文字列

必須: 条件による

X-Amz-Signature

署名する文字列と派生署名キーから計算された 16 進符号化署名を指定します。

条件: HTTP 認証ヘッダーではなくクエリ文字列に認証情報を含める場合は、このパラメータを指定します。

型: 文字列

必須: 条件による

X-Amz-SignedHeaders

正規リクエストの一部として含まれていたすべての HTTP ヘッダーを指定します。署名付きヘッダーの指定に関する詳細については、「IAM ユーザーガイド」の「署名付き AWS API リクエストの作成」を参照してください。

条件: HTTP 認証ヘッダーではなくクエリ文字列に認証情報を含める場合は、このパラメータを指定します。

型: 文字列

必須: 条件による

AWS RoboMaker エンドポイントとクォータ

AWS RoboMaker のサービスエンドポイントおよび Service Quotas を以下に示します。AWS の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。標準の AWS エンドポイントに加えて、一部の AWS のサービスは選択されたリージョンで FIPS エンドポイントを提供します。詳細については、「AWS のサービスエンドポイント」を参照してください。

Service Quotas (制限とも呼ばれます) は、AWS アカウントのサービスリソースまたはオペレーションの最大数です。詳細については、「AWS のサービスクォータ」を参照してください。

サービスエンドポイント

リージョン名	リージョン	エンドポイント	プロトコル	
米国東部 (オハイオ)	us-east-2	robomaker.us-east-2.amazonaws.com	HTTPS	
米国東部 (バージニア北部)	us-east-1	robomaker.us-east-1.amazonaws.com	HTTPS	
米国西部 (オレゴン)	us-west-2	robomaker.us-west-2.amazonaws.com	HTTPS	
アジアパシフィック (シンガポール)	ap-southeast-1	robomaker.ap-southeast-1.amazonaws.com	HTTPS	
アジアパシフィック (東京)	ap-northeast-1	robomaker.ap-northeast-1.amazonaws.com	HTTPS	

リージョン名	リージョン	エンドポイント	プロトコル	
欧州 (フランクフルト)	eu-central-1	robomaker.eu-central-1.amazonaws.com	HTTPS	
欧州 (アイルランド)	eu-west-1	robomaker.eu-west-1.amazonaws.com	HTTPS	
AWS GovCloud (米国西部)	us-gov-west-1	robomaker.us-gov-west-1.amazonaws.com	HTTPS	

サービスクォータ

名前	デフォルト	引き上げ可能	説明
バッチのタイムアウト	サポートされている各リージョン: 14	No	シミュレーションジョブバッチの最大タイムアウト (日数)。
同時 GPU シミュレーションジョブ	サポートされている各リージョン: 1	はい	現在のリージョンにおける、このアカウントで実行できる同時 GPU シミュレーションジョブの最大数。
ワールド同時エクスポートジョブ	サポートされている各リージョン: 3	はい	現在のリージョンおける、このアカウントで実

名前	デフォルト	引き上げ可能	説明
			行できるワールド同時エクスポートジョブの最大数。
ワールド同時生成ジョブ	サポートされている各リージョン: 3	はい	このリージョンにおける、このアカウントで実行できるワールド同時生成ジョブの最大数。
同時デプロイジョブ	サポートされている各リージョン: 20	はい	このアカウントで現在のリージョンで実行できる同時デプロイジョブの最大数。
同時シミュレーションジョブバッチ	サポートされている各リージョン: 5	はい	このアカウントで現在のリージョンにおいて実行できる同時シミュレーションジョブバッチの最大数。
同時シミュレーションジョブ	サポートされている各リージョン: 1	はい	このアカウントで現在のリージョンで実行できる同時シミュレーションジョブの最大数。
フリート	サポートされている各リージョン: 20	はい	このアカウントで現在のリージョンに作成できるフリートの最大数。

名前	デフォルト	引き上げ可能	説明
1 分あたりの GPU シミュレーションジョブの作成レート	サポートされている各リージョン: 2	No	現在のリージョンにおける、このアカウントで作成できる GPU シミュレーションジョブの最大数 (1 分あたり)。
バッチの最小タイムアウト	サポートされている各リージョン: 5	No	シミュレーションジョブバッチに指定できる最小タイムアウト (分数)。
最小シミュレーション期間	サポートされている各リージョン: 5	No	シミュレーションジョブ用に指定できる最小期間 (分)。
ロボットアプリケーション	サポートされている各リージョン: 40	はい	このアカウントで現在のリージョンに作成できるロボットアプリケーションの最大数。
ロボット	サポートされている各リージョン: 100	はい	このアカウントで現在のリージョンに作成できるロボットの最大数。
フリートあたりのロボット	サポートされている各リージョン: 100	はい	フリートに登録できるロボットの最大数。

名前	デフォルト	引き上げ可能	説明
1 分あたりのシミュレーションジョブの作成レート	us-east-1: 10 us-west-2: 10 他のサポートされている各リージョン: 5	No	現在のリージョンにおける、このアカウントで作成できるシミュレーションジョブの最大数 (1 分あたり)。
シミュレーションアプリケーション	サポートされている各リージョン: 40	はい	このアカウントで現在のリージョンに作成できるシミュレーションアプリケーションの最大数。
シミュレーション期間	サポートされている各リージョン: 14	No	シミュレーションジョブで実行 (再起動など) できる最大期間 (日)。
バッチごとのシミュレーションジョブリクエスト	サポートされている各リージョン: 20	はい	StartSimulationJobBatch 呼び出しで送信できるシミュレーションジョブリクエストの最大数
ソースのサイズ	サポートされている各リージョン: 5 GB	No	ロボットアプリケーションまたはシミュレーションアプリケーションの任意のソースの最大サイズ (GB)。
ロボットアプリケーションあたりのバージョン	サポートされている各リージョン: 40	はい	ロボットアプリケーション用に作成できるバージョンの最大数。

名前	デフォルト	引き上げ可能	説明
シミュレーションアプリケーションあたりのバージョン	サポートされている各リージョン: 40	はい	シミュレーションアプリケーション用に作成できるバージョンの最大数。
アカウントあたりのワールドテンプレート	サポートされている各リージョン: 40	はい	このリージョンにおける、このアカウントで作成できるワールドテンプレートの最大数。
エクスポートジョブあたりのワールド	サポートされている各リージョン: 1	No	ワールドエクスポートジョブリクエストに含まれるワールドの最大数。
ジェネレーションジョブあたりのワールド	サポートされている各リージョン: 50	No	1 つのワールド生成ジョブリクエストに含まれるワールドの最大数。

AWS RoboMaker のトラブルシューティング

以下のセクションでは、AWS RoboMaker のシミュレーション、IDE、Simulation WorldForge の使用時に発生する可能性のあるエラーや問題のトラブルシューティングに関するアドバイスを提供します。ここに記載されていない問題が見つかった場合は、このページの下部で [フィードバックを提供します] ボタンを使用して報告することができます。

シミュレーションジョブのログは、CloudWatch Logs コンソールで確認できます。デフォルトでは、AWS RoboMaker はアプリケーション用に生成されたシミュレーションジョブのログをアップロードします。`streamOutputToCloudWatch` を `True` に設定すると、ツールにも同じ動作が適用されます。シミュレーションジョブで カスタムアップロード設定の追加 がアップロードするよう設定することもできます。

詳細については、「AWS RoboMaker でのログ記録とモニタリング」を参照してください。

セクション

- シミュレーションジョブ
- Simulation WorldForge

シミュレーションジョブ

問題：シミュレーションジョブが失敗しました。

以下の質問を参考にして根本原因を特定し、推奨アクションを実行してください。

Amazon S3 リソースは AWS RoboMaker と同じリージョンにありますか?

ロボットアプリケーション、シミュレーションアプリケーション、および出力は、AWS RoboMaker と同じリージョンにあることが必要です。アプリケーションのソースとシミュレーションジョブの出力の場所を確認してください。

ロボットアプリケーションは異常終了しましたか?

シミュレーション用にロボットアプリケーションをセットアップするときに問題がありました。Amazon CloudWatch でシミュレーションジョブのロボットアプリケーションを確認してください。

ログにはシミュレーションジョブの詳細画面からアクセスします。[Logs] (ログ) を選択し、ログストリームを選択します。特定の問題を見つけるには、フィルターを使用します。例えば、WARNING または ERROR を使用します。

アプリケーションの .so ファイルが欠落していませんか?

アプリケーションがクラッシュした場合、依存する共有オブジェクト (.so) ファイルが欠落している場合があります。環境のアプリケーションバンドルを抽出し、必要な共有オブジェクトライブラリが /usr/local/lib または /usr/lib に存在することを確認してください。依存関係がパッケージ .xml ファイルに追加されていることを確認します。

AWS CLI でロールの ARN を使用しましたか?

AWS CLI から create-simulation-job を呼び出す場合は、ロール名だけでなく、ロールの完全な Amazon リソースネーム (ARN) を使用します。

ロールには AWS RoboMaker の信頼ポリシーがありますか?

AWS CLI から create-simulation-job を呼び出して IAM ロールの完全な Amazon リソースネーム (ARN) を渡す場合、信頼ポリシーの特権が十分ではない可能性があります。以下のように、ロールに robomaker.amazonaws.com と信頼関係があることを確認します。

```
{"Version": "2012-10-17",
   "Statement": {"Effect": "Allow",
       "Principal": { "Service": "robomaker.amazonaws.com" },
       "Action": "sts:AssumeRole",
       "Condition": {"StringEquals": {"aws:SourceAccount": "account#" // Account where
 the simulation job resource is created
           },
           "StringEquals": {"aws:SourceArn":
 "arn:aws:robomaker:region:account#:simulation-job/*"
           }
       }
   }
}
```

条件キーは、サービス間のトランザクション中に AWS サービスが混乱した代理として使用されるのを防ぐことができます。条件キーに関する詳しい情報については、「SourceAccount」と「SourceArn」を参照してください。

ロールアクセスの表示と IAM ロールへの信頼ポリシーの追加の詳細については、「ロールの修正」を参照してください。

Amazon S3 に発行するためのアクセス許可がロールにありますか?

シミュレーションジョブの出力 Amazon S3 バケットを指定する場合、ロールにはバケットへの書き込みアクセス許可が必要です。信頼ポリシーを更新して書き込みアクセス許可を含めてください。以下の信頼ポリシーの例は、読み取りアクセス許可、リストアクセス許可、および Amazon S3 バケットへの書き込みアクセス許可を追加します。

```
{"Action": "s3:ListBucket",
   "Resource": [
       "my-bucket/*"
   ],
   "Effect": "Allow"
},{"Action": [
       "s3:Get*",
       "s3:List*"
   ],
   "Resource": [
       "my-bucket/*"
   ],
   "Effect": "Allow"
},{"Action": "s3:Put*",
   "Resource": [
       "my-bucket/*"
   ],
   "Effect": "Allow"
}
```

ロールには、CloudWatch に発行するアクセス許可がありますか?

IAM ロールのアクセス許可ポリシーを更新して CloudWatch へのアクセスを含めてください。

```
{"Effect": "Allow",
  "Action": [
      "logs:CreateLogGroup",
      "logs:CreateLogStream",
      "logs:PutLogEvents",
      "logs:DescribeLogStreams"
   ],
   "Resource": "*"
}
```

アプリケーションに不一致のエンティティタグがありませんか?

エンティティタグ (ETag) は、シミュレーションの作成時に提供される Amazon S3 オブジェクトのハッシュです。ETag は、オブジェクトのコンテンツに加えた変更のみを反映し、メタデータに加えた変更は反映しません。ロボットアプリケーションやシミュレーションバンドルの内容を、AWS RoboMaker で使用する前に Amazon S3 で変更すると、バージョンの不一致が発生します。

この問題を解決するには、ロボットアプリケーションやシミュレーションアプリケーションの新しいバージョンを作成し、更新したアプリケーションバンドルのキーの場所を指定します。詳細については、「アプリケーションバージョンの作成」または「シミュレーションアプリケーションバージョンの作成」を参照してください。

サブネットの Elastic Network Interface (ENI) の制限を超えていますか?

AWS RoboMaker は、シミュレーションジョブの実行先であるサブネットの同時実行シミュレーションジョブごとに 1 つの ENI を使用します。これらの ENI ごとに IP アドレスを割り当てる必要があります。この問題を解決する方法は以下のとおりです。

未使用の ENI を削除し、サブネットの IP アドレスを解放します。未使用の ENI を削除するには、「ネットワークインターフェイスの削除」を参照してください。

AWS Management Console を使用して、特定の AWS リージョンの ENI に対してサービス制限の引き上げをリクエストします。

起動コマンドは適切に設定されていますか?

シミュレーションが複雑な場合やコンテナイメージが大きい場合は、シミュレーションジョブの起動に数分かかることがあります。AWS RoboMaker でシミュレーションジョブの準備に 25 分以上かかる場合は、起動コマンドに問題がある可能性があります。ジョブをキャンセルして新しいシミュレーションジョブを作成します。問題が解決しない場合は、AWS サポートまでお問い合わせください。

CloudWatch Logs を使用すると、シミュレーションとロボットアプリケーションの実行ログにエラーがないかどうかを確認できます。ターミナルのカスタマイズされたツールを追加して、実行中のシミュレーションジョブに接続し、トラブルシューティングを行うこともできます。

サブネットは、AWS RoboMaker でサポートされているゾーンにありますか?

AWS RoboMaker でサポートされている AWS アベイラビリティーゾーンのうちの 2 つでサブネットを指定してください。サポートされている AWS アベイラビリティーゾーンのリストは、API レスポンスに記載されています。

ワールドファイルのモデル参照は正しいですか?

CloudWatch Logs を使用して、ワールドファイルのすべてのモデルが正しいことを確認します。モデルが見つからない場合は、以下のエラーが表示されます。

```
[Wrn] [ModelDatabase.cc:340] Getting models from[http://models.gazebosim.org/]. This
 may take a few seconds.
[Wrn] [ModelDatabase.cc:212] Unable to connect to model database using [http://
models.gazebosim.org//database.config]. Only locally installed models will be
 available.
[Err] [ModelDatabase.cc:414] Unable to download model[model://model_name]
[Err] [SystemPaths.cc:429] File or path does not exist[""]
Error [parser.cc:581] Unable to find uri[model://model_name]
```

Simulation WorldForge

問題：ワールド生成ジョブが失敗しました。

ワールド生成ジョブが完了しなかった場合は、ワールドカウント `floorplanCount * interiorCountPerFloorplan` が 1 より大きく 50 未満であるか確認してください。

問題：ワールドエクスポートジョブが失敗したのはなぜですか？

以下の質問を参考にして根本原因を特定し、推奨アクションを実行してください。

AWS RoboMaker の信頼ポリシーがありますか?

AWS CLI から `create-world-export-job` を呼び出して IAM ロールの完全な Amazon リソースネーム (ARN) を渡す場合、信頼ポリシーの特権が十分ではない可能性があります。以下のように、ロールに `robomaker.amazonaws.com` と信頼関係があることを確認します。

```
{"Version": "2012-10-17",
   "Statement": {"Effect": "Allow",
       "Principal": { "Service": "robomaker.amazonaws.com" },
       "Action": "sts:AssumeRole",
       "Condition": {"StringEquals": {"aws:SourceAccount": "account#" // Account where
 the simulation job resource is created
           },
           "StringEquals": {"aws:SourceArn":
 "arn:aws:robomaker:region:account#:simulation-job/*"
           }
       }
   }
```

```
}
```

条件キーは、サービス間のトランザクション中に AWS サービスが混乱した代理として使用されるのを防ぐことができます。条件キーに関する詳しい情報については、「SourceAccount」と「SourceArn」を参照してください。

Amazon S3 に発行するためのアクセス許可がロールにありますか?

エクスポートジョブの出力 Amazon S3 バケットを指定する場合、ロールにはバケットへのアクセス許可が必要です。信頼ポリシーを更新してアクセス許可を含めてください。

```
{"Effect": "Allow",
 "Action": [
   "s3:AbortMultipartUpload",
   "s3:GetObject",
   "s3:PutObject"
 ],
 "Resource": "my-bucket"
}
```

エクスポートジョブに対して指定されているバケットを変更または削除しましたか?

エクスポートジョブ中にバケットを更新すると、エクスポートジョブの ResourceNotFound エラーが発生することがあります。

問題：ワールドイメージに問題があります。

以下の質問を参考にして根本原因を特定し、推奨アクションを実行してください。

出入口にドアがないのはなぜですか？

ドアは、バージョン 2 以降のテンプレートを使用している場合しか追加できません。バージョン 1 テンプレートを新しいバージョンに更新できます。詳細については、「シミュレーションワールドテンプレートのバージョン、機能、および変更」を参照してください。

AWS RoboMaker Simulation WorldForge では一意でランダムなワールドが作成されるため、指定したドア設定は生成時にワールドに存在しない可能性があります。例えば、テンプレートでリビングルームとキッチンの間にドアを指定し、それらの部屋の間に開放壁があるとします。出入口の代わりに開放壁があるので、そこにはドアを追加できないはずです。

ドアが部屋の入口を塞いでるのはなぜですか？

部屋の入口を塞ぐドアは、ロボットへのチャレンジに使用できる状況です。このチャレンジをロボットに提示しないワールドを作成するには、以下の作業のいずれかを行ってください。

ワールドテンプレートから別のワールドを生成する。新しいワールドで生成されたドアは入口を塞がないかもしれません。

ワールドテンプレート内のドアの開放率を変更する。

ワールドイメージ内の壁が、シミュレーションジョブ内またはエクスポートされたワールド内の壁よりも短いのはなぜですか?

Simulation WorldForge ワールドを壁に隠れることなく表示できるようにするために、AWS RoboMaker ではワールドイメージ内の壁が切り取られます。壁の高さは、作成したワールドのワールドテンプレートで指定した高さになります。

バージョン 2 以降のテンプレートで生成されたワールドでは、ワールドイメージにおいてドアモデルが切り取られません。ワールドイメージ内のドアの高さは、作成したワールドのドアの高さと同じになります。

サポートポリシー

以下のセクションでは、のサポート変更について説明します AWS RoboMaker。

サポートの変更: 2022 年 12 月 15 日

2022 年 6 月 27 日に、 AWS Cloud9 開発環境機能を開発環境機能に移行し、 AWS RoboMaker サポートを終了しました。2022 年 12 月 15 日以降、 AWS RoboMakerで以前に作成した開発環境にはアクセスできません。

なぜサポートを終了したのですか？

AWS Cloud9 新しい機能、柔軟性、およびリージョンサポートの拡張により、開発エクスペリエンスが向上します。 AWS Cloud9では、Amazon Linux と Ubuntu のプラットフォームオプション、コスト削減設定、任意のロボットやシミュレーションソフトウェアを柔軟に使用および構成できます。AWS Cloud9の使用開始に関する詳細については、「AWS Cloud9 ユーザーガイド」を参照してください。

の新しい開発環境 AWS Cloud9

AWS Cloud9 以前と同じ開発環境機能にアクセスできます AWS RoboMaker。 AWS Cloud9 コンソールを使用して開発環境を作成し、新しい機能を活用してください。 AWS Cloud9 開発環境を設定してロボットとシミュレーションアプリケーションを構築およびシミュレートする方法については、AWS Robotics ブログの「ロボットアプリケーションの構築とシミュレーション」を参照してください。 AWS Cloud9

AWS RoboMaker 既存の開発環境

2022 年 6 月 27 日より前に起動された環境には、 AWS Cloud9 コンソールからアクセスできます。NICE DCV 機能を保持するには、「 AWS Cloud9での口ボットアプリケーションの構築とシミュレート」の「NICE DCV の設定」をご確認ください。

サポートの変更: 2022 年 5 月 2 日

2022 年 5 月 2 日、既存の `robots`、`fleets`、`deployments` のジョブリソースをアカウントから削除しました。これらの AWS RoboMaker アプリケーションデプロイリソースを削除しても、物理ハードウェアには影響しません。 AWS IoT Greengrass Version 2などの他の方法でも、引き続きアプリケーションを物理口ボットにデプロイできます。

以下の API アクションは非推奨になりました。

- CancelDeploymentJob
- DeleteFleet
- DeleteRobot
- DeregisterRobot
- DescribeDeploymentJob
- DescribeFleet
- DescribeRobot
- ListDeploymentJobs
- ListFleets
- ListRobots
- SyncDeploymentJob

推奨されるアクション

次のアクションを実行することをお勧めします。

1. まだ実施していない場合は、ロボットとシミュレーションアプリケーションをサポートされているコンテナイメージに移行します。ロボットアプリケーション、シミュレーションアプリケーション、シミュレーションジョブを Docker ベースのワークフローに移行する方法については、「ROS アプリケーションのコンテナへの移行」を参照してください。
2. に移行 AWS IoT Greengrass Version 2。詳細については、「AWS IoT Greengrass Version 2 デベロッパーガイド」を参照してください。 AWS IoT Greengrass Version 2 デプロイメントについて詳しくは、 AWS Robotics に関する以下のブログ記事を参照してください。
   - Docker による ROS AWS IoT Greengrass Version 2 ロボットのデプロイと管理
   - を使用して ROS アプリケーションをスナップとしてデプロイ AWS IoT Greengrass Version 2

サポートの変更: 2022 年 3 月 15 日

2022 年 3 月 15 日に、シミュレーションジョブに影響が及ぶ可能性のある 2 AWS RoboMaker つの変更がシミュレーションに適用されました。

1. AWS RoboMaker シミュレーションジョブをコンテナイメージに移行しました。つまり、ロボットとシミュレーションアプリケーションをサポートされているコンテナイメージに移行する必要があります。
2. シミュレーションでは、プリインストールされたロボットオペレーティングソフトウェア (ROS)、Gazebo、Ubuntu のベースイメージの販売を停止しました。 AWS RoboMaker ROS と Gazebo ベースのシミュレーションは引き続き実行できますが、そのメカニズムは変わります。ロボットアプリケーションは一般的なソフトウェアスイートへ、シミュレーションアプリケーションはシミュレーションランタイムのソフトウェアスイートへ更新してください。

なぜこのような変更を加えたのでしょうか？

AWS RoboMaker あらゆるロボットとシミュレーションソフトウェアの拡張構成をサポートするようになったため、シミュレーションの実行中に任意のロボットとシミュレーションソフトウェアを使用および構成できます。ROS Kinetic などの旧バージョンや ROS2 Galactic などの新バージョンを含め、ロボットアプリケーションでは引き続き ROS を使用することができますが、ROS を使用しなくてもカスタムのロボットアプリケーションを実行することができます。さらに、 AWS RoboMaker でシミュレーションを実行しながら、任意のシミュレーションソフトウェアを使用できるようになりました。

これからどうなるのか？

移行されていない既存のロボットおよびシミュレーションアプリケーションの使用は制限されますが、サポートされているソフトウェアスイートとコンテナイメージに移行することはできます。2022 年 3 月 15 日より前に開始され、期間が 2022 年 3 月 15 日を超えるシミュレーションジョブとシミュレーションジョブバッチは、引き続き完了まで実行されます。

2022 年 1 月 31 日にサポートが終了

2022 年 1 月 31 日に、 AWS IoT Greengrass Version 2 AWS RoboMaker デプロイ機能をアプリケーションデプロイ機能に移行し、サポートを終了しました。 AWS IoT Greengrass Version 2 既存のアプリケーションデプロイ機能をサポートし、新機能とデプロイエクスペリエンスの向上を提供します。2022 年 1 月 31 日以降、 AWS RoboMakerでは新しいアプリケーションデプロイリソース (`robots`、`fleets`、`deployments`) を作成できなくなります。

以下の API アクションは非推奨になりました。

- `CreateDeploymentJob`
- `CreateFleet`

- `CreateRobot`
- `RegisterRobot`

2021 年 4 月 30 日にサポートが終了

2021 年 4 月 30 日以降、 AWS RoboMakerでは新しい ROS Kinetic、Gazebo 7.1、ROS Dashingまたは Ubuntu 16.04 リソースを作成できなくなりま、す。ただし、 AWS RoboMaker 既存のリソースはアカウントに残ります。アップグレードしないと、機能内の ROS Kinetic、Gazebo 7.1、ROS Dash、および Ubuntu 16.04 リソースの機能が変更されたり、機能しなくなったりする可能性があります。 AWS RoboMaker

次のソフトウェアスイートの組み合わせは廃止されます。

- ROS Kinetic、Gazebo 7.1、Ubuntu 16.04
- ROS Kinetic、Gazebo 9、Ubuntu 16.04
- ROS Dashing、Gazebo 9、Ubuntu 16.04

非推奨の影響が及ぶ分野は以下の通りです。

- AWS Cloud9 統合開発環境 (IDE)
  - ROS Kinetic と ROS Dashing をベースとする既存のすべての IDE にアクセスできます。引き続き IDE 内での作業が可能です。ビルドとバンドルのプロセスの正常な実行は保証されません。
  - ROS Kinetic と ROS Dashing をベースとする IDE を新たに作成することはできません。
- ロボットアプリケーションとシミュレーションアプリケーション
  - ROS Kinetic と ROS Dashing をベースとするロボットアプリケーションを新たに作成することはできなくなります。
  - ROS Kinetic と Gazebo 7.1、ROS Kinetic と Gazebo 9、または ROS Dashing と Gazebo 9 を使用してシミュレーションアプリケーションを新たに作成することはできなくなります。
  - 非推奨の ROS バージョンと Gazebo バージョンを使用して、既存のロボットアプリケーションまたはシミュレーションアプリケーションの新しいバージョンを作成することはできなくなります。
- シミュレーションジョブとシミュレーションバッチ
  - Kinetic、Dashing、または Gazebo 7.1 を使用して、ロボットアプリケーションおよびシミュレーションアプリケーションで新しいシミュレーションジョブを作成することはできなくなります。

非推奨日より前に起動され、その実行期間が非推奨日を過ぎたシミュレーションジョブは、完了するまで正常に実行されます。シミュレーションジョブの最長期間が 14 日間であれば、これらのジョブは非推奨後から最長で 14 日間実行できます。

- デプロイジョブ
  - Kinetic または Dashing をベースとするロボットアプリケーションのデプロイジョブを作成することはできなくなります。
- サンプルアプリケーションとクラウド拡張機能
  - ROS Kinetic および ROS Dashing をベースとするアプリケーションでは、クラウド拡張機能はサポートされなくなります。クラウド拡張は ROS Kinetic と ROS Dashing のワークスペースにインストールできますが、動作する場合と動作しない場合があります。
  - サンプルアプリケーションを起動するために ROS ディストリビューションとして ROS Kinetic も ROS Dashing も選択できなくなる可能性があります。サンプルアプリケーションは、既存の ROS Kinetic IDE と ROS Dashing IDE に引き続きダウンロードできます。ただし、これらはサポートされなくなったため、破損する可能性があります。

ドキュメント履歴

次の表には、AWS RoboMaker のサービスとドキュメントに対し、機能と廃止がいつ適用されたのかが記載されています。

変更	説明	日付
AWS RoboMaker シミュレーション用のパブリック ECR	パブリック ECR イメージの使用がサポートされるようになりました。プライベート ECR リポジトリを作成せずに、AWS RoboMaker のロボットとシミュレーションアプリケーションを作成できます。	2023 年 1 月 26 日
IDE の廃止	AWS RoboMaker IDE を廃止しました	2022 年 12 月 15 日
RUG のプリインストール廃止	ロボットオペレーティングソフトウェア (ROS)、Ubuntu、Gazebo のベースイメージのプリインストールを廃止し、AWS RoboMaker のシミュレーションジョブをコンテナイメージに移行しました。	2022 年 3 月 15 日
アプリケーションデプロイの廃止	AWS RoboMaker のアプリケーションデプロイを廃止しました。	2022 年 1 月 31 日
クラウド拡張の廃止	AWS RoboMaker 向けのクラウド拡張を廃止しました。	2022 年 1 月 31 日
サンプルの廃止	AWS RoboMaker 用の自動運転補強、ナビゲーション、人	2020 年 5 月 15 日

変更	説明	日付
	物検出、音声コマンドサンプルを廃止しました。	
タグのサポート	多くの AWS RoboMaker リソースにタグのサポートを追加しました。	1/24/2019
新しいサービスとガイド	これは AWS RoboMaker の最初のリリースおよび AWS RoboMaker 開発者ガイドです。	2018 年 11 月 27 日

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。