

AWS Robot Delivery Challenge 環境構築について

目次

ロボットを動かしてみよう

- 全体の流れ
- AWS Robomakerの準備
- ロボットコントローラーの用意
- アプリケーション環境の構築
- 地図作成とナビゲーション

予選リーグへの戦略を立てよう

- 予選リーグ測定について
- アプリケーションコードの更新と実行

各ツールの操作のコツとチューニングポイント

- Gazeboについて
- Rvizについて
- 地図作りのコツ
- マップ作成のチューニングポイント
- ナビゲーションのチューニングポイント

ロボットを動かしてみよう

全体の流れ

ロボットをシミュレーション上で動かせるようになるまでの行程は大きく分けて4つあります。まずは、AWSマネジメントコンソールからログインしてAWSRoboMaker用の開発環境を整えます。

Robotics

AWS RoboMaker

インテリジェントなロボット アプリケーションを容易に開 発、テスト、デプロイしま す。

AWS RoboMaker は、開発者が容易にロボットアプリケーションを開発、テスト、デプロイできるほか、AWS サービスを使用してインテリジェントなロボット機能を構築することができる新しいサービスです。

AWS RoboMaker の使用開始

すぐに簡単に使用できるサンプルアプリケーションによる、AWS RoboMaker を試してください。サンプルアプリケーションシミュレーションジョブを実行し、稼働中であることを確かめます。RoboMaker 開発環境を開き、コードをカスタマイズして編集します。

[サンプルアプリケーションを試す](#)

料金 (米国)

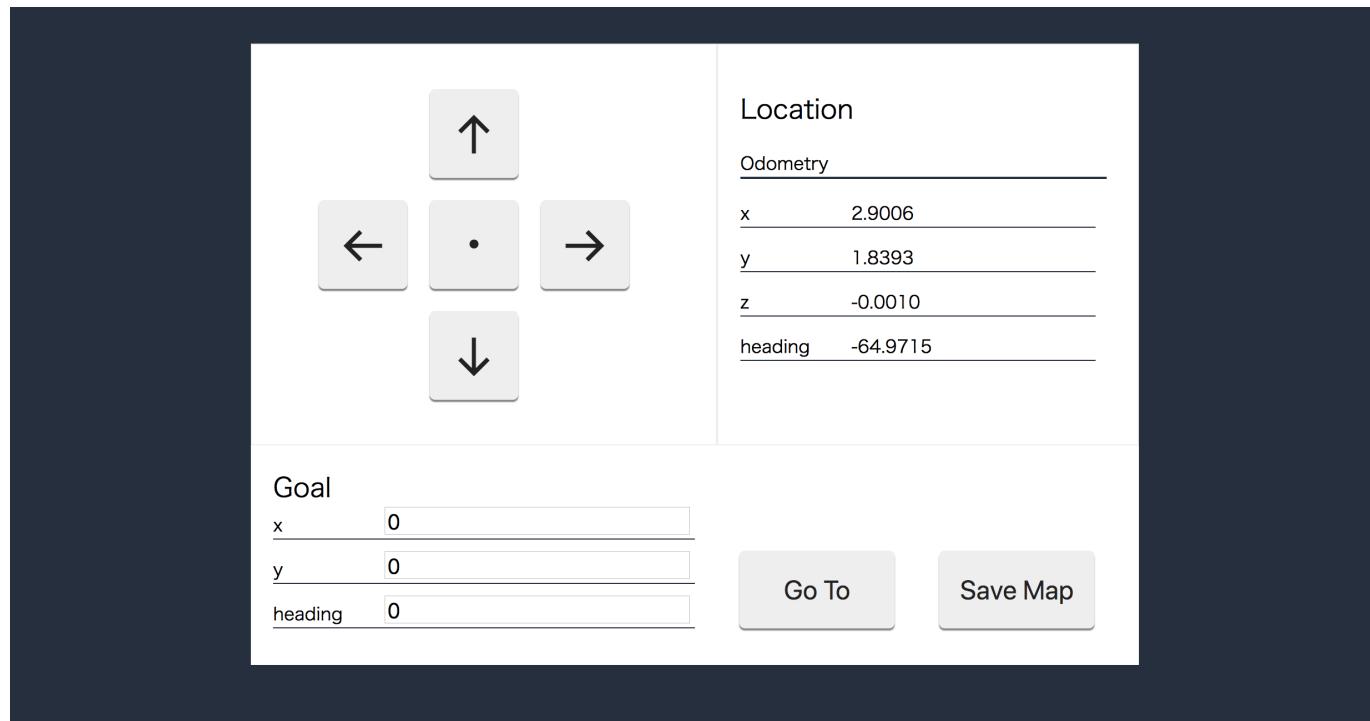
AWS RoboMaker では、最低料金ではなく、実際に使用した分のみお支払いいただけます。ROS 用 RoboMaker クラウド拡張機能、RoboMaker 開発環境、RoboMaker シミュレーション、および RoboMaker フリート管理を含む、各 RoboMaker 機能につき個別に料金が請求されます。

仕組み

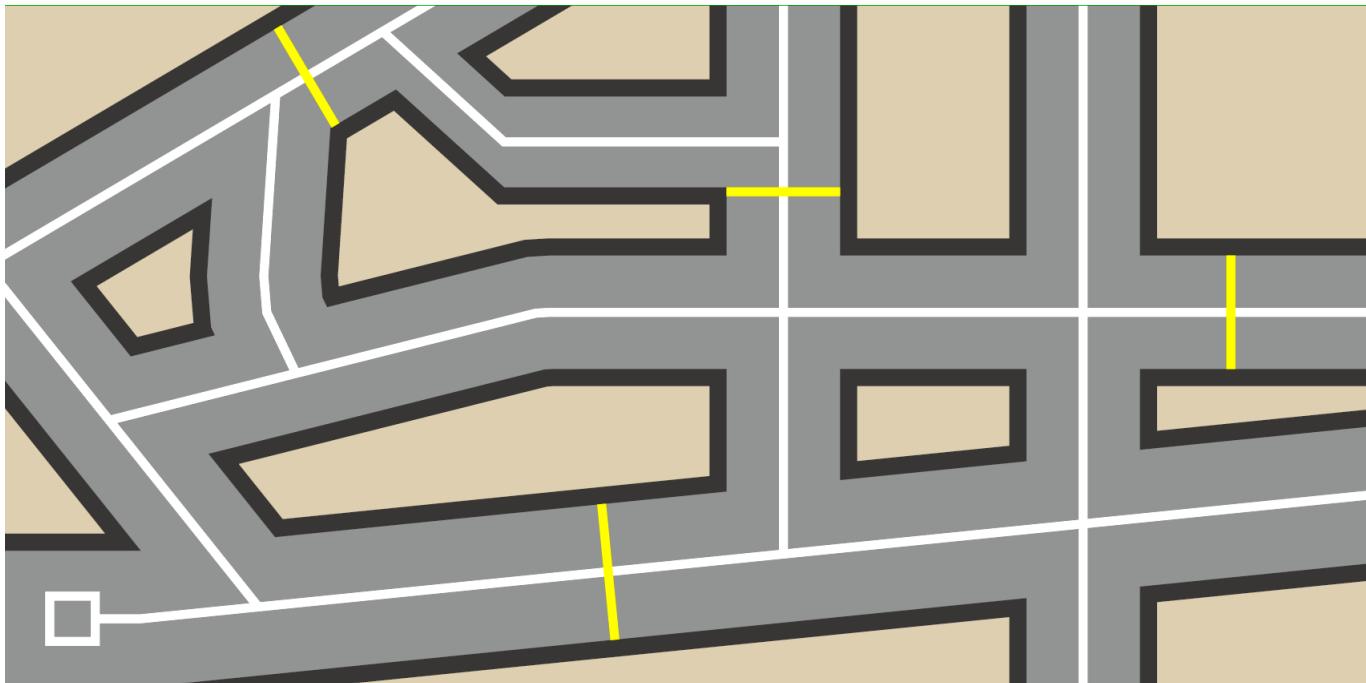


フィードバック 日本語 © 2008 - 2020, Amazon Web Services, Inc. またはその関連会社。無断転用禁止。 プライバシーポリシー 利用規約

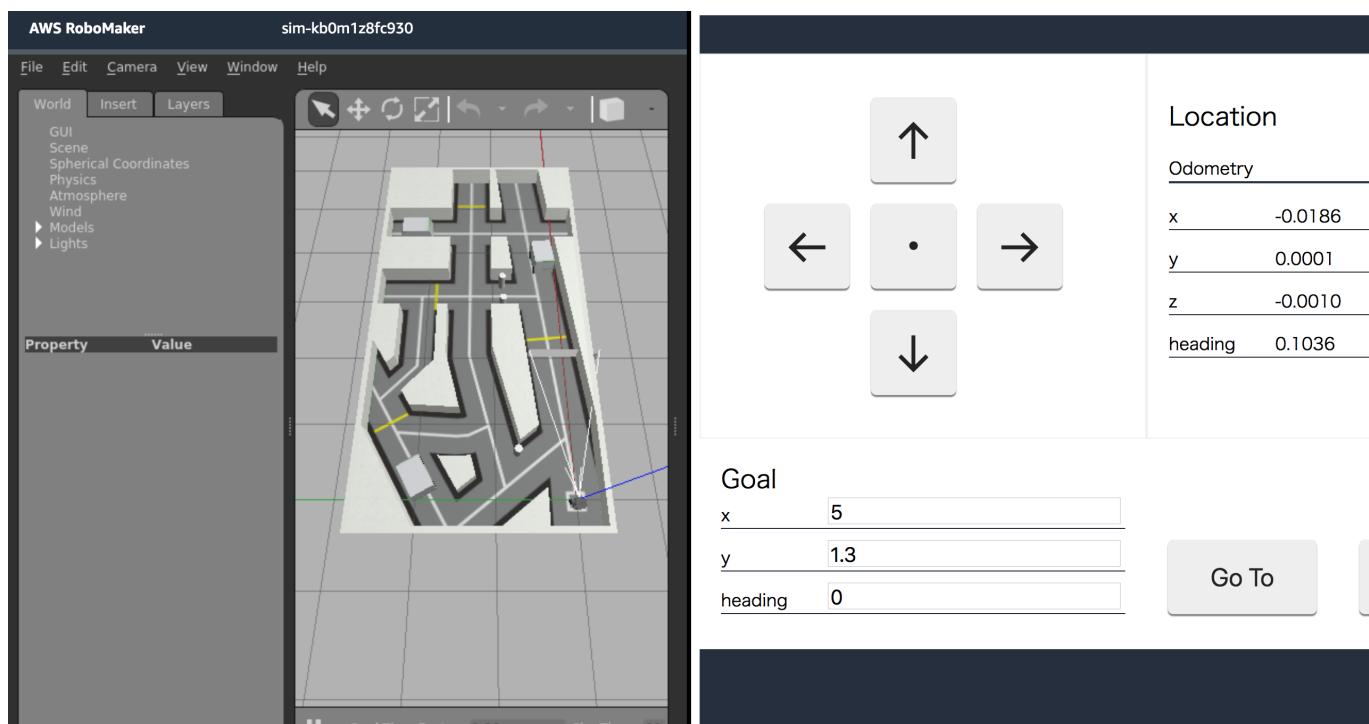
次に下の写真のようなコントローラをブラウザで表示し、操作する用意をします。



そして最初の開発環境内で数行コードを実行して（こちらのコードは用意しております）アプリケーションの構築をします。最後に下の写真のような道をシミュレーション環境内でロボットに走らせてコントローラーを使って地図の作成と自動運転のテストをする、という流れです。



予選では上の地図の中の左下のスタート地点から右の黄色いラインまで行くタイムを競います。



これから一つずつ説明していきます。

AWS RoboMakerの準備

1. AWSマネジメントコンソールを開き、AWS RoboMakerサービスを開きます。

Welcome to AWS Educate Starter Account

Your Starter Account Status

- Active** full access
- \$75 credits (estimated)
- 364d 23:59:32 remaining term
- 0:60 session time

[Account Details](#) **AWS Console**

What AWS services can I use in my Starter Account?

What regions can I use with a Starter Account?

Are Service Linked Roles supported?

I can't start any resources. What happened?

AWS マネジメントコンソール

AWS のサービス

サービスを検索する
名称、キーワード、頭文字を入力できます。

例: Relational Database Service、データベース、RDS

▼ 最近アクセスしたサービス

- AWS RoboMaker
- CloudWatch
- CloudFormation
- Cloud9
- S3

▶ すべてのサービス

ソリューションの構築
シンプルなウィザードと自動化されたワークフローで作業を開始しましょう。

仮想マシンを起動する ウェブアプリを構築する

外出先でリソースにアクセスする

AWS コンソールモバイルアプリを使用してマネジメントコンソールにアクセスします。
詳細は[こちらから](#)

AWS を試す

Amazon Redshift
データレイクにクエリを拡張できる、高速かつシンプルで、費用対効果の高いデータウェアハウス。
詳細は[こちらから](#)

AWS Fargate を使ってサーバーレスコンテナを実行
AWS Fargate が実行され、サーバーやクラスターを管理することなく、コンテナをスケーリングします。
詳細は[こちらから](#)

Amazon S3 によるスケーラブルで耐久性に優れたセキュアなバックアップと復元

2. 左のナビゲーションペインから 開発環境 を選び [環境の作成] ボタンを押して開発環境を作成を開始します。

The screenshot shows the AWS RoboMaker website. On the left, there's a navigation sidebar with sections like '開発' (Development), 'シミュレーション' (Simulation), 'フリートの管理' (Fleet Management), and 'リソース' (Resources). The main content area features a large banner with the text 'AWS RoboMaker インテリジェントなロボットアプリケーションを容易に開発、テスト、デプロイします。' (AWS RoboMaker makes it easy to develop, test, and deploy intelligent robot applications). Below the banner, there's a section titled '仕組み' (Architecture) with two diagrams: one of a robot head and another of a cloud connected to a computer. A note says 'AWS RoboMaker は、開発者が簡単にロボットアプリケーションを開発、テスト、デプロイできるほか、AWS サービスを使用してインテリジェントなロボット機能を構築することができる新しいサービスです。' (AWS RoboMaker makes it easy for developers to create intelligent robot applications by developing, testing, and deploying them directly from the AWS Cloud using familiar tools and services like ROS and AWS Lambda). To the right, there's a box titled 'AWS RoboMaker の使用開始' (Getting Started with AWS RoboMaker) with a button 'サンプルアプリケーションを試す' (Try a Sample Application).

3. 開かれた「AWS RoboMaker 開発環境を作成する」ウインドウではプリインストールされたROSディストリビューションはKineticかMelodicを選びます。インスタンスタイプはデフォルトのm4.largeを選択、VPCはリストから(デフォルト)と表示されているものを選びサブネットは任意の1つを選びます。(選択するサブネットはパブリックサブネットである必要があります)

This screenshot shows the first step of the 'Create Development Environment' wizard. The title is 'AWS RoboMaker 開発環境を作成する'. The '全般' (General) tab is selected. Under '名前' (Name), the name 'roboton-summit' is entered. Under 'プリインストールされた ROS ディストリビューション' (Pre-installed ROS Distribution), 'ROS Melodic' is selected. Under 'インスタンスタイプ' (Instance Type), 'm4.large' is chosen. Under 'IAM ロール' (IAM Role), 'AWSRoboMakerServiceRoleForAWSCloud9' is listed. At the bottom, there are tabs for 'ネットワーク' (Networking) and 'VPC info'.

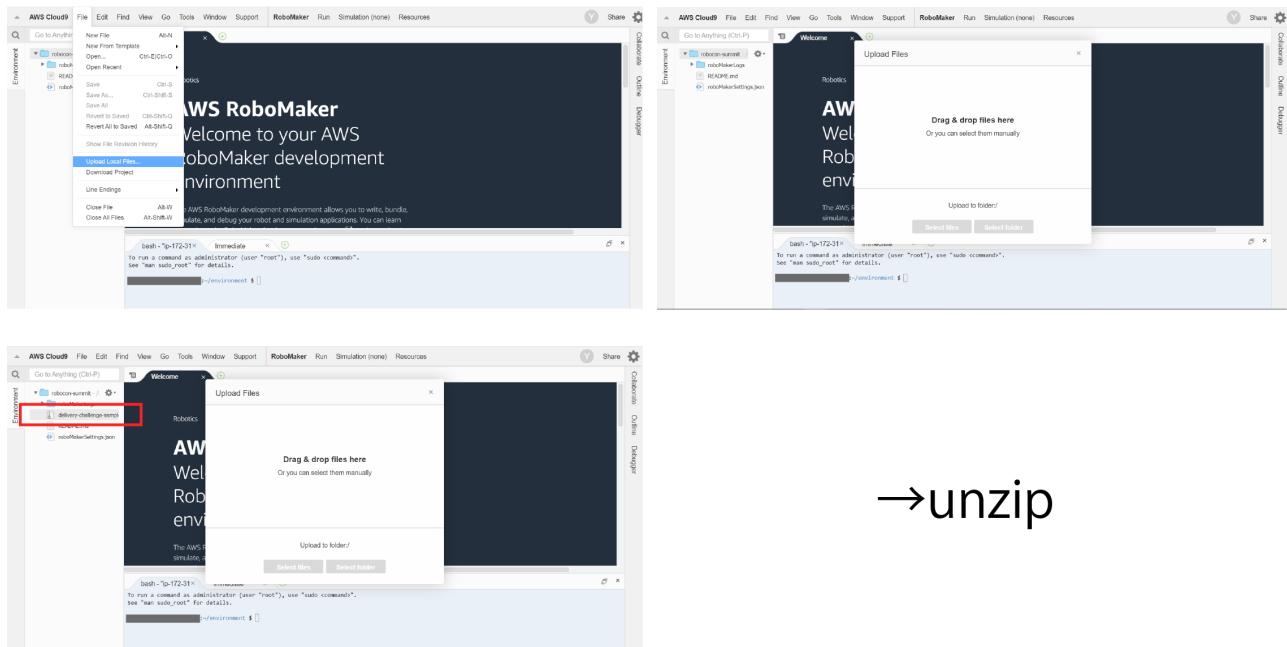
This screenshot shows the second step of the wizard, 'AWS RoboMaker 開発環境を作成する'. The 'ネットワーク' (Networking) tab is selected. It shows a dropdown for 'VPC info' containing 'vpc-eec343389 (デフォルト)' and a dropdown for 'サブネット' (Subnet) containing 'subnet-f87745a3'. A note at the bottom explains that AWS Cloud9環境を使用する場合は、次のベストプラクティスをお勧めします。

This screenshot shows the AWS Cloud9 IDE interface. A terminal window displays the message 'Welcome to your AWS RoboMaker development environment'. Below it, a message says 'We are creating your AWS Cloud9 environment. This can take a few minutes.' The AWS Cloud9 logo is visible in the background.

4. AWSのクラウドIDEサービス、AWS Cloud9をベースにAWS RoboMaker用に機能拡張した開発環境が開きます。

5. メニューから File -> Upload Local Files... を選択します。

6. ポップアップウィンドウ「Upload Files」が開きます。



→unzip

7. あらかじめダウンロードしておいた delivery-challenge-sample.zip ファイルを「Drag & drop files here」の領域にドラッグ&ドロップしてファイルを開発環境にアップロードします（または [Select files] ボタンをクリックして、ファイルを選択することもできます）。
8. ファイルが開発環境に取り込まれます。ファイルのアップロードが完了したらこのウィンドウは不要なので、「Upload Files」 ウィンドウ右上の x ボタンをクリックして、ウィンドウを閉じます。
9. 開発環境下方にあるターミナルウィンドウをクリック、次のコマンドを入力します。

```
unzip delivery-challenge-sample.zip
```

これでAWS RoboMakerの開発環境の作成、サンプルアプリケーションの取り込みが完了しました！次はコントローラーの用意をしていきましょう

ロボットコントローラーの用意

1. アプリケーションが提供するブラウザインターフェースから AWS にアクセスできるようにするための設定を行います。まず AWS マネジメントコンソールで CloudFormation を開きます。

AWS マネジメントコンソール

The screenshot shows the AWS Management Console homepage. In the top left, there's a search bar with "CloudFormation" typed into it. Below the search bar, the "CloudFormation" service is listed under "AWS のサービス". To the right, there are sections for "外出先でリソースにアクセスする" (Access resources from outside) and "AWS を試す" (Try AWS). Under "AWS を試す", there are links for "Amazon Redshift", "AWS Fargate", and "Amazon S3".

2. [スタックの作成] -> 新しいリソースを使用(標準) を選択します。

The screenshot shows the "CloudFormation > スタック" page. On the left, there's a sidebar with options like "CloudFormation", "StackSets", "エクスポート", "デザイナー", and "CloudFormation レジストリ". The main area shows a table for "スタック (1)". The table has columns for "スタックの名前", "ステータス", "作成時刻", and "説明". One row is visible: "aws-cloud9-robocon-summit-6387f55..." with status "CREATE_COMPLETE" and timestamp "2020-02-12 15:11:18 UTC+0900". At the bottom, there are buttons for "新しいリソースを使用 (標準)" and "既存のリソースを使用 (リソースをインポート)".

3. 「スタックの作成」で [テンプレートファイルのアップロード] を選択します。前提条件はテンプレートの準備完了のままで大丈夫です。↓
4. テンプレートファイルのアップロードの [ファイルの選択] ボタンをクリックします。
5. ファイルの選択ウィンドウが開かれます。あらかじめダウンロードしておいた Cognito_setting_CFn.yaml を選択します。

三 **ステップ 1
テンプレートの指定**

ステップ 2
スタックの詳細を指定

ステップ 3
スタックオプションの設定

ステップ 4
レビュー

スタックの作成

前提条件 - テンプレートの準備

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

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

テンプレートの指定

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

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

Amazon S3 URL テンプレートファイルのアップロード

テンプレートファイルのアップロード

ファイルの選択 Cognito_setting_CFn.yaml
JSON または YAML 形式のファイル

6. 「スタックの作成」に戻ったら [次へ] をクリックして次に進みます。
7. 「スタックの詳細を指定」でスタックの名前を入力、[次へ] をクリックして次に進みます。
 ※注意 スタックの名前にはマイナス(-)を含むことができません。-を含む名前にしてしまうとエラーが発生するので注意してください。

CloudFormation > スタック > スタックの作成

三 **ステップ 1
テンプレートの指定**

ステップ 2
スタックの詳細を指定

ステップ 3
スタックオプションの設定

ステップ 4
レビュー

スタックの作成

スタックの詳細を指定

スタックの名前
roboco202d

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

パラメータ
パラメータは、テンプレートで定義されます。また、パラメータを使用すると、スタックを作成または変更する際にカスタム値を入力できます。

パラメータなし
テンプレートで定義されているパラメータはありません

キャンセル 戻る 次へ

CloudFormation > スタック > スタックの作成

三 **ステップ 1
テンプレートの指定**

ステップ 2
スタックの詳細を指定

ステップ 3
スタックオプションの設定

ステップ 4
レビュー

スタックの詳細を指定

スタックの名前
roboco202d

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

パラメータ
パラメータは、テンプレートで定義されます。また、パラメータを更新すると、スタックを作成または変更する際にカスタム値を入力できます。

パラメータなし
テンプレートで定義されているパラメータはありません

キャンセル 戻る 次へ

8. 「スタックオプションの設定」は特に変更はありません。そのまま [次へ] をクリックして次に進みます。
9. 「レビュー」は画面一番下、AWS CloudFormationによって IAM リソースが作成される場合があることを承認します。のチェックボックスをチェックして、[スタックの作成] をクリックします。

三	スタックの作成オプション
	失敗時のロールバック 有効 タイムアウト - 削除保護 無効

▶ クイック作成リンク

機能

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

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

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

キャンセル

戻る

変更セットの作成

スタックの作成

フィードバック 日本語

© 2008 - 2020, Amazon Web Services, Inc. またはその関連会社。無断転用禁止。 プライバシーポリシー 利用規約

10. 事前に定義されたAWSリソースの作成が開始されます。作業が完了するまでしばらく待ってから [出力] タブを開きます。出力タブのリスト表示に キーが IdentityPoolID で値が出力されていれば準備が完了

了です。値に表示されている文字列は後から使用します。テキストエディターなどに値をコピーしておくと良いでしょう。

The screenshot shows the AWS RoboMaker console interface. At the top, there are buttons for '削除' (Delete), '更新する' (Update), 'Stackアクション' (Stack Actions), and 'Stackの作成' (Create Stack). Below these are tabs for 'Stackの情報' (Stack Information), 'イベント' (Events), 'リソース' (Resources), '出力' (Output) [highlighted in orange], 'パラメータ' (Parameters), and 'テンプレート' (Template). Under the '出力' tab, the heading is '出力 (1)'. A search bar contains the placeholder '検索結果の出力'. A gear icon is available for settings. The main table has columns: 'キー' (Key), '値' (Value), '説明' (Description), and 'エクスポート名' (Export Name). One row is shown: 'IdentityPoolID' with a red box highlighting the value 'ap-northeast-'.

アプリケーション環境の構築

- delivery-challenge-sample の中にある ws_resources.yaml ファイルを開きます。

The screenshot shows the AWS Cloud9 IDE interface. The top navigation bar includes AWS Cloud9, File, Edit, Find, View, Go, Tools, Window, Support, RoboMaker, Run, Simulation (none), and Resources. The left sidebar shows the 'Environment' tree with 'robocon-summit' and 'delivery-challenge-sample' selected. The main editor area shows the 'ws_resources.yaml' file with the following content:

```
cognito_identity_pool_id: # set the IdentityPoolID here
```

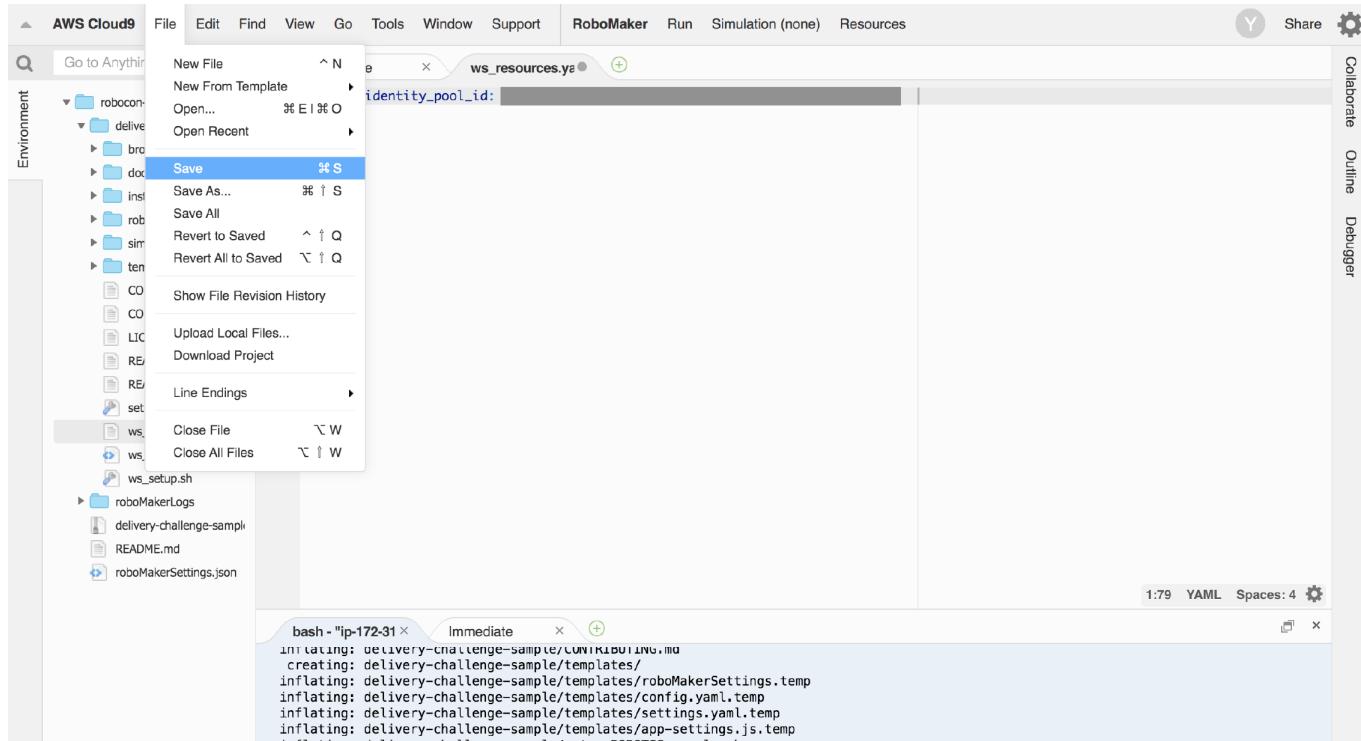
The code editor has tabs for 'ws_resources.yaml' and 'ws_setup.py'. The bottom status bar indicates '1:1 YAML Spaces: 4'.

中の # set the IdentityPoolID here の箇所を ブラウザインターフェースのための準備で取得した IdentityPoolID の値に書き換え、変更を保存します。

```
cognito_identity_pool_id: # set the IdentityPoolID here
```

変更後の例

```
cognito_identity_pool_id: us-east-1:aaaaaaaa-bbbbcccc-dddd-
111111111111
```

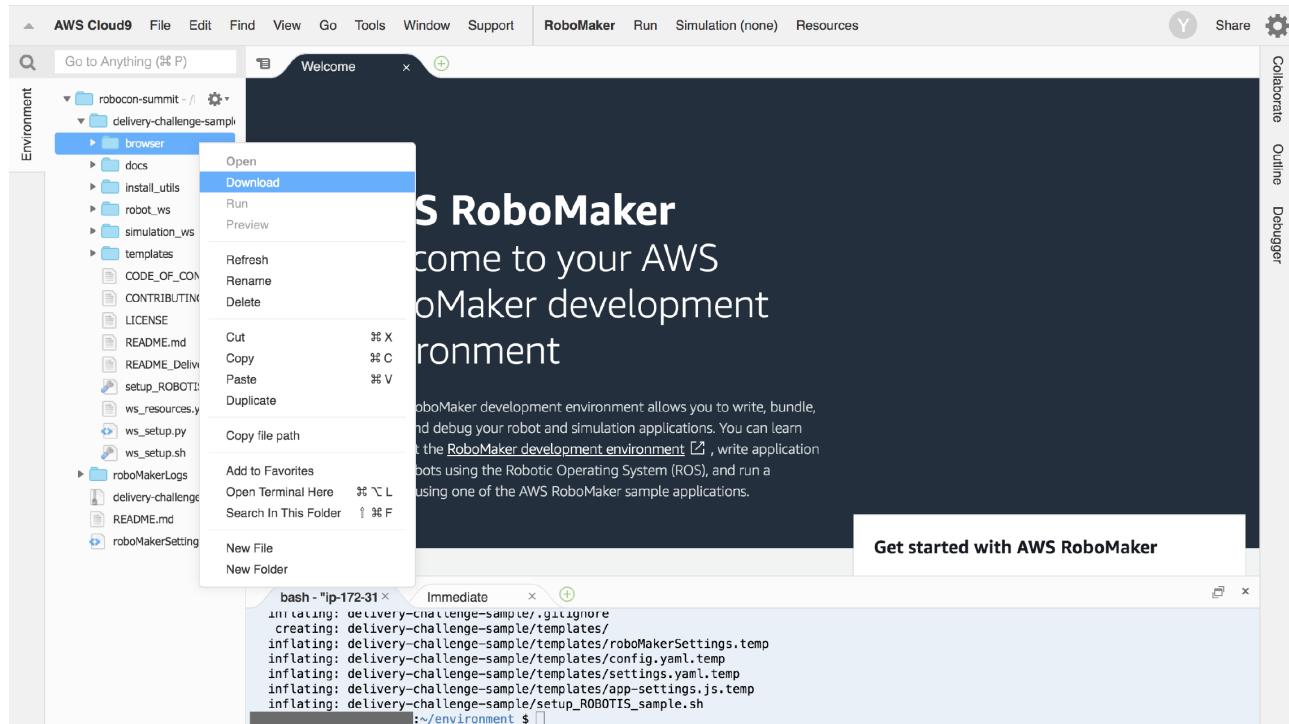


2. ターミナル領域で次を実行して環境構築を開始します

```
cd ~/environment
cd delivery-challenge-sample
./ws_setup.sh
```

競技用アプリケーションの構築が開始されます。完了には 15分から30分ほどかかることがあります。

3. browserフォルダ右クリック、メニューから Download を選びます。ダウンロードしたファイルの browser/robot_controller.html にリモート操作などの機能が提供されます。Webブラウザーで開きます。



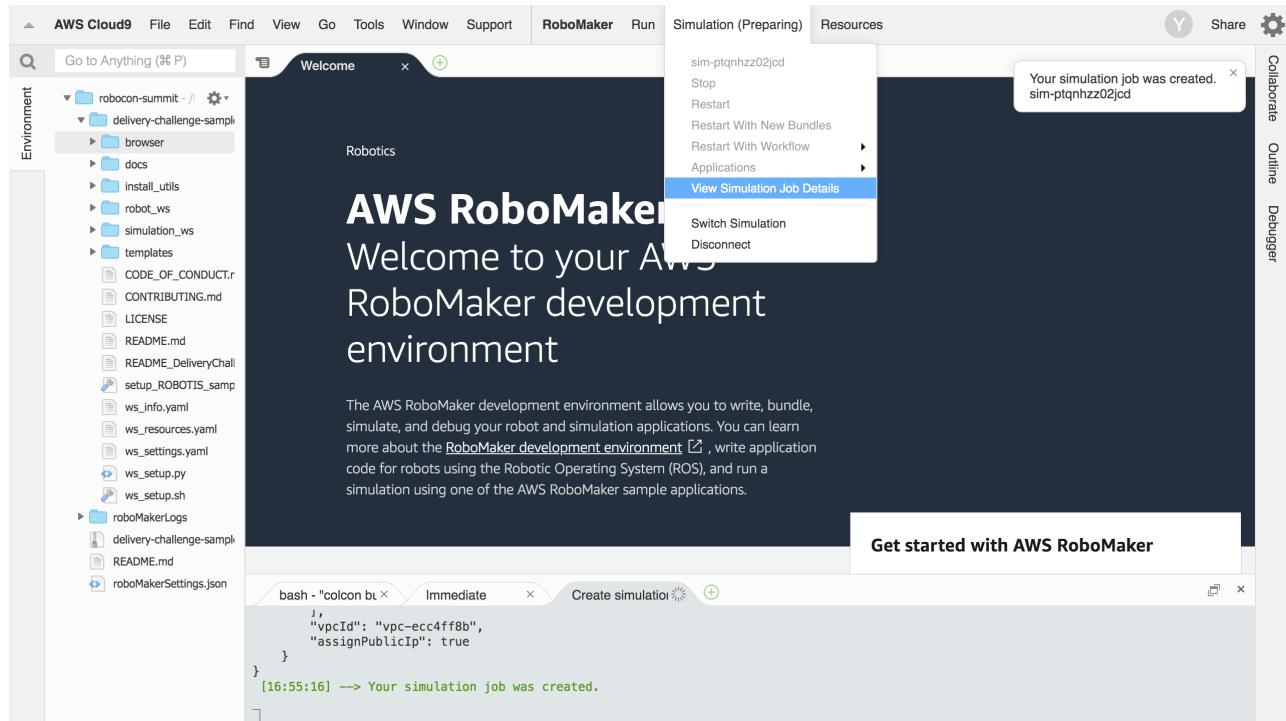
地図作成とナビゲーション

地図の作成

- 開発環境メニュー Run -> Launch Simulation -> Delivery Challenge controller を選択してシミュレーションジョブを開始します。



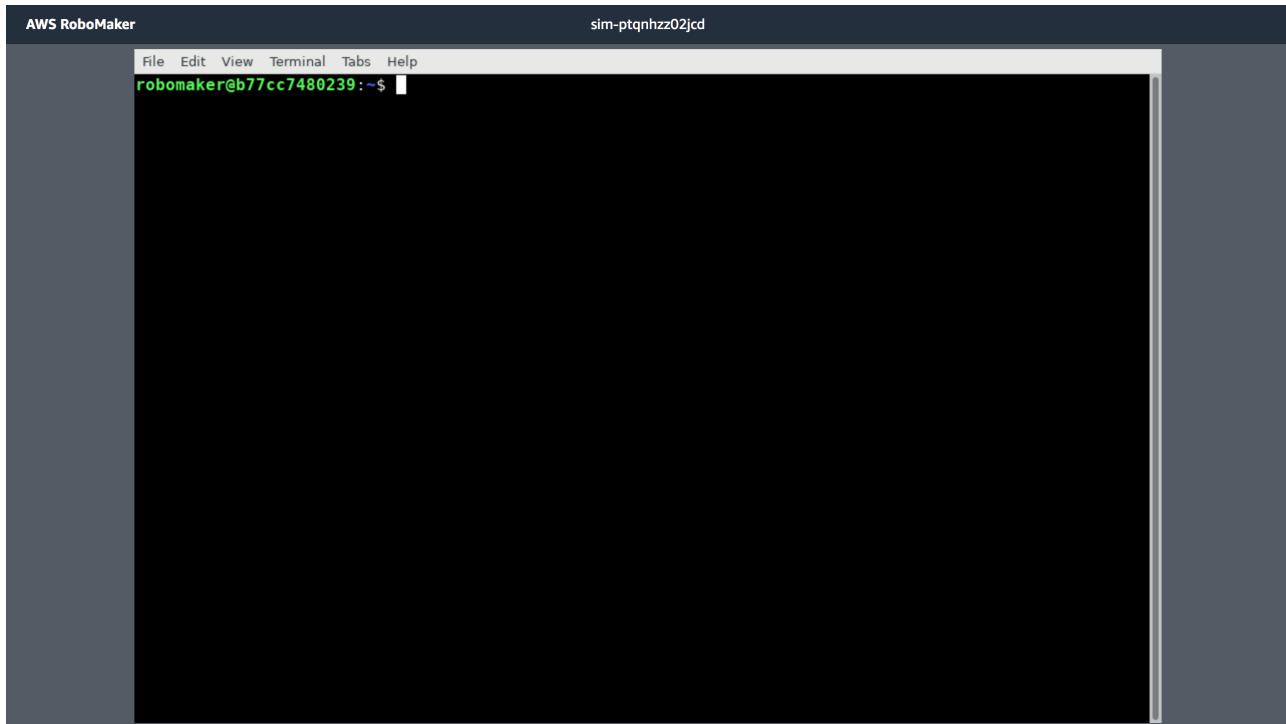
- 開発環境メニュー Simulation -> View Simulation Job Details を開き起動したシミュレーションの詳細画面を開きます。



3. シミュレーションの詳細画面 [Gazebo] アイコンをクリックすることでシミュレーション画面を開くことができます。シミュレーション内のロボットはブラウザーアルファフェースを通じて操作することができます。

The screenshot shows the AWS RoboMaker Simulation Details page for a simulation job named 'sim-ptqnhzz02jcd'. The page displays basic information such as the ID, status (実行中 - Running), and ARN. Below this, there is a 'Simulation Tools' section with four tabs: Terminal, Gazebo, Rqt, and RViz. Each tab has an 'info' button next to it. The Gazebo tab is currently selected, showing a Gazebo logo icon.

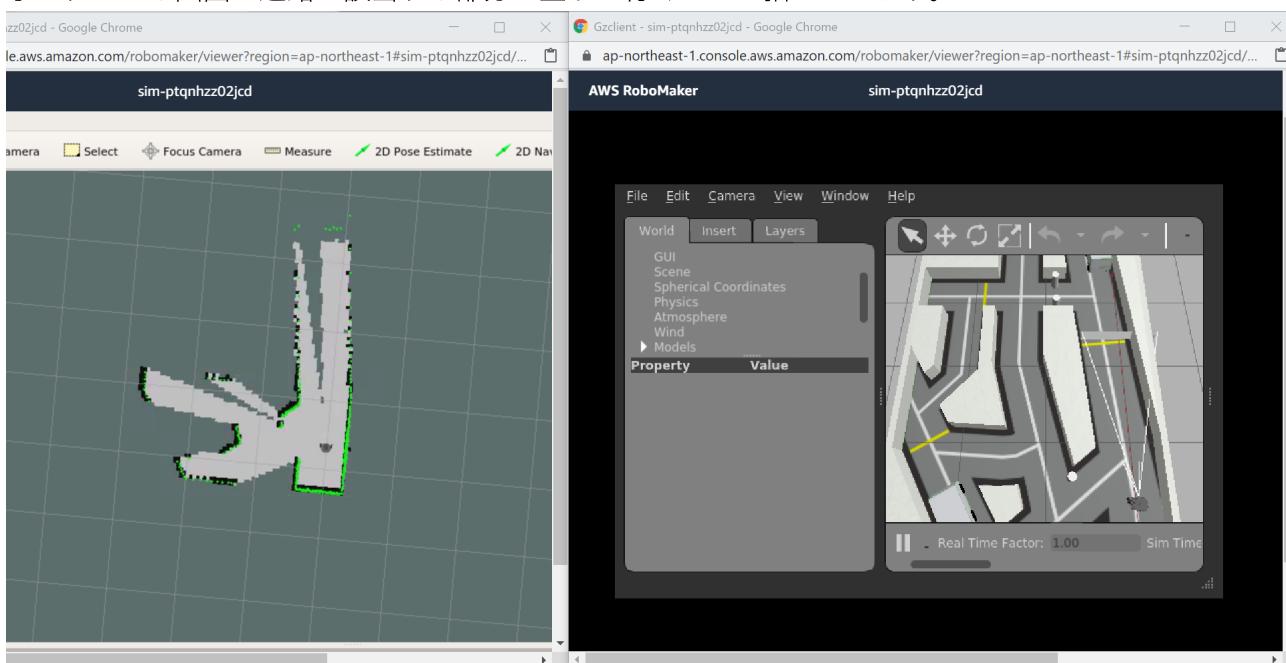
4. シミュレーションの詳細画面 [Terminal] アイコンをクリックし、ターミナルを開きます。次を実行することでマップの作成を開始することができます。



```
for dp in `find ~/workspace/bundle-store -maxdepth 2 -name setup.sh`;
do
    BUNDLE_CURRENT_PREFIX=${dp:0:-9}
    source ${dp} --extend
done

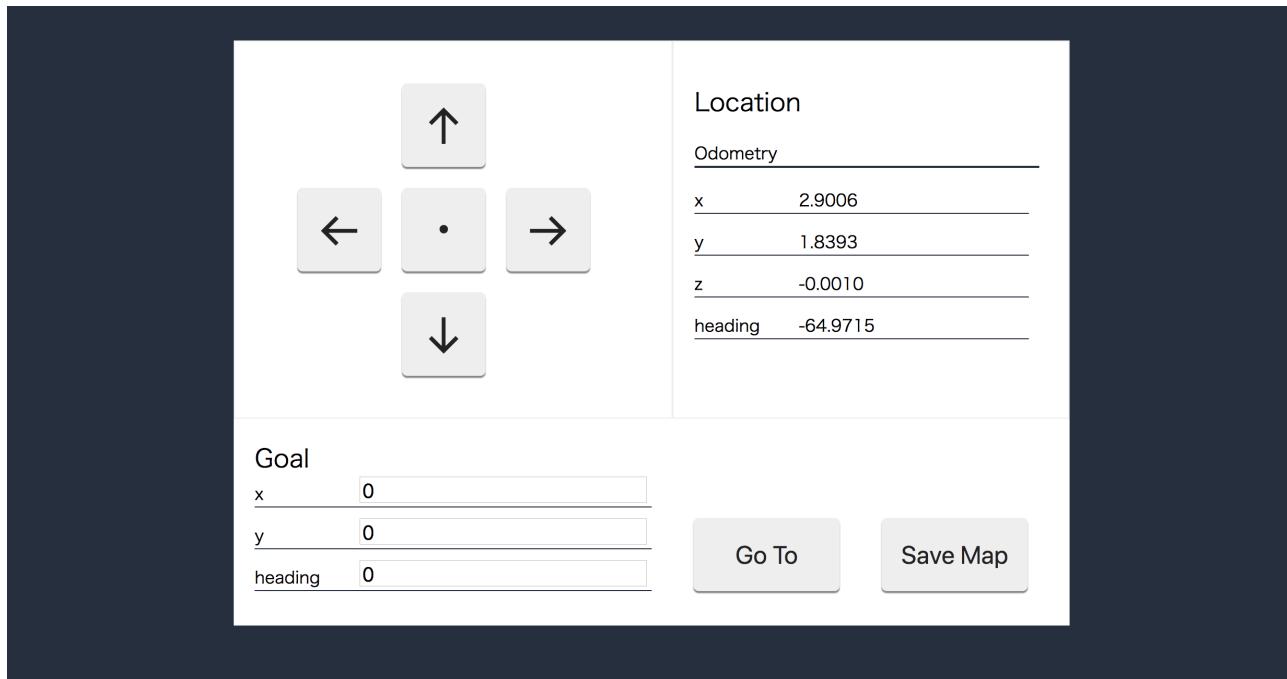
export TURTLEBOT3_MODEL=burger
roslaunch delivery_robot_sample slam.launch
```

5. ブラウザインターフェースでロボットを操作して環境の地図を作成します。環境の地図を作る、
とはGazeboの画面をみながらロボットを操作して道路を一通り走らせることで、ターミナル画面に表
示されている画面で道路に該当する部分を塗って行くことを指しています。



(Gazeboの操作について詳しくは[Gazeboについて](#)を、地図の作り方について詳しくは[地図作りのコツ](#)を参照)

- 地図が完成したらブラウザインターフェースで [Save Map] ボタンをクリックします。地図情報は Amazon S3 に保存されます。（保存先バケット名は robot_ws/src/delivery_robot_sample/settings/settings.yaml で確認することができます）



- 地図の保存がうまく行ったらシミュレーションジョブの詳細画面 [アクション] -> [キャンセル] を選んでシミュレーションジョブを終了させます。

シミュレーションは使った時間と処理の重さに応じた従量課金（処理の重さにも依存しますが、1時間使って約 \$3）です。不要に長くシミュレーションを起動したままにして、クレジットを使い果たさないように気をつけてください！ なお、シミュレーションは起動時の設定で1時間半たてば自動終了するように設定されています。この時間の調整は roboMakerSettings.json ファイルを編集することにより可能です。

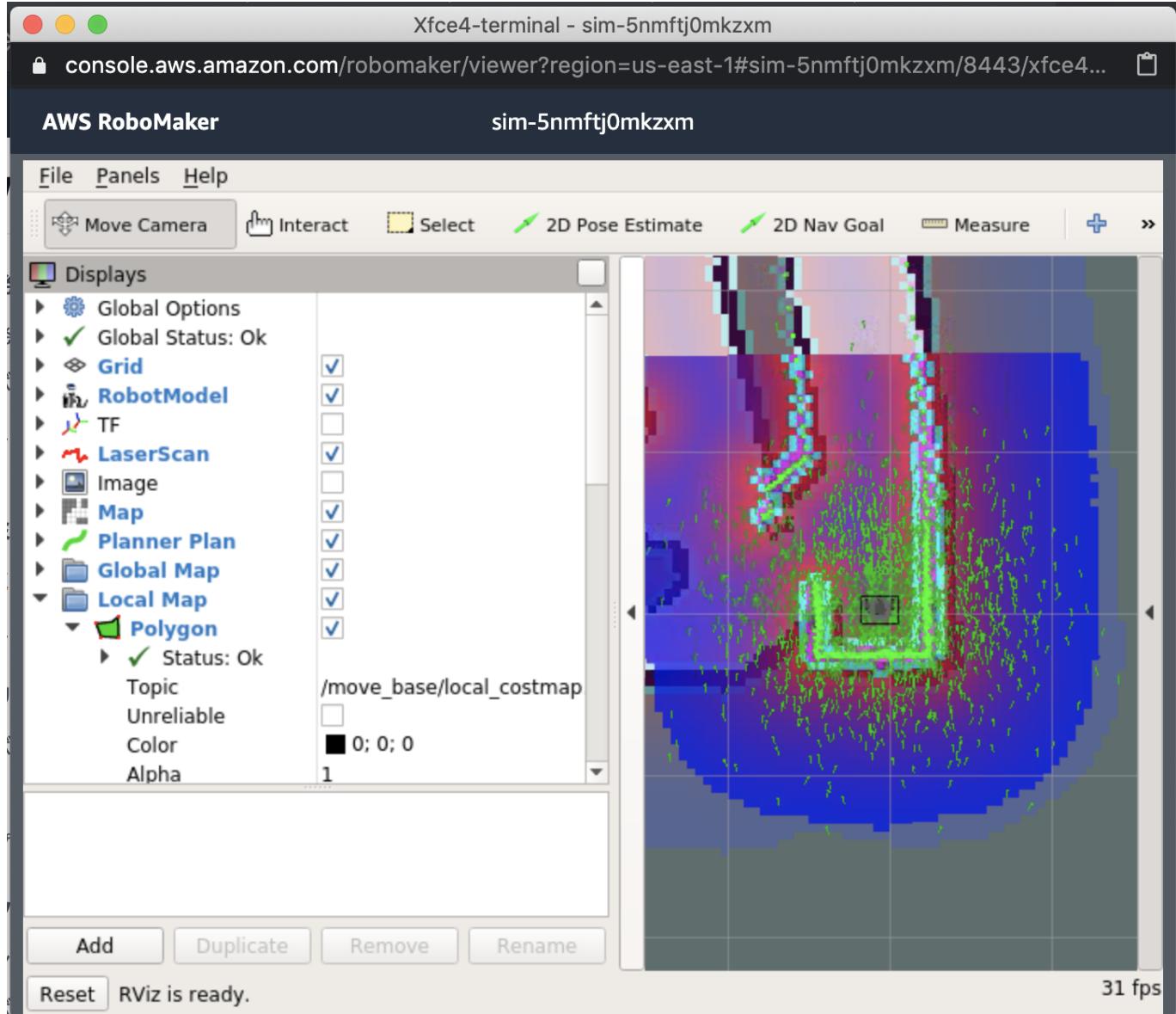
ナビゲーション

地図が保存されたら、保存された地図を使って指定された場所への自動走行が可能になります。

1. 開発環境メニュー Run -> Launch Simulation -> Delivery Challenge navigation を選択してシミュレーションジョブを開始します。
2. 開発環境メニュー Simulation -> View Simulation Job Details を開き起動したシミュレーションの詳細画面を開きます。
3. [Gazebo] アイコンをクリックすることでシミュレーション画面を開くことができます。シミュレーション内のロボットはブラウザインターフェースを通じて操作することができます。
4. (オプション) シミュレーションの詳細画面 [Terminal] アイコンをクリックし、ターミナルを開きます。次を実行することで自動計算されたナビゲーションルートやロボットが認識している周辺情報などを視覚化することができます。

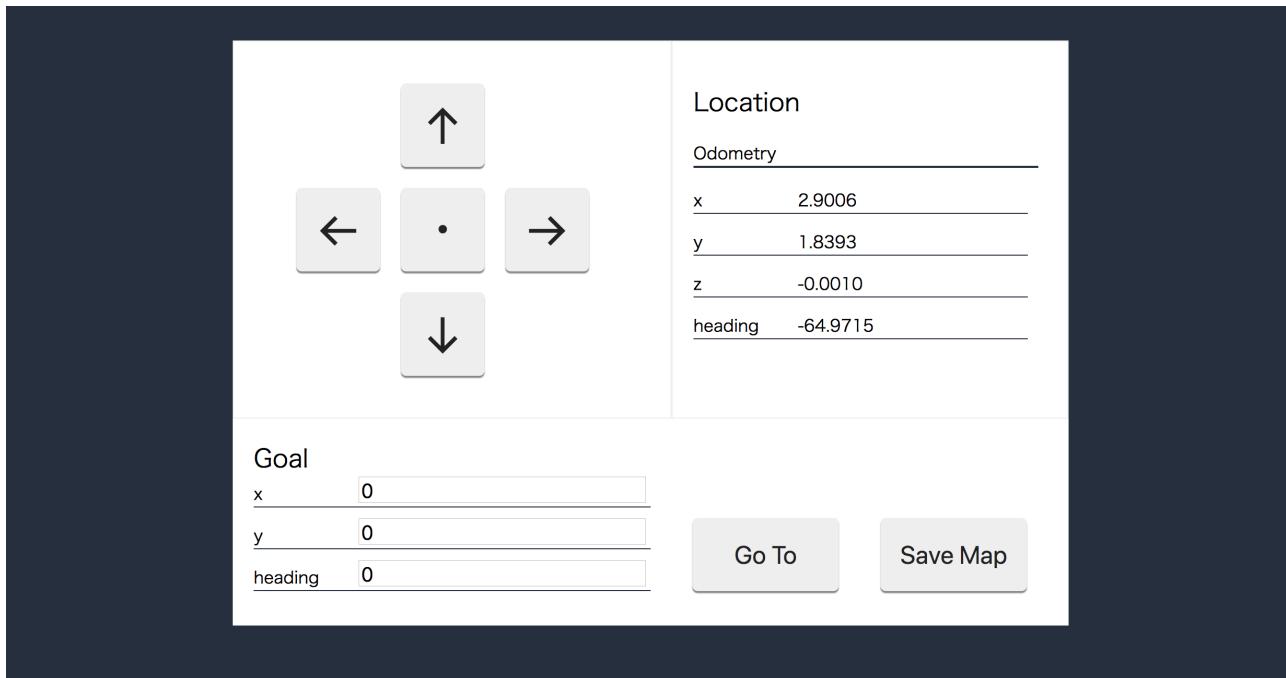
```
for dp in `find ~/workspace/bundle-store -maxdepth 2 -name setup.sh`;
do
  BUNDLE_CURRENT_PREFIX=${dp:0:-9}
  source ${dp} --extend
done

export TURTLEBOT3_MODEL=burger
roslaunch delivery_robot_sample rviz_for_navigation.launch
```



5. ブラウザインターフェースで「Goal」に向かう場所の座標を入力します。（現在のロボットの位置座標はブラウザインターフェースの「Location」で確認することができます。あらかじめロボットを目標位置に移動させて、その場所の座標情報をメモしておくことで、その場所への移動を何度も繰り返すことができます）

り返すことができます)



6. ブラウザインターフェースで「Go To」ボタンをクリックします。ロボットが、目標位置に向けて移動を開始します。

シミュレーションでの作業が一通り終わったらシミュレーションジョブの詳細画面 [アクション] -> [キャンセル] を選んでシミュレーションジョブを終了させましょう。シミュレーションは使った時間と処理の重さに応じた従量課金（処理の重さにも依存しますが、1時間使って約 \$3）です。不要に長くシミュレーションを起動したままにして、クレジットを使い果たさないように気をつけてください！ なお、シミュレーションは起動時の設定で1時間半たてば自動終了するように設定されています。この時間の調整は roboMakerSettings.json ファイルを編集することにより可能です。

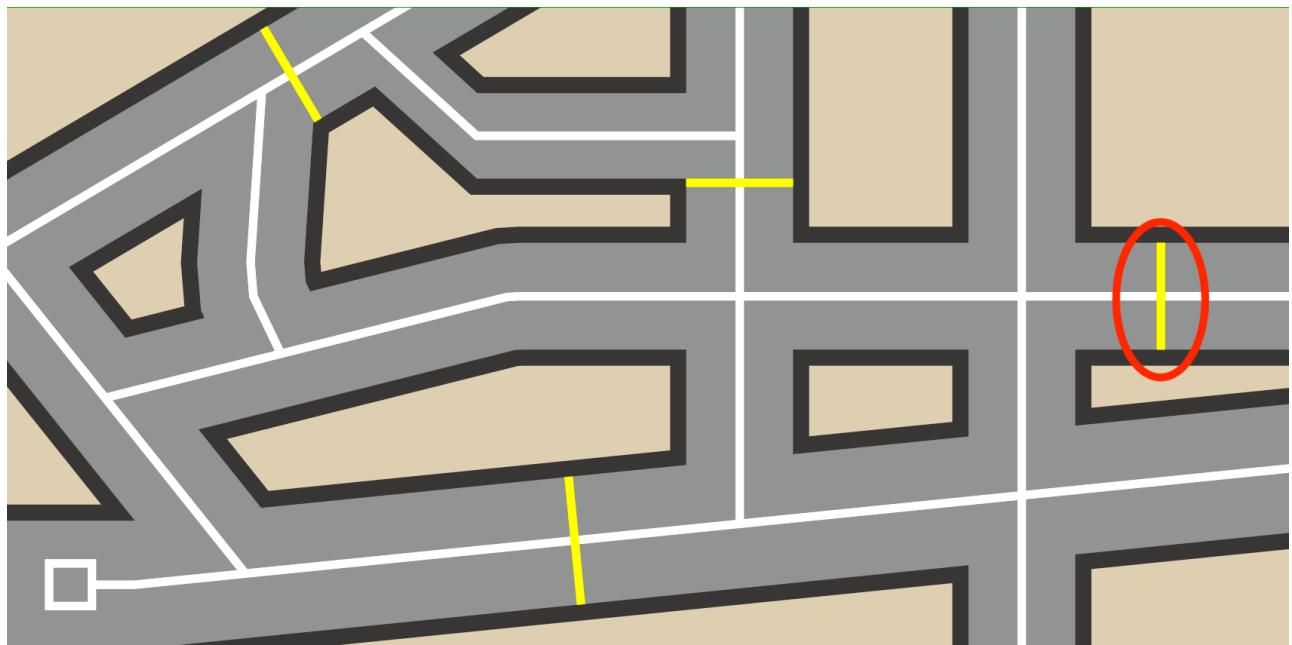
タイム計測方法

タイム測定はダウンロードしたブラウザインターフェースの browser/game_console.html から行います。タイム計測を [Start] ボタンをクリックして開始します。[Start] ボタンをクリックするとロボットが初期位置に戻され、タイマーが開始します。ゴールに到着すると Status の表示が Finished に代わり、タイマーが止まります。

予選リーグへの戦略を立てよう

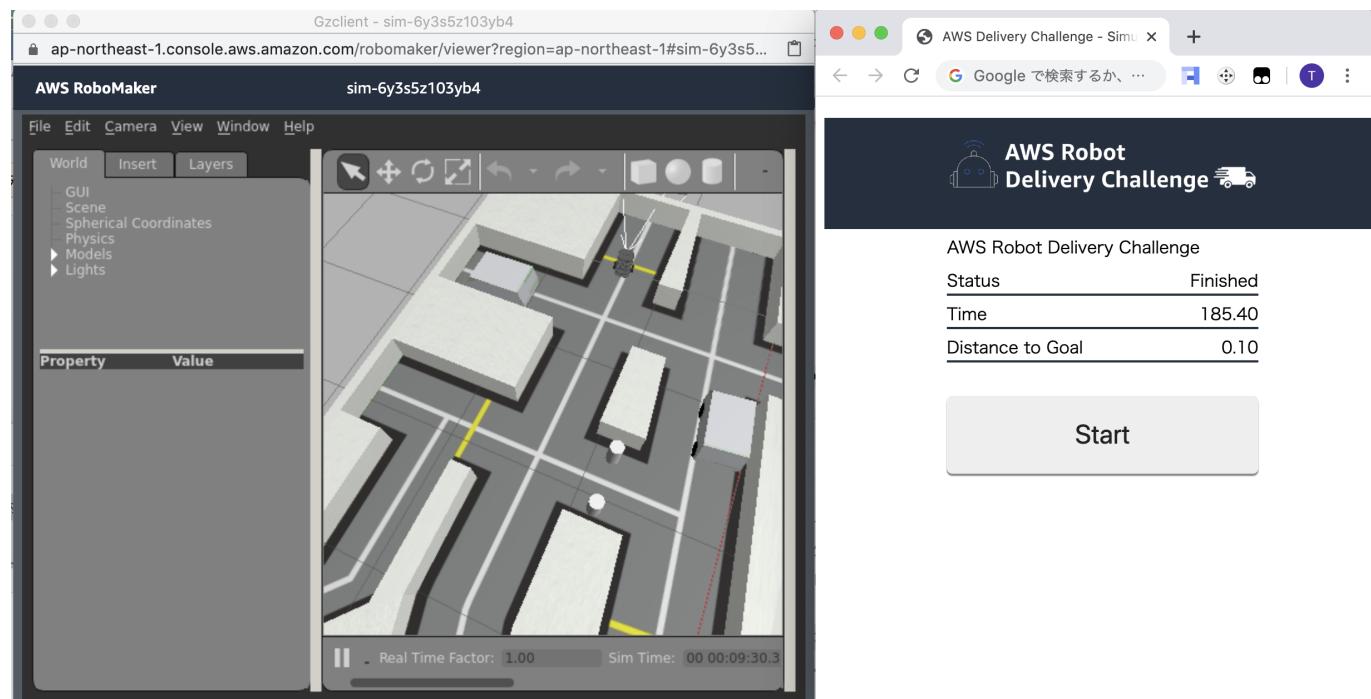
予選リーグ測定について

1. 予選リーグは RoboMaker のシミュレーション環境で指定された目的にまでの到着時間を競うタイムトライアルです。シミュレーションワールドの中で次の赤丸の場所を目指します。



2. ブラウザインターフェースの browser/game_console.htmlを開き、[タイム計測方法](#)にあるようにタイムを計測します。ゴールに到着すると Status の表示が Finished に代わり、タイマーが止まります。この状態を、Gazebo の画面と合わせてスクリーンショットを結果として報告します。(下のような写真を送ってください。)

注意： 計測中、リモートコントロールによって直接のロボットの移動速度を操作することは NG とします。
測定開始後にナビゲーションでの座標指定や、運転を開始するためのコマンドを送ることは可能です



アプリケーションコードの更新と実行

提供の地図を作る機能とナビゲーション機能はサンプルアプリケーションとして提供しています。サンプルアプリケーションとして提供されている箇所は robot_ws/src/delivery_challenge_sample 以下のファイルで、この箇所の変更は自由です。サンプルアプリケーションの中身の概要については [アプリケーションコードの詳細と改良の方法\(Detail\)](#) のドキュメントを参照してください。

各ツールの操作のコツとチューニングポイント

Gazeboについて

[Gazebo] は3Dの物理的なシミュレーション機能を含むシミュレータです。 ウィンドウのサイズは角をドラッグすることで変更することができます。 シミュレーション画面の視点はマウスを使って変更することができます。 マウスでの視点の操作方法は次のページの一番下に説明されています。

http://gazebosim.org/tutorials?cat=guided_b&tut=guided_b2

Rvizについて

「地図の作成」、「ナビゲーション」、それぞれの作業でターミナルでコマンドを入力した後に起動されるウィンドウは rviz と呼ばれるツールです。 rviz はロボットが認識している情報を視覚化するツールです。 今回の例では ロボットの LiDAR センサーが認識した障害物を赤い点(点群)で、また生成している地図を画像として表示しています。 ウィンドウのサイズは角をドラッグすることで変更することができます。 マウスを使って視覚化されている画面の視点を調整することができます。 左クリックしながらのマウスの移動で回転、[Shift] + 左ボタンで画面を上下左右に移動、右ボタンでズームをすることができます。

地図作りのコツ

地図画像の中で黒い場所が障害物で、無色の箇所は未知の領域です。 地図はロボットに搭載されている LiDAR センサーからの情報をもとに作成されます。 LiDAR センサーは周囲360度の障害物までの距離を測定することができるセンサーです。 地図を作るために全てのルートを通過しなければならないというわけではありません。 LiDAR センサーが壁を障害物として認識できれば、通過していない場所でもその場所の地図情報は形成されます。 例えば狭くて通過が難しいルートがあった時、無理してそこを通過する必要はないかもしれません。 その狭い道につながる別のルートがあれば、その別ルートから反対側に移動することにより、狭い通路を挟む周辺の地図をつくることができます。

ブラウザインターフェースで [Save Map] ボタンをクリックするとファイルは一度 /tmp 配下に保存され、 次に Amazon S3 に保存されます。 Amazon S3 での保存先は
robot_ws/src/delivery_robot_sample/settings/settings.yaml ファイルで設定されています。

保存されるファイルは map.pgm と map.yaml ファイルの 2 つです。 map.pgm ファイルはグレースケールの 画像ファイルで、作成された地図データになります。 これは画像ファイルなので、画像編集ソフトで編集することも可能です。 画像データの中で、黒い箇所が障害物として認識されます。 例えば地図の中でロボットを走行させたくない領域がある場合、その周辺を黒く塗りつぶすことで、ロボットにそこに障害物があると解釈させることができます。 大幅に変更するとロボットのセンサーが捉えた情報と地図が一致しなくなり、ロボットが自身の位置を判断することができなくなるので、行きすぎた編集には注意が必要です。

マップ作成のチューニングポイント

マップの精度などに満足がいかない場合、例えば次のチューニングが考えられます。

- gmapping 以外の SLAM の手法を利用する (SLAM そのほか用語については「サンプルアプリケーション各機能の内部仕様の概要」の「地図の作成」を参照してください)
 - サンプルアプリケーションは SLAM メソッドとして gmapping を利用しています。 ROBOTIS サンプルプログラムは標準で gmapping 以外に cartographer, hector, karto, frontier_exploration に対応できるように作られています。 これらを今回のサンプルアプリケーションの中で使うには robot_ws/src/delivery_robot_sample/launch/slam.launch ファイルの slam_methods の値を変更

し、またこのメソッドをアプリケーションの中にバンドルするよう、`package.xml` というファイルを編集済みます。

- 例: `cartographer` を使う場合
- `robot_ws/src/delivery_robot_sample/launch/slam.launch`、`value` を `cartographer` に

```
<launch>
  <include file="$(find turtlebot3_slam)/launch/turtlebot3_slam.launch">
    <arg name="slam_methods" value="cartographer"/>
  </include>
</launch>
```

- `robot_ws/src/delivery_robot_sample/package.xml`、`<exec_depend>` に `cartographer` を加える

```
:          :
<!!-- <exec_depend>slam_gmapping</exec_depend> -->
<exec_depend>cartographer</exec_depend>
<exec_depend>cartographer_ros</exec_depend>
<exec_depend>cartographer_ros_msgs</exec_depend>
<exec_depend>cartographer_rviz</exec_depend>
<!!-- <exec_depend>hector_mapping</exec_depend>
<exec_depend>slam_karto</exec_depend>
<exec_depend>frontier_exploration</exec_depend>
<exec_depend>navigation_stage</exec_depend> -->
</package>
```

ファイルを変更したらビルド、バンドル作業をおこない、変更内容をロボットアプリケーションに反映される必要があります。方法は、Detailファイルで説明の「ビルド、バンドルを行う」、を参照してください。

ナビゲーションのチューニングポイント

狭い道を通れずにロボットがスタックするなど、ナビゲーションがうまく行かない場合、例えば次のチューニングが考えられます。

- ナビゲーションの各パラメータを調整する
 - ナビゲーションスタックを起動する際、障害物からどこまでの範囲を通過できないエリアとして処理するかなどをパラメータ値で与えています。これらの値を調整することで、例えば狭い道を経路の候補から外せたり、通れるようにさせることができます。パラメータは `robot_ws/src/turtlebot3/turtlebot3_navigation/param/costmap_common_param_burger.yaml` の中にあります。
 - チューニングについて詳しくは次の ROBOTIS 社のサイトを参考にしてください。
<http://emanual.robotis.com/docs/en/platform/turtlebot3/navigation/#tuning-guide>
 - ファイルを変更したらビルド、バンドル作業をおこない、変更内容をロボットアプリケーションに反映される必要があります。方法は、Detailファイルで説明の「ビルド、バンドルを行う」、を参照してください。
- 地図ファイルを編集する

- 地図ファイルを画像編集ファイルなどで編集、更新して、ナビゲーションを最適化できる可能性があります。地図ファイルとして保存されているのは map.pgm と map.yaml ファイルの 2 つです。map.pgm ファイルはグレースケールの画像ファイルで、作成された地図データになります。これは画像ファイルなので、画像編集ソフトで編集することも可能です。画像データの中で、黒い箇所が障害物として認識されます。例えば地図の中でロボットを走行させたくない領域がある場合、その周辺を黒く塗りつぶすことで、そこに障害物があると解釈させることができます。あまり大きく編集すると、ロボットのセンサーが捉えた情報と地図が一致しなくなり、ロボットが自身の位置を判断することができなくなるので、行きすぎた編集には注意が必要です。
- robot_ws/src/delivery_robot_sample/settings/settings.yaml の s3_bucker の値が S3 で保存されているバケットの名前で、バケット内 s3_prefix で指定された場所に地図が保存されています。
<https://s3.console.aws.amazon.com/s3> から該当するバケットにアクセス、ダウンロードと、アップロード、更新を行うことができます。地図はリアルタイムでロボットアプリケーションに反映させるわけではありません。更新したら、再度アプリケーションを起動して最新の地図でのナビゲーションを開始します。