

AWS Robot Delivery Challenge 環境構築 Linux 編

このドキュメントは AWS Robot Delivery Challenge の配布アプリケーションを Linux デスクトップ環境で実行する方法を説明しています。このドキュメントの内容は上級者向けです。まずは HOWTO.pdf の内容を確認してください。

事前準備

- 以降の手順を進めるには事前に次がインストールされている必要があります。
 - ROS (Melodic または Kinetic、参考 <http://wiki.ros.org/ROS/Installation>)
 - Gazebo (参考 http://gazebo.org/tutorials/tut=install_ubuntu&cat=install)
 - python2, python3, pip, pip3, curl
- AWS CLI を使えるようにします。CLIから管理者権限でAWSアカウントにアクセスできるようクレデンシャルの設定を行います。
 - 参考：https://docs.aws.amazon.com/ja_jp/cli/latest/userguide/cli-chap-welcome.html
- `.bashrc` で ROS 環境のセットアップスクリプトを読み込むようにしておきます。

```
ROS_DISTRO=melodic # もしくは kinetic
echo "source /opt/ros/$ROS_DISTRO/setup.bash" >> ~/.bashrc
. ~/.bashrc
```

ソースコードのダウンロード

- アプリケーションのソースコードをダウンロードします。
入手したソースコードをローカルの開発環境に展開してください。

ブラウザーインターフェースのための準備

- アプリケーションが提供するブラウザー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。
- [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。
- 「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。
- 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。
- ファイルの選択ウィンドウが開かれます。ダウンロードしたソースコードの browser フォルダの中の setup の中にある `Cognito_setting_CFn.yaml` を選択します。
- 「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（任意ただしマイナス('-')記号は含めないでください）を入力、**[次へ]**をクリックして次に進みます。
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。
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。AWSマネージメントコンソールでの作業はここまでです。

アプリケーション環境の構築

1. ソースコードの `delivery-challenge-sample` の中にある `ws_resources.yaml` ファイルを開きます。中の **`set the IdentityPoolID here`** の箇所を **ブラウザー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-bbbb-cccc-dddd-111111111111
```

2. 次を実行して環境構築を開始します

```
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。

メモ：環境構築のプロセスでは複数のリソースが AWS 上に作成されます。作成されたリソースの情報は `ws_resources.yaml` ファイルに記録されます。途中でエラーが出た時は、問題を修正して、再度 `ws_setup.sh` を実行します。 `ws_setup.sh` 実行の際、 `ws_resources.yaml` に記録されているリソースについては、再度生成することなく、 `ws_resources.yaml` の内容を引き継ぎます。 `ws_setup.sh` は `ws_resources.yaml` に加え `ws_settings.yaml` と `ws_info.yaml` というファイルも書き出します。途中でエラーが出たことが理由で再度 `ws_setup.sh` を実行する場合は `ws_settings.yaml` と `ws_info.yaml` については削除してから実行を行なった方がより確実な再実行が行われます。（これら3つのファイルは `ws_setup.sh` の中で行われている一つ一つの処理を記録しているもので、記録が残っている作業については再実行の際はスキップするよう作用します）

サンプルアプリケーションの使い方

ブラウザーインターフェースの事前準備でダウンロードしたファイルの browser/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。ブラウザーインターフェースを通して次の操作ができます。

- ロボットのリモート操作
- 地図のクラウドへの保存 (サンプル SLAM アプリケーションが動いている時のみ動作)
- 目的地への自動移動(ナビゲーション) (サンプル Navigation アプリケーションが動いている時のみ)
- 現在位置の確認 - これはナビゲーションで目的地を指定するときのガイドになる数字です。ロボットに向かわせたいところにロボットを移動させて座標情報を記録しておきましょう。

地図の作成

ナビゲーションを行うにはナビゲーション機能が利用する環境の地図が必要です。地図の作成は次のようにして行います。

1. ターミナルを開き、次のコマンドを入力してシミュレーション環境を起動します。

```
cd delivery-challenge-sample/simulation_ws
source ./install/local_setup.bash

roslaunch delivery_challenge_simulation create_stage.launch gui:=true
```

2. 新たにターミナルを開く、次のコマンドを入力してロボットアプリケーションを起動します。

```
cd delivery-challenge-sample/robot_ws

source ./install/local_setup.bash

export TURTLEBOT3_MODEL=burger
roslaunch delivery_robot slam_robot.launch
```

3. ブラウザーインターフェースでロボットを操作して環境の地図を完成させます。
4. 地図が完成したらブラウザー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）

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. ターミナルを開き、次のコマンドを入力してシミュレーション環境を起動します。

```
cd delivery-challenge-sample/simulation_ws
source ./install/local_setup.bash

roslaunch delivery_challenge_simulation create_stage.launch gui:=true
```

2. 新たにターミナルを開く、次のコマンドを入力してロボットアプリケーションを起動します。

```
cd delivery-challenge-sample/robot_ws

source ./install/local_setup.bash

export TURTLEBOT3_MODEL=burger
roslaunch navigation_robot navigato_in_robot.launch
```

3. ブラウザー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）
4. ブラウザー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。

予選リーグ測定

予選リーグ測定のためにはアプリケーションを AWS RoboMaker のシミュレーションで動かす必要があります。次にローカルのデスクトップ環境で作成したアプリケーションの AWS RoboMaker シミュレータでの実行方法を説明します。

1. AWS RoboMaker のシミュレーションで動かすためのロボットアプリケーションのバンドルを作る

```
cd robot_ws
colcon bundle
```

2. S3 にロボットアプリケーションをアップロード、AWS RoboMakerにロボットアプリケーションを登録する

```
cd robot_ws
aws s3 cp aws bundle/output.tar s3:
<bucket_name>/bundle/bundle/robot_app.tar
```

(<bucket_name> は ws_resources.yaml の bucket_name の値で読み変えます)

RoboMaker へのアプリケーションの登録の仕方については次を参考にしてください。

https://docs.aws.amazon.com/ja_jp/robomaker/latest/dg/application-create-simjob.html#application-simjob-robotapp

3. AWS RoboMaker のシミュレーションで動かすためのロボットアプリケーションのバンドルを作る

```
cd simulation_ws
colcon bundle
```

4. S3 にシミュレーションアプリケーションをアップロード、AWS RoboMakerにシミュレーションアプリケーションを登録する

ファイルのアップロードは次のようにして行います

```
cd simulation_ws
aws s3 cp aws bundle/output.tar s3:
<bucket_name>/bundle/bundle/simulation_app.tar
```

(<bucket_name> は ws_resources.yaml の bucket_name の値で読み変えます)

RoboMaker へのアプリケーションの登録の仕方については次を参考にしてください。

https://docs.aws.amazon.com/ja_jp/robomaker/latest/dg/application-create-simjob.html#application-simjob-simapp

5. シミュレーションジョブを開始する

参照： https://docs.aws.amazon.com/ja_jp/robomaker/latest/dg/application-create-simjob.html#application-create-simulation-job

(AWS RoboMaker のシミュレータでアプリケーションを実行するのに必要な必要なリソースはすでに作られ、作られたリソースの情報は ws_resources.yaml に記述されています)

シミュレーションが起動したら、計測を開始します。ゴール地点や、時間の計測の方法については、HOWTO.pdf の「予選リーグ測定について」を参考にしてください。

アプリケーションコードの更新と実行

robot_ws/delivery_robot_sample 配下のプログラムはサンプルプログラムです。書き換える。または新たなロボットアプリケーションを作成してコンテストに望むことが可能です。

robot_ws/delivery_robot_sample 配下を変更した時の変更の反映方法は次の通りです。

```
cd delivery-challenge-sample/robot_ws
rosdep install --from-paths src --ignore-src -r -y
colcon build
```

License

This library is licensed under the MIT-0 License. See the LICENSE file.