

Google Home™ しりとリアプリ - サポート #410

CircleCIで自作Dockerイメージを使用するようにする

2018/08/04 18:41 - みぞ @mizo0203

ステータス:	却下	開始日:	2018/08/04
優先度:	通常	期日:	
担当者:	みぞ @mizo0203	進捗率:	0%
カテゴリ:		予定工数:	0.00時間
対象バージョン:	v0.2-recreation	作業時間:	0.00時間
説明			
関連するチケット:			
関連している サポート #453: CircleCI にて古い .deb ファイルがキャッシュされないように修正		完了	2018/10/14

履歴

#1 - 2018/09/27 10:22 - みぞ @mizo0203

- 対象バージョンを v0.1-demo にセット

#2 - 2018/10/07 00:09 - a a

- 題名を CircleCI 上の Docker イメージ circleci/python:2.7.15 に Google Cloud SDK をインストールから CircleCI で自作Dockerイメージを使用するようにする に変更

- 対象バージョンを v0.1-demo から v0.2-recreation に変更

#3 - 2018/10/14 15:56 - みぞ @mizo0203

- 関連している サポート #453: CircleCI にて古い .deb ファイルがキャッシュされないように修正 を追加

#4 - 2018/10/14 16:03 - みぞ @mizo0203

- ステータスを 新規 から 却下 に変更

自作 Docker イメージを使用する場合、Google Cloud SDK がリリースされる毎に Docker イメージを再ビルドして Docker hub にプッシュしなければならない。

既存の Docker イメージ circleci/python:2.7.15 に最新の Google Cloud SDK をインストールするように circle.yml で設定するほうが運用コストが低い。

なので、circleci/python:2.7.15 を自作 Docker イメージに置き換えるのではなく、既知の問題である [#453](#) を修正する。

#5 - 2018/10/14 16:07 - みぞ @mizo0203

別記の理由により、ボツになった Dockerfile

FROM circleci/python:2.7.15

```
# https://github.com/GoogleCloudPlatform/cloud-sdk-docker/blob/master/Dockerfile
```

```
# https://cloud.google.com/sdk/docs/downloads-apt-get
```

```
RUN sudo apt-get update && /
    sudo apt-get install -y lsb-release /
    apt-transport-https && /
    export CLOUD_SDK_REPO="cloud-sdk-$(lsb_release -c -s)" && /
    echo "deb http://packages.cloud.google.com/apt $CLOUD_SDK_REPO main" | sudo tee -a /etc/apt/sources.list.d/google-cloud-sdk.list
    curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add - && /
    sudo apt-get update && /
    sudo apt-get install -y google-cloud-sdk /
    google-cloud-sdk-app-engine-python /
    google-cloud-sdk-app-engine-python-extras && /
    gcloud config set core/disable_usage_reporting true && /
    gcloud config set component_manager/disable_update_check true && /
    gcloud config set metrics/environment github_docker_image && /
    gcloud --version && /
    docker --version
```