

# .Debian

銀河系唯一のDebian専門誌

2020年11月21日

Python、原稿システム特集



# 会 強 勉 研 究 会 の ア ー キ ブ ト

---

## 目次

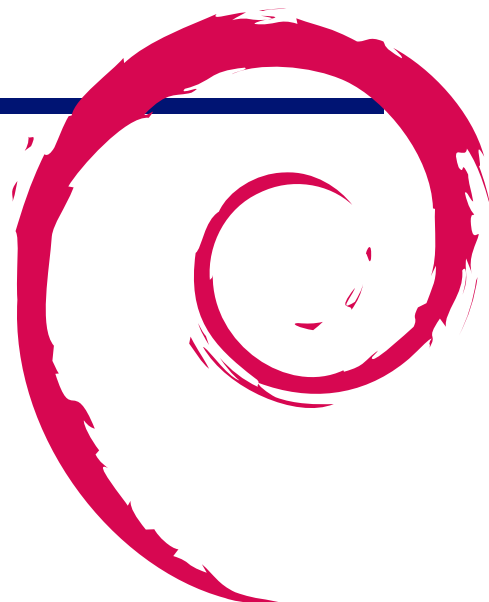
1	最近の Debian 関連のミーティング報告	2	2.10 dancercj	3	
1.1	2020 年 10 月度 東京エリア・関西合同 Debian 勉強会	2	2.11 sambar_gamer	4	
2	事前課題	3	2.12 henrich	4	
2.1	dictoss	3	3	Debian での Python	5
2.2	koedoyoshida	3	3.1	はじめに	5
2.3	yosuke_san	3	3.2	Python について	5
2.4	tanykazy	3	3.3	Debian における Python 情報の取り扱いについて	5
2.5	YukiharuYABUKI	3	3.4	終わりに	8
2.6	kenhys	3	3.5	参考情報	8
2.7	ipv6waterstar	3	4	Debian 勉強会資料のディレクトリ構成変更	9
2.8	space320jt	3	4.1	Debian 勉強会資料ディレクトリ構成変更の目的	9
2.9	yy-y-ja-jp	3	4.2	ディレクトリ構成変更の実際	9
			5	メモ	12

---

# 1 最近の Debian 関連のミーティング報告

杉本 典充

---



## 1.1 2020 年 10 月度 東京エリア・関西合同 Debian 勉強会

2020 年 10 月 17 日 (土) に東京エリア Debian 勉強会と関西 Debian 勉強会の合同でオンラインによる Debian 勉強会を開催しました。参加者は 13 名でした。

セミナーは上川さんによる「Debian での Rust」の発表を行いました。発表では Rust を始める人が読むとよいドキュメント<sup>\*1</sup>の紹介があり、Debian で提供する Rust 関連のパッケージのインストール方法とよく利用するコマンド "cargo" の説明、発表の場でコーディングを行ってみることで参加者に Rust がどのようなプログラム言語であるかを伝えました。

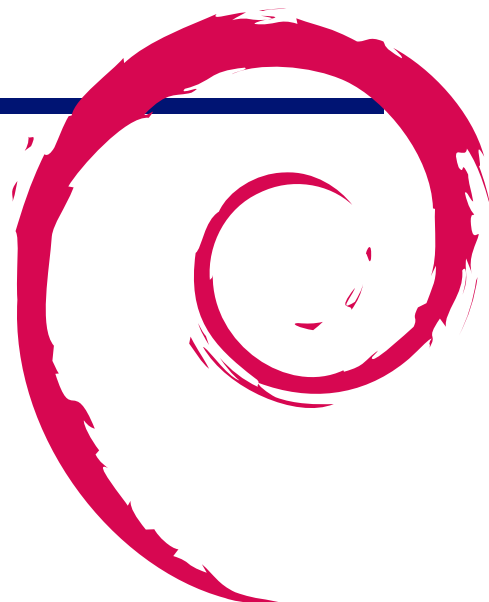
その後、Rust や Debian の最近の動向について情報交換を行いました。

---

<sup>\*1</sup> <https://doc.rust-jp.rs/>

## 2 事前課題

杉本 典充



今回の事前課題は以下です。

1. <https://mentors.debian.net/> の web サイトを知っていますか
2. debexpo という web アプリケーションを知っていますか
3. 次のプログラム言語でどのように debian パッケージが作られているか興味があるものを教えてください。

### 2.1 dictoss

1. 知っています
2. 知っています
3. go, golang, nodejs, python

### 2.2 koedoyoshida

1. 知っています
2. 知りません
3. python, その他

### 2.3 yosuke\_san

1. 知っています
2. 知りません
3. rust, python

### 2.4 tanykazy

1. 知っています
2. 知りません
3. go, golang, python, その他

### 2.5 YukiharuYABUKI

1. 使っています、または使ったことがあります
2. 知りません
3. python, その他

### 2.6 kenhys

1. 使っています、または使ったことがあります
2. 知っています
3. (回答なし)

### 2.7 ipv6waterstar

1. 使っています、または使ったことがあります
2. 知りません
3. emacs, その他

### 2.8 space320jt

1. 知りません
2. 知りません
3. python

### 2.9 yy\_y-ja\_jp

1. 知っています
2. 知っています
3. rust, go, golang, nodejs, emacs

### 2.10 dancerj

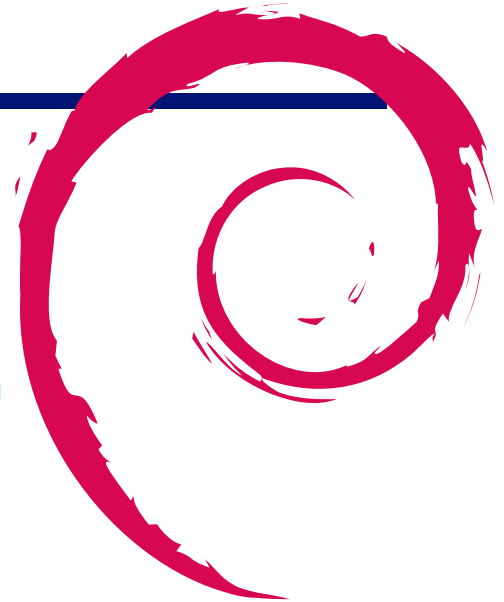
1. 使っています、または使ったことがあります
2. 知りません
3. nodejs, emacs

## 2.11 sambar\_gamer

1. 知っています
2. 知りません
3. python

## 2.12 henrich

1. 使っています、または使ったことがあります
2. 知っています
3. rust, golang, nodejs, python, ruby, perl



## 3 Debian での Python

杉本典充

### 3.1 はじめに

Debian では多くのプログラム言語の実行環境、ライブラリ及びアプリケーションのパッケージを多数提供しています。今回は Python について Debian ではどのようにパッケージをメンテナンスしているのかを調べてみました。

### 3.2 Python について

Python とは Debian などの linux 環境のみならず他の OS でも実行することが可能なマルチプラットフォームのプログラミング言語の 1 つです。2015 年に機械学習のプラットフォームである tensorflow がオープンソース化され、tensorflow には Python のライブラリが豊富なことから Python の人気が上がってきています。

Python の web サイト (<https://www.python.org/>) では初心者向けのチュートリアル<sup>\*2</sup>、Python の言語仕様やクラス・関数の仕様書<sup>\*3</sup>などの多くの情報を公開しています。特に開発者向けの情報では PEP (Python Enhancement Proposals)<sup>\*4</sup>と呼ばれる Python をよりよくするための提案の募集と公開しており<sup>\*5</sup>、コーディング規約やライブラリの廃止勧告、互換性の説明など多岐にわたる情報を公開しています。

### 3.3 Debian における Python 情報の取り扱いについて

#### 3.3.1 基本的な情報

Debian における Python 関連の情報は以下ページにまとまっています。

- Python - Debian Wiki <https://wiki.debian.org/Python/>
- Debian Python Policy <https://www.debian.org/doc/packaging-manuals/python-policy/>

Debian Project では、Python の実行環境をメンテナンスするチーム (cpython-team) とライブラリ及びアプリケーションをメンテナンスするチーム (Debian Python Team) の 2 つに分かれて活動しています。2 つのチームの主な違いを表 1 に示します。

「Debian Python Policy」<sup>\*6</sup> を読んでみて、パッケージの話で大事そうなところを以下に挙げてみます。

- python3 へ移行しましょう
- python のインタプリタのディレクトリ配置やファイル名の規則 (例: /usr/bin/python3、/usr/bin/python3.8)

<sup>\*2</sup> <https://www.python.org/about/gettingstarted/>

<sup>\*3</sup> <https://docs.python.org/3/>

<sup>\*4</sup> <https://www.python.org/dev/peps/>

<sup>\*5</sup> Debian にも DEP という似たような仕組みがあります。 <https://dep-team.pages.debian.net/>

<sup>\*6</sup> <https://www.debian.org/doc/packaging-manuals/python-policy/>

表 1 Debian Project における Python のチーム情報

チーム名	cpython-team	Debian Python Team (DPT)
役割	実行環境のメンテナンス	ライブラリとアプリケーションのメンテナンス
メーリングリスト	debian-python@lists.debian.org	(同左)
IRC	#debian-python	(同左)
wiki	-	<a href="https://wiki.debian.org/Teams/PythonTeam">https://wiki.debian.org/Teams/PythonTeam</a>
git リポジトリ	<a href="https://salsa.debian.org/cpython-team">https://salsa.debian.org/cpython-team</a>	<a href="https://salsa.debian.org/python-team">https://salsa.debian.org/python-team</a>

- python ライブラリのパスの規則 (debian パッケージでは /usr/lib/python3,2.7/dist-packages/ 配下に配置する)
- python のライブラリやアプリケーションの debian パッケージ名の命名規則 (python3 系用であれば python3-\* の形式とすると、モジュール名にアンダースコアが入っている場合は debian パッケージ名ではハイフンに置換すること)
- python プログラムについて
  - シェバンには #!/usr/bin/env interpreter\_name は使わないこと
  - #!/usr/bin/python3 または #!/usr/bin/python3.y として debian が提供するデフォルトのバージョンを指定すること
- python の debian パッケージ化を支援するための debhelper の拡張として dh-python パッケージがある

### 3.3.2 cpython-team について

cpython-team チームの [https://salsa.debian.org/groups/cpython-team/-/group\\_members](https://salsa.debian.org/groups/cpython-team/-/group_members) に表示されるメンバーは Matthias Klose (@doko) さん 1 名のみです。

Klose さんがメンテナンスしている <https://tracker.debian.org/pkg/python3.9> を確認してみると、maintainer にはチームでなく Klose さんの名前が書いてあります。そのため 1 人でメンテナンスしているようです。

2020 年 11 月時点の unstable においては python-2.7、3.8、3.9 のパッケージを提供しており、現在は python-3.9 のデバッグ作業を行っています\*7。

### 3.3.3 Debian Python Team (DPT) について

Debian Python Team (DPT) のメンバーは <https://wiki.debian.org/Teams/PythonTeam> に記載があり、2020 年 11 月現在では 20 名でメンテナンスしているようです。チームでメンテナンスしているパッケージの一覧は次の URL で見ることができます。

- <https://qa.debian.org/developer.php?email=team%2Bpython%40tracker.debian.org>

Debian Python Team (DPT) に参加するには チームのポリシードキュメントを読み、応諾した上で参加することを求めています。チームのポリシーを記載したドキュメントは以下のページにあります。

- <https://salsa.debian.org/python-team/tools/python-modules/blob/master/policy.rst>

このページに記載しているポリシーにはおおよそ以下の内容が書かれています。

- チームへ参加方法
  - 以下 3 つを書いたメールを [debian-python@lists.debian.org](mailto:debian-python@lists.debian.org) へ送信する

\*7 <https://bugs.debian.org/cgi-bin/pkgreport.cgi?tag=python3.9;users=debian-python@lists.debian.org>

- \* なぜ参加したいのか
- \* <https://salsa.debian.org> にログインするアカウント名
- \* ポリシードキュメント ( policy.rst ) を読み、受諾することの宣言
- パッケージメンテナンスについて
  - debian パッケージ内の debian/control ファイルの Maintainer フィールド及び Uploaders フィールドの意味
  - git-buildpackage を使ったパッケージメンテナンスの説明

また、Python のパッケージを作るにあたっては debian/control、debian/rules などのファイルの書き方の説明が以下のページ記載されています。

- <https://wiki.debian.org/Python/LibraryStyleGuide>
- <https://wiki.debian.org/Python/AppStyleGuide>

### 3.3.4 pypi パッケージを debian パッケージに変換する

プログラム言語にはそれぞれライブラリやアプリケーションをパッケージ化して配布するサイトを持っていることが多いです。そして Debian ではそれぞれのプログラム言語のライブラリを debian パッケージへ変換する debian パッケージを配布しており、以下のページに情報があります。

- <https://wiki.debian.org/AutomaticPackagingTools>

Python におけるライブラリやアプリケーションの配布サイトは <https://pypi.org/> が有名です。上記 URL で紹介している Python 向けの debian パッケージへの変換ツールは「python3-stdeb」パッケージに入っている「py2dsc-deb」コマンドがよさそうに思いました\*8。

試しに py2dsc-deb コマンドを使って pypi が配布するパッケージを debian パッケージに変換してみました。debian/compat バージョンは "9" として変換しているようです。

```
$ sudo apt-get install python3-stdeb python-all dh-python
$ wget https://files.pythonhosted.org/packages/53/20/4019a739b2eefe9282d3822ef6a225250af964b117356971bd55e274193c/ws4py-0.5.1.tar.gz
$ py2dsc-deb ws4py-0.5.1.tar.gz
$ ls -l deb_dist
l 合計 128
-rw-r--r-- 1 norimitu norimitu 38224 11月 19 13:58 python3-ws4py_0.5.1-1_all.deb
drwxr-xr-x 8 norimitu norimitu 4096 11月 19 13:58 ws4py-0.5.1
-rw-r--r-- 1 norimitu norimitu 916 11月 19 13:58 ws4py_0.5.1-1.debian.tar.xz
-rw-r--r-- 1 norimitu norimitu 869 11月 19 13:58 ws4py_0.5.1-1.dsc
-rw-r--r-- 1 norimitu norimitu 5480 11月 19 13:58 ws4py_0.5.1-1_amd64.buildinfo
-rw-r--r-- 1 norimitu norimitu 1024 11月 19 13:58 ws4py_0.5.1-1_amd64.changes
-rw-r--r-- 1 norimitu norimitu 5419 11月 19 13:58 ws4py_0.5.1-1_source.buildinfo
-rw-r--r-- 1 norimitu norimitu 1377 11月 19 13:58 ws4py_0.5.1-1_source.changes
-rw-r--r-- 2 norimitu norimitu 51408 2月 26 2020 ws4py_0.5.1.orig.tar.gz

$ cat deb_dist/ws4py-0.5.1/debian/compat
9
```

また、python のパッケージをインストールするコマンド "pip" でインストールしたときに binding ライブラリのコンパイルも一緒に行うライブラリ "mysqlclient" を試してみると、以下のコマンドで debian パッケージに変換できました。

\*8 pypi2deb パッケージは依存パッケージがとても多く perl 関連のパッケージが大量にインストールされます。



```

$ sudo apt-get install python3-stdeb python3-all-dev dh-python libmysqlclient-dev
$ wget https://files.pythonhosted.org/packages/a5/e1/e5f2b231c05dc51d9d87fa5066f90d1405345c54b14b0b11a1c859020f21/mysqlclient-2.0.1.tar.gz
$ py2dsc-deb mysqlclient-2.0.1.tar.gz
$ ls -l deb_dist
合計 264
drwxr-xr-x 11 norimitu norimitu 4096 11 月 19 14:15 mysqlclient-2.0.1
-rw-r--r-- 1 norimitu norimitu 1236 11 月 19 14:15 mysqlclient_2.0.1-1.debian.tar.xz
-rw-r--r-- 1 norimitu norimitu 927 11 月 19 14:15 mysqlclient_2.0.1-1.dsc
-rw-r--r-- 1 norimitu norimitu 6192 11 月 19 14:15 mysqlclient_2.0.1-1_amd64.buildinfo
-rw-r--r-- 1 norimitu norimitu 1394 11 月 19 14:15 mysqlclient_2.0.1-1_amd64.changes
-rw-r--r-- 1 norimitu norimitu 5795 11 月 19 14:15 mysqlclient_2.0.1-1_source.buildinfo
-rw-r--r-- 1 norimitu norimitu 1464 11 月 19 14:15 mysqlclient_2.0.1-1_source.changes
-rw-r--r-- 2 norimitu norimitu 87807 7 月 3 03:55 mysqlclient_2.0.1.orig.tar.gz
-rw-r--r-- 1 norimitu norimitu 90744 11 月 19 14:15 python3-mysqlclient-dbg_2.0.1-1_amd64.deb
-rw-r--r-- 1 norimitu norimitu 48584 11 月 19 14:15 python3-mysqlclient_2.0.1-1_amd64.deb

$ cat deb_dist/mysqlclient-2.0.1/debian/compat
9

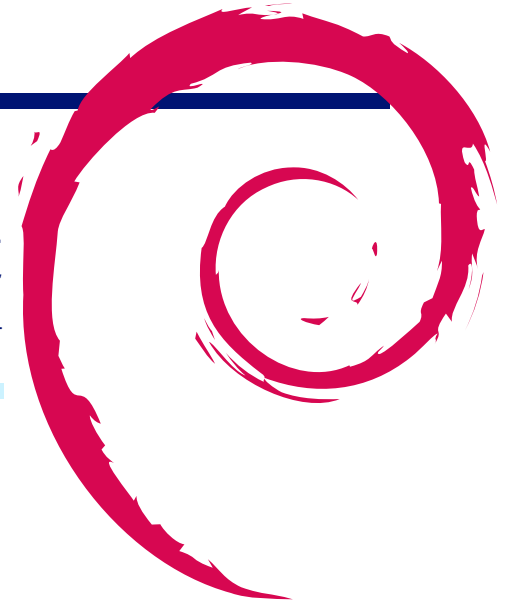
```

### 3.4 終わりに

debian における Python パッケージをどういうチームがどのようにパッケージを作成、メンテナンスしているか調べてみました。皆さんも Python や他のプログラム言語のライブラリやアプリケーションを debian パッケージ化し、debian ライフをお楽しみください。

### 3.5 参考情報

- <https://wiki.debian.org/Python/>
- <https://www.debian.org/doc/packaging-manuals/python-policy/>
- <https://salsa.debian.org/python-team/tools/python-modules/blob/master/policy.rst>
- <https://wiki.debian.org/Python/LibraryStyleGuide>
- <https://wiki.debian.org/Python/AppStyleGuide>



## 4 Debian 勉強会資料のディレクトリ構成変更

上川純一

### 4.1 Debian 勉強会資料ディレクトリ構成変更の目的

Debian 勉強会開始当初はそこまでファイル数が多くなかったのでディレクトリ構成はシンプルで、画像などはその月のサブディレクトリに置いて TeX ファイルやスタイルファイルなどはトップレベルに配置するというようにしていました。しかし 15 年たった現在においては 800 以上のファイルが初期状態で配置され、TeX コンパイルすると中間ファイルをふくめて 10000 近いファイル数が配置されるようになってきました。月に TeX ファイルは 4 個くらい追加されるので状況はどんどん悪くなります。

- debianmeetingresumeYYYYMM.tex – TeX ファイル
- debianmeetingresumeYYYYMM-kansai.tex – 関西の勉強会の TeX ファイル
- debianmeetingresumeYYYYMM-presentation.tex – プレゼンテーション用の TeX ファイル
- debianmeetingresumeYYYYMM-kansai-presentation.tex – 関西の勉強会の TeX ファイル
- imageYYYYMM/ – 画像ファイルなどの配置場所

また、初めてチェックアウトした人が Make コマンドをうつと 8 コアの十分速い最近のマシンでも 15 分くらいコンパイルにかかります。遅めのマシンだと数時間かかるということもあるようです。Git commit 際にビルドできるかどうかチェックするために pre-commit-hook を使うという運用にしていたのですが、状況によっては Git commit に長時間待たされることになるのです。全体に影響を及ぼす大掛かりな変更などもできませんでした。それは流石に不便すぎるので今回ソースを年ごとに分割してみることにしました。

### 4.2 ディレクトリ構成変更の実際

ディレクトリ構成は新しいファイルについては以下のように変更しました。

- YYYY/Makefile – そのサブディレクトリ用の Makefile。
- YYYY/debianmeetingresumeYYYYMM.tex – TeX ファイル
- YYYY/debianmeetingresumeYYYYMM-kansai.tex – 関西の勉強会の TeX ファイル
- YYYY/debianmeetingresumeYYYYMM-presentation.tex – プレゼンテーション用の TeX ファイル
- YYYY/debianmeetingresumeYYYYMM-kansai-presentation.tex – 関西の勉強会の TeX ファイル
- YYYY/imageYYYYMM/ – 画像ファイルなどの配置場所

チェックアウトしたあとの一連の作業は以下になると思います。

```
$ cd 2020
$ edit debianmeetingresume202011.tex
$ make
$ git commit
$ git pull --rebase
$ git push
```

今までディレクトリ構成の変更については何度か途中で諦めているので今回はできるだけ挫折しないようにやりやすいところからはじめています。まず2020年からはじめました。2017年までさかのぼって変更しました。また方針として、リファクタリングは後回しして必要なファイルはシンボリックリンクで持ってきています。2016年より以前はそれ以降の年から依存されているファイルがあるので移動するにはなんらかのリファクタリングが必要になるかと思います。utils/mkdirlinks.sh にスクリプトをおいていますが、使いまわしている画像ファイルなどによって依存関係が結構ありました。

```
beamerthemeKansaiDebian.sty
beamerthemeKyoto.sty
beamerthemeTokyo.sty
image200502
image200607
image200703
image200707
image200802
image201006
image2012-natsu
kansaimonthlyreport.sty
monthlypresentation.sty
monthlyreport.sty
```

今後の課題としては、再利用しやすい画像やスタイルファイルなどについては再利用専用のディレクトリなどに分割するのが良いかなと思っています。

#### 4.2.1 UTF-8 化

全員が全部のファイルをビルドさせられる状態だと文字コードの一括変換等はあまりやりたくなかったのですが、今回の変更で基本的には2020/サブディレクトリだけをビルドし直せば良いようになったので一括でUTF-8に変更しました。

スクリプトを書いて find + xargs で実行しました。utils/convert\_to\_utf8.sh

```
$ find -name '*.tex' | xargs bash ./utils/convert_to_utf8.sh
$ find -name '*.sty' | xargs bash ./utils/convert_to_utf8.sh
```

#### 4.2.2 CI の有効化

CI を有効にしたかったのですが大きすぎておそすぎるという問題がありました。Github にミラーを作ってそこから Cloud Build 連携で力技で8コアの Cloud Build インスタンスを利用して15分以内にビルドできています。結果一回30円くらいだったと思います。

Salsa の CI <sup>\*9</sup> では最近のサブディレクトリ(2020, 2021)だけをビルドするようにして試してます。dictoss さん提供の gitlab runner 経由で7分でビルドが終わっているようです。

本当は salsa の master に push される前にビルドのチェックを走らせたいのですができるのかな？

CI の設定ファイルが複数あるのも今後の課題です。

- cloudbuild.yaml – cloud build で使っている。
- utils/docker/cloudbuild.yaml – docker image 作成用
- .gitlab-ci.yml – salsa の設定
- debian/control – 2013年ころまで pbuilder ベースで走らせていた CI 用、現在は何に利用しているのか？

最初は Debian 環境に LaTeX 環境のインストールとセットアップからするように設定していたのですが、様子を見ている限りだと LaTeX 関連のパッケージに何分もかかっているようでした。Docker Image をたまに更新するよう

---

\*9 .gitlab-ci.yml 参照

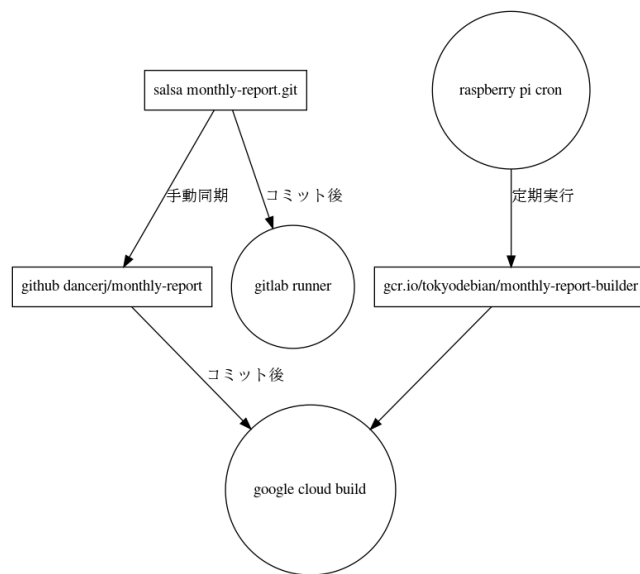


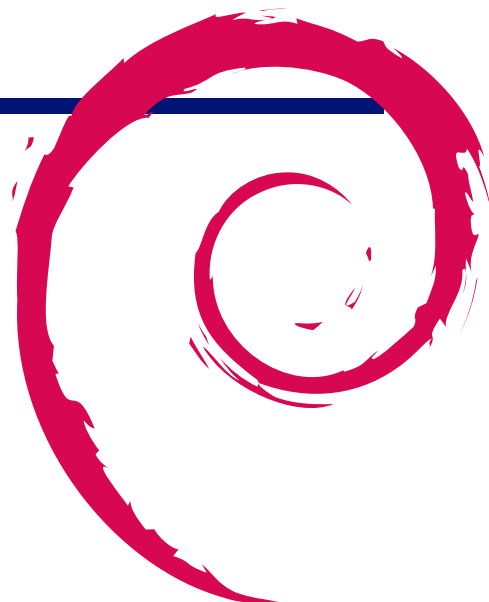
図 1 CI の流れ

にしてそれを利用すればより高速になる気がしており、試しに Docker Image を事前に作成するようにしたら Cloud Build の時間が 13 分くらいに短縮されました。

Docker Image はとりあえず手元の raspberry pi 3 で定期的に更新するようにしています。

## 5 メモ

---









**Debian 勉強会資料**

2020年11月21日 初版第1刷発行

東京エリア Debian 勉強会（編集・印刷・発行）

---