

、 (・ 肉 ・)
ノ ログ と Docker

タイトルズレました

^_^

o/^(.;-▽)つ ミ ギャハッハッズレてる!ズレてる!
と_)__つノ ☆ バンバン

^_^

o/^(°Д°)つ
と_)__つノ

^_^

o/^(°Д°)つ
と_)__つノ

自己紹介

- にく
- I like
 - コンサドーレ札幌
 - Emacs
 - Ruby
 - Elixir/Phoenix

自己紹介

- github_id :: niku
- twitter_id :: niku_name
- web :: <http://niku.name/>

今日話すこと

- <http://niku.name/>
 - どうやって表示しているか
 - どうやってコンテンツ更新しているか

どうやって表示しているか

構成 : Debian - Apache - Rack - 独自アプリケーション nwiki - コンテンツ nikulog

- Apache Webサーバー
- Rack サーバーとアプリケーションを繋ぐミドルウェア
- nwiki git のファイルの内容を web に表示する
- nikulog コンテンツ (git管理)

つらみ

- アプリケーション開発者にとっての趣味サーバー管理
- サーバー管理者にとっての趣味アプリケーション開発(?)
- 「これ、どうやるんだったっけ……」

つらみ

- 知識としてはある
- たまにしかやらないので毎回調べるはめになる
- [apache 設定 やり方] [検索]

つらみ対策

プロビジョニングツール

- chef / itamae レシピ書けます
- いいものだ
- (僕は)サーバー管理の手数を減らしたい
- 選定や知識のアップグレードの時間を投資し続けることができるだろうか？

つらみ対策

わかれていると管理しやすいのでは

- ミニマム構成のサーバー部分
 - 穴が少ない
- 気軽にアップデートでき、動作検証しやすいアプリケーション

つらみ対策

- ミニマム構成のサーバー => Docker だけインストールした Debian
- 気軽にアップデートでき、動作検証しやすいアプリケーション => Dockerイメージ

Dockerイメージを載せるサーバー

さくらVPSにDebian8をインストールして、コマンドを手で打ちました

```
$ sudo update-locale LANG=C.UTF-8
$ sudo sh -c "echo 'deb http://ftp.jp.debian.org/debian jessie-backports main' \  
> /etc/apt/sources.list.d/jessie-backports.list"
$ sudo apt-get update
$ sudo apt-get install -y docker.io
```

Docker

構成 : Debian - Apache - Rack - 独自アプリケーション nwiki - コンテンツ nikulog

niku/debian

- PackerでOSイメージを自分で作る方法を知っていた
- Packerを使ってVagrantのBoxを作る方法を一つずつ説明する
- \ (`・肉・ `) ノログ
- (URL長くて入らなかった)

niku/debian

OS イメージ作成, サーバー起動,
サーバーログイン

```
% git clone https://github.com/niku/uti.git
% cd uti/share/lib/server/host/
uti/share/lib/server/host% packer build jessie64.json
uti/share/lib/server/host% vagrant up
uti/share/lib/server/host% vagrant ssh
```


niku/debian

Debianサーバー内でDockerイメージ生成, DockerHubへpush

```
$ uname -a
Linux packer-virtualbox-iso-1430355314 3.16.0-4-amd64 #1 SMP \
Debian 3.16.7-ckt9-3-deb8u1 (2015-04-24) x86_64 GNU/Linux
$ cat /etc/debian_version
8.0
$ sudo sh -c 'echo "deb http://ftp.jp.debian.org/debian sid main" > /etc/apt/sources.list.d/sid.list'
$ sudo apt-get update
$ sudo apt-get install -y docker.io git debootstrap
$ git clone https://github.com/docker/docker.git
$ sudo docker/contrib/mkimage.sh \
--dir debian-image \
--tag niku/debian \
debootstrap --variant=minbase \
jessie http://ftp.jp.debian.org/debian
$ cd debian-image
$ sudo docker login --username="niku" --email="my_email_address@example.com" --password="my_password"
$ sudo docker push niku/debian
```

niku/nwiki

AutomatedBuild

- Github リポジトリを更新
- Github から Dockerhub へ Webhook でお知らせ
- Dockerhub が Dockerfile を元に Docker イメージをビルド

niku/nwiki

やりかた

- Dockerfile のある github リポジトリを Dockerhub に AutomatedBuild で登録

The screenshot shows the Dockerhub interface for configuring Automated Builds for the repository 'niku/nwiki'. The page title is 'Edit Automated Build settings for niku/nwiki'. Under the 'Repo Name' section, the name 'niku/nwiki' is entered in a text field. A note below states: 'You can't change the Repo Name after the build has been created'. In the 'Active' section, there is a checked checkbox. Below this, it says 'When active we will build when new pushes occur'. At the bottom, there are four input fields: 'Type' (a dropdown menu set to 'Branch'), 'Name' (a text field containing 'master'), 'Dockerfile Location' (a text field containing '/'), and 'Docker Tag Name' (a text field containing 'latest'). A blue 'Save' button is located to the right of the 'Docker Tag Name' field.

Search... Browse Repos Documentation Community Help niku

Edit Automated Build settings for niku/nwiki

Repo Name:
niku/nwiki

You can't change the Repo Name after the build has been created

Active:

When active we will build when new pushes occur

Type	Name	Dockerfile Location	Docker Tag Name
Branch	master	/	latest

niku/nwiki

やりかた

- github の webhook に Dockerhub を指定

The screenshot shows the GitHub repository settings for 'niku / nwiki'. At the top right, there are buttons for 'Unwatch' (1), 'Unstar' (4), and 'Fork' (1). On the left sidebar, the 'Webhooks & Services' section is selected. The main content area is divided into two sections: 'Webhooks' and 'Services'. The 'Webhooks' section has an 'Add webhook' button and a description: 'Webhooks allow external services to be notified when certain events happen within your repository. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).' The 'Services' section has an 'Add service' button and a description: 'Services are pre-built integrations that perform certain actions when events occur on GitHub. For more information on services check out our [Service Hooks Guide](#).' Below the 'Services' section, a list of services is shown, with 'Docker' selected and highlighted by a red box. To the right of the 'Docker' entry are edit and delete icons.

niku/nwiki

nwiki/Dockerfile

```
FROM niku/debian
MAINTAINER niku

RUN apt-get update && \
    apt-get -y install build-essential \
    ruby2.1-dev \
    cmake \
    pkg-config \
    git \
    apache2 \
    libapache2-mod-passenger && \
    echo "(設定ファイルの内容)" > /etc/apache2/sites-available/nwiki.conf && \
    a2dissite 000-default && \
    a2ensite nwiki && \
    git clone https://github.com/niku/nwiki.git /var/www/nwiki && \
    cd /var/www/nwiki && gem install bundler && bundle install

EXPOSE 80 443
CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

niku/nikulog

同様にAutomatedBuildを利用

niku/nikulog

nikulog/Dockerfile

```
FROM niku/nwiki
MAINTAINER niku

RUN git clone https://github.com/niku/nikulog /var/www/nikulog && \
    cd /var/www/nikulog && \
    git checkout config && \
    git checkout master && \
    git fetch && \
    git reset --hard origin/master && \
    echo "(設定ファイルの内容)" > /var/www/nwiki/config.ru && \
    mkdir -p /var/www/nikulog/tmp && \
    touch /var/www/nikulog/tmp/restart.txt
```

サーバーとイメージ準備完了

```
$ sudo docker pull niku/nikulog:latest  
$ sudo docker run --publish 80:80 --name nikulog -d niku/nikulog
```

で表示できるようになった (^o^)/

niku.name 更新のしくみ

更新は？ $\Sigma(\square;)$

niku.name 更新のしくみ

Docker導入以前

```
git
|
| Push
|
niku.nameのgit
|
| gitのhookスクリプト
|
github (niku/nikulog)
```

niku.name 更新のしくみ

参考にした

- DockerHubのAutomated Buildsをフックして最新のDockerコンテナをデプロイする | SOTA
- <http://deeeet.com/writing/2015/01/08/dockerhub-hook/>

niku.name 更新のしくみ

Docker導入後

```
git
|
| Push
|
github (niku/nikulog)
|
| githubのWebhook
|
Dockerhub (niku/nikulog)
|
| DockerimageのAutomatedBuild
| DockerhubのWebhook
|
niku.nameのとあるポート
|
| (docker pull して再起動)
|
niku/nikulog 最新のdocker
```

niku.name 更新待受

[Browse Repos](#)[Documentation](#)[Community](#)[Help](#)

niku

AUTOMATED BUILD REPOSITORY

Updated 3 days, 16 hours ago

niku / nikulog

Pull this repository

docker pull niku/nikulog

No description set

☆ 0 🗨 0 📄 54

<- basic info

Webhooks

Add Webhook

Webhooks will be called after a successful repository push is made. The webhook call will be a HTTP POST with the following JSON payload (shown below). If you want to test it out, we recommend using a tool like [requestb.in](#)

When creating a webhook chain, subsequent URLs will be called once the previous callback has been contacted (by POSTing a [valid JSON payload](#) to the `callback_url` found in the request body).

See the [docs](#) for more detail.

> Click to Show Example Response

56535ac9-f2ba-4aad-b407-9447b4d2d546

Edit

Test

`http://niku.name:4000/refresh-nikulog`

niku.name 更新待受

Webhook待受サーバー

- bketelsen/captainhook
- <https://github.com/bketelsen/captainhook>
- HTTPリクエストがくると設定したjsonのとおりコマンドを実行する

niku.name 更新待受

サーバーに追加でアプリケーションを入れたくない

- niku/yobirinex
- <https://github.com/niku/yobirinex>
- 実行環境(ErlangVM)も同梱できるのでtarファイル展開した以下で完結する

niku.name 更新待受

ErlangVM同梱HTTPサーバーの作りかた

- ElixirでHTTPのやりとりをする
- `http://niku.name/articles/2015/07/03/Elixir%E3%81%A7HTTP%E3%81%AE%E3%82%84%E3%82%8A%E3%81%A8%E3%82%8A%E3%82%92%E3%81%99%E3%82%8B`

niku.name 更新待受

更新用設定ファイル

```
% cat configdir/refresh-nikulog.json
{
  "scripts": [
    {
      "command": "sudo",
      "args": ["docker", "pull", "niku/nikulog:latest"]
    },
    {
      "command": "sudo",
      "args": ["docker", "kill", "nikulog"]
    },
    {
      "command": "sudo",
      "args": ["docker", "rm", "nikulog"]
    },
    {
      "command": "sudo",
      "args": ["docker", "run", "--publish", "80:80", "--name", "nikulog", "-d", "niku/nikulog"]
    }
  ]
}
```

niku.name 更新待受

```
git
|
| Push
|
github (niku/nikulog)
|
| githubのWebhook
|
Dockerhub
|
| DockerimageのAutomatedBuild
| DockerhubのWebhook
|
niku.nameのとあるポート
|
| (docker pull して再起動)
|
niku/nikulog 最新のdocker
```

その他/話せなかったこと1

- Dockerhub のビルド遅くて数分から十数分待つことがある。急いでないのので別に問題にしてい
- Webhook飛んでこないことある。Dockerhubならwebから手動でビルド, niku.nameならローカルからcurlでリクエストしてる

その他/話せなかったこと2

- docker-compose使っていない
Vagrant の docker プロビジョンでまだ対応していない
- Vagrant から Docker を使ったときの docker exec デバッグのやりかたを書きたかったけど時間なかった
- Packerでイメージつくるときの DEBCONF_DEBUG=5の有用さ

まとめ

- niku.nameの表示
 - Dockerイメージ
- niku.nameの更新
 - GithubのWebhook
 - DockerHubのAutomatedBuild
 - DockerHubのWebhook
 - 独自Webhook待受サーバー