

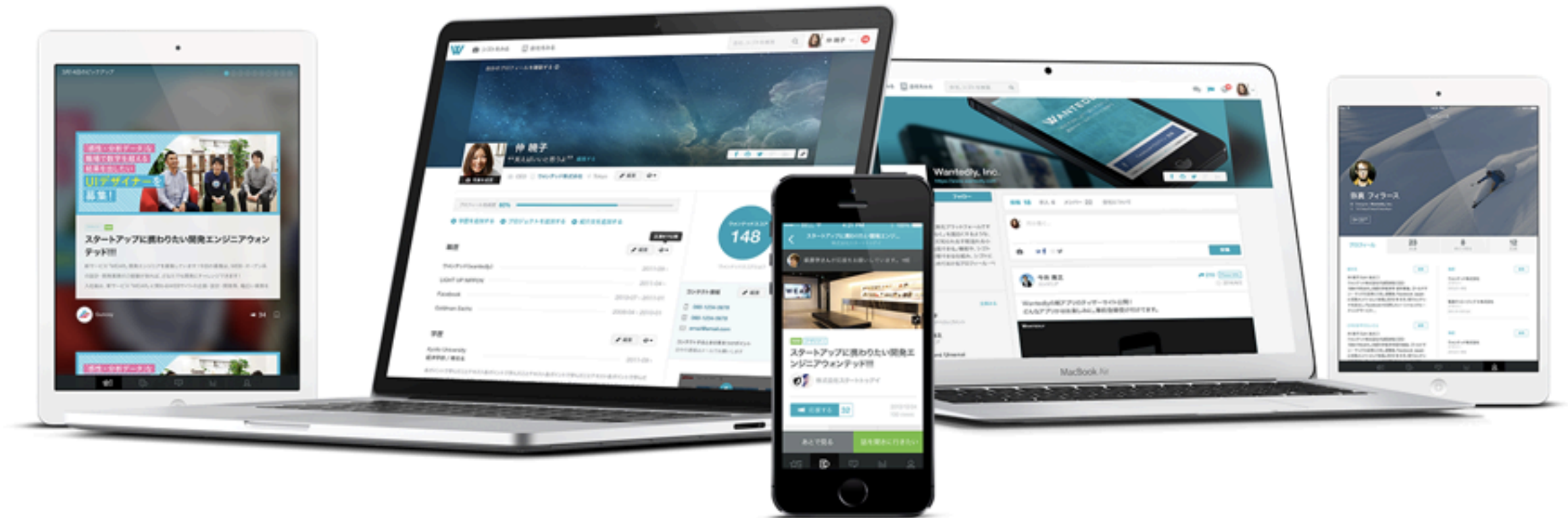
NGINX
IMAGE
SERVER

SEIGO UCHIDA



@spesnova

WANTEDLY



今日の話

nginx_small_light の利用事例

利用箇所と導入した経緯

募集ページ (リサイズ)

日本での「GitHub」活用を促進させるGitHubberをウォンテッド! - GitHubの求人 - Wantedly

W 募集をみる 会社をみる フィードをみる ユーザーや会社、募集を検索 ★気になるリスト

中途採用 > 日本 > セールス

セールス

日本での「GitHub」活用を促進させるGitHubberをウォンテッド!

 GitHub

埋め込む いいね! 64 Tweet 95 Bookmark 4



2015/01/17 社員とあなたの共通の友達 (12) +7 4916 応援する 63

会社アイコン（上下に余白）

ヤフー株式会社の最新情報 - Wantedly

募集をみる 会社をみる フィードをみる

ユーザーや会社、募集を検索

★気になるリスト

 ヤフー株式会社
<http://www.yahoo.co.jp/>

フォローする 64人のフォロワー

ヤフー株式会社に働いています

Tweet いいね! 9

社員とあなたの共通の友達  + 156

設立年月 1996年1月 社員数 4514人

会社について 投稿 募集 (11) メンバー (134) 統計情報

ヤフー株式会社の募集

ヤフー地図の革新をあなたの手

会社アイコン (上下に余白)



プロフィール背景（ブレンダー）

The image shows a screenshot of a user profile on the Wantedly website. The profile background is a blurred image of a person, highlighted with a red border. A red arrow points from the bottom left towards the background, and another red arrow points from the bottom right towards the background. The profile information includes the name 内田 誠悟 (Uchida Makiko), a bio "うっちーさんの優しさで、心のインフラを整えてもらいたいです。", and a quote "Just Do It". The user is identified as an Engineer at Wantedly Inc. The profile has 77 connections and 23 introductions. The skills section lists 30 Oshare Engineers, 28 Hubot, and 26 Docker. The social media section shows icons for Facebook, Twitter, and LinkedIn. A large blue circle displays the Wantedly Score of 90.

内田 誠悟 プロフィール - Wantedly

Wantedly

募集をみる 会社をみる フィードをみる

ユーザーや会社、募集を検索

★気になるリスト

自分のプロフィールを編集する

うっちーさんの優しさで、心のインフラを整えてもらいたいです。

内田 誠悟

“Just Do It”

エンジニア Wantedly Inc.

プロフィール 77 つながり 23 紹介文

ソーシャル

フォンテッド・スコア 90

スキル・特徴

- 30 オシャレエンジニア
- 28 Hubot
- 26 Docker

会社ページ背景（ブレンダー）

Wantedly, Inc.の最新情報 - Wantedly

募集をみる 会社をみる フィードをみる ユーザーや会社、募集を検索 ★気になるリスト

Wantedly, Inc.
http://site.wantedly.com

フォローする 471人のフォロワー +社員を招待 Tweet いいね! 290

社員とあなたの共通の友達 (422) + 416

会社について 投稿 募集 (16) メンバー (42) 統計情報

ここに書く...

写真を選択する Twitter Facebook 投稿する

なにを投稿すればよいですか?

私たちについて

「シゴトでココロオドル人をふやす」というミッションを掲げ、シゴトSNSのWantedly（ウオンテッドリー）を展開しています。Wantedlyには、シゴト仲間とつながり、つながりを管理する機能や、自分のシゴトに関する情報をまとめておけるプロフィールページの他、会社に遊びにいける機能があります。

モバイル (WebP)



※ 2015/02/18 時点でAppStore, Google Play に申請中

通信帯域により取得サイズを変更



Quality	Size
良い	100KB
悪い	25KB
すごく悪い	6KB 16分の1

37 of 45

iOS 通信のパフォーマンス改善 ・ iOSオールスターズ登壇資料

20,640 views

Youhei Sugigami (4 SlideShares) , エンジニア at Wantedly
+ Follow

nginx_small_light 導入の経緯

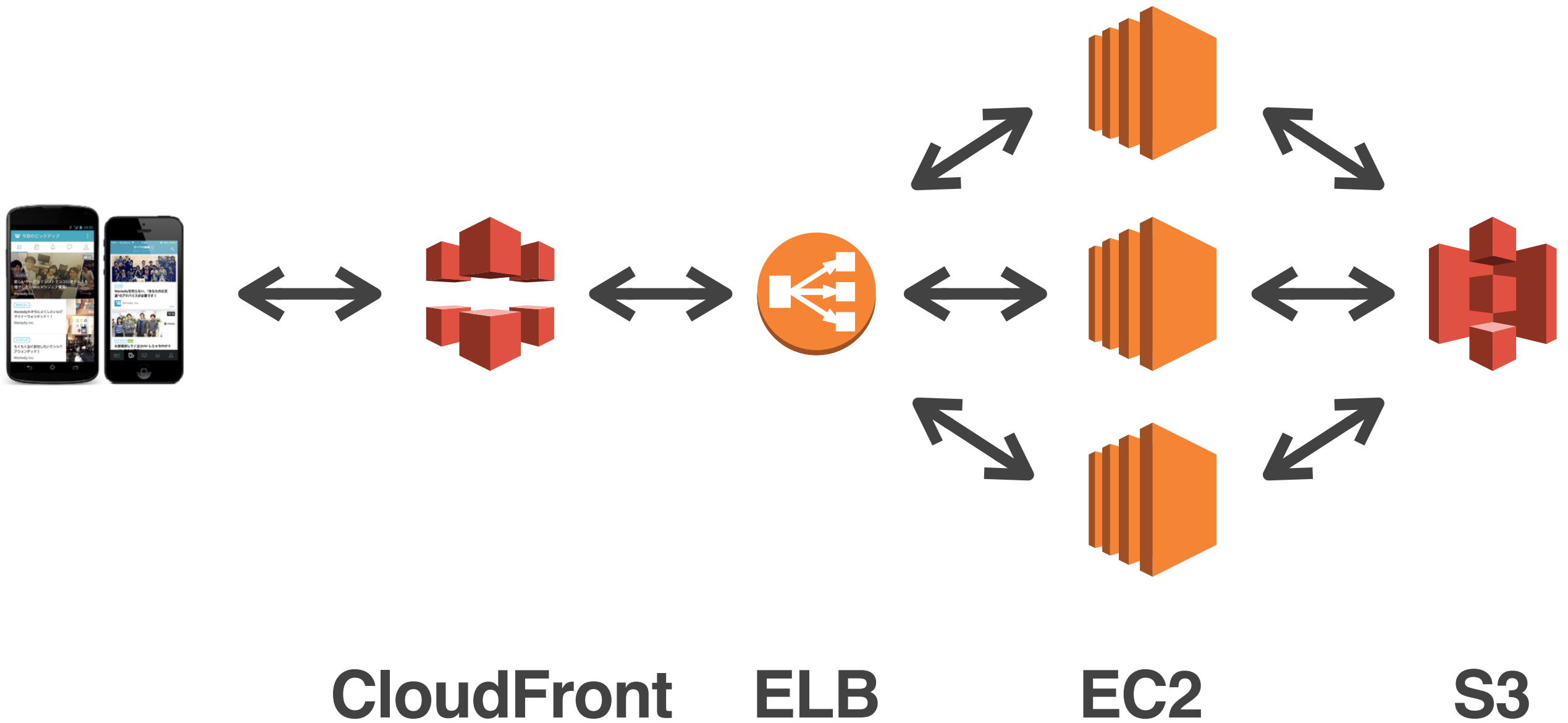
- バックグラウンドジョブ
- 時間かかる
- 動的にリサイズしたい

WebP 導入の経緯

- **Wantedly 海外進出**
- **ネットワーク環境の違い**
- **通信量の改善**
- **モバイルチームの支援**

サーバ構成

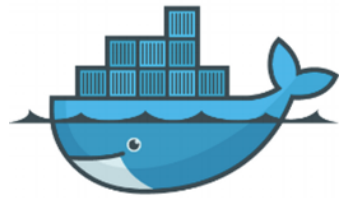
NGINX IMAGE SERVER



サーバ構成



Nginx コンテナ



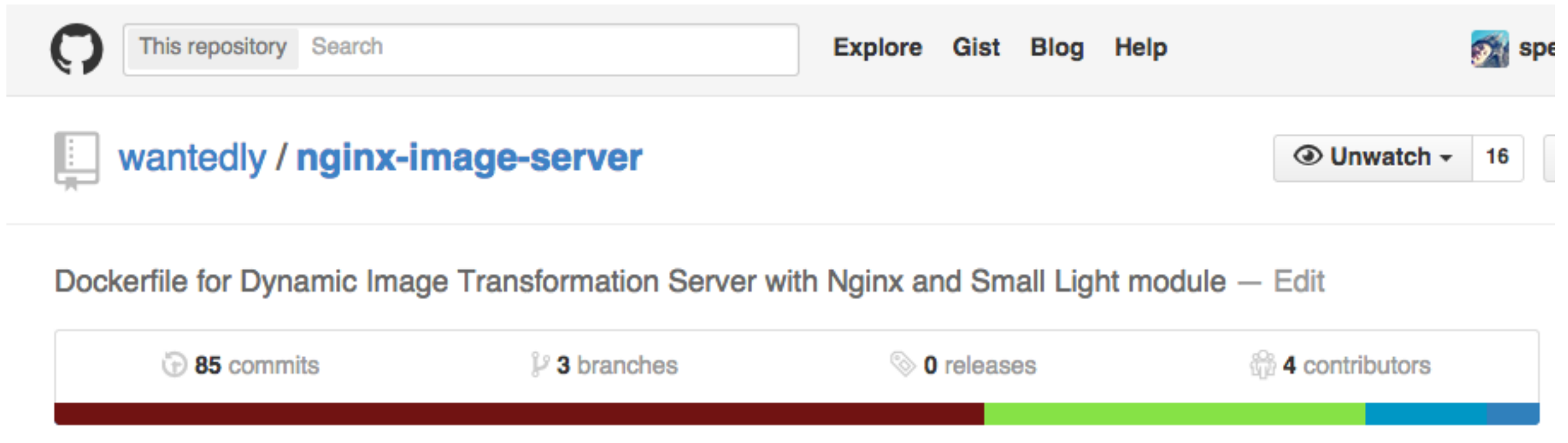
Docker デーモン



CoreOS サーバ

画像サーバの構築と運用

GitHub Flow



1. Vagrant で CoreOS VM の起動
2. Dockerfile や nginx.conf の編集
3. ビルドとテスト
4. Pull Request

コンテナのCI (infrataster)

The screenshot displays the Wercker CI interface for a build. The browser window title is "wercker". The breadcrumb navigation is "HOME / APPS / WANTEDLY / NGINX-IMAGE-SERVER / #2FEDDF5". A green banner at the top indicates "Build passed on 1 nginx_small_light-v0.6.4" with a "Deploy to" button. Below this, it shows the build was performed by "SPESNOVA" about a month ago, with a total time of 17 minutes and 27 seconds. The "Commit" section shows a commit with the message "#2feddf5 Use nginx_small_light v0.6.4 to fix semaphore error" and a link to a GitHub pull request. The "Build steps" section lists four successful steps: "get code" (0 sec), "setup environment" (1 min 9 sec), "environment variables" (2 sec), and "Check docker version" (1 sec). A help icon is visible in the bottom right corner.

wercker

HOME / APPS / WANTEDLY / NGINX-IMAGE-SERVER / #2FEDDF5

✓ Build passed on 1 nginx_small_light-v0.6.4 [Deploy to](#)

BY SPESNOVA, ABOUT A MONTH AGO ⌚ TOTAL TIME OF 17 MIN 27 SEC

Commit [View changes »](#)

#2feddf5 Use nginx_small_light v0.6.4 to fix semaphore error
See https://github.com/cubicdaiya/nginx_small_light/pull/20 for more about v0.6.4 - *Seigo Uchida*

Build steps [Refresh](#)

- ✓ get code 0 sec [Expand](#)
- ✓ setup environment 1 min 9 sec [Expand](#)
wercker-labs/docker defined in wercker.yml.
- ✓ environment variables 2 sec [Expand](#)
- ✓ Check docker version 1 sec [Expand](#)

?

Merge => Quay.io をフック

Use ngx_small_light v0.6.4 by spesnova · Pull Request #16 · wantedly/nginx-image-server · GitHub

GitHub This repository Search Explore Features Enterprise Blog Sign up Sign In

wantedly / nginx-image-server Watch 16 Star 31 Fork 0

Use ngx_small_light v0.6.4 #16

Merged awakia merged 1 commit into master from ngx_small_light-v0.6.4 on Jan 15

Conversation 2 Commits 1 Files changed 2 +2 -2

spesnova commented on Jan 15 Owner

WHY

To fix semaphore error
See [cubicdaiya/nginx_small_light#20](#) for more about v0.6.4 and the error.

Use ngx_small_light v0.6.4 to fix semaphore error 2feddf5

dtan4 was assigned by spesnova on Jan 15

spesnova commented on Jan 15 Owner

@dtan4 Please review this :)
CC: @awakia

awakia commented on Jan 15 Owner

Labels: None yet
Milestone: No milestone
Assignee: dtan4
3 participants

Quay.io で自動ビルド

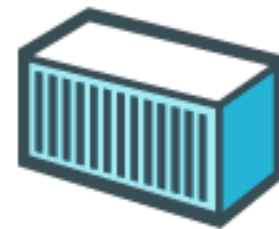
The screenshot displays the Quay.io web interface for a repository named 'wantedly/nginx-image-server'. The page title is 'Build History'. A green button labeled '+ New Dockerfile Build' is visible in the top right corner. The main content area shows a list of build IDs on the left: e029f7c, cc42ae4 (selected), 21cf6a8, 977c7bb, and 8437459. The selected build 'cc42ae4' is detailed in a light blue box, indicating it was 'Triggered by: Push to GitHub repository wantedly/nginx-image-server'. Below this, a status bar shows 'Dockerfile build completed and pushed' with a 'Build Package' icon and 'Started: 15 days ago'. A large black box contains the build logs, showing steps like 'Priming cache for build', 'Pulling base image', and 'Building image from Dockerfile'. The logs include commands such as 'FROM quay.io/wantedly/buildpack-deps:14.04', 'ENV NGINX_VERSION 1.6.2', 'ENV NGX_SMALL_LIGHT_VERSION 0.6.4', and 'COPY' commands for configuration files. At the bottom of the logs, it says 'Pushing image built from Dockerfile'. A 'Run Build Again' button is located at the bottom left of the log area, and the 'Build ID: 08c0e4ca-19ac-47e7-a55b-e690c0ad4f9e' is displayed at the bottom right. The top navigation bar includes links for 'Tour', 'Repositories', 'Docs', 'Tutorial', 'Pricing', and 'Organizations', along with a user profile for 'spesnova'.

デプロイ



Build

Develop an app using Docker containers with any language and any toolchain.



Ship

Ship the "Dockerized" app and dependencies anywhere - to QA, teammates, or the cloud - without breaking anything.

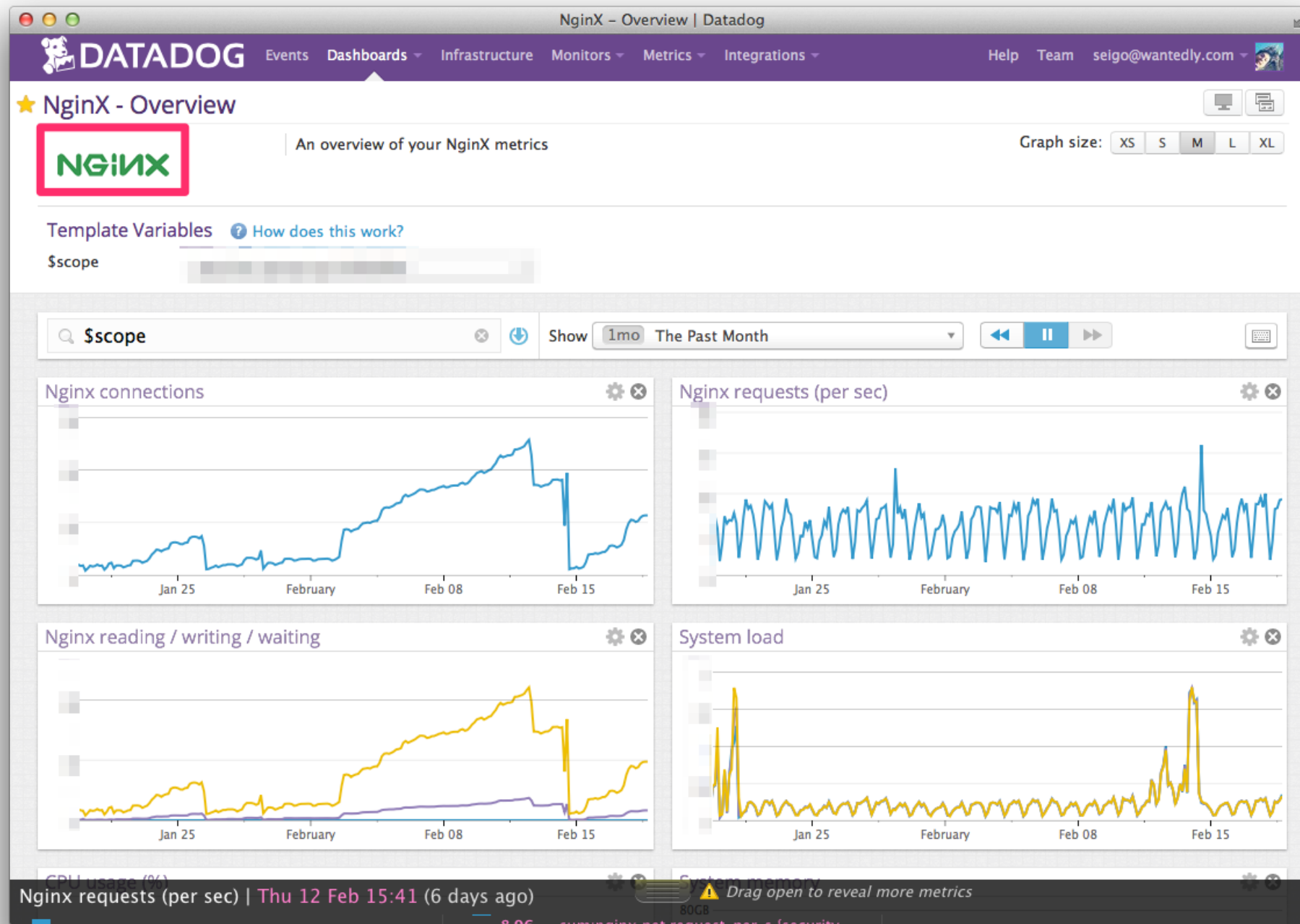


Run

Scale to 1000s of nodes, move between data centers and clouds, update with zero downtime and more.

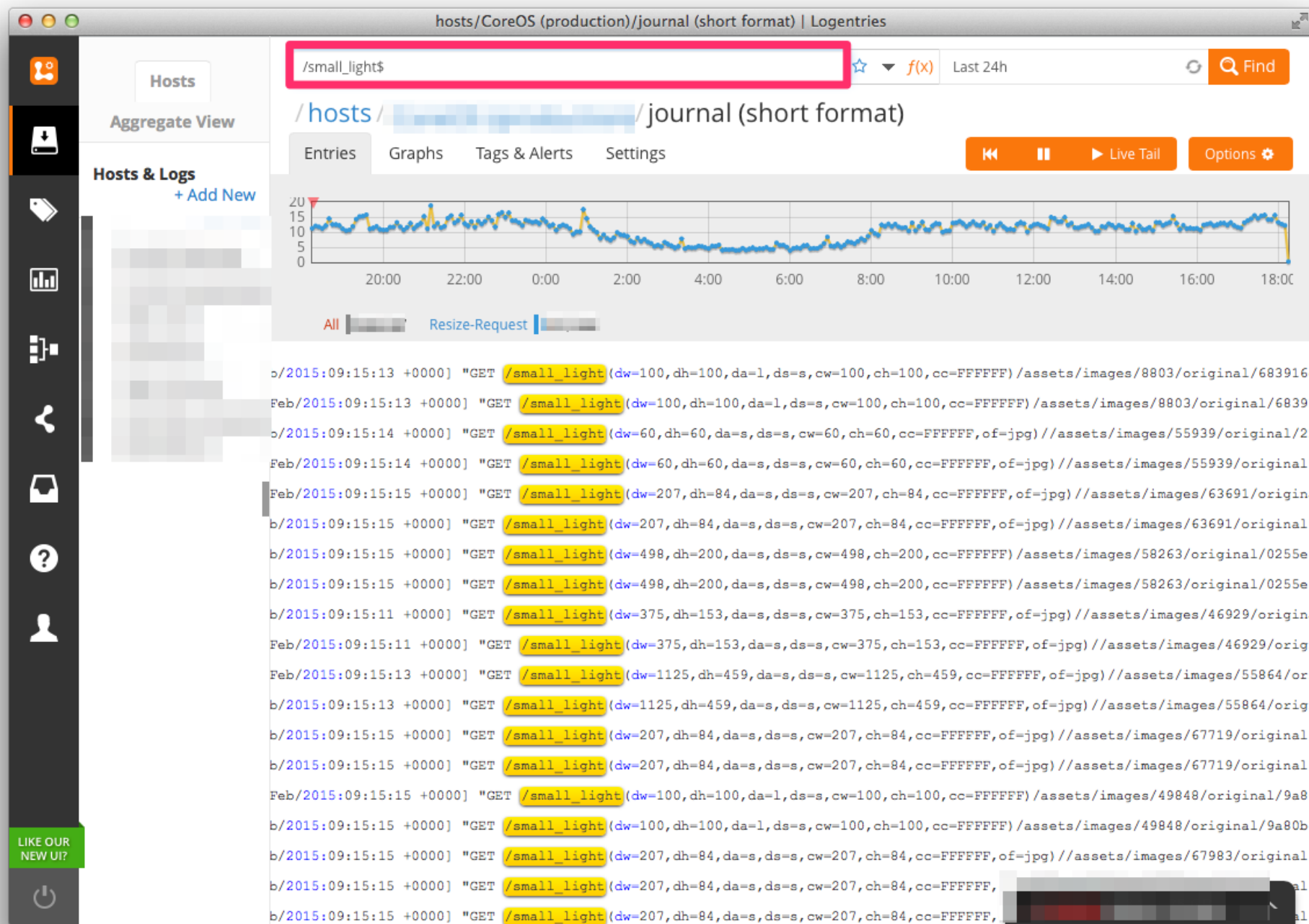
1. **CoreOS(EC2) インスタンスの追加**
2. **cloud-config でプロビジョニング**
3. **systemd unit としてコンテナが起動**
4. **古いインスタンスの削除**

モニタリング



Datadog で Docker/EC2/Nginx を見る

ログ



systemd **journal** を全て **Logentries** に送る
必要に応じてフィルタリングして見る

遭遇した問題

WebP の変換がおかしい

WebP の変換がおかしい



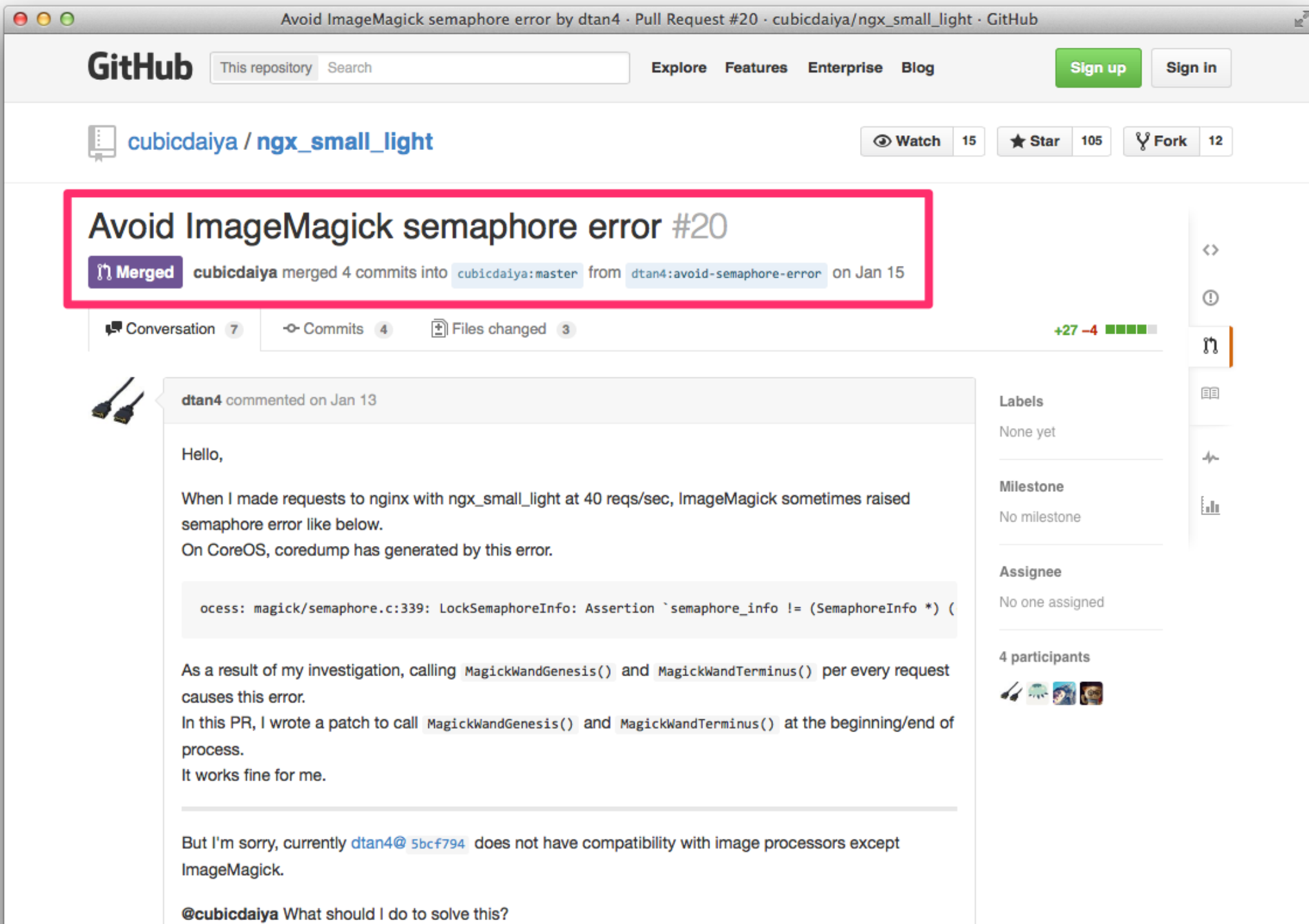
WebP の変換がおかしい

ImageMagick のバージョンが悪かった

⇒ ちゃんと **CHANGELOG** 見よう

Nginx が coredump 吐く

Nginx が coredump 吐く



The screenshot shows a GitHub pull request page for the repository 'cubicdaiya / ngx_small_light'. The pull request is titled 'Avoid ImageMagick semaphore error #20' and is marked as 'Merged'. It was merged by 'cubicdaiya' on January 15, 2015, merging 4 commits from the branch 'dtan4:avoid-semaphore-error' into 'cubicdaiya:master'. The pull request shows 7 conversations, 4 commits, and 3 files changed. A diff view shows a change of +27 lines and -4 lines. The main content is a comment from 'dtan4' dated January 13, 2015. The comment describes a semaphore error in ImageMagick when using ngx_small_light at 40 requests per second on CoreOS, which caused a coredump. The error message is: `rocess: magick/semaphore.c:339: LockSemaphoreInfo: Assertion `semaphore_info != (SemaphoreInfo *) (`. The comment explains that the error is caused by calling `MagickWandGenesis()` and `MagickWandTerminus()` for every request. The pull request includes a patch to call these functions at the beginning and end of the process, which resolved the issue for the user. The comment also mentions that the user's current environment does not have compatibility with image processors except ImageMagick and asks for advice from 'cubicdaiya' on how to solve this.

Avoid ImageMagick semaphore error #20

Merged cubicdaiya merged 4 commits into cubicdaiya:master from dtan4:avoid-semaphore-error on Jan 15

Conversation 7 Commits 4 Files changed 3 +27 -4

dtan4 commented on Jan 13

Hello,

When I made requests to nginx with ngx_small_light at 40 reqs/sec, ImageMagick sometimes raised semaphore error like below.

On CoreOS, coredump has generated by this error.

```
rocess: magick/semaphore.c:339: LockSemaphoreInfo: Assertion `semaphore_info != (SemaphoreInfo *) (
```

As a result of my investigation, calling `MagickWandGenesis()` and `MagickWandTerminus()` per every request causes this error.

In this PR, I wrote a patch to call `MagickWandGenesis()` and `MagickWandTerminus()` at the beginning/end of process.

It works fine for me.

But I'm sorry, currently dtan4@5bcf794 does not have compatibility with image processors except ImageMagick.

@cubicdaiya What should I do to solve this?

Labels: None yet

Milestone: No milestone

Assignee: No one assigned

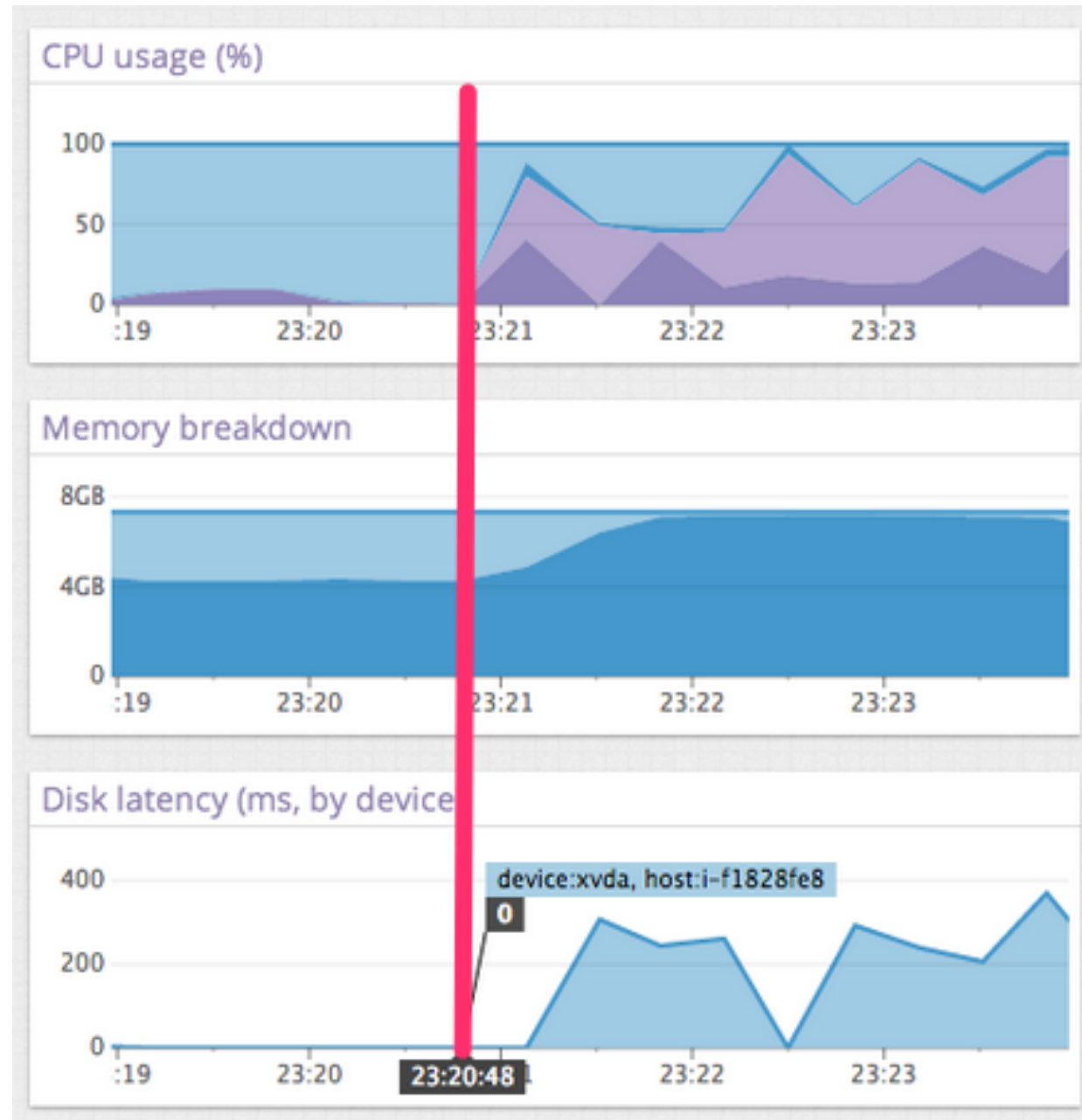
4 participants

nginx が coredump 吐く

nginx-image-serve の ImageMagick 初期化
のタイミングの問題だった

⇒ ユーザ **フィードバック** する

Nginx が異常に高負荷になる



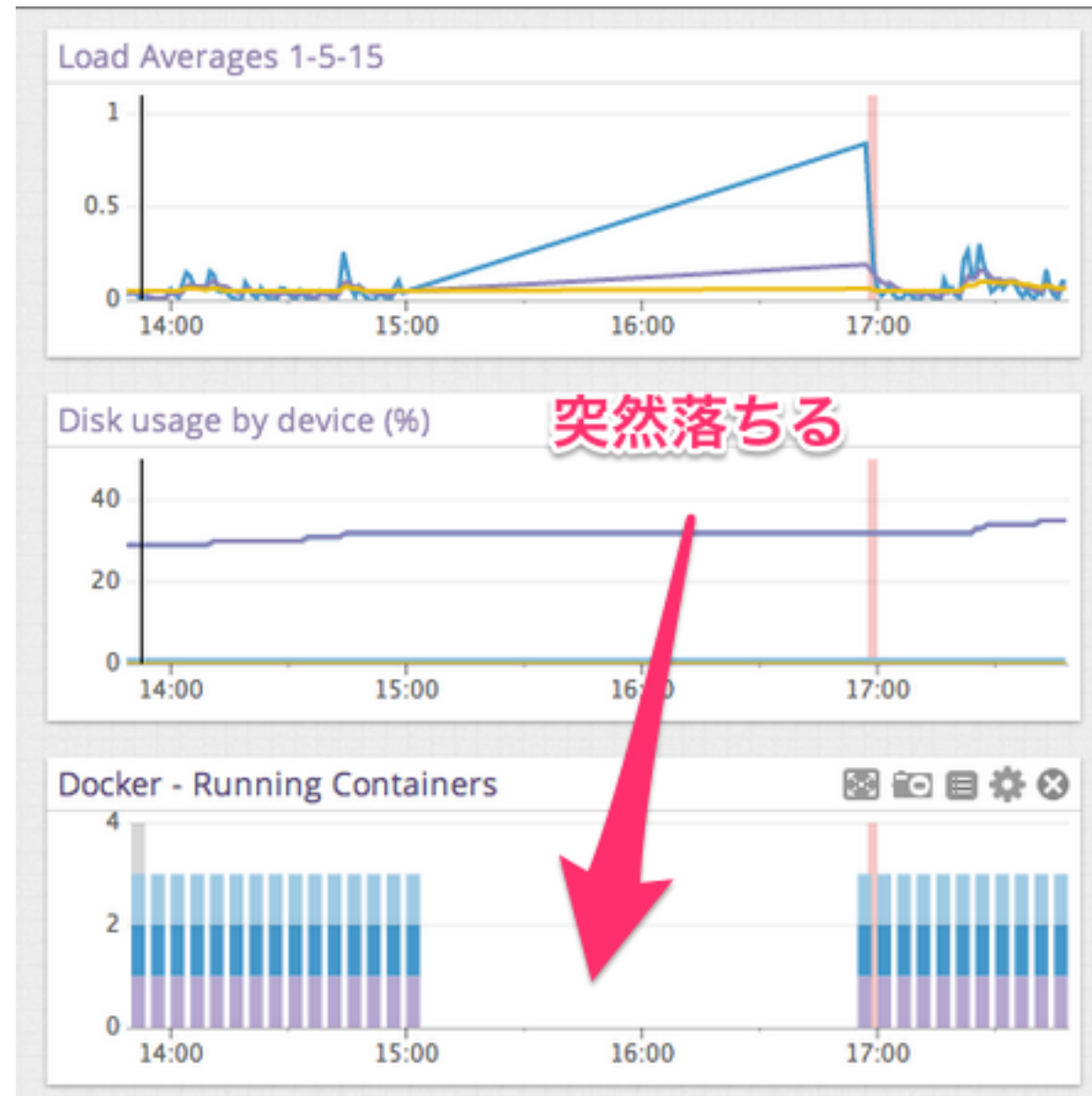
Nginx が異常に高負荷になる

異常に大きいサイズがリクエストされてた

⇒ 画像サーバ側でサイズ制限した

⇒ ログとサーバメトリクス同時に見る

AWS 上で CoreOS がクラッシュ



AWS 上で CoreOS がクラッシュ

The screenshot shows a GitHub issue page for the repository 'coreos/bugs'. The issue title is 'CoreOS crash on AWS with (many versions) #204', which is highlighted with a red box. The issue is marked as 'Closed' and was opened by 'jonboulle' on Dec 11, 2014, with 65 comments. The issue description, also highlighted with a red box, reads: 'From @i0n on October 30, 2014 11:23 Hey there chaps. Experiencing some very strange behaviour using CoreOS 444.5.0 stable on ec2. Cluster boots fine and I am able to schedule services with fleet (which come up and start working), but then after a random length of time machines in the cluster will crash forcing me to restart the VM in question to recover. Only clue in the logs is: "ERROR reconcile.go:58: Unable to determine agent's desired state: timeout reached" Has anyone else run into this? I have raised this as a fleet issue as the log hints at that, but I guess it could be etcd? Copied from original issue: [coreos/fleet#1017](#)'. Below this is a comment from 'jonboulle' on Dec 11, 2014, replying to '@bcwaldon' on Oct 30, 2014: '@i0n What do you mean by "machines in the cluster will crash"? Are your VMs just going away? Are services crashing? Any info in dmesg?'. The right sidebar shows the issue has 'bug' and 'os' labels, no milestone, no assignee, and 12 participants.

AWS 上で CoreOS がクラッシュ

kernel のバグだった

⇒ **継続的なOSのアップデート**

まとめ

まとめ

- `ngx_small_light` 重宝しています！
- `ngx_http_secure_link_module` 導入したい
- ImageMagick のビルド、一度動いたらイメージに焼いてしまいたい
- Docker で全部同じフローでデプロイできる