

2019-11-28 DNSOPS.JP BoF

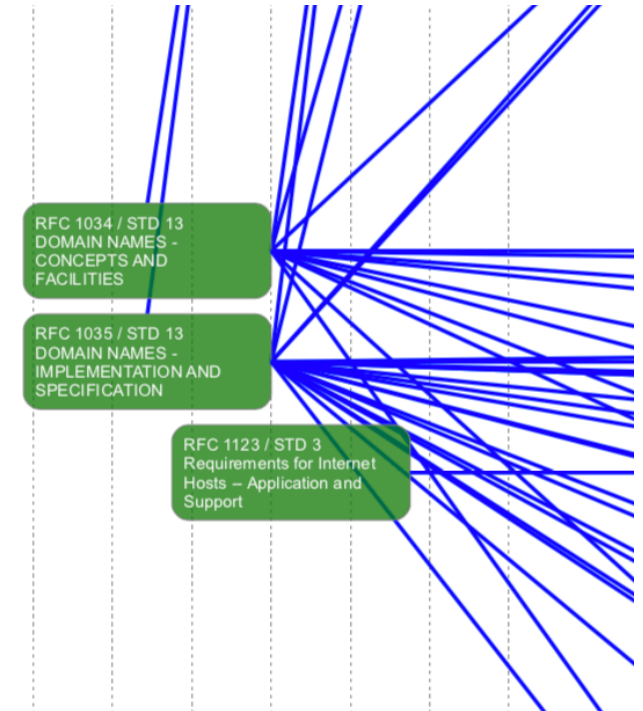
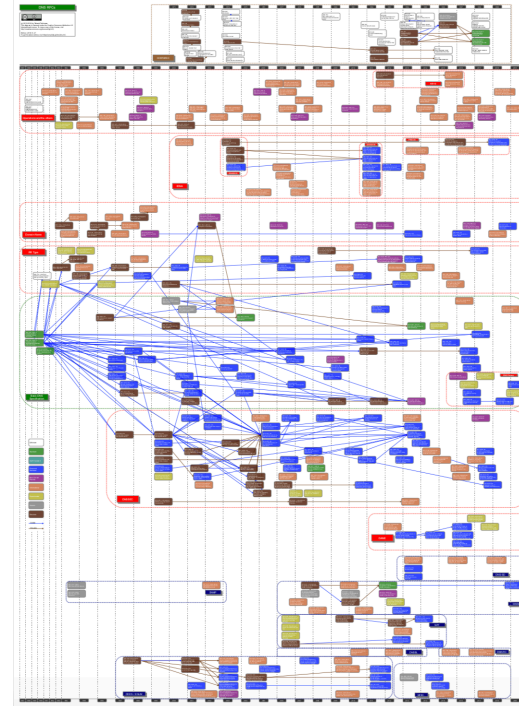
OctoDNSとGitLab CI/CDを利用した 複数DNSプロバイダー構成の運用



株式会社ハートビーツ 滝澤隆史

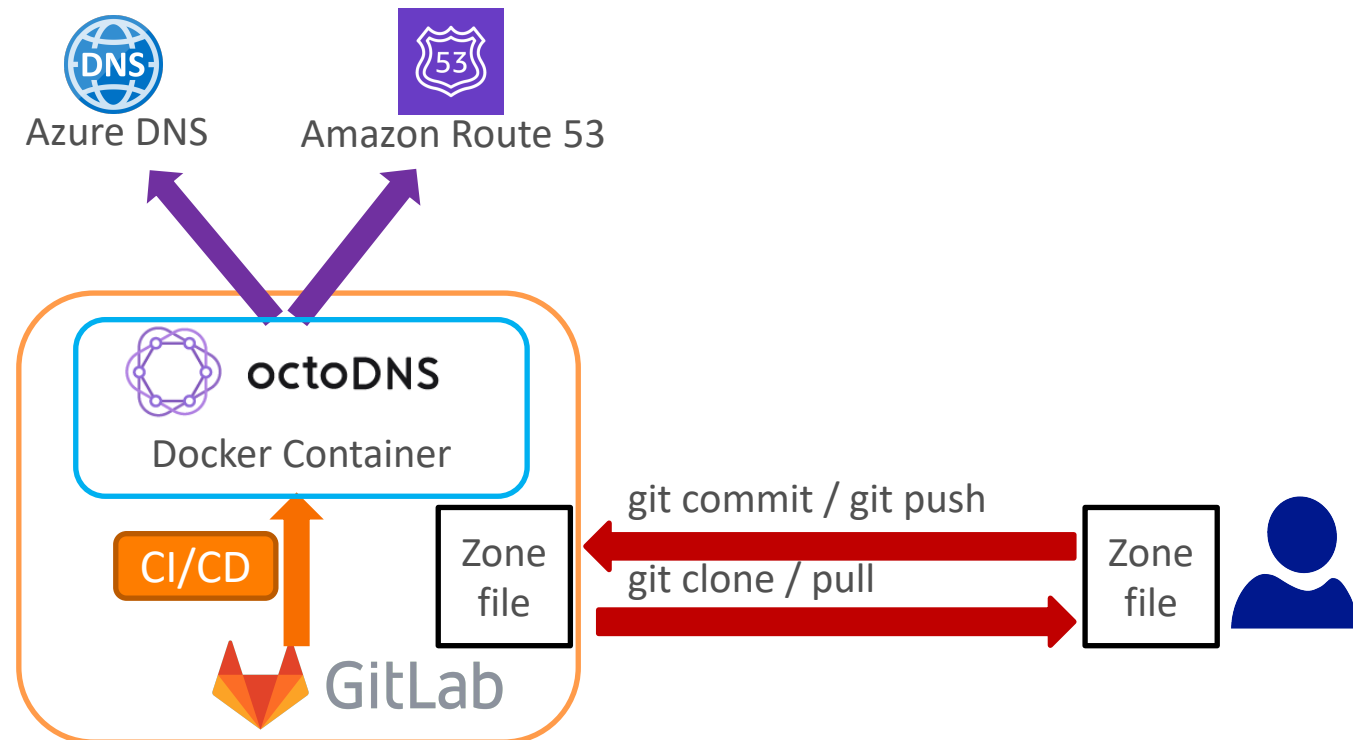
私は誰

- 滝澤 隆史 (たきざわ たかし)
- Twitter: @ttkzw
- DNSで遊んでいる人
 - DNS RFC系統図
 - <https://emaillab.jp/dns/dns-rfc/>
- 所属: 株式会社ハートビーツ
 - 事業内容
 - MSP (監視・運用代行)、受託開発 (PHP + React (TypeScript))
 - 滝澤は社内システム基盤の面倒を見ている
 - DNS権威サーバー
 - ここのお話をします



本日のお話を 3 行で説明すると

- DNSゾーン管理ツール OctoDNS と
- SCM（ソースコード管理）ツール GitLab のCI/CD機能を使って、
- 複数DNSプロバイダー構成を運用する事例を紹介します。



背景（きっかけ）

課題認識

- 2016年10月のDynのDNS権威サーバーへの大規模DDoS
- 自前運用の弊社のDNS権威サーバー
 - DDoSが発生したらこの1000分の1程度でも回線飽和で使い物にならなくなるという課題認識
 - 弊社はお客様へ監視サービスを提供しており、アラートメールの送信ができなくなったり、監視システムへアクセスできなくなったりするのは事業上の大問題
 - この時点ではこの課題を社内のGitLabのIssueに起票しただけ

オープンソースのツールの認識

- 2017年春に複数DNSプロバイダー対応したOSSの管理ツールがあることを認識し、調査を実施
 - Stack Exchange社開発のDNSControl (2017-03-14 First public release)
 - GitHub社開発のOctoDNS (2017-03-14 First public release)
- この時点では、利用したいDNSプロバイダーやリソースレコードの要件には合わなかったため、検討を中断した。
 - 利用を検討していたDNSプロバイダーがNSレコードの追加に対応していなかったということもあった。これは後ほど解消した。
- この後、「課題は認識しているが、緊急性が高いわけではないため、優先度が高くない」という典型的な状況になり、2年ほど放置されてしまった。

データセンター閉鎖

- ラックを借りているデータセンターが閉鎖することになった。
- DNS権威サーバーをどこかに移転しなければならない
- 「課題は認識しているが、緊急性が高いわけではないため、優先度が高くない」
 - 別な意味で緊急性が高くなってしまった。

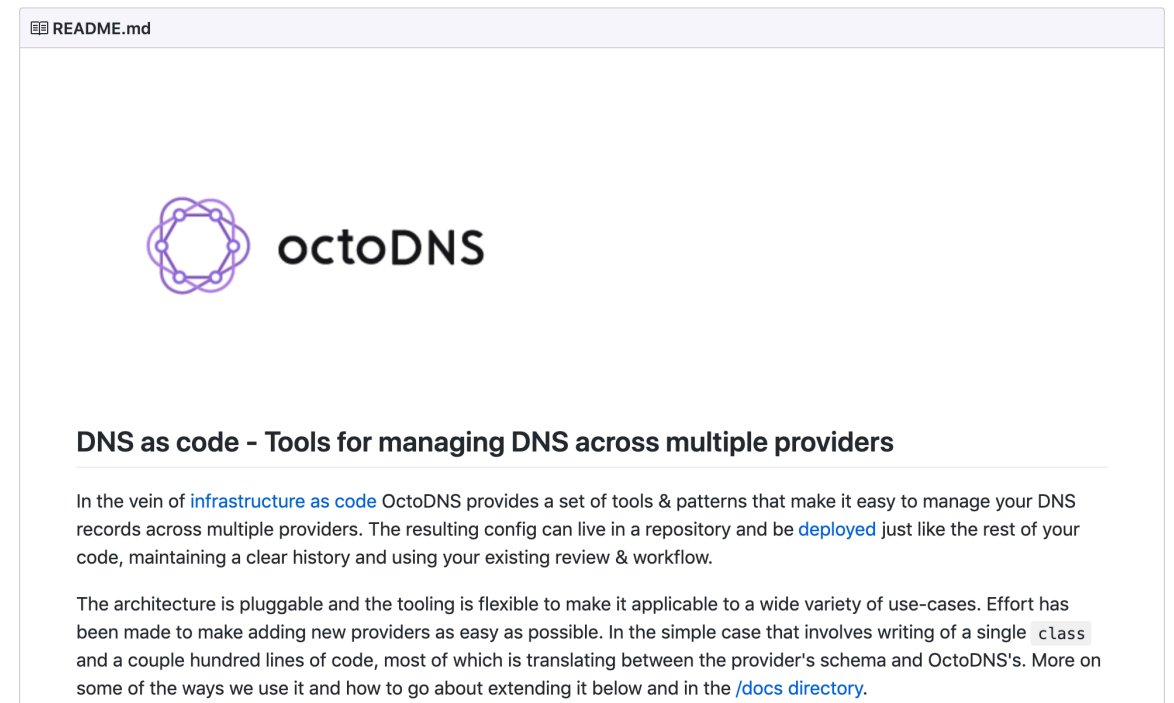
プロジェクト開始

- ということで調査を再開
- 以前の調査では要件が合わなかったDNSプロバイダーとリソースレコードの件が解消されていた。
- 結果として、OctoDNSを使うことにした。
 - OctoDNSの方が設定ファイルがシンプルという理由

OctoDNS

OctoDNSとは

- GitHub社が開発・保守しているオープンソースソフトウェア
- DNS as code - Tools for managing DNS across multiple providers
- <https://github.com/github/octodns>



OctoDNSとは

- CLIツール
 - octodns-sync
 - ソースプロバイダーのゾーン情報をターゲットプロバイダーにAPIで同期する

```
$ octodns-sync --config-file=./config/production.yaml --doit  
...
```



OctoDNSの設定ファイル（公式サイトでのREADME.mdより）

config/production.yaml

プロバイダー定義
(認証情報含む)

```
---
providers:
  config:
    class: octodns.provider.yaml.YamlProvider
    directory: ./config
    default_ttl: 3600
    enforce_order: True
  dyn:
    class: octodns.provider.dyn.DynProvider
    customer: 1234
    username: 'username'
    password: env/DYN_PASSWORD
  route53:
    class: octodns.provider.route53.Route53Provider
    access_key_id: env/AWS_ACCESS_KEY_ID
    secret_access_key: env/AWS_SECRET_ACCESS_KEY
```

ゾーン毎の
プロバイダーの指定

```
zones:
  example.com.:
    sources:
      - config
    targets:
      - dyn
      - route53
```

config/example.com.yaml

```
---
"":
  ttl: 60
  type: A
  values:
    - 1.2.3.4
    - 1.2.3.5
```

ソース
プロバイダーの
ゾーンファイル
(YAML形式)

OctoDNSの実行例（公式サイト README.md より）

- デフォルトはDRY RUNとして動作する。

```
$ octodns-sync --config-file=./config/production.yaml
...
*****
* example.com.
*****
* route53 (Route53Provider)
*   Create <ARecord A 60, example.com., [u'1.2.3.4', '1.2.3.5']>
*   Summary: Creates=1, Updates=0, Deletes=0, Existing Records=0
* dyn (DynProvider)
*   Create <ARecord A 60, example.com., [u'1.2.3.4', '1.2.3.5']>
*   Summary: Creates=1, Updates=0, Deletes=0, Existing Records=0
*****
...
```

- 実際に反映するためには --doit オプションを付ける。

```
$ octodns-sync --config-file=./config/production.yaml --doit
...
```

OctoDNSが対応しているプロバイダー

- AzureProvider
- Akamai
- CloudflareProvider
- ConstellixProvider
- DigitalOceanProvider
- DnsMadeEasyProvider
- DnsimpleProvider
- DynProvider
- EtcHostsProvider
- GoogleCloudProvider
- MythicBeastsProvider
- Ns1Provider
- OVH
- PowerDNSProvider
- RackspaceProvider
- Route53
- Selectel
- Transip
- ソースプロバイダー
 - AxfrSource
 - ZoneFileSource
 - TinyDnsFileSource
- コンフィグ
 - YamlProvider

採用した構成

- ソースプロバイダー
 - ZoneFileSource (ゾーンファイル)
 - 既存のゾーンファイルをそのまま利用でき、CLIベースの作業手順のまま利用できるため。
- ターゲットプロバイダー
 - Route53Provider (Amazon Route53)
 - AzureProvider (Azure DNS)



GitLab CI/CD

octodns-syncをどこで実行するか

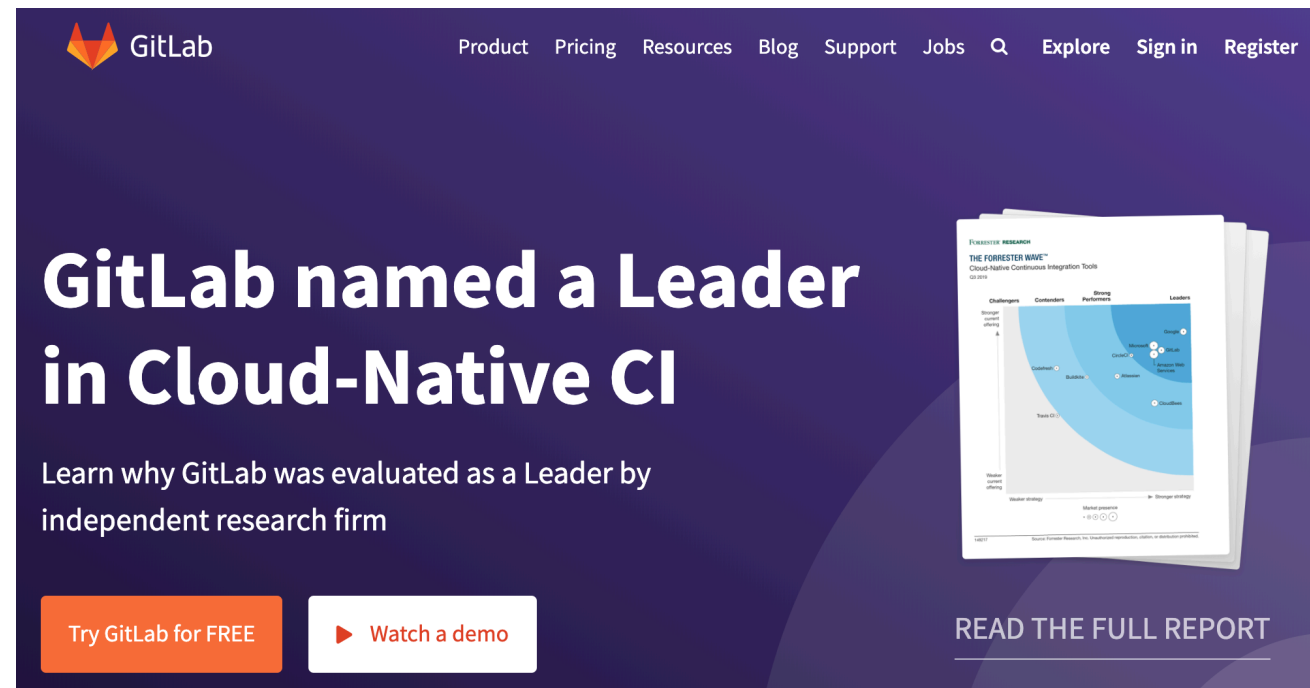
- 既存のシステム
 - DNS権威サーバーのホストにSSHログインして、Subversion管理下のゾーンファイルを直接編集していた。
 - BIND 9が動いている
 - NSDに移行する計画はあったんですけどね

octodns-syncをどこで実行するか

- 新しいシステム
 - DNS権威サーバーのホストがない！
 - ソースコード管理にGitLabを利用しているので、GitLab CI/CDを利用して、
Dockerコンテナ内でoctodns-syncを実行するようにした。

GitLabとは

- GitLab社が開発しているSCM（ソースコード管理）ツール
 - CI/CD (Continuous Integration/Continuous Delivery)機能もある
- <https://about.gitlab.com/>



The banner features the GitLab logo and navigation menu at the top. The main headline reads "GitLab named a Leader in Cloud-Native CI". Below this, it states "Learn why GitLab was evaluated as a Leader by independent research firm". On the right, there is a stack of report covers, with the top one titled "THE FORRESTER WAVE™ Cloud-Native Continuous Integration Tools Q3 2019". The report cover shows a Gartner-style quadrant chart with GitLab positioned in the "Leader" quadrant. At the bottom left, there are two buttons: "Try GitLab for FREE" and "Watch a demo". At the bottom right, there is a link "READ THE FULL REPORT".

GitLab

Product Pricing Resources Blog Support Jobs Q Explore Sign in Register

GitLab named a Leader in Cloud-Native CI

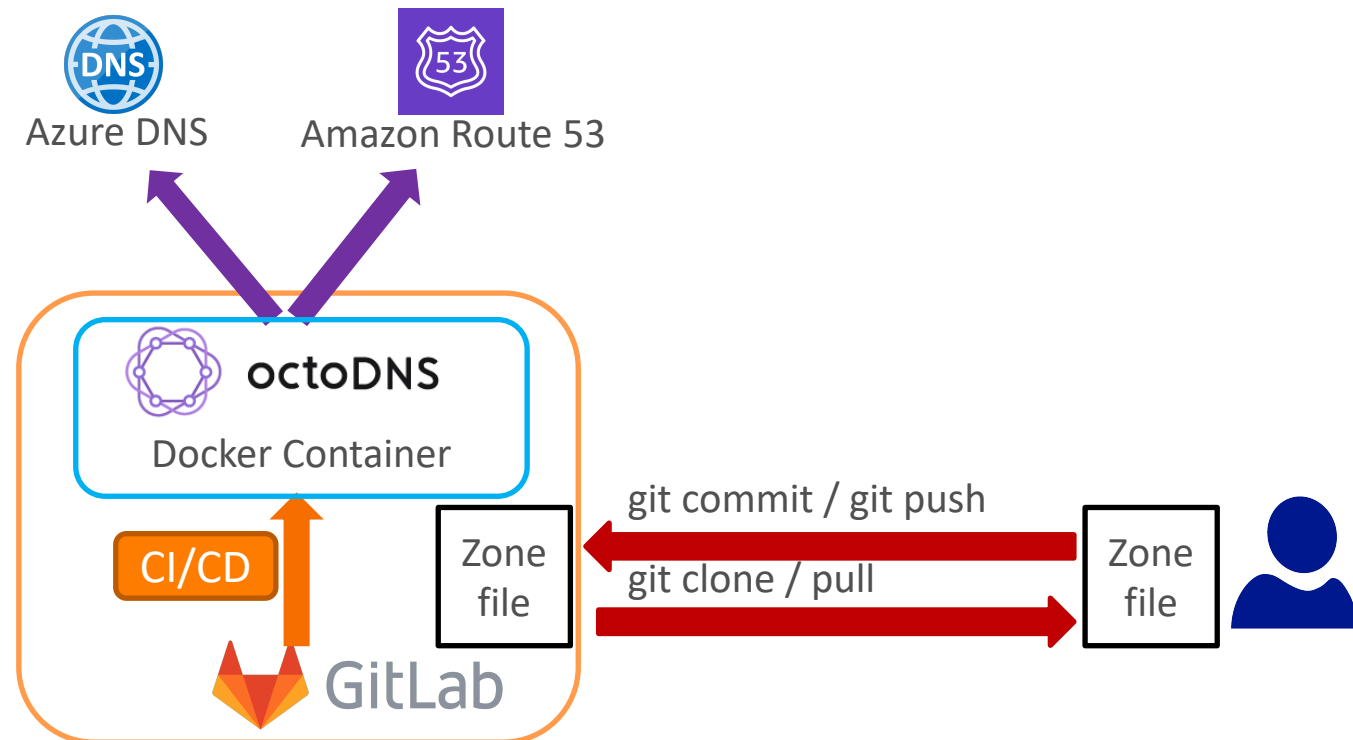
Learn why GitLab was evaluated as a Leader by independent research firm

[Try GitLab for FREE](#) [Watch a demo](#)

[READ THE FULL REPORT](#)

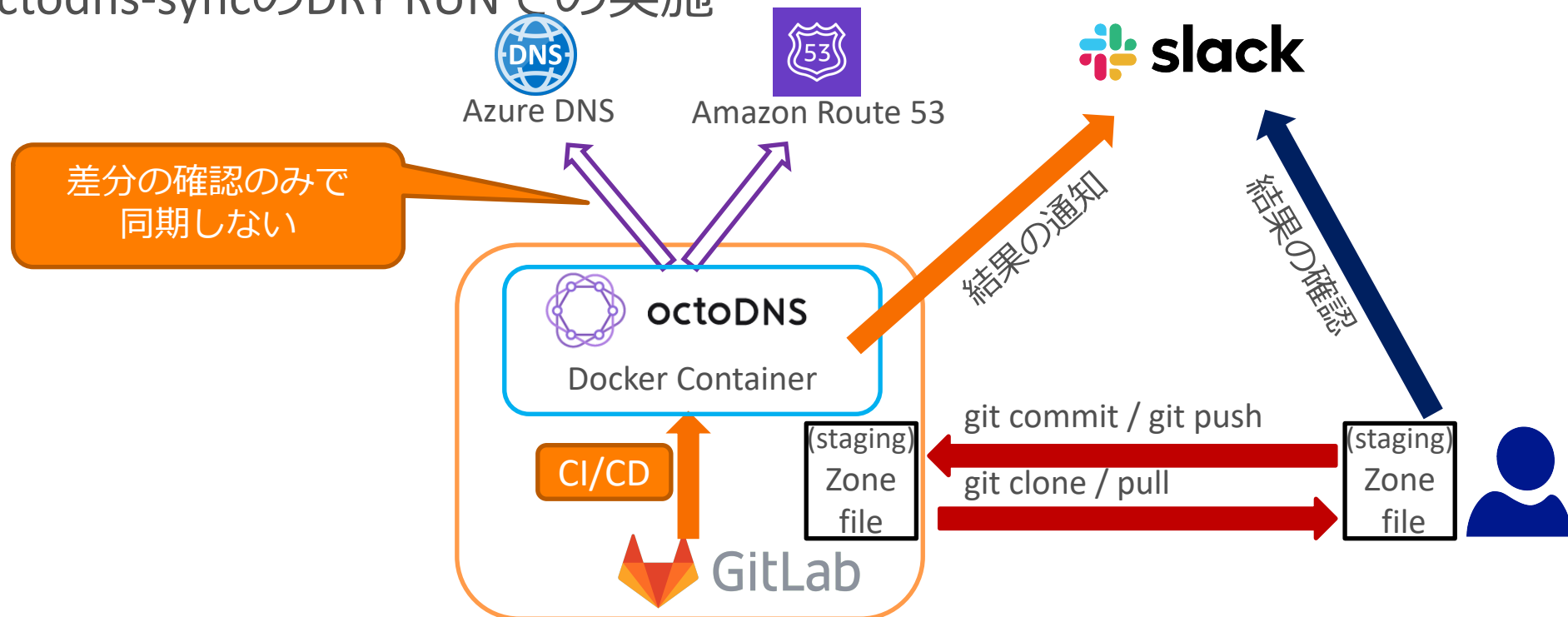
基本構成

- GitLab CI/CDにより、
- OctoDNSをインストールしたDockerコンテナを起動して、
- octodns-syncを実行するようになる。



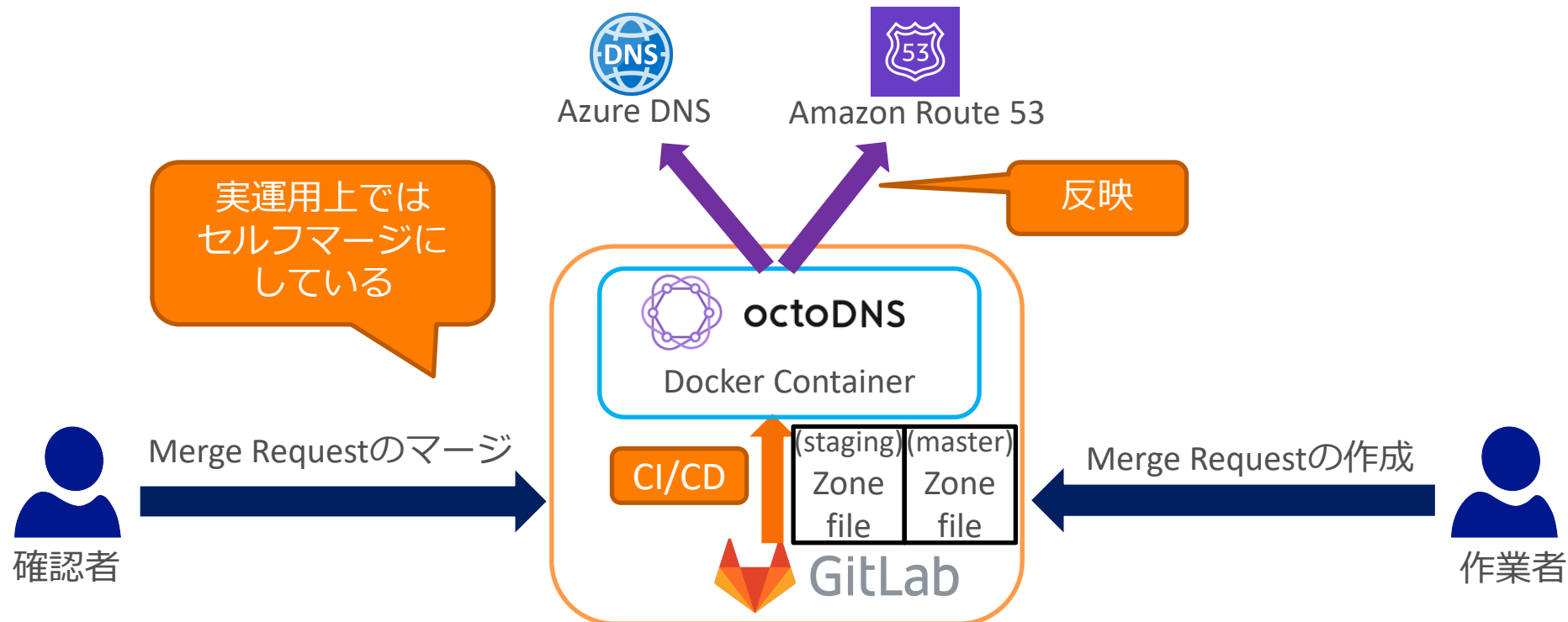
テスト

- 何も確認せずにDNSプロバイダーのゾーンに反映するのは危険
- stagingブランチを用意して、テストを行う
 - ゾーンファイルの構文チェック (named-checkzone)
 - octodns-syncのDRY RUNでの実施



Merge Requestの作成とマージ

- 作業者はstagingブランチからmasterブランチへのMerge Requestを作成する。
- 確認者はMerge Requestの内容を確認してマージする。
- GitLab CI/CDによりoctodns-syncを--doitオプションを付けて実行し、DNSプロバイダーのゾーンに反映する。



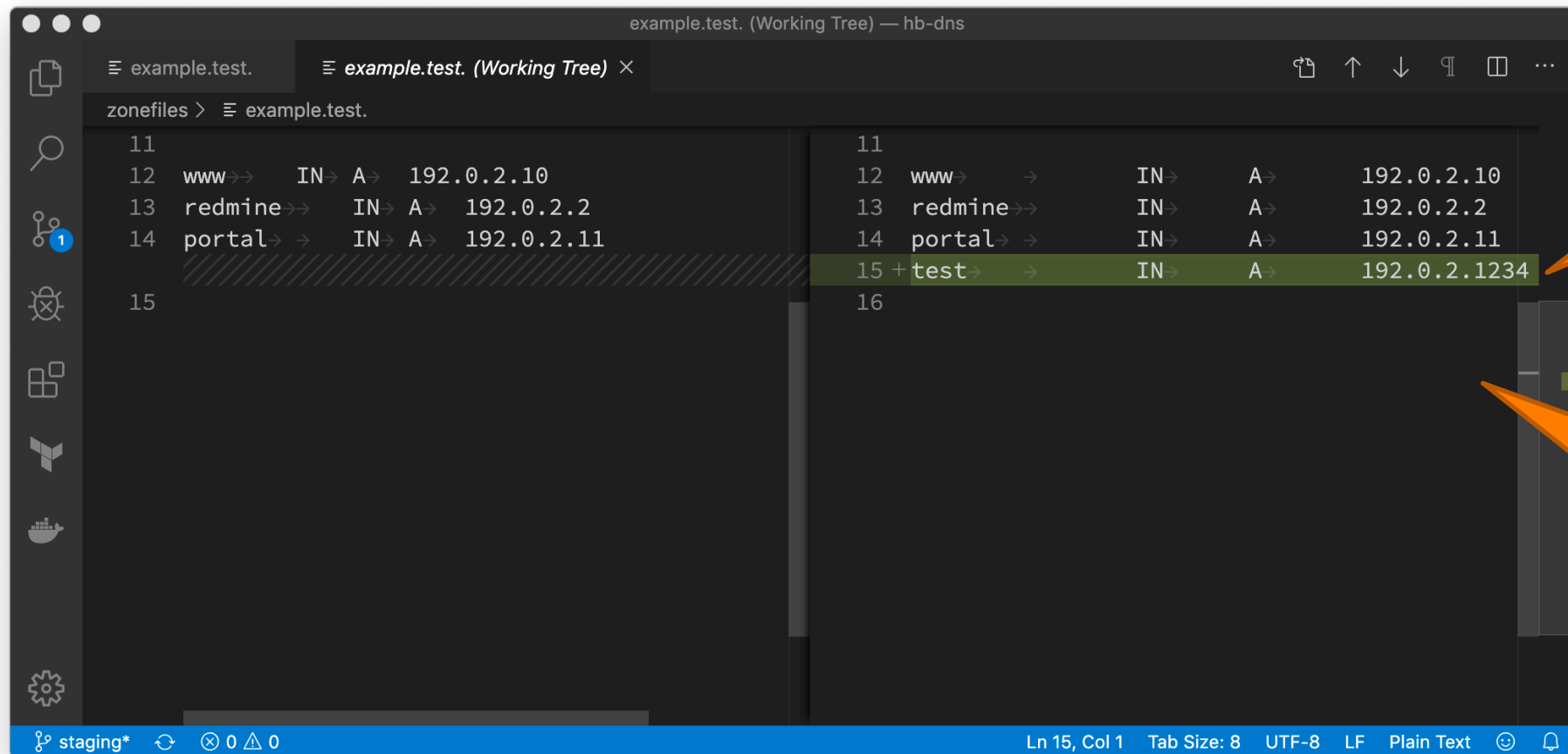
実行例の紹介

実行例の紹介

- 失敗する実行例
 - 構文ミス
- 成功する実行例

失敗する実行例

- stagingブランチからgit pullして、ゾーンファイルを編集し、差分を確認し、git commit & git pushする。



```

example.test. (Working Tree) — hb-dns
example.test. example.test. (Working Tree) ×
zonefiles > example.test.
11
12 www-> IN A 192.0.2.10
13 redmine-> IN A 192.0.2.2
14 portal-> IN A 192.0.2.11
15
16
11
12 www-> IN A 192.0.2.10
13 redmine-> IN A 192.0.2.2
14 portal-> IN A 192.0.2.11
15 + test-> IN A 192.0.2.1234
16
staging* 0 0 0 Ln 15, Col 1 Tab Size: 8 UTF-8 LF Plain Text

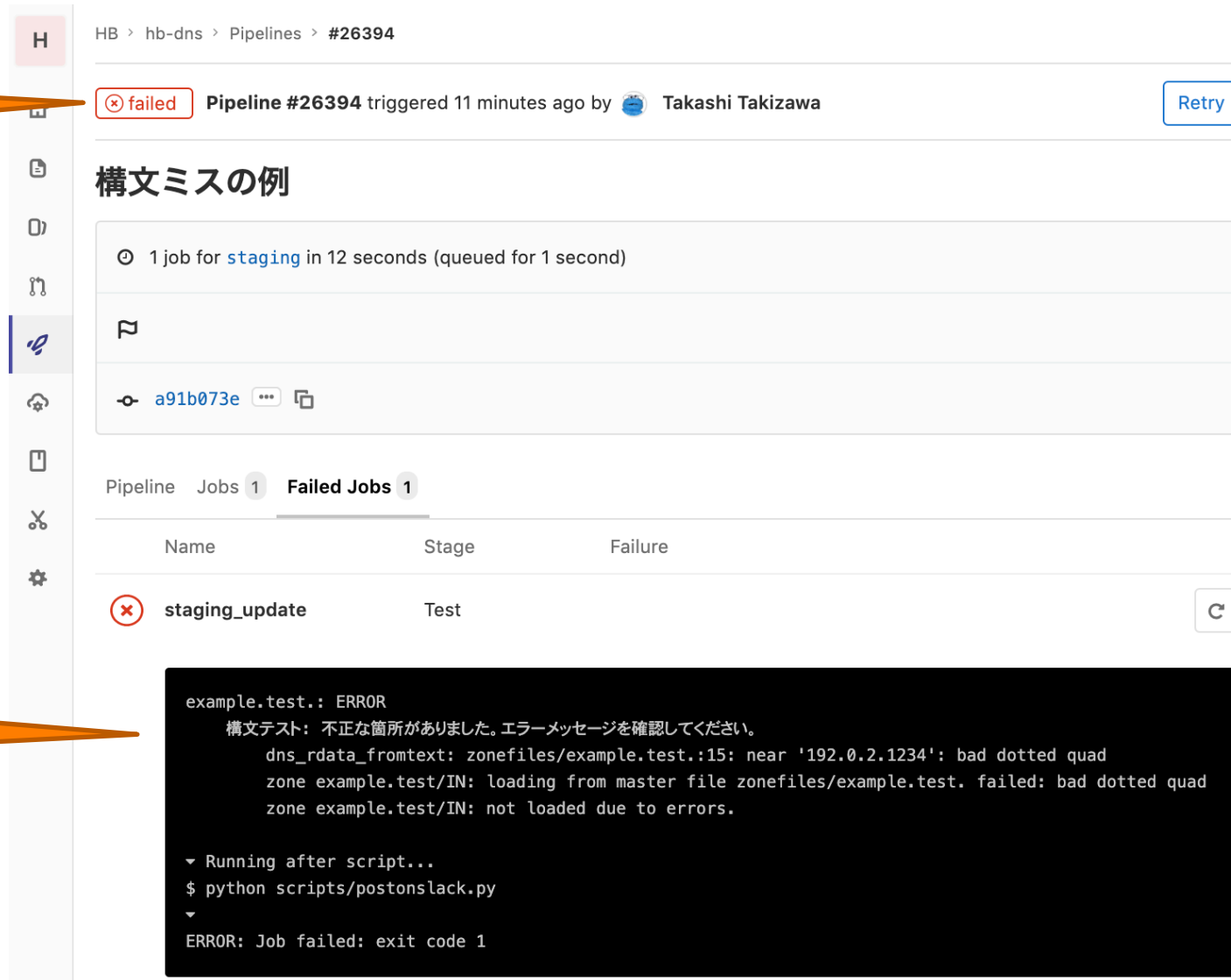
```

差分の確認
(構文ミス)

実際の手順はvim
で行うけど、
VSCodeでgit pushま
でやってみた例

GitLab CI/CDのジョブの確認画面（失敗する実行例）

ステータス



HB > hb-dns > Pipelines > #26394

⊗ failed Pipeline #26394 triggered 11 minutes ago by Takashi Takizawa Retry

構文ミスの例

⌚ 1 job for `staging` in 12 seconds (queued for 1 second)

`a91b073e`

Pipeline Jobs **1** Failed Jobs **1**

Name	Stage	Failure
⊗ staging_update	Test	

```
example.test.: ERROR
  構文テスト: 不正な箇所がありました。エラーメッセージを確認してください。
  dns_rdata_fromtext: zonefiles/example.test.:15: near '192.0.2.1234': bad dotted quad
  zone example.test/IN: loading from master file zonefiles/example.test. failed: bad dotted quad
  zone example.test/IN: not loaded due to errors.

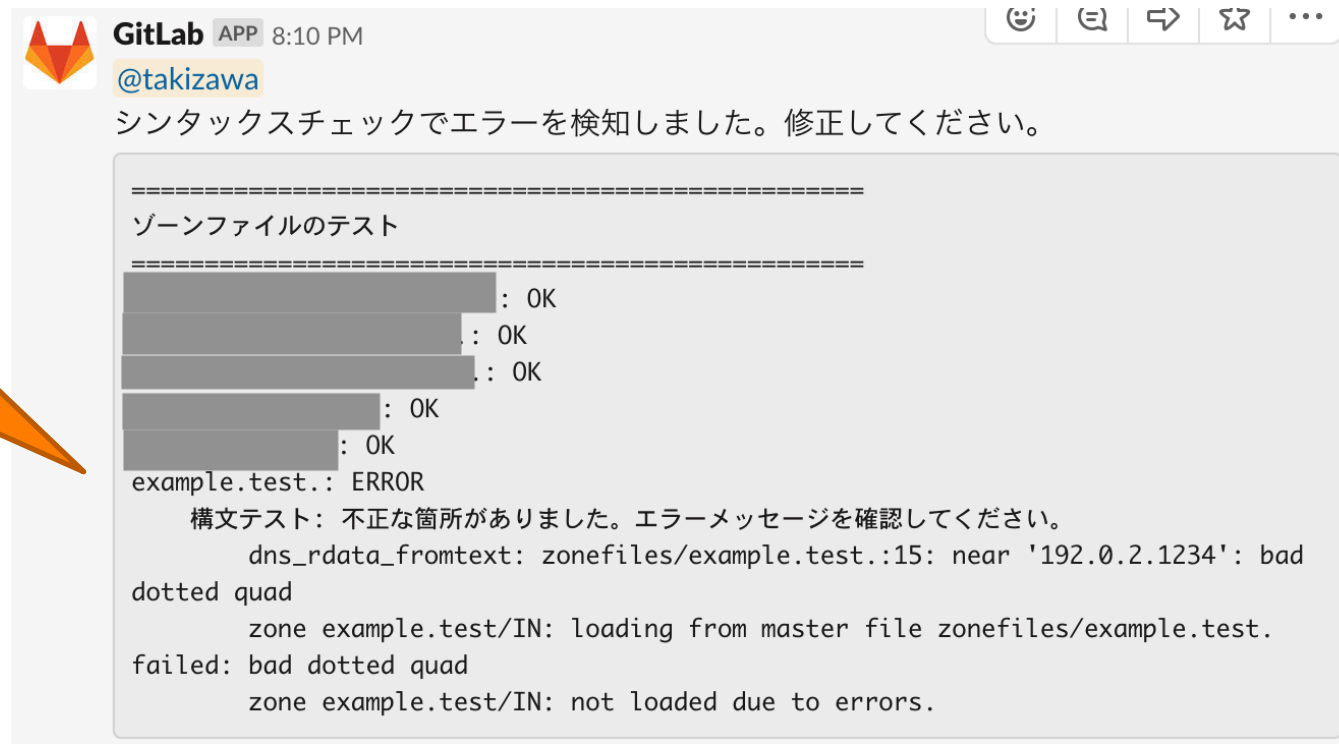
  ▼ Running after script...
  $ python scripts/postonslack.py
  ▼
  ERROR: Job failed: exit code 1
```

ジョブの出力

Slackへのジョブ失敗の通知

Slackへの通知

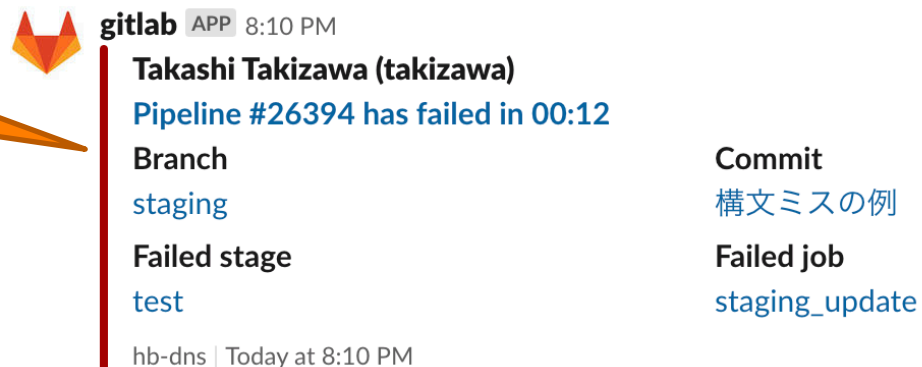
GitLabの画面を開かなくてもすむようにジョブの出力内容を通知



```

=====
ゾーンファイルのテスト
=====
[REDACTED]: OK
[REDACTED]: OK
[REDACTED]: OK
[REDACTED]: OK
[REDACTED]: OK
example.test.: ERROR
  構文テスト: 不正な箇所がありました。エラーメッセージを確認してください。
    dns_rdata_fromtext: zonefiles/example.test.:15: near '192.0.2.1234': bad
dotted quad
    zone example.test/IN: loading from master file zonefiles/example.test.
failed: bad dotted quad
    zone example.test/IN: not loaded due to errors.
  
```

GitLabの Slack Integrationによる通知

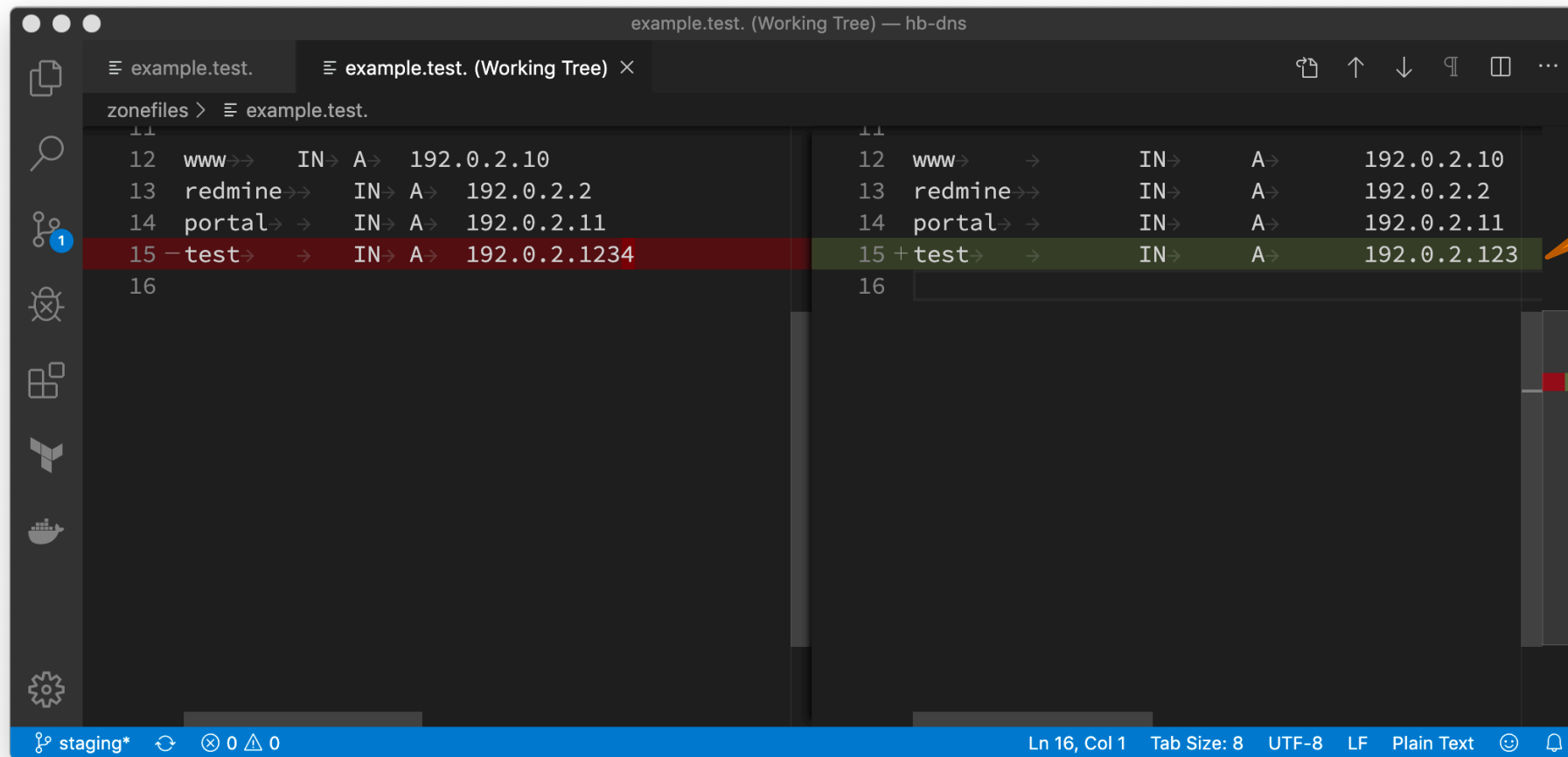


```

gitlab APP 8:10 PM
Takashi Takizawa (takizawa)
Pipeline #26394 has failed in 00:12
Branch
staging
Failed stage
test
Commit
構文ミスの例
Failed job
staging_update
hb-dns | Today at 8:10 PM
  
```

成功する実行例

- stagingブランチからgit pullして、ゾーンファイルを編集し、差分を確認し、git commit & git pushする。



```

example.test. (Working Tree) — hb-dns
example.test. example.test. (Working Tree) ×
zonefiles > example.test.
11
12 www→→ IN→ A→ 192.0.2.10
13 redmine→→ IN→ A→ 192.0.2.2
14 portal→→ IN→ A→ 192.0.2.11
15 -test→→ IN→ A→ 192.0.2.1234
16
11
12 www→→ IN→ A→ 192.0.2.10
13 redmine→→ IN→ A→ 192.0.2.2
14 portal→→ IN→ A→ 192.0.2.11
15 +test→→ IN→ A→ 192.0.2.123
16
staging* 0 0 Ln 16, Col 1 Tab Size: 8 UTF-8 LF Plain Text
  
```

差分の確認
(正しい)

GitLab CI/CDのジョブの確認画面（成功する実行例）

H

Home

Search

Jobs

Repositories

Groups

Projects

Settings

More

```

desired=140.224.210.in-addr.arpa.
2019-11-18T11:22:24 [139752525883200] INFO AzureProvider[azuredns] populate: found 257
records, exists=True
2019-11-18T11:22:24 [139752525883200] INFO AzureProvider[azuredns] plan: No changes
2019-11-18T11:22:24 [139752525883200] INFO Manager
*****
* example.test.
*****
* route53 (Route53Provider)
* Create <ARecord A 900, test.example.test., ['192.0.2.123']> (zonefile)
* Summary: Creates=1, Updates=0, Deletes=0, Existing Records=4
* azuredns (AzureProvider)
* Create <ARecord A 900, test.example.test., ['192.0.2.123']> (zonefile)
* Summary: Creates=1, Updates=0, Deletes=0, Existing Records=4
*****
  <div data-bbox="265 628 425 735" data-label="Text" style="color: #4f81bd;">
    <div data-bbox="265 628 395 645" data-label="Text" style="color: #4f81bd;">
      <div data-bbox="265 650 425 668" data-label="Text" style="color: #4f81bd;">
        <div data-bbox="265 675 275 690" data-label="Image" style="color: #4f81bd;">
          <div data-bbox="265 700 275 715" data-label="Image" style="color: #4f81bd;">
            <div data-bbox="265 720 335 735" data-label="Text" style="color: #4f81bd;">
              Job succeeded
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

staging_update Retry

Duration: 48 seconds

Timeout: 1h (from project) ?

Runner: docker-runner (#4)

Tags: docker

Commit 56867e02 📄

成功する例

✔ Pipeline #26395 for staging

test ▼

➔ ✔ staging_update

ジョブの成功

Slackへのジョブ成功の通知

Slackへの通知

GitLabの画面を開かなくてもすむようにジョブの出力内容を通知

GitLabの
Slack Integration
による通知



GitLab APP 8:22 PM

@takizawa

レコードの差分です。
意図した通りの更新内容か確認してください。

```
*****  
* example.test.  
*****  
* route53 (Route53Provider)  
*   Create <ARecord A 900, test.example.test., ['192.0.2.123']> (zonefile)  
*   Summary: Creates=1, Updates=0, Deletes=0, Existing Records=4  
* azuredns (AzureProvider)  
*   Create <ARecord A 900, test.example.test., ['192.0.2.123']> (zonefile)  
*   Summary: Creates=1, Updates=0, Deletes=0, Existing Records=4  
*****
```



gitlab APP 8:22 PM

Takashi Takizawa (takizawa)

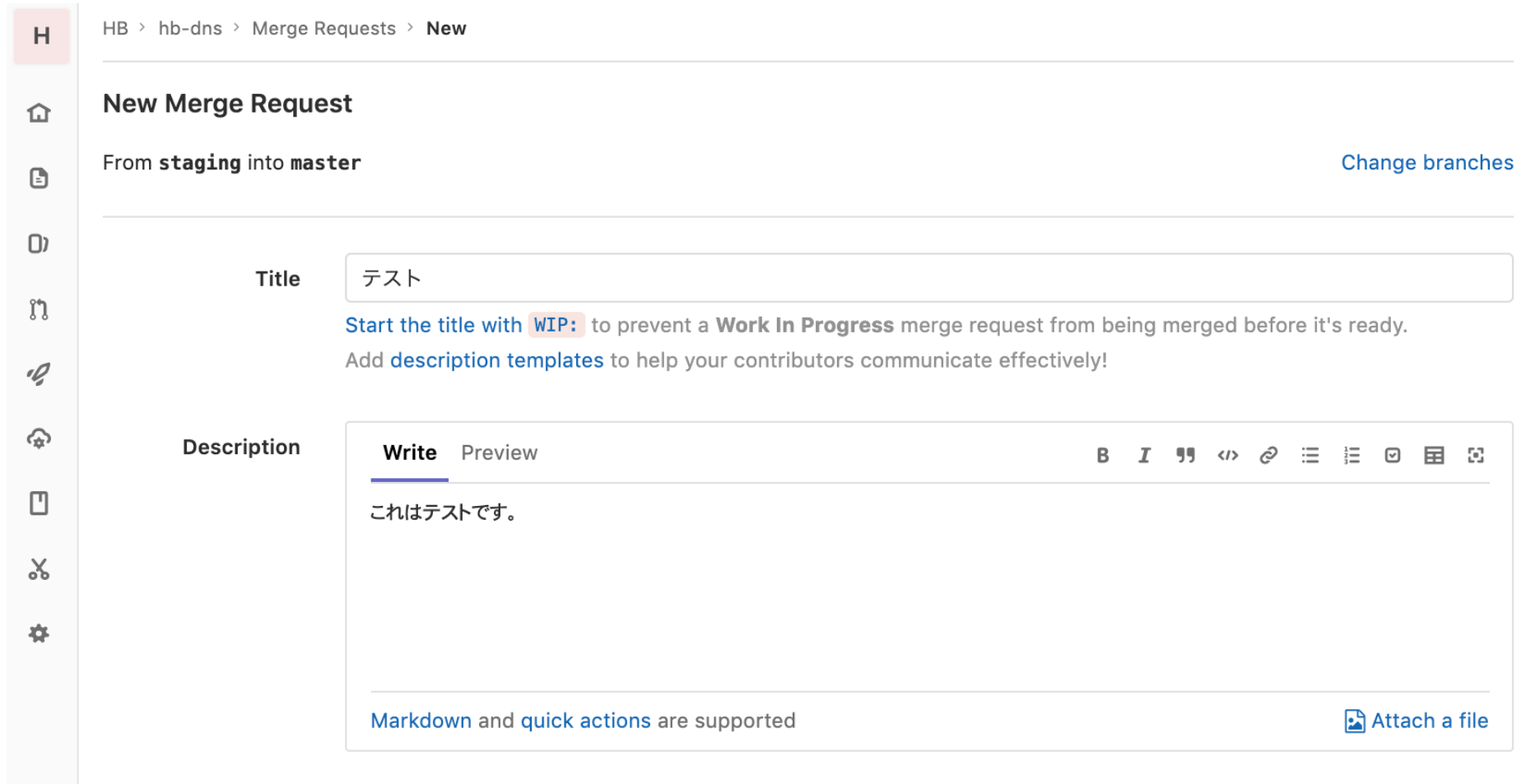
Pipeline #26395 has passed in 00:48

Branch
staging

Commit
成功する例

hb-dns | Today at 8:22 PM

GitLab CI/CD: Merge Requestの作成



HB > hb-dns > Merge Requests > New

New Merge Request

From **staging** into **master** [Change branches](#)

Title

Start the title with **WIP:** to prevent a **Work In Progress** merge request from being merged before it's ready.
Add [description templates](#) to help your contributors communicate effectively!

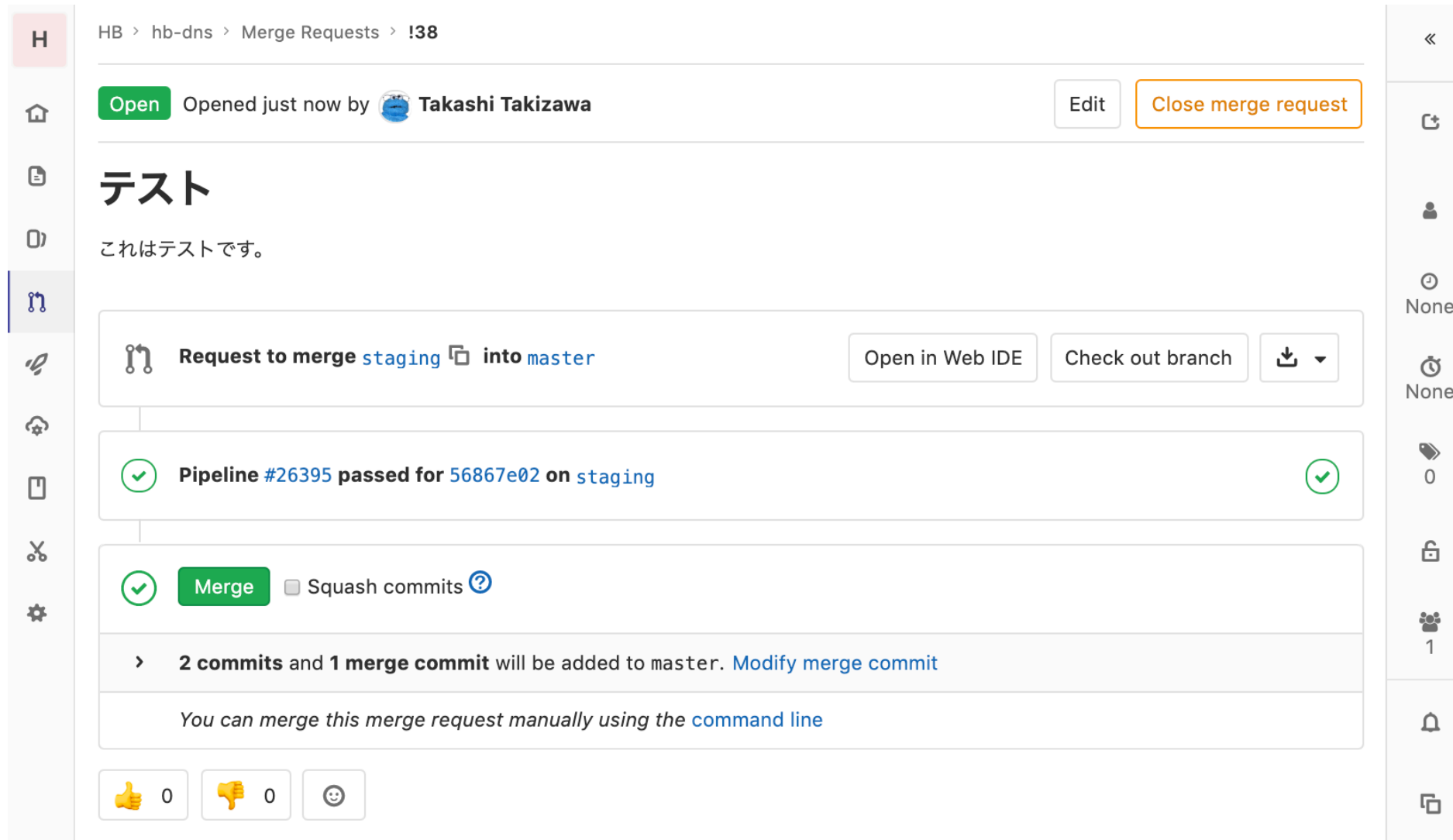
Description

Write Preview **B I " <> @ ☰ ☷**

これはテストです。

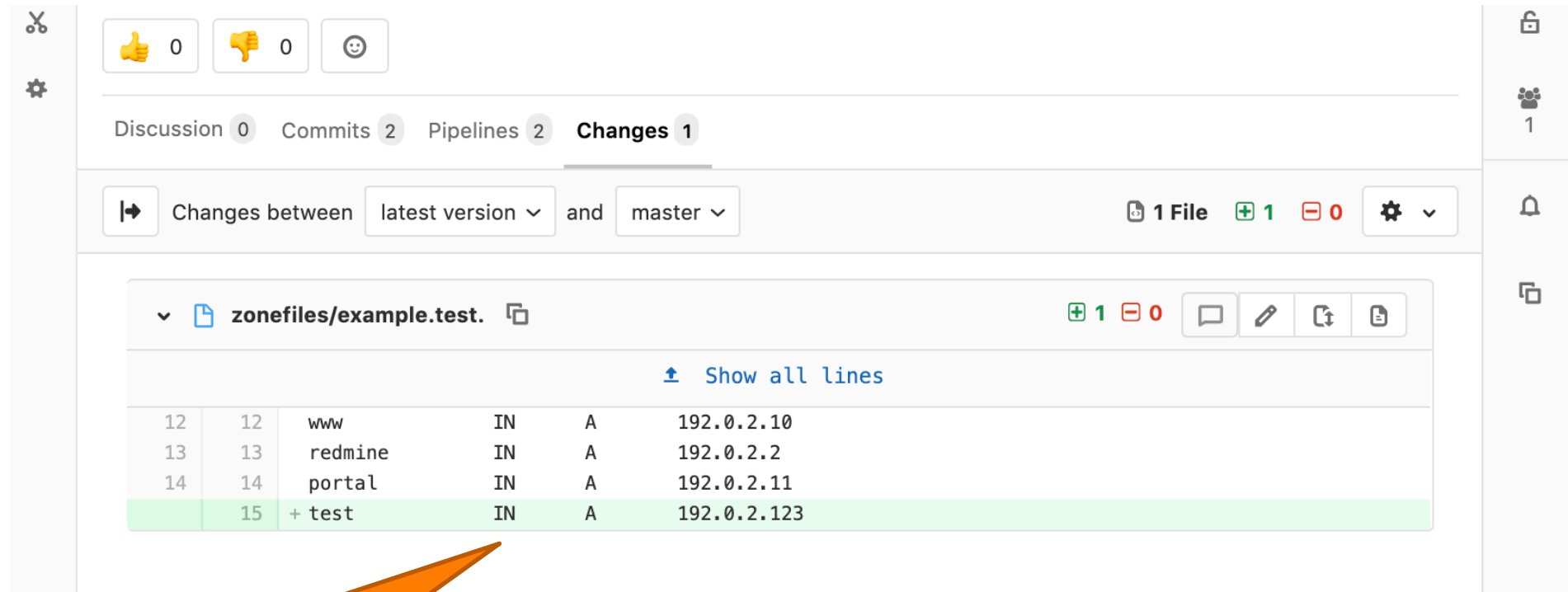
[Markdown](#) and [quick actions](#) are supported [Attach a file](#)

GitLab CI/CD: Merge Requestの確認とマージ



The screenshot shows a GitLab Merge Request interface. At the top, the breadcrumb navigation reads "HB > hb-dns > Merge Requests > !38". Below this, a green "Open" button is followed by the text "Opened just now by Takashi Takizawa". To the right are "Edit" and "Close merge request" buttons. The main title of the merge request is "テスト" (Test), with the description "これはテストです。" (This is a test). A section titled "Request to merge staging into master" includes buttons for "Open in Web IDE", "Check out branch", and a download icon. Below this, a green checkmark indicates "Pipeline #26395 passed for 56867e02 on staging". The "Merge" button is highlighted in green, with a checkbox for "Squash commits" and a help icon. A summary line states "2 commits and 1 merge commit will be added to master. Modify merge commit". A note at the bottom says "You can merge this merge request manually using the command line". At the very bottom, there are thumbs up/down and a smiley face icon, all with a count of 0. The left sidebar contains navigation icons, and the right sidebar contains utility icons.

GitLab CI/CD: Merge Requestの確認とマージ



Discussion 0 Commits 2 Pipelines 2 **Changes 1**

Changes between latest version and master 1 File +1 -0

zonefiles/example.test. +1 -0

Show all lines

12	12	www	IN	A	192.0.2.10
13	13	redmine	IN	A	192.0.2.2
14	14	portal	IN	A	192.0.2.11
	15	+ test	IN	A	192.0.2.123

変更内容を最終確認し、
問題なければMergeボタン
をクリックする。

GitLab CI/CDのジョブの確認画面（同期の実施）

H

Home

Jobs

Builds

Groups

Settings

Search

Help

```

*****
* example.test.
*****
* route53 (Route53Provider)
*   Create <ARecord A 900, test.example.test., ['192.0.2.123']> (zonefile)
*   Summary: Creates=1, Updates=0, Deletes=0, Existing Records=4
* azuredns (AzureProvider)
*   Create <ARecord A 900, test.example.test., ['192.0.2.123']> (zonefile)
*   Summary: Creates=1, Updates=0, Deletes=0, Existing Records=4
*****

2019-11-18T11:40:08 [140041703348032] INFO  Route53Provider[route53] apply: making
changes
2019-11-18T11:40:08 [140041703348032] INFO  Route53Provider[route53] _apply:
zone=example.test., len(changes)=1
2019-11-18T11:40:09 [140041703348032] INFO  Route53Provider[route53] _apply:
sending change request for batch of 1 mods, 1 ResourceRecords
2019-11-18T11:40:09 [140041703348032] INFO  AzureProvider[azuredns] apply: making
changes
2019-11-18T11:40:10 [140041703348032] INFO  Manager sync:   2 total changes
▼
▼
▼
Job succeeded

```

production_update Retry

Duration: 50 seconds

Timeout: 1h (from project) ?

Runner: docker-runner (#4)

Tags: docker

Commit [8122b18b](#) 📄

Merge branch 'staging' into 'master'

✔ **Pipeline #26396** for **master**

test ▼

➔ ✔ **production_update**

ジョブの成功

Slackへのmasterブランチのジョブの実行結果の通知

GitLabの
Slack Integration
による通知



gitlab APP 8:36 PM

Takashi Takizawa (takizawa) opened !38 テスト in HB/hb-dns

Takashi Takizawa (takizawa) merged !38 テスト in HB/hb-dns

Takashi Takizawa (takizawa)

Pipeline #26396 has passed in 00:50

Branch

master

hb-dns | Today at 8:40 PM

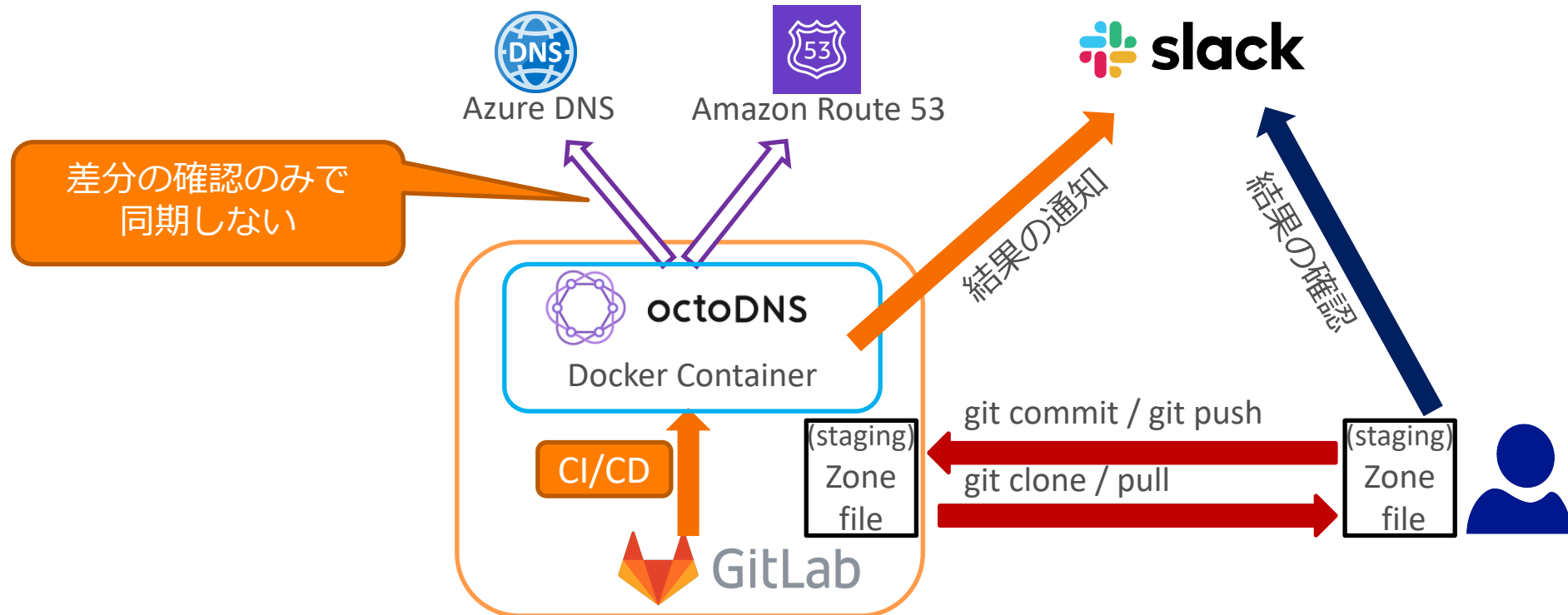
Commit

Merge branch 'staging' into 'master'

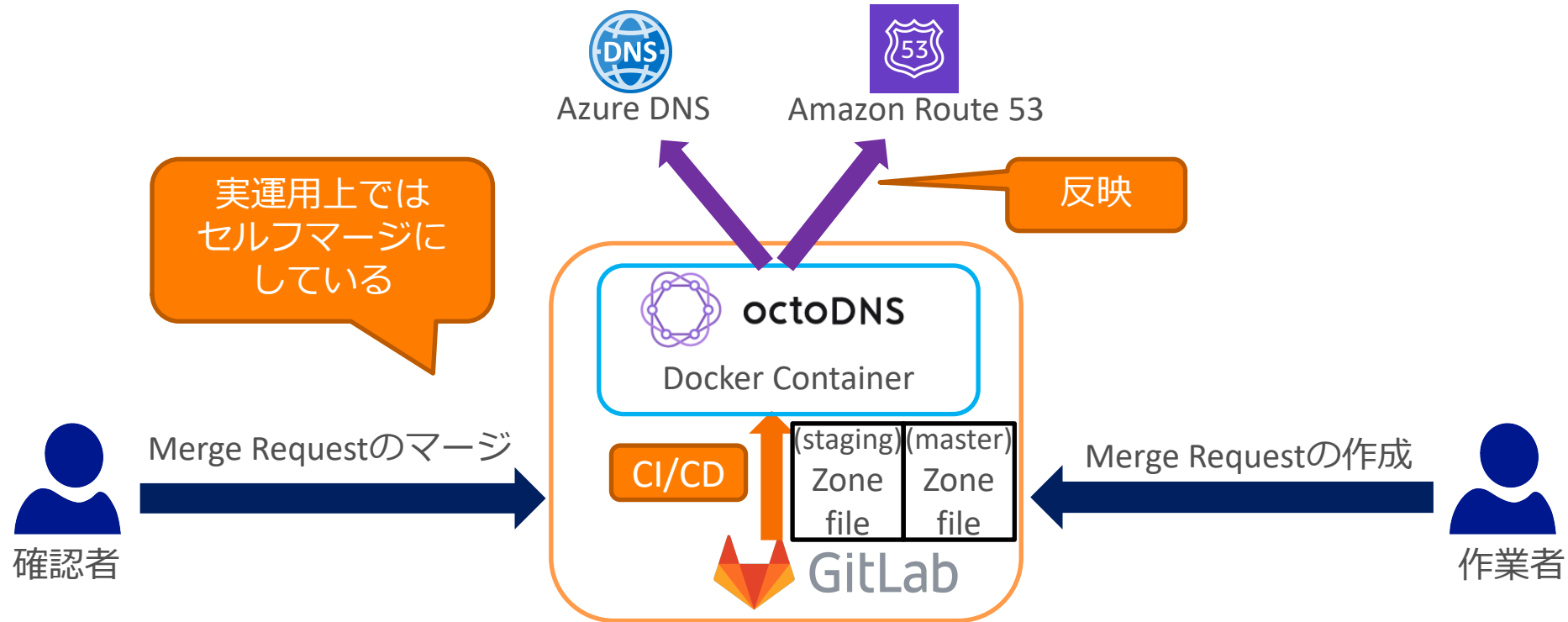
実行例は以上！

- 実際はdigで確認もするけど、例としては省略。

振り返り: stagingブランチ

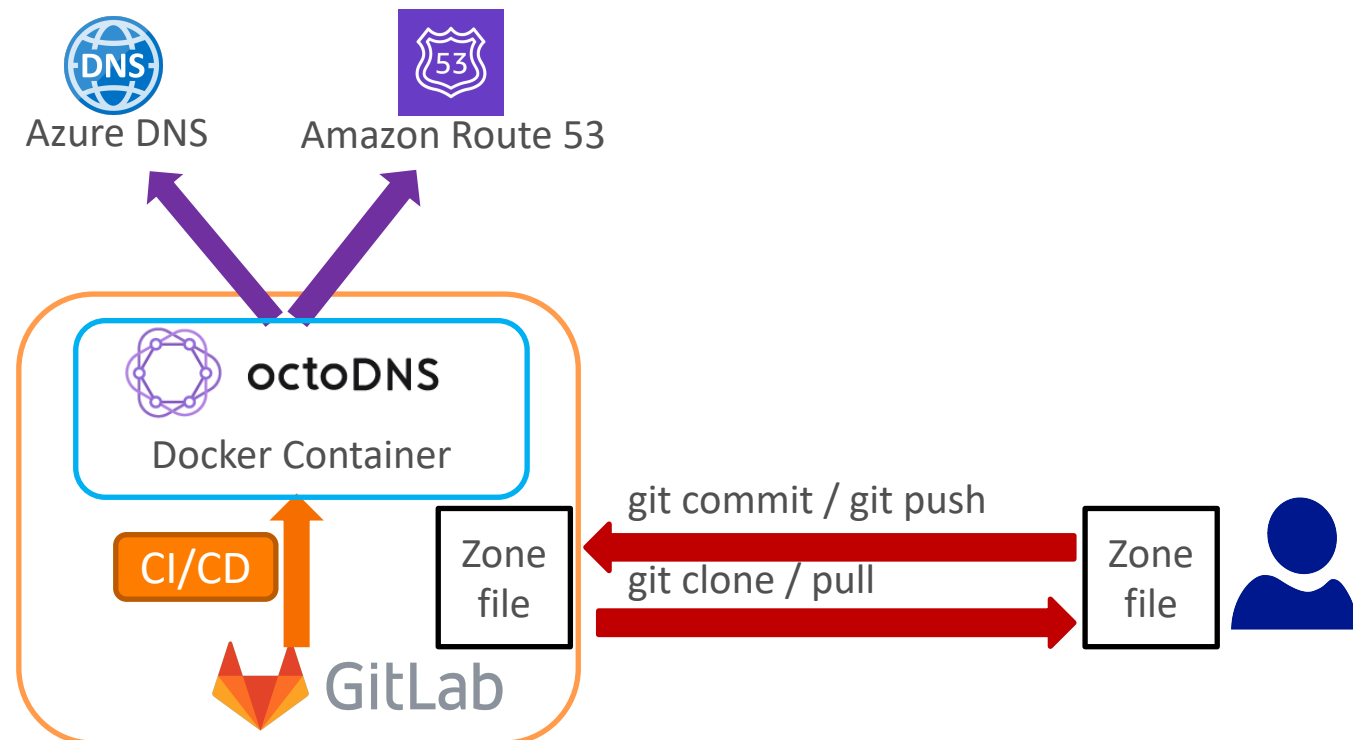


振り返り: masterブランチ



まとめ

- DNSゾーン管理ツール OctoDNS と
- SCM（ソースコード管理）ツール GitLab のCI/CD機能を使って、
- 複数DNSプロバイダー構成を運用する事例を紹介しました



おまけ: OctoDNSの知見

OctoDNSの知見

- OctoDNS v0.9.8時点での情報
- リソースレコード(RR)
 - SOAレコードとNSレコードは同期されない。
 - プロバイダーにより対応しているRRタイプは異なる。
- 安全機能 (octodns.provider.plan.Planクラスのraise_if_unsafe()で定義)
 - 10個以上のRRが存在するときに
 - 全体の30%以上のRRが更新される場合は中断する
 - 全体の30%以上のRRが削除される場合は中断する
 - --force オプションを付けるとこの安全機能を見捨てる

OctoDNSの知見

- プロバイダー毎の設定方法
 - 各プロバイダーのPythonモジュールのDocstringとして記述されている。

```
>>> import octodns.provider.route53
>>> help(octodns.provider.route53.Route53Provider)
Help on class Route53Provider in module octodns.provider.route53:

class Route53Provider(octodns.provider.base.BaseProvider)
|   Route53Provider(id, access_key_id=None, secret_access_key=None, max_changes=1000,
client_max_attempts=None, session_token=None, *args, **kwargs)
|
|   AWS Route53 Provider
|
|   route53:
|       class: octodns.provider.route53.Route53Provider
|       # The AWS access key id
|       access_key_id:
|       # The AWS secret access key
|       secret_access_key:
|       # The AWS session token (optional)
|       # Only needed if using temporary security credentials
|       session_token:
```

OctoDNSの知見

- ZoneFileSource
 - SOAレコードは必須（同期されないけど）
 - ファイル名は絶対ドメイン名にする。
 - ./zonefiles/example.test.

```
class ZoneFileSource(AxfrBaseSource):
    '''
    Bind compatible zone file source

    zonefile:
        class: octodns.source.axfr.ZoneFileSource
        # The directory holding the zone files
        # Filenames should match zone name (eg. example.com.)
        directory: ./zonefiles
```