

EASEL WebUI
AWS 環境構築手順書

株式会社 EASEL

目次

1.概要.....	1
2.本ドキュメントの流れ.....	2
3.アカウントの作成.....	3
3.1.AWS アカウント.....	3
3.2.Google アカウント (ES920LRBG2-K 利用時のみ).....	3
4.Elastic Beanstalk でアプリケーション作成.....	4
4.1.ログイン.....	4
4.2.リージョン確認.....	5
4.3.Elastic Beanstalk の選択.....	5
4.3.アプリケーションの作成.....	6
4.3.1.アプリケーション名の設定.....	7
4.3.2.プラットフォームの設定.....	7
4.3.3.アプリケーションコードの設定.....	8
4.3.3.オプション設定.....	10
4.3.3.アプリケーション作成開始.....	14
5.ローカル環境(PC)から EC2(仮想マシン)に接続.....	16
6.AWS S3 アクセスキー作成.....	19
7.編集と再アップロード.....	20
7.1.環境ファイル (.env) 編集.....	20
7.3.メールのソースコード編集.....	23
7.4.ソースコード環境の再アップロード.....	24
7.2.管理者用ログイン ID 及びパスワード作成.....	25
8.WEB UI 表示確認.....	27
9.Elastic Beanstalk からのイベントを受信する.....	28
10.ローカル環境(PC)から AWS RDS に接続.....	30
11. エクスポート／インポート.....	35
11.1.弊社サーバーからのエクスポート.....	35
11.2.構築環境へのインポート.....	35
12. WebUI の追加.....	37
12.1. ES920LRBG2-K WebUI に ES920LRTH2-K(ES920LRTH3-K) WebUI パッケージを適用.....	37
12.2. ES920LRTH2-K(ES920LRTH3-K) WebUI に ES920LRBG2-K WebUI パッケージを適用.....	37
12.3.利用アプリケーションの選択.....	39

1.概要

本ドキュメントは、EASEL 提供 WebUI を Amazon Web Service(以降、AWS と略)上に構築する為の手順書となり、下記キットのいずれかを購入し、弊社 WebUI を利用されている方が対象となります。

- ・ ES920LRBG2-K (位置通知基本キット)
- ・ ES920LRTH2-K (温湿度管理基本キット)
- ・ ES920LRTH3-K (CO2/温湿度管理基本キット)

AWS 上に構築する際、本ドキュメントでは AWS Elastic Beanstalk(以降、Elastic Beanstalk と略)を使用することを前提としています。Elastic Beanstalk は、AWS 上に比較的短時間に WebUI を構築することができ、その後もリソース制御やスケーリングを簡単に行えるサービスです。

Elastic Beanstalk: <https://aws.amazon.com/jp/elasticbeanstalk/>

尚、本 WebUI には PHP WEB フレームワークの Laravel v5.5 をベースに作成しています。

2.本ドキュメントの流れ

本ドキュメントでは、3章～8章にて WebUI を Elastic Beanstalk を使用して構築する手順について説明しています。



Elastic Beanstalk での WebUI 構築フロー

9章では、AWS Relational Database System(RDS)のテーブルデータに、MySQL Workbench を使用してローカル環境から閲覧・操作できる方法について説明しています。

10章では、弊社 WebUI の使用したデータをエクスポートし、構築した WebUI へインポートする方法について説明しています。

最後の 11 章では、ES920LRBG2-K WebUI と ES920LRTH2-K(ES920LRTH3-K) WebUI の WebUI を一

つの環境上に構築するための方法について説明しています。

3.アカウントの作成

AWS 及び Google のアカウント登録

AWS と Google にアカウント登録します。(既に登録済みの場合この手順は不要です。)

3.1.AWS アカウント

AWS : <https://aws.amazon.com/jp/>

※AWS の登録にはキャッシュカードが必要です。

3.2.Google アカウント(ES920LRBG2-K 利用時のみ)

Google アカウントは、Google Maps を利用する際の Key を作成するために必要です。

<https://accounts.google.com/signup/v2/webcreateaccount?hl=ja&flowName=GlifWebSignIn&flowEntry=SignUp>

(注) Google Maps は 2018 年 6 月 11 日より Google Maps Platform となりました。

Google からキーを取得しないサイトではマップ表示ができないようになっています。

また従来の API 体系にも変更が出ています。キーの取得に関してはこちらを参考に取得してください。

<https://developers.google.com/maps/documentation/javascript/get-api-key?hl=ja>

(公式) (英語)

本 WebUI 用では、API は Maps、Routes、Routes の全てを選択してください。

また、Google Maps の API は、ある一定の無料枠を超えると従量課金になりことにも留意ください。

<https://developers.google.com/maps/documentation/javascript/usage-and-billing?hl=ja>

(公式) (英語)

4.Elastic Beanstalk でアプリケーション作成

4.1.ログイン

AWS マネジメントコンソールにログインします。



AWS メイン画面からコンソールへのログイン

上記画面のボタンをクリックしてログイン画面に遷移します。

- ・ アカウントを作成したルートアカウントの場合は、登録に使用したメールアドレスとパスワードを入力してログインします。
- ・ IAM(AWS Identity and Access Management)ユーザーの場合は、アカウント、ユーザー名、パスワードを入力してログインします。

4.2.リージョン確認

AWS のリージョン表示が、「東京」リージョンであることを確認してください。

画面右上端に現在選択されているリージョンが表示されています（下画像の赤右枠部）になります。

リージョンが「東京」と表示されていない場合は、ドロップリストから「アジアパシフィック（東京） ap-northeast-1」を選択してください。選択しますと、リージョン表示が「東京」に変わり、以降の AWS 上での作業は自動的に東京リージョンで実施することになります。

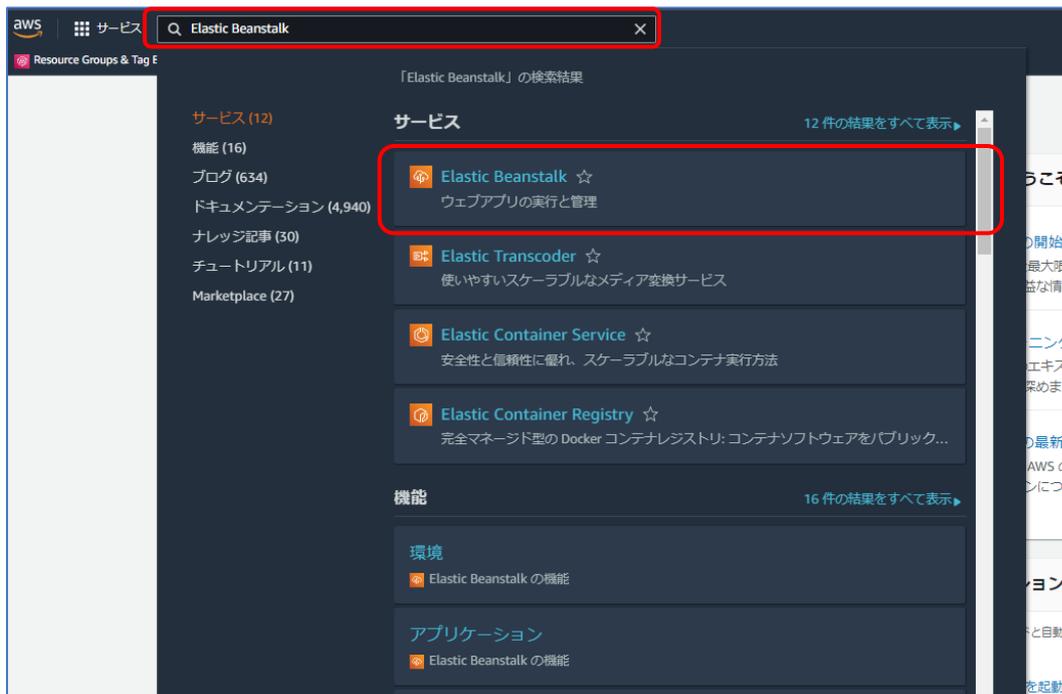


リージョン確認

4.3.Elastic Beanstalk の選択

ログインすると、コンソールのホーム画面が表示されます。

画面左上の検索窓に「Elastic Beanstalk」を入力して表示された検索結果から Elastic Beanstalk を選択します。



コンソール画面のサービス検索結果

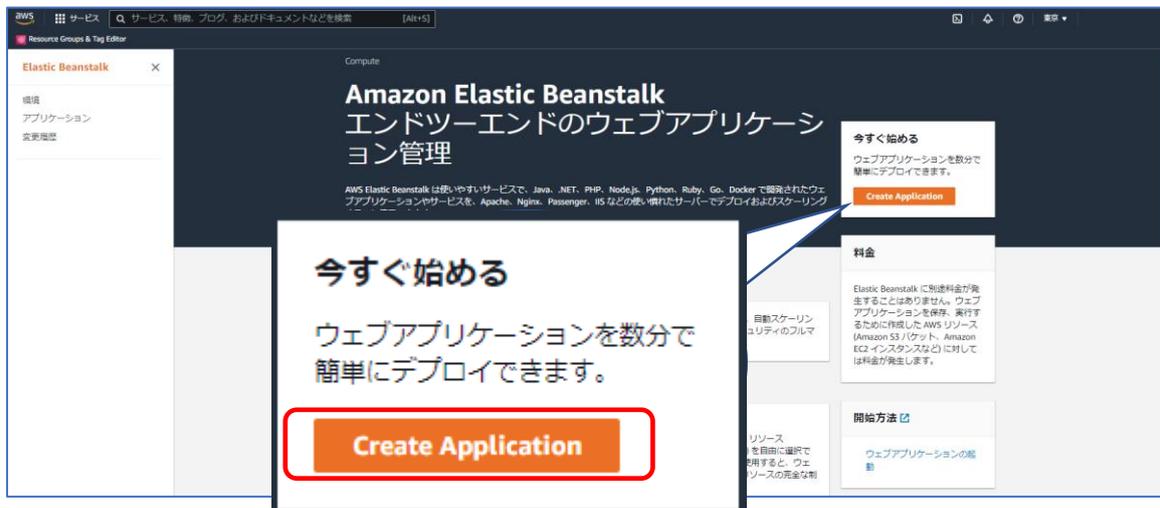


Elastic Beanstalk の画面

※Elastic Beanstalk を既に利用している場合は、Elastic Beanstalk で作成したアプリケーションの一覧が表示されます。

4.3. アプリケーションの作成

画面右の「今すぐ始める」パネルにある「Create Application」をクリックします。

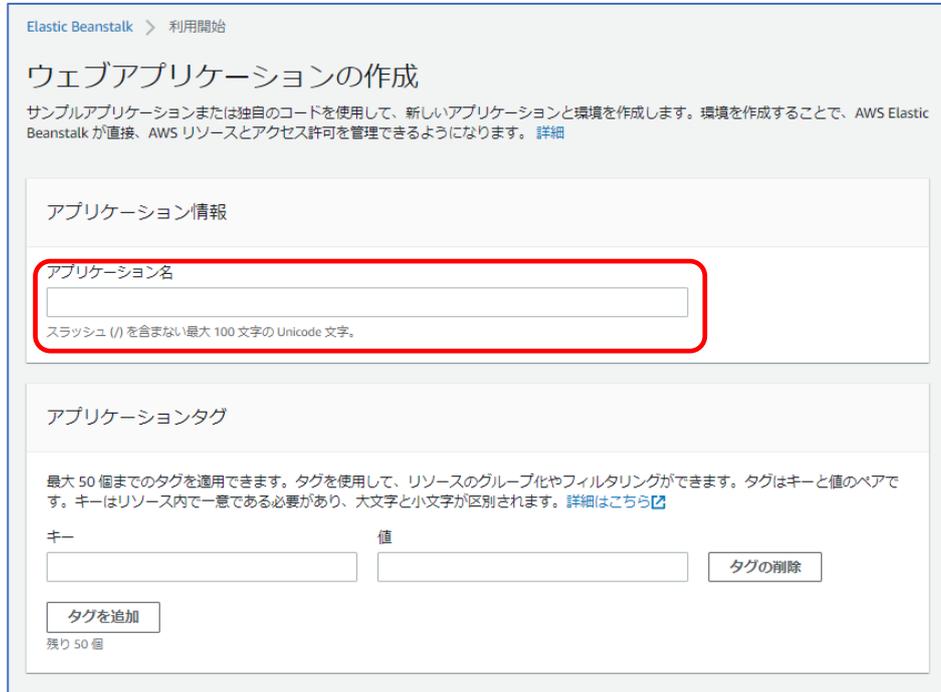


新しいアプリケーションの作成

クリックすると、「ウェブアプリケーションの作成」画面に遷移します。

4.3.1.アプリケーション名の設定

アプリケーション情報のアプリケーション名を入力します。アプリケーション名は、Elastic Beanstalk の環境名として今後「アプリケーション名-env」で表示します。また、Elastic Beanstalk で作成する Web アプリケーションのドメインの先頭にも「アプリケーション名-env」が含まれます。



Elastic Beanstalk > 利用開始

ウェブアプリケーションの作成

サンプルアプリケーションまたは独自のコードを使用して、新しいアプリケーションと環境を作成します。環境を作成することで、AWS Elastic Beanstalk が直接、AWS リソースとアクセス許可を管理できるようになります。 [詳細](#)

アプリケーション情報

アプリケーション名

スラッシュ (/) を含まない最大 100 文字の Unicode 文字。

アプリケーションタグ

最大 50 個までのタグを適用できます。タグを使用して、リソースのグループ化やフィルタリングができます。タグはキーと値のペアです。キーはリソース内で一意である必要があり、大文字と小文字が区別されます。 [詳細はこちら](#)

キー 値

タグの削除

タグを追加

残り 50 個

ウェブアプリケーションの作成

4.3.2.プラットフォームの設定

次にプラットフォームの設定を行います。

- ・プラットフォーム⇒「PHP」を選択。
- ・プラットフォームのブランチ「PHP 7.4 running on 64bit Amazon Linux2」を選択
- ・プラットフォームのバージョン Recommended のバージョンを選択



プラットフォーム

プラットフォーム

PHP

プラットフォームのブランチ

PHP 7.4 running on 64bit Amazon Linux 2

警告
廃止されたプラットフォームのブランチを本稼働環境で使用することは推奨されません。 [詳細はこちら](#)

プラットフォームのバージョン

3.3.15 (Recommended)

4.3.3.アプリケーションコードの設定

アプリケーションコードは「コードのアップロード」を選択し、ソースコード元では、ソースコード元は「ローカルファイル」を選択し、「ファイル選択」から WebUI のソースコードの zip ファイル(*)を指定します。

アプリケーションコード

サンプルアプリケーション
サンプルコードを今すぐ開始する。

コードのアップロード
コンピュータからソースバンドルをアップロードするか、Amazon S3 からソースバンドルをコピーします。

ソースコード元

バージョンラベル
このバージョンのアプリケーションコードの一意の名前。

ソースコード元
最大サイズは 512 MB

ローカルファイル

パブリック S3 URL

ファイルがアップロードされていません

アプリケーションコードの設定

(*) CD-ROM に同梱された WebUI のソースコードは、2 つの zip ファイルからなります。まずベースパッケージ(es920web_base_package.zip)を PC の適当なフォルダに解凍します。次に、もう一方の Web アプリケーションパッケージの zip も解凍先に先程のペースパッケージ解凍先を指定します。

名前	更新日時	種類	サイズ
.ebextensions	2022/06/15 17:56	ファイル フォルダ	
.platform	2022/06/15 20:28	ファイル フォルダ	
app	2022/06/15 19:04	ファイル フォルダ	
bootstrap	2022/06/15 17:54	ファイル フォルダ	
config	2022/06/15 17:54	ファイル フォルダ	
database	2022/06/15 17:53	ファイル フォルダ	
public	2022/06/15 17:55	ファイル フォルダ	
resources	2022/06/15 17:53	ファイル フォルダ	
routes	2022/06/15 17:55	ファイル フォルダ	
storage	2022/06/15 17:53	ファイル フォルダ	
tests	2022/06/15 17:53	ファイル フォルダ	
vendor	2022/06/15 17:54	ファイル フォルダ	
.env	2022/06/15 21:17	ENV ファイル	1 KB
.env.example	2020/01/23 19:41	EXAMPLE ファイル	1 KB
.gitattributes	2020/01/23 19:41	Git Attributes ソー...	1 KB
.gitignore	2020/01/23 19:41	Git Ignore ソースフ...	1 KB
artisan	2020/01/23 19:41	ファイル	2 KB
composer.json	2020/01/26 22:45	JSON ソースファイル	2 KB
composer.lock	2020/01/26 22:45	LOCK ファイル	184 KB
package.json	2020/01/23 19:41	JSON ソースファイル	2 KB
phpunit.xml	2020/01/23 19:41	XML ドキュメント	2 KB
readme.md	2020/01/23 19:41	Markdown ソースフ...	4 KB
server.php	2020/01/23 19:41	PHP ファイル	1 KB
webpack.mix.js	2020/01/23 19:41	JS ファイル	1 KB

2つのzipを解凍して展開されたソースコード

初回のソースコードアップロードでは、解凍したディレクトリの直下にある .ebextensions と .platform フォルダ以外を選択してzipを作成してください。

名前	更新日時	種類	サイズ
.ebextensions	2022/06/15 17:56	ファイルフォルダー	
.platform	2022/06/15 20:28	ファイルフォルダー	
app	2022/06/15 19:04	ファイルフォルダー	
bootstrap	2022/06/15 17:54	ファイルフォルダー	
config	2022/06/15 17:54	ファイルフォルダー	
database	2022/06/15 17:53	ファイルフォルダー	
public	2022/06/15 17:55	ファイルフォルダー	
resources	2022/06/15 17:53	ファイルフォルダー	
routes	2022/06/15 17:55	ファイルフォルダー	
storage	2022/06/15 17:53	ファイルフォルダー	
tests	2022/06/15 17:53	ファイルフォルダー	
vendor	2022/06/15 17:54	ファイルフォルダー	
.env	2022/06/15 21:17	ENV ファイル	1 KB
.env.example	2020/01/23 19:41	EXAMPLE ファイル	1 KB
.gitattributes	2020/01/23 19:41	Git Attributes ソー...	1 KB
.gitignore	2020/01/23 19:41	Git Ignore ソースフ...	1 KB
artisan	2020/01/23 19:41	ファイル	2 KB
composer.json	2020/01/26 22:45	JSON ソースファイル	2 KB
composer.lock	2020/01/26 22:45	LOCK ファイル	184 KB
package.json	2020/01/23 19:41	JSON ソースファイル	2 KB
phpunit.xml	2020/01/23 19:41	XML ドキュメント	2 KB
readme.md	2020/01/23 19:41	Markdown ソースフ...	4 KB
server.php	2020/01/23 19:41	PHP ファイル	1 KB
webpack.mix.js	2020/01/23 19:41	JS ファイル	1 KB

.ebextensions、.platform フォルダは最初のアップロード用 zip 作成には含めない

最初のアップロード用 zip 作成範囲

解凍・展開したソースファイルのトップディレクトリ

zip化したファイルをアップロードデータに指定します。

ソースコード元

バージョンラベル
このバージョンのアプリケーションコードの一意の名前。

ソースコード元
最大サイズは 512 MB

ローカルファイル
 パブリック S3 URL

ファイル名: test_web.zip
 ファイルがアップロードされました

▶ アプリケーションコードタグ

ソースコード(zip)のアップロード

4.3.3.オプション設定

アプリケーションアップロード完了後、右下にある「より多くのオプションの設定」ボタンをクリックします。

ソースコード元

バージョンラベル
このバージョンのアプリケーションコードの一意の名前。
test-web-source

ソースコード元
最大サイズは 512 MB

ローカルファイル
 パブリック S3 URL

📁 ファイルを選択
ファイル名: test_web.zip
🟢 ファイルがアップロードされました

▶ アプリケーションコードタグ

キャンセル **より多くのオプションの設定** アプリケーションの作成

「ウェブアプリケーションの作成」画面最下の「より多くのオプションの設定」の選択

Elastic Beanstalk の環境設定画面に遷移します。

Elastic Beanstalk > 利用開始

Testwebui-env の設定

プリセット
ユースケースに一致するプリセットから開始するか、カスタム設定を選択して推奨値を解除して、サービスのデフォルト値を使用します。

設定のプリセット

単一インスタンス (無料利用対象)
 単一インスタンス (スポットインスタンスを使用)
 高可用性
 高可用性 (スポットインスタンスとオンデマンドインスタンスを使用)
 カスタム設定

プラットフォーム

PHP 7.4 running on 64bit Amazon Linux 2/3.3.15
プラットフォームのバージョンを変更する

ソフトウェア 編集

AWS X-Ray 無効	ログのストリーミング: 無効 (デフォルト)	環境プロパティ: 0
ログのローテーション: 無効 (デフォルト)		

インスタンス 編集

Elastic Beanstalk 環境設定画面

本画面では、サーバーインスタンスの選択や、ロードバランサー(負荷分散)、データベース、スケーリングの様々な設定等ができます。本ドキュメントでは、最小限での WebUI の構築を優先して以下の項目のみを行います。

- ・ソフトウェア：Web サーバー(ソフトウェア)の選択とドキュメントルート設定
- ・データベース：WebUI で使用するデータベースの設定
- ・通知：Elastic Beanstalk 環境に変化があった場合、指定メールのアドレスで通知
- ・セキュリティ：Web サーバーにローカル環境から SSH 接続できるように設定

4.3.1.ソフトウェア

「ソフトウェア」の編集ボタンをクリックすると、「ソフトウェアの変更」画面に遷移します。



ソフトウェア設定

ソフトウェアの変更

以下の設定はコンテナの動作を制御し、キーと値のペアを OS 環境の値として渡せるようにします。 [詳細情報](#)

コンテナオプション

プロキシサーバー
Specifies which proxy server to be used for client connections. By choosing Apache, Elastic Beanstalk defaults to the latest version of Apache https that it supports. [Learn more](#) Static file mappings and gzip compression will not take effect if the proxy server is set to "None".
Nginx

ドキュメントのルート
パブリック向けウェブのルートとして機能するプロジェクトの子ディレクトリです。ルートドキュメントがプロジェクトディレクトリに保存されている場合は、この設定を「/」のままにします。ルートドキュメントがディレクトリ内 (/public など) にある場合、この値をそのディレクトリと一致するように設定します。値は「/」文字で開始します。また、先頭に「*」はつけられません。「*」（ワイルドカード）です。
/

メモリ制限
PHP 環境に割り当てられたメモリの量を指定します。この値は、/etc/php.d/ ディレクトリにある ini 設定ファイルに書き込まれます。
256M

Zlib 出力の圧縮
PHP が出力に圧縮を使用するかどうかを指定します。この値は、/etc/php.d/ ディレクトリにある ini 設定ファイルに書き込まれます。
Off

URL で fopen を許可します
PHP のファイル読み取りがウェブサイトや FTP サーバーなどのリモートロケーションからデータを取得できるかどうかを指定します。この値は、/etc/php.d/ ディレクトリにある ini 設定ファイルに書き込まれます。
On

表示エラー
エラーメッセージを出力に含めるかどうかを指定します。この値は、/etc/php.d/ ディレクトリにある ini 設定ファイルに書き込まれます。
Off

最大実行時間 (秒)
環境がスクリプトを終了するまでの最長時間を指定します。これは、書き込みが不十分なうちに、スクリプトがサーバーに到達付けられないように役に立ちます。
60

「ソフトウェアの変更」画面

ソフトウェアの変更画面ではコンテナオプション、AWS X-Ray、S3 ログストレージ、CloudWatch Logs へのインスタンスログのストリーミング、環境プロパティ等の設定ができますが、ここではコンテナオプションの変更点のみ記載します。

- ・プロキシサーバー⇒Apache に設定
- ・ドキュメントのルート⇒/public に設定



コンテナオプション

プロキシサーバー
Specifies which proxy server to be used for client connections. By choosing **Apache**, Elastic Beanstalk defaults to the latest version of Apache https that it supports. [Learn more](#) Static file mappings and gzip compression will not take effect if the proxy server is set to "None".

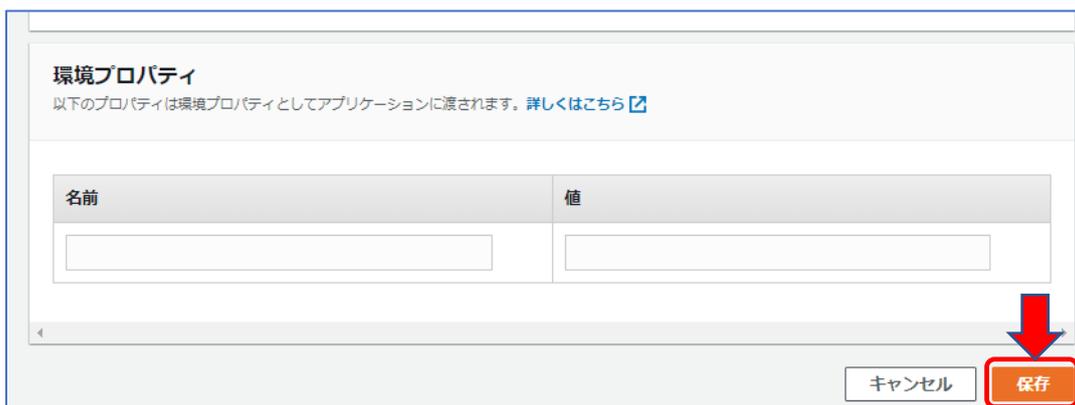
Apache

ドキュメントのルート
パブリック向けウェブのルートとして機能するプロジェクトの子ディレクトリです。ルートドキュメントがプロジェクトディレクトリに保存されている場合は、この設定を "A" のままにします。ルートドキュメントが子ディレクトリ内 (/public など) にある場合、この値をそのディレクトリと一致するように設定します。値は "/" 文字で開始します。また、先頭に "." はつけられません。"." (ピリオド) です。

/public

「ソフトウェアの変更」 > 「コンテナオプション」の変更

変更後に画面最下の「保存」ボタンをクリックすると設定が保存され、環境設定画面に戻ります。



環境プロパティ
以下のプロパティは環境プロパティとしてアプリケーションに渡されます。詳しくはこちら

名前	値
<input type="text"/>	<input type="text"/>

キャンセル 保存

4.3.2. データベース

「データベース」の編集ボタンをクリックすると、「データベースを追加」画面に遷移します。

この環境は VPC の一部ではありません。

データベース 編集

エンジン: --	インスタンスクラス: --	マルチアベイラビリティゾーン: --
	ストレージ (GB): --	

タグ 編集

タグ:

データベース設定

本設定を行う事で Elastic Beanstalk 環境作成時にデータベースとして Amazon Relational Database Services(以降 RDS と略)のインスタンスを作成します。

データベースを追加

環境と統合された RDS SQL データベースを追加します。 [詳細はこちら](#)

スナップショットを復元 - オプション

以前に使用したデータベースから既存のスナップショットを復元します。

スナップショット

なし ▼ ↺

データベース設定

環境のデータベースのエンジンとインスタンスタイプを選択します。

エンジン

mysql ▼

エンジンバージョン

8.0.28 ▼

インスタンスクラス

db.t2.micro ▼

データベース編集 (データベースを追加)

本画面では下記データベースの設定を行います。

- エンジン ⇒mysql を選択
- エンジンバージョン⇒5.7.xx (xx の数値が大きいものを選択)
- インスタンスクラス⇒db.t2.micro

インスタンスにより性能が異なります。性能が良いインスタンスほど金額

が高くなります。ここでは安価な db.t2.micro を選択します（後から変更することも可能です）。

- ・ストレージ ⇒5(GB)
データベースの容量です。容量が大きいほど金額が高くなります。ここでは最小の 5GB を選択します(後から変更することも可能です)。
- ・ユーザー名 ⇒AWS RDS にアクセスする際に必要なユーザー名を入力します。
- ・パスワード ⇒AWS RDS にアクセスする際に必要なパスワードを入力します。

※ユーザー名とパスワードは、後の WebUI の環境設定編集で使用しますので、一時的にメモに残す等忘れないようにしてください。

- ・アベイラビリティ⇒低(1つのAZ)
冗長構成の有無を選択します。「低(1つのAZ)」は単一の RDS での運用となり、冗長構成は無となります。「高(マルチAZ)」を選択すると、冗長構成のデータベースを構築しますが、その分の使用料金が掛かります(後で変更可能です)。ここではコストが掛からない「低(1つのAZ)」を選択します。
- ・データベース削除ポリシー⇒スナップショットの作成
Elastic Beanstalk 環境を削除する時のデータベースを残すか削除するかの設定になります。環境の削除時にデータベースも削除する場合は、「削除」を選択します。環境を削除してもデータベースを利用する場合は、「スナップショット」の場合、必要な場合にスナップショットから復元して使用することが可能となり、「保持」の場合、そのまま別環境で使用することが可能になります。

変更後に画面最下の「保存」ボタンをクリックすると設定が保存され、環境設定画面に戻ります。

加するとき、またはスタンドアロンデータベースを作成するときに、スナップショットからデータベースを復元できます。データベースのスナップショットの保存には料金が発生する場合があります。

保持
デカップリングされたデータベースは、引き続き Elastic Beanstalk の外部で使用および運用できます。

削除
Elastic Beanstalk はデータベースを終了します。データベースは使用できなくなります。

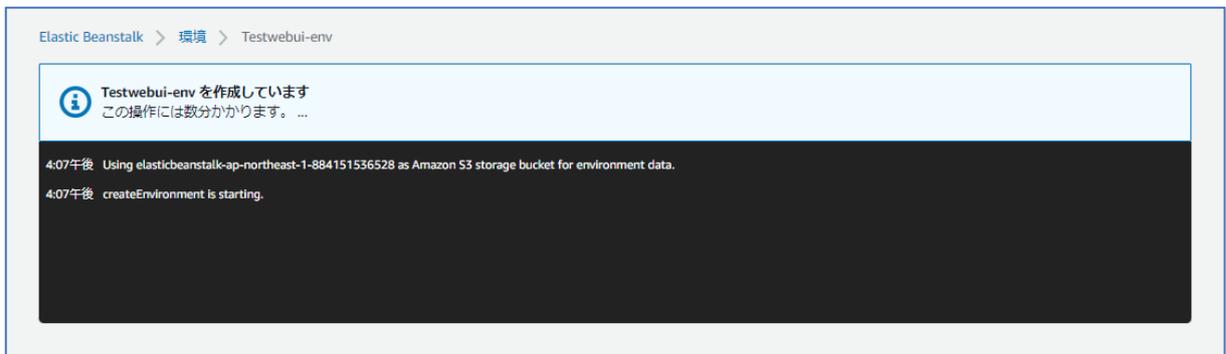
キャンセル 保存

4.3.3.アプリケーション作成開始

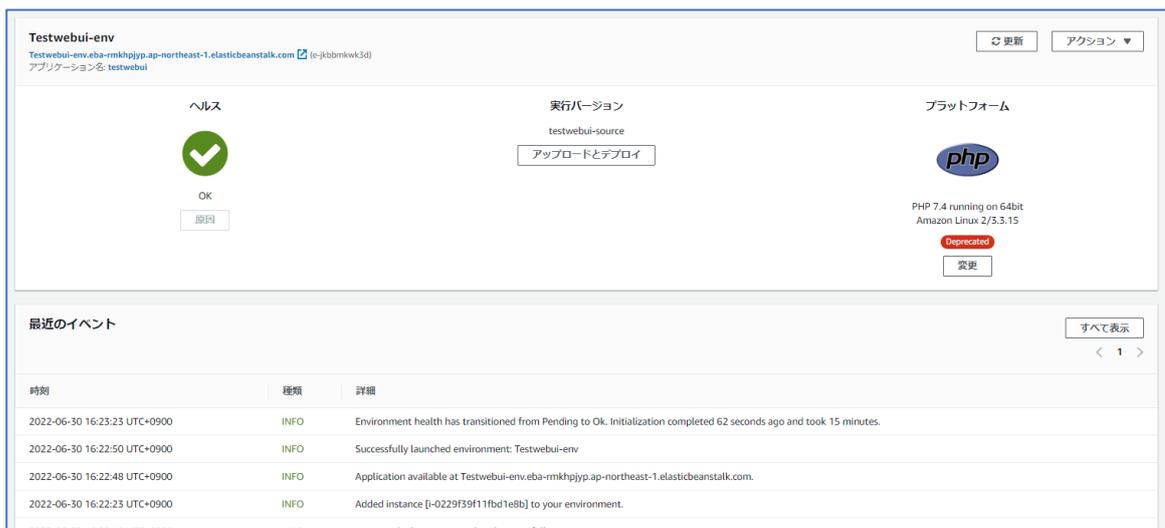
最後に環境設定画面の最下にある「アプリの作成」ボタンをクリックします。



「アプリ作成」ボタンをクリックすると、Elastic Beanstalk のアプリ環境作成画面に切り替わります。



Elastic Beanstalk のアプリ環境作成には約 15~20 分掛かります。環境が出来上がると、下記の画面に切り替わります。



5. ローカル環境から EC2(仮想マシン)に接続

Elastic Beanstalk で作成された仮想マシン(AWS EC2)にローカル環境から TeraTerm 等のターミナルソフトを使用して SSH 接続できるようにします。

本設定前に予めキーペアを作成する必要があります。ブラウザの別タブにて AWS EC2 の画面を開きます。

【キーペアの作成】

- ① 左メニューの「ネットワーク&セキュリティ」から「キーペア」を選択します。

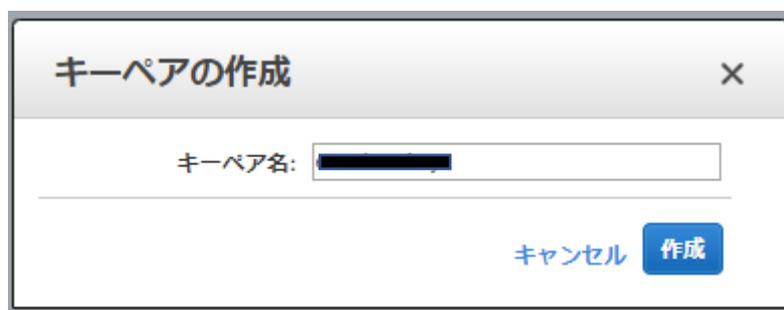


EC2 ネットワーク&セキュリティ選択

- ② 画面左上にある「キーペアの作成」を選択します。



- ③ キーペア名入力ダイアログが出るのでキーペア名を入力し、「作成」ボタンを選択します。



キーペア作成ダイアログ

- ④ 作成と同時に、キーファイル(.pem ファイル)がブラウザからダウンロードされます。

ダウンロードした pem ファイルはローカル環境から Elastic Beanstalk で作成した Web サーバー (EC2) に SSH 接続するために必要になります。

【セキュリティにキーペア名設定】

① Elastic Beanstalk の画面からサイドメニューにある、作成した環境の「設定」をクリックします。



設定画面 (セキュリティ設定)

Elastic Beanstalk の環境設定画面に遷移するので、「セキュリティ」の編集ボタンをクリックして「セキュリティの変更」画面に遷移します。



Elastic Beanstalk セキュリティ設定

② 「仮想マシンアクセス許可」の EC2 キーペアのドロップリストから先程作成したキーペア名を選択し、右下の「適用」を選択します。

セキュリティの変更

サービスロール

サービスロール
aws-elasticbeanstalk-service-role ▼ ⌂

仮想マシンアクセス許可

EC2 キーペア
[REDACTED] ▼ ⌂

IAM インスタンスプロフィール
aws-elasticbeanstalk-ec2-role ▼ ⌂

キャンセル 続行 適用

設定画面（セキュリティ変更）

- ③ Warnings が表示されますが、右下の「確認」ボタンをクリックします。

サービスメッセージ

warnings (1) errors

⚠ Changes to option EC2KeyName settings will not take effect immediately. Each of your existing EC2 instances will be replaced and your new settings will take effect then.

aws:autoscaling:launchconfiguration:EC2KeyName "null" to "[REDACTED]"

キャンセル 確認

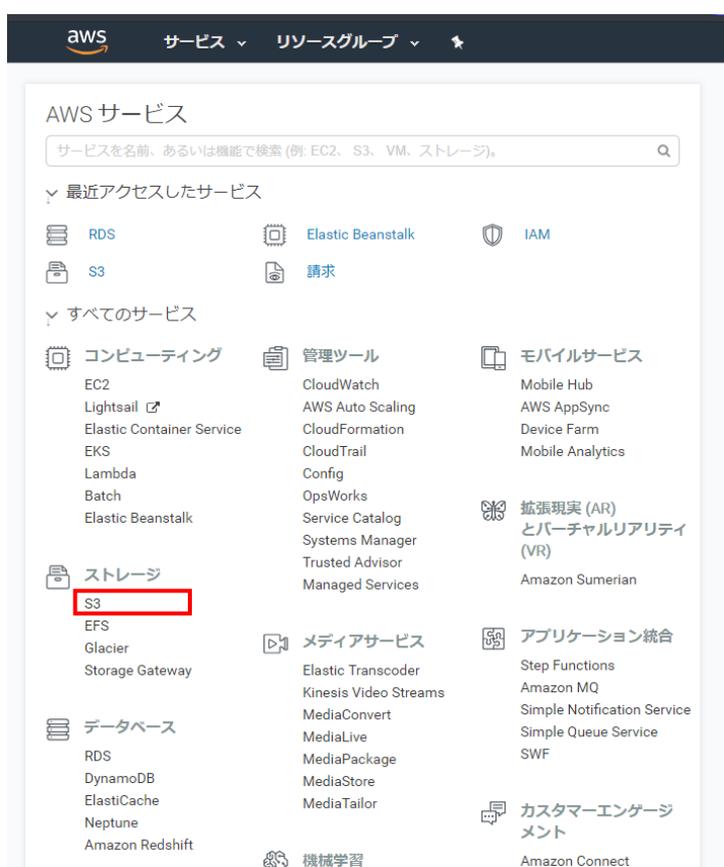
設定画面（Warning 表示）

Elastic Beanstalk が環境の更新作業に入ります。（完了まで数分掛かる場合があります。）

6.AWS S3 アクセスキー作成

AWS 上のストレージである S3(Simple Storage Service)にアクセスするための設定を行います。
AWS S3 のバケットは、Elastic Beanstalk がアプリケーション作成時に自動で作成されます。
本 WebUI は、データベースのバックアップとソースコードのバックアップを S3 に保存できるようになっています。データベースのバックアップを RDS のスナップショットに委任する場合は、本設定は不要です。

AWS マネジメントコンソール画面左上の AWS のアイコンから、AWS のメインコンソール画面に遷移し、全てのサービスの中から「ストレージ」⇒「S3」を選択します。



AWS マネジメントコンソール画面 (S3 選択)

既に elasticbeanstalk-ap-northeast-1-xxxxxxxxxxxx というバケットが確認できます。

ここには既にアップロードしたソースコード(.zip)が置いてあります。

このストレージに WebUI プログラムからアクセスできるように、S3 へのアクセスキーとシークレットキーを取得し、取得したキーの値を後ほど説明する WEB UI の環境ファイル(.env)に設定します。

AWS_KEY と AWS_SECRET の取得方法は以下の URL を参照してください。

<https://aws.amazon.com/jp/developers/access-keys/>

7.編集と再アップロード

Elastic Beanstalk の環境が作成されたので、その環境を WebUI で使用できるようにソースコード内の環境ファイルとソースコードの一部を編集します。

[編集内容]

- 1.環境ファイル(.env)設定
- 2.管理者用ログイン ID 及びパスワード作成
- 3.WEB UI 用メールのソースコード編集

7.1.環境ファイル (.env) 編集

環境ファイル(.env)は、WebUI のソース解凍したフォルダのディレクトリ直下にあります。オリジナルの.env ファイルはリネームしてバックアップしておいてください。

■APP_KEY

ローカル環境に php の composer(*)がインストールされている場合、解凍したディレクトリにて「php artisan key:generate」コマンドを入力することで.env の APP_KEY にキーが設定されます。

(*)composer とは、PHP の必要なライブラリ群やパッケージを管理するためのツールです。

ローカル環境に composer がインストールされていない場合は、composer 公式サイトからダウンロードください。

<https://getcomposer.org/download/>

注意)現在の composer のバージョンが 2.*ですが、弊社から提供している WebUI が composer のバージョン 1.*に依存しているため、警告やエラーが発生する場合は、下記コマンドにて 1.*バージョンをに下げて試してみてください。

composer のバージョンを 2.x から 1.x にする

```
composer self-update -1
```

composer のバージョンを 1.x から 2.x に戻す

```
composer self-update -2
```

次に、.env ファイルを開いて編集していきます。

■APP_NAME

WEB UI アプリケーション名です。

画面ヘッダ部左上の表示や、ユーザーへのメール配信テキストに埋め込まれます。

■ APP_ENV, APP_DEBUG, APP_LOG_LEVEL

開発中の場合、下記の設定にしてください。エラー発生時にブラウザにエラー情報が出力されます。

APP_ENV='local'

APP_DEBUG='true'

APP_LOG_LEVEL='debug'

■ APP_URL

独自ドメインを設定していない場合は、Elastic Beanstalk の URL を記載します。

URL は Elastic Beanstalk のアプリケーションのダッシュボード画面中央に環境 ID と共に URL が表示されています。

■ データベースの設定

DB_CONNECTION=mysql

DB_HOST=(Elastic Beanstalk で構築した DB のエンドポイント) (例: xx~xx.rds.amazonaws.com)

DB_PORT=(Elastic Beanstalk で構築した DB のエンドポイントのポート番号)(例: 3306)

※DB_HOST、DB_PORT 共に Elastic Beanstalk のダッシュボードの「設定」画面の「データベース」で確認できます。

DB_DATABASE=ebdb

DB_USERNAME=(DB 作成時に入力したユーザー名)

DB_PASSWORD=(DB 作成時に入力したパスワード)

■ AWS S3 アクセス用キー(AWS S3 を利用しない場合は、入力不要)

AWS_KEY=取得した AWS キー

AWS_SECRET=取得した AWS 秘密キー

AWS_REGION=ap-northeast-1(東京リージョン)

AWS_BUCKET=バケット名

■ Google Maps API キー(ES920LRBG2-K 利用時のみ)

Google アカウントを作成後、Google Maps 用キーを取得した後こちらに設定します。

GMAPS_KEY=(Google Maps API キー)

■メール設定

WebUI にユーザーを登録した際の登録通知案内メール等を送信する場合のアドレスを設定します。

例)

```
MAIL_DRIVER=smtp
```

```
MAIL_HOST= (メールホスト) (例: smtp.xxxx.jp)
```

```
MAIL_PORT=465
```

```
MAIL_USERNAME=(使用するメールアドレス)
```

```
MAIL_PASSWORD= (上記メールアドレスのパスワード)
```

```
MAIL_ENCRYPTION=ssl
```

■ストレージ選択

WebUI の EC2 サーバー内に保存('local'を設定)するか、S3 に保存('s3'を設定)するかを設定します。

```
// バックアップ格納ストレージ
```

```
BACKUP_STORAGE='s3'
```

```
// フロアマップ画像格納ストレージ(ES920LRBG2-K のみ)
```

```
FLOORMAP_STORAGE='local'
```

■期限切れ通知メール設定

WebUI の有効期限が近づいた時にユーザーにメール通知する場合に設定します。

設定有効の場合、毎午前 9 時に確認し、該当ユーザーにメールで通知します。

```
// 有効期限切れ事前メール通知 有効(true) 無効(false)
```

```
USER_EXPIRE_NOTIFICATION=false
```

```
// ES920LRBG2-K 期限切れ日の 30 日前に通知
```

```
USER_GPSAPP_DIFF_EXPIRED_DATE=30
```

```
// ES920LRTH2(TH3)-K 期限切れ日の 30 日前に通知
```

```
USER_THAPP_DIFF_EXPIRED_DATE=30
```

7.3.メールのソースコード編集

ユーザーに通知するメール用の設定です。通知メールは下記の2種類になります。

- ・登録完了メール（認証付き）
- ・パスワードリセット完了メール

- ・WebUI 登録完了メール（認証付き）の文言変更
以下のソースコード内の文言を編集します。

```
¥app¥Mail¥EmailVerification.php
¥resources¥views¥emails¥verification.blade.php
¥resources¥views¥vendor¥notifications¥email.blade.php
```

- ・WebUI パスワードリセット完了メールの文言変更
以下のソースコード内の文言を編集します。

```
¥app¥Mail¥PasswordResetNotifyMail.php
¥app¥Notifications¥CustomPasswordReset.php
¥resources¥views¥emails¥userpasswordreset.blade.php
```

7.4. ソースコード環境の再アップロード

1.~3.の設定後に、今回は前回を含めなかった.ebextensions フォルダも含めて zip 圧縮します (zip ファイル名は前回とバージョン番号を変更したものにします)。

名前	更新日時	種類	サイズ
.ebextensions	2018/05/18 10:56	ファイルフォルダ	
app	2018/05/18 10:56	ファイルフォルダ	
bootstrap	2018/05/18 10:56	ファイルフォルダ	
config	2018/05/18 10:56	ファイルフォルダ	
database	2018/05/18 10:56	ファイルフォルダ	
nbproject	2018/05/18 10:56	ファイルフォルダ	
public	2018/05/18 10:56	ファイルフォルダ	
resources	2018/05/18 10:56	ファイルフォルダ	
routes	2018/05/18 10:56	ファイルフォルダ	
storage	2018/05/18 10:56	ファイルフォルダ	
tests	2018/05/18 10:56	ファイルフォルダ	
vendor	2018/05/18 10:56	ファイルフォルダ	
.env	2018/05/18 13:24	ENV ファイル	
.env.example	2018/01/25 18:17	EXAMPLE ファイル	
.gitattributes	2018/01/25 18:17	GITATTRIBU	
.gitignore	2018/01/25 18:17	GITIGNORE ファイル	1 KB
_ide_helper.php	2018/02/21 16:17	PHP ファイル	438 KB
artisan	2018/01/25 18:17	ファイル	2 KB
composer.json	2018/04/27 17:52	JSON File	2 KB
composer.lock	2018/04/27 17:35	LOCK ファイル	173 KB
list.txt	2018/02/28 17:32	テキスト文書	7 KB
package.json	2018/01/25 18:17	JSON File	2 KB
phpunit.xml	2018/01/25 18:17	XML ドキュメント	2 KB
readme.md	2018/01/25 18:17	MD ファイル	4 KB
server.php	2018/01/25 18:17	PHP ファイル	1 KB
webpack.mix.js	2018/01/25 18:17	JavaScript ファイル	1 KB

.ebextensions フォルダも含めた、全てのフォルダ、ファイルを zip 圧縮。

解凍したソースファイルのトップディレクトリ

ダッシュボード画面から、デプロイを選択してアップロードします。

時刻	種類	詳細
2018-05-18 13:36:32 UTC+0900	INFO	Environment health has transitioned from Info to Ok. Application update completed 83 seconds ago and took 2 minutes.
2018-05-18 13:34:56 UTC+0900	INFO	Environment update completed successfully.
2018-05-18 13:34:56 UTC+0900	INFO	New application version was deployed to running EC2 instances.
2018-05-18 13:33:54 UTC+0900	INFO	Deploying new version to instance(s).
2018-05-18 13:33:32 UTC+0900	INFO	Environment health has transitioned from Ok to Info. Application update in progress (running for 14 seconds).

環境ファイル編集後のアップロード

(※) 今回.ebextensions フォルダを含めて再アップロードしたのは、同フォルダ内にある init.config ファイル (ソースコードアップロード完了時に Elastic Beanstalk が実行する初期設定用ファイル) にて

データベーステーブル作成のコマンド(`php artisan migrate`)を実行するためです。

先の設定でデータベースは作成済みのため、`.extensions` をアップロードすることによりアップロード完了時に `php artisan migrate` が実行され RDS にデータベーステーブルが作成されます。

7.2.管理者用ログイン ID 及びパスワード作成

WebUI の管理者登録は WebUI が作成された EC2 インスタンスにログインしコマンド入力で行います。

- ① ローカル環境のターミナルソフトウェア(Teraterm など)より、SSH 接続にて Elastic Beanstalk で作成された EC2 インスタンス(名前は作成したアプリケーション環境と同名)にログインします。

ログイン名：`ec2-user`

秘密鍵：`AWS`

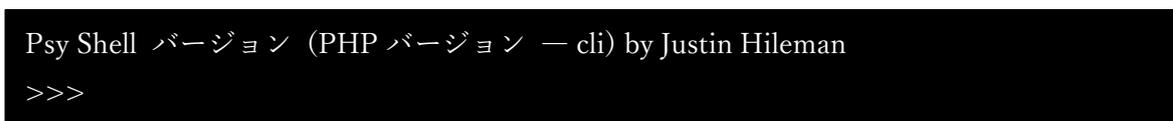


```

Elastic Beanstalk
Amazon Linux AMI

This EC2 instance is managed by AWS Elastic Beanstalk. Changes made via SSH
WILL BE LOST if the instance is replaced by auto-scaling. For more information
on customizing your Elastic Beanstalk environment, see our documentation here:
http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/customize-containers-ec2.h
tml
[ec2-user@ip: ~]$
```

- ② `sudo su -` コマンドを入力し、管理者権限ユーザーになります。
- ③ Elastic Beanstalk の EC2 では、WebUI のソースコードは `/var/app/current/` 配下に展開されます。(通常の EC2(Linux)の場合は、`/var/www/html/` 配下)
`cd /var/app/current` コマンドでソースコードが配置されたディレクトリに移ります。
- ④ `tinker` を起動し、管理者ユーザーを登録します。
`php artisan tinker` を入力し、`tinker` を起動します。
起動すると、下記のような画面がコンソール上に出力されます。



```

Psy Shell バージョン (PHP バージョン - cli) by Justin Hileman
>>>
```

>>>の後に下記の App¥Admin::create()を実行します。

```
>>>App¥Admin::create(['name'=>'admin','email'=>'dummy@easel5.com','password'=>b  
crypt('admin')]);
```

上記コード例は、ログイン ID : admin パスワード:admin とした場合のものになります。

赤字部分を適切なものを入力して実行してください。

‘name’ : 管理者用 WebUI にログインする ID を入力します。

‘email’ : 管理者用メールアドレス。ダミーのため、適当なアドレスでも可能です。(未入力は不可)

‘password’ : 管理者用 WebUI にログインするパスワードを入力します。

いずれも半角英数字にて記載します。

エラーが出ず、登録内容が表示されれば管理者用 WebUI にログインできるようになります。

exit と入力して tinker を終了します。

ここで入力した管理者 ID とパスワードは忘れないようにしてください。

8.WebUI 表示確認

デプロイ作業が完了したら、URL にアクセスします。
WebUI のログイン画面が表示されたら成功です。



ES920LRBG2-K

ログイン

ログインID

パスワード

次回から入力を省略

[パスワードをお忘れの方はこちら](#)

ユーザーログイン画面(ES920LRBG2-K)

管理者用ログイン URL は [http://\(ドメイン\)/admin/login](http://(ドメイン)/admin/login) になります。



ES920LRBG2-K管理者用

管理者ログイン

ログインID

パスワード

次回からパスワード省略

管理者ログイン画面(ES920LRBG2-K)

先程作成した管理者用のログイン ID とパスワードを入力してログインします。

9.Elastic Beanstalk からのイベントを受信する

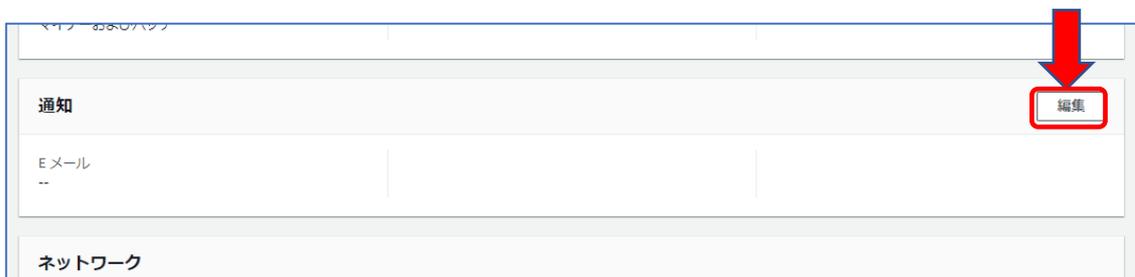
作成した Elastic Beanstalk 環境からのイベント(高負荷によるステータス変化、再起動、アップデート完了等)を AWS SNS(Simple Notification Service)を利用してメール受信することができます。

- ④ Elastic Beanstalk の画面からサイドメニューにある、作成した環境の「設定」をクリックします。



設定画面 (セキュリティ設定)

- ⑤ Elastic Beanstalk の環境設定画面に遷移するので、「通知」の編集ボタンをクリックして「通知の変更」画面に遷移します。



Elastic Beanstalk 通知設定

- ⑥ 「Eメール通知」のEメールに受信するメールアドレスを入力し、画面最下の「適用」を選択します。



Elastic Beanstalk 通知の変更

Elastic Beanstalk が環境の更新作業に入ります。(完了まで数分掛かる場合があります。)

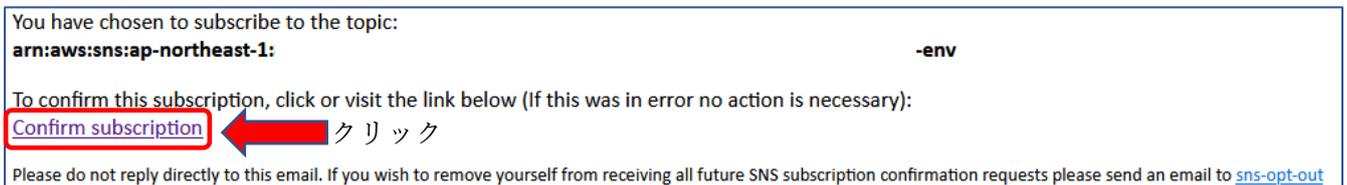
完了すると、「AWS Notification」メールが届いていることが確認できます。

メールが届いていない場合は、メールアドレスが正しく入力されているか確認してください。

★	📧	件名	👤	差出人	🕒	送信日時
☆		AWS Notification - Subscription Confirmation	•	AWS Notifications	🕒	17:50

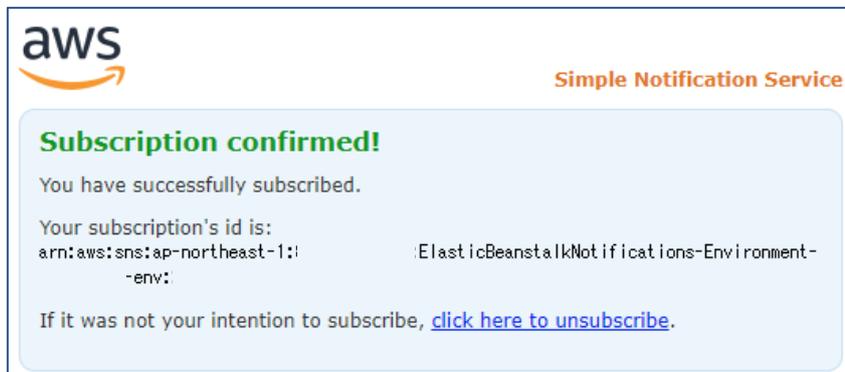
メーラーでのメール受信確認

メール本文にある「Confirm subscription」をクリックします。



メール受信内容

ブラウザ側で通知確認完了の画面が表示されたら、設定完了となります。



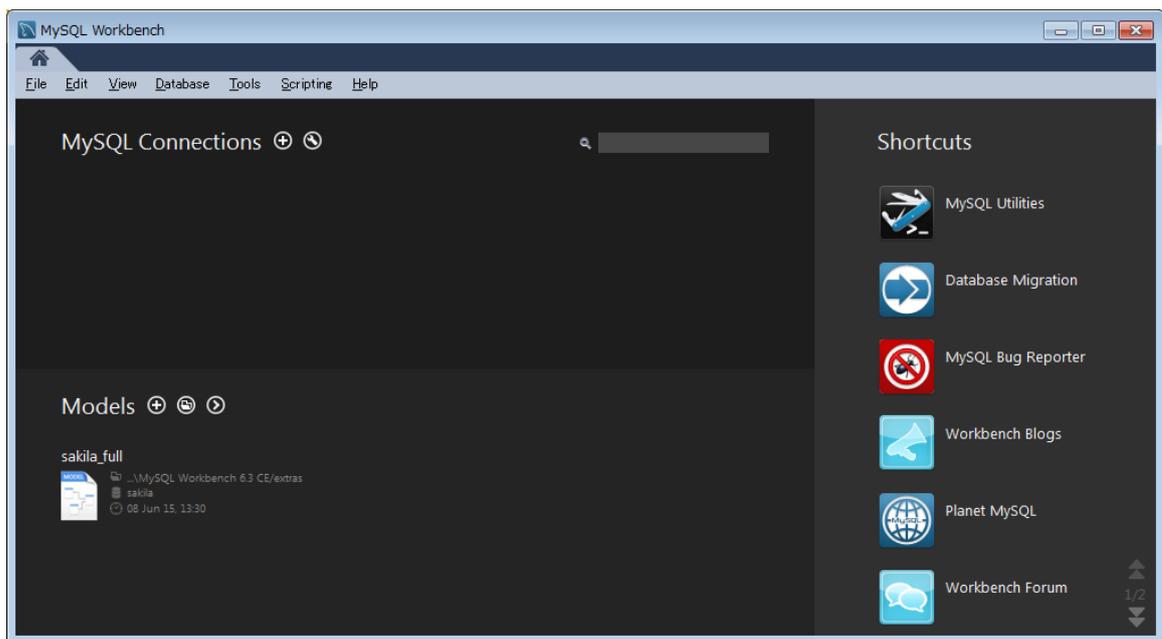
ブラウザ画面 (サブスクリプション確認完了)

10. ローカル環境(PC)から AWS RDS に接続

ローカル環境から AWS RDS のテーブル参照、インポート、エクスポートを行う方法として、MySQL Workbench が使用できます。

MySQL Workbench:(<https://www.mysql.com/jp/products/workbench/>)

※MySQL Workbench をインストールするには、Oracle の Web account を取得する必要があります (無料)。

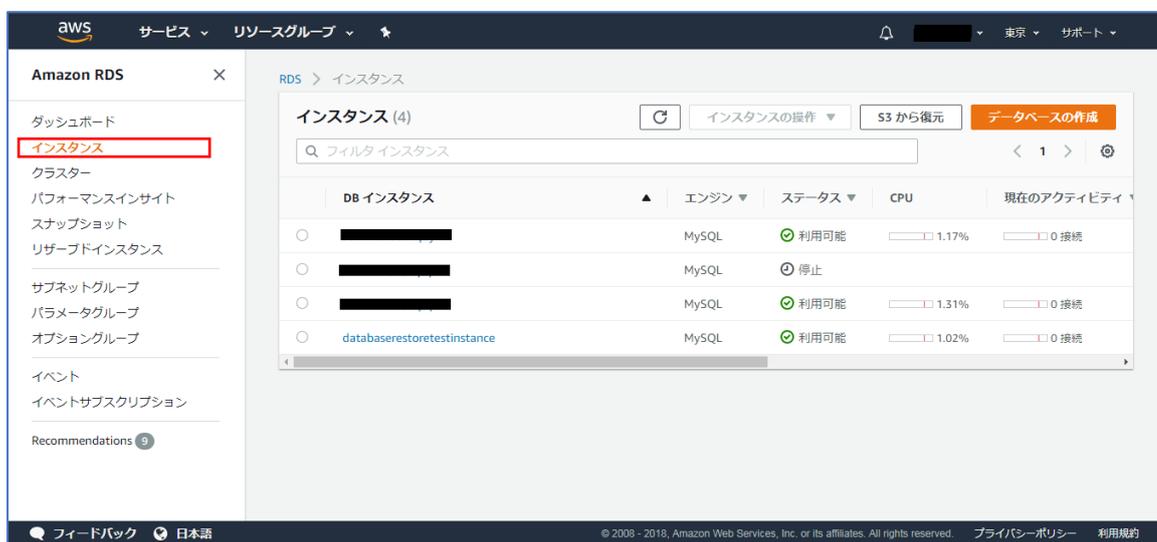


MySQL Workbench

ローカル環境から MySQL Workbench を使用して、AWS RDS MySQL の DB インスタンスにアクセスするためには、事前にローカル環境の IP を AWS RDS のセキュリティグループに登録する必要があります。

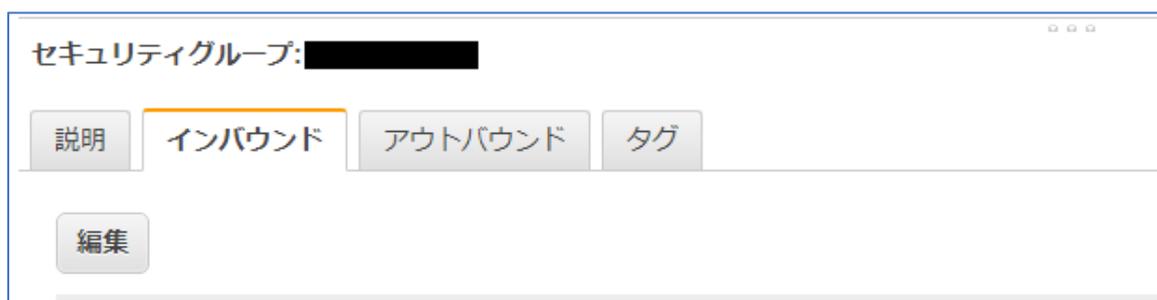
以下に AWS RDS のセキュリティグループの登録から MySQL Workbench を使用して DB インスタンスにアクセスする手順を説明します。

- ① AWS のサービスから RDS を選択し、「インスタンス」から WebUI で使用している DB インスタンスを選択します。



AWS RDS DB インスタンスの選択

- ② DB インスタンス画面の「接続」のセキュリティグループから「セキュリティグループ-Inbound」を選択します。
- ③ ②実行後、EC2 のダッシュボードに遷移するので画面下の「インバウンド」タブを選択し、「編集」ボタンを選択します。



EC2 ダッシュボード セキュリティグループ設定

④ 「インバウンドのルール編集」にて「ルールの追加」ボタンを選択すると、ルールの行が追加されます。

・タイプは「MySQL/Aurora」を選択

・ソースは「マイ IP」を選択することで、横にローカル環境の IP アドレスが表示され、ローカル環境から DB インスタンスへのアクセスを許可することになります。

タイプ	プロトコル	ポート範囲	ソース	説明
MySQL/Aurora	TCP	3306	マイ IP	例: 管理者のデスクトップの SS
MySQL/Aurora	TCP	3306	カスタム	例: 管理者のデスクトップの SS

ルールの追加

注意: 既存のルールを編集すると、編集したルールが削除されて、新しい詳細を含む新しいルールが作成されます。これにより、そのルールに依存するトラフィックは、新しいルールが作成されるまで短時間切断されます。

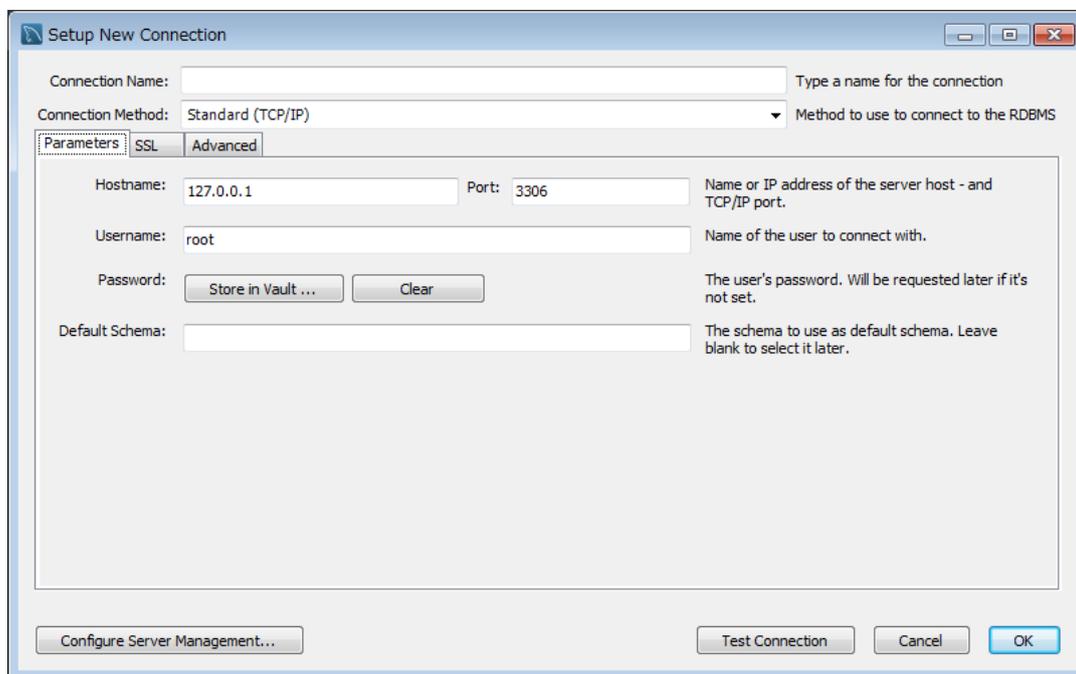
キャンセル 保存

インバウンドのルール編集のルールの追加

※ローカル環境の IP アドレスが DHCP で割り当てられている場合、ローカル環境 PC を再起動等により IP アドレスが前回と変更となった場合は、再度、本設定にて IP アドレスを設定し直す必要があります。

設定が完了したら右下の「保存」ボタンで設定を保存します。

- ⑤ ローカル環境にインストールした MySQL Workbench を起動します。
- ⑥ 「Home」画面の「MySQL Connections」より「+」のボタンを選択し、データベース接続設定 (Setup New Connection)画面を開きます。



MySQL Workbench データベース接続設定

各パラメータを入力していきます。

Connection Name:本設定を識別する名称（テスト接続成功後に入力）

Connection Method: Standard(TCP/IP)

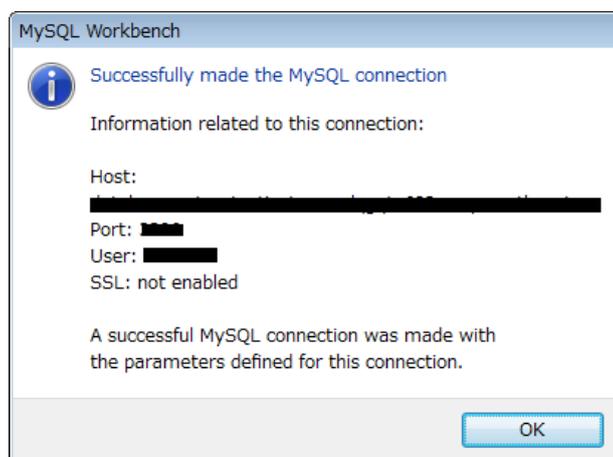
Parameters

Hostname:エンドポイントを入力します。

Port : エンドポイントのポート番号を入力します。

Username : データベース作成時に入力したマスターユーザー名を入力します。

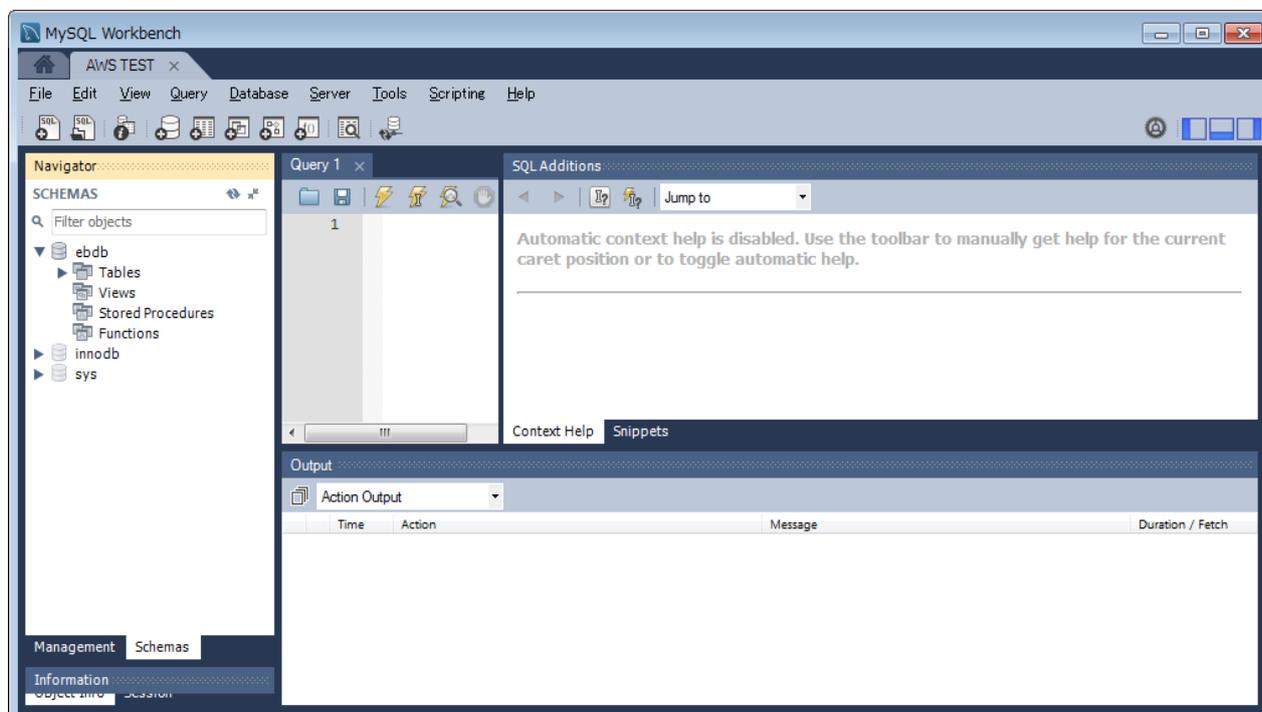
- ⑦ 画面右下の「Test Connection」を選択し、パスワードが聞かれます。マスターパスワードを入力して続行すると、無事接続できたことが確認できます。



MySQL Workbench データベース接続テスト成功

- ⑧ Connection Name に本設定名称(例：AWS RDS アプリ名)を入力して、画面右下の「OK」ボタンを選択すると、設定情報が MySQL Workbench に保存されます。

- ⑨ MySQL Workbench の HOME 画面上に設定した DB 接続情報が表示されているので、クリックしてパスワードを入力すると、AWS の DB インスタンスに接続され、インスタンス内のテーブルが参照できるようになります。



MySQL Workbench データベース接続テスト成功

11. エクスポート / インポート

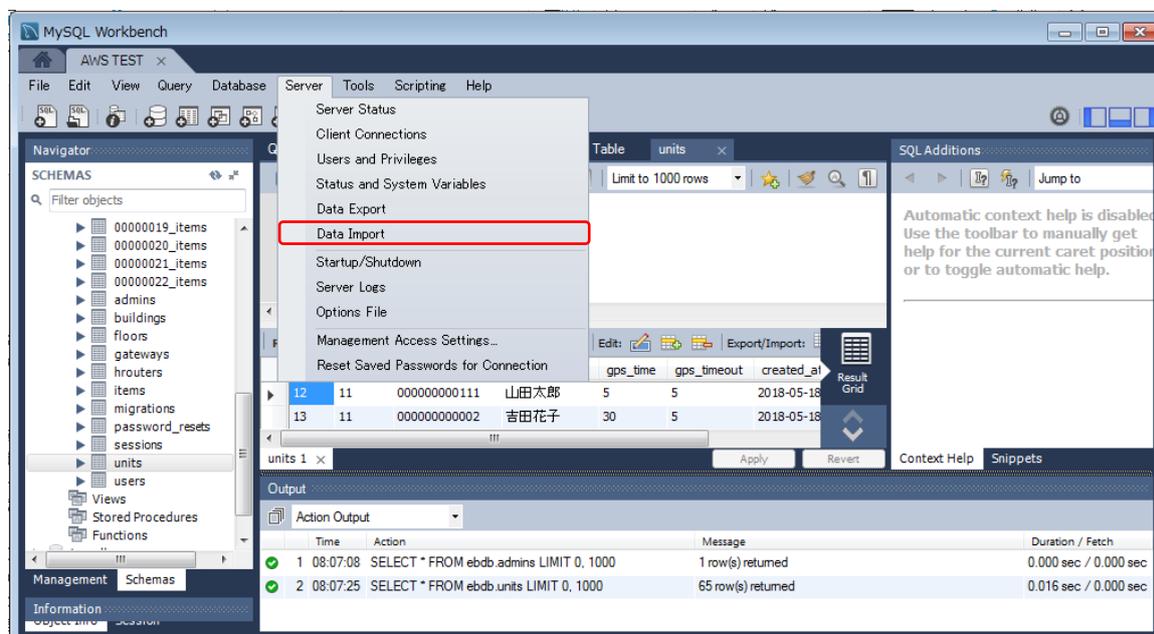
弊社クラウドサーバーからデータをエクスポートし、構築した環境へインポートする場合には以下の手順で行います。

11.1. 弊社サーバーからのエクスポート

- ① 弊社 WebUI にてエクスポートファイル作成要求を実行します。
(エクスポートファイル作成要求は、ES920LRBG2-K、ES920LRTH2-K のどちらの WebUI からも行えます。)
- ② エクスポートファイルをダウンロードします。
ファイル名：es920web_user_8 桁数値.sql
※注意：ファイル名にある 8 桁の数値は弊社データベースのユーザー管理テーブルの id に相当します。既に WebUI 環境を構築し使用されている場合は、ユーザー管理テーブルの id が使用されていない事を確認してください。

11.2. 構築環境へのインポート

- ③ MySQL Workbench のメニュー Server>Data Import から②でダウンロードしたエクスポートファイルを指定して、AWS RDS にデータをインポートします。



MySQL Workbench データインポート

- ④ ブラウザから Web UI の管理画面にログインし、対象データを利用する登録ユーザー情報に移動します。

ユーザー情報 一覧に戻る

基本情報

ID	ログインID	メールアドレス	会社	利用日数	利用アプリ	登録日時	利用開始日時
13	demouser	imaizumi@easel5.com	EASEL	221	ES920LRBG2-K ES920LRTH2-K	2018-05-20 13:38:19	2019-05-20 13:38:48

編集 削除

アクティベート

数値8桁で入力ください。 実行

登録デバイス数

ユニット		ルーター		ゲートウェイ
ES920LRBG	BLEタグ	屋内	屋外	
18	1	6	1	8

デバイス分布マップ

サーバー通知データ使用量

使用量 (MByte)	最大 (MByte)	未使用 (MByte)
0.12500000	64	63.875

データ確認

ユーザー情報画面

- ⑤ ユーザー情報画面の「アクティベート」パネルに、エクスポートファイル名にあった数値8桁を入力し、「実行」ボタンを押下すると、該当ユーザー用に適用され、インポートしたデータが使用できるようになります。

※入力する数値8桁は、既に別な登録ユーザーの ID と同値の場合、実行エラーとなり、インポートしたデータは使用できません。

アクティベート

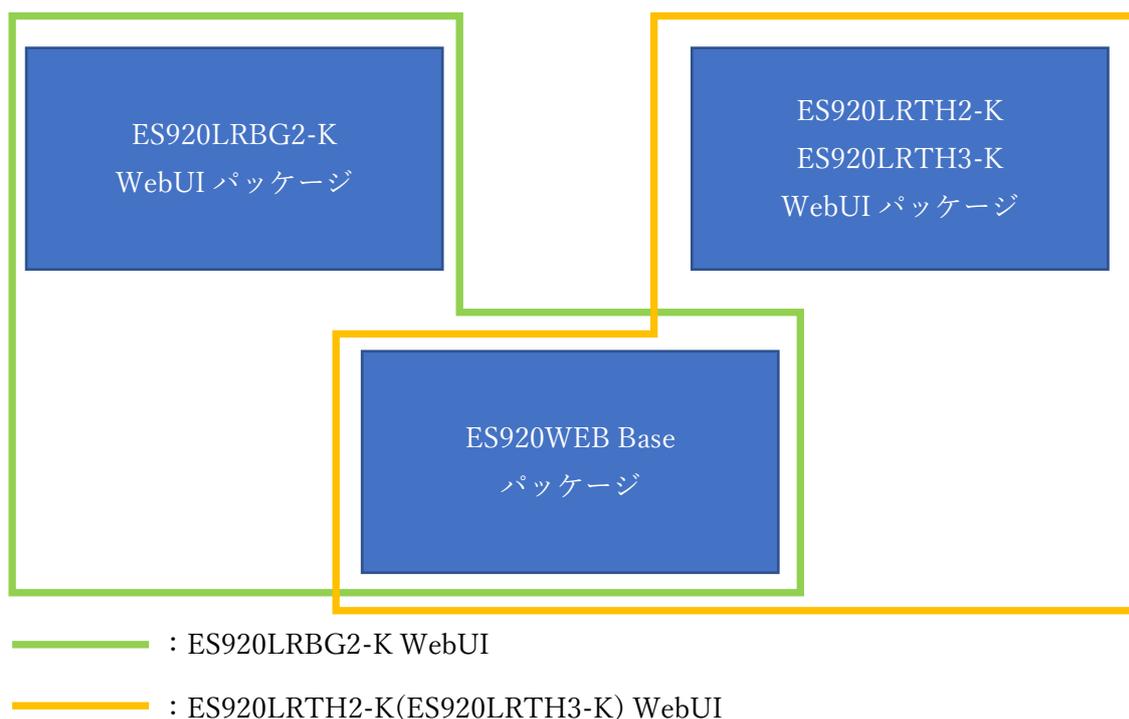
数値8桁で入力ください。 実行

アクティベート設定パネル

12.WebUI の追加

ES920LRBG2-K と ES920LRTH2-K (ES920LRTH3-K) の WebUI は、ベースとなる ES920WEB Base とそれぞれの WebUI パッケージから構成されています。

一方の WebUI 環境を構築済みの場合、もう一方の WebUI パッケージを追加することで、両 WebUI を使用することが可能です。



WebUI パッケージ構成図

12.1. ES920LRBG2-K WebUI に ES920LRTH2-K(ES920LRTH3-K) WebUI パッケージを適用

前提条件として、ES920LRBG2-K をご購入で WebUI 構築済みのお客様を対象とした記載となります。

【パッケージ適用手順】

- ① es920lrthx-k_package.zip を解凍します。
- ② 解凍したフォルダを ES920LRBG2-K WebUI 環境に上書きします。
- ③ フォルダを zip で固めて、Elastic Beanstalk にデプロイします。

12.2. ES920LRTH2-K(ES920LRTH3-K) WebUI に ES920LRBG2-K WebUI パッケージを適用

前提条件として、ES920LRTH2-K または ES920LRTH3-K をご購入で WebUI 構築済みのお客様を対象とした記載となります。

※ES920LRBG2-K WebUI では、地図に Google Map を使用しています。

Google Map 利用には別途 Google と契約し、KEY を取得する必要があります。

【パッケージ適用手順】

- ① es920lrbg2-k_package.zip を解凍します。
- ② 解凍したフォルダを ES920LRTH2-K(ES920LRTH3-K) WebUI の環境に上書きします。
- ③ env ファイルを編集します。
 - ・ Google Map Key 情報追加
- ④ フォルダを zip で固めて、Elastic Beanstalk にデプロイします。

12.3.利用アプリケーションの選択

登録済みユーザーが両 WebUI を利用するためには、WebUI 管理者ユーザーが、登録ユーザー毎に個別に設定する必要があります。※両 WebUI を使用するには、ES920LRBG2-K、ES920LRTH2(TH3)-K の両パッケージが必要となります。

管理者用 WebUI にログインし、追加するユーザーの詳細情報画面にて、「WebUI 追加」パネルにある「追加」ボタンを押下することで、もう一方の未使用の WebUI が追加されます。

ID	ログインID	メールアドレス	会社	登録日時	編集	削除
0000001	demouser	demouser@easel5.com	EASEL	2022-04-07 15:32:03	編集	削除

利用アプリ	利用開始日時	利用日数	残り日数	延長
ES920LRBG2-K	-	0	0	延長
ES920LRTH2(TH3)-K	2022-04-07	84	350	延長

ゲートウェイ	温度感測ユニット
2台	15台

ユーザー情報画面

また、両 WebUI 利用環境構築後の新規ユーザー登録時には、新規登録ユーザーの利用アプリを必ず選択する必要があります。

ユーザー登録

ログインID

会社名

メールアドレス

パスワード

パスワード(確認用)

利用アプリ ES920LRBG2-K(位置検出)
 ES920LRTH2(TH3)-K(温湿度・CO2/温湿度)

使用期限

登録

ユーザー登録画面