

# AWS OpsWorksのご紹介

Amazon Data Services Japan  
2013/06/25



# Agenda

📦 AWS OpsWorksとは

📦 OpsWorksの特長

📦 OpsWorks 利用の流れ

📦 OpsWorksのメリット

📦 Chefとは

📦 OpsWorksのライフサイクルイベント

📦 どのようなアプリケーションがOpsWorksに向いているのか？

📦 OpsWorksの機能詳細

📦 OpsWorksの料金

📦 Link

📦 APPENDIX



# AWS OpsWorksとは

- ❑ 開発者やシステム管理者向けの統合アプリケーション管理ソリューション
- ❑ アプリケーション全体のライフサイクルをモデル化、コントロール、自動化が可能



# OpsWorksの特長

## 柔軟性

- スクリプトでインストールを行う、あらゆるソフトウェアをサポート
- Chefフレームワークを使用
- 独自のChefレシピまたはコミュニティで作成された何百もの設定を活用可能

## 高機能

- 環境全体のテンプレートを定義可能
- テンプレートを保守・バージョン管理可能
- 変更を実行中のインスタンスに適用可能

## 安全

- IAMユーザのSSHアクセスをスタック単位で管理可能

## 使いやすさ

- LB,AP,DB等の各コンポーネントをレイヤーとしてモデル化可能
- リソースおよびその状態を1つの場所で参照可能

## 効率的

- デプロイ、ロールバック、パッチ管理、自動インスタンススケールリング、自動復旧のサポート



# OpsWorks利用の流れ

① スタックの作成

② レイヤーの作成

③ レシピの作成・設定  
(ビルトインレシピ利用)

④ レイヤーに  
インスタンス追加・起動

⑤ レシピによって  
パッケージインストール、  
設定



User



AWS Management  
Console



## Stack

Load Balancerレイヤー

レシピ



App Serverレイヤー

レシピ



Databaseレイヤー

レシピ





# OpsWorksのメリット

- 📦 より早い開始が可能
- 📦 自動化によるコスト削減
- 📦 大規模デプロイの簡易化
- 📦 スクリプト化された構成によりエラーを削減

# Chefとは？

■ サーバの構築手順にある作業を自動化するツール

現状



ChefがCookbook内の  
レシピを元に自動  
構築



あるべき状態



Cookbook

レシピ	レシピ
package "httpd" install	httpd.co nf 編集

- ・ レシピは構築手順書。
- ・ Rubyベースのためプログラミング可能
- ・ 何度でもレシピを実行しても同じ結果になることが重要。あるべき状態に収束

→ 「べき等性」

# Chefレシピファイルの例

## 例 : default.rbファイル (rubyベース)

```
# haproxy パッケージインストール
case node[:platform]
  when 'ubuntu'
    package 'haproxy' do
      action :install
    end
  end
end
```



# Chefの種類

## 📦 Chef-Solo

- Chefのスタンドアロン版。オープンソースで公開
- **OpsWorksはChef-Soloを採用**

## 📦 Chefクライアント／Chefサーバ

- クライアントがサーバに接続、サーバ側で構成管理

## 📦 Chefホスティングサービス

- Opscode社が提供。ノード数に応じて課金。サポート有

## 📦 プライベートなChef

- Chefホスティングをプライベートネットワーク内に構築



# OpsWorksライフサイクルイベント

各イベントごとに登録されているレシピが実行される

イベント	いつ実行されるか？	Chefレシピの例
Setup	インスタンス初期化時、または正常に起動した時	Apache, Ruby, Passenger, Ruby on Railsのインストールするレシピ
Configure	スタックの状態が変化するたびに実行。 例: 新規インスタンスがアプリケーションサーバーレイヤーに正常に追加された時	ロードバランサーレイヤー設定を更新して、インスタンス追加を反映するレシピ
Deploy	アプリケーションがデプロイされるたびに実行	アプリケーションのチェックアウトやダウンロードを実行し、アプリケーションを再ロードするレシピ
Undeploy	アプリケーションが削除された時	データベーステーブルの削除などのクリーンアップを実行するレシピ
Shutdown	インスタンスが実際に停止する45秒前	サービスをシャットダウンするレシピ



# レイヤーを使ったアプリケーションのモデル化

- インスタンス起動時に設定
  - ボリューム、EIPs, セキュリティグループ
  - ソフトウェア、サービス、Chefレシピ

■ 好きなソフトウェアを利用可能



Mount Point	RAID Level	# Disks	Total Size (G)
	None	1	

Packages	Security Groups	Auto He
		Toggle



# 自動化タスク

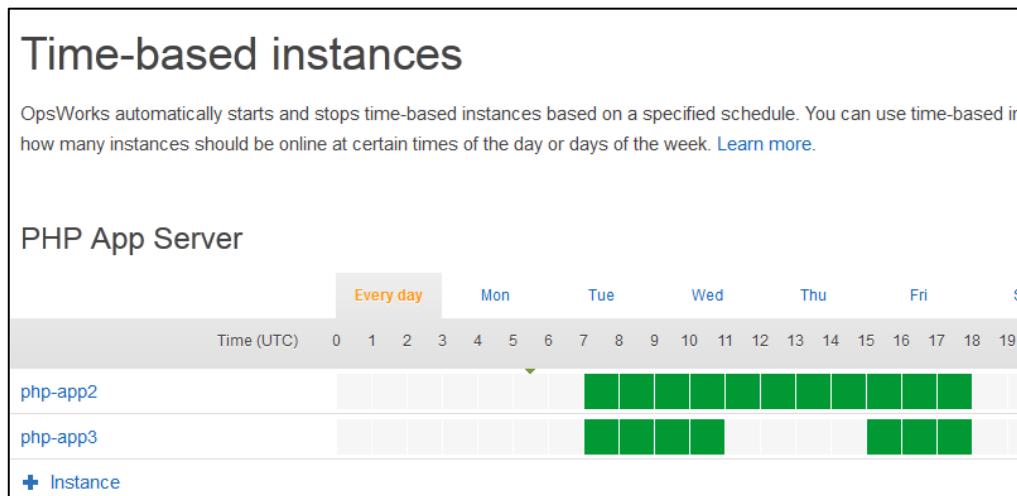
📦 インストール、デプロイ、構成

📦 EBS RAID

📦 Auto Scaling

- Time-based
- Load-based

📦 自動復旧





# どのようなアプリケーションが OpsWorksに向いているか？

## 📦 良い例

- ソーシャル・ブラウザ向けゲーム
- SaaSプロバイダー
- Eコマース
- ロケーションベースサービス
- 基本的なWebやモバイルアプリ

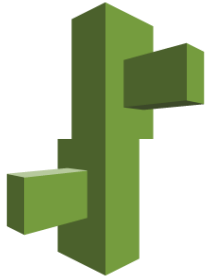
## 📦 悪い例（現時点の）

- 商用エンタープライズアプリケーション
  - スクリプトでパッケージインストールできない場合
- バッチ処理
  - カスタムAMI利用等
- ソースコードリポジトリと本番環境が通信できない場合



# AWSのアプリケーション管理ソリューション

使い易さ



Elastic Beanstalk



OpsWorks



CloudFormation



EC2



フレキシブル



# OpsWorksの機能詳細（1）

リソース設定	多くのアプリケーションに必要なリソースをサポート 例: EC2インスタンス、EBSボリューム、EIPs
ソフトウェア管理	インスタンス上で必要に応じてソフトウェアを構成。Chefのレシピを使ってライフサイクル管理が可能。Amazon Linux, Ubuntuをサポート
アプリケーションデプロイ	Git, SVN, アーカイブチェックアウト、部分デプロイ、ロールバック、ツール導入のフックをサポート。複数のアプリケーションをTierごとにデプロイ可能
監視	CloudWatchまたはビルトインのGangliaを利用可能



# OpsWorksの機能詳細(2)

ロードバランサー	ビルトインのHAProxy、ELB、またはChefレシピを使ったカスタムレイヤーでロードバランサーを作成可能
データベース	Chefレシピを使って、既存のRDSやDynamoDBへアプリケーションから接続を構成可能。ビルトインのMySQLまたはChefレシピを使ってカスタムレイヤーでデータベースを作成可能
Auto Scaling	時間もしくは負荷に応じたAuto Scalingが可能
自動復旧	インスタンスに障害が発生した場合、OpsWorksは新規インスタンスと交換することが可能
アクセスコントロール	リソースへのアクセスを管理可能





# OpsWorksの料金体系

- ❑ OpsWorks自体は無料
- ❑ OpsWorksにて作成されたAWSリソース（EC2インスタンス、EBSボリューム、Elastic IPアドレスなど）
- ❑ OpsWorksはログを作成し、お客様のアカウントのAmazon S3バケットに保存するため、その課金が発生



# Link

## AWS OpsWorks User Guide

- <http://docs.aws.amazon.com/opsworks/latest/userguide/welcome.html>

**ご清聴ありがとうございました！**



# APPENDIX



# Chef-Soloによるhttpdインストール(1)

## 1. Chef-Soloインストール

```
$ curl -L http://www.opscode.com/chef/install.sh | sudo bash
```

## 2. Cookbook作成

```
$ knife cookbook create apache -o cookbooks
```

## 3. Chef-Solo設定ファイル2つ作成

```
$ cat > solo.rb  
file_cache_path "/tmp/chef-solo"  
cookbook_path ["/home/ec2-user/cookbooks"]
```

```
$ cat > localhost.json  
{  
  "run_list" : [  
    "recipe[apache]"  
  ]  
}
```

# Chef-Soloによるhttpdインストール(2)

## 📦 4. レシピ作成・編集

```
$ cat > ./cookbooks/apache/recipes/default.rb  
package "httpd" do  
  action :install  
end
```

## 📦 5. Chef-Soloを実行してパッケージインストール

```
$ sudo chef-solo -c solo.rb -j ./localhost.json
```