

株式会社エクストランス

# X-MON3

X-MON Chef 連携設定リファレンス

2016/7/27 版

## まえがき

本書は X-MON3 系列のエスカレーション機能を使い、Chef を利用した Web サイトのメンテナンス画面への切り替えの方法を解説します。

そのため、基本的な LinuxOS の一般的な操作、用語などについては知識をご理解の上でお読みください。

また、本稼働中のシステムへのインストール作業などは十分に検証を行ったうえで導入するようにしてください。

いかなるシステムへの影響が発生しても、弊社は責任を負いかねますのでご了承ください。

本書執筆用に使用した環境は以下となります。

OS : CentOS 6.5 64bit 版

Web サーバ : apache 2.2.15 ( mod\_rewrite 有効 )

また、本書で実施する手順は、全て Chef-solo を使った構成となります。

本書以外のマニュアルについては X-MON サポートページにログインしてご確認ください。

<https://x-mon.jp/support/>

2014 年 07 月

改定履歴
2014 年 07 月 初版

The Chef™ Mark and Chef Logo are either registered trademarks/service marks or trademarks/ service marks of Chef, in the United States and other countries and are used with Chef Inc's permission.

We are not affiliated with, endorsed or sponsored by Chef Inc.

Copyright © 2004-2016 X-TRANS, Inc. All Rights Reserved.

## 目次

---

1	Chef のインストール .....	3
2	ユーザ環境のセットアップ .....	4
3	Chef-solo 環境の設定 .....	5
4	Cookbook の作成 .....	6
4.1	メンテナンス画面を表示する仕組み .....	6
4.2	.htaccess の作成 .....	6
4.3	レシピの作成 .....	6
4.3.1	障害発生時のレシピ .....	7
4.3.2	障害復旧時のレシピ .....	7
4.3.3	レシピのテスト実行 .....	8
5	X-MON でのエスカレーション設定 .....	10
5.1	エスカレーションコマンドの設定 .....	10
5.2	エスカレーションの設定 .....	12

## 1 Chef のインストール

---

Chef のインストールは簡単です。Opscode が提供している Chef 配布用スクリプトを実行します。

インストール作業は root ユーザにて実行します。

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

## 2 ユーザ環境のセットアップ

Chef を実行するユーザを作成します。本書では「chef-ope」というユーザを作成し、使用する想定となっています。別のユーザアカウントを利用したい場合は、該当箇所を置き換えてください。

```
# useradd chef-ope
# passwd chef-ope
# usermod -G wheel chef-ope
```

また、Chef では root 権限の操作が必要になります。visudo にて以下の項目の変更を行ってください。

**56 行目の requiretty をコメントアウトし、リモートで sudo を使えるようにする**

```
#Defaults    requiretty
```

**108 行目のコメントを外し、wheel グループのユーザは sudo をパスワード無しで使用できるようにする**

```
%wheel ALL=(ALL)    NOPASSWD: ALL
```

また、X-MON からエスカレーションを実行するために、X-MON サーバの公開鍵を chef-ope ユーザの authorized\_keys に登録が必要です。設定の手順はオンラインヘルプをご確認ください。

### 3 Chef-solo 環境の設定

Chef-solo の環境を作る際に、knife solo のコマンドを使用します。次のコマンドを入力し、gem 経由でインストールしてください。

```
# /opt/chef/embedded/bin/gem install knife-solo
```

ここからの作業は、chef-ope ユーザで実施します。まず、knife コマンドのセットアップを行います。

```
# sudo su - chef-ope
$ cd ~
$ knife configure
```

プロンプトで入力を求められますが、全て enter キーを押してください。  
続いて、Chef のリポジトリを作成します。次のコマンドを入力してください。

```
$ knife solo init chef-repo
```

これで chef-ope アカウントのホームディレクトリに、Chef のリポジトリが作成されます。

最後の Chef-solo の設定ファイルを作成します。

```
$ vi /home/chef-ope/chef-repo/solo.rb
```

以下のように設定を行います。

```
file_cache_path "/tmp/chef-repo"
cookbook_path ["/home/chef-ope/chef-repo/cookbooks"]
```

これで Chef-solo を使うための事前準備が完了しました。

## 4 Cookbook の作成

Web サイトのメンテナンス画面への切り替えを行うための Cookbook のひな形を最初に作成します。次のコマンドを実行してください。

```
$ knife cookbook create maintenance -o /home/chef-ope/chef-repo/cookbooks/
```

これで「maintenance」という Cookbook のひな形が作成されました。

### 4.1 メンテナンス画面を表示する仕組み

Web サイトでメンテナンス表示に切り替えを行う方法はいくつか存在します。

ここでは、できるだけ既存のコンテンツに影響を与えないように、`mod_rewrite` を使って、メンテナンス中は全てのリクエストを、`maintenance.html` に遷移させる方法を使います。

`maintenance.html` は別途、作成を行い予めサーバ上に設置しておいてください。

### 4.2 .htaccess の作成

まずは、Cookbook の中に障害時に設置する `.htaccess` ファイルを作成します。

```
$ vi /home/chef-ope/chef-  
repo/cookbooks/maintenance/templates/default/htaccess.erb
```

ファイルの中身は以下のように記載してください。

```
ErrorDocument 503 /maintenance.html  
  
<IfModule mod_rewrite.c>  
  RewriteEngine On  
  RewriteCond %{REQUEST_URI} !=/maintenance.html  
  RewriteRule ^.*$ - [R=503,L]  
</IfModule>
```

### 4.3 レシピの作成

Chef はレシピと呼ばれるファイルに記載した手順通りの処理を行います。ここでは、障害発生時のレシピと、障害復旧時のレシピの作成を行います。

#### 4.3.1 障害発生時のレシピ

障害発生時には先ほど作成しておいた `.htaccess` を Web サーバのドキュメントルートに設置します。ドキュメントルートは「`/var/www/html`」としています。別の場所を設定している場合は、該当箇所を置き換えてください。

```
$ vi /home/chef-ope/chef-repo/cookbooks/maintenance/recipes/default.rb
```

ファイルの中身は以下のように記載してください。

```
template "/var/www/html/.htaccess" do
  source "htaccess.erb"
  mode 0644
  owner "root"
  group "root"
  action :create
end
```

続いて実行時の設定用 JSON ファイルを作成します。

```
$ vi /home/chef-ope/chef-repo/problem.json
```

ファイルの中身は以下のように記載してください。

```
{
  "run_list": ["maintenance::default"]
}
```

#### 4.3.2 障害復旧時のレシピ

障害復旧時には設置されている `.htaccess` を Web サーバのドキュメントルートから削除します。ドキュメントルートは「`/var/www/html`」としています。別の場所を設定している場合は、該当箇所を置き換えてください。

```
$ vi /home/chef-ope/chef-repo/cookbooks/maintenance/recipes/recovery.rb
```

ファイルの中身は以下のように記載してください。

```
file "/var/www/html/.htaccess" do
  action :delete
end
```

続いて実行時の設定用 JSON ファイルを作成します。

```
$ vi /home/chef-ope/chef-repo/recovery.json
```

ファイルの中身は以下のように記載してください。

```
{
  "run_list": ["maintenance::recovery"]
}
```

#### 4.3.3 レシピのテスト実行

まず、次のコマンドを実行してください。

```
$ sudo /usr/bin/chef-solo -c /home/chef-ope/chef-repo/solo.rb -j /home/chef-ope/chef-repo/problem.json
```

実行完了後は、次のコマンドを実行し、`.htaccess` が作成されていることを確認してみましよう。

```
$ ls -la /var/www/html
```

`.htaccess` が無事に作成されているようであれば、Web サーバにブラウザから接続し、メンテナンス画面が表示される事を確認して下さい。

メンテナンス画面が表示された場合は、次のコマンドを実行し、正常表示に戻ることを確認して下さい。

```
$ sudo /usr/bin/chef-solo -c /home/chef-ope/chef-repo/solo.rb -j /home/chef-ope/chef-repo/recovery.json
```

その後、同じように `.htaccess` が削除されていることを確認してみましよう。

```
$ ls -la /var/www/html
```

先ほど、存在していた `.htaccess` が削除されているようでしたら、改めて Web サーバにブラウザから接続し、リダイレクト処理が行われない事を確認して下さい。

問題がないようであれば、事前準備は完了です。

## 5 X-MON でのエスカレーション設定

X-MON から Chef を実行するには「通知・エスカレーション関連設定一覧」から、エスカレーションコマンドの設定を行います。

### 5.1 エスカレーションコマンドの設定

**通知・エスカレーション関連設定一覧**

<p><b>メール送信設定</b></p> <p>X-MONからメールを送信する際のメールサーバ設定を行います。</p>
<p><b>通知メール文章設定</b></p> <p>通知メールの文章を任意の設定に変更します。</p>
<p><b>警告灯設定</b></p> <p>警告灯デバイスの登録や、点灯/点滅の設定を行います。</p>
<p><b>Twilio連携設定</b></p> <p>Twilioアカウントの管理やX-MONからの電話通知設定を行います。</p>
<p><b>エスカレーションコマンド一覧</b></p> <p>エスカレーションコマンドの一覧表示、登録を行います。</p>
<p><b>エスカレーションコマンド設定</b></p> <p>エスカレーションコマンドの設定を行います。</p>

Chef を実行する際には、「linux\_remote\_command」を使用します。エスカレーション設定の新規登録から、以下の内容で登録を行なって下さい。

コマンド名称	Chef 実行-障害発生
実行コマンド	linux_remote_command
引数	<pre>--ipaddress \$HOSTADDRESS\$ --username chef-ope -- prog '/usr/bin/chef-solo -c /home/chef-ope/chef-repo/solo.rb -j /home/chef-ope/chef-repo/problem.json' --state \$SERVICESTATE\$ --notification \$NOTIFICATIONTYPE\$</pre>

コマンド名称	Chef 実行-障害復旧
実行コマンド	linux_remote_command
引数	<pre>--ipaddress \$HOSTADDRESS\$ --username chef-ope -- prog '/usr/bin/chef-solo -c /home/chef-ope/chef-repo/solo.rb -j /home/chef-ope/chef-repo/recovery.json' --state \$SERVICESTATE\$ --notification \$NOTIFICATIONTYPE\$</pre>

--prog で指定するのは、先ほどテスト実行で実行したコマンドになります。  
linux\_remote\_command では、自動的に sudo を付与しますので、sudo の記述は不要  
です。

chef-solo に対して指定している引数は、以下の内容となります。

引数	概要
-c	chef-solo コマンドが参照する設定ファイルを指定します。
-j	実行するレシピや設定が記述された JSON ファイルを指定しま す。

X-MON のエスカレーションで Chef を実行する場合は、対象とする JSON ファイル  
ごとに、エスカレーションコマンド設定を追加する必要があります。

## 5.2 エスカレーションの設定

障害時にメンテナンス画面に切り替えを行いたいサービスのエスカレーション設定を行います。以下の画像のように設定を行なって下さい。

The image displays two screenshots of the Chef escalation settings interface. Each screenshot includes a 'NEW!' badge, status toggles (有効/無効), and a '削除' (Delete) button.

**Top Screenshot (Chef実行-障害発生):**

- 実行開始回数: 1 回目から実行する
- 実行回数:  制限する (1 回実行する),  制限しない
- 実行内容:  メール通知 (大阪ユーザグループ),  コマンド (Chef実行-障害発生)
- 対象ステータス:  WARNING,  CRITICAL,  UNKNOWN,  復旧時

**Bottom Screenshot (Chef実行-障害復旧):**

- 実行開始回数: 1 回目から実行する
- 実行回数:  制限する (1 回実行する),  制限しない
- 実行内容:  メール通知 (大阪ユーザグループ),  コマンド (Chef実行-障害復旧)
- 対象ステータス:  WARNING,  CRITICAL,  UNKNOWN,  復旧時

エスカレーションの実行タイミングで、実行回数は 1 回にしておいてください。エスカレーションが実行される度に、.htaccess ファイルが再作成される事になります。