

VSP setup

Ubuntu

<http://articles.slicehost.com/2010/4/30/ubuntu-lucid-setup-part-1>

<http://articles.slicehost.com/2010/4/30/ubuntu-lucid-setup-part-2>

Ruby

<http://articles.slicehost.com/2009/1/6/ubuntu-intrepid-ruby-on-rails>

RVM

<http://d.hatena.ne.jp/hrsth/20100609/1276047000>

MySQL

<http://articles.slicehost.com/2008/7/8/ubuntu-hardy-installing-mysql-with-rails-and-php-options>

gem

passenger

http://redmine.jp/tech_note/apache-passenger/

nokogiri

```
sudo apt-get install libxml2-dev libxslt1-dev
```

```
sudo apt-get install ruby1.8-dev
```

```
sudo gem install nokogiri
```

EC2 backup

- ・ Keypairをインスタンスのディレクトリ「/mnt」にコピーする
WinSCPを使って、現在使用しているインスタンスのディレクトリ「/mnt」にPrivate Key File (pk-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.pem) とX.509 certificate file (cert-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.pem) をコピーする。

- ・ SSHから「/mnt」にデータのバックアップを行う
ec2-bundle-vol -d /mnt -k /mnt/" Private Key File" -c /mnt/" X.509 certificate file" -u "AWS account ID"
" Private Key File" と" X.509 certificate file" は先ほど、「/mnt」にコピーした物。"AWS account ID"はAWS登録時に自動的に割り振られたID(AWS Account Activityページ上部のAccount Number)。

- ・ SSHからAMIが作成されていることを確認する
ls -l /mnt/image.*

- ・ SSHから作成したAMIをS3にアップロード
ec2-upload-bundle -b "your-s3-bucket" -m /mnt/image.manifest.xml -a "aws-access-key-id" -s "aws-secret-access-key"
"your-s3-bucket"には、「S3 Organizer」で作成したディレクトリを入れる。

4.作成したAMIの登録

- ・ コマンドプロンプトからS3にアップロードしたAMIを登録する。成功すると、AMIのイメージ名が表示される。

```
C:\>ec2-register "your-s3-bucket"/image.manifest.xml  
IMAGE ami-XXXXXXXX
```

- ・ コマンドプロンプトからAMI登録の確認をする。登録時に表示されたイメージ名があればOK。

```
C:\>ec2-describe-images -o self -o amazon
```

5.作成したAMIを起動してみる

AMIの起動方法は、2つあり「Elasticfox」とコマンドプロンプトを使用する方法です。

Bundler 1.1.0 で YAML がパースエラーを起こす。

<http://makandra.com/notes/1105-fix-couldn-t-parse-yaml-error-after-upgrading-bundler-config/boot.rb> に以下を追加。

```
require 'yaml'  
YAML::ENGINE.yamler= 'syck'
```

Flex 4.5 の spark.components.Image のプログレスバー

Flex 4.5 で spark 版の Image コンポーネントが追加されてキャッシュ機構とかスムーズリサイズ、レターボックス化なんかの機能が使えるようになった。

<http://opensource.adobe.com/wiki/display/flexsdk/Spark+Image>

で、ロード中にプログレスバーを表示できるようにもなったんだけど、これがうまく動かない。

SkinParts で指定されている id を割り当てられているにも関わらず、スキン側のプログレスバーが無視されている感じ。

で spark 内のソースを追ってみたら、

```
override protected function getCurrentSkinState():String
{
    var enableLoadingState:Boolean = getStyle("enableLoadingState");

    if (_invalid)
        return "invalid";
    else if (!enabled)
        return "disabled";
    else if (_loading && enableLoadingState) // ← ここ
        return "loading";
    else if (imageDisplay && imageDisplay.source && _ready)
        return "ready";
    else
        return "uninitialized";
}
```

なんて箇所を発見。つまり、enableLoadingState っていうスタイルがないとローディング中の State が認識されない。(プロパティじゃないところが軽いトラップ。)

というわけで、

```
s|Image {
    enableLoadingState: true;
}
```

とか

```
<s:Image enableLoadingState="true" />
```

って指定してやれば無事にプログレスバーが表示されるよ、というお話。

(上で挙げたアドビのページだと <s:Image enablePreload="true" /> ってあるけどこれはたぶんウソ、もしくはベータ時代のお話。そんなものはなかった。)

Devise で Omniauthable 利用時に Rememberable が効かない

omniauth_callbacks_controller に Devise::Controller::Rememberable を include

```
class Users::OmniauthCallbacksController < Devise::OmniauthCallbacksController
  include Devise::Controllers::Rememberable # 追加
```

```
  def twitter
    user = User.find ...
```

```
    ...
    if user.persisted?
```

```
      ...
      remember_me( user ) # 追加
    end
```

```
  end
end
```

あと database_authenticatable を使っていない時は config/initializers/devise.rb で

```
config.use_salt_as_remember_token = false
```

としておく必要あり。(true だとマイグレーションで t.rememberable でも remember_token カラムを作ってくれない)