

# クラウド入門 演習参考資料

# 1章 CloudClient

## 1.1 CloudClient について

## 1.2 CloudClient を使ってみよう

- 1.2.1. CloudClient の設定をする
- 1.2.2. インスタンスを起動する
- 1.2.3. putty の秘密鍵を作成する
- 1.2.4. Apache をインストールする
- 1.2.5. Apache の設定をする
- 1.2.6. 仮想マシンイメージを作成する
- 1.2.7. 動作確認をする
- 1.2.8. インスタンスを終了する

# 2章 WebAPI

## 2.1 Typica について

## 2.2 Typica を使ってみよう

- 2.2.1. eclipse プロジェクトをインポートする
- 2.2.2. 登録されている仮想マシンイメージの一覧を取得する
- 2.2.3. 演習のヒント
  - 2.2.3.1. インスタンスを起動する
  - 2.2.3.2. 起動しているインスタンスの一覧を取得する
  - 2.2.3.3. インスタンスを終了する

# 3章【参考】教室外から接続するには

## 3.1 edubasecloud に接続するには

- 3.1.1. PacketiX VPN のインストール
- 3.1.2. Putty のインストール
- 3.1.3. CloudClient のインストール

## 3.2 実行環境の構築

- 3.2.1. java-jdk のインストール
- 3.2.2. eclipse のインストール
- 3.2.3. winscp のインストール

## 図の索引

図 1:設定の変更.....	5
図 2:クラウドクライアントの設定 .....	6
図 3:認証情報取得.....	7
図 4:Regions の設定 .....	8
図 5:新規キーペアの作成.....	9
図 6:新規キーペアの保存.....	9
図 7:キーペアの管理.....	10
図 8:pem ファイルの保存場所.....	11
図 9:起動する仮想マシンイメージの選択 .....	12
図 10:仮想マシンの起動 .....	12
図 11:仮想マシン起動ダイアログ .....	13
図 12:仮想マシンの起動確認 .....	13
図 13:シェルの起動 .....	14
図 14:鍵変換.....	14
図 15:外部鍵のインポート .....	15
図 16:鍵の生成.....	15
図 17:警告.....	15
図 18:秘密鍵の保存.....	16
図 19:秘密鍵の保存場所.....	16
図 20:シェルの起動 .....	17
図 21:セキュリティ警告 .....	17
図 22:イメージの追加 .....	18
図 23:イメージ追加ダイアログ .....	19
図 25:作成したイメージの確認.....	19
図 24:イメージ作成完了 .....	19
図 26:Web ブラウザからの確認 .....	20
図 27:仮想マシンの停止 .....	21
図 28:eclipse プロジェクトのインポート .....	22
図 29:選択.....	23
図 30:プロジェクトのインポート .....	23
図 31:フォルダの参照.....	24
図 32:プロジェクトのインポート完了 .....	24
図 33:インポートしたプロジェクトの確認 .....	25
図 34:設定の変更.....	26
図 35:クラウドクライアントの設定確認 .....	27
図 36:アプリケーションの実行 .....	28
図 37:イメージのリスと取得.....	29

# 1章 CloudClient

## 1.1 CloudClient について

edubase Cloud を操作する GUI ツールです。

詳しくは、「e-cloud\_quick\_start.ppt」や「manual\_student.pdf」を参照してください。

## 1.2 CloudClient を使ってみよう

Apache をインストールした仮想マシンイメージを作成し、マシンが立ち上がったときに、Apache が起動するように設定をしてみましょう。

用意するもの

- 。 接続情報の資料

### 1.2.1. CloudClient の設定をする

まず最初に CloudClient を使用するために設定を行います。

Windows の「スタート→すべてのプログラム→クラウドクライアント→CloudClient」を起動します。

次に「ファイル→設定」をクリックし、クラウドクライアントの設定を行います。(図 1:設定の変更)

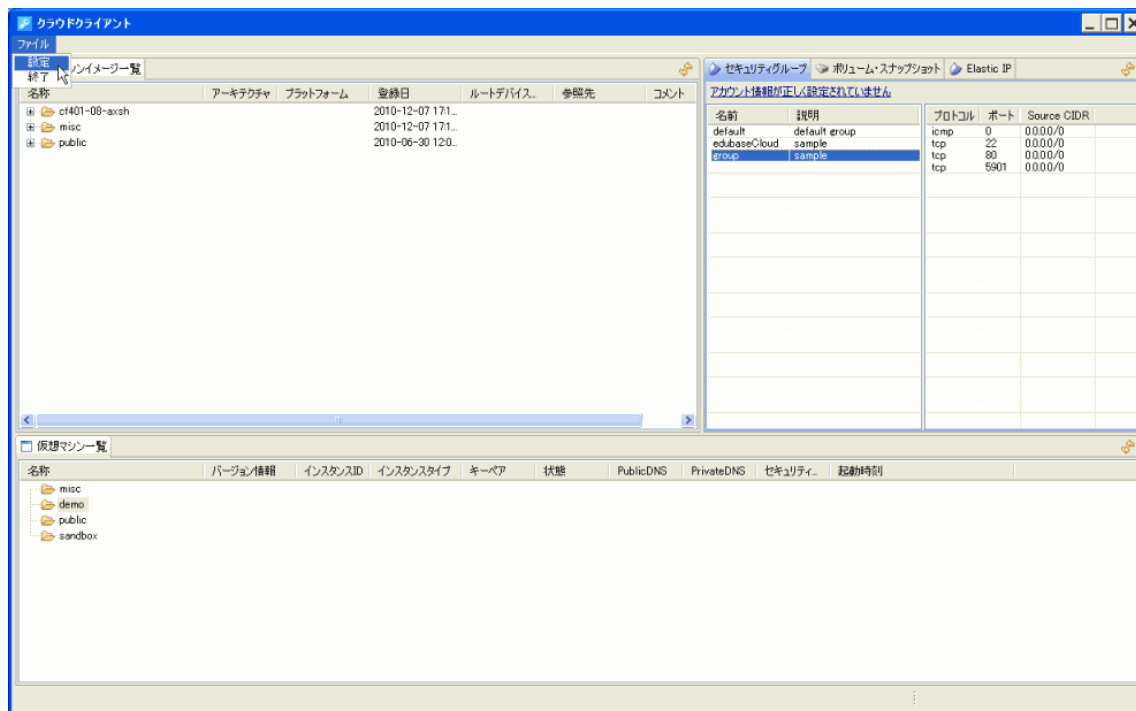


図 1: 設定の変更

左側の「クラウドクライアント」をクリックして、クラウドクライアントの項目を表示します。

接続情報の資料を元に以下の項目を入力し「プロジェクトチーム名取得」ボタンをクリックします。

(図 2:クラウドクライアントの設定)

- 接続 URL
- ログイン ID
- パスワード
- プロジェクトチーム名

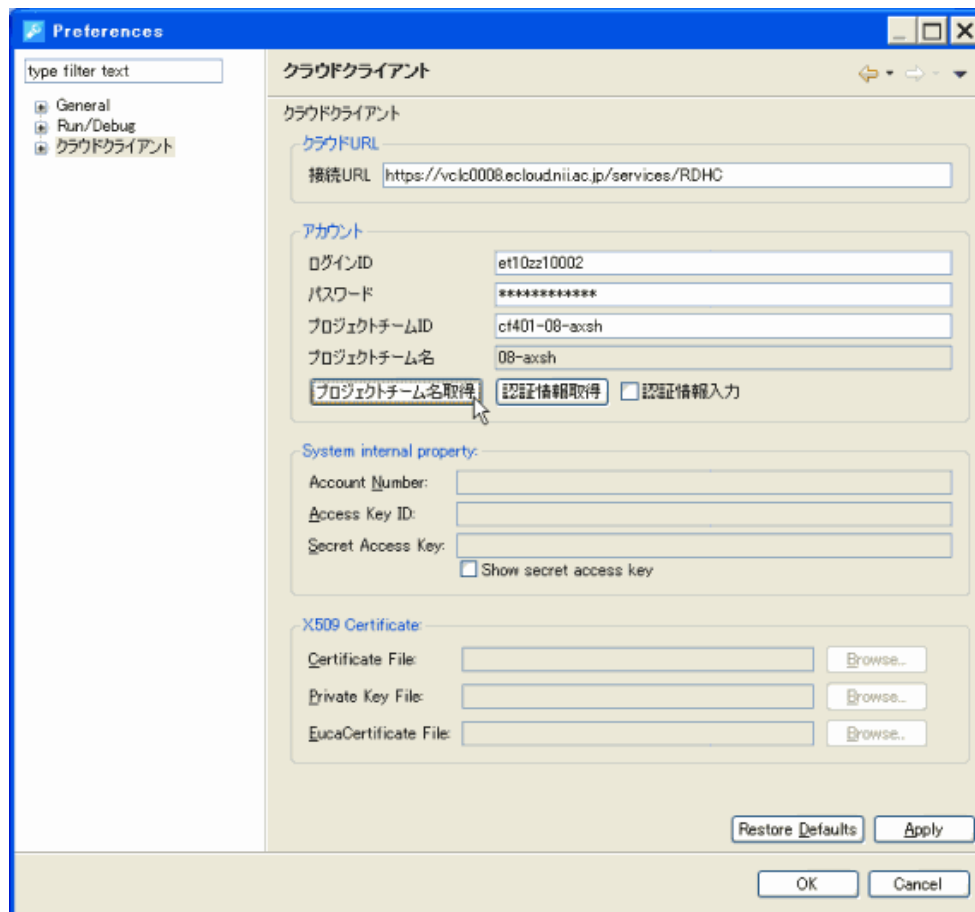


図2:クラウドクライアントの設定

次に「認証情報取得」ボタンをクリックし、System internal property と X509 Certificate の内容が取得されることを確認し「Apply」ボタンをクリックします。(図 3:認証情報取得)  
表示に変化はありませんので、次に進んでください。

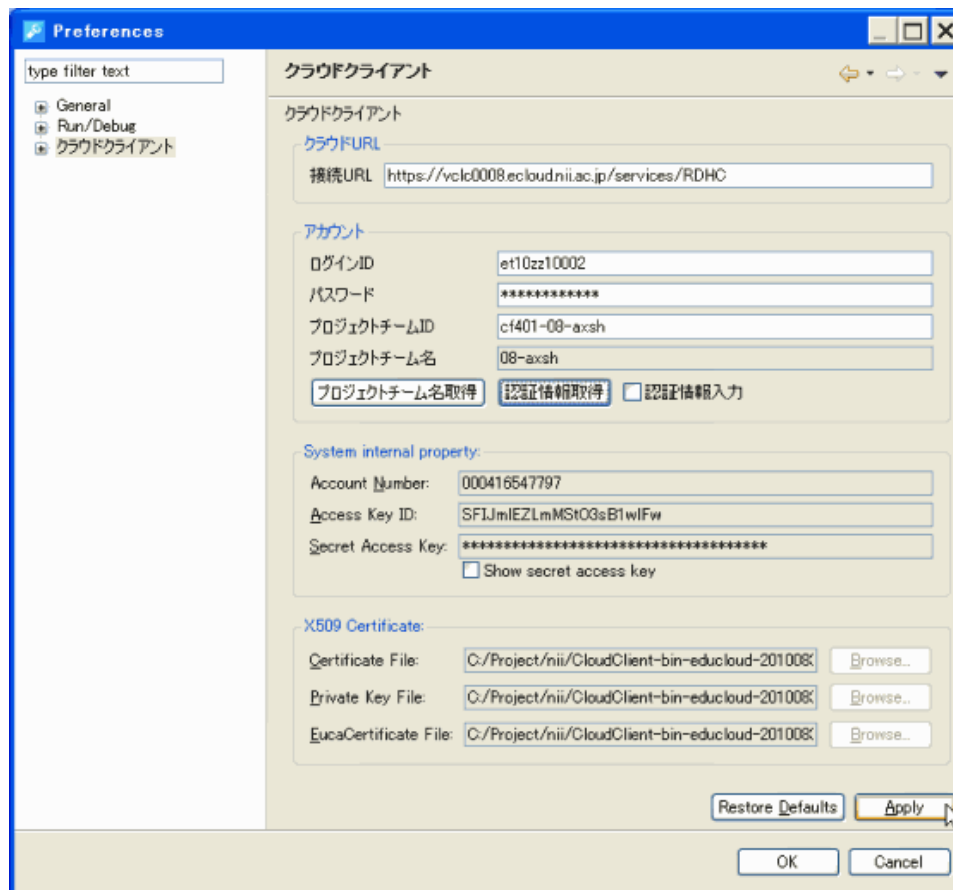


図 3: 認証情報取得

次に左側のクラウドクライアントのタブを開き「Regions」を設定します。  
Ec2:Region の項目で「Eucalyptus」を選択します。(図 4:Regions の設定)

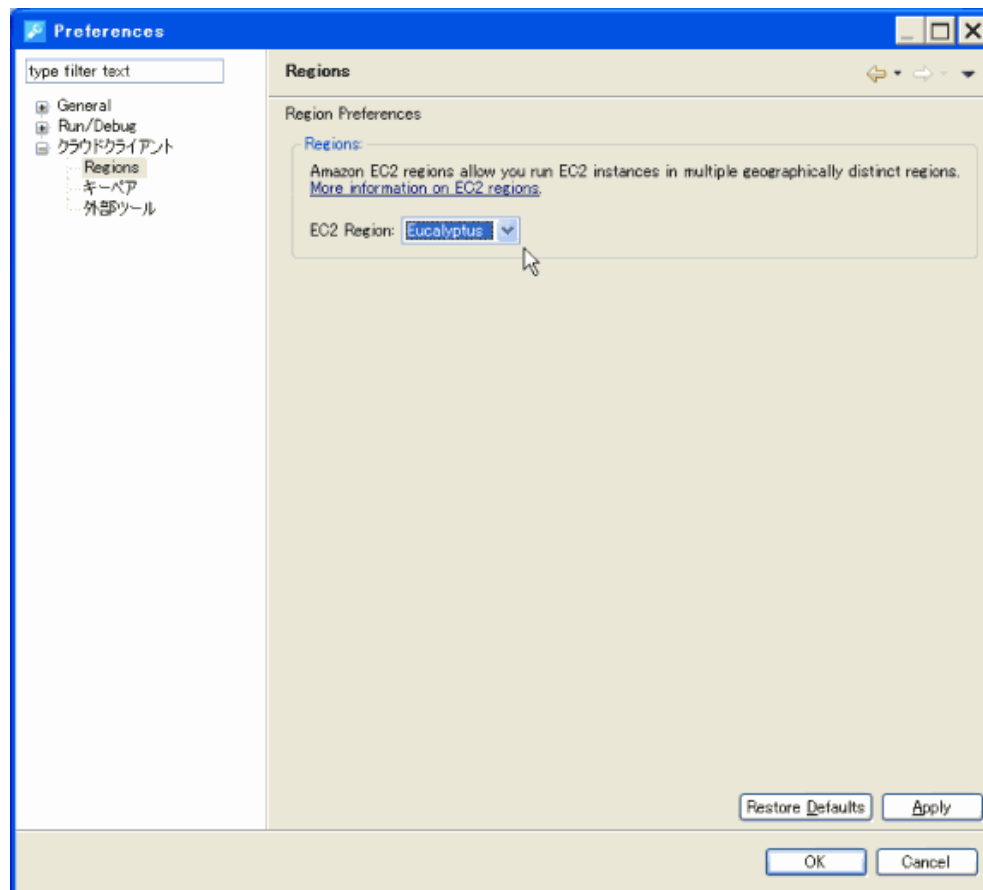


図 4: Regions の設定

次にキーペアの作成をします。

キーペアの一覧を右クリックし「新規キーペアの作成」をクリックします。(図 5:新規キーペアの作成)

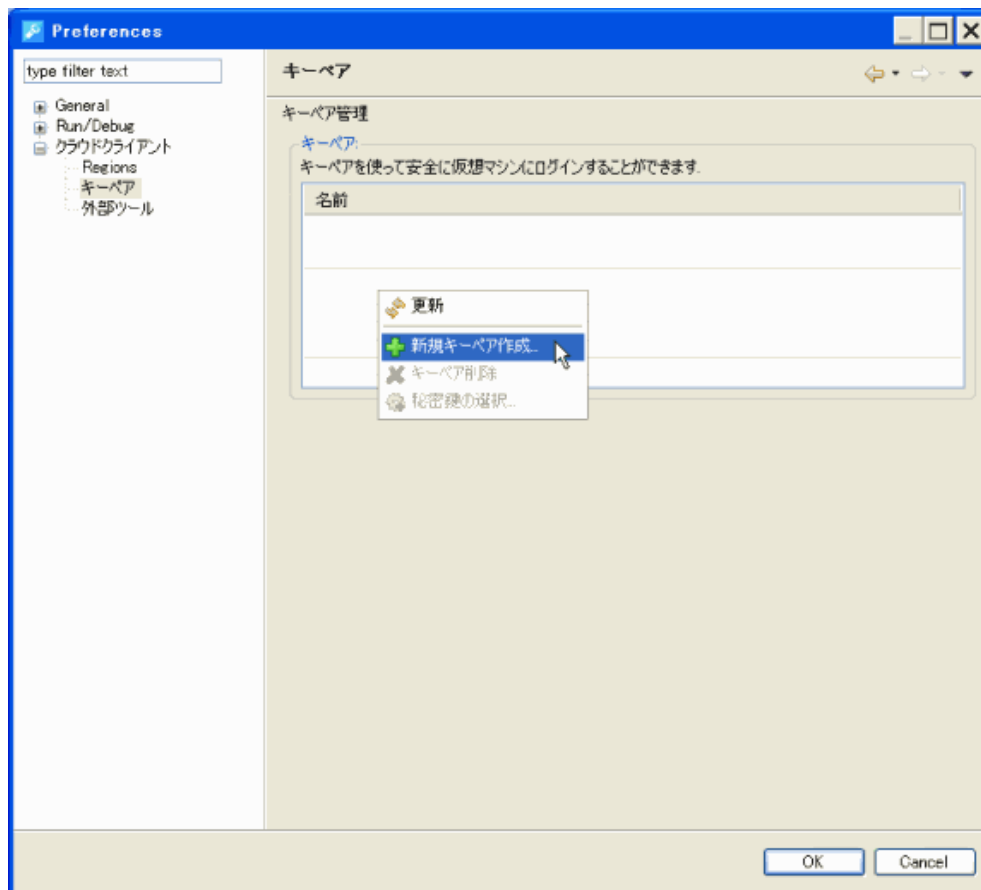


図5: 新規キーペアの作成

キーペア名に「ログイン ID」を入力し、「OK」ボタンをクリックします。(図 6: 新規キーペアの保存)

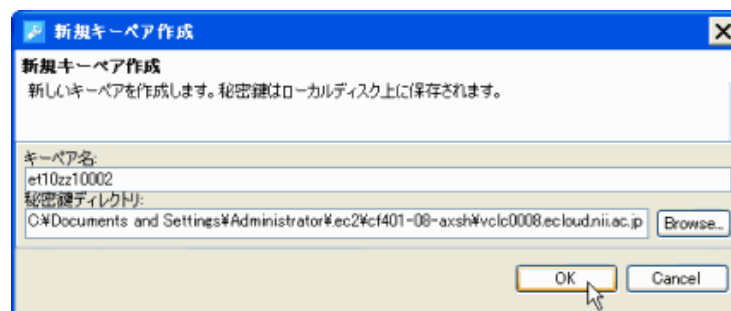


図6: 新規キーペアの保存

また秘密鍵ディレクトリは変更しないでください。

作成されたキーペアに緑色のチェックマークが入っていることを確認してください。(図 7: キーペアの管理)



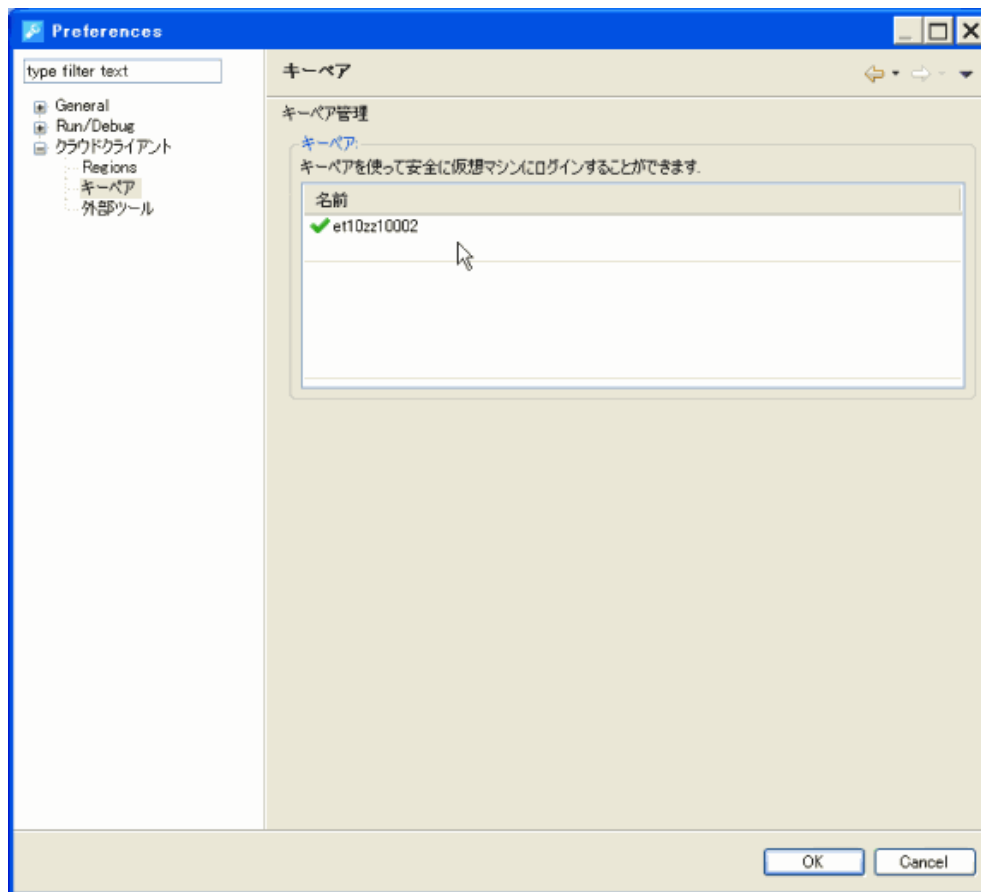


図7: キーペアの管理

「U:¥usr¥.ec2¥[プロジェクト ID]¥[クラウド名]」の中に「ログイン ID」という pem キーが作成されていることを確認してください。(図 8: pem ファイルの保存場所)

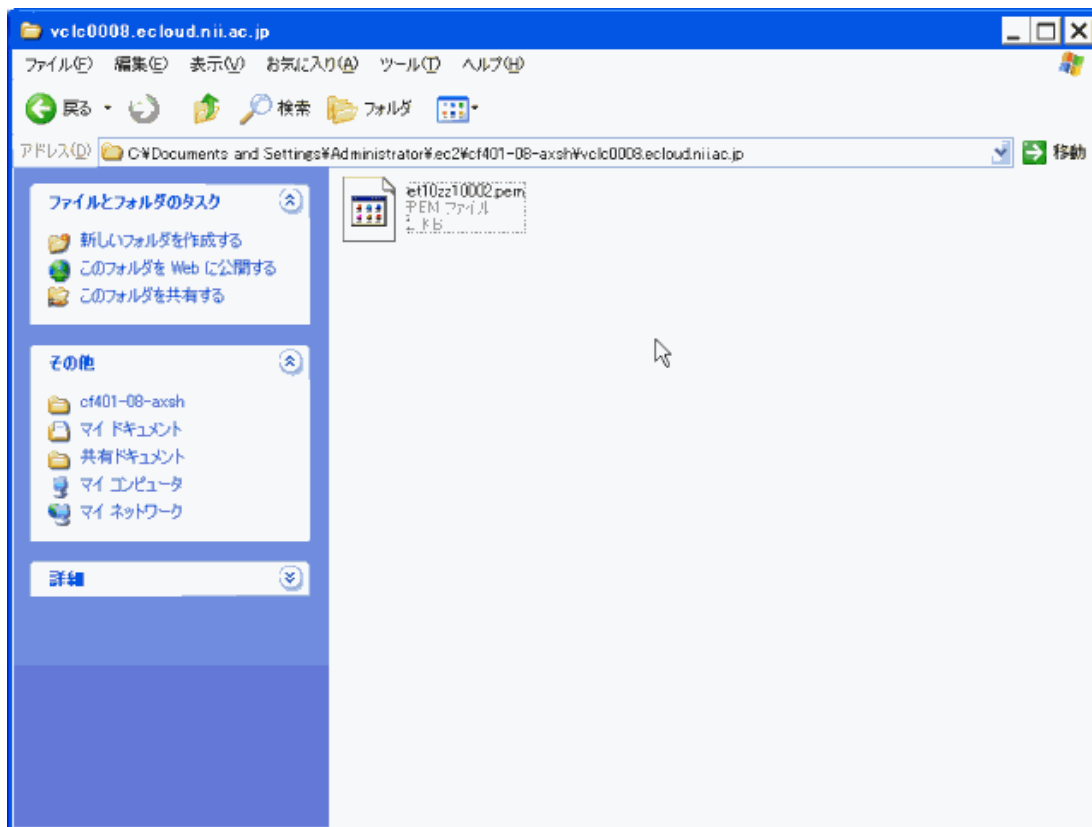


図 8:pem ファイルの保存場所

### 1.2.2. インスタンスを起動する

Windows の「スタート→すべてのプログラム→クラウドクライアント→CloudClient」を起動し、仮想マシンイメージ一覧の misc フォルダの中にある今回起動する「CentOS」イメージを選びます。(図 9: 起動する仮想マシンイメージの選択)

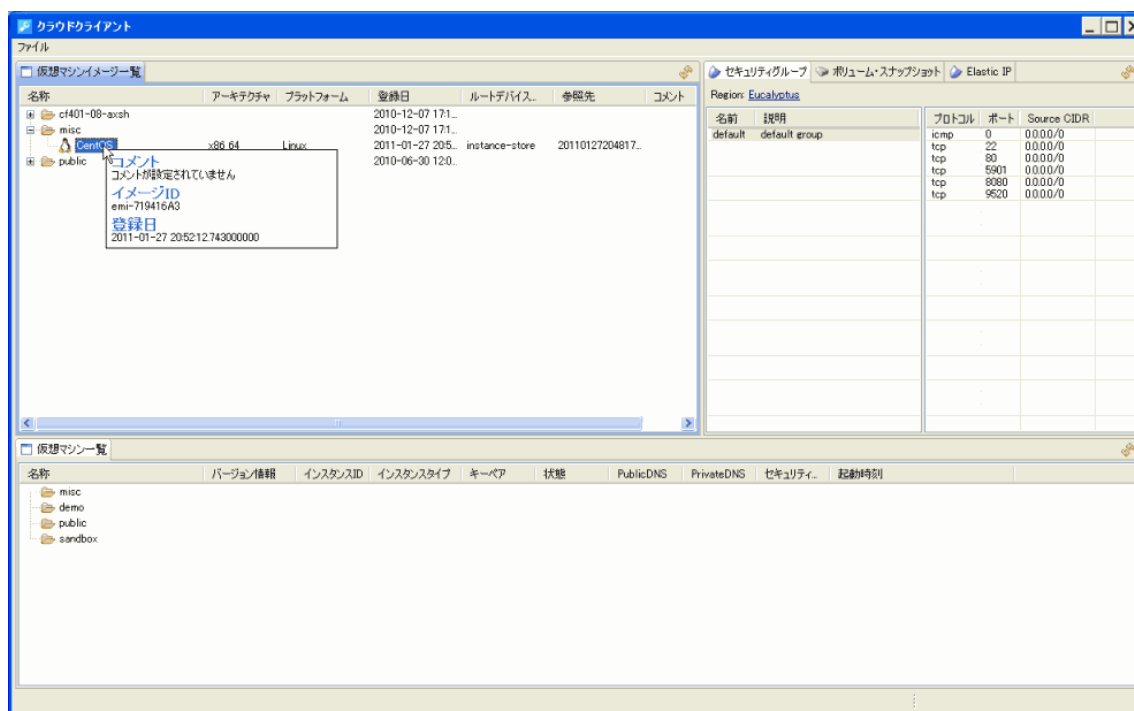


図 9: 起動する仮想マシンイメージの選択

選んだイメージを右クリックし、「仮想マシン起動」をクリック(図 10:仮想マシンの起動)すると仮想

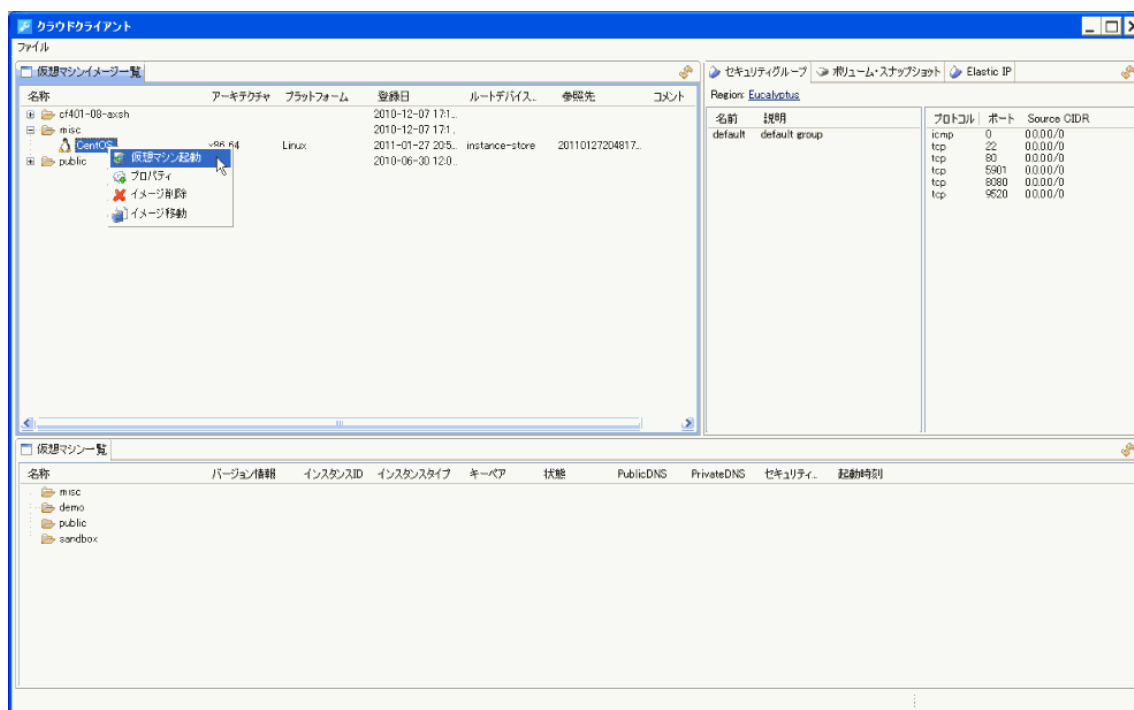


図 10: 仮想マシンの起動

マシン起動ダイアログ(図 11:仮想マシン起動ダイアログ)が表示されます。

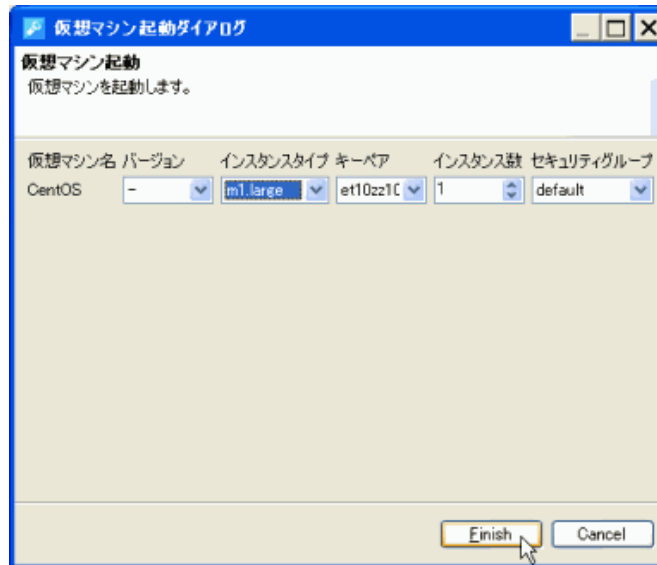


図 11: 仮想マシン起動ダイアログ

以下の項目を選択し「Finish」ボタンをクリックして仮想マシンを起動します。

- バージョン: -
- インスタンスタイプ: m1.large
- キーペア ログイン ID
- インスタンス数 1
- セキュリティグループ default

仮想マシン一覧の misc フォルダの中に先ほどの設定で起動している仮想マシンがあることを確認してください。(図 12: 仮想マシンの起動確認)

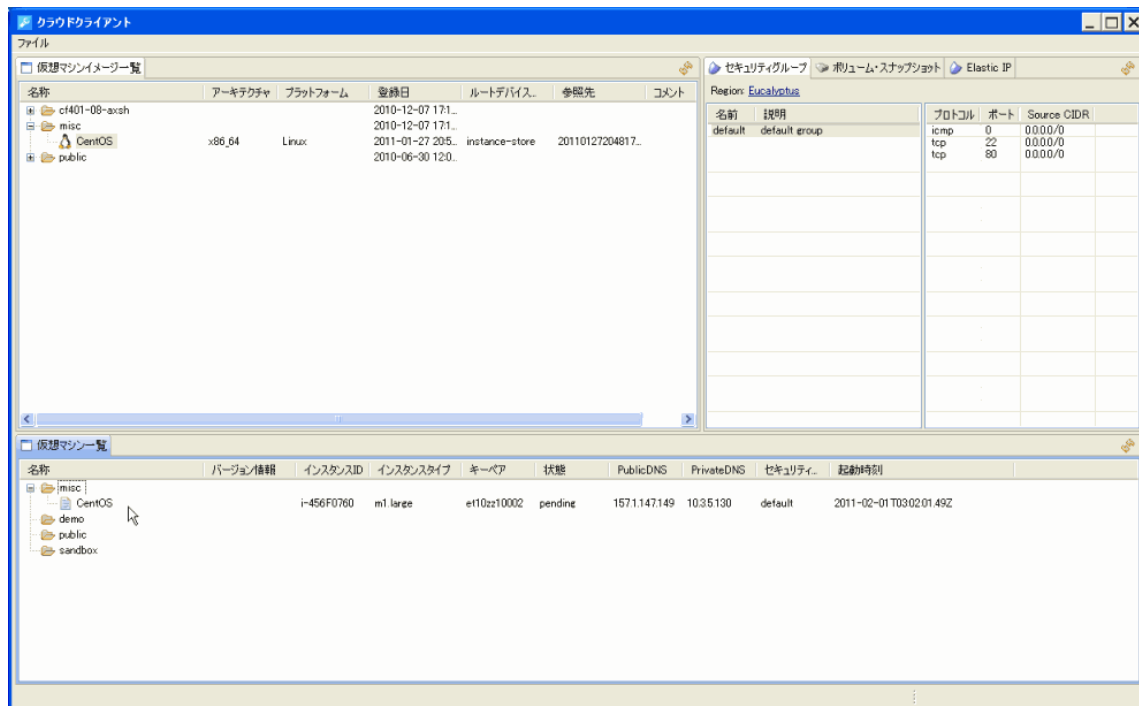


図 12: 仮想マシンの起動確認

仮想マシン一覧の右側にある「リフレッシュ」を押すと情報が更新されます。仮想マシンの状態が

pending から running になるまで確認してください。  
なお起動するまで、数分かかります。

### 1.2.3. Putty の秘密鍵を作成する

初めて edubase Cloud で起動した仮想マシンに putty でログインするとき秘密鍵の作成が行われます。

先ほど起動した仮想マシンにログインしてみましょう。仮想マシン一覧の misc フォルダの中にある「CentOS」を右クリックし、「シェルの起動」を選択する(図 13:シェルの起動)と puttygen 鍵変換のダイアログが表示されます。「OK」ボタンをクリックします。(図 14:鍵変換)

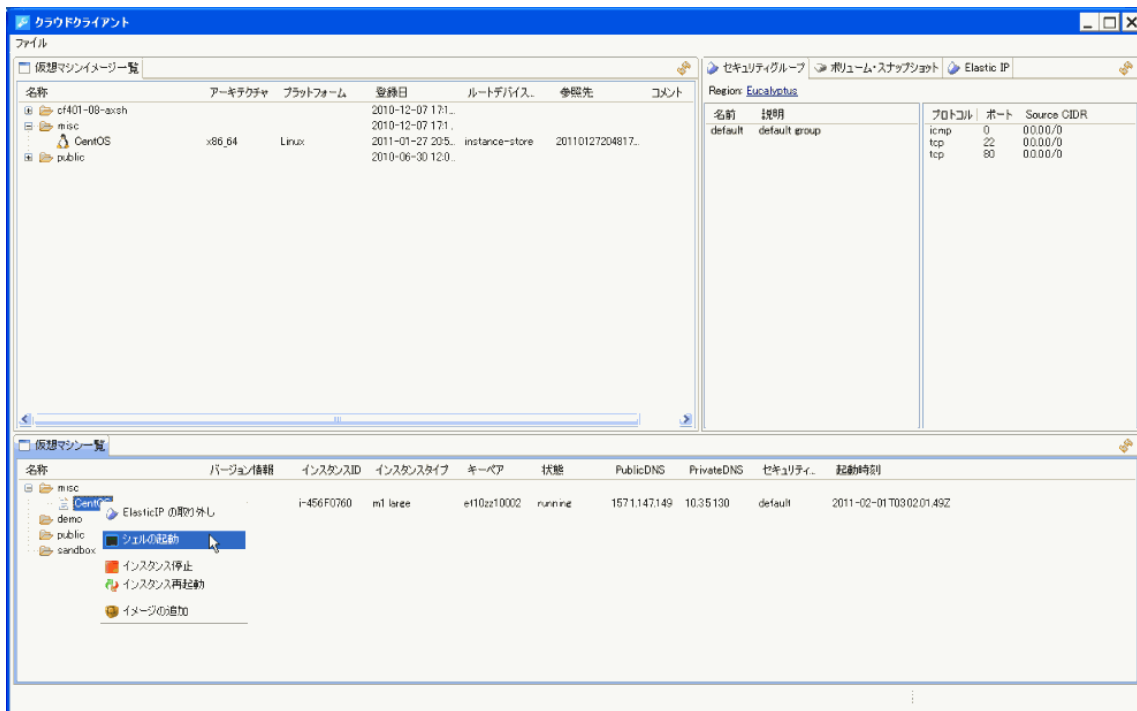


図 13: シェルの起動

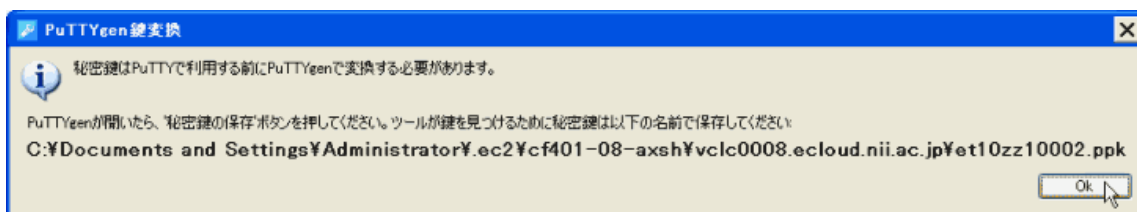


図 14: 鍵変換



図 15: 外部鍵のインポート

外部の鍵のインポートに成功したというダイアログが表示されます。(図 15: 外部鍵のインポート) 「OK」ボタンをクリックします。

鍵生成ダイアログが表示されるので、「Save Private key」ボタンをクリックします。(図 17: 警告)

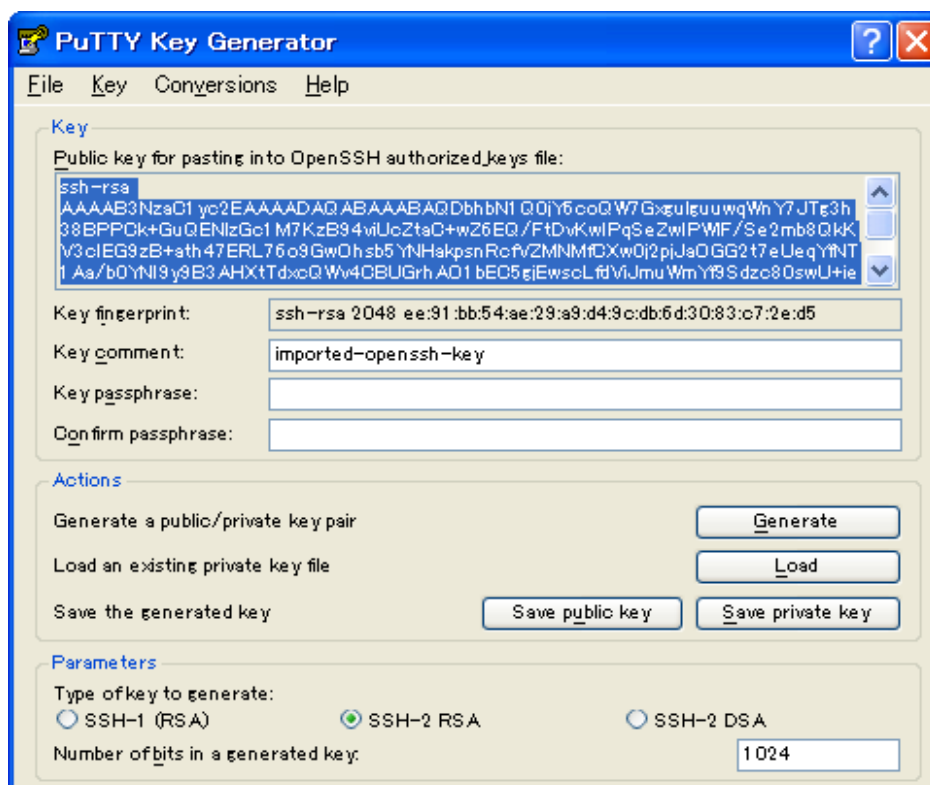


図 16: 鍵の生成

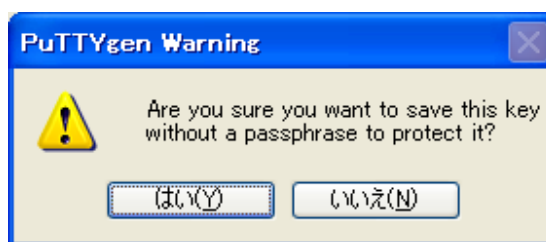


図 17: 警告

警告ダイアログが表示されますが「はい」をクリックします。(図 21:セキュリティ警告)

ファイル名に「ログイン ID」を入力し「保存」ボタンをクリックします。(図 18:秘密鍵の保存)

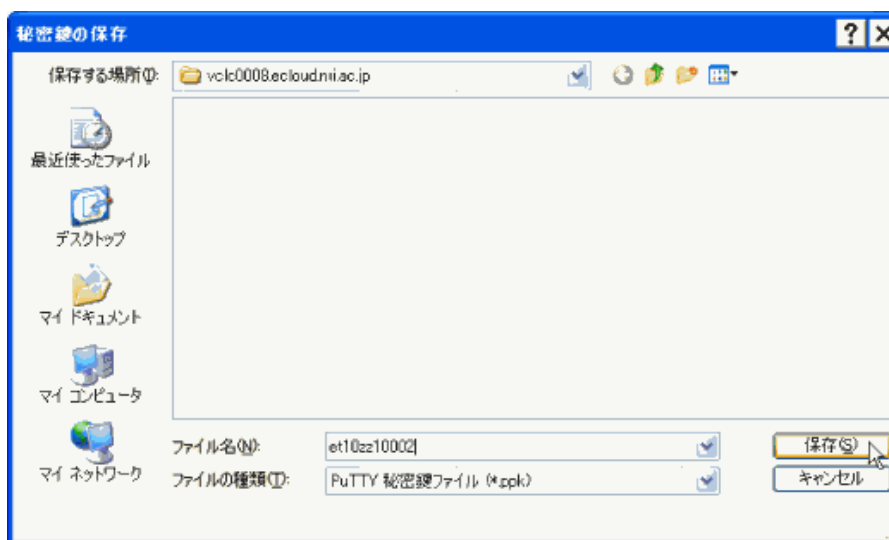


図 18: 秘密鍵の保存

「U:¥usr¥.ec2¥[プロジェクト ID]¥[クラウド名]」フォルダの中に「ログイン ID」という名前の ppk ファイ

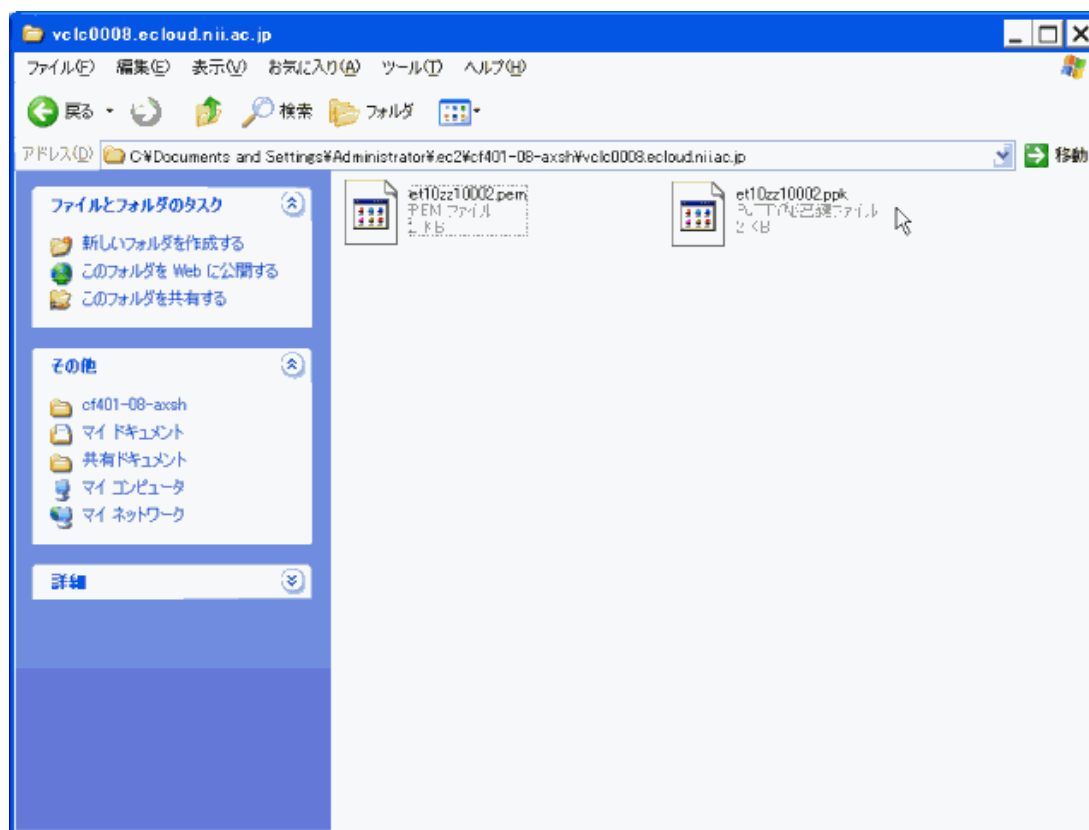


図 19: 秘密鍵の保存場所

ルがあることを確認します。(図 19:秘密鍵の保存場所)

#### 1.2.4. Apache をインストールする

仮想マシン一覧の misc フォルダの中にある「CentOS」を右クリックし、「シェルの起動」を選択します。(図 20:シェルの起動)



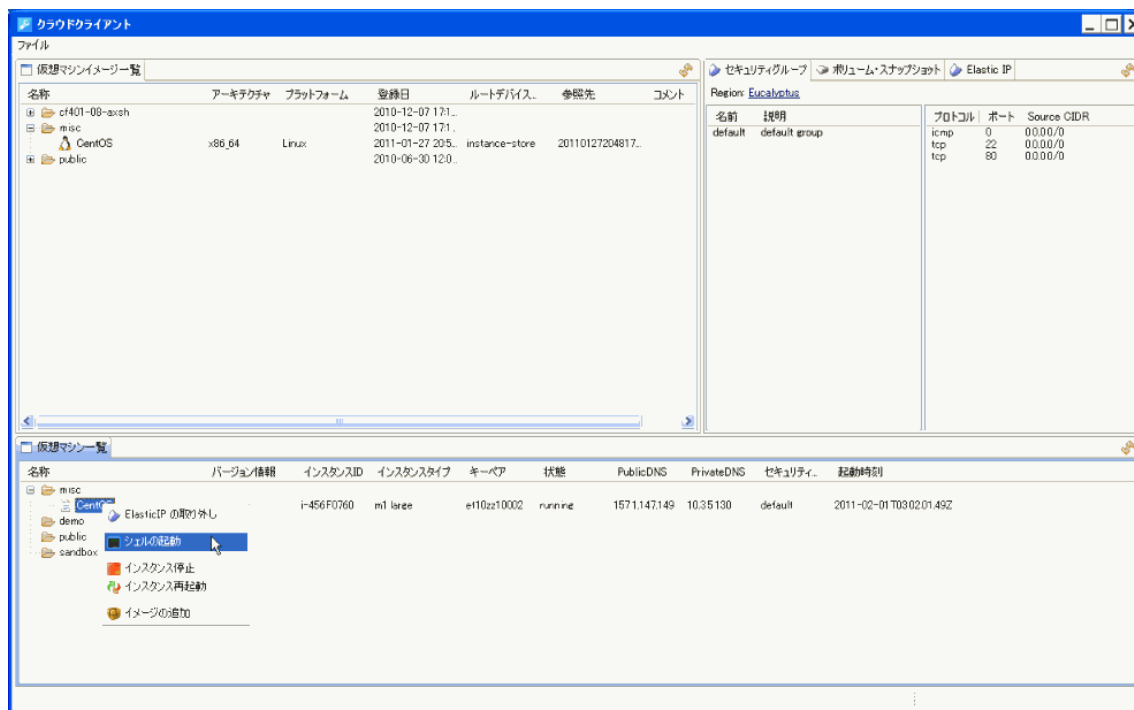


図 20: シェルの起動

セキュリティ警告ダイアログが表示されます。「はい」ボタンをクリックします。(図 32: プロジェクトのイ



図 21: セキュリティ警告

ンポート完了)

以下のコマンドで Apache をインストールします

```
-bash-3.2# yum -y install httpd
```

## 1.2.5. Apache を設定する

マシンの起動時に Apache が起動するように以下のコマンドで設定します。

```
-bash-3.2# chkconfig httpd on
```

以下のコマンドで設定を確認します。3、4、5番が on になっていることを確認してください。

```
-bash-3.2# chkconfig --list httpd
httpd          0:off 1:off 2:on  3:on  4:on  5:on  6:off
```

## 1.2.6. 仮想マシンイメージを作成する

先ほど Apache の設定を行った仮想マシンのイメージを作成します。最初に以下のコマンドで仮想マシンが現在使用している容量を確認します。

```
-bash-3.2# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1        4.0G  1.3G  2.5G  34% /
/dev/sda2        5.5G  140M  5.0G   3% /mnt
```

現在「1.3G」使用しているので、今回は「2G」で作成します。

仮想マシン一覧の misc フォルダの中の「CentOS」選びます。

選んだイメージを右クリックし、「イメージの追加」を選択する(図 22:イメージの追加)とイメージ追加

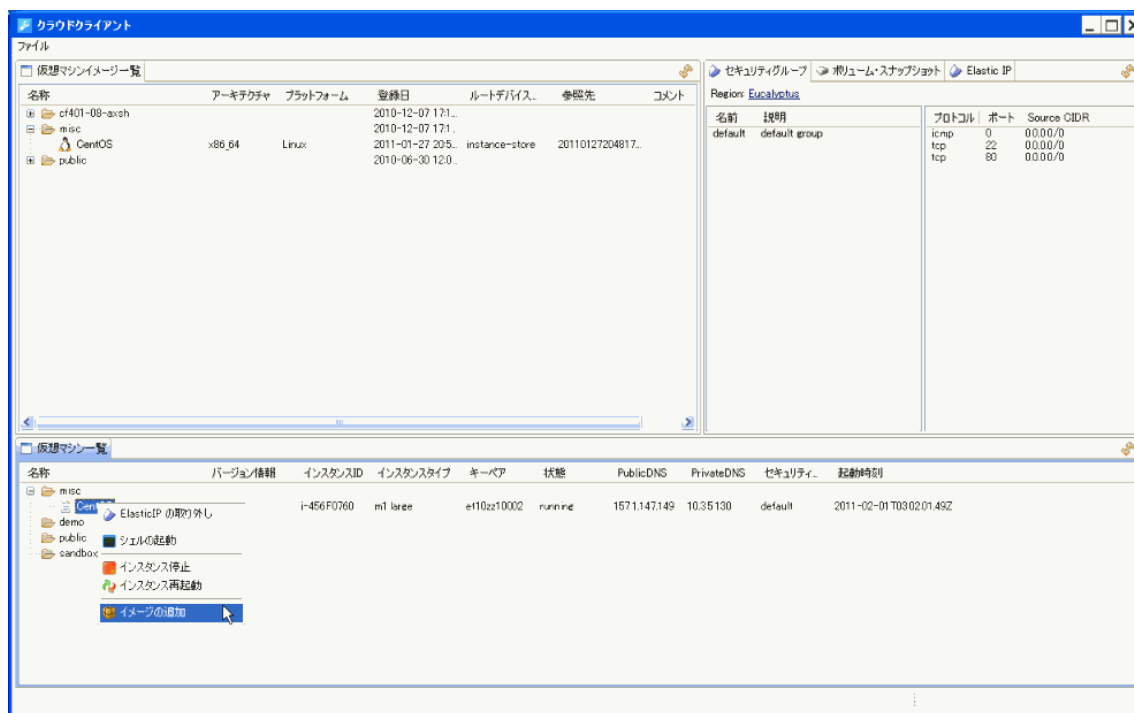


図 22: イメージの追加

ダイアログが表示されます。(図 23:イメージ追加ダイアログ)

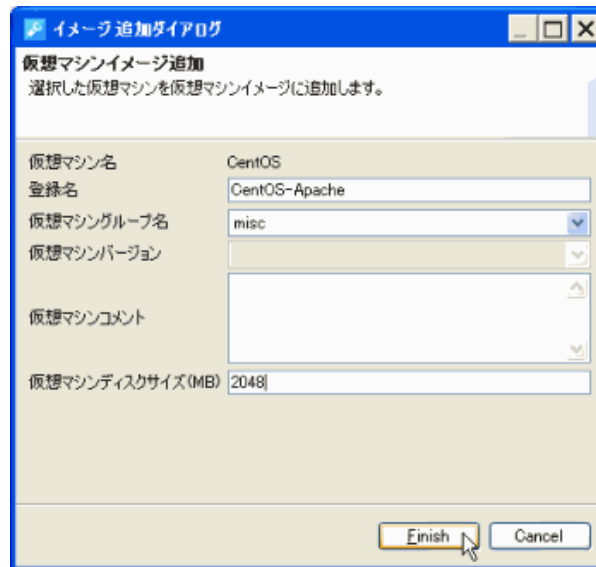


図 23: イメージ追加ダイアログ

以下の項目を入力し、「Finish」ボタンをクリックしてイメージを作成します。

- 登録名: CentOS-Apache
- 仮想グループ名 misc
- 仮想マシンディスクサイズ 2048

クラウドクライアントの画面右下に登録の進捗が表示されます。

仮想マシンイメージの登録が終わると完了のダイアログ(図 24:イメージ作成完了)が表示され、仮想マシンイメージ一覧の misc フォルダの中に登録した名前を追加されます。(図 25:作成したイメージの確認)

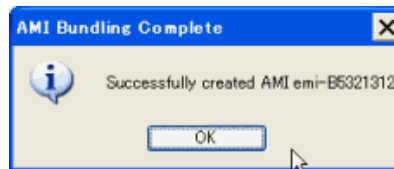


図 24: イメージ作成完了

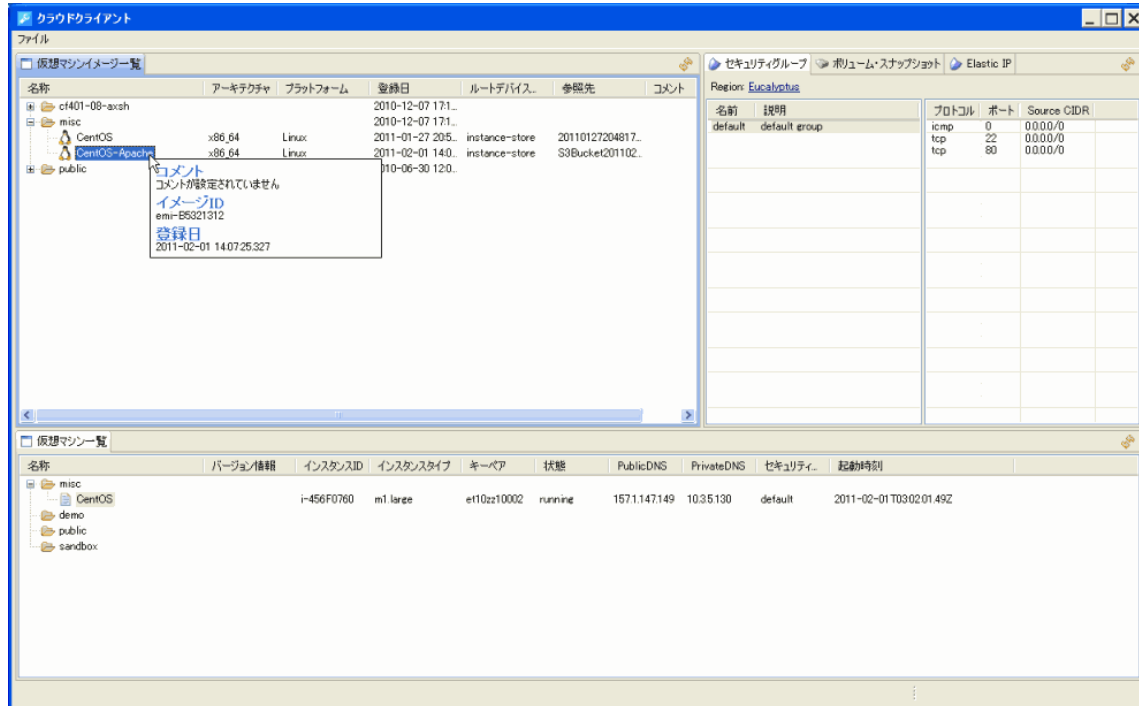


図 25: 作成したイメージの確認

### 【参考】イメージを作成する時の注意点

現在起動しているイメージの **HDD サイズ**が実際に使用しているサイズの2倍必要です。  
イメージを作成しているときに容量がなくなってイメージが作成できなくなります。

イメージを作成する場合は十分な容量を確保して作成してください。

### 1.2.7. 動作確認をする

作成したインスタンスを起動します。(1.2.2を参照)

Web ブラウザで以下の **URL** にアクセスして、画面が表示されることを確認してください。(図 26: Web ブラウザからの確認)

URL: [http://\[仮想マシンの PublicDNS\]](http://[仮想マシンの PublicDNS])

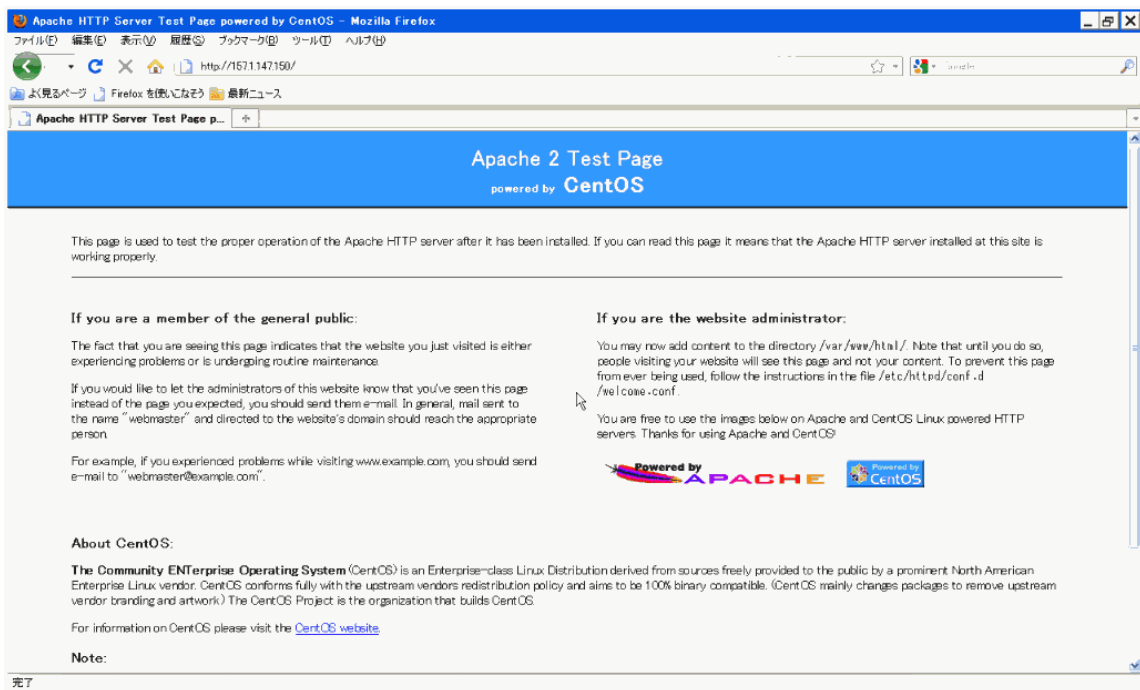


図 26: Web ブラウザからの確認

## 1.2.8. インスタンスを終了する

仮想マシン一覧の misc フォルダの中にある仮想マシンを右クリックし、「インスタンス停止」を選択

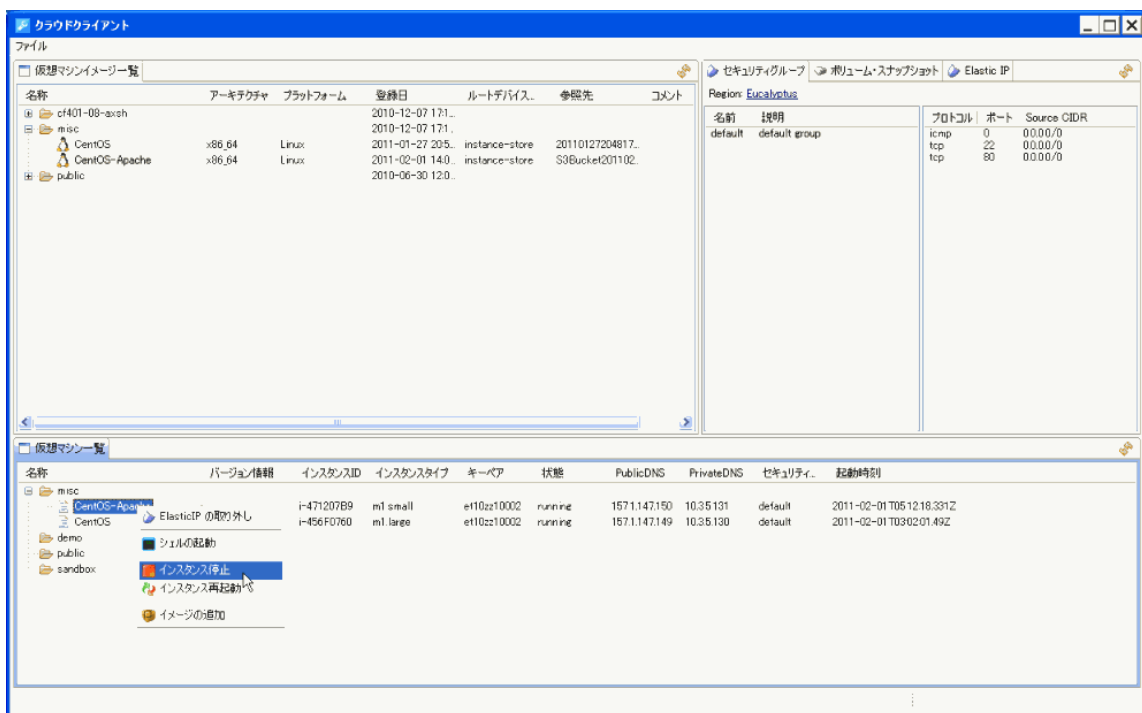


図 27: 仮想マシンの停止

します。(図 27: 仮想マシンの停止)

## 2章 WebAPI

### 2.1 Typica について

Typica は amazon のクラウドサービス (SQS,EC2,CloudWatch,AutoScaling,ELB,SimpleDB,SNS,FPS,DevPay) で使える QueryAPI です。

Java で利用できます。

Eucalyptus Community Cloud でテストがされていますので edubase Cloud でも利用することができます。

詳しくは、以下の URL を参照してみてください。

URL: <http://code.google.com/p/typica/>

### 2.2 Typica を使ってみよう

実際に java で typica を使って edubase Cloud を操作してみましよう。

用意するもの

- typica-sample.zip(ファイル)

ダウンロードしてきた typica-sample.zip を解凍し「U:¥usr¥workspace」フォルダの中に入れておいてください。

#### 2.2.1. eclipse プロジェクトをインポートする

デスクトップの eclipse アイコン[eclipse3.4.2(JRE1.6)]から eclipse を起動します。

ワークスペースランチャーが表示される場合はワークスペースを「U:¥usr¥workspace」に設定して、「OK」ボタンをクリックしてください。

「ファイル→インポート」をクリックします。(図 28:eclipse プロジェクトのインポート)

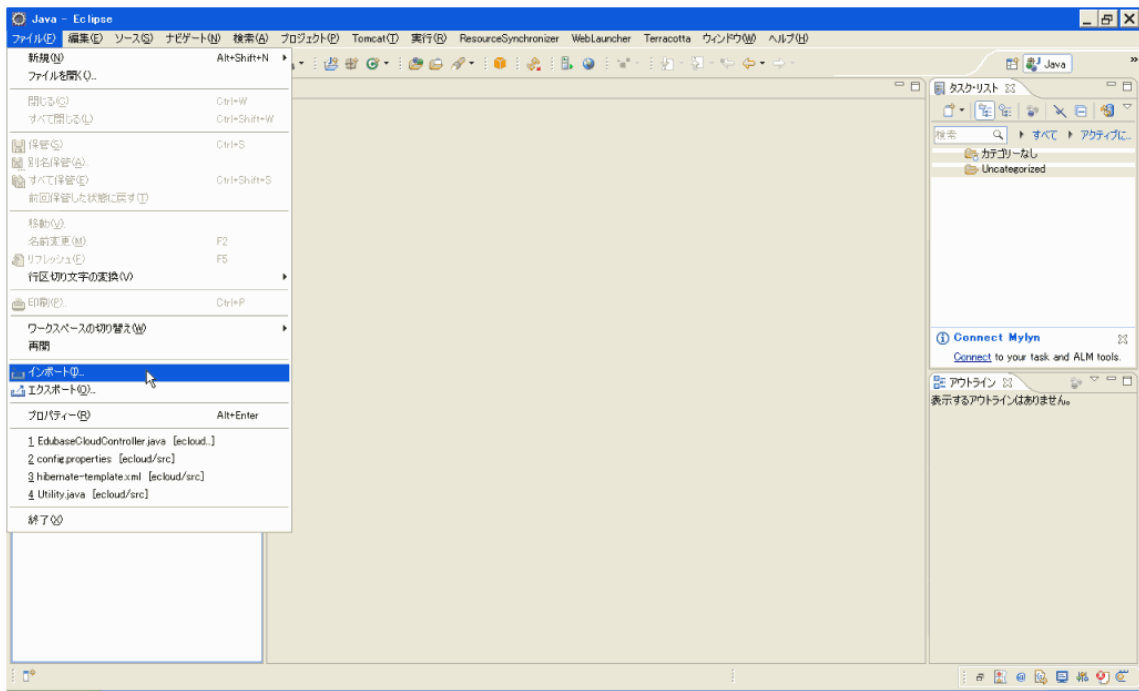


図 28: eclipse プロジェクトのインポート

「一般」フォルダの「既存プロジェクトをワークスペースへ」を選択し、「次へ」をクリックします。(図

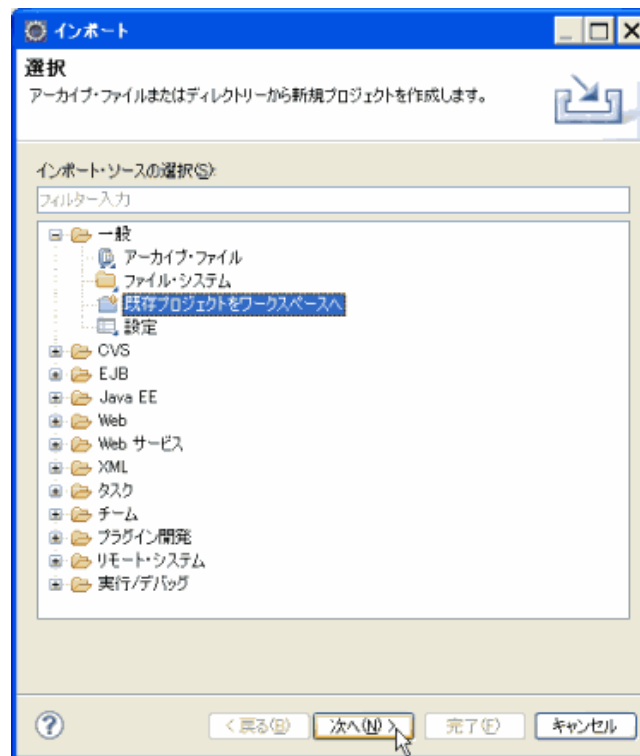


図 29: 選択

29:選択)

「ルート・ディレクトリの選択」を選択し、「参照」ボタンをクリックします。(図 30:プロジェクトのインポート)

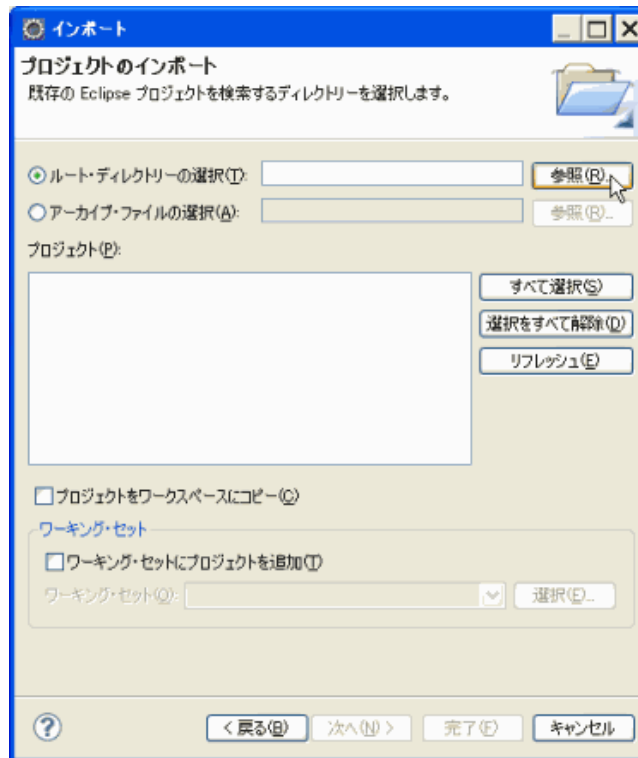


図 30: プロジェクトのインポート

「U:\usr¥workspace」フォルダ以下にある「typica-sample」を選択し、「OK」ボタンをクリックします。

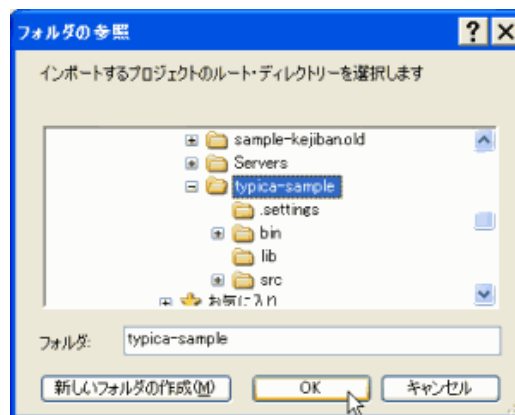


図 31: フォルダの参照

(図 31:フォルダの参照)

「終了」ボタンをクリックする(エラー: 参照先が見つかりません。)とパッケージ・エクスプローラーに「typica-sample」が表示されます。(図 33:インポートしたプロジェクトの確認)



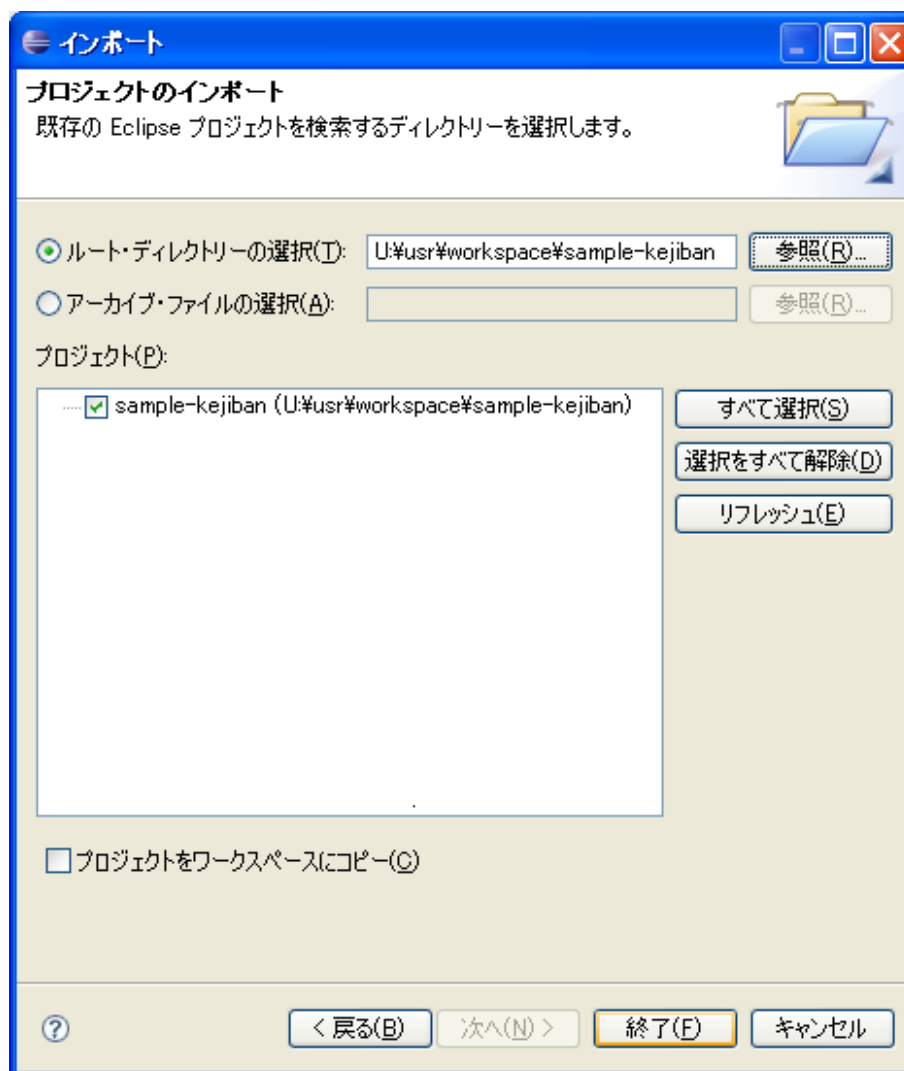


図 32: プロジェクトのインポート完了

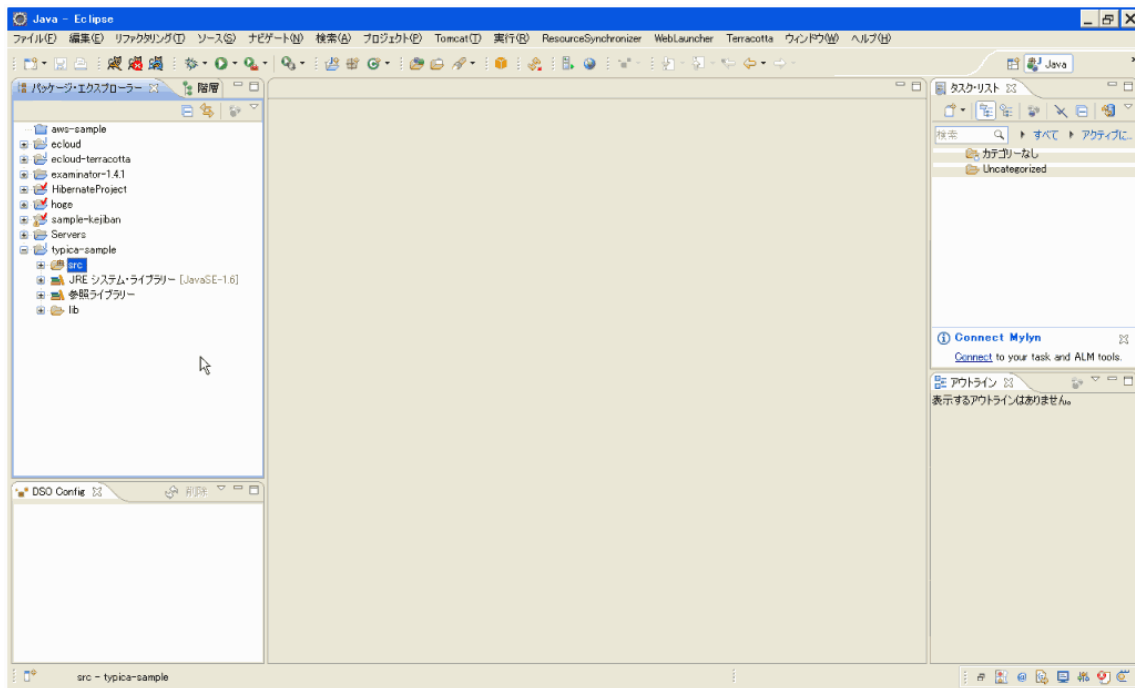


図 33: インポートしたプロジェクトの確認

### 2.2.2. 登録されている仮想マシンイメージの一覧を取得する

それではインポートしたプロジェクトを使って現在使っているクラウドの仮想マシンイメージの一覧を取得してみましょう。

パッケージ・エクスプローラーの中の「typica-sample→src→jp.ac.nii.typica→EdubaseCloudController」をダブルクリックしてソースコードを表示してください。

ソースコード:EdubaseCloudController.java

```
package jp.ac.nii.typica;

import java.util.ArrayList;
import java.util.List;
import com.xerox.amazonws.ec2.ImageDescription;
import com.xerox.amazonws.ec2.Jec2;

public class EdubaseCloudController {

    static Jec2 ec2;
    private static void init() throws Exception {
        String awsSecretKey = "awsSecretKey";
        String awsAccessId = "awsAccessId";
        String endpoint = "endpoint";
        ec2 = new Jec2(awsAccessId, awsSecretKey, false, endpoint);
        ec2.setSignatureVersion(1);
        ec2.setResourcePrefix("/services/Eucalyptus");
    }
}
```

```

    }

    public static List<ImageDescription> describeImages() throws Exception {
        List<String> params = new ArrayList<String>();
        List<ImageDescription> images = ec2.describeImages(params);
        return images;
    }

    public static void main(String[] args) throws Exception {
        init();
        List<ImageDescription> images = describeImages();
        for(ImageDescription img : images) {
            System.out.println(img);
        }
    }
}

```

まず、edubaseCloud への接続情報を記述します。

```

private static void init() throws Exception {
    String awsSecretKey = "awsSecretKey";
    String awsAccessId = "awsAccessId";
    String endpoint = "endpoint";

    ec2 = new Jec2(awsAccessId, awsSecretKey, false, endpoint);
    ec2.setSignatureVersion(1);
    ec2.setResourcePrefix("/services/Eucalyptus");
}

```

以下の項目の部分を修正します。

- awsSecretKey
- awsAccessId
- endpoint

Windows の「スタート→すべてのプログラム→クラウドクライアント→CloudClient」を起動し、「ファイル→設定」を選択します。(図 34:設定の変更)

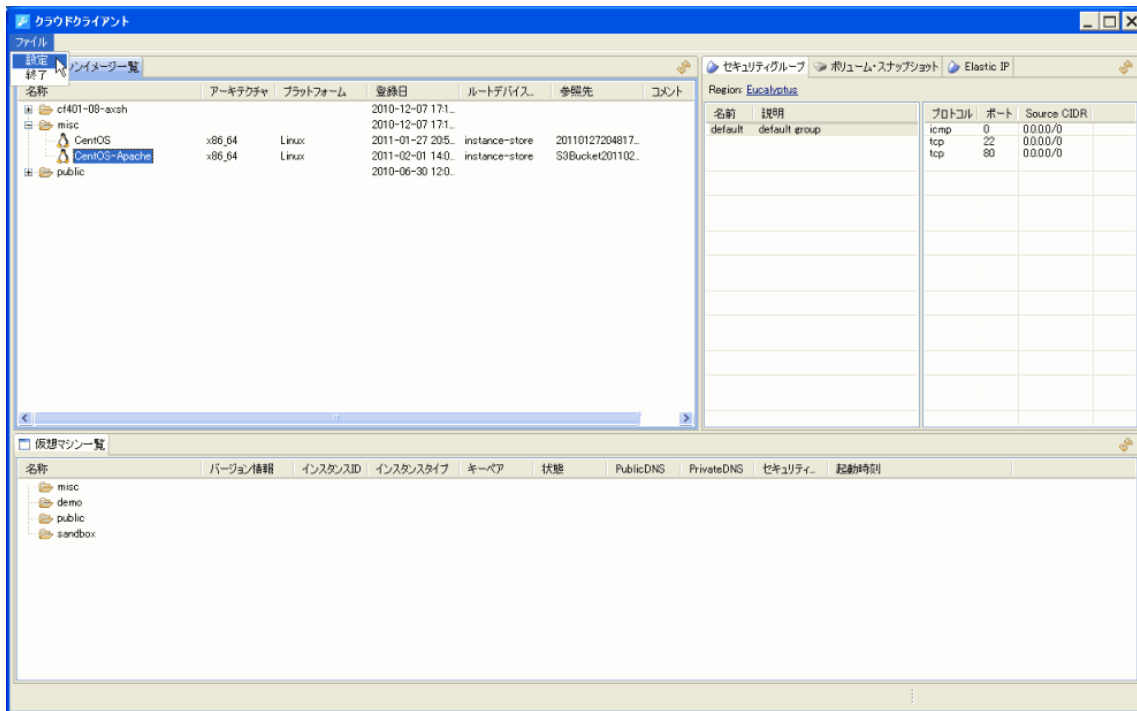


図 34: 設定の変更

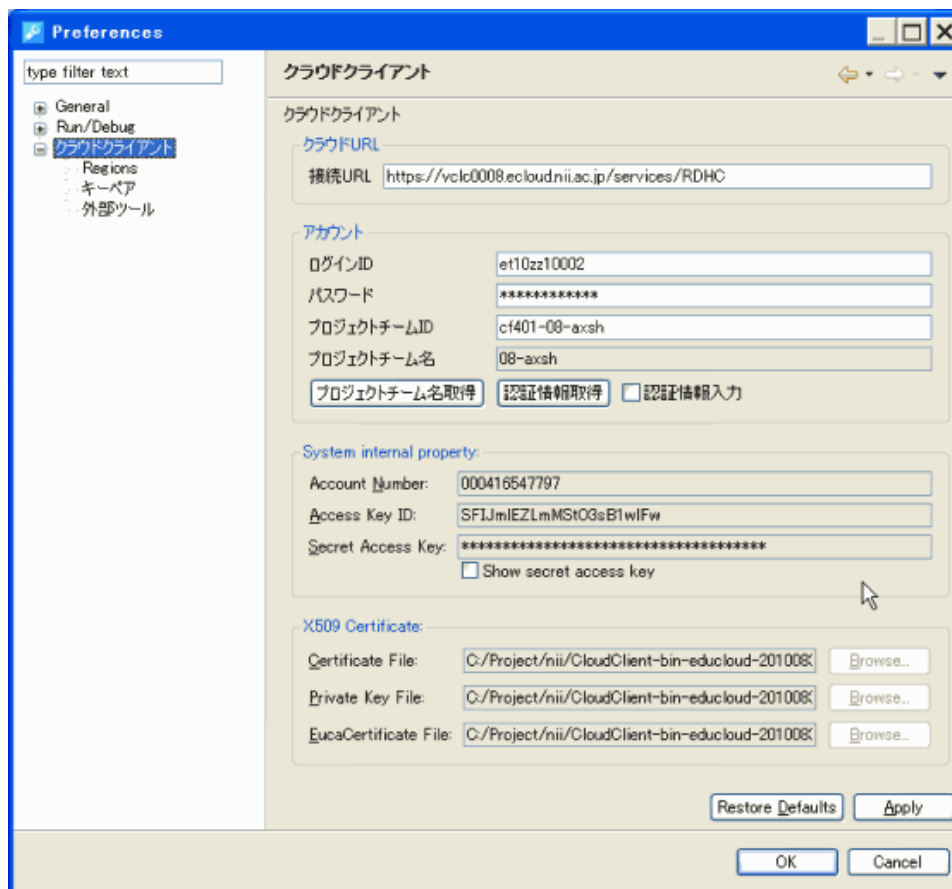


図 35: クラウドクライアントの設定確認

クラウドクライアントの設定を確認します。(図 35:クラウドクライアントの設定確認)

ソースコードの当該箇所にコピーします。

config.properties	クラウドクライアントの Preserceces ダイアログ
aws Sercret Key	Secret Access Key
aws Access Id	Access Key ID
endpoint	接続 URL のホスト名 (例: vclc0008.eccloud.nii.ac.jp)

次に現在登録されているイメージの list を取得するメソッドを確認してください。

```
public static List<ImageDescription> describeImages() throws Exception{
    List<String> params = new ArrayList<String>();
    List<ImageDescription> images = ec2.describeImages(params);
    return images;
}
```

ここでは、現在登録されているイメージを全て取得し、結果をリストにして返しています。

最後に実行メソッドを確認してください。

```
public static void main(String[] args) throws Exception {
    init();
    List<ImageDescription> images = describeImages();
    for(ImageDescription img : images) {
        System.out.println(img);
    }
}
```

ここでは、先ほど作成した接続情報を読み取り、イメージのリストを取得し、結果をプリントしています。

それでは実行してみましょう。

パッケージ・エクスプローラーにある「typica-sample→src→jp.ac.nii.typica→EdubaseCloudController」を右クリックし、「実行→Java アプリケーション」を選択します。(図 36:アプリケーションの実行)



```
Java - typica-sample/src/p/c/nni/typica/EdubaseCloudController.java - Eclipse
ファイル 編集 拡張機能 リファクタリング ツール ナビゲーション 検索 フォジカト Tomcat 実行 ResourceSynchronizer WebLauncher TerraCotta Mayas ウインドウ ヘルプ
コンソール
(終了) EdubaseCloudController [Java アプリケーション] C:\%systemdrive%\jvms\java6\bin\javaw.exe 2011/02/01 16:22:41
Image [Image]-B5021313, Loc=S3Bucket20110112412325/S3Bundle201101124123255.manifest.xml, ovm=cf401-08-axsh, state=available iSPublic=false, arch=x86_64, imgTy
Image [Image]-E7c12416D3, Loc=20100908151433_ramdisk_bucket/20100908151433_ramdisk.manifest.xml, ovm=admmin, state=available iSPublic=true, arch=x86_64, imgTy
Image [Image]-797616D3, Loc=20101127204817_ramdisk_bucket/20101127204817_ramdisk.manifest.xml, ovm=admmin, state=available iSPublic=true, arch=x86_64, imgTy
Image [Image]-E449D160B, Loc=20100908152246_kernel_bucket/20100908152246_kernel.manifest.xml, ovm=admmin, state=available iSPublic=true, arch=x86_64, imgTy=ke
Image [Image]-783816D4, Loc=20100908152246_ramdisk_bucket/20100908152246_ramdisk.manifest.xml, ovm=admmin, state=available iSPublic=true, arch=x86_64, imgTy=ke
Image [Image]-E07D14D3, Loc=centos53-1386-machine/centos-5.3-x86_194.manifest.xml, ovm=admmin, state=available iSPublic=true, arch=1386, imgTy=machine, kern
Image [Image]-E4B41955, Loc=centos53-32bit-ramdisk/initrd-2.6.18-194.17.4.el5xen.img.manifest.xml, ovm=admmin, state=available iSPublic=true, arch=1386, imgTy
Image [Image]-E27F4BCES, Loc=CentOS54-x86_64-Trac-Subversion/centOS54-x86_64-Trac-Subversion.manifest.xml, ovm=admmin, state=available iSPublic=true, arch=x86_6
Image [Image]-B585131C, Loc=S3Bucket20110112414172/S3Bundle201101124141722.manifest.xml, ovm=cf401-08-axsh, state=available iSPublic=false, arch=x86_64, imgTy
Image [Image]-B585131C, Loc=S3Bucket201101124172/S3Bundle2011011241723.manifest.xml, ovm=cf401-08-axsh, state=available iSPublic=false, arch=x86_64, imgTy=ke
Image [Image]-E41516D3, Loc=20100924161823_kernel_bucket/20100924161823_kernel.manifest.xml, ovm=cf401-08-axsh, state=available iSPublic=true, arch=x86_64, imgTy=ke
Image [Image]-E8217611, Loc=S3Bucket2011011222022/S3Bundle20110112220221.manifest.xml, ovm=cf401-08-axsh, state=available iSPublic=false, arch=x86_64, imgTy
Image [Image]-B6D3C131, Loc=S3Bucket20110121424815/S3Bundle20110121424815.manifest.xml, ovm=cf401-08-axsh, state=available iSPublic=false, arch=x86_64, imgTy
Image [Image]-E7c17316FF, Loc=20100908151848_ramdisk_bucket/20100908151848_ramdisk.manifest.xml, ovm=admmin, state=available iSPublic=true, arch=x86_64, imgTy=ke
Image [Image]-B689131B, Loc=S3Bucket20110117184212/S3Bundle20110117184212.manifest.xml, ovm=cf401-08-axsh, state=available iSPublic=false, arch=x86_64, imgTy=ke
Image [Image]-E7c17316FF, Loc=20101127204817_ramdisk_bucket/20101127204817_ramdisk.manifest.xml, ovm=cf401-08-axsh, state=available iSPublic=false, arch=x86_64, imgTy=ke
Image [Image]-E42D1605, Loc=20101127204817_kernel_bucket/20101127204817_kernel.manifest.xml, ovm=admmin, state=available iSPublic=true, arch=x86_64, imgTy=ke
Image [Image]-B5821312, Loc=S3Bucket20110201140429/S3Bundle20110201140429.manifest.xml, ovm=cf401-08-axsh, state=available iSPublic=false, arch=x86_64, imgTy=ke
Image [Image]-E047F513D, Loc=centos53-1386-kernel/vmlinuz-2.6.24-19-xen.manifest.xml, ovm=admmin, state=available iSPublic=true, arch=1386, imgTy=kernel, kern
Image [Image]-E7c17316FF, Loc=centos53-1386-ramdisk/initrd-2.6.24-19-xen.manifest.xml, ovm=admmin, state=available iSPublic=true, arch=1386, imgTy=ramdisk, kern
Image [Image]-E840130C, Loc=S3Bucket20110112152403/S3Bundle20110112152403.manifest.xml, ovm=cf401-08-axsh, state=available iSPublic=false, arch=x86_64, imgTy=ke
Image [Image]-E7c17316FF, Loc=20100908151848_ramdisk_bucket/20100908151848_kernel.manifest.xml, ovm=admmin, state=available iSPublic=true, arch=x86_64, imgTy=ke
Image [Image]-B689131B, Loc=S3Bucket20110112412426/S3Bundle20110112412426.manifest.xml, ovm=cf401-08-axsh, state=available iSPublic=false, arch=x86_64, imgTy=ke
Image [Image]-E7c17316FF, Loc=centos53-1386-machine/wsgm.img.manifest.xml, ovm=admmin, state=available iSPublic=true, arch=1386, imgTy=machine, kernel
Image [Image]-B259130D, Loc=S3Bucket20110112103132/S3Bundle20110112103132.manifest.xml, ovm=cf401-08-axsh, state=available iSPublic=false, arch=x86_64, imgTy
Image [Image]-E7c17316FF, Loc=centos53-apache-kernel/vmlinuz-2.6.27.21-0.1-xen.manifest.xml, ovm=admmin, state=available iSPublic=true, arch=x86_64, imgTy=kern
Image [Image]-E83A0130, Loc=S3Bucket2011011231108/S3Bundle2011011231108.manifest.xml, ovm=cf401-08-axsh, state=available iSPublic=false, arch=x86_64, imgTy=ke
Image [Image]-E84D160A, Loc=centos53-32bit-kernel/vmlinuz-2.6.18-194.17.4.el5xen.manifest.xml, ovm=admmin, state=available iSPublic=true, arch=1386, imgTy=ke
Image [Image]-E7c17316FF, Loc=centos53-lisportal-machine/portal.img.manifest.xml, ovm=admmin, state=available iSPublic=true, arch=1386, imgTy=machine, kernel
Image [Image]-E7c17316FF, Loc=centos53-trac-kernel/initrd-2.6.27.21-0.1-xen.manifest.xml, ovm=admmin, state=available iSPublic=true, arch=x86_64, imgTy=ramdisk
Image [Image]-E7c17316FF, Loc=centos53-trac-kernel/vmlinuz-2.6.27.21-0.1-xen.manifest.xml, ovm=admmin, state=available iSPublic=true, arch=x86_64, imgTy=kernel
Image [Image]-E77B516C, Loc=2010112518351_ramdisk_bucket/2010112518351_ramdisk.manifest.xml, ovm=admmin, state=available iSPublic=true, arch=x86_64, imgTy=ke
Image [Image]-E7c17316FF, Loc=CentOS54-x86_64-Apache-MySQL-PHP/centOS54-x86_64-Apache-MySQL-PHP.manifest.xml, ovm=admmin, state=available iSPublic=true, arch=x86
Image [Image]-B5D0131A, Loc=S3Bucket20101217125330/S3Bundle20101217125330.manifest.xml, ovm=cf401-08-axsh, state=available iSPublic=false, arch=x86_64, imgTy
```

図 37: イメージのリスと取得

### 2.2.3. インスタンスを起動する

/\*\*

```

* Requests reservation of a number of instances.
* <p>
* This will begin launching those instances for which a reservation was
* successfully obtained.
* <p>
* If less than <code>minCount</code> instances are available no instances
* will be reserved.
*
* @param lc object containing launch configuration
* @return A {@link com.xerox.amazonws.ec2.ReservationDescription} describing the instances
that
*      have been reserved.
* @throws EC2Exception wraps checked exceptions
*/
public ReservationDescription runInstances(LaunchConfiguration lc) throws EC2Exception {
}

```

```

/**
* Launches the given AMI one time. The min and max values are '1'.
*
* @param imageId the AMI to launch
*/
public LaunchConfiguration(final String imageId) {

/** The name of the key file to access the AMI via ssh. */
private String keyName;

/** The minimum number of AMIs to launch. */
private int minCount;

/** The maximum (desired) number of AMIs to launch. */
private int maxCount;

/** The security group to launch the AMI in. */
private List<String> securityGroup;

/** The size of the hardware to launch the AMI in. */
private InstanceType instanceType = InstanceType.DEFAULT;

```

#### 2.2.4. 起動しているインスタンスの一覧を取得する

以下のメソッドを使い起動しているインスタンスの一覧を取得するスクリプトを作成してみましょう。

```

/**
* Gets a list of running instances.
* <p>

```

```

* If the array of instance IDs is empty then a list of all instances owned
* by the caller will be returned. Otherwise the list will contain
* information for the requested instances only.
*
* @param instanceIds An array of instances ({@link
com.xerox.amazonws.ec2.ReservationDescription.Instance#instanceId}).
* @return A list of {@link com.xerox.amazonws.ec2.ReservationDescription} instances.
* @throws EC2Exception wraps checked exceptions
*/
public List<ReservationDescription> describeInstances(String[] instanceIds) throws
EC2Exception {
}

```

## 2.2.5. インスタンスを終了する

以下のメソッドを使いインスタンスを終了するスクリプトを作成してみましょう。

```

/**
* Terminates a selection of running instances.
*
* @param instanceIds A list of instances ({@link
com.xerox.amazonws.ec2.ReservationDescription.Instance#instanceId}).
* @return A list of {@link InstanceStateChangeDescription} instances.
* @throws EC2Exception wraps checked exceptions
* TODO: need to return request id
*/
public List<InstanceStateChangeDescription> terminateInstances(List<String> instanceIds)
throws EC2Exception {
}

```



## 3章 教室外から接続するには

### 3.1 edubasecloud に接続するには

#### 3.1.1. PacketiX VPN のインストール

参考資料「edubase Cloud はじめの一步」に従い VPN の設定をしてください。

#### 3.1.2. Putty のインストール

以下のサイトから「Putty」をダウンロードし、インストールウィザードに従って、インストールしてください。

URL: <http://yebisuya.dip.jp/Software/PuTTY/>

#### 3.1.3. CloudClient のインストール

VPN がうまくインストールできていれば以下のサイトが閲覧可能になります。

URL: <http://start.ecloud.nill.ac.jp/>

サイトにアクセスし、右の方にある「初めての方へ」の「3:クイックスタート(トピック)」をクリックし、クイックスタートから資料をダウンロードしてください。

参考資料「e-cloud\_quick\_start.ppt」を参考に CloudClient の設定をしてください。

### 3.2 実行環境の構築

#### 3.2.1. java-jdk のインストール

以下のサイトから「jdk」をダウンロードし、インストールウィザードに従って、jdk をインストールしてください。

URL: <http://java.sun.com/javase/ja/6/download.html>

#### 3.2.2. eclipse のインストール

以下のサイトから「Eclipse IDE for Java EE Developers」をダウンロードし、アーカイブを解凍して所定のディレクトリにコピーします。

URL: <http://www.eclipse.org/downloads/packages/release/galileo/sr2>

#### 3.2.3. Winscp のインストール

以下のサイトから「WinSCP 4.2.9」をダウンロードし、インストールウィザードに従って、インストールしてください。

URL: <http://winscp.net/eng/download.php>