

Web 三層モデル(1)

参考資料

目次

1章 サンプルプログラム「掲示板」	4
1.1. 掲示板について	4
1.2. 掲示板を動かしてみよう	4
1.2.1. 各インスタンスに接続するためにセキュリティグループの設定をする.....	5
1.2.2. eclipse 上で起動スクリプトを動かすための設定をする	8
1.2.3. 起動スクリプトを実行する	13
1.2.4. 動作確認をする	15
2章 Apache2+Tomcat をスケールアウトする	18
2.1. スケールアウトするには	18
2.2. 演習のヒント	21
2.2.1. スケールアウトスクリプトを作成する.....	21
2.2.2. 動作確認をする	22
3章 【参考】手動でスケールアウトする場合	23
3.1. インスタンスを起動する.....	23
3.2. AppServer の設定をする	23
3.2.1. サンプルプログラムを編集する.....	23
3.2.2. Pagent を起動し鍵を登録する	23
3.2.3. AppServer にサンプルプログラムを配置する.....	25
3.2.4. Tomcat を起動する	26
3.3. WebServer の設定をする	26
3.3.1. proxy_ajp.conf の編集.....	26
3.3.2. Apache(httpd)を再起動する	26
3.4. LB の設定をする.....	26
3.4.1. nginx.conf の編集.....	26
3.4.2. nginx を再起動する.....	27
3.5. 動作確認する	27
3.5.1. Web ブラウザから確認する.....	27
3.5.2. データベースを確認する	27
3.5.3. AppServer の log を確認する.....	27
3.5.4. 起動しているマシンを終了する。	27

図の索引

図 1:動作環境の構成	4
図 2:全体の概要図	5
図 3:新規グループの追加	6
図 4:セキュリティグループの追加	6
図 5:セキュリティグループの追加確認	7
図 6:パーミッションの追加	7
図 7:セキュリティグループの設定	8
図 8:プロジェクトのインポート	8
図 9: 既存プロジェクトをワークスペースへ	9
図 10:参照先の選択	9
図 11:フォルダの選択	10
図 12:参照の完了	10
図 13:プロジェクトのインポート完了	11
図 14: クラウドクライアントの設定	12
図 15:クラウドクライアントの設定確認	13
図 16:起動スクリプトの実行	14
図 17:実行ログの確認	14
図 18:インスタンスの起動確認	15
図 19:インスタンスの起動完了	15
図 20:WEB ブラウザの表示	16
図 21:シェルの起動	16
図 22:停止スクリプトの実行	17
図 23:サーバーの停止	18
図 24:起動スクリプト実行時の全体構成	19
図 25:スケールアウトスクリプト実行時の全体構成	21
図 26:PAGENT の場所	24
図 27:秘密鍵のリスト	24
図 28:秘密鍵の登録	25
図 29:秘密鍵の登録確認	25
図 30:APPSERVER へログイン	25

1章 サンプルプログラム「掲示板」

1.1. 掲示板について

サンプルプログラム「掲示板」は LB、WebServer、AppServer、DBServer の構成 (図 1:動作環境の構成) で動作し、記事の新規書き込みと、書き込まれた記事の内容を閲覧できる Web システムです。

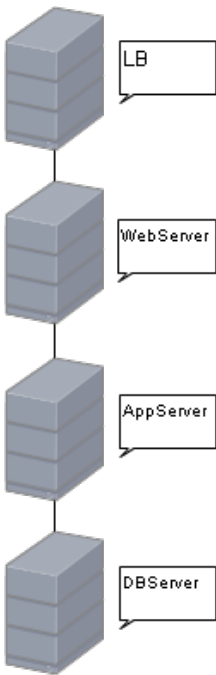


図 1:動作環境の構成

1.2. 掲示板を動かしてみよう

今回は起動スクリプトを実行して掲示板プログラムを動かします。

用意するもの

- ☐ ecloud.zip(ファイル)
- ☐ config.properties.zip(ファイル)

ecloud.zip は zip をを解凍して「U:usr¥workspace」以下に入れてください。

config.properties.zip は zip を解凍して自分の使用している「プロジェクト名」フォルダの中にあるものを「U:usr¥workspace¥ecloud¥src」の中に上書きして入れてください。

以下の図(図 2:全体の概要図)が起動スクリプトを使って掲示板を動かす全体の概要になります。

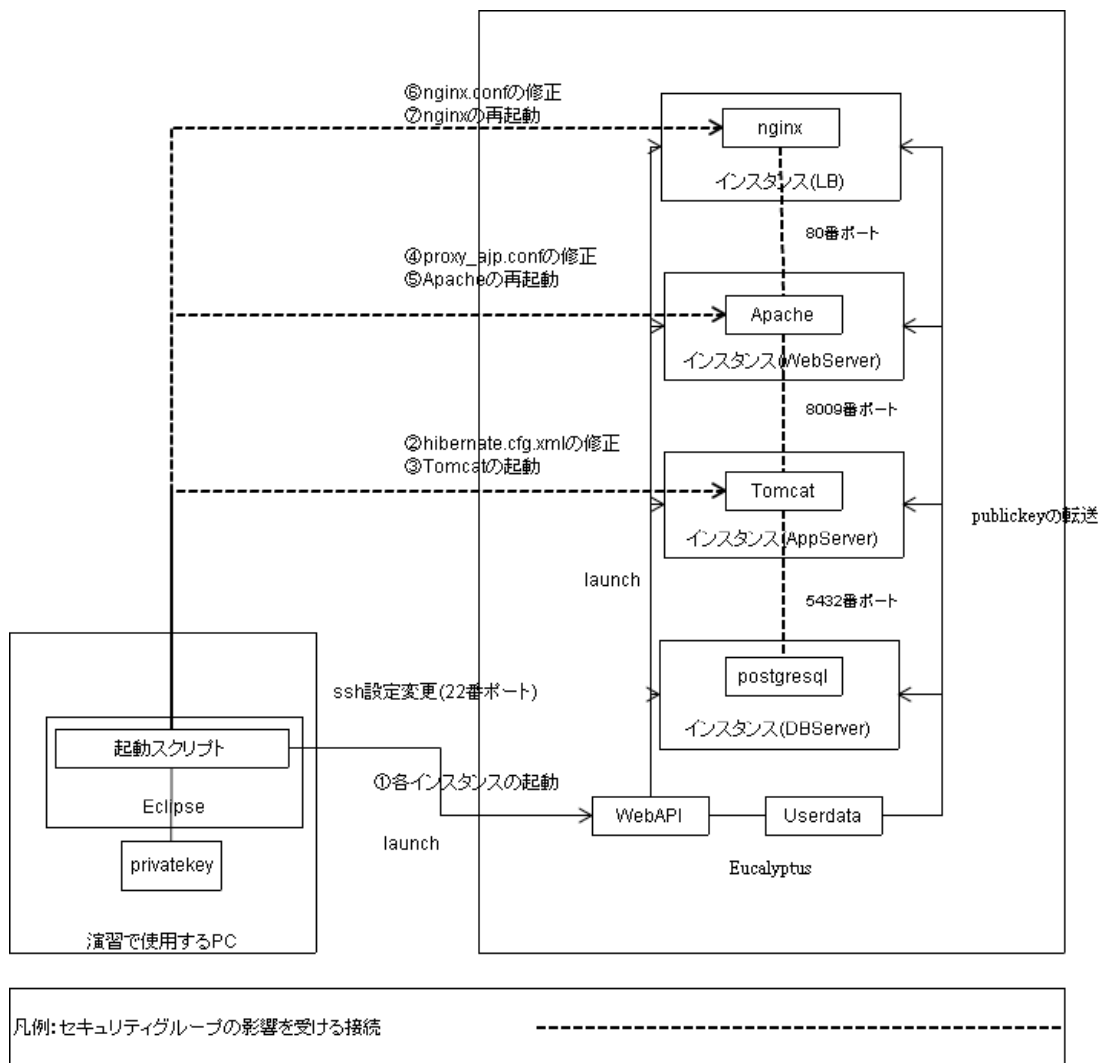


図 2:全体の概要図

- WebAPI 経由で各サーバーを起動する
- ssh、scp を使用し各サーバーのアプリケーションの設定を書き換える

今回起動スクリプトを実行するためにやらなければならないことは以下になります。

- 各インスタンスに接続するためにセキュリティグループの設定をする
- eclipse 上で起動スクリプトを動かすための設定をする

まずはセキュリティグループの設定から行いましょう。

1.2.1. 各インスタンスに接続するためにセキュリティグループの設定をする

まず最初に、起動する仮想マシンのセキュリティーの設定を行います。

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、「セキュリティグループ」タブをクリックします。

セキュリティグループの名前一覧を右クリックし、「新規グループの作成」選択します。(図 3:新規グ

ループの追加)

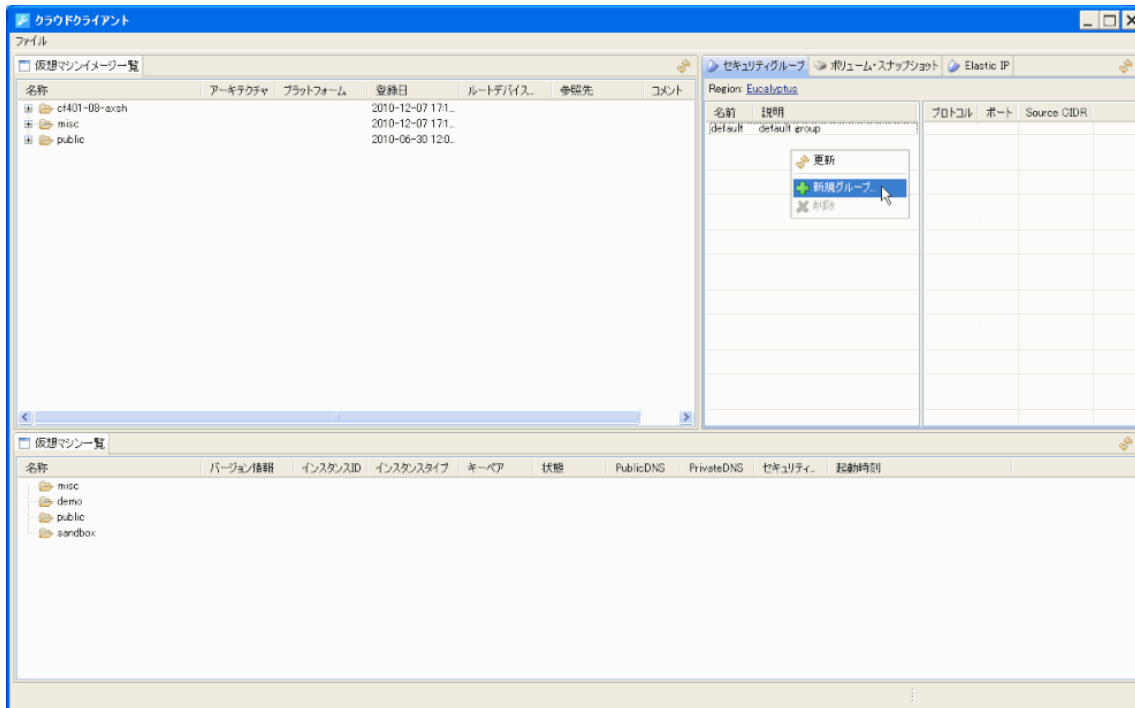


図 3:新規グループの追加

「セキュリティグループ名」に「ログイン ID」、「説明」に「sample」を入力し、「OK」ボタンをクリックします。(図 4:セキュリティグループの追加)

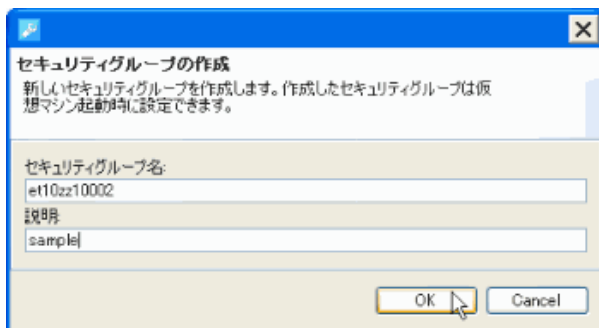


図 4:セキュリティグループの追加

セキュリティグループに「ログイン ID」名が追加されたのを確認してください。(図 5:セキュリティグループの追加確認)

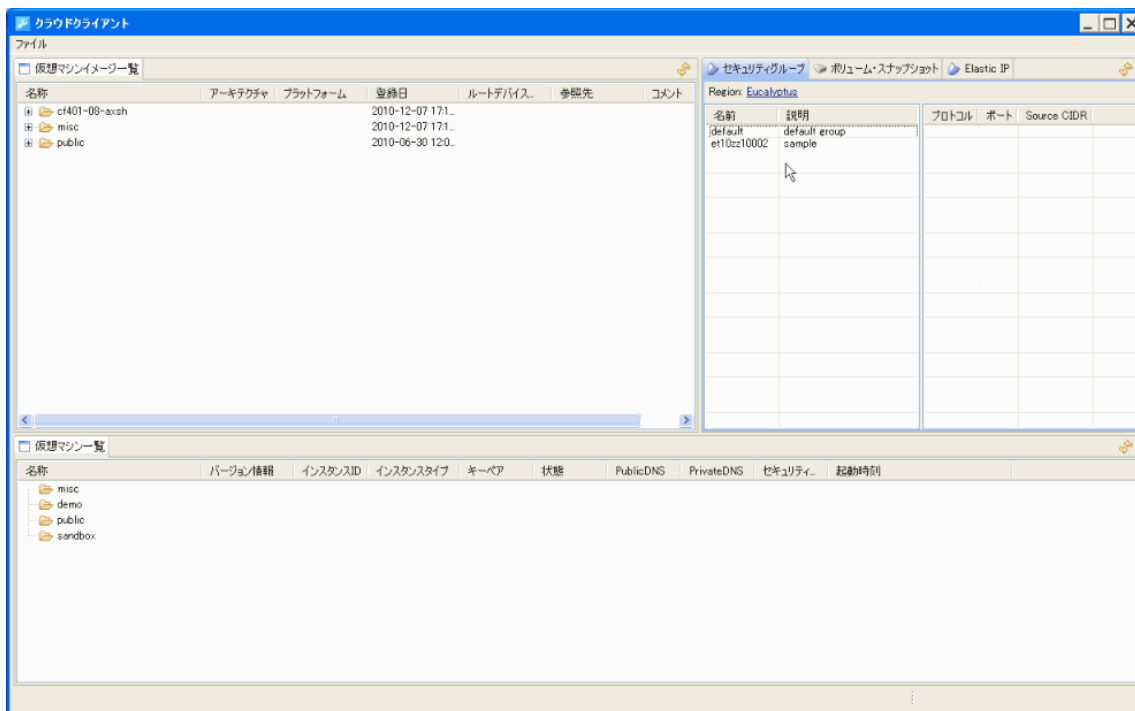


図 5:セキュリティグループの追加確認

次に今作成した、「ログイン ID」をクリックし、右側の一覧を右クリックして「パーミッションの追加」を選択します。(図 6:パーミッションの追加)

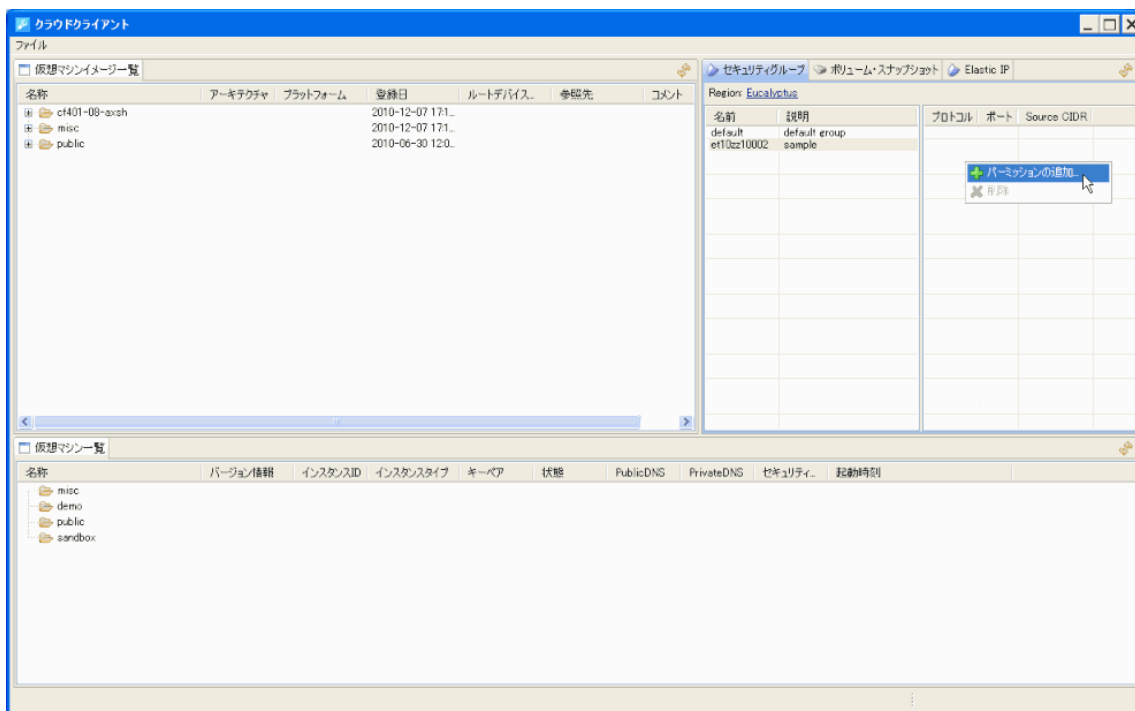


図 6:パーミッションの追加

以下の3つの項目を作成してください。(図 7:セキュリティグループの設定)

- ☐ Protocol: icmp
- ☐ Port or Port Range: 0
- ☐ Network Mask: 0.0.0.0/0

☐ Protocol: tcp
☐ Port or Port Range: 22
☐ Network Mask: 0.0.0.0/0

☐ Protocol: tcp
☐ Port or Port Range: 80
☐ Network Mask: 0.0.0.0/0

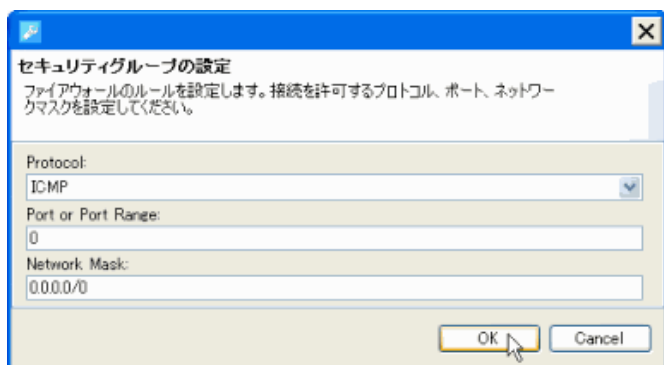


図 7:セキュリティグループの設定

1.2.2. eclipse 上で起動スクリプトを動かすための設定をする

次に、先ほど「U:\usr\workspace」の中においた掲示板を起動するためのスクリプトが入っている「ecloud」をインポートします。

デスクトップの eclipse アイコン[eclipse3.4.2(JRE1.6)]から「eclipse」を起動し、「ファイル→インポート」をクリックします。(図 8:プロジェクトのインポート)

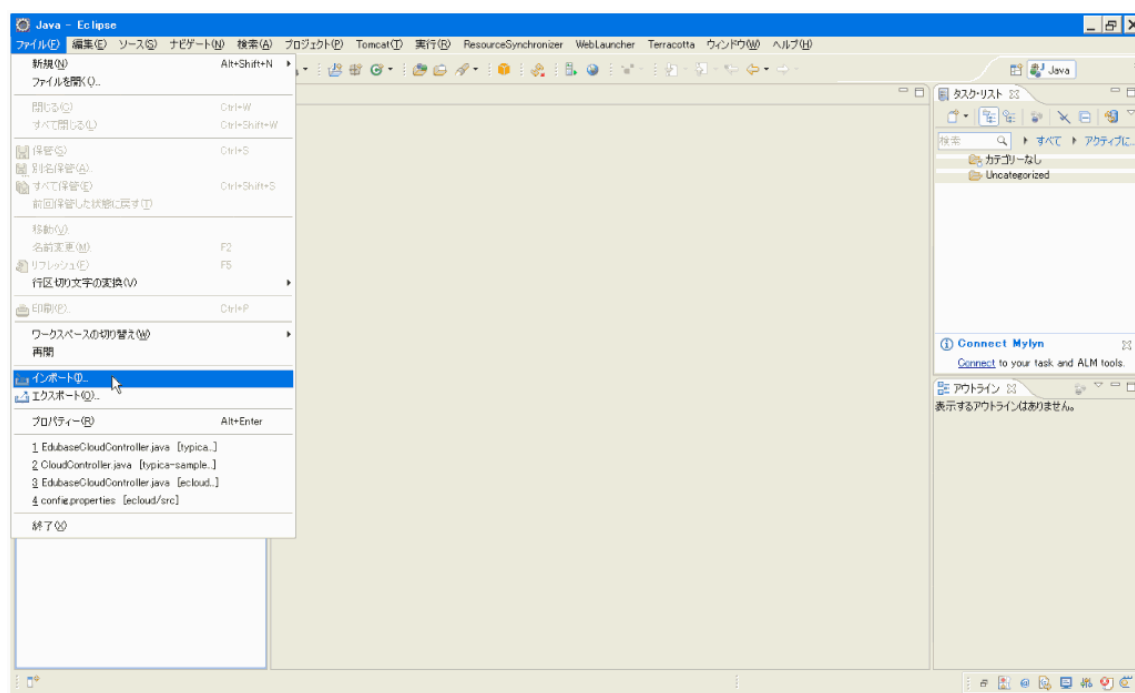


図 8:プロジェクトのインポート

「一般→既存プロジェクトをワークスペースへ」を選択し、「次へ」ボタンをクリックします。(図 9: 既存プロジェクトをワークスペースへ)

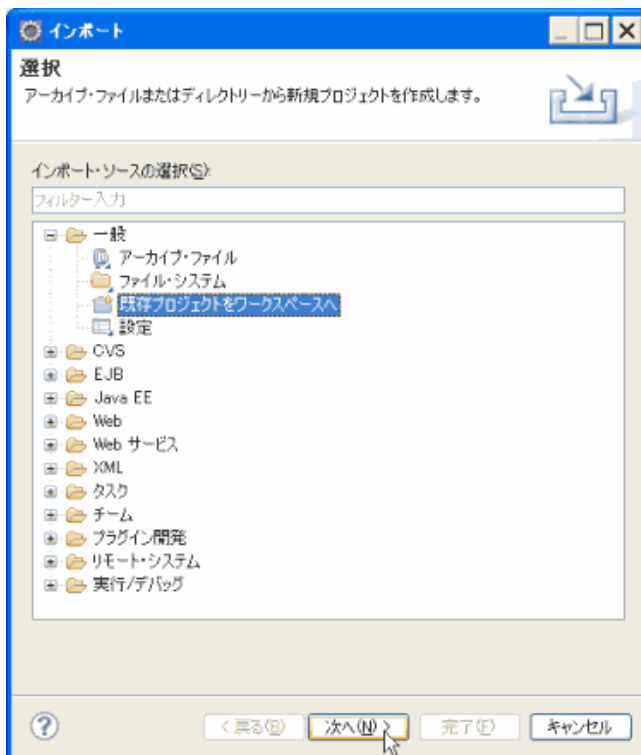


図 9: 既存プロジェクトをワークスペースへ

「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 10: 参照先の選択)

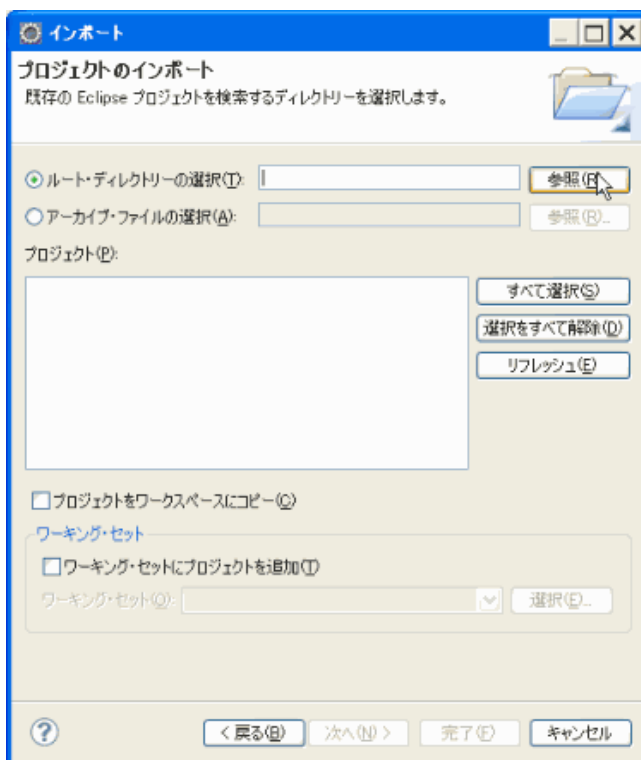


図 10: 参照先の選択

「フォルダの参照」で「U:\usr¥workspace」フォルダの中にある、「eccloud」を選択し、「OK」ボタンを

クリックします。(図 11:フォルダの選択)

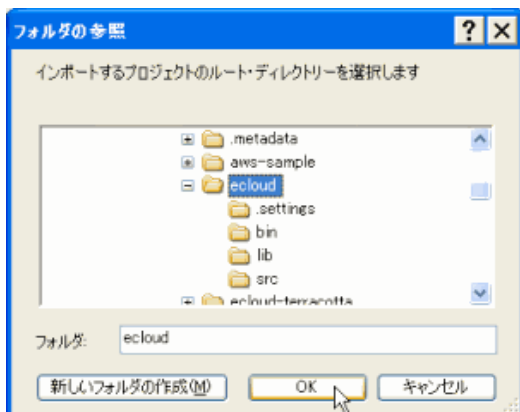


図 11:フォルダの選択

「終了」ボタンをクリックする(図 12:参照の完了)とパッケージ・エクスプローラーに「ecloud」が表示されます。(図 13:プロジェクトのインポート完了)

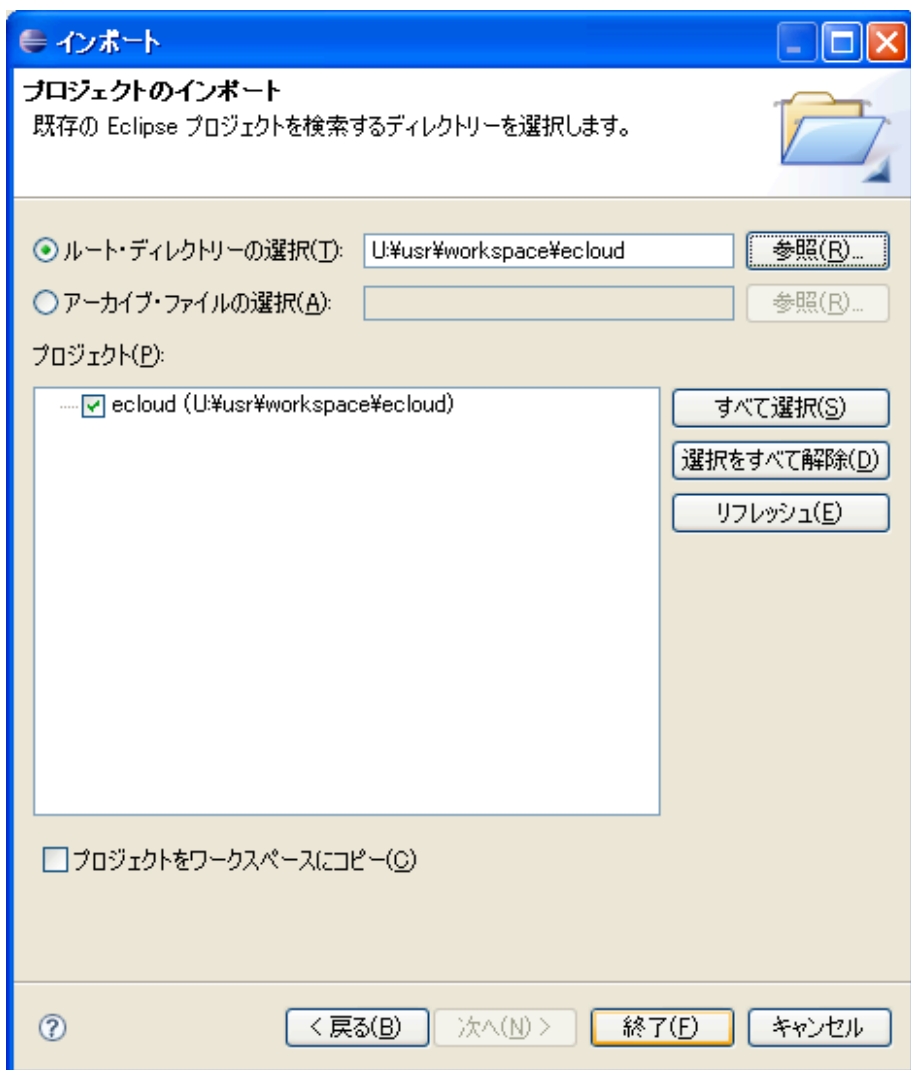


図 12:参照の完了

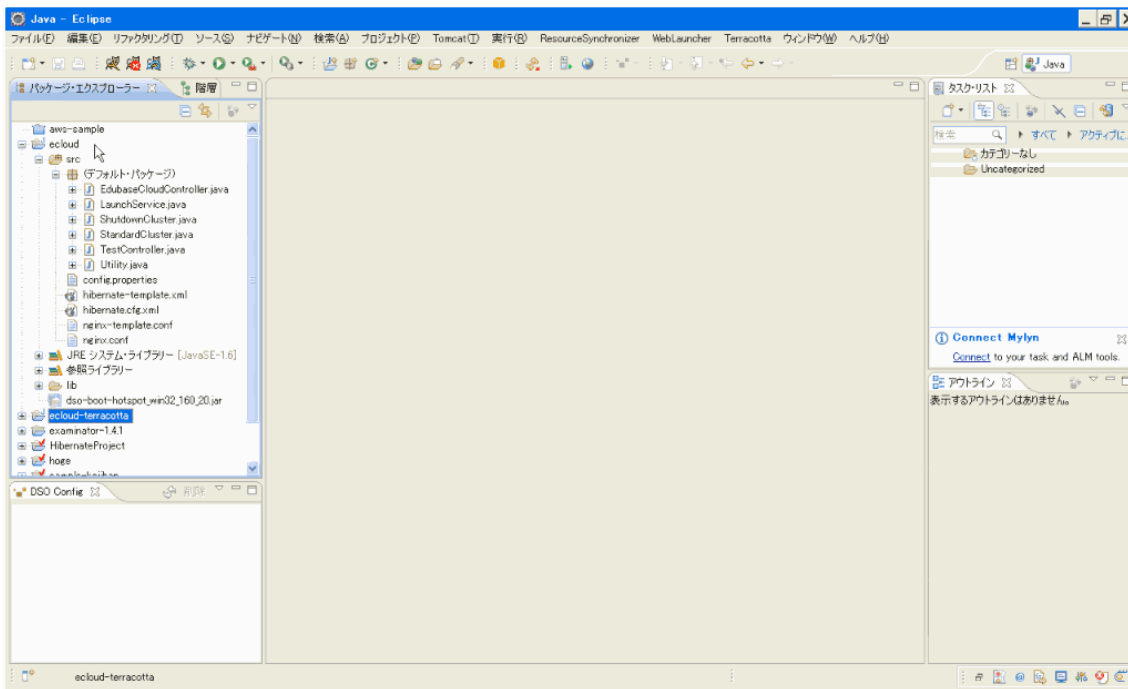


図 13:プロジェクトのインポート完了

スクリプトを起動するために「config.properties」を編集します。
先ほどインポートした「ecloud→src→config.properties」を開き各項目を記述してください。

ソースコード: config.properties

```
# Eucalyptus Credential
secretKey=secretKey
accessKey=accessKey

# Eucalyptus Web API
hostname=hostname
resourcePrefix=/services/Eucalyptus
port=8773

# Instance
keyName=ログインユーザーID
securityGroup=ログインユーザーID

# Machine Images
dbImageId=イメージ ID
appImageId=イメージ ID
webImageId=イメージ ID
lbImageId=イメージ ID

# Configuration files for Process (nginx and Tomcat)
srcDir= U:¥¥usr¥¥workspace¥¥ecloud¥¥src¥¥

# Private Key
keyFile=U:¥¥usr¥¥.ec2¥¥[プロジェクト名]¥¥[hostname]¥¥[ログインユーザーID].pem
```

以下の項目の部分修正します。

- ☐ secretKey
- ☐ accessKey

- ☐ hostname
- ☐ keyName
- ☐ securityGroup
- ☐ keyFile

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 14: クラウドクライアントの設定)

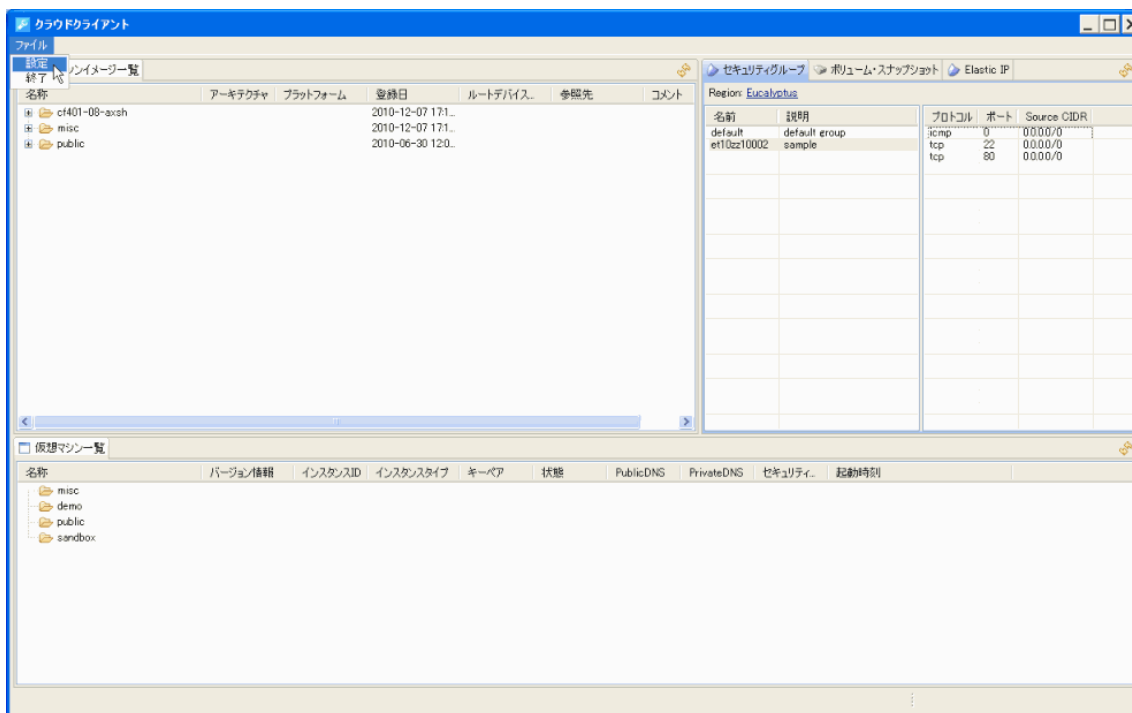


図 14: クラウドクライアントの設定

クラウドクライアントの設定を確認します。(図 15:クラウドクライアントの設定確認)

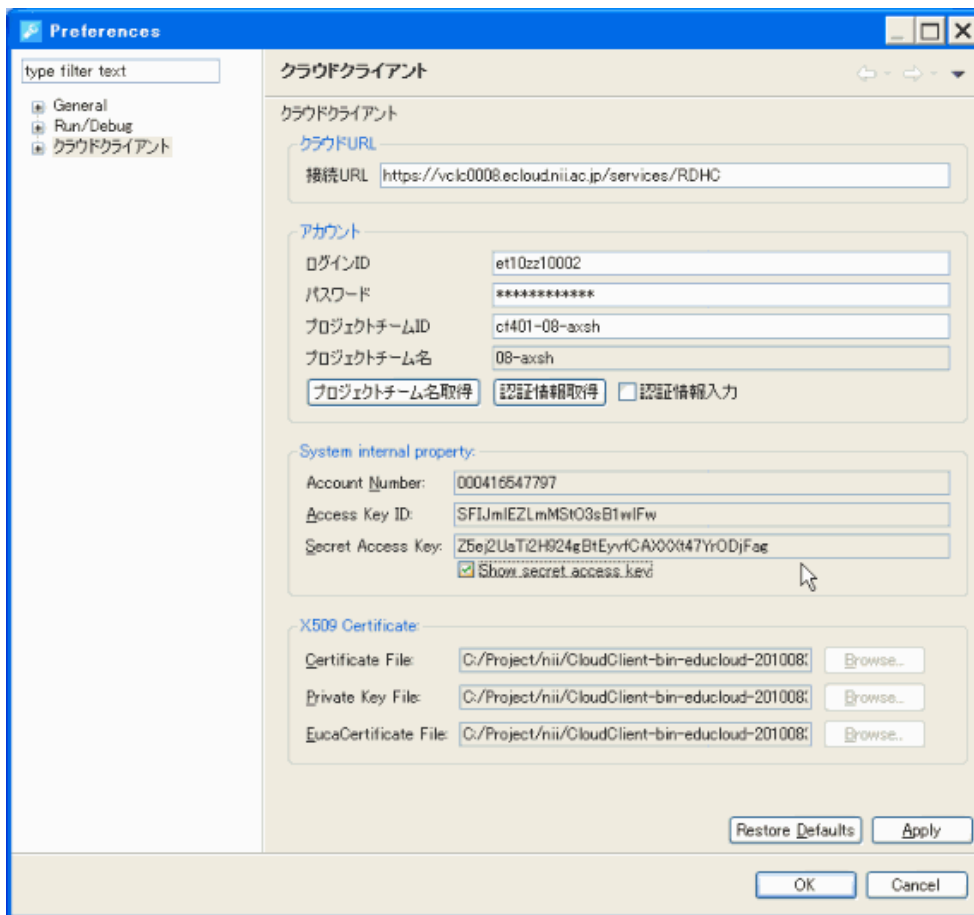


図 15:クラウドクライアントの設定確認

config.properties	クラウドクライアントの Preserceces ダイアログ
secretKey	Secret Access Key
accessKey	Access Key ID
hostname	接続 URL のホスト名(例: vcl0008.eccloud.nii.ac.jp)

「keyname」、「securityGroup」には自分が使用しているログインユーザーID をいれます。

「keyFile」には「U:\usr\ec2\[プロジェクト名]\[hostname]\[ログインユーザーID].pem」を入れます。

1.2.3. 起動スクリプトを実行する

それでは起動スクリプトを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「eccloud→src→jp.ac.nii.sample→LaunchCluster」を右クリックし「実行→Java Application」を実行します。(図 16: 起動スクリプトの実行)

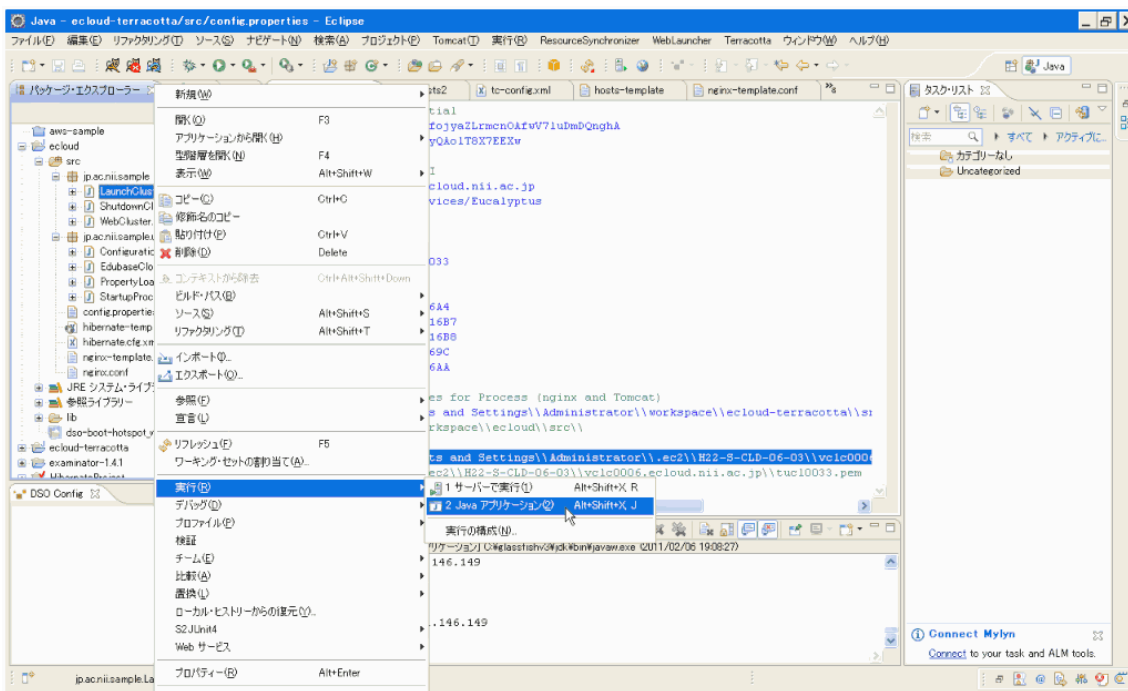


図 16: 起動スクリプトの実行

以下のように Script の実行ログが表示されます。(図 17:実行ログの確認)

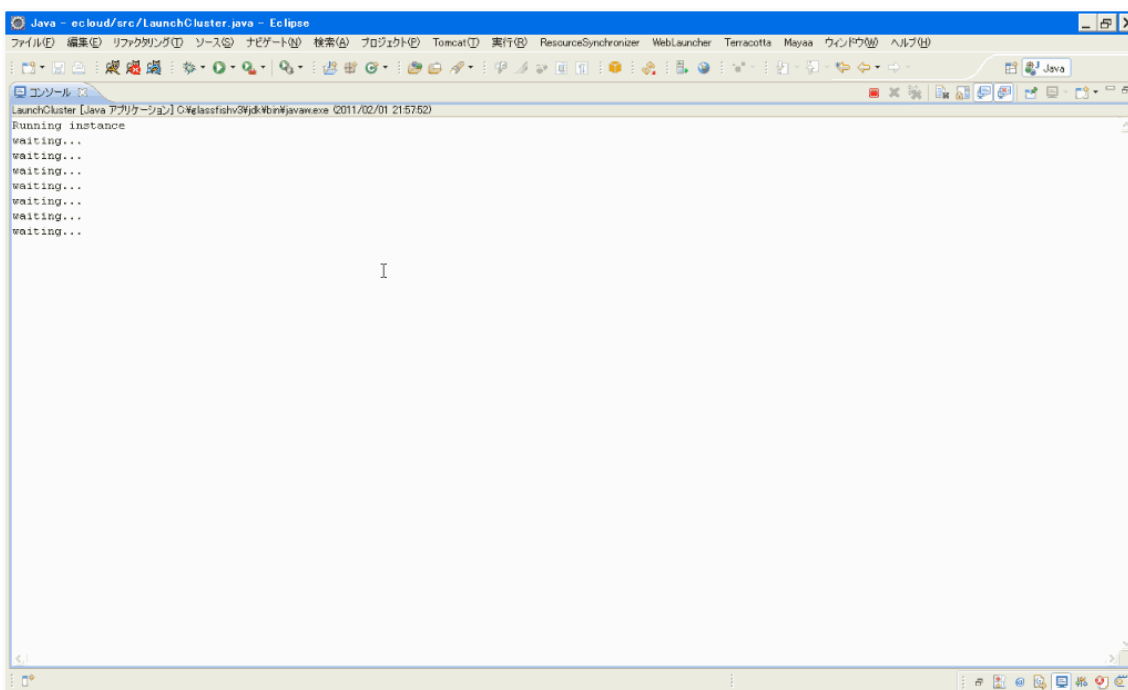


図 17:実行ログの確認

「CloudClient」の仮想マシン一覧の右にある「リフレッシュ」ボタンを押すとサーバーが一台ずつ起動してくるのが確認できます。(図 18:インスタンスの起動確認)
またこの起動処理には時間がかかります。

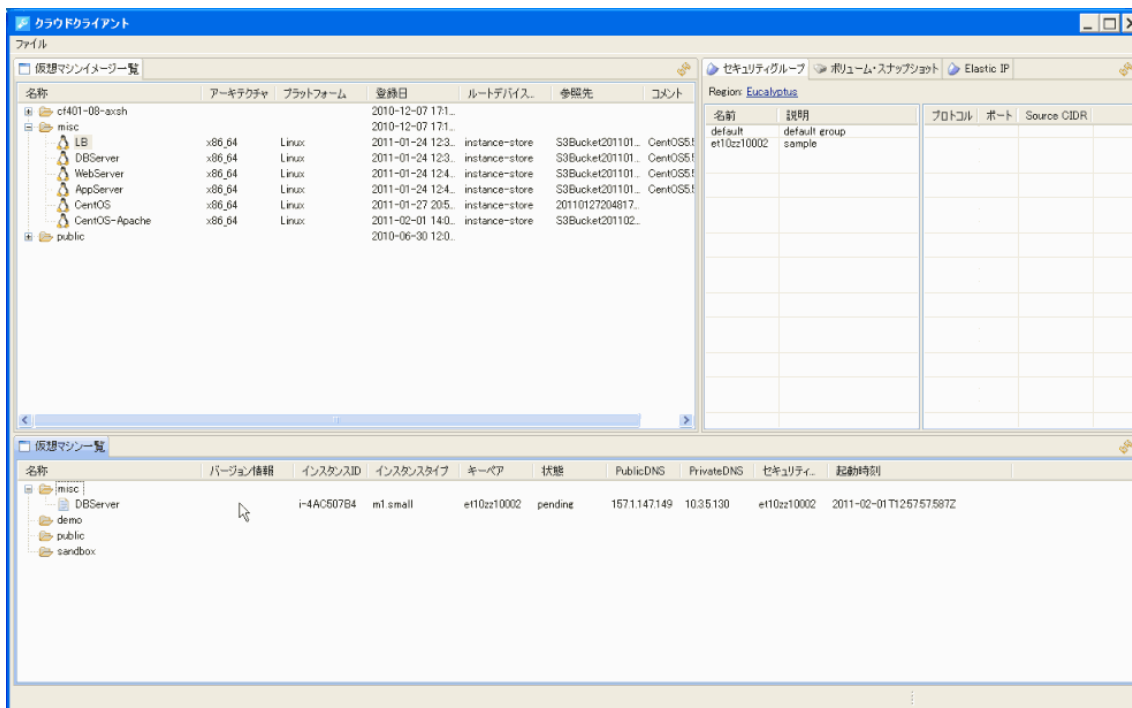


図 18:インスタンスの起動確認

全ての Server の状態が「running」になるのを確認したら動作確認してみましょう。(図 19:インスタンスの起動完了)

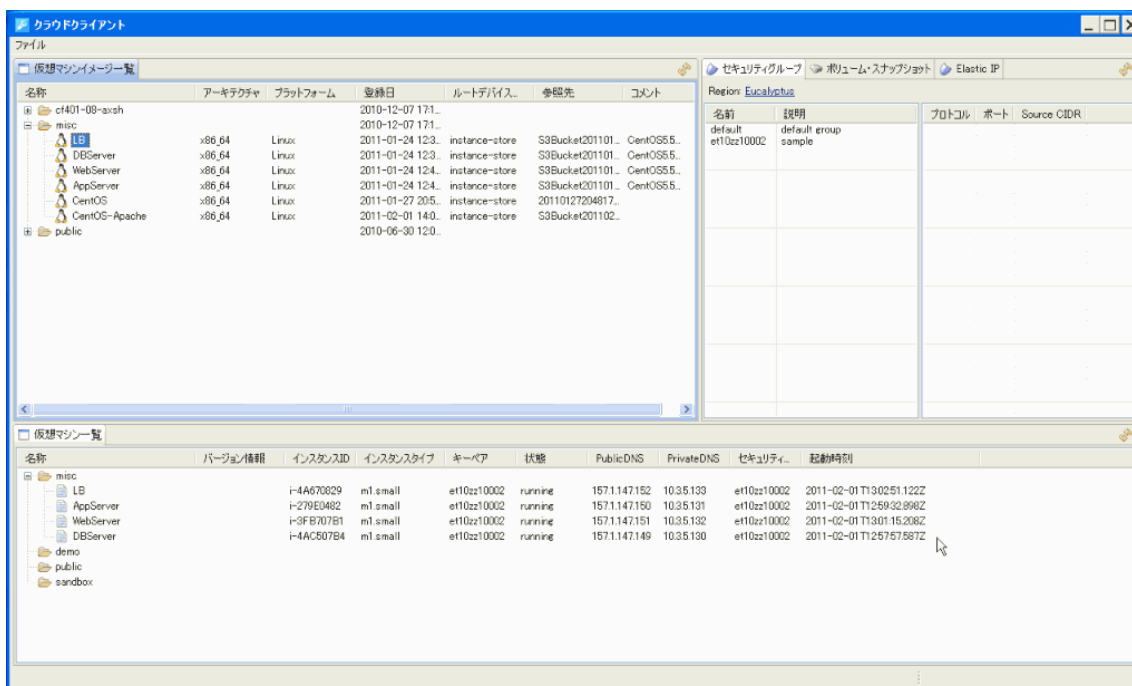


図 19:インスタンスの起動完了

1.2.4. 動作確認をする

1.2.4.1.Web ブラウザから確認する

Web ブラウザを起動して以下の URL を見てください。(図 20:Web ブラウザの表示)

URL: [http://\[LB のパブリック IP\]/sample-kejiban/](http://[LB のパブリック IP]/sample-kejiban/)

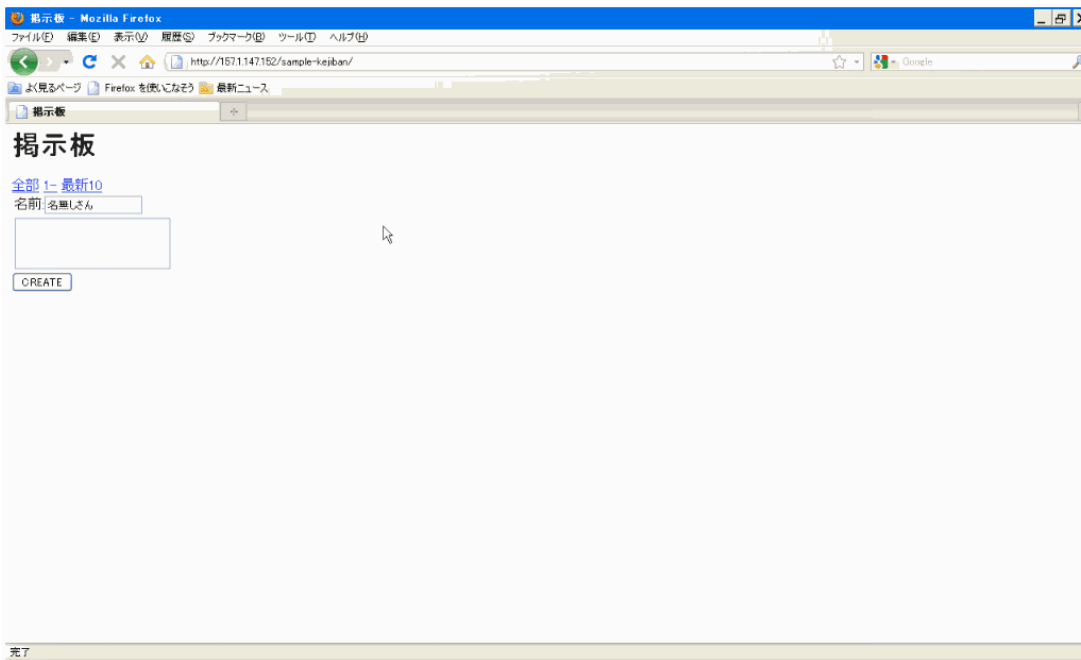


図 20: Web ブラウザの表示

1.2.4.2. データベースを確認する

それでは実際にデータが書き込まれているか確認しましょう。
まず Web ブラウザで「名前」と「コメント」を入力し書き込みを行いましょ。

「CloudClient」の「仮想マシン一覧」から起動している DBServer の「名称」を右クリックし「シェルの起動」を選択します。(図 21:シェルの起動)

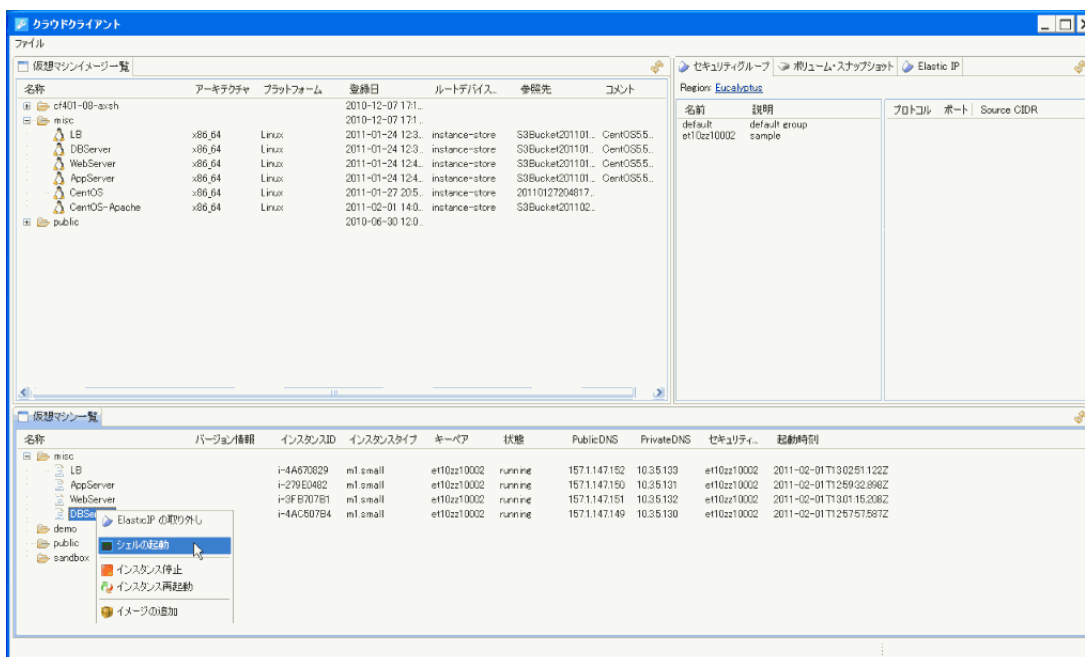


図 21:シェルの起動

以下のコマンドでデータベースの中を見てみましょう。

まずユーザー「postgres」になります。


```
-bash-3.2# su - postgres
-bash-3.2$
```

次に対象のデータベースにログインします。

```
-bash-3.2$ psql demo
psql (8.4.5)
Type "help" for help.

demo=#
```

テーブルの中身を確認します。

```
demo=# select * from res;

 id |   name   | entry   |          date
-----+-----+-----+-----
  1 | nanashisan | hogehoge | 2011-01-17 02:55:39.607
(1 rows)

demo=#
```

レコードが増えていることを確認してください。

1.2.4.3.起動しているマシンを終了する

最後に立ち上げたマシンを停止しましょう。

「ecloud」プロジェクトの「src→jp.ac.nii.sample→ShutdownCluster」を右クリックし「実行→Java Application」を実行します。(図 22:停止スクリプトの実行)

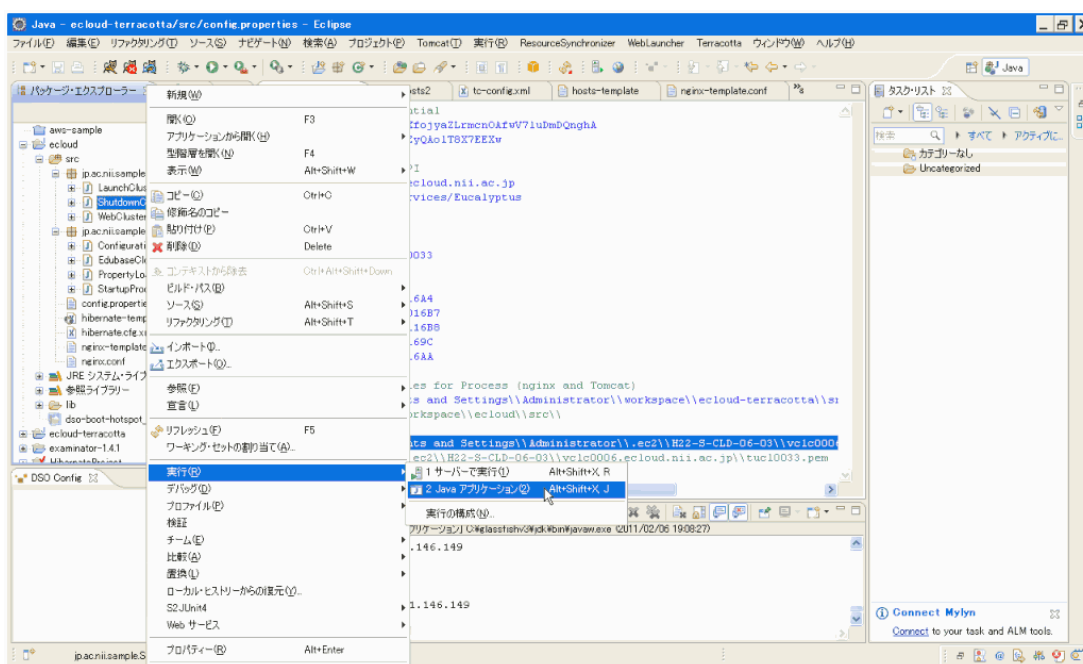


図 22:停止スクリプトの実行

「CloudClient」の仮想マシン一覧の右にある「リフレッシュ」ボタンを押すと全てのサーバーが停止の処理を行っているのが確認できます。(図 23:サーバーの停止)

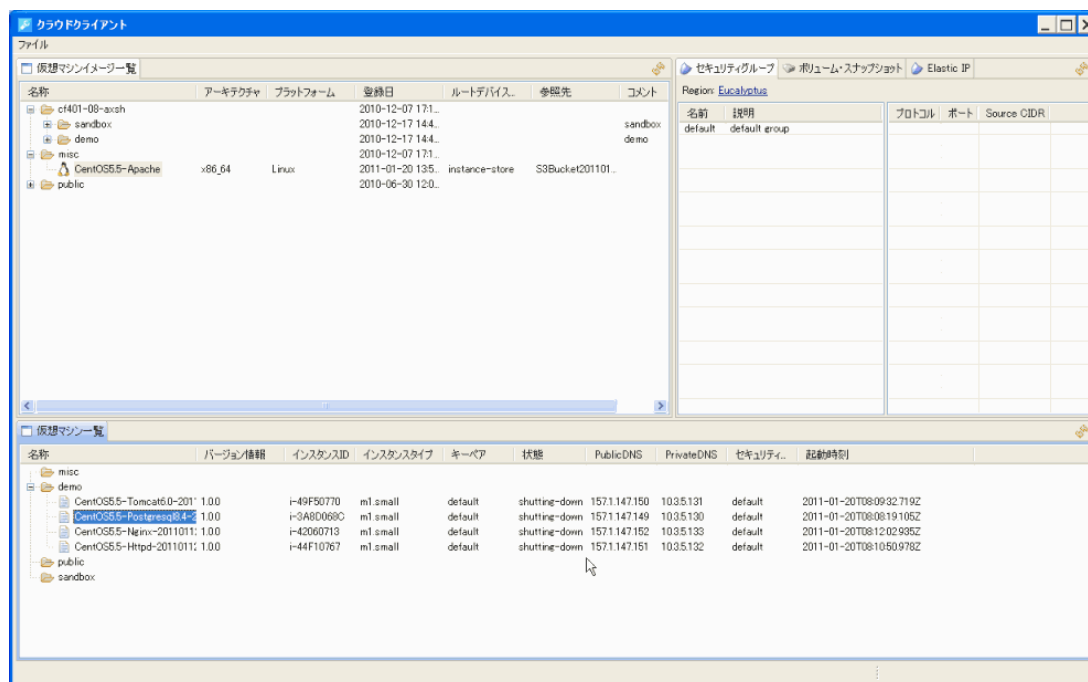


図 23:サーバーの停止

2章 Apache2+Tomcat をスケールアウトする

2.1. スケールアウトするには

先ほど実行したスクリプトは以下のような手順で動きます。(図 24:起動スクリプト実行時の全体構成)

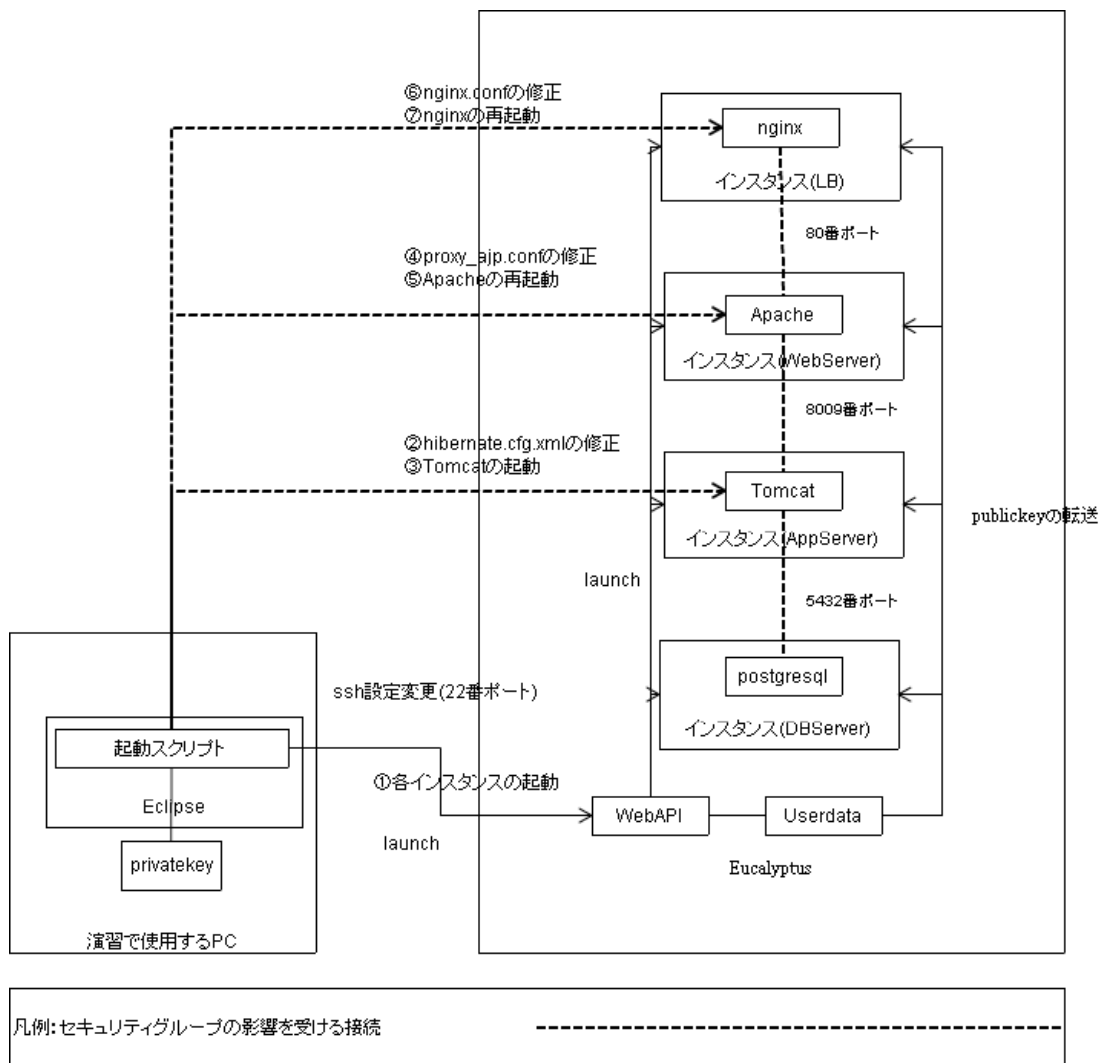


図 24: 起動スクリプト実行時の全体構成

- 各インスタンスを WebAPI 経由で起動する
- hibernate.cfg.xml の編集
- Tomcat の起動
- proxy_ajp.conf の修正
- Apache の再起動
- nginx の修正
- nginx の再起動

各ファイルの場所と変更箇所は以下になります。

hibernate.cfg.xml

ファイルは AppServer の「/opt/sample-kejiban/src/main/webapp/WEB-INF/classes/」にあります。

変更箇所は、以下になります。

```
<property name="hibernate.connection.url">jdbc:postgresql://DBServer のプライベート
IP:5432/demo</property>
```

proxy_ajp.conf

ファイルは WebServer の「/etc/httpd/conf.d/」にあります。

変更箇所は以下になります。

```
ProxyPass /sample-kejiban ajp://AppServer のプライベート IP:8009/sample-kejiban
```

nginx.conf

ファイルは Lb の「/usr/local/nginx/conf/」にあります。

変更箇所は以下になります。

```
http {  
-----略-----  
    upstream backend {  
        server WebServer のプライベート IP;  
    }  
-----略-----  
}
```

各サーバーの起動、再起動方法は以下になります。

Tomcat の起動方法

```
-bash-3.2# /etc/init.d/tomcat start
```

Apache(httpd)の再起動方法

```
-bash-3.2# /etc/init.d/httpd start
```

nginx の再起動方法

```
-bash-3.2# /etc/init.d/nginx start
```

今回作成するスクリプトは以下の手順で実行されます。(図 25:スケールアウトスクリプト実行時の全体構成)

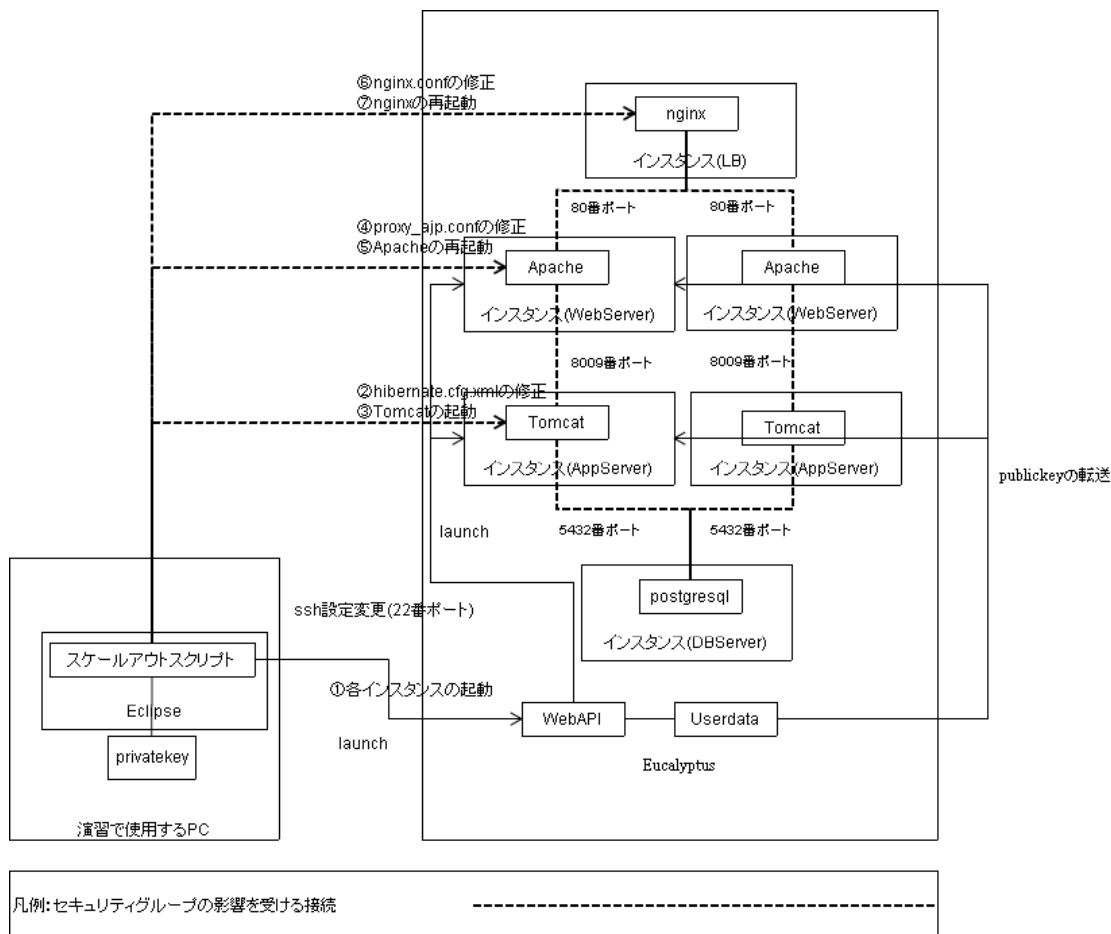


図 25:スケールアウトスクリプト実行時の全体構成

- インスタンス(AppServer)とインスタンス(WebServer)を WebAPI 経由で起動する
- hibernate.cfg.xml の編集
- Tomcat の起動
- proxy_ajp.conf の修正
- Apache の再起動
- nginx の修正
- nginx の再起動

起動スクリプトとの違いは WebAPI 経由で起動するインスタンスは AppServer と WebServer の2つになり、nginx には今回起動したインスタンス(WebServer)だけではなく、既に起動しているインスタンス(WebServer)のプライベート IP をすべて記述することになります。

2.2. 演習のヒント

2.2.1. スケールアウトスクリプトを作成する

本書「2.1スケールアウトするには」と起動スクリプトを参考に、AppServer と WebServer を増やすスケールアウトスクリプトを作成してみましょう。

2.2.2. 動作確認をする

2.2.2.1.Web ブラウザから確認する

本書「1.2.4.1Web ブラウザから確認する」を参考に動作確認を行ってください。

2.2.2.2.データベースを確認する

本書「1.2.4.2データベースを確認する」を参考に動作確認を行ってください。

2.2.2.3.AppServer の log を確認する

「CloudClient」の「仮想マシン一覧」から起動している AppServer の「名称」を右クリックし「シェルの起動」を選択します。

下記コマンドを実行しログを確認します。

```
-bash-3.2# tail -F /opt/tomcat/logs/catarlina.out
```

Web ブラウザをリロードして2台の APPServer に交互にアクセスがくるか確認する。

2.2.2.4.起動しているマシンを終了する

本書「1.2.4.3起動しているマシンを終了する」を参考に動作確認を行ってください。

3章 【参考】 手動でスケールアウトする場合

3.1. 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→クラウドクライアント」を起動し、仮想マシンイメージ一覧から DBServer と APPServer を起動してください。

3.2. AppServer の設定をする

3.2.1. サンプルプログラムを編集する

3.2.1.1. サンプルプログラムをインポートする

本書「1.2.2eclipse 上で起動スクリプトを動かすための設定をする」を参考に「sample-kejiban」プロジェクトをインポートします。

3.2.1.2. hibernate.cfg.xml の編集

「sample-kejiban→src/main/java→hibernate.cfg.xml」を開きます。
「property」の「hibernate.connection.url」を下記のように変更します。

```
<property name="hibernate.connection.url">jdbc:postgresql://DBServer のプライベート  
IP:5432/demo</property>
```

3.2.2. Pagent を起動し鍵を登録する

ここではスクリプトで各インスタンスに接続するために必要な鍵の登録を行います。「C:¥Program Files¥PuTTY」フォルダの中にある「pagent.exe」を起動します。(図 26:Pagent の場所)

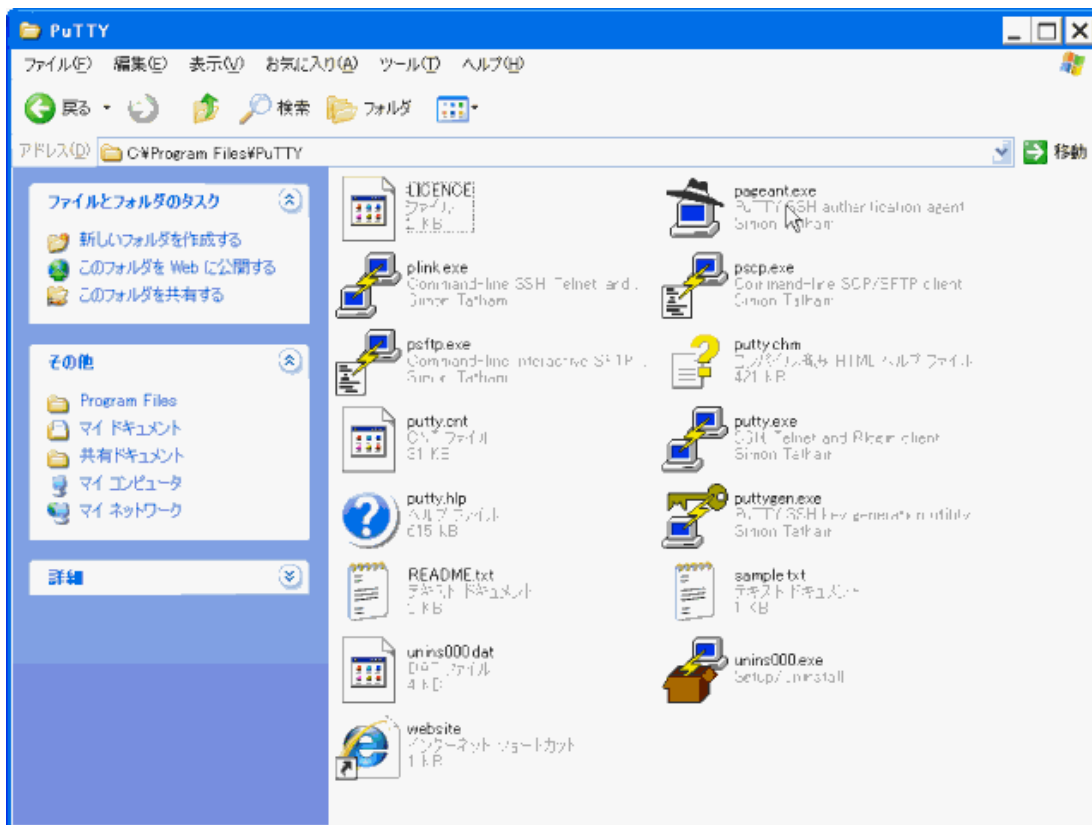


図 26:Pageant の場所

起動すると Windows のタスクバーにアイコンが出ます。

次に Windows のタスクバーに表示されている pageant のアイコンをダブルクリックして pageant の Key List を表示します。

新規に鍵を登録するために「Add Key」ボタンをクリックします。(図 27:秘密鍵のリスト)

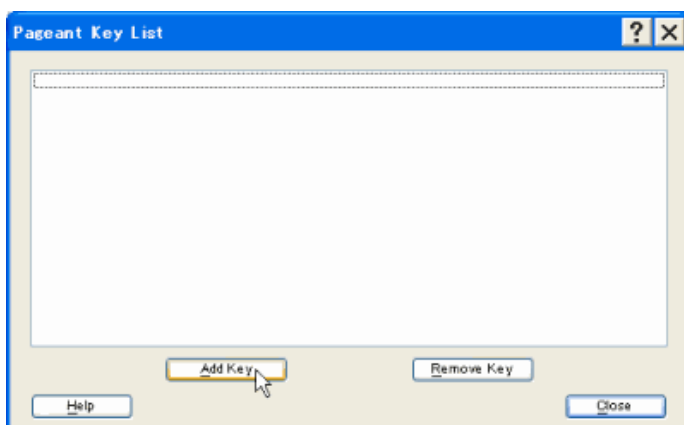


図 27:秘密鍵のリスト

前回作成した「ログイン ID.ppk」という名前の ppk ファイルを選択して「開く」ボタンをクリックします。(図 28:秘密鍵の登録)

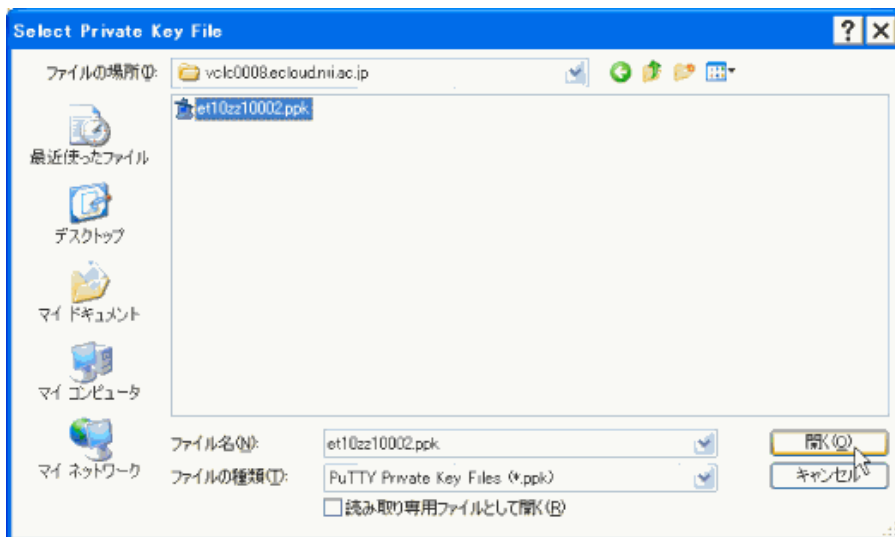


図 28:秘密鍵の登録

鍵が登録されていることを確認します。(図 29:秘密鍵の登録確認)

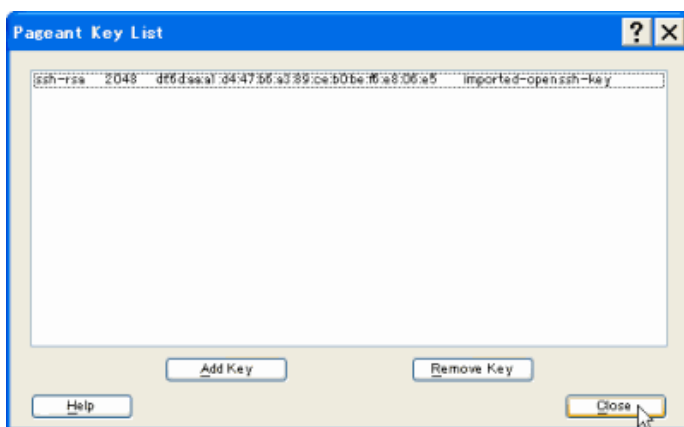


図 29:秘密鍵の登録確認

3.2.3. AppServer にサンプルプログラムを配置する

Windows の「スタート→すべてのプログラム→WinSCP→WinSCP」を起動し、「ホスト名」に「AppServer のパブリック IP」、「ユーザ名」に「root」を入力して、ログインします。(図 30:AppServer へログイン)

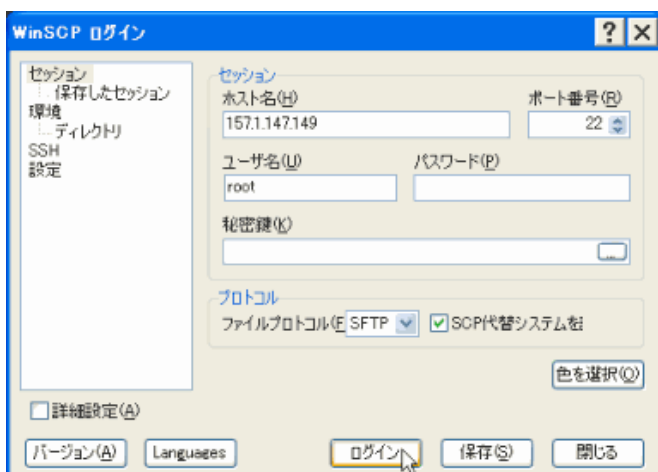


図 30:AppServer へログイン

先ほど編集した「sample-kejiban」プロジェクトを「/opt/」ディレクトリに転送します。
次に Tomcat の設定をします。

「/opt/sample-kejiban/src/main/webapp」フォルダを「/opt/tomcat/webapps/」の中に「sample-kejiban」という名前でコピーしてください。

```
-bash-3.2# cp -r sample-kejiban/src/main/webapp /opt/tomcat/webapps/  
-bash-3.2# mv /opt/tomcat/webapps/webapp /opt/tomcat/webapps/sample-kejiban
```

3.2.4. Tomcat を起動する

以下のコマンドで Tomcat を起動してください。

```
-bash-3.2# /etc/init.d/tomcat start  
  
Starting tomcat:  
Using CATALINA_BASE:   /opt/tomcat  
Using CATALINA_HOME:   /opt/tomcat  
Using CATALINA_TMPDIR: /opt/tomcat/temp  
Using JRE_HOME:        /usr/java/default  
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar  
  
-bash-3.2#
```

3.3. WebServer の設定をする

3.3.1. proxy_ajp.conf の編集

「/etc/httpd/conf.d/proxy_ajp.conf」ファイルに AppServer のプライベート IP を追加します。

```
-bash-3.2# vi /etc/httpd/conf.d/proxy_ajp.conf  
  
ProxyPass /sample-kejiban ajp://AppServer のプライベート IP:8009/sample-kejiban
```

3.3.2. Apache(httpd)を再起動する

以下のコマンドで httpd を再起動します。

```
-bash-3.2# vi /etc/init.d/httpd restart
```

3.4. LB の設定をする

3.4.1. nginx.conf の編集

「/usr/local/nginx/conf/nginx.conf」ファイルに WebServer のプライベート IP を追加します。

```
-bash-3.2# vi /usr/local/nginx/conf/nginx.conf
```

```
http {
```

```
-----略-----
```

```
    upstream backend {
```

```
        server WebServer のプライベート IP;
```

```
    }
```

```
-----略-----
```

```
}
```

3.4.2. nginx を再起動する

以下のコマンドで nginx を再起動します。

```
-bash-3.2# vi /etc/init.d/nginx restart
```

3.5. 動作確認する

3.5.1. Web ブラウザから確認する

本書「1.2.4.1 Web ブラウザから確認する」を参考に動作確認を行ってください。

3.5.2. データベースを確認する

本書「1.2.4.2 データベースを確認する」を参考に動作確認を行ってください。

3.5.3. AppServer の log を確認する

本書「2.2.2.3 AppServer の log を確認する」を参考に動作確認を行ってください。

3.5.4. 起動しているマシンを終了する。

本書「1.2.4.3 起動しているマシンを終了する」を参考に動作確認を行ってください。