

PuppetMCollective による自動化 参考資料

目次

1 はじめに.....	3
2 前提条件.....	3
3 PUPPET+MCCOLLECTIVE による自動化構築手順	3
3.1. CONFIG ファイルの設定	3
3.1.1. アクセスキーとシークレットキーの設定をする	3
3.1.2. セキュリティグループの設定をする.....	4
3.1.3. キーペアの設定をする.....	4
3.1.4. イメージ ID を確認する.....	5
3.2. LB、DB、MAIL サーバを構築する	6
3.2.1. LB、DB、Mail サーバを起動する.....	6
3.2.2. デプロイ、LB、DB、Mail サーバの IP アドレスを取得する.....	7
3.2.3. MCollective と Puppet の設定をする	9
3.2.4. Puppet の certificate の設定をする	11
3.3. デプロイサーバの PUPPET で MCCOLLECTIVESERVER を設定する	11
3.4. WEB サーバを構築する.....	12
3.4.1. Web サーバを起動する	12
3.4.2. Web サーバの起動設定をする.....	13
3.4.3. Puppet の certificate の設定をする	15
3.5. PUPPET AGENT をロックする	16
3.6. LB サーバへ WEB サーバの IP アドレスを追加する	16
3.7. WEB ブラウザからの確認.....	17
3.7.1. browser サーバにログイン.....	17
3.7.2. hosts の設定	18
3.7.3. 掲示板アプリケーションの確認	19

図の索引

図 1 クラウドクライアント	4
図 2 VNC の接続先サーバの設定	18
図 3 VNC のパスワード入力	18
図 4 スクリーンサーバ	18

1はじめに

この手順書は edubaseCloud 上に Puppet と MCollective を使いシステムの自動化をする手順書である。

2前提条件

- ・デプロイサーバが起動していること。
- ・デプロイサーバ上で war の export が終わっていること。
(「01_掲示板サービスの本番環境構築」の P23「3.5.deploy サーバの設定」の最後まで完了している事)

3Puppet+Mcollective による自動化構築手順

3.1.Config ファイルの設定

LB、WEB、DB、Mail サーバを起動するためのアクセスキー、シークレットキー、セキュリティグループ、キーペアの設定をおこなう。

Config ディレクトリに移動する。

```
deploy# cd /root/work/deploy/config
```

デプロイサーバにある設定ファイルを修正する。

```
deploy# vi ./config.yml
```

以下は config.yml の設定の説明である。

config.yml を修正するとき、:(コロン)の後にスペースを一つ入れて入力してください。

3.1.1.アクセスキーとシークレットキーの設定をする

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」を起動する。
次に「ファイル→設定」をクリックし、CloudClient を表示する。

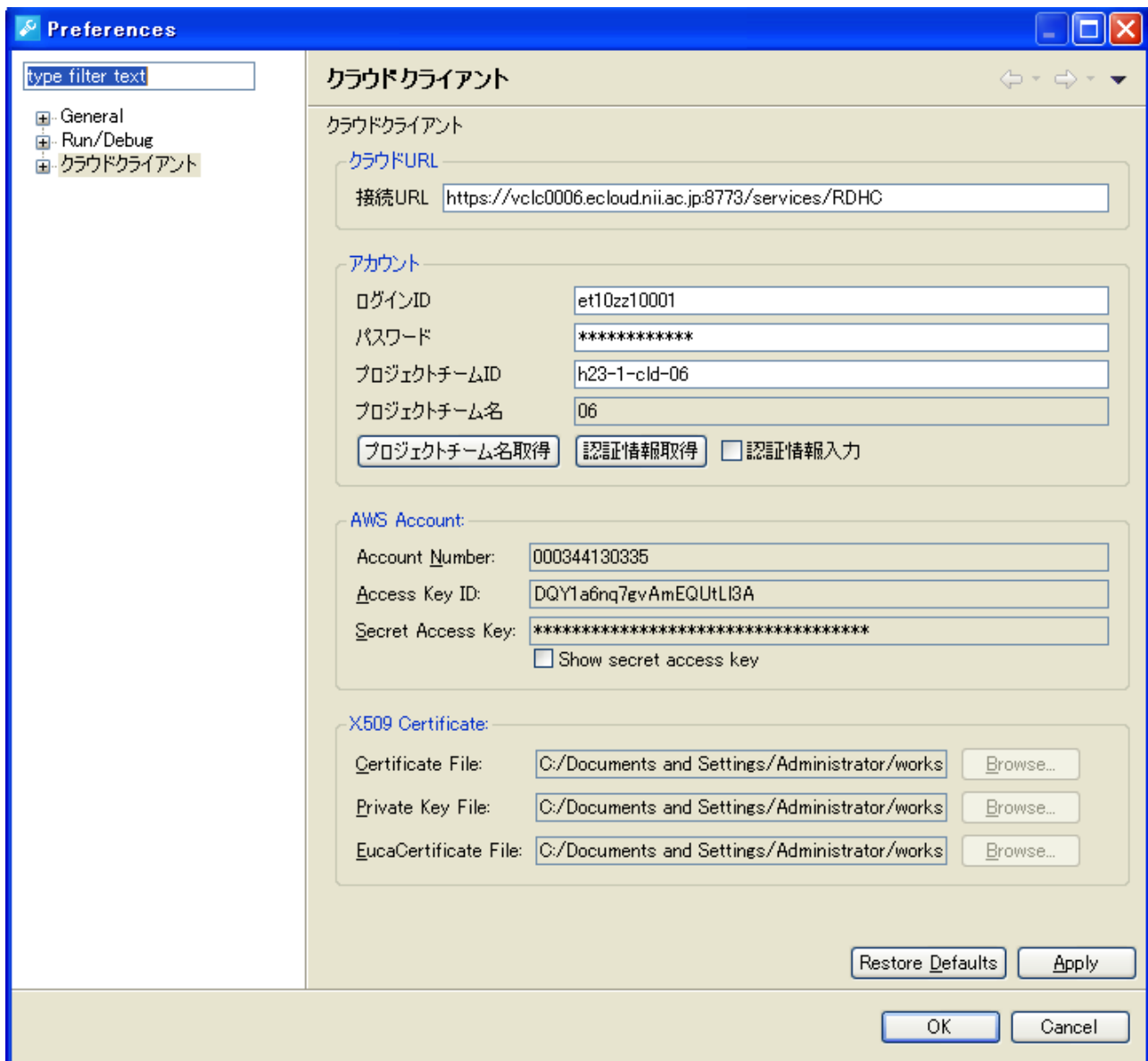


図 1 クラウドクライアント

AWS Account の Access Key ID を access_key に入力をする。

access_key:

AWS Account の Secret Access Key を secret_key に入力をする。

secret_key:

3.1.2. セキュリティグループの設定をする

事前に作成したセキュリティグループ ID (group_id) の変更をする。

group_id: tuc|****

3.1.3. キーペアの設定をする

事前に作成したキーペアの名前の変更をする。

key_name: tuc|****

3.1.4.イメージ ID を確認する

CloudClient のイメージ ID と config.yml 内のイメージ ID が一致することを確認する

```
images:  
  lb: emi-*****  
  web: emi-*****  
  db: emi-*****  
  mail: emi-*****  
  monitor: emi-*****
```

3.2.LB、DB、Mail サーバを構築する

3.2.1.LB, DB, Mail サーバを起動する

以下のコマンドをデプロイサーバで実行する。

インスタンスを起動する。

deploy ディレクトリに移動する

```
deploy# cd /root/work/deploy
```

lb,db,mail サーバを起動する。出力結果は同じである。

lb サーバを起動する。

```
deploy# ./bin/deploy instances launch lb
```

db サーバを起動する。

```
deploy# ./bin/deploy instances launch db
```

mail サーバを起動する。

```
deploy# ./bin/deploy instances launch mail
```

実行結果

```
I, [2012-03-24T12:18:06.406861 #1139] INFO -- : New RightAws::Ec2 using single-  
threaded mode  
  
I, [2012-03-24T12:18:06.407027 #1139] INFO -- : Launching instance of image  
emi-5EA60645 for DQY1a6nq7gvAmEQUtLI3A, key: tucI0003, groups: tucI0003  
  
I, [2012-03-24T12:18:06.407617 #1139] INFO -- : Opening new HTTPS connection to  
vcIc0006.ecIoud.nii.ac.jp:8773  
  
i-*****
```

実行後のログの最後にインスタンス **ID** が出力されるので、lb、db、mail の instance_id をメモしておく。次の設定で利用する。

Instance_id のメモ例

```
lb:  
i-*****  
db:  
i-*****  
mail:  
i-*****
```

3.2.2. デプロイ、LB、DB、Mail サーバの IP アドレスを取得する

この後、設定を行う上で必要となるデプロイ、LB、DB、Mail サーバの IP アドレスを取得する。

deploy ディレクトリに移動する。

```
deploy# cd /root/work/deploy
```

deploy サーバの IP アドレスを取得する。

```
deploy# /sbin/ip route get 8.8.8.8
```

実行結果

```
8.8.8.8 via 10.3.5.129 dev eth0 src 10.3.5.130  
cache mtu 1500 advmss 1460 hoplimit 64
```

lb サーバの IP アドレスを取得する。

```
deploy# ./bin/deploy instances describe --instanceids={lb サーバの instance_id}  
--key=ipaddr
```

実行結果

```
I, [2012-03-24T12:36:53.317214 #1201] INFO -- : New RightAws::Ec2 using single-  
threaded mode  
I, [2012-03-24T12:36:53.327502 #1201] INFO -- : Opening new HTTPS connection to  
vclc0006.eccloud.nii.ac.jp:8773  
["10.3.4.131"]
```

db サーバの IP アドレスを取得する。

```
deploy# ./bin/deploy instances describe --instanceids={db サーバの instance_id}  
--key=ipaddr
```

実行結果

```
I, [2012-03-24T12:36:53.317214 #1201] INFO -- : New RightAws::Ec2 using single-  
threaded mode  
I, [2012-03-24T12:36:53.327502 #1201] INFO -- : Opening new HTTPS connection to  
vclc0006.eccloud.nii.ac.jp:8773  
["10.3.4.132"]
```

mail サーバの IP アドレスを取得する。

```
deploy# ./bin/deploy instances describe --instanceids={mail サーバの instance_id}  
--key=ipaddr
```

実行結果

```
I, [2012-03-24T12:36:53.317214 #1201] INFO -- : New RightAws::Ec2 using single-  
threaded mode  
  
I, [2012-03-24T12:36:53.327502 #1201] INFO -- : Opening new HTTPS connection to  
vc1c0006.eccloud.nii.ac.jp:8773  
  
["10.3.4.133"]
```

Instance_id のメモに各サーバの IP アドレスを追記する。

```
deploy:  
10.3.4.130  
lb:  
i-*****  
10.3.4.131  
db:  
i-*****  
10.3.4.133  
mail:  
i-*****  
10.3.4.132
```


3.2.3.MCollective と Puppet の設定をする

db, lb, mail サーバに対して Puppet と MCollective の設定を行う。

CloudClient で lb, db, mail サーバの状態が running になっていることを確認してください。

各サーバの hosts ファイルを用いて、deploy.nii.localdomain を今回使用しているデプロイサーバと紐づけて、Puppet Client と MCollective Server を起動する。

deploy ディレクトリに移動する。

```
deploy# cd /root/work/deploy
```

lb サーバの設定を行う。

lb サーバの hosts にデプロイサーバの設定を追記する。(実行結果なし)

```
deploy# ./ bin/deploy ssh exec {lb サーバの IP アドレス} 'echo "{デプロイサーバの IP アドレス} deploy.nii.localdomain" >> /etc/hosts'
```

lb サーバで puppet を起動する

```
deploy# ./ bin/deploy ssh exec {lb サーバの IP アドレス} "/etc/init.d/puppet start"
```

実行結果

```
* Starting puppet agent
debug: Failed to load library 'selinux' for feature 'selinux'
...
... #Puppet の Debug メッセージが表示されます
...
debug: Finishing transaction 70328380861720
... done.
```

lb サーバで mcollective を起動する

```
deploy# ./ bin/deploy ssh exec {lb サーバの IP アドレス} "/etc/init.d/mcollective start"
```

実行結果

```
Starting mcollective: *
```

同様に db サーバの設定を行う。

db サーバの hosts にデプロイサーバの設定を追記する。(実行結果なし)

```
deploy# ./ bin/deploy ssh exec {db サーバの IP アドレス} 'echo "{デプロイサーバの IP アドレス} deploy.nii.localdomain" >> /etc/hosts'
```

db サーバで puppet を起動する

```
deploy# ./ bin/deploy ssh exec {db サーバの IP アドレス} "/etc/init.d/puppet start"
```

実行結果

```
* Starting puppet agent
debug: Failed to load library 'selinux' for feature 'selinux'
...
... #Puppet の Debug メッセージが表示されます
...
debug: Finishing transaction 70328380861720
...done.
```

db サーバで mcollective を起動する

```
deploy# ./ bin/deploy ssh exec {db サーバの IP アドレス} "/etc/init.d/mcollective
start"
```

実行結果

```
Starting mcollective: *
```

同様に mail サーバの設定を行う。

mail サーバの hosts にデプロイサーバの設定を追記する。(実行結果なし)

```
deploy# ./ bin/deploy ssh exec {mail サーバの IP アドレス} 'echo "{デプロイサーバ
の IP アドレス} deploy.nii.localdomain" >> /etc/hosts'
```

mail サーバで puppet を起動する

```
deploy# ./ bin/deploy ssh exec {mail サーバの IP アドレス} "/etc/init.d/puppet
start"
```

実行結果

```
* Starting puppet agent
debug: Failed to load library 'selinux' for feature 'selinux'
...
... #Puppet の Debug メッセージが表示されます
...
debug: Finishing transaction 70328380861720
...done.
```

mail サーバで mcollective を起動する

```
deploy# ./ bin/deploy ssh exec {mail サーバの IP アドレス} "/etc/init.d/mcollective
start"
```

実行結果

```
Starting mcollective: *
```

3.2.4.Puppet の certificate の設定をする

デプロイサーバの上で起動している PuppetMaster と lb、db、mail の上で起動している PuppetAgent との通信をするために認証を行う。

puppetca コマンドを実行して、puppet-master と puppet-agent の通信の許可をする。

puppet-master に接続している puppet-agent の一覧を表示する。

```
deploy# puppetca -l
```

実行結果

```
db.nii.localdomain (42:E5:19:4D:AF:3D:DD:83:6C:91:C0:0C:90:6A:B1:32)
lb.nii.localdomain (EF:11:73:71:83:7B:88:52:7D:36:62:75:4B:81:F1:0F)
mail.nii.localdomain (1A:99:75:1F:35:DF:90:67:4A:32:63:F4:F3:17:25:00)
```

puppetmaster と puppet-agent の間の通信を許可する。

```
deploy# puppetca -s --all
```

実行結果

```
notice: Signed certificate request for db.nii.localdomain
notice: Removing file Puppet::SSL::CertificateRequest db.nii.localdomain at
'/var/lib/puppet/ssl/ca/requests/db.nii.localdomain.pem'

notice: Signed certificate request for lb.nii.localdomain

notice: Removing file Puppet::SSL::CertificateRequest lb.nii.localdomain at
'/var/lib/puppet/ssl/ca/requests/lb.nii.localdomain.pem'

notice: Signed certificate request for mail.nii.localdomain

notice: Removing file Puppet::SSL::CertificateRequest mail.nii.localdomain at
'/var/lib/puppet/ssl/ca/requests/mail.nii.localdomain.pem'
```

3.3.デプロイサーバの Puppet で MCollectiveServer を設定する

デプロイサーバ上で Puppet と Mcollective のサービスを起動する。

PuppetServer を起動する。

```
deploy# service puppet start
```

実行結果

```
*Starting puppet agent
debug: Failed to load library 'selinux' for feature 'selinux'
debug: Puppet::Type::User::ProviderUser_role_add: file rolemod does not exist
debug: Puppet::Type::User::ProviderDirectoryservice: file /usr/bin/dscl does not exist
```

```
debug: Puppet::Type::User::ProviderPw: file pw does not exist
debug: Failed to load library 'ldap' for feature 'ldap'
~~~~~中略~~~~~
debug: /File[/var/lib/puppet/clientbucket]: The container
/var/lib/puppet/clientbucket will propagate my refresh event
debug: /File[/var/lib/puppet/client_data]/ensure: created
debug: /File[/var/lib/puppet/client_data]: The container
/var/lib/puppet/client_data will propagate my refresh event
debug: Finishing transaction 70070770287880
```

MCollectiveServer を起動する。

```
deploy# service mcollective start
```

実行結果

```
Starting mcollective: *

[ OK ]
```

3.4.Web サーバを構築する

掲示板サービスを Web サーバ上で実行するための環境を構築する。

3.4.1.Web サーバを起動する

以下のコマンドをデプロイサーバで実行し、Web サーバを起動する。

deploy ディレクトリへ移動する。

```
deploy# cd /root/work/deploy
```

web サーバを起動する。

```
deploy# ./bin/deploy instances launch web
```

実行結果

```
I, [2012-03-25T13:17:08.830560 #1147] INFO -- : New RightAws::Ec2 using single-
threaded mode
I, [2012-03-25T13:17:08.830722 #1147] INFO -- : Launching instance of image
emi-1D751460 for cVy1n9VFKkGhICgaMxIWhA, key: tucl0002, groups: tucl0002
I, [2012-03-25T13:17:08.831315 #1147] INFO -- : Opening new HTTPS connection to
vclc0006.eccloud.nii.ac.jp:8773
i-*****
```

web サーバの IP アドレスを取得する。

```
deploy# ./bin/deploy instances describe --instanceids={web サーバの instance_id}  
--key=ipaddr
```

実行結果

```
I, [2012-03-24T12:36:53.317214 #1201] INFO -- : New RightAws::Ec2 using single-  
threaded mode  
I, [2012-03-24T12:36:53.327502 #1201] INFO -- : Opening new HTTPS connection to  
vclc0006.ecloud.nii.ac.jp:8773  
["10.3.4.134"]
```

起動時のログから instance_id を、更に instance_id から IP アドレスが出力されるので、メモに追記しておく。次の設定で利用する。

instance_id と IP アドレスをメモする

```
web:  
i-*****  
10.3.4.134
```

3.4.2.Web サーバの起動設定をする

web サーバに対して Puppet と MCollective の設定を行う。

CloudClient で web サーバの状態が running になっていることを確認する。

Web サーバ上で動く掲示板サービスでは mail と db サーバを利用しているため、接続先の **IP アドレス**を取得する。

以下のコマンドをデプロイサーバ上で実行する。

mail サーバの ip アドレスを取得する

```
deploy# mco facts ipaddress -F fqdn=/^mail/ -v
```

実行結果

```
Report for fact: ipaddress  
  
10.3.4.133 found 1 times  
  
mail.nii.localdomain  
  
---- rpc stats ----  
Nodes: 1 / 1  
Pass / Fail: 0 / 0  
Start Time: Sun Mar 25 13:18:01 +0000 2012  
Discovery Time: 2002.81ms  
Agent Time: 47.08ms  
Total Time: 2049.89ms
```

db サーバの ip アドレスを取得する

```
deploy# mco facts ipaddress -F fqdn=/^db/ -v
```

実行結果

```
Determining the amount of hosts matching filter for 2 seconds .... 1
Report for fact: ipaddress
```

```
10.3.7.142 found 1 times
```

```
db.nii.localdomain
```

```
---- rpc stats ----
```

```
Nodes: 1 / 1
```

```
Pass / Fail: 0 / 0
```

```
Start Time: Sun Mar 25 13:18:13 +0000 2012
```

```
Discovery Time: 2002.72ms
```

```
Agent Time: 41.72ms
```

```
Total Time: 2044.44ms
```

web サーバの設定を行う。

web サーバの hosts にデプロイサーバの設定を追記する。(実行結果なし)

```
deploy# ./ bin/deploy ssh exec {web サーバの IP アドレス} 'echo "{デプロイサーバの IP アドレス} deploy.nii.localdomain" >> /etc/hosts'
```

web サーバの hosts に db サーバの設定を追記する。(実行結果なし)

```
deploy# ./ bin/deploy ssh exec {web サーバの IP アドレス} 'echo "{db サーバの IP アドレス} db.nii.localdomain" >> /etc/hosts'
```

web サーバの hosts に mail サーバの設定を追記する。(実行結果なし)

```
deploy# ./ bin/deploy ssh exec {web サーバの IP アドレス} 'echo "{mail サーバの IP アドレス} mail.nii.localdomain" >> /etc/hosts'
```

web サーバで puppet を起動する

```
deploy# ./ bin/deploy ssh exec {web サーバの IP アドレス} "/etc/init.d/puppet start"
```

実行結果

```
* Starting puppet agent
```

```
debug: Failed to load library 'selinux' for feature 'selinux'
```

```
...
```

```
... #Puppet の Debug メッセージが表示されます
```

```
...
```

```
debug: Finishing transaction 70328380861720
```

```
... done.
```

web サーバで mcollective を起動する

```
deploy# ./ bin/deploy ssh exec {web サーバの IP アドレス} "/etc/init.d/mcollective start"
```

実行結果

```
Starting mcollective: *
```

3.4.3.Puppet の certificate の設定をする

デプロイサーバの上で起動している PuppetMaster と web の上で起動している PuppetAgent との通信をするために認証を行う。

puppetca コマンドを実行して、puppet-master と puppet-agent の通信の許可をする。

puppet-master に接続している puppet-agent の一覧を表示する。

```
deploy# puppetca -l
```

実行結果

```
web.i-45f6084e.nii.localdomain (2D:EE:05:07:18:33:D1:30:C6:CE:8E:AA:99:DF:A3:5D)
```

puppetmaster と puppet-agent の間の通信を許可する。

```
deploy# puppetca -s --all
```

実行結果

```
notice: Signed certificate request for web.i-45f6084e.nii.localdomain
notice:      Removing      file      Puppet::SSL::CertificateRequest      web.i-
45f6084e.nii.localdomain      at      '/var/lib/puppet/ssl/ca/requests/web.i-
45f6084e.nii.localdomain.pem'
```

3.5.Puppet Agent をロックする

意図しないタイミングで Puppet Agent が起動しないように Puppet Agent をロックする。

```
deploy# mco puppetd disable -F fqdn=/^web/ -v
```

実行結果

```
Determining the amount of hosts matching filter for 2 seconds .... 1

* [ =====> ] 1 / 1

deploy.nii.localdomain      : OK
  {:output=>"Lock created"}

---- rpc stats ----
      Nodes: 1 / 1
    Pass / Fail: 1 / 0
    Start Time: Wed Mar 28 09:41:16 +0000 2012
  Discovery Time: 2002.73ms
    Agent Time: 27.48ms
    Total Time: 2030.21ms
```

3.6.LB サーバへ Web サーバの IP アドレスを追加する

LB に Web サーバを設定する。

以下の手順をデプロイサーバ上で実行する。

- 1.起動中の web サーバの IP アドレスを取得しリストを作成する
- 2.mcollective から puppet を使用して lb サーバの nginx.conf を更新する
- 3.mcollective から lb サーバの nginx を再起動する。

deploy ディレクトリに移動する。

```
deploy# cd /root/work/deploy
```

現在起動している web サーバの IP アドレスを取得しリストを作成する。

```
deploy# mco facts ipaddress -F fqdn=/^web/ -j | /root/work/deploy/bin/retrieve
ip mco --format file > /var/tmp/nginx/nginx.ipset
```

mcollective から puppet を使用して lb サーバの nginx.conf を更新する。

```
deploy# mco puppetd runonce -I lb.nii.localdomain -v
```

実行結果

```
* [ =====> ] 1 / 1
```



```
lb.nii.localdomain : OK

{:stopped=>0,      :status=>"idling",      :idling=>1,      :running=>0,      :output=>
  "Signalled daemonized puppet agent to run (process 574); Currently
  idling;      last      completed      run      875      seconds
  ago",      :enabled=>1,      :lastrun=>1332681323}

---- rpc stats ----
      Nodes: 1 / 1
    Pass / Fail: 1 / 0
    Start Time: Sun Mar 25 13:29:58 +0000 2012
  Discovery Time: 0.00ms
    Agent Time: 51.26ms
    Total Time: 51.26ms
```

mcollective から lb サーバの nginx を再起動する。

```
deploy# mco service nginx restart -F fqdn=/^lb/ -v
```

実行結果

```
* [ =====> ] 1 / 1

lb.nii.localdomain      status=running
      OK

---- service summary ----
      Nodes: 1 / 1
    Statuses: started=1
  Elapsed Time: 1.07 s
```

3.7.web ブラウザからの確認

3.7.1.browser サーバにログイン

Windows の「スタート→すべてのプログラム→RealVNC→VNC ビューワ4→VNC ビューワの起動」を実行する。

「サーバ名」に「**browser**サーバのパブリックIP:1」を入力し「OK」ボタンをクリックする。

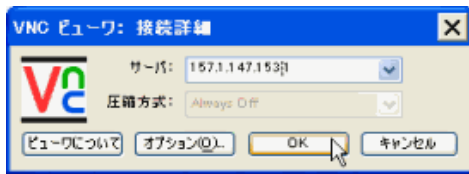


図 2VNC の接続先サーバの設定

「パスワード」に「**1qaz2wsx**」を入力し、ログインする。

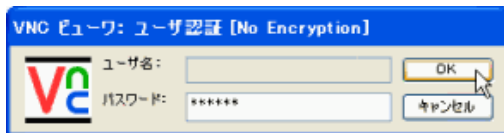


図 3VNC のパスワード入力

スクリーンセーバが起動していて、パスワードが求められる場合は「**1qaz2wsx**」を入力する。

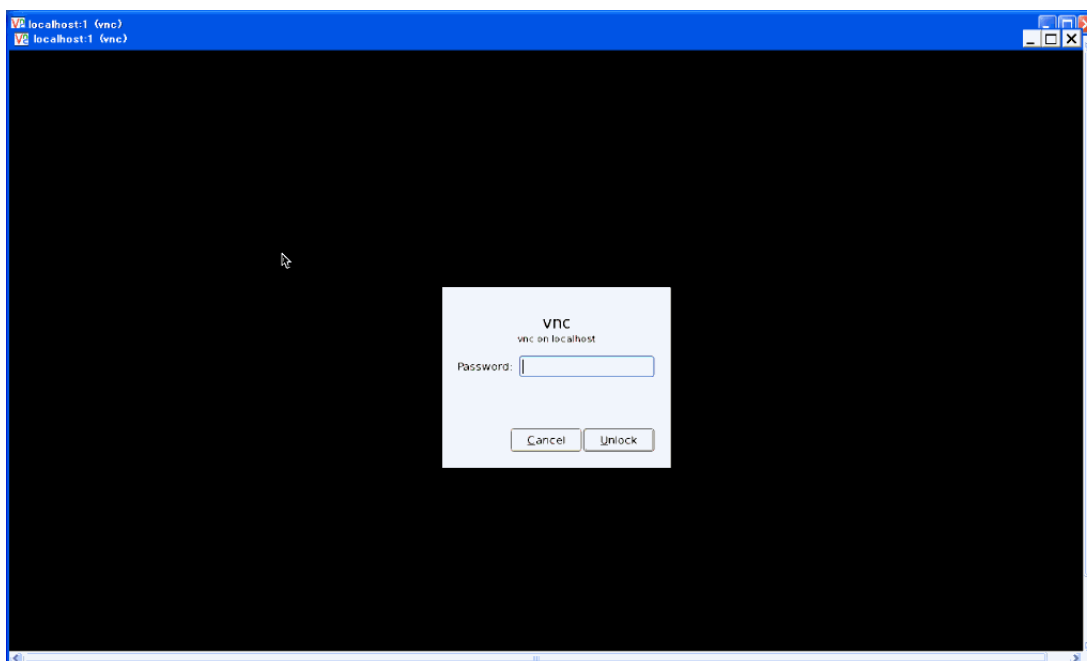


図 4スクリーンセーバ

3.7.2.hosts の設定

掲示板アプリケーションのアクセスするために/etc/hosts に lb サーバと mail サーバの IP アドレスを記述する。

lb サーバと mail サーバのプライベート IP アドレスを変更する。

```
browser$ sudo vi /etc/hosts
{lb サーバのプライベート IP} lb.nii.localdomain
{mail サーバのプライベート IP} mail.nii.localdomain
```

3.7.3.掲示板アプリケーションの確認

browser サーバ上の Firefox を起動しブックマークから掲示板アプリケーションにアクセスし動作の確認する。

ブックマークからアクセスするのは以下の URL になっている。

<code>http://lb.nii.localdomain/keijiban/top</code>
