

より高度なアプリケーションの配布 参考資料

目次

1 はじめに.....	3
2 前提条件.....	3
3 事前準備.....	3
3.1. 現在起動している WEB サーバの情報を収集する	3
3.2. テストが完了した WAR ファイルをコミットする	4
3.3. PUPPET が配布できる場所に WAR ファイルを配置する	4
4 アプリケーションを配布するするフロー.....	5
4.1. 必要台数の WEB サーバを起動および設定をする.....	5
4.1.1. 必要台数の web サーバを起動する.....	5
4.1.2. web サーバの Puppet, MCollective, 掲示板アプリケーションの設定をする	6
4.1.3. Puppet agent を Puppet master に登録する.....	8
4.1.4. 起動した Web サーバの TOMCAT を再起動する	8
4.1.5. Puppet Agent をロックする.....	8
4.2. 旧 WEB サーバの MCOLLECTIVE を停止する	9
4.3. ロードバランサの設定を変更する.....	9
4.4. 監視の設定を変更する	10
4.5. 旧 WEB サーバを停止する	12
5 WAR ファイルが配置されている事を確認する	14
6 WEB ブラウザからの確認.....	15
6.1. BROWSER サーバにログイン	15
6.2. HOSTS の設定.....	16
6.3. 掲示板アプリケーションの確認	16
7 ロードバランサのログを確認する	17
7.1. ロードバランサへのアクセス状況を確認する	17
7.2. ロードバランサのエラー状況を確認する	18

図表目次

図 1VNC の接続先サーバの設定	15
図 2VNC のパスワード入力	15
図 3スクリーンセーバ	15

1はじめに

本書は、war 形式の Web アプリケーションを Web サーバにデプロイするにあたり、新たに Web サーバ用のインスタンスを必要台数分起動し、ロードバランサの設定を変更することでデプロイを完了させる方法の手順を記述している。

この手順における作業は全てデプロイサーバで行う。

2前提条件

- Puppet(+MCollective)による構築の自動化が実施された環境であること。
- 掲示板アプリケーションのスケールアウトが実施された環境であること。
- 更新対象の war ファイルが用意されていること。
- 現在、2台の Web サーバが稼働していることとする。
- 前回デプロイした war の次のバージョンをデプロイ対象とし、本手順書では「02」をデプロイ対象として記述する。

3事前準備

3.1.現在起動している Web サーバの情報を収集する

現在起動している Web サーバ(旧 Web サーバ)のインスタンス ID と IP アドレスを記録する。

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

実行結果

Report for fact: ipaddress

10.3.7.XXX found 1 times

web.i-XXXXXXXXX.nii.localdomain

10.3.7.XXX found 1 times

web.i-XXXXXXXXX.nii.localdomain

現在起動している Web サーバのインスタンス ID と IP アドレスを記録する。

旧 Web1: i-XXXXXXXXX 10.3.7.XXX

旧 Web2: i-XXXXXXXXX 10.3.7.XXX

3.2. テストが完了した war ファイルをコミットする

作業ディレクトリに移動

```
deploy# cd /var/tmp/keijiban/
```

Subversion をチェックアウトしたディレクトリに、更新対象の war ファイルをコピーする。

```
deploy# cp -p /var/tmp/wars/02/keijiban.war keijiban.war
```

Subversion の状態を確認する。

```
deploy# svn status
```

実行結果

```
M      keijiban.war
```

変更をコミットする

```
deploy# svn commit -m '02'
```

実行結果

```
Sending      keijiban.war
Transmitting file data .
Committed revision 3.
```

3.3. Puppet が配布できる場所に war ファイルを配置する

作業ディレクトリに移動する。

```
deploy# cd /var/tmp/
```

Subversion から war ファイルを export する。

```
deploy# svn export svn+ssh://localhost/root/work/repos/keijiban/keijiban.war
/var/tmp/keijiban.war
```

実行結果

```
A      /var/tmp/keijiban.war
Export complete.
```

4アプリケーションを配布するするフロー

1. 必要台数の Web サーバを起動および設定をする
2. 旧 Web サーバの mcollective を停止する
3. ロードバランサの設定を変更する
4. 監視の設定を変更する
5. 旧 Web サーバを停止する

※ロードバランサと Nagios の設定を生成する際に mcollective から取得した情報から行っているため、新 Web サーバを起動したあとすぐに、旧 Web サーバの mcollective の停止を行う。

4.1.必要台数の Web サーバを起動および設定をする

「掲示板アプリケーションのスケールアウト」で行った手順のうち、ロードバランサの設定以降を実施せずに Web サーバの起動および設定を行う。ただし、起動および設定は複数台を一度に行う。

4.1.1.必要台数の web サーバを起動する

webapi を使用して必要台数の web サーバを新規に起動する。

作業ディレクトリに移動する

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

web のマシンイメージを起動する

```
deploy# ./bin/deploy instances launch web --count=2
```

実行結果

```
I, [2012-03-24T13:02:51.987045 #1995] INFO -- : New RightAws::Ec2 using single-threaded mode
I, [2012-03-24T13:02:51.987208 #1995] INFO -- : Launching instance of image emi-1FE61454 for cVy1n9VFKkGhICgaMxIWhA, key: tucl0004, groups: tucl0004
I, [2012-03-24T13:02:51.987781 #1995] INFO -- : Opening new HTTPS connection to vclc0006.eccloud.nii.ac.jp:8773
i-YYYYYYYY
i-YYYYYYYY
```

web サーバ台数分の IP アドレスを取得する。

```
deploy# ./bin/deploy instances describe --instanceids= {新web1の instance_id}
{新web2の instance_id} --key=ipaddr | ./bin/retrieve ip cloud
```

実行結果

```
10.3.4.135 10.3.4.136
```

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。

新 Web1:

i-YYYYYYYY

10.3.4.135

新 Web2:

i-YYYYYYYY

10.3.4.136

以降これを、「新バージョンの Web サーバ情報」と呼ぶ。

4.1.2.web サーバの Puppet,MCollective,掲示板アプリケーションの設定をする

mail サーバの ip アドレスを取得する

```
deploy# mco facts ipaddress -F fqdn=/^mail/ -j | ./bin/retrieve ip mco
```

実行結果

10.3.5.133

db サーバの ip アドレスを取得する

```
deploy# mco facts ipaddress -F fqdn=/^db/ -j | ./bin/retrieve ip mco
```

実行結果

10.3.5.134

作業ディレクトリに移動

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

起動した全ての Web サーバに対して以下を行う

- ssh ログインし/etc/hosts に deploy, mail, db サーバの IP アドレスを追加
- puppet を起動する
- mcollective を起動する

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BのIPアドレス} "/etc/init.d/mcollective
start"
```

実行結果

```
Starting mcollective: *
```

4.1.3.Puppet agent を Puppet master に登録する

puppetmaster にリクエストがきていることを確認する。

```
deploy# puppetca -l
```

実行結果

```
web.i-YYYYYYYY.nii.localdomain (88:67:53:76:EF:A5:B7:7A:E5:08:E1:38:73:CB:3E:49)
web.i-YYYYYYYY.nii.localdomain (72:A7:38:73:CB:3E:49:67:53:76:EF:A5:B7:7A:E5:08)
```

puppetmaster との通信を許可する。通信を許可した段階で puppet が puppetmaster の manifest を元に各サーバでの設定を開始する。

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

実行結果

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

4.1.4.起動した Web サーバの TOMCAT を再起動する

起動した Web サーバの TOMCAT を再起動する。

```
deploy# mco service tomcat6 restart -F fqdn=/^web. {新 Web サーバ InstanceID}/ -v
```

実行結果

```
Determining the amount of hosts matching filter for 2 seconds .... 1
(中略)
---- service summary ----
      Nodes: 1 / 1
      Statuses: started=1
      Elapsed Time: 16.20 s
```

4.1.5.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
```

4.2.旧 Web サーバの mcollective を停止する

作業ディレクトリに移動

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

旧 Web サーバの mcollective を停止します

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

実行結果

```
Shutting down mcollective: *
```

4.3.ロードバランサの設定を変更する

作業ディレクトリに移動

```
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 client から puppet を使用して lb サーバの設定を更新する

```
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,      :outpu
t=>      "Signalled daemonized puppet agent to run (process 617); Currently
idling;      last      completed      run      591      seconds
ago",      :enabled=>1,      :lastrun=>1337752247}
```

```
----- rpc stats -----
      Nodes: 1 / 1
    Pass / Fail: 1 / 0
    Start Time: Wed May 23 06:00:38 +0000 2012
  Discovery Time: 0.00ms
    Agent Time: 44.49ms
    Total Time: 44.49ms
```

mcollective client から puppet を使用して 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: 0.75 s
```

4.4.監視の設定を変更する

作業ディレクトリに移動

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

各サーバ郡の IP アドレスのリストを作成する

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

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

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

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

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

mcollective client から puppet を使用して監視サーバの設定を更新する

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

実行結果

```
* [ =====> ] 1 / 1  
  
monitor.nii.localdomain          : OK  
  
{:stopped=>0,      :status=>"idling",      :idling=>1,      :running=>0,      :output=>  
  "Signalled daemonized puppet agent to run (process 1161); Currently  
  idling;          last      completed      run      299      seconds  
ago",      :lastrun=>1337752986,      :enabled=>1}  
  
---- rpc stats ----  
      Nodes: 1 / 1  
      Pass / Fail: 1 / 0  
      Start Time: Wed May 23 06:08:05 +0000 2012  
      Discovery Time: 0.00ms  
      Agent Time: 46.99ms  
      Total Time: 46.99ms
```

mcollective client から旧 Web サーバの ganglia-monitor を停止する

```
deploy# ./bin/deploy ssh exec {旧 Web サーバ IP アドレス} '/etc/init.d/ganglia-  
monitor stop'
```

実行結果

```
Stopping Ganglia Monitor Daemon: gmond
```

mcollective client から新 Web サーバの ganglia-monitor を起動する

```
deploy# mco service ganglia-monitor start -F fqdn=/^web.{新 Web サーバ InstanceID}/  
-v
```

実行結果

```
Determining the amount of hosts matching filter for 2 seconds .... 1  
  
* [ =====> ] 1 / 1  
  
web.i-480A0750.nii.localdomain      status=stopped
```

OK

```
----- service summary -----  
Nodes: 1 / 1  
Statuses: stopped=1  
Elapsed Time: 0.74 s
```

mcollective client から gmetad を再起動する

```
deploy# mco service gmetad restart -F fqdn=/^monitor/ -v
```

実行結果

```
Determining the amount of hosts matching filter for 2 seconds .... 1  
  
* [ ======> ] 1 / 1  
  
monitor.nii.localdomain          status=running  
OK  
  
----- service summary -----  
Nodes: 1 / 1  
Statuses: started=1  
Elapsed Time: 0.76 s
```

mcollective client から nagios を再起動する

```
deploy# mco service nagios3 restart -F fqdn=/^monitor/ -v
```

実行結果

```
* [ ======> ] 1 / 1  
  
monitor.nii.localdomain          status=running  
OK  
  
----- service summary -----  
Nodes: 1 / 1  
Statuses: started=1  
Elapsed Time: 16.15 s
```

4.5.旧 Web サーバを停止する

作業ディレクトリに移動

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

旧バージョンの Web サーバを停止します

```
deploy# ./bin/deploy instances terminate --instanceids= {旧 Web サーバ InstanceID}  
{旧 Web サーバ InstanceID}
```

出力結果

```
I, [2012-05-23T06:16:00.638893 #3507] INFO -- : New RightAws::Ec2 using single-  
threaded mode  
I, [2012-05-23T06:16:00.639457 #3507] INFO -- : Opening new HTTPS connection to  
vclc0006.ecloud.nii.ac.jp:8773  
terminate instances i-xxxxxx i-xxxxxx
```

5war ファイルが配置されている事を確認する

新 Web サーバに war ファイルが配置されている事を確認する

```
deploy# mco shellcmd --cmd="ls -l /var/lib/tomcat6/webapps" -F fqdn=/^web/ -v
```

実行結果

```
* [ =====> ] 2 / 2

[web.i-475C0824.nii.localdomain] exit=0:
total 14940
drwxr-xr-x 3 root    root        4096 Mar  8 10:03 ROOT
drwxr-xr-x 4 tomcat6 tomcat6     4096 Mar 25 08:55 keijiban
-rw-r--r-- 1 root    root      15266541 Mar 25 08:55 keijiban.war

[web.i-2AB005B5.nii.localdomain] exit=0:
total 14940
drwxr-xr-x 3 root    root        4096 Mar  8 10:03 ROOT
drwxr-xr-x 4 tomcat6 tomcat6     4096 Mar 25 08:55 keijiban
-rw-r--r-- 1 root    root      15266541 Mar 25 08:55 keijiban.war
```

6web ブラウザからの確認

6.1.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。

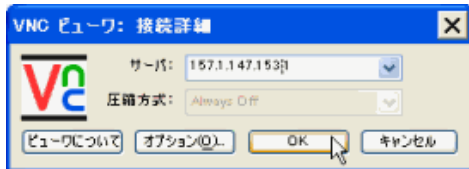


図 1VNC の接続先サーバの設定

「パスワード」に「1qaz2wsx」を入力し、ログインする。

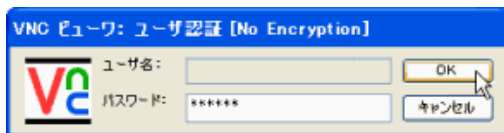


図 2VNC のパスワード入力

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」を入力します。

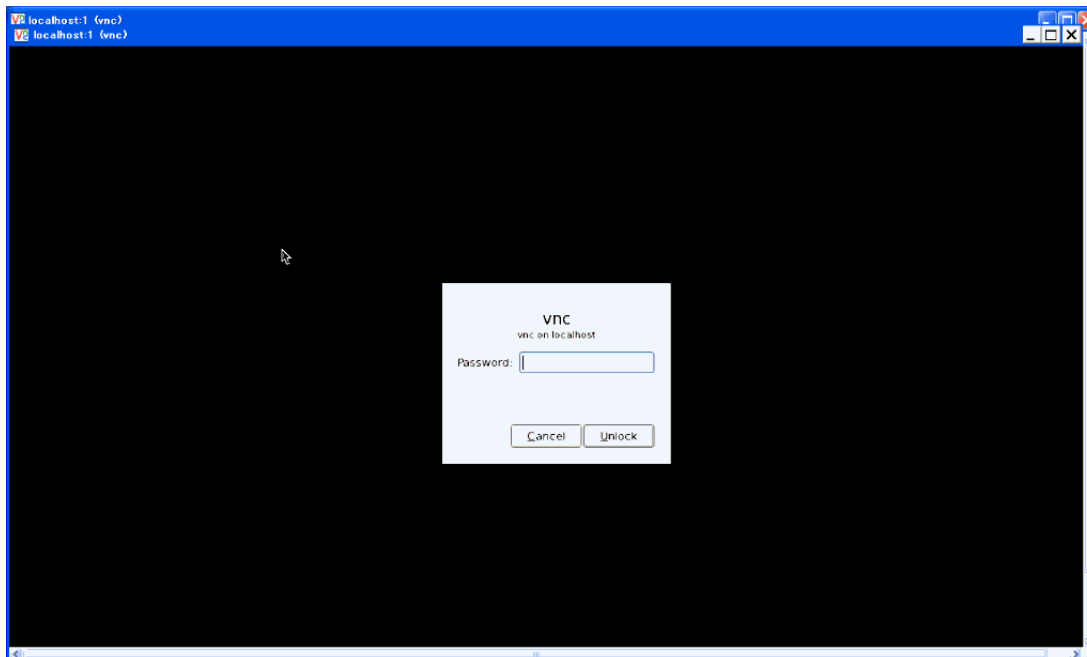


図 3スクリーンセーバ

6.2.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 アドレスの確認をする。

```
browser$ sudo vi /etc/hosts
```

確認内容

```
{/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
```

6.3.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をします。

ブックマークからアクセスするのは以下の URL になっています。

```
http://lb.nii.localdomain/keijiban/top
```

画面上部が「掲示板 ver.2(02)」となっていれば、デプロイ完了である。

7ロードバランサのログを確認する

7.1.ロードバランサへのアクセス状況を確認する

Web サーバを複数台で運用する際にアクセス状況を確認するには、ロードバランサの Nginx ログを確認する。

Nginx ログを10行表示する

```
deploy# mco shellcmd --cmd="tail /var/log/nginx/access.log" -F fqdn=/^lb/ -v
```

実行結果

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

[lb.nii.localdomain] exit=0:
136.187.36.110 - - [25/Mar/2012:17:56:35 +0000] "GET /keijiban/entry/index
HTTP/1.1" 200 1311 "-" "Java/1.6.0_24"
136.187.36.110 - - [25/Mar/2012:17:56:35 +0000] "POST /keijiban/entry/ HTTP/1.1"
302 0 "-" "Java/1.6.0_24"
136.187.36.110 - - [25/Mar/2012:17:56:35 +0000] "POST /keijiban/entry/ HTTP/1.1"
302 0 "-" "Java/1.6.0_24"
136.187.36.110 - - [25/Mar/2012:17:56:35 +0000] "GET /keijiban/entry/index
HTTP/1.1" 200 1311 "-" "Java/1.6.0_24"
136.187.36.110 - - [25/Mar/2012:17:56:35 +0000] "GET /keijiban/entry/index
HTTP/1.1" 200 1311 "-" "Java/1.6.0_24"
136.187.36.110 - - [25/Mar/2012:17:56:35 +0000] "GET /keijiban/entry/index
HTTP/1.1" 200 1311 "-" "Java/1.6.0_24"
136.187.36.110 - - [25/Mar/2012:17:57:04 +0000] "GET
/keijiban/entry/verifyToken?token=mQ9IOfoRdX HTTP/1.1" 200 1129 "-"
"Java/1.6.0_24"
136.187.36.110 - - [25/Mar/2012:17:57:04 +0000] "GET
/keijiban/entry/verifyToken?token=cQPHxdAJZK HTTP/1.1" 200 1129 "-"
"Java/1.6.0_24"
136.187.36.110 - - [25/Mar/2012:17:57:04 +0000] "GET
/keijiban/entry/verifyToken?token=EKS4nmBtum HTTP/1.1" 200 1129 "-"
"Java/1.6.0_24"
136.187.36.110 - - [25/Mar/2012:17:57:04 +0000] "GET
/keijiban/entry/verifyToken?token=N8UZga9Lt5 HTTP/1.1" 200 1129 "-"
"Java/1.6.0_24"
```

7.2.ロードバランサのエラー状況を確認する

前回確認した後、デプロイ時に404エラーが発生していないことを確認する。

Nginx で404エラーが発生しているか確認する

```
deploy# mco shellcmd --cmd='grep " 404 " /var/log/nginx/access.log | tail' -F  
fqdn=/^lb/ -v
```

実行結果

```
* [ =====> ] 1 / 1  
  
[lb.nii.localdomain] exit=0:  
136.187.36.110 -- [25/Mar/2012:18:17:50 +0000] "GET /keijiban/top HTTP/1.1" 404  
991 "-" "Java/1.6.0_24"  
136.187.36.110 -- [25/Mar/2012:18:17:51 +0000] "GET /keijiban/article/40  
HTTP/1.1" 404 1012 "-" "Java/1.6.0_24"  
136.187.36.110 -- [25/Mar/2012:18:17:51 +0000] "GET /keijiban/article/39  
HTTP/1.1" 404 1012 "-" "Java/1.6.0_24"  
136.187.36.110 -- [25/Mar/2012:18:17:51 +0000] "GET /keijiban/article/38  
HTTP/1.1" 404 1012 "-" "Java/1.6.0_24"  
136.187.36.110 -- [25/Mar/2012:18:17:51 +0000] "POST /keijiban/comment/  
HTTP/1.1" 404 1006 "-" "Java/1.6.0_24"  
136.187.36.110 -- [25/Mar/2012:18:17:51 +0000] "POST /keijiban/comment/  
HTTP/1.1" 404 1006 "-" "Java/1.6.0_24"  
136.187.36.110 -- [25/Mar/2012:18:17:52 +0000] "GET /keijiban/article HTTP/1.1"  
404 1003 "-" "Java/1.6.0_24"  
136.187.36.110 -- [25/Mar/2012:18:17:52 +0000] "GET /keijiban/article/  
HTTP/1.1" 404 1006 "-" "Java/1.6.0_24"  
136.187.36.110 -- [25/Mar/2012:18:17:52 +0000] "GET /keijiban/top HTTP/1.1" 404  
991 "-" "Java/1.6.0_24"  
136.187.36.110 -- [25/Mar/2012:18:17:52 +0000] "GET /keijiban/top/ HTTP/1.1"  
404 994 "-" "Java/1.6.0_24"
```