
Constructing a Clustering Server - MagicInfo Premium Server

Document Name	Constructing a clustering for MagicInfo Premium Server		
Date	2013-04-01	Version	1.0
Written by	Sam-ryong Kim	Approved by	Eun-sook Seong

Revision History

Version	Date	Revision detail	Written by	Approved by
1.0	2013-04-01	First edition	Sam-ryong Kim	Eun-sook Seong

Contents

- 1 Overview 4
 - 1.1 Purpose 4
 - 1.2 Coverage 4
- 2 Considerations before extending the Server 5
 - 2.1 Performance increase using hardware 5
 - 2.2 Performance increase using software 5
- 3 Server structure 7
 - 3.1 Small-size server structure..... 7
 - 3.2 Large-size server structure..... 8
- 4 Constructing the small-size server 10
 - 4.1 Installing DB 10
 - 4.2 Installing Tomcat 10
- 5 Constructing the large-size server 11
 - 5.1 Installing DB 11
 - 5.2 Installing Tomcat 11
 - 5.3 Installing HAProxy 11
 - 5.4 Storage settings 11
 - 5.5 Installing Memcached 11
 - 5.6 VNC server settings..... 11
 - 5.7 MagicInfo Player settings 11
- 6 Installing and constructing HAProxy 12
 - 6.1 Install preparations..... 12
 - 6.2 Install 13
 - 6.3 Configuring settings 17
- 7 Installing and constructing Memcached 24
 - 7.1 Memcached settings (Linux) 24
 - 7.1.1 Install preparations 24
 - 7.1.2 Installing 25
 - 7.1.3 Configuring the settings..... 25
 - 7.2 Memcached settings (Windows) 27
 - 7.2.1 Install preparations 27
 - 7.2.2 Installing 27
 - 7.2.3 Configuring the settings..... 27
 - 7.3 Configuring the memcached settings from MagicInfo 28
- 8 VNC tunneling server settings..... 30
 - 8.1 For the small-size structure..... 30
 - 8.2 For the large-size structure 32
- 9 Constructing a storage server 34
- 10 JVM memory tuning 37
- 11 FAQ..... 39

1 Overview

1.1 Purpose

This document provides instructions on how to construct the Premium Server according to the number of clients for MagicInfo Premium Server (hereinafter Premium Server). Premium Server is designed to operate dozens to hundreds of thousands of devices. The application provides customers with a variety of methods to construct the server and is easily extendable.

Open source software items used to extend the Premium Server are compatible with Linux. This document targets engineers who are familiar with the Linux operating system.

1.2 Coverage

This document is divided largely into the following sections:

- Considerations before extending the Premium Server
- Premium Server structure according to the number of clients
- Installing and configuring HAProxy which is a layer 4 software load balancer
- Configuring memcached which is a cache server

Basic instructions on how to install Premium Server can be found in the user guide and are omitted in this document.

2 Considerations before extending the Server

A server administrator may extend a server in order to prevent the server from excessive load and guarantee zero downtime. Increasing the number of servers does not guarantee a performance improvement. It may be time consuming and infeasible for an administrator to manage servers at one time.

If the customer finds the server slow and requests technical support, it is recommended to deal with the hardware rather than increasing the number of servers. For example, to double the server performance, increase the hardware performance by two times, instead of adding two servers. Additional costs are likely with high prices for hardware. Hardware with higher performance allows an administrator's effectiveness in managing many devices to be more manageable. This can influence customer satisfaction.

It is recommended to take into account the hardware performance, the server administrator's technical skills, and the server administration type (e.g., virtual machine administration) when providing advice on extending the server.

It is best to consider both the hardware and software.

2.1 Performance increase using hardware

This is the easiest way to increase server performance without adding more servers or software. In this situation, the administrator's overall workload may not necessarily increase. Additional cost is inevitable.

If the CPU or memory is insufficient from a system monitoring standpoint, simply upgrade or replace them. For example, if the performance of a 2G memory server degrades due to full memory usage, purchase and install additional memory on the server machine.

Changing the CPU or memory may not solve the problem in some cases. If the server is slow although the CPU and memory are sufficient, the issue may be linked to the capacity of the server. For example, if the number of clients has increased when the server capacity is 1000 tps and 1500 tps capacity is required, the server machine may need to be replaced. A considerable amount of additional cost will be incurred in this case.

In addition to replacing the hardware, another way to improve the performance is to create a separate DB server, considering the maintenance cost.

2.2 Performance increase using software

Hardware performance can be improved easily without increasing the administrator's workload. Software can be complicated to install and may increase the number of servers and software for the administrator to handle.

Premium Server provides four options to improve the software performance.

- Operation of a separate PostgreSQL DB server
 - If Tomcat and PostgreSQL are running on the same server and the server slows down, run them on different servers.
 - Run Tomcat from one server and Postgres from the other.
 - This option is suitable when the recommended number of clients is between 500 and 1,500 (the recommended number may vary depending on the hardware performance).
- Operation of two Tomcat servers or more

- This option addresses potential failures and improves performance using a redundant server.
- Suitable when the recommended number of clients is between 1,500 and 10,000.
- It is possible to operate two Tomcat servers or more and one DB server or more.

- Operation of a separate cache server (memcached)
 - This option is suitable when 10,000 clients or more are used.
 - In addition to Tomcat and DB servers, a memcached server, which is a separate cache server, is required.
 - Since data is buffered in the memcached server, the chance of a bottleneck decreases even if there are a large number of clients. This can vastly improve the server performance.

- Distributed DB
 - If there is excessive load on the DB server despite one of the above three options, it is advisable to use two DB servers or more.
 - Operate the DB servers in the master-slave format (distribute data using software called pgpool).

Above are recommended options. It is possible to install using the appropriate option depending on the situation. Factors that should be taken into account include hardware performance, the administrator's skills to use software, and the cost. It is greatly dependent on the administrator's skills.

Also, it is possible to apply the redundant server, failover or load balancing function to Premium Server by configuring a layer 4 software load balancer such as HAProxy.

3 Server structure

3.1 Small-size server structure

Small-size structure provides two options. The first option is to install Tomcat and PostgreSQL on a single server. The second is to install these on different servers. Details on installing them on a single server have not been changed and thus are not mentioned in this document.

The small-size structure should be sufficient for most situations.

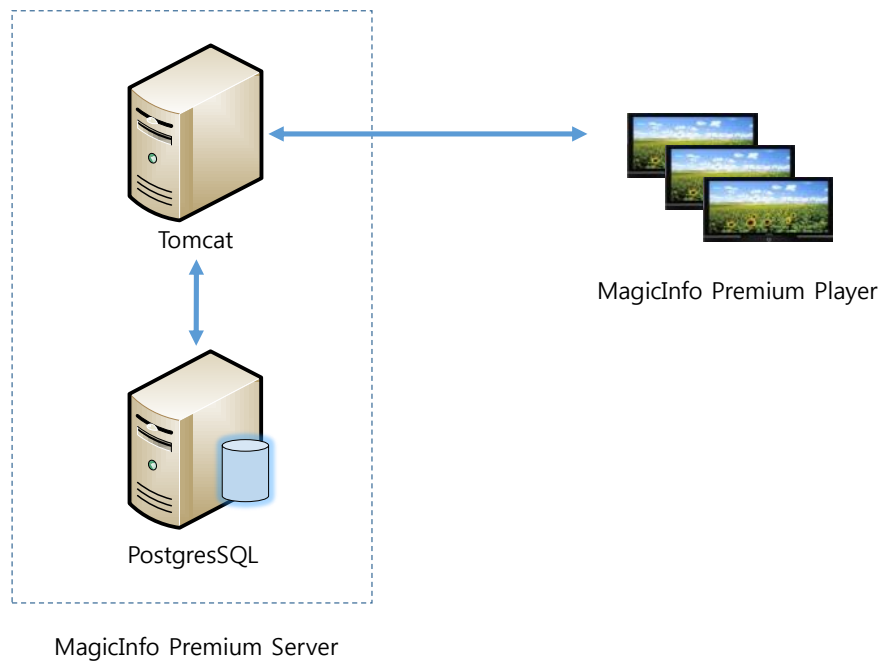


Figure 1 Small-size structure

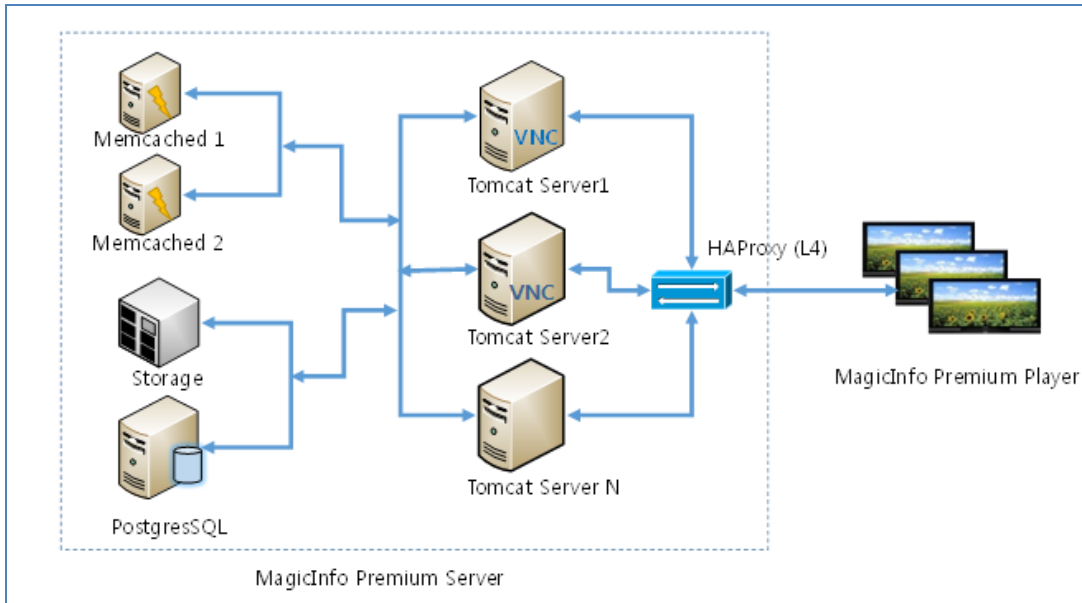
The server structure above is recommended when clients number in the thousands. The available number of clients may vary depending on the hardware specifications.

- Applicable situations
 - Where clients number in the thousands or less (applicability depends the hardware specifications)
 - When a single server is used and the server is slow
 - When the number of clients is expected to increase gradually
- Advantages
 - Tomcat and DB are separated resulting in distributed workload of the server.
 - The cause of a failure can be found easily.
- Disadvantages
 - Additional administrators may be required if there is an increase in the number of servers.
 - Server maintenance cost increases.

3.2 Large-size server structure

The large-size structure is suitable when thousands of LFDs or more are in use where a service type like SaaS is required. It is the most complex structure and recommended for large-scale franchise enterprises, customers that need a centralized management, and advertising companies.

The complex structure requires a server administrator and in certain cases a dedicated group to manage the server.



Two or more Tomcat servers are used in this server structure. To use multiple Tomcat servers, HAProxy, which is a layer 4 software load balancer, is required to perform the load balancing and failover functions. If the customer already has a layer 4 hardware load balancer, it can be used instead. To connect a layer 4 hardware load balancer, consult the manufacturer of the device. In addition, the memcached server is added. The memcached server can be used to store memory distributed between Tomcat servers. This prevents the server performance from degrading when the number of Tomcat servers increases. The memcached server can be used along with the DB server.

(Memcached is a sort of distributed cache server widely used in the SNS field.)

- Applicable situations
 - Where clients number in the thousands to hundreds of thousands
 - Where the number of LFDs keeps increasing
 - Suitable for large-scale franchise enterprises/advertising companies/SaaS-type business owners
 - Where zero downtime service is required in the occurrence of a server failure
- Advantages
 - Zero downtime service is provided in the occurrence of a failure (failover).
 - Workload of a Tomcat server is distributed (load balancing).
 - An increase in the number of LFDs can be resolved by increasing the number of Tomcat servers.
 - Flexible server structure can be obtained.
 - Server performance does not degrade even if the number of LFDs increases.
- Disadvantages
 - Complex server structure
 - A server administrator and a dedicated administrative group are required.

- The minimum number of servers is five (for efficient management using a VM solution).
 - HAProxy Server: one
 - Tomcat Server: two
 - Storage Server: one
 - PostgreSQL Server(+ memcached) : one

A separate slave DB can be configured if there is excessive load in PostgreSQL DB. It is possible to connect a 3rd party backup server to ensure stable operation of the server.

4 Constructing the small-size server

It is very simple to construct the small-size server. All you have to do is to install a Tomcat server and a PostgreSQL server. Tomcat is only supported if MagicInfo Premium Server is already installed. This allows the server to be constructed with ease. Note that the following procedures should be observed when installing.

4.1 Installing DB

The first thing to do before installing Tomcat is to construct a DB server. Install the DB server according to the following process. This is to check whether the DB server can be connected when the Tomcat server is installed.

- 1) Installing the PostgreSQL server on the DB server
 - A. Install the PostgreSQL server.
 - B. Register the PostgreSQL port exception in Windows Firewall (port: 5432)
- 2) Installing MagicInfo Premium Server DB (refer to the MagicInfo user guide)
 - A. Run the MagicInfo Premium Server installer.
 - B. Select the option to install "DB" only.

4.2 Installing Tomcat

After the DB server is installed, install the Tomcat server on another server. Most of the options are installed automatically as long as the settings are configured properly.

- 1) Installing MagicInfo Premium Server Tomcat
 - A. Run the MagicInfo Premium Server installer.
 - B. Select the option to install "WAS" only.
- 2) Go to <http://localhost:7001/MagicInfo> from the server, and perform an operation test.

5 Constructing the large-size server

For this server structure, two or more Tomcat servers are installed and load balancing is available using a layer 4 software load balancer such as HAProxy. This is different from the small-size structure. In addition, the memcached server configuration is required.

5.1 Installing DB

[Install referring to chapter 4.1.](#)

5.2 Installing Tomcat

[Install referring to chapter 4.2.](#)

5.3 Installing HAProxy

[Install referring to chapter 7.](#)

5.4 Storage settings

[Install referring to chapter 10.](#)

5.5 Installing Memcached

[Install referring to chapter 8.](#)

5.6 VNC server settings

[Install referring to chapter 9.2.](#)

5.7 MagicInfo Player settings

Make sure the server IP for MagicInfo Player is the same as the HAProxy server IP.

6 Installing and constructing HAProxy

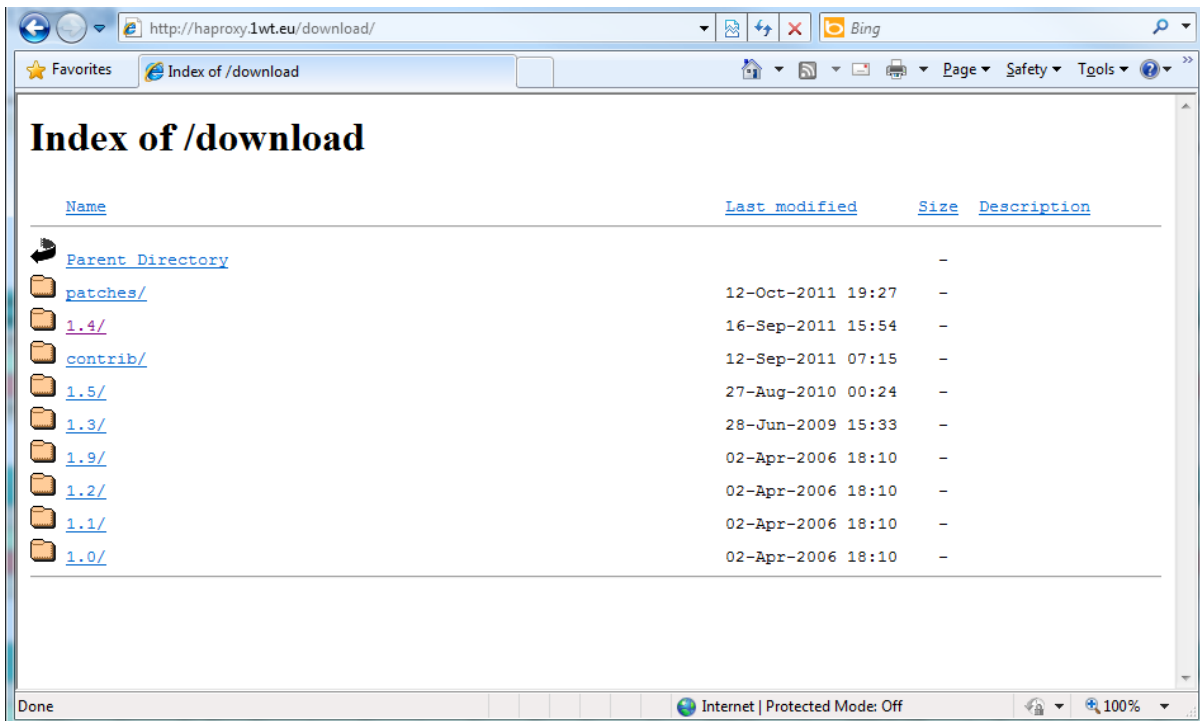
HAProxy is an open source software which provides a layer 4 software load balancing function. If the customer already has a layer 4 hardware load balancer such as H/W L4, HAProxy is not needed. For information on layer 4 hardware load balancing settings, contact the manufacturer of the device. Samsung Electronics does not provide service for 3rd party load balancers.

HAProxy software is required for the large-size server structure.

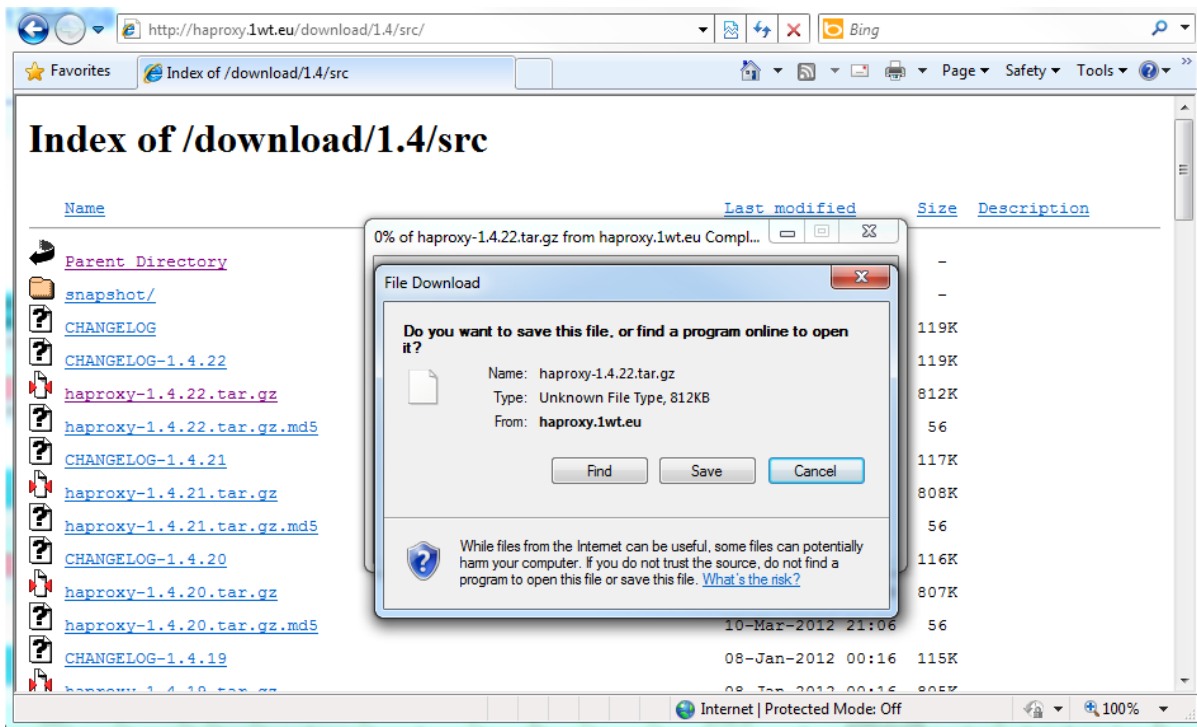
6.1 Install preparations

HAProxy can be downloaded from the following official website.

- Download address: <http://haproxy.1wt.eu/download/>



Instructions on install and settings in this document are for version 1.4.22 which is a stable build.



<http://haproxy.1wt.eu/download/1.4/src/haproxy-1.4.22.tar.gz>

A Linux server is required to install HAProxy. Details on installing HAProxy in this document are applicable to CentOS 6.0 using the following system settings.

- System settings example
 - OS : CentOS 6.0
 - HAProxy version : HAProxy 1.4.22
 - Load Balancer Server IP (CentOS) : 192.168.10.100
 - MagicInfo Premium Server
 - ◆ Server 1 IP : 192.168.10.101
 - ◆ Server 2 IP : 192.168.10.102
 - Port: 7001 (http), 21(ftp)

6.2 Install

- 1) Unzip the file.
 # tar xzf haproxy-1.4.22.tar.gz

```
[root@localhost ~]# tar xzf haproxy-1.4.22.tar.gz
[root@localhost ~]# █
```

- 2) Move the folder.
cd haproxy-1.4.22

```
[root@localhost ~]# tar xzf haproxy-1.4.22.tar.gz
[root@localhost ~]# cd haproxy-1.4.22
[root@localhost haproxy-1.4.22]# █
```

3) Edit Makefile.

vi Makefile

Change "PREFIX = /usr/local" to "PREFIX = /usr".

```
# DESTDIR is not set by default and is used for installation only.
# It might be useful to set DESTDIR if you want to install haproxy
# in a sandbox.
# PREFIX is set to "/usr/local" by default and is used for installation only.
# SBINDIR is set to "$(PREFIX)/sbin" by default and is used for installation
# only.
# MANDIR is set to "$(PREFIX)/share/man" by default and is used for
# installation only.
# DOCDIR is set to "$(PREFIX)/doc/haproxy" by default and is used for
# installation only.
#
# Other variables :
# DLMALLOC_SRC : build with dmalloc, indicate the location of dmalloc.c.
# DLMALLOC_THRES : should match PAGE_SIZE on every platform (default: 4096).
# PCREDIR : force the path to libpcre.
# IGNOREGIT : ignore GIT commit versions if set.
# VERSION : force haproxy version reporting.
# SUBVERS : add a sub-version (eg: platform, model, ...).
# VERDATE : force haproxy's release date.

#### Installation options.
DESTDIR =
PREFIX = /usr/
```

4) Configure the compilation settings.

32bit : # make TARGET=linux26 ARCH=x86

64bit : # make TARGET=linux26 ARCH=x86_64

```
[root@localhost ~]# tar xzf haproxy-1.4.22.tar.gz
[root@localhost ~]# cd haproxy-1.4.22
[root@localhost haproxy-1.4.22]# make TARGET=linux26 ARCH=x86
```

```

-c -o ebtrees/ebmbtree.o ebtrees/ebmbtree.c
gcc -Iinclude -Iebtree -Wall -O2 -g -fno-strict-aliasing -DTPROXY -DCONF I
G_HAP_CRYPT -DENABLE_POLL -DENABLE_EPOLL -DENABLE_SEPOLL -DNETFILTER -DUSE_GETSO
CKNAME -DCONFIG_HAPROXY_VERSION="\1.4.22\" -DCONFIG_HAPROXY_DATE="\2012/08/09\"
-c -o ebtrees/ebsttree.o ebtrees/ebsttree.c
gcc -Iinclude -Iebtree -Wall -O2 -g -fno-strict-aliasing -DTPROXY -DCONF I
G_HAP_CRYPT -DENABLE_POLL -DENABLE_EPOLL -DENABLE_SEPOLL -DNETFILTER -DUSE_GETSO
CKNAME -DCONFIG_HAPROXY_VERSION="\1.4.22\" -DCONFIG_HAPROXY_DATE="\2012/08/09\"
-c -o ebtrees/ebimtree.o ebtrees/ebimtree.c
gcc -Iinclude -Iebtree -Wall -O2 -g -fno-strict-aliasing -DTPROXY -DCONF I
G_HAP_CRYPT -DENABLE_POLL -DENABLE_EPOLL -DENABLE_SEPOLL -DNETFILTER -DUSE_GETSO
CKNAME -DCONFIG_HAPROXY_VERSION="\1.4.22\" -DCONFIG_HAPROXY_DATE="\2012/08/09\"
-c -o ebtrees/ebistree.o ebtrees/ebistree.c
gcc -g -o haproxy src/haproxy.o src/sessionhash.o src/base64.o src/protocols.o
src/uri_auth.o src/standard.o src/buffers.o src/log.o src/task.o src/time.o src/
fd.o src/pipe.o src/regex.o src/cfgparse.o src/server.o src/checks.o src/queue.o
src/client.o src/proxy.o src/stick_table.o src/proto_uvst.o src/proto_http.o sr
c/stream_sock.o src/appsession.o src/backend.o src/lb_chash.o src/lb_fwlc.o src/
lb_fwrr.o src/lb_map.o src/stream_interface.o src/dumpstats.o src/proto_tcp.o sr
c/session.o src/hdr_idx.o src/ev_select.o src/signal.o src/acl.o src/pattern.o s
rc/memory.o src/freq_ctr.o src/auth.o src/ev_poll.o src/ev_epoll.o src/ev_sepoll
.o ebtrees/ebtree.o ebtrees/eb32tree.o ebtrees/eb64tree.o ebtrees/ebmbtree.o ebtrees/
ebsttree.o ebtrees/ebimtree.o ebtrees/ebistree.o -lcrypt
[root@localhost haproxy-1.4.22]#

```

5) Perform compilation.

make install

```

-c -o ebtrees/ebmbtree.o ebtrees/ebmbtree.c
gcc -Iinclude -Iebtree -Wall -O2 -g -fno-strict-aliasing -DTPROXY -DCONF I
G_HAP_CRYPT -DENABLE_POLL -DENABLE_EPOLL -DENABLE_SEPOLL -DNETFILTER -DUSE_GETSO
CKNAME -DCONFIG_HAPROXY_VERSION="\1.4.22\" -DCONFIG_HAPROXY_DATE="\2012/08/09\"
-c -o ebtrees/ebsttree.o ebtrees/ebsttree.c
gcc -Iinclude -Iebtree -Wall -O2 -g -fno-strict-aliasing -DTPROXY -DCONF I
G_HAP_CRYPT -DENABLE_POLL -DENABLE_EPOLL -DENABLE_SEPOLL -DNETFILTER -DUSE_GETSO
CKNAME -DCONFIG_HAPROXY_VERSION="\1.4.22\" -DCONFIG_HAPROXY_DATE="\2012/08/09\"
-c -o ebtrees/ebimtree.o ebtrees/ebimtree.c
gcc -Iinclude -Iebtree -Wall -O2 -g -fno-strict-aliasing -DTPROXY -DCONF I
G_HAP_CRYPT -DENABLE_POLL -DENABLE_EPOLL -DENABLE_SEPOLL -DNETFILTER -DUSE_GETSO
CKNAME -DCONFIG_HAPROXY_VERSION="\1.4.22\" -DCONFIG_HAPROXY_DATE="\2012/08/09\"
-c -o ebtrees/ebistree.o ebtrees/ebistree.c
gcc -g -o haproxy src/haproxy.o src/sessionhash.o src/base64.o src/protocols.o
src/uri_auth.o src/standard.o src/buffers.o src/log.o src/task.o src/time.o src/
fd.o src/pipe.o src/regex.o src/cfgparse.o src/server.o src/checks.o src/queue.o
src/client.o src/proxy.o src/stick_table.o src/proto_uvst.o src/proto_http.o sr
c/stream_sock.o src/appsession.o src/backend.o src/lb_chash.o src/lb_fwlc.o src/
lb_fwrr.o src/lb_map.o src/stream_interface.o src/dumpstats.o src/proto_tcp.o sr
c/session.o src/hdr_idx.o src/ev_select.o src/signal.o src/acl.o src/pattern.o s
rc/memory.o src/freq_ctr.o src/auth.o src/ev_poll.o src/ev_epoll.o src/ev_sepoll
.o ebtrees/ebtree.o ebtrees/eb32tree.o ebtrees/eb64tree.o ebtrees/ebmbtree.o ebtrees/
ebsttree.o ebtrees/ebimtree.o ebtrees/ebistree.o -lcrypt
[root@localhost haproxy-1.4.22]# make install

```



```
gcc -Iinclude -Iebtree -Wall -O2 -g -fno-strict-aliasing -DTPROXY -DCONF I
G_HAP_CRYPT -DENABLE_POLL -DENABLE_EPOLL -DENABLE_SEPOLL -DNETFILTER -DUSE_GETSO
CKNAME -DCONFIG_HAPROXY_VERSION="\1.4.22\" -DCONFIG_HAPROXY_DATE="\2012/08/09\"
-c -o ebtree/ebistree.o ebtree/ebistree.c
gcc -g -o haproxy src/haproxy.o src/sessionhash.o src/base64.o src/protocols.o
src/uri_auth.o src/standard.o src/buffers.o src/log.o src/task.o src/time.o src/
fd.o src/pipe.o src/regex.o src/cfgparse.o src/server.o src/checks.o src/queue.o
src/client.o src/proxy.o src/stick_table.o src/proto_uxst.o src/proto_http.o sr
c/stream_sock.o src/appsession.o src/backend.o src/lb_chash.o src/lb_fwlc.o src/
lb_fwrr.o src/lb_map.o src/stream_interface.o src/dumpstats.o src/proto_tcp.o sr
c/session.o src/hdr_idx.o src/ev_select.o src/signal.o src/acl.o src/pattern.o s
rc/memory.o src/freq_ctr.o src/auth.o src/ev_poll.o src/ev_epoll.o src/ev_sepoll
.o ebtree/ebtree.o ebtree/eb32tree.o ebtree/eb64tree.o ebtree/ebmbtree.o ebtree/
ebstree.o ebtree/ebimtree.o ebtree/ebistree.o -lcrypt
[root@localhost haproxy-1.4.22]# make install
install -d /usr/local/sbin
install haproxy /usr/local/sbin
install -d /usr/local/share/man/man1
install -m 644 doc/haproxy.1 /usr/local/share/man/man1
install -d /usr/local/doc/haproxy
for x in configuration architecture haproxy-en haproxy-fr; do \
    install -m 644 doc/$x.txt /usr/local/doc/haproxy ; \
done
[root@localhost haproxy-1.4.22]#
```

HAProxy can be installed with ease using the process above.

6.3 Configuring settings


1) Add groups or users.

```
# groupadd haproxy
```

```
[root@localhost haproxy-1.4.22]# groupadd haproxy
```

useradd haproxy -g haproxy

```
[root@localhost haproxy-1.4.22]# useradd haproxy -g haproxy
```



2) Configure the HAProxy settings.

vi /etc/haproxy/haproxy.cfg

```
global
    log 127.0.0.1 local0
    log 127.0.0.1 local1 notice
    #log loghost local0 info
    maxconn 4096
    #debug
    #quiet
    user haproxy
    group haproxy

defaults
    log global
    mode http
    #option httplog
    option dontlognull
    retries 3
    option redispatch
    maxconn 2000
    timeout 5000
    clitimeout 50000
    srvtimeout 50000

listen WEB 192.168.10.100:7001
    mode http
    stats enable
    stats uri /haproxy
    balance roundrobin
    cookie JSESSIONID prefix
    option httpclose
    option forwardfor
    #option httpchk HEAD /check.txt HTTP/1.0
    server WEB1 192.168.10.101:7001 cookie A check
    server WEB2 192.168.10.102:7001 cookie B check

listen FTP 192.168.10.100:21
    mode tcp
    option tcplog
    balance roundrobin
    server FTP1 192.168.10.101:21 weight 10 minconn 30 maxconn 1000 check
    server FTP2 192.168.10.102:21 weight 10 minconn 30 maxconn 1000 check

listen VNC 192.168.10.100:5901
    mode tcp
    option tcplog
    server VNC1 192.168.10.101:5901 check #active
    server VNC2 192.168.10.102:5901 check backup #standby

listen VNC_Viewer 192.168.10.100:5500
    mode tcp
    option tcplog
    server VNC_Viewer1 192.168.10.101:5500 check #active
    server VNC_Viewer2 192.168.10.102:5500 check backup #standby
```

3) Configure the Hosts settings.

vi /etc/hosts

127.0.0.1	localhost.localdomain localhost
::1	localhost6.localdomain6 localhost6
192.168.10.101	webA
192.168.10.102	webB

4) Configure the Log settings.

vi /etc/sysconfig/syslog

```

SYSLOGD_OPTIONS="-m 0 -r"
KLOGD_OPTIONS="-x"
SYSLOG_UMASK=077
# vi /etc/syslog.conf
*.info;mail.none;authpriv.none;cron.none          /var/log/messages
authpriv.*                                          /var/log/secure
mail.*                                              -/var/log/maillog
cron.*                                              /var/log/cron
*.emerg                                             *
uucp,news.crit                                     /var/log/spooler
local7.*                                           /var/log/boot.log
local0.*                                           /var/log/haproxy.log
local1.*                                           /var/log/haproxy.log
    
```



```
SYSLOGD_OPTIONS="-m 0 -r"
KLOGD_OPTIONS="-x"
SYSLOG_UMASK=077
# vi /etc/syslog.conf
*.info;mail.none;authpriv.none;cron.none      /var/log/messages
authpriv.*                                     /var/log/secure
mail.*                                          -/var/log/maillog
cron.*                                          /var/log/cron
*.emerg                                         *
uuwp,news.crit                                 /var/log/spooler
local7.*                                       /var/log/boot.log
local0.*                                       /var/log/haproxy.log
local1.*                                       /var/log/haproxy.log
~
~
~
~
~
~
~
~
~
~
"/etc/sysconfig/syslog" 14L, 391C
```

5) Configure the Firewall settings.

From the HAProxy server, include the ports to forward such as 7001, 21 and 5901 in the Firewall rules. If you do not want to forward the ports, disable the Firewall function as follows.


```
# service iptables stop
```

```
[root@localhost home]# # service iptables stop
[root@localhost home]# █
```

6) Move the folder.

```
# cd examples
```


```
[root@localhost haproxy-1.4.22]# cd examples/
```



7) Register the service

```
# cp haproxy.init /etc/rc.d/init.d/haproxy
```

```
[root@localhost examples]# cp haproxy.init /etc/rc.d/init.d/haproxy
```



8) Start the service.

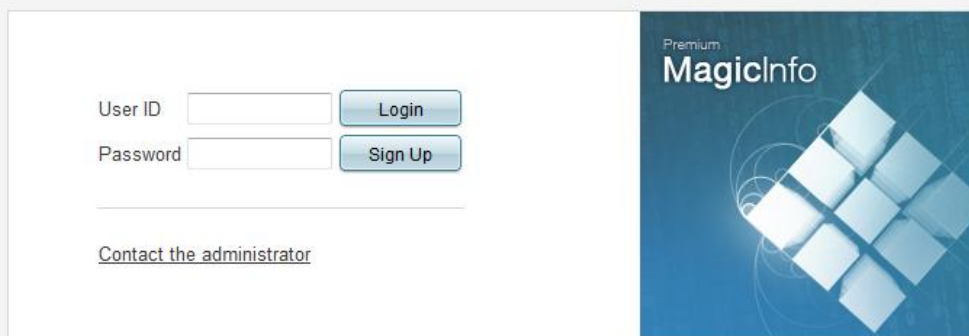
service haproxy restart

```
[root@localhost haproxy-1.4.22]# service haproxy restart
/etc/init.d/haproxy: line 1: kconfig:: command not found
Shutting down haproxy: [ OK ]
Starting haproxy: [ OK ]
[root@localhost haproxy-1.4.22]#
```

9) Check the operating status.

Go to the following address (HAProxy server IP) and check to see if MagicInfo can be accessed. If it can be accessed, the system is working properly. Distribute content to LFDs to confirm that the FTP is working properly.

<http://192.168.10.100:7001/MagicInfo/>



Copyright 2009 - 2013 SamsungElectronics All rights reserved.

7 Installing and constructing Memcached

Memcached is a distributed cache server and used in the large-size server structure.

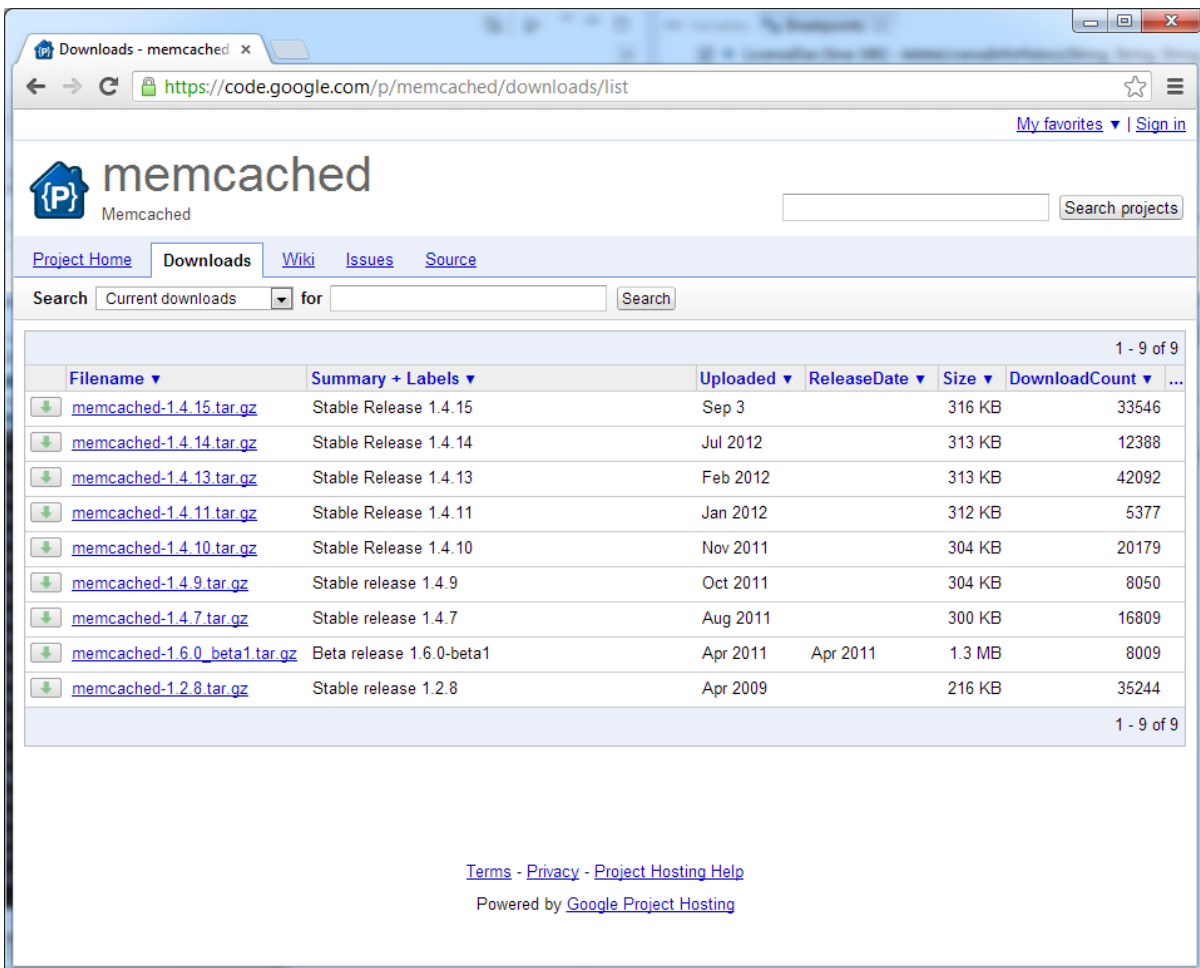
7.1 Memcached settings (Linux)

7.1.1 Install preparations

Download Memcached and Libevent from the following sites respectively.

1) Downloading Memcached

<http://code.google.com/p/memcached/downloads/list>



2) Downloading Libevent

<http://libevent.org/>



7.1.2 Installing

1) Installing Libevent

```
tar xzvf libevent-2.0.19-stable.tar.gz
cd libevent-2.0.19-stable
./configure
make; make install
```

2) Installing Memcached

```
tar xzvf memcached-1.4.14.tar.gz
cd memcached-1.4.14
./configure -prefix=/usr/local/memcached -with-libevent=/usr/local/
make; make install
```

7.1.3 Configuring the settings

1) Running the software

Run the software to configure the cache memory and port settings using the following options.

```
./bin/memcached -d -m 2048 -l 0.0.0.0 -p 11211 -u <userid>
```

- Run with d daemon
- Allot m cache memory (e.g., 2048 -> 2Gb)
- l IP binding is 0.0.0.0
- p port is 11211
- User account to run u

2) Operation test

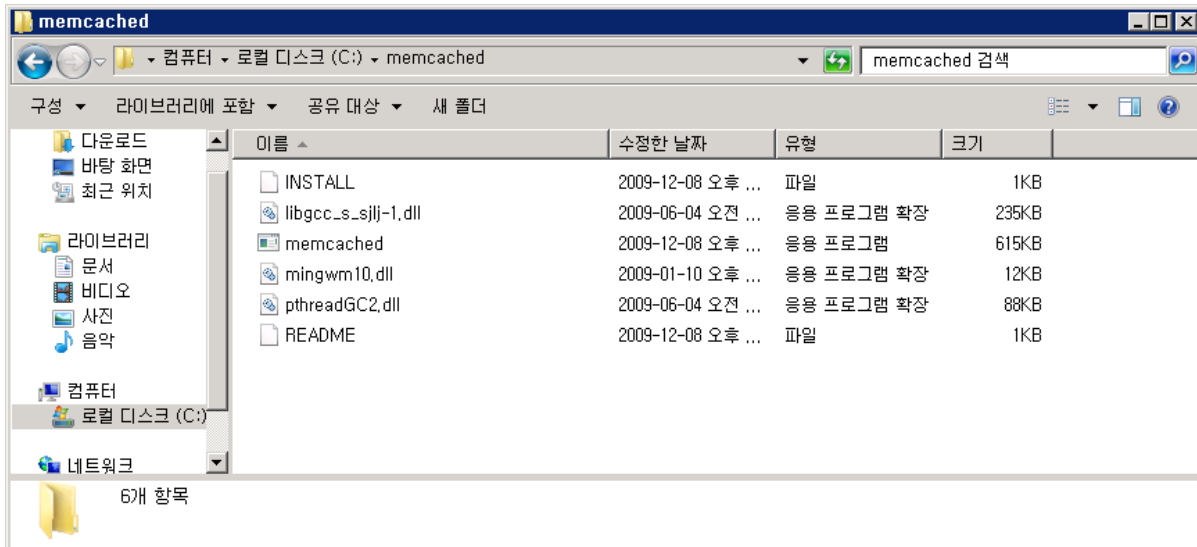
Check the memcached operation using telnet. If the server cannot be accessed, open the corresponding port using the Firewall settings on the Linux server or disable the Firewall service to check the operation status.

```
[root@localhost memcached-1.4.14]# ./memcached -d -m 1024 -l 0.0.0.0 -p 11211 -u
memcached
[root@localhost memcached-1.4.14]# telnet 127.0.0.1 11211
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
stats
STAT pid 5960
STAT uptime 10
STAT time 1363066284
STAT version 1.4.14
STAT libevent 2.0.19-stable
STAT pointer_size 32
STAT rusage_user 0.000000
STAT rusage_system 0.008998
STAT curr_connections 5
STAT total_connections 6
STAT connection_structures 6
STAT reserved_fds 20
STAT cmd_get 0
STAT cmd_set 0
STAT cmd_flush 0
STAT cmd_touch 0
STAT get_hits 0
```

7.2 Memcached settings (Windows)

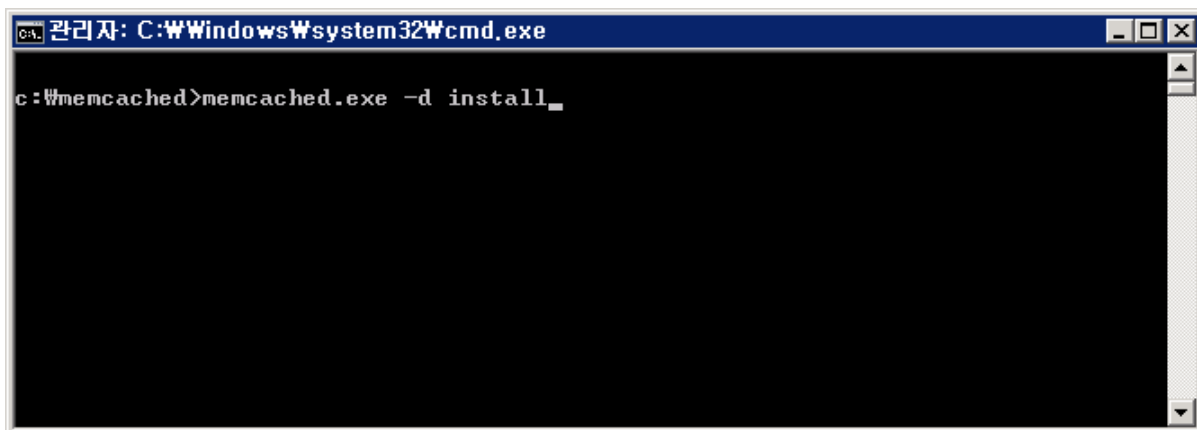
7.2.1 Install preparations

Download and unzip the Windows version of memcached 1.4.



7.2.2 Installing

Open the command window using the cmd command. Install the service using the memcached.exe -d install command.



If msvc71.dll is not available and installation is not possible, copy msvc71.dll to c:\windows\system32\ (msvc71.dll can be searched and then downloaded.)

- Delete : c:\memcached>memcached.exe -d uninstall
- ※ Refer to c:\memcached>memcached -help for further information.

7.2.3 Configuring the settings

```
C:\memcached>memcached.exe -d start
C:\memcached>memcached.exe -d stop
```

- (1) If memcached is run, it will listen on port 11211.
This can be found through c:\>netstat -na.

Check to see if the corresponding port is accessible with the following command. If not accessible, add port 11211 to the firewall.

c:\>telnet 127.0.0.1 11211

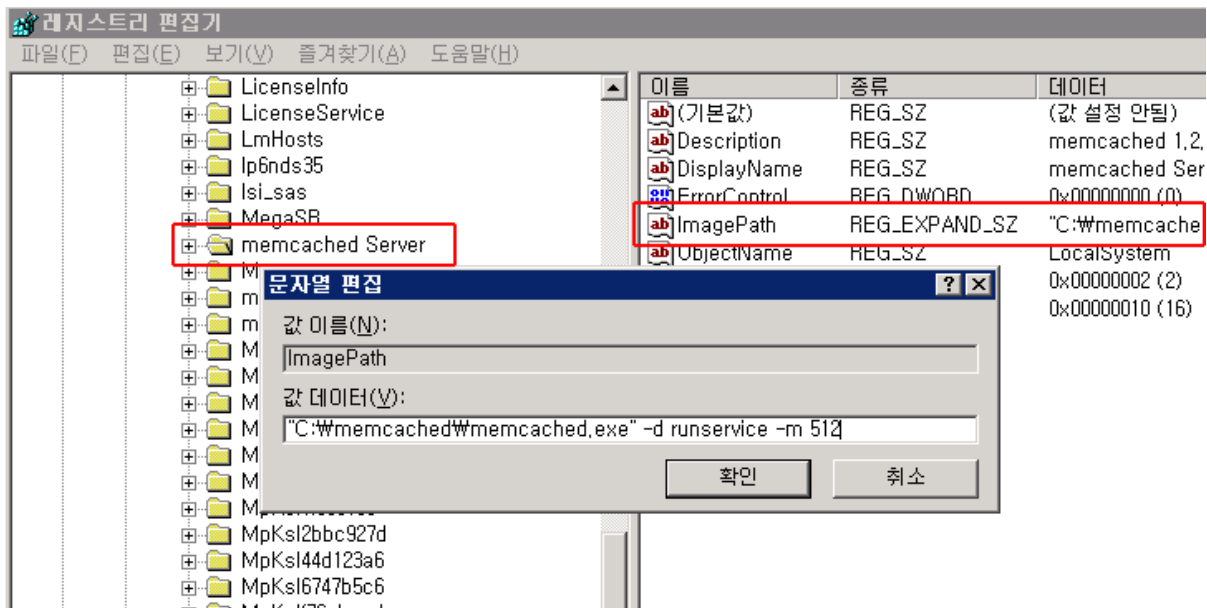
(2) Server memory is limited to 64MB. It is possible to edit the registry to configure the maximum memory limit.

- Run > regedit
- Edit ImagePath for

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\memcached Server

- as follows:

"C:\memcached\memcached.exe" -d runservice -m 512 (set to 512MB)



7.3 Configuring the memcached settings from MagicInfo

To use the memcached server, configure the memcached server settings from config.properties used for MagicInfo. After changing the settings as follows, restart the server.

```
#####
# cache option
# options : db, memcached (experiment), jndi
# - jndi(memory) : only 1 server (default)
# - db : mutple servers 2~5 under 5000 units
# - memcache : more servers over 5000 units
#####
cache.type = memcached

#####
# memcached Configuration (experiment)
# ex) memcached.url = 192.168.1.100:11211 192.168.1.101:11211
#####
memcached.use = Y
memcached.url = 192.168.1.103:11211
```

If multiple memcached servers are used, configure the settings as follows.

```
#####  
# cache option  
# options : db, memcached (experiment), jndi  
# - jndi(memory) : only 1 server (default)  
# - db : mutiple servers 2-5 under 5000 units  
# - memcache : more servers over 5000 units  
#####  
cache.type = memcached  
  
#####  
# memcached Configuration (experiment)  
# ex) memcached.url = 192.168.1.100:11211 192.168.1.101:11211  
#####  
memcached.use = Y  
memcached.url = 192.168.1.103:11211 192.168.1.104:11211
```

8 VNC tunneling server settings

8.1 For the small-size structure

If two or more Tomcat servers are used, use the VNC tunneling server on only one of the Tomcat servers or set up the Tomcat servers in an active-standby configuration.

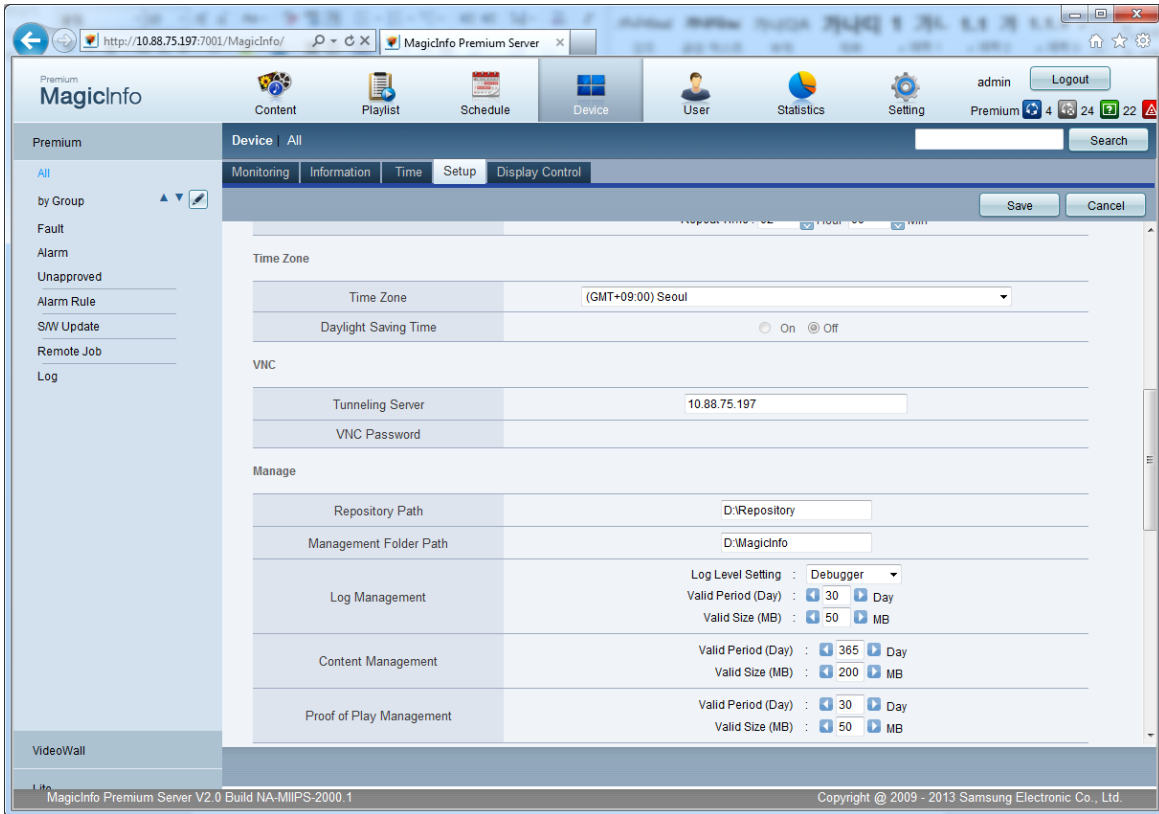
Install the VNC tunneling server on a separate server, on one of the Tomcat servers, or on each server. After installing the VNC tunneling server, from MagicInfo Premium i Player, enter the VNC tunneling server IP address or the address set from haproxy as the tunneling server IP address.

- Installing the server
 - If multiple Tomcat servers are used, install the tunneling server on a separate server or one of the Tomcat servers.
On Tomcat servers, it should be installed automatically.
 - Configure the settings to forward the VNC port to the VNC tunneling server from HAProxy.

```
listen VNC 192.168.10.100:5901
    mode tcp
    option tcplog
    server VNC1 192.168.10.101:5901 check #active
    server VNC2 192.168.10.101:5901 check backup #standby

listen VNC_Viewer 192.168.10.100:5500
    mode tcp
    option tcplog
    server VNC_Viewer1 192.168.10.101:5500 check #active
    server VNC_Viewer2 192.168.10.101:5500 check backup #standby
```

- It is possible to change the tunneling server address from i Player. Alternatively, go to [Device]-[Setup]-[tunneling server address] in Premium Server and remotely change the address.



The following shows a simple structure for a VNC connection.

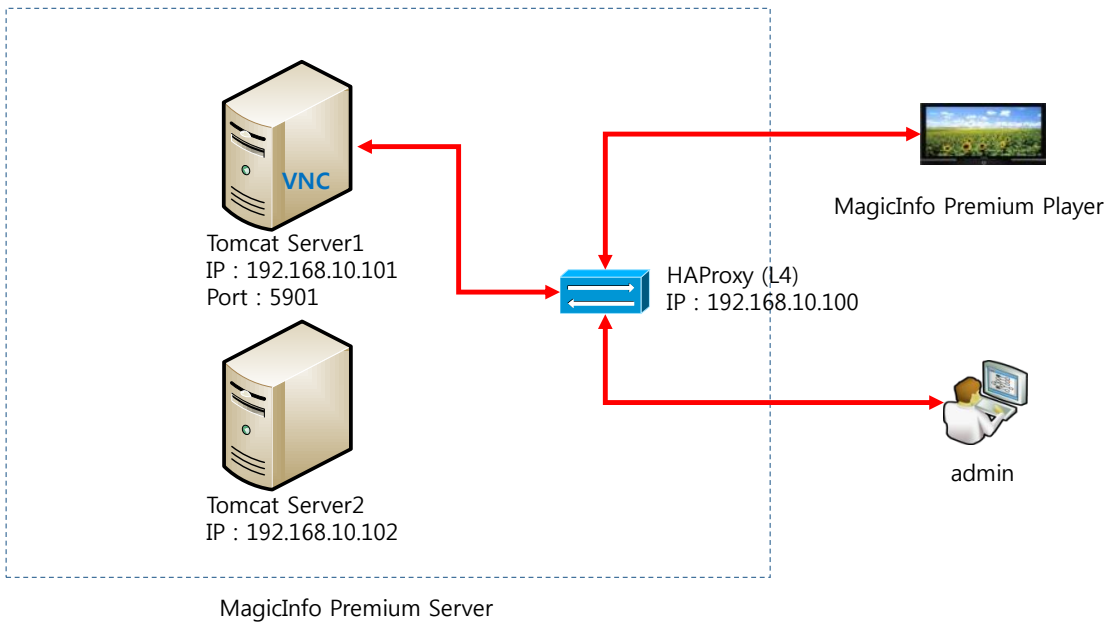


Figure 2 Structure with one VNC server

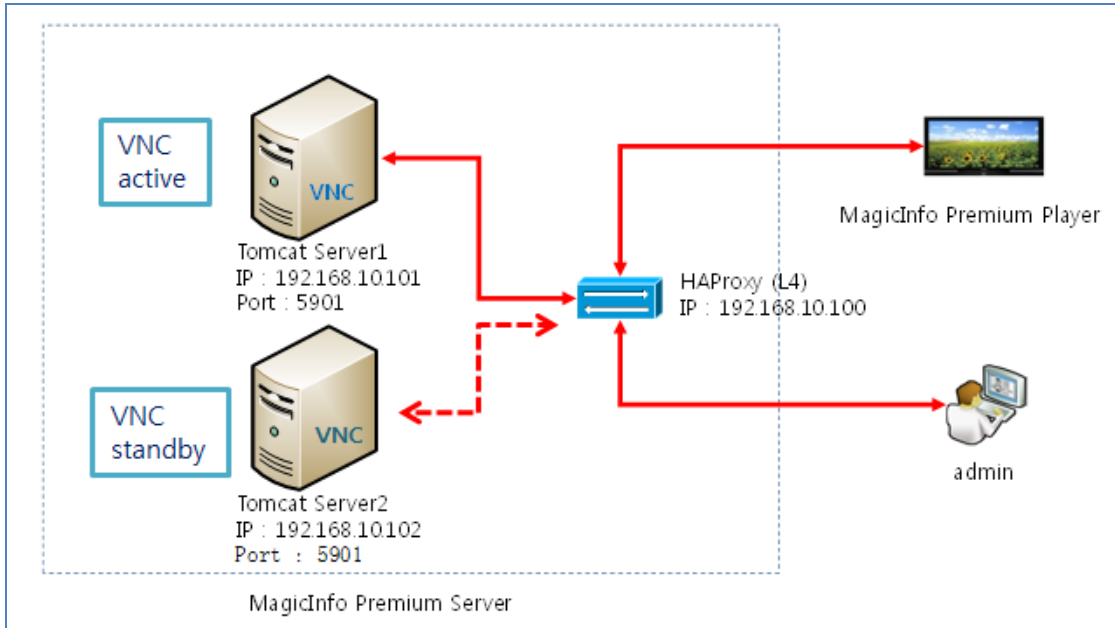


Figure 3 Configuration with two or more VNC servers (Active - Standby)

8.2 For the large-size structure

If tens of thousands of LFDs or more are used in the large-size server structure, it is not possible to manage the LFDs from a single VNC server. More than one VNC server is required in this case. However, the clustering method will not work and you have to manually distribute the LFDs so that they can access different VNC servers respectively.

For example, if 10,000 LFDs are used, configure 5,000 of them for VNC server 192.168.10.101 and the rest for VNC server 192.168.10.102.

- Installing the server
 - It is possible to change the tunneling server address from i Player. Alternatively, go to [Device]-[Setup]-[tunneling server address] in Premium Server and remotely configure the settings.
 - No settings are required from HAProxy (delete port 5901 from the forward ports).

Find below the structure.

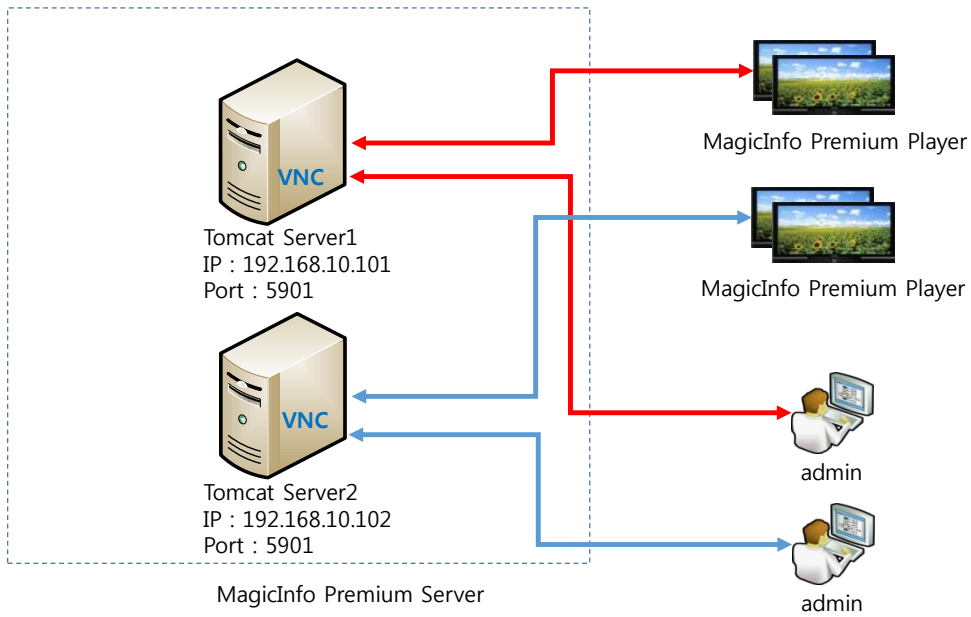


Figure 4 Structure with multiple VNC servers

9 Constructing a storage server

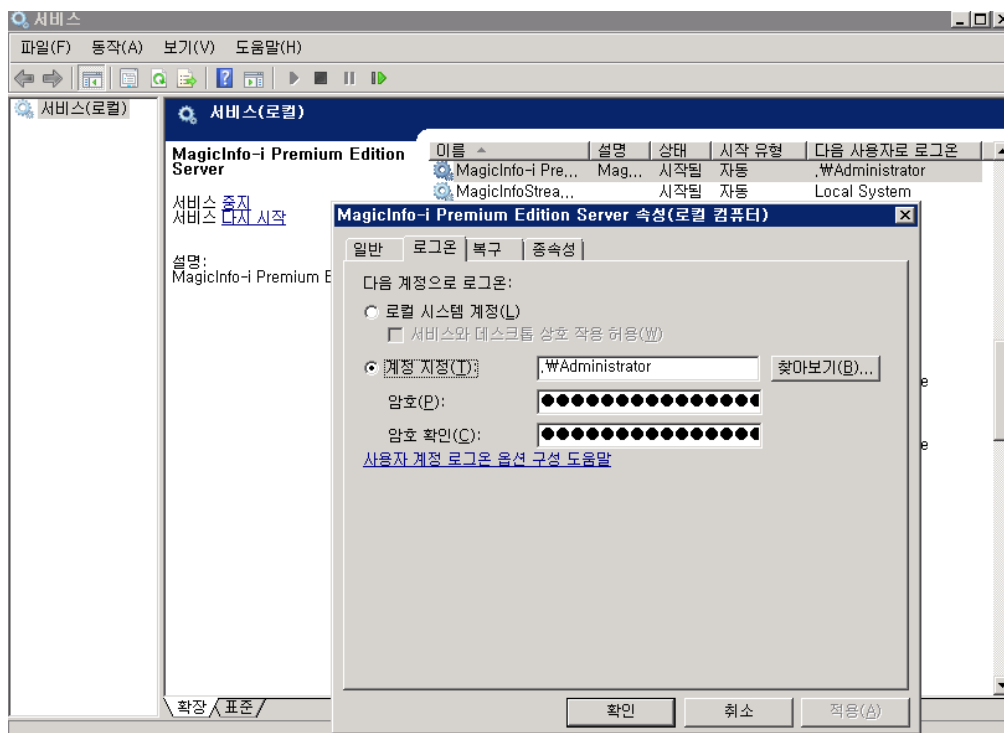
Content management is one of the important roles of MagicInfo Premium Server. If more than one Tomcat server is used, content should be managed from a single storage server.

Storage server products vary and allow you to configure settings to share content folders from each Tomcat server using the mklink function in Windows. It is recommended to use a separate storage server, considering the server performance and management.

- 1) Constructing the storage server
 - A. Prepare a Windows server.
(IP example: 192.168.10.104)
 - B. Create a folder to set it as a shared folder.
(Path example: \\192.168.10.104\runtime)
- 2) Setting the passwords for the Tomcat server and Storage server
 - A. Configuring the MagicInfo service logon settings for the Tomcat server

Make sure the user password for WAS is identical to that of Network Drive.

Go to [Control Panel] - [Administrative Tools] - [Services]. Set the service password for the MagicInfo-i Premium server.

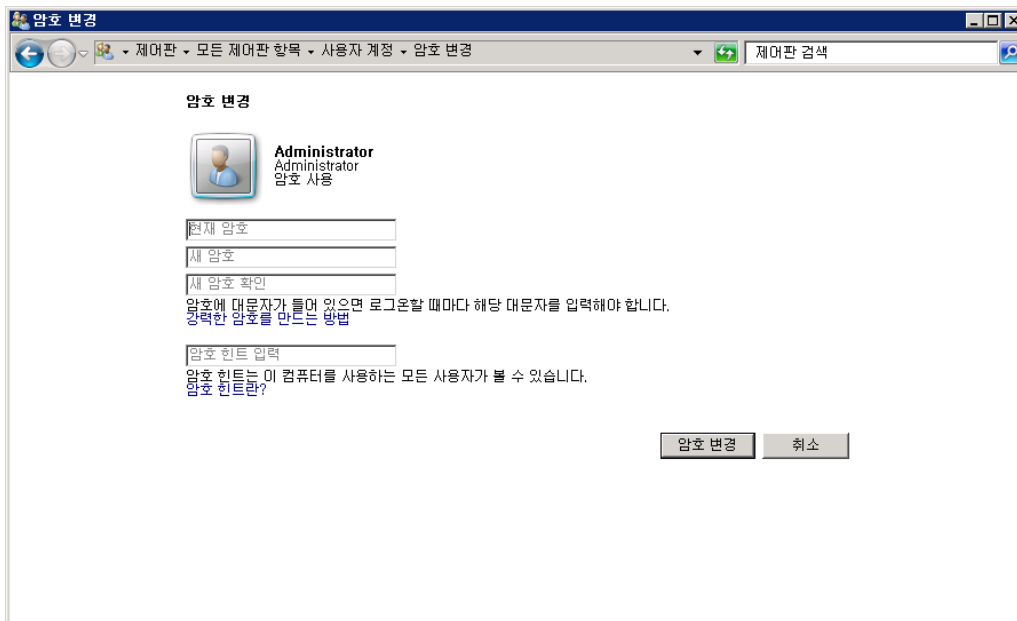


- Right-click on the service - Select [Properties]
- Go to the [Log On] tab and select [This account]. Enter the ID (administrator) and password. Select OK to apply the changes.

- B. Storage server password setting
Remotely access the server assigned to the network drive. Change the administrator password to the password of the Tomcat server.

Go to [Control Panel] - [User Accounts] - [Change your password].

Change the password. Select [OK] to apply the change.



- 3) Specifying the shared folder on a Tomcat server
 - A. Stop the Tomcat server.
 - B. Rename the "C:\MagicInfo-i Premium\runtime" folder.
(E.g., runtime -> runtime_back)
 - C. Open the cmd window from each Tomcat server. Enter the commands.
The following commands link a folder from another server to the current folder.

```
mklink /D "runtime" "\\192.168.10.104\runtime"
```

```

Administrator: Command Prompt
C:\MagicInfo-i Premium>dir/w
Volume in drive C has no label.
Volume Serial Number is 22B0-4D72

Directory of C:\MagicInfo-i Premium

[.]          [..]          [bin]          [conf]          [ddl]
[runtime_back] [server]
              0 File(s)          0 bytes
              7 Dir(s) 499,677,167,616 bytes free

C:\MagicInfo-i Premium>mklink /D "runtime" "www192.168.10.104\runtime"
    
```

- D. Copy the contents of the "runtime_back" folder to the "runtime" folder.
- E. Start the Tomcat server.

10 JVM memory tuning

Manually configure the settings using the suitable JVM memory options according to server performance.

1) JVM memory settings

- A. Open the MagicInfo install folder\tomcat\bin\service.bat file from editor.
(ex: c:\MagicInfo Premium\tomcat\bin\service.bat)

It is possible to configure the JVM PermSize and HeapSize settings. Configure the settings according to server performance.

About options

```
-XX:PermSize=256m : Initial PermSize
-XX:MaxPermSize=256m : Maximum PermSize
-Xms1024m : Initial Heap Memory Size of JVM
-Xmx1024m : Maximum Heap Memory Size of JVM
--JvmMs 1024 : Initial Heap Memory Size of the service
--JvmMx 1024 : Maximum Heap Memory Size of the service
```

Original file content

```
"%EXECUTABLE%" //US//%SERVICE_NAME% ++JvmOptions "-
Djava.io.tmpdir=%CATALINA_BASE%\temp;-
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager;-
Djava.util.logging.config.file=%CATALINA_BASE%\conf\logging.properties;-
XX:PermSize=128m;-XX:MaxPermSize=128m" --JvmMs 256 --JvmMx 512 --
Startup auto
```

Edited file content

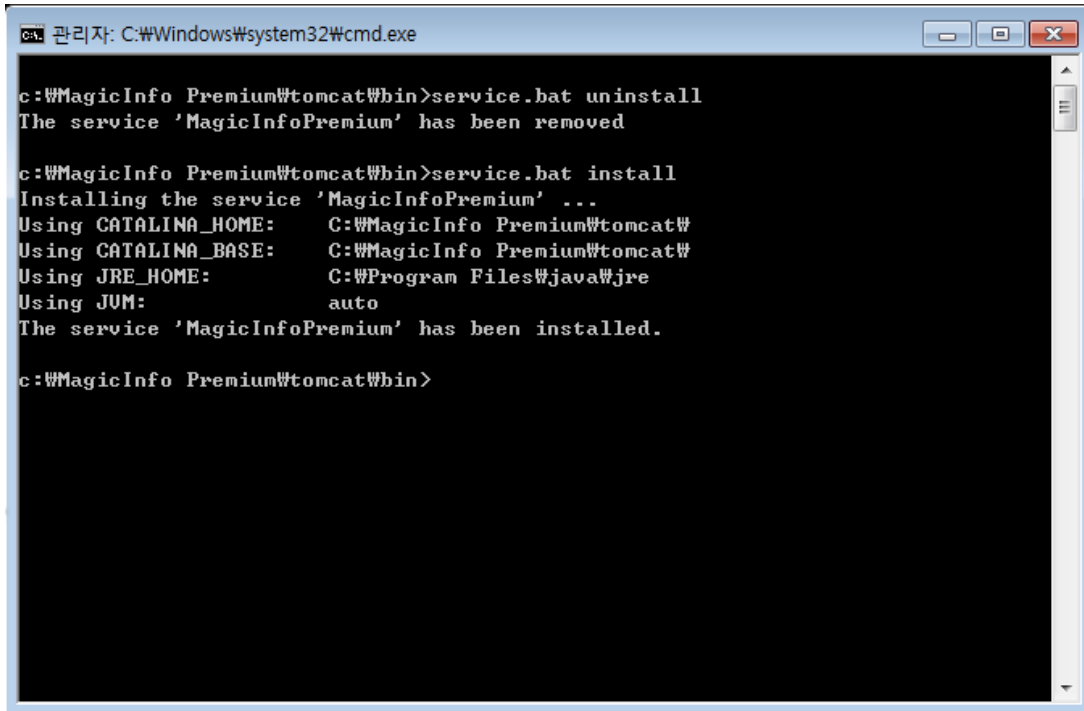
```
"%EXECUTABLE%" //US//%SERVICE_NAME% ++JvmOptions "-
Djava.io.tmpdir=%CATALINA_BASE%\temp;-
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager;-
Djava.util.logging.config.file=%CATALINA_BASE%\conf\logging.properties;-
Xms1024m;-Xmx1024m;-XX:PermSize=256m;-XX:MaxPermSize=256m" --
JvmMs 1024 --JvmMx 1024 --Startup auto
```

2) Installing the service

Run a command prompt. Uninstall or install the service.

```
c:\MagicInfo Premium\tomcat\bin>ervice.bat uninstall
```

```
c:\MagicInfo Premium\tomcat\bin>service.bat install
```



```
관리자: C:\Windows\system32\cmd.exe

c:\MagicInfo Premium\tomcat\bin>service.bat uninstall
The service 'MagicInfoPremium' has been removed

c:\MagicInfo Premium\tomcat\bin>service.bat install
Installing the service 'MagicInfoPremium' ...
Using CATALINA_HOME:    C:\MagicInfo Premium\tomcat\
Using CATALINA_BASE:    C:\MagicInfo Premium\tomcat\
Using JRE_HOME:         C:\Program Files\java\jre
Using JUM:               auto
The service 'MagicInfoPremium' has been installed.

c:\MagicInfo Premium\tomcat\bin>
```

11 FAQ

- 1) **When using multiple Tomcat servers, which server (MAC address) should be used to obtain a MagicInfo license?**

If multiple Tomcat servers are used, it is possible to obtain a license regardless of the server.

For example, if Tomcat1 and Tomcat2 servers are used, obtain a license based on the MAC address of either the Tomcat1 server or Tomcat2 server. Then, enter the obtained license.

- 2) **In the occurrence of a failure with one of two Tomcat servers, is it possible to continue using the server?**

If one server encounters a failure and the other is working properly, it is possible to continue using the working server. However, this can cause excessive load on the single working server. It is best to fix the problematic server as soon as possible.

- 3) **Is it possible to use a streaming server when multiple Tomcat servers are used?**

It will work but is not recommended. The reason is because it can cause excessive load on the Tomcat server and greatly affect server performance. This will result in degradation in the performance of all the connected devices. At present, a streaming server is recommended only for a small-size structure with 50 clients or less.

- 4) **If there are 5,000 clients or less and the customer wants a stable server operation, what is the best setup?**

The small-size structure uses only one Tomcat server. There may be a risk of downtime if the server fails to function properly. If the customer wants a more stable server operation, suggest the large-size structure and provide them with necessary advice.

- 5) **The network condition is poor and we want to operate content download servers in different locations. Is this possible?**

Unfortunately, this feature is not available. There is no plan to develop one for the time being.