High Availability
And
Disaster Recovery
Trademarks

Deepnet Unified Authentication, MobileID, QuickID, PocketID, SafeID, GridID, FlashID, SmartID, TypeSense, VoiceSense, MobilePass, DevicePass, RemotePass and Site Stamp are trademarks of Deepnet Security Limited. All other brand names and product names are trademarks or registered trademarks of their respective owners.

Copyrights

Under the international copyright law, neither the Deepnet Security software or documentation may be copied, reproduced, translated or reduced to any electronic medium or machine readable form, in whole or in part, without the prior written consent of Deepnet Security.

Licence Conditions

Please read your licence agreement with Deepnet carefully and make sure you understand the exact terms of usage. In particular, for which projects, on which platforms and at which sites, you are allowed to use the product. You are not allowed to make any modifications to the product. If you feel the need for any modifications, please contact Deepnet Security.

Disclaimer

This document is provided “as is” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

This document could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the document. Deepnet Security may make improvements of and/or changes to the product described in this document at any time.

Contact

If you wish to obtain further information on this product or any other Deepnet Security products, you are always welcome to contact us.

Deepnet Security Limited
Northway House
1379 High Road
London N20 9LP
United Kingdom

Tel:   +44(0)20 8343 9663
Fax:   +44(0)20 8446 3182
Web:   www.deepnetsecurity.com
Email: support@deepnetsecurity.com
# Table of Contents

Introduction ..................................................................................... 4
Architecture ..................................................................................... 4
    Scenario 1 ................................................................................................................................... 4
    Scenario 2 ................................................................................................................................... 5
    Scenario 3 ................................................................................................................................... 5
    Scenario 4 ................................................................................................................................... 6
Installation ...................................................................................... 7
    Install the Primary Server ................................................................................... 7
        Scenario 1 & 2 ............................................................................................................................. 7
        Scenario 3 & 4 ............................................................................................................................. 7
    Install the Replica Servers ................................................................................... 8
        Prerequisites ................................................................................................................................ 8
        Installation .................................................................................................................................. 8
    Configure SQL Servers ........................................................................................ 9
        Scenario 2: Master/Slave............................................................................................................... 9
        Scenario 3/4: Master/Master ........................................................................................................ 9
Disaster Recovery ............................................................................. 9
Appendix A: MySQL Master-Master Replication ................................... 10
    Prerequisites ................................................................................................... 10
    Preparation ..................................................................................................... 10
    Configuration................................................................................................. 10
Introduction

Availability refers to the percentage of time that a system is available to a user. When increasing the level of availability, your goal is to attain the level of “high availability” (HA) or “fault tolerant” availability. HA systems must be maintainable so that administrators can service a failed component in the system without shutting down the entire operation.

DualShield is designed to avoid “single-point failure”. A customer can deploy multiple instances of the DualShield servers to achieve high availability. After adding and configuring multiple instances of the DualShield servers in your network, you can configure failover and balance the load across all instances of the DualShield servers.

To do so, you must place a load-balancer in front of the cluster of multiple instances of the DualShield servers.

Architecture

DualShield servers store data in a SQL database which is provided by any commercial SQL database server such as MySQL, MS-SQL or Oracle etc. Depending on your network environments and requirements, you can set up a HA system in various architectures. Below are some examples:

Scenario 1

Multiple DualShield servers with a single SQL server

In this scenario, we deploy multiple copies of the DualShield servers at a single site. All DualShield servers must be perfect replica of the primary DualShield server, and they are all connected to one single SQL database server.

A load balancer fronts the cluster of DualShield servers to distribute the traffic. All DualShield servers are active at any time. Should a DualShield server cease to function, the load balancer will stop distributing the traffic to the server.
Scenario 2

Multiple DualShield servers with a SQL failover cluster

This scenario is an enhanced model of the previous scenario. Instead of one SQL server which is subject to single-point failure, we create a master/slave failover cluster of two SQL servers. Similarly, we add a secondary load balancer to a failover cluster of load balancer.

Scenario 3

Multiple DualShield servers with multiple SQL servers in a single site

In Scenario 2, there are two SQL servers installed but only one of them is active at any time as they work in the active/passive model. In this scenario, we create a cluster of SQL servers that are all active at any time (master/master). This requires that all SQL servers are perfect replica of the primary SQL server and they are configured to real-time replication.
Scenario 4

Multiple DualShield servers with multiple SQL servers in multiple sites

This scenario demonstrates how you can deploy DualShield system in different, multiple geographically separated locations. For the simplicity of illustration, the diagram displays only one DualShield server at each site. You can, if needed, install multiple DualShield servers at each site as demonstrate in Scenario 1.
Installation

Install the Primary Server

Scenario 1 & 2
To deploy a HA system for the DualShield platforms in a model similar to scenario 1 or 2, in which multiple DualShield servers are connected to a single SQL database server or a single SQL failover cluster, you must install the primary DualShield platform in such way that the DualShield server is connected to an external SQL database server. In other words, you must first install a separate SQL database server or utilise an existing SQL database server in your network.

To install the primary server, follow the installation procedure as detailed in the “DualShield Authentication Platform - Installation Guide”. At the step 8: ”Database Connection”, select the option: “Use an existing SQL server”.

Scenario 3 & 4
To deploy a HA system for the DualShield platforms in a model similar to scenario 3 or 4, in which each DualShield server is connected to a separate SQL database server, you can install the primary DualShield server with connection to either an external SQL database server, or the internal MySQL database server that is included in the installation package. However, only MySQL is recommended and supported because of the simplicity in setting up master-master replication model with MySQL. Although it is possible to set up master-master replication with other types SQL servers, we do not currently provide technical support for those types of SQL servers.
Therefore, the simplest way is to install the primary DualShield server together with the MySQL server included in the package. To install the primary DualShield server, follow the installation procedure as detailed in the "DualShield Authentication Platform - Installation Guide". At the step 8: “Database Connection”, select the option: “Install a copy of MySQL server”.

Install the Replica Servers

DualShield includes a utility program, `dsclone`, that enables you to easily create a backup image of the primary DualShield server and install replica servers from the image. If the internal MySQL server was also installed when the primary DualShield was installed, then the MySQL server and database will be also included in the backup image and will be installed on the replica server machines.

Prerequisites

1. The replica server machine should have similar hardware and OS to the primary machine.
2. Windows OS on both servers must have the same drive letter and the same path, e.g. c:\windows

Installation

1. unzip "dsrvclone.zip" to a folder, e.g. c:\temp where it is readable and writable;
2. open windows console with administrator privilege, go to the folder, e.g. c:\temp;
3. run dsrvclone.exe. It will self-extract its contents to a subfolder called "clonepack";
4. go into the subfolder, e.g. c:\temp\clonepack
5. in the command line, run "dsclone backup";
6. wait it to finish;
7. copy the entire folder, e.g. c:\temp\clonepack, to the replica machine;
8. run "dsclone deploy" on the replica machine and follow the instructions.

Configure SQL Servers

Scenario 2: Master/Slave
If you have deployed DualShield HA system in a model similar to scenario 2, you will need to configure your SQL servers to work in Master/Slave model.

For MySQL, please refer to the following article:
http://xorl.wordpress.com/2011/03/13/how-to-mysql-masterslave-replication/

For other types of SQL server, please refer to the technical documents of the software.

Scenario 3/4: Master/Master
If you have deployed DualShield HA system in a model similar to scenario 3 or 4, you will need to configure your SQL servers to work in Master/Master model.

For MySQL, if you decided to configure it manually please refer to the following article:
http://www.neocodesoftware.com/replication/

Or, you can use a utility provided by Deepnet Security called MYSQLMM to configure it for you automatically. Please refer to the Appendix A: MySQL Master-Master Replication.

For other types of SQL server, please refer to the technical documents of the software.

Disaster Recovery
In the event when a DualShield server machine, either the primary or a replica, has ceased to function and has to be completed rebuilt, you can easily reinstall the DualShield server from the backup image created by the DSCLONE utility.

It is important to note that, however, the backup image created by DSCLONE does not contain any data in the DualShield database. You must regularly backup your SQL database using the backup function provided by your SQL server.
Appendix A: MySQL Master-Master Replication

In this tutorial we will create a dual master (master-master) MySQL replication cluster for DualShield using a utility program called MySQLMM provided by Deepnet Security.

We'll call the two MySQL servers **Server A** and **Server B**. In a dual master setup each server functions as both a master and a slave to the other server.

**Prerequisites**

A copy of DualShield (with MySQL) has been successfully installed on Server A and operating as the primary server, and a copy of DualShield has been successfully cloned on Server B from Server A using the DualShield clone utility.

**Preparation**

Copy MySQLMM.zip file to Server A, unzip its contents to a folder, for instance, c:\temp. This folder must be **read/write-able**. There 6 files in the package.

![Folder structure with files]

**Configuration**

1. Stop the DualShield service on both Server A and B (**VERY IMPORTANT!!!**)

2. Configure MySQL on Server B so that it is accessible from Server A.

2.1 Go to **Server B**

2.2 Launch Windows Console (Run as Administrator)

2.3 Change to MySQL bin folder, e.g.

   C:\Program Files\Deepnet DualShield\mysql\bin

2.3 Enter the following commands:

   ```
   $ mysql -u root -p
   Enter password: MySQL-Root-Password
   ```
3. Check that MySQL on Server B is accessible from Server A.

3.1 Go to **Server A**
3.2 Launch Windows Console (Run as Administrator)
3.3 Change to MySQL bin folder, e.g.

```
C:\Program Files\Deepnet DualShield\mysql\bin
```

2.3 Enter the following commands:

```
mysql -h Server-B-IP-Address -u root -p
```

Enter password: **MySQL-Root-Password**

Below is an example:

4. Change to the folder where MySQLMM has been unpacked. Enter the command below:
MySQL Dual Master Replication Setup

```
mysqldualmaster.bat Server-A-IP-Address Server-B-IP-Address "MySQL-Root-Password" "DualShield-Path"
```

Notice that MySQL-Root-Password and DualShield-Path are enclosed by quotes. Also, do not provide 127.0.0.1 as the IP address of Server A.

---

Next it reminds you that DualShield service must be stopped on both servers before configuring MySQL master-master replication setup.

---

It also reminds you that MySQL on server B must be made accessible from server A.

---

Finally, it prompts you to confirm that the parameters that you provided are correct.

---

The script will now carry jobs such as backup database on server A, restore it on server B, create slave user on both servers, etc.

---

It will have also modified a file called `das.ini` on server A that is located in the "mysql" sub folder in the DualShield folder. However, it won’t be able to modify the `das.ini` file on server B. You will have to modify it manually. For your convenience, it generates the necessary part in the file `dasB.iniappend` in the folder where MYSQLMM resides. You just need to append the content of `dasB.iniappend` to the `das.ini` file on server B.
On Server A, in the folder where MYSQLMM resides:

```
server-id = 2
replicate.same-server-id = 0
auto-increment.increment = 2
auto-increment.offset = 2
master-host = 192.168.230.3
master-user = dsrobot
master-password = changeme
master-connect-retry = 60
replicate-do-db = dualshield
log-bin = c:\Program Files\Deepnet DualShield\mysql\log-bin.log
binlog-do-db = dualshield
```

On Server B, in the "MySQL" subfolder where DualShield is installed:

```
server-id = 2
replicate.same-server-id = 0
auto-increment.increment = 2
auto-increment.offset = 2
master-host = 192.168.311.200
master-user = dsrobot
master-password = changeme
master-connect-retry = 60
replicate-do-db = dualshield
log-bin = c:\Program Files\Deepnet DualShield\mysql\log-bin.log
binlog-do-db = dualshield
```

After editing the das.ini file on Server B, **You must restart the MySQL service on server B (The service name is MySQL(DUAL))** before returning to server A to continue the script execution.

Type "yes" to let the script do the rest.

You should see:

```
-- Restart MySQL service on server A --
The MySQL(DUAL) service is stopping.
The MySQL(DUAL) service was stopped successfully.

The MySQL(DUAL) service is starting.
The MySQL(DUAL) service was started successfully.

-- Wait for 5 seconds for fully starting of MySQL service --

-- Generate sync script for server A --
-- Execute sync script on server A --
```

<table>
<thead>
<tr>
<th>File</th>
<th>Position</th>
<th>Binlog_Do_DB</th>
<th>Binlog_Ignore_DB</th>
</tr>
</thead>
</table>

log-bin.000001  98      dualshield

-- Generate sync script for server B --
-- Execute sync script on server B --

File    Position        Binlog_Do_DB    Binlog_Ignore_DB
log-bin.000001  98      dualshield

-- Wait another 10 seconds for synchronization --

-- Show slave status on server A --
*************************** 1. row **************** ***********
Slave_IO_State: Waiting for master to send event
                Master_Host: 192.168.230.97
                Master_User: dsrobot
                Master_Port: 3306
                Connect_Retry: 60
                Master_Log_File: log-bin.000001
                Read_Master_Log_Pos: 98
                Relay_Log_File: nanoart-win2k31-relay-bin.000002
                Relay_Log_Pos: 233
                Relay_Master_Log_File: log-bin.000001
                Slave_IO_Running: Yes
                Slave_SQL_Running: Yes
                Replicate_Do_DB: dualshield
                Replicate_Ignore_DB: dualshield
                Replicate_Do_Table: dualshield
                Replicate_Ignore_Table: dualshield
                Replicate_Wild_Do_Table: dualshield
                Replicate_Wild_Ignore_Table: dualshield
                Last_Errno: 0
                Last_Error: 
                Skip_Counter: 0
                Exec_Master_Log_Pos: 98
                Relay_Log_Space: 233
                Until_Condition: None
                Until_Log_File: 
                Until_Log_Pos: 0
                Master_SSL_Allowed: No
                Master_SSL_CA_File: 
                Master_SSL_CA_Path: 
                Master_SSL_Cert: 
                Master_SSL_Cipher: 
                Master_SSL_Key: 
                Seconds_Behind_Master: 0
-- Show slave status on server B --
*************************** 1. row **************** ***********
Slave_IO_State: Waiting for master to send event
                Master_Host: 192.168.230.3
                Master_User: dsrobot
                Master_Port: 3306
                Connect_Retry: 60
                Master_Log_File: log-bin.000001
                Read_Master_Log_Pos: 98
                Relay_Log_File: nanoart-win2k33-relay-bin.000002
                Relay_Log_Pos: 233
                Relay_Master_Log_File: log-bin.000001
                Slave_IO_Running: Yes
                Slave_SQL_Running: Yes
                Replicate_Do_DB: dualshield
                Replicate_Ignore_DB: dualshield
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
  Last_Errno: 0
  Last_Error:
  Skip_Counter: 0
  Exec_Master_Log_Pos: 98
  Relay_Log_Space: 233
  Until_Condition: None
  Until_Log_File:
  Until_Log_Pos: 0
  Master_SSL_Allowed: No
  Master_SSL_CA_File:
  Master_SSL_CA_Path:
  Master_SSL_Cert:
  Master_SSL_CIPHER:
  Master_SSL_Key:
  Seconds_Behind_Master: 0

If you see the following statements:

  Slave_IO_Running: Yes
  Slave_SQL_Running: Yes

Then the Master-Master replication setup has been successful.

Finally, start the DualShield service on both servers.

Reference


http://www.nparikh.org/unix/mysql.php