Information Management software

Automated Cluster Controlled HADR (High Availability Disaster Recovery) Configuration Setup using the IBM DB2 High Availability Instance Configuration Utility (db2haicu)

June 2009

Authors:
Steve Raspudic, IBM Toronto Lab (stevera@ca.ibm.com)
Saurabh Sehgal, IBM Toronto Lab (ssehgal@ca.ibm.com)
Malaravan Ponnuthurai, IBM Toronto Lab (malarava@ca.ibm.com)

Table of Contents

1. Introduction and Overview

This paper describes two distinct configurations of an automated IBM® DB2® for Linux®, UNIX®, Solaris, AIX® and Windows® failover solution. The configurations are based on the DB2 High Availability Disaster Recovery (HADR) feature and the DB2 High Availability Instance Configuration Utility (db2haicu) available with DB2 Version 9.7.

Target Audience for this paper:

- DB2 database administrators
- Disaster recovery architects
- High availability architects

2. Before you begin

Below you will find information on knowledge requirements, as well as hardware and software configurations used to set up the topology depicted in Sections 4 and 5 below. It is important that you read this section prior to beginning any setup.

2.1 Knowledge Prerequisites

- Understanding of DB2 9.7 and HADR*
- Understanding of High Availability and Disaster Recovery concepts
- Basic understanding of operating system concepts

*Information on DB2 HADR can be found here:
http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.admin.ha.doc/doc/c0011267.html

2.2 Software Versions used in setup

For both topologies covered in Sections 4 and 5, the same software configuration was used:

- DB2 9.7
- Linux Enterprise Server (Linux 2.6.16.46-0.12-smp # x86_64 x86_64 x86_64 GNU/Linux)

3. Overview of Important Concepts

3.1 The db2haicu utility

Db2haicu is a tool available with DB2 9.7 and stands for the "DB2 High Availability Instance Configuration Utility". This utility takes in user input regarding the software and hardware environment of a DB2 instance, and configures the instance for High Availability using the Tivoli System Automation (TSA) Cluster Manager. During this configuration process, all necessary resources, dependencies and equivalencies are automatically defined to TSA. **Note***:* TSA does not need to be manually installed on your system as it is pre-packaged with DB2 9.7.

Note that this tool first became available with DB2 9.5 and the examples displayed here would also apply without change to DB2 9.5.

Two input methods can be used to provide the necessary data to db2haicu. The first method is the interactive mode, where the user is prompted for input at the command line. The second input method is the XML mode, where db2haicu can parse the necessary data from a user defined XML file.

The db2haicu interactive mode is covered in Section 4 and the db2haicu XML mode is covered in Section 5.

3.2 HADR Overview

The HADR feature of DB2 9.7 allows a database administrator (DBA) to have one "hot standby" copy of any DB2 database such that, in the event of a primary database failure, a DBA can quickly switch over to the "hot standby" with minimal interruption to database clients. (See Fig.1 below for a typical HADR environment.)

However, an HADR primary database does not automatically switch over to its standby database in the event of failure. Instead, a DBA must manually perform a takeover operation when the primary database has failed.

The db2haicu tool can be used to automate such an HADR system. During the db2haicu configuration process, the necessary HADR resources and their relationships are defined to the cluster manager. Failure events in the HADR system can then be detected automatically and takeover operations can be executed without manual intervention.

3.3 Typical HADR topologies

A typical HADR topology contains two nodes – a primary node to host the primary HADR database, and a standby node to host the standby HADR database. The nodes are connected to each other over a network to accommodate transaction replication between the two databases.

Two automated HADR topologies are implemented in this paper: a *single* network HADR topology, automated using the db2haicu XML mode, and a *multiple* network HADR topology, automated using the db2haicu interactive mode.

4. Setting up an automated multiple network HADR topology using the db2haicu Interactive mode

The configuration of an automated multiple network HADR topology, as illustrated in Fig. 1, is described in the steps below.

Notes:

1. There are two parts to this configuration. The first part describes the preliminary steps needed to configure a multiple network HADR topology. The second part describes the use of db2haicu's interactive mode to automate the topology for failovers.

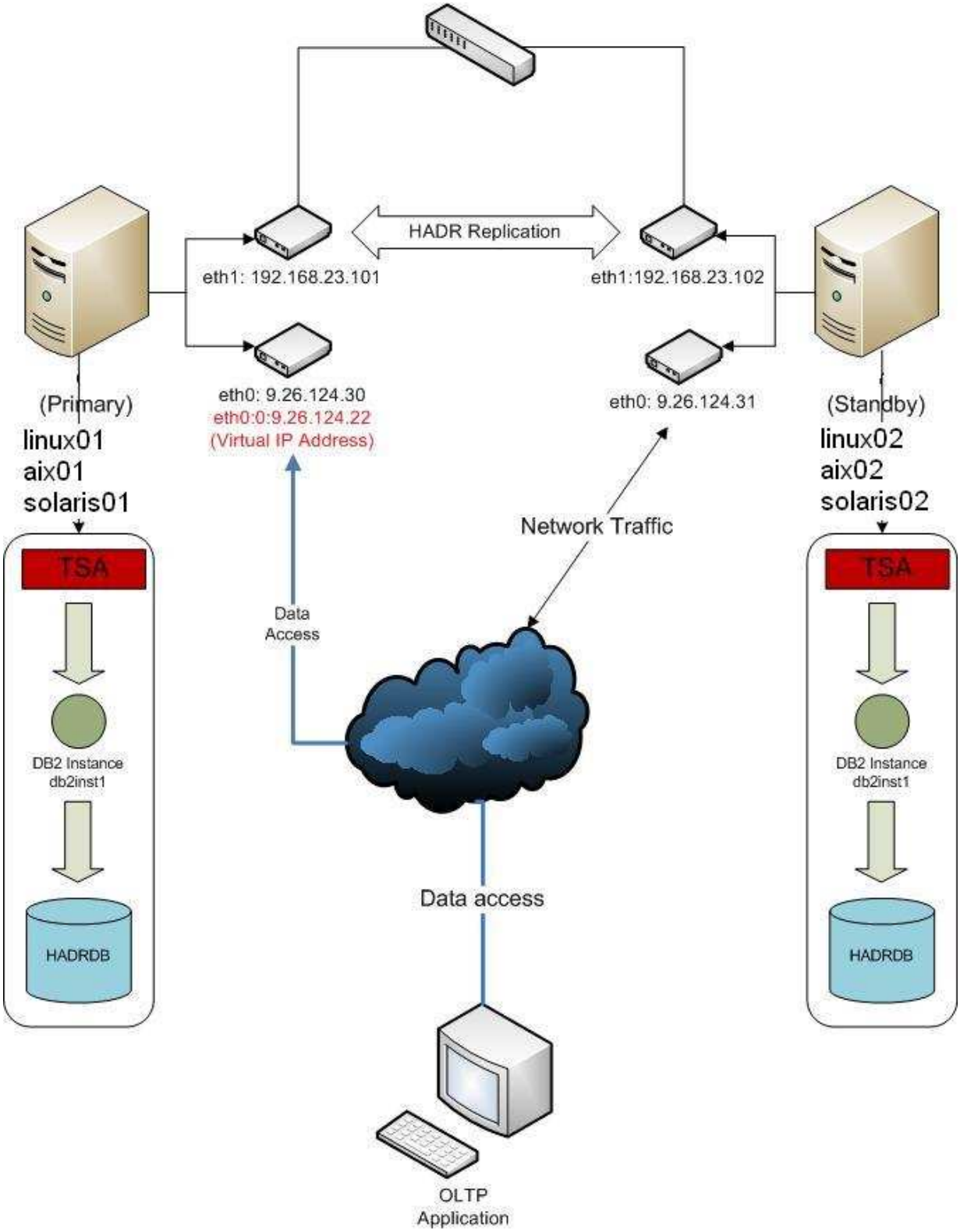
2. The parameters used for various commands described below are based on the topology illustrated in Fig. 1. You must change the parameters to match your own specific environment.

4.1 Topology Configuration:

This topology makes use of two nodes: the primary node to host the primary HADR database and the standby node to host the standby HADR database.

The nodes are connected to each other using two distinct networks: a *public* network and a *private* network. The public network is defined to host the virtual IP address that allows clients to connect to the primary database. The private network is defined to carry out HADR data replication between the primary and standby nodes.

Fig. 1. Automated Multiple Network HADR Topology



4.1.1 Basic Network setup:

The two machines used for this topology contain two network interfaces each. For Linux environments, we will take the network adapters to be named eth0 and eth1 (same naming on each machine). We will take it that eth0 is the 'public' adapter (connected to the public network) and eth1 is the 'private' adapter (commonly only connected to the other node in the cluster). Note that for AIX, the adapters are generally named en0 (and en1), and for Solaris, adapters are generally named hme0 (and hme1). Keep that in mind as you read the following example and you are working with AIX or Solaris environments.

1. The eth0 network interfaces are connected to each other through the external network cloud forming the *public* network. We assigned the following static IP addresses to the eth0 adapters on the primary and standby nodes:

Primary node (linux01)
eth0: 9.26.124.30 (255.255.255.0)

Standby node (linux02)
eth0: 9.26.124.31 (255.255.255.0)

2. The eth1 network interfaces are connected directly to each other using a switch forming a *private* network. The following static IP addresses are assigned to the eth1 network interfaces:

Primary node (linux01):
eth1: 192.168.23.101 (255.255.255.0)

Standby node (linux02):
eth1: 192.168.23.102 (255.255.255.0)

3. Make sure that the primary and standby node names are mapped to their corresponding public IP addresses in the /etc/hosts file:

9.26.124.30 linux01 linux01.fullyQualifiedDomain.com
9.26.124.31 linux02 linux02.fullyQualifiedDomain.com

Confirm that the above two lines are present at each /etc/hosts file on each of the machines linux01 and linux02.

4. Make sure that the file ~/sqllib/db2nodes.cfg for the instance residing on the machine linux01 has contents as follows:

0 linux01

Then ensure that the file ~/sqllib/db2nodes.cfg for the instance residing on the machine linux02 has contents as follows:

0 linux02

5. Ensure that the values linux01 and linux02 are used to refer to these two nodes when prompted by the db2haicu setup tool.

6. Ensure that the 'hostname' command when executed on the linux01 machine returns 'linux01'. Similarly, ensure that the 'hostname' command when executed on the linux02 machine returns 'linux02'.

7. The primary and the standby machines should be able to ping each other over both networks. Issue the following commands on both the primary and the standby machines and make sure that they complete successfully:

```
% ping linux01
% ping linux02
% ping 192.168.23.101
% ping 192.168.23.102
```

4.1.2 HADR Database setup

We create a primary DB2 instance named 'db2inst1' on the primary node, and a standby instance 'db2inst1' on the standby node[1]. The HADR databases named "HADRDB" are set up between the two instances.

1) The following parameters were used to configure the standby HADR database on the standby instance 'db2inst1'.

```
/home/db2inst1% db2 get db cfg for hadrdb | grep HADR
 HADR database role                                      = STANDBY
 HADR local host name   (HADR_LOCAL_HOST)                = 192.168.23.102
 HADR local service name  (HADR_LOCAL_SVC)               = 55555
 HADR remote host name   (HADR_REMOTE_HOST)              = 192.168.23.101
 HADR remote service name     (HADR_REMOTE_SVC)          = 55555
 HADR instance name of remote server  (HADR_REMOTE_INST) = db2inst1
 HADR timeout value    (HADR_TIMEOUT)                    = 120
 HADR log write synchronization mode     (HADR_SYNCMODE) = SYNC
 HADR peer window duration (seconds)  (HADR_PEER_WINDOW) = 300
```

2) The following parameters were used to configure the primary HADR database on the primary instance 'db2inst1'.

```
/home/db2inst1% db2 get db cfg for hadrdb | grep HADR
 HADR database role                                      = PRIMARY
 HADR local host name   (HADR_LOCAL_HOST)                = 192.168.23.101
 HADR local service name  (HADR_LOCAL_SVC)               = 55555
 HADR remote host name   (HADR_REMOTE_HOST)              = 192.168.23.102
 HADR remote service name     (HADR_REMOTE_SVC)          = 55555
 HADR instance name of remote server  (HADR_REMOTE_INST) = db2inst1
 HADR timeout value    (HADR_TIMEOUT)                    = 120
 HADR log write synchronization mode     (HADR_SYNCMODE) = SYNC
 HADR peer window duration (seconds)  (HADR_PEER_WINDOW) = 300
```

3) Note that the configuration values HADR_REMOTE_HOST and HADR_LOCAL_HOST on the standby and primary databases reflect the IP addresses assigned to the eth1 NICs.

4) Set the HADR_PEER_WINDOW configuration parameter to a large enough value to ensure that the HADR peer window does not expire before the standby machine attempts to take over the hard primary role. In our environment, 300 seconds was sufficient to ensure this.

5) Ensure that the clocks of the two nodes are synchronized.

6) The HADR_SYNCMODE was set to 'SYNC' and will be used for both examples in this paper. Note that 'SYNC' mode provides a somewhat higher level of data replication protection than 'NEARSYNC' mode. For further details concerning the HADR synchronization modes, consult the DB2 documentation.

We must make sure that the HADR databases are in "peer" state before proceeding with the db2haicu tool. Issue the following command to check HADR status on either the primary or the standby instance: [2]

db2pd –hadr –db *database name*

You should see output similar to this:

```
/home/db2inst1% db2pd -hadr -db hadrdb

Database Partition 0 -- Database HADRDB -- Active -- Up 0 days 00:14:23

HADR Information:
Role     State              SyncMode HeartBeatsMissed   LogGapRunAvg (bytes)
Primary Peer                Sync      0                 0

…….
continues
```

[1]For more information on how to create DB2 instances, consult the DB2 documentation.

[2]This paper will not go into details of how to configure HADR replication. For more information, consult the DB2 documentation.

4.1.3 Cluster preparation

Before using the db2haicu tool, the primary and the standby nodes must be prepared with the proper security environment.

1) With root authority, issue the following command on both the primary and the standby nodes:

```
root@linux01# preprpnode linux01 linux02
.....
root@linux02# preprpnode linux01 linux02
```

This command only needs to be run *once* per node and *not* for every DB2 instance that is made highly available.

4.1.4 Network Time Protocol:

The time and dates on the standby and the primary machines must be synchronized as closely as possible. This is absolutely critical to ensure smooth failovers during primary machine failures.

The Network time protocol can be used for this purpose. Refer to your operating system documentation for information on how to configure NTP for your system. Configure NTP for both the primary and standby database hosting machines.

4.1.5 Client reroute:

The client reroute feature allows a DB2 client application to recover from a lost database connection in case of a network failure. In both the HADR configurations discussed in this paper, we use a virtual IP address that allows clients to connect to the primary database. Identify the IP address that will be created and failed over as part of the HADR failover. Also, identify the port number for the instance TCP/IP listener via checking the value of the SVCENAME DBM CFG parameter. Make sure that the TCP/IP listener is started (for both instances with db2set DB2COMM=tcpip) and that the SVCENAME parameter is set to a consistent unused value (we will use 55445 for the TCP/IP listener ports on both instances).

1) Issue the following command on both the standby and the primary nodes to configure the virtual IP address for client reroute:

```
/home/db2inst1/% db2 update alternate server for database
hadrdb using hostname 9.26.124.22 port 55445
….
/home/db2inst1/% db2 update alternate server for database
hadrdb using hostname 9.26.124.22 port 55445
```

In this example, 9.26.124.22 is the virtual IP address (which we will define subsequently with db2haicu) and port 55445 is found in the SVCENAME DBM CFG parameter.

4.2 The db2haicu Interactive setup mode

After the preceding preliminary configuration steps are completed, the db2haicu tool can be used to automate HADR failover.

Db2haicu must be run first on the standby instance and then on the primary instance for the configuration to complete. The details involving the process are outlined in the following section.

Note: The "…" above a db2haicu message indicates continuation from a message displayed in a previous step.

Creating a Cluster Domain

Log on to the standby instance and issue the following command:

db2haicu

The following welcome message will be displayed on the screen:

```
/home/db2inst1% db2haicu
Welcome to the DB2 High Availability Instance Configuration Utility (db2haicu).

You can find detailed diagnostic information in the DB2 server diagnostic log file called
db2diag.log. Also, you can use the utility called db2pd to query the status of the cluster
domains you create.

For more information about configuring your clustered environment using db2haicu, see
the topic called 'DB2 High Availability Instance Configuration Utility (db2haicu)' in the
DB2 Information Center.

db2haicu determined the current DB2 database manager instance is db2inst1. The
cluster configuration that follows will apply to this instance.

db2haicu is collecting information on your current setup. This step may take some time
as db2haicu will need to activate all databases for the instance to discover all paths ...
When you use db2haicu to configure your clustered environment, you create cluster
domains. For more information, see the topic 'Creating a cluster domain with db2haicu' in
the DB2 Information Center. db2haicu is searching the current machine for an existing
active cluster domain ...
db2haicu did not find a cluster domain on this machine. db2haicu will now query the
system for information about cluster nodes to create a new cluster domain ...

db2haicu did not find a cluster domain on this machine. To continue configuring your
clustered environment for high availability, you must create a cluster domain; otherwise,
db2haicu will exit.

Create a domain and continue? [1]
1. Yes
2. No
```

We must now create a cluster domain.

1) Type '1' and press "Enter" at the following initial prompt.

```
…
Create a domain and continue? [1]
1. Yes
2. No
1
```

2) Enter a unique name of the domain you want to create and the number of nodes contained in the domain (2 in our case). We decided to name our domain "hadr_linux_domain".

```
…
Create a unique name for the new domain:
hadr_linux_domain
How many cluster nodes will the domain hadr_linux_domain contain?
2
```

3) Follow the prompts to enter the name of the primary and the standby nodes and confirm domain creation.

```
            …
Enter the host name of a machine to add to the domain:
linux01
Enter the host name of a machine to add to the domain:
linux02
db2haicu can now create a new Creating domain hadr_linux_domain in the
cluster ...

db2haicu can now create a new domain containing the 2 machines that you
specified. If you choose not to create a domain now, db2haicu will exit.

Create the domain now? [1]
1. Yes
2. No
1
Creating domain hadr_linux_domain in the cluster...
Creating domain hadr_linux_domain in the cluster was successful.
```

Quorum configuration

After the domain creation has completed, a quorum must be configured for the cluster domain. The supported quorum type for this solution is a "network quorum". A network quorum is a pingable IP address that is used to decide which node in the cluster will serve as the "active" node during a site failure, and which nodes will be offline.

You will be prompted by db2haicu to enter quorum configuration values:

```
…
You can now configure a quorum device for the domain. For more
information, see the topic "Quorum devices" in the DB2 Information
Center. If you do not configure a quorum device for the domain, then a
human operator will have to manually intervene if subsets of machines in
the cluster lose connectivity.
Configure a quorum device for the domain called hadr_linux_domain? [1]
1. Yes
2. No
```

From the preceding prompt:

1) Type '1' and press "Enter" to create the quorum

```
…
1
The following is a list of supported quorum device types:
  1. Network Quorum
Enter the number corresponding to the quorum device type to be used: [1]
```

2) Type '1' and press Enter again to choose the Network Quorum type. Then, follow the prompt to enter the IP address you would like to use as a network tiebreaker.

```
…
Specify the network address of the quorum device:
9.26.4.5
Configuring quorum device for domain hadr_linux_domain...
Configuring quorum device for domain hadr_linux_domain was successful.
```

Quorum configuration is now completed. Note that you may use any IP address as the quorum device, so long as the IP address is pingable from both nodes. Use an IP address that is known to be reliably available on the network. The IP address of the DNS server is usually a reasonable choice here.

Network setup

After the quorum configuration, you must define the public and the private networks of your system to db2haicu. If network failure detection is important to your configuration, you *must* follow the prompts and add the networks to the cluster at this point. All network interfaces are automatically discovered by the db2haicu tool.

An example is shown below:

```
Create networks for these network interface cards? [1]
1. Yes
2. No
1
Enter the name of the network for the network interface card: eth0 on
cluster node: linux01
1. Create a new public network for this network interface card.
2. Create a new private network for this network interface card.
Enter selection:
1
Are you sure you want to add the network interface card eth0 on cluster
node linux01 to the network db2_public_network_0? [1]
1. Yes
2. No
1
Adding network interface card eth0 on cluster node linux01 to the network
db2_public_network_0 ...
Adding network interface card eth0 on cluster node linux01 to the network
db2_public_network_0 was successful.
Enter the name of the network for the network interface card: eth0 on
cluster node: linux02
1. db2_public_network_0
2. Create a new public network for this network interface card.
3. Create a new private network for this network interface card.
Enter selection:
1
Are you sure you want to add the network interface card eth0 on cluster
node linux02 to the network db2_public_network_0? [1]
1. Yes
2. No
1
Adding network interface card eth0 on cluster node linux02 to the network
db2_public_network_0 ...
Adding network interface card eth0 on cluster node linux02 to the network
db2_public_network_0 was successful.
Enter the name of the network for the network interface card: eth1 on
cluster node: linux02
1. db2_public_network_0
2. Create a new public network for this network interface card.
3. Create a new private network for this network interface card.
Enter selection:
3
Are you sure you want to add the network interface card eth1 on cluster
node linux02 to the network db2_private_network_0? [1]
1. Yes
2. No
```

```
1
Adding network interface card eth1 on cluster node linux02 to the network
db2_private_network_0 ...
Adding network interface card eth1 on cluster node linux02 to the network
db2_private_network_0 was successful.
Enter the name of the network for the network interface card: eth1 on
cluster node: linux01
1. db2_private_network_0
2. db2_public_network_0
3. Create a new public network for this network interface card.
4. Create a new private network for this network interface card.
Enter selection:
1
Are you sure you want to add the network interface card eth1 on cluster
node linux01 to the network db2_private_network_0? [1]
1. Yes
2. No
1
Adding network interface card eth1 on cluster node linux01 to the network
db2_private_network_0 ...
Adding network interface card eth1 on cluster node linux01 to the network
db2_private_network_0 was successful.
```

Note that it is not possible to add two NICs with different subnet masks and different assigned IP addresses to the same common network. For example, in this configuration, if one tries to define eth1 and eth0 to the same network using db2haicu, the input will be rejected.

Cluster Manager selection

After the network definitions, db2haicu prompts for the Cluster Manager software being used for the current HA setup.

For our purpose, we select TSA:

```
The cluster manager name configuration parameter (high availability
configuration parameter) is not set. For more information, see the topic
"cluster_mgr - Cluster manager name configuration parameter" in the DB2
Information Center. Do you want to set the high availability
configuration parameter?
The following are valid settings for the high availability configuration
parameter:
  1.TSA
  2.Vendor
Enter a value for the high availability configuration parameter: [1]
1
Setting a high availability configuration parameter for instance db2inst1
to TSA.
Adding DB2 database partition 0 to the cluster ...
Adding DB2 database partition 0 to the cluster was successful.
```

The db2haicu utility will automatically add the DB2 single partition instance running the standby HADR database to the specified Cluster Manager at this point.

Automating HADR failover

Right after the DB2 standby single partition instance resource has been added to the cluster domain, the user will be prompted to confirm automation for the HADR database in question:

```
…
Do you want to validate and automate HADR failover for the HADR database
HADRDB? [1]
1. Yes
2. No
```

From the preceding prompt:

1) Type '1' and press "Enter"

The db2haicu tool must be run on both the standby and the primary instances for the HADR setup to complete. After the preceding steps have been completed, a message indicating this condition is displayed on the screen, and we must proceed to the primary machine to complete the setup:

```
   ….
Adding HADR database HADRDB to the domain ...

The HADR database HADRDB has been determined to be valid for high
availability. However, the database cannot be added to the cluster from
this node because db2haicu detected this node is the standby for the HADR
database HADRDB. Run db2haicu on the primary for the HADR database HADRDB
to configure the database for automated failover.

All cluster configurations have been completed successfully. db2haicu
exiting ...
```

Primary Instance setup

1) After the standby instance has been configured, log on to the primary DB2 instance and issue the db2haicu command again. The welcome message will be displayed, and the first step would be to select a Cluster Manager for the setup. We again select TSA.

2) Db2haicu will then proceed to adding the DB2 single partition resource instance running the primary database to the cluster, and prompt the user for confirmation of automating HADR failover.

The steps are shown below:

```
…
The cluster manager name configuration parameter (high availability
configuration parameter) is not set. For more information, see the topic
"cluster_mgr - Cluster manager name configuration parameter" in the DB2
Information Center. Do you want to set the high availability
configuration parameter?
The following are valid settings for the high availability configuration
parameter:
  1.TSA
  2.Vendor
Enter a value for the high availability configuration parameter: [1]
1
Setting a high availability configuration parameter for instance db2inst1
to TSA.
Adding DB2 database partition 0 to the cluster ...
Adding DB2 database partition 0 to the cluster was successful.
Do you want to validate and automate HADR failover for the HADR database
HADRDB? [1]
1. Yes
2. No
1
Adding HADR database HADRDB to the domain ...
The cluster node 192.168.23.102 was not found in the domain.
Please re-enter the host name.
linux02
The cluster node 192.168.23.101 was not found in the domain.
Please re-enter the host name.
linux01
Adding HADR database HADRDB to the domain ...
Adding HADR database HADRDB to the domain was successful.
…
```

Virtual IP Address setup

Once the HADR database resource has been added to the cluster from the primary instance (as shown in the preceding section), db2haicu will prompt the user to create a virtual IP address:

```
…
Do you want to configure a virtual IP address for the HADR database
HADRDB? [1]
1. Yes
2. No
```

1) At the preceding prompt, type ‘1’ and press Enter to continue. Enter the value of the virtual IP address that you configured with the client reroute setup (Section 4.1.5).

```
…
1
Enter the virtual IP address:
9.26.124.22
…
```

2) Enter the subnet mask associated with the virtual IP address, followed by the network on which this IP address will reside. We assign this IP address to the public network to allow external clients to connect.

```
…
Enter the subnet mask for the virtual IP address 9.26.124.22:
[255.255.255.0]
255.255.245.0

Select the network for the virtual IP 9.26.124.22:
1. db2_private_network_0
2. db2_public_network_0
Enter selection:
2

Adding virtual IP address 9.26.124.22 to the domain ...
Adding virtual IP address 9.26.124.22 to the domain was successful.
```

You must make sure that your IP address and subnet mask values are well formed and correspond with the subnet mask of the network you chose. All invalid inputs will be rejected.

After the virtual IP address configuration, the Automated Cluster controlled HADR configuration is complete.
Issue the "lssam" command as root to see the resources created during this process.

```
Online IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Online
      '- Online IBM.Application:db2_db2inst1_linux01_0-rs
            '- Online IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Online IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
      '- Online IBM.Application:db2_db2inst1_linux02_0-rs
            '- Online IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Nominal=Online
      |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs
            |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
            '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
      '- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
            |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
            '- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
Online IBM.Equivalency:db2_db2inst1_linux01_0-rg_group-equ
      '- Online IBM.PeerNode:linux01:linux01
Online IBM.Equivalency:db2_db2inst1_linux02_0-rg_group-equ
      '- Online IBM.PeerNode:linux02:linux02
Online IBM.Equivalency:db2_db2inst1_db2inst1_HADRDB-rg_group-equ
      |- Online IBM.PeerNode:linux01:linux01
      '- Online IBM.PeerNode:linux02:linux02
Online IBM.Equivalency:db2_private_network_0
      |- Online IBM.NetworkInterface:eth1:linux02
      '- Online IBM.NetworkInterface:eth1:linux01
Online IBM.Equivalency:db2_public_network_0
      |- Online IBM.NetworkInterface:eth0:linux01
      '- Online IBM.NetworkInterface:eth0:linux02
```

The command "db2pd –ha" can also be issued from the instance owner ID to examine the state of the resources:

```
/home/db2inst1% db2pd -ha
         DB2 HA Status
Instance Information:
Instance Name               = db2inst1
Number Of Domains           = 1
Number Of RGs for instance     = 2

Domain Information:
Domain Name                 = hadr_linux_domain
Cluster Version             = 2.5.3.0
Cluster State               = Online
Number of nodes             = 2

Node Information:
Node Name                   State
--------------------        ------------------
linux02                     Online
linux01                     Online

Resource Group Information:
Resource Group Name             = db2_db2inst1_db2inst1_HADRDB-rg
Resource Group LockState        = Unlocked
Resource Group OpState          = Online
Resource Group Nominal OpState = Online
Number of Group Resources      = 2
Number of Allowed Nodes         = 2
  Allowed Nodes
  -------------
  linux01
  linux02
Member Resource Information:
  Resource Name                 = db2_db2inst1_db2inst1_HADRDB-rs
  Resource State                = Online
  Resource Type                 = HADR
  HADR Primary Instance         = db2inst1
  HADR Secondary Instance       = db2inst1
  HADR DB Name                  = HADRDB
  HADR Primary Node             = linux01
  HADR Secondary Node           = linux02

  Resource Name                 = db2ip_9_26_124_22-rs
  Resource State                = Online
  Resource Type                 = IP

continues …
```

5. Setting up an automated single network HADR topology using the db2haicu XML mode

The configuration of an automated single network HADR topology, as illustrated in Fig. 2, is described in the steps below. Similar to the previous section:

1. There are two parts to this configuration. The first part describes the steps needed to configure a single network HADR topology. The second part describes the use of db2haicu's XML mode to automate the topology for failovers.

2. The parameters used for the configuration described below are based on the topology illustrated in Fig. 2. You must change the parameters to match your own specific environment.

5.1 Topology setup

This topology contains a total of two nodes with one network interface per node. One of these nodes is the primary node (e.g., linux01, aix01, solaris01) that hosts the primary HADR database, and the second node is the 'hot' standby node (e.g., linux02, aix02, solaris02) that hosts the standby HADR database.

HADR data replication along with any other network traffic uses the single network formed between the two network interfaces of the primary and standby nodes. We define this single network as the *public* network.

5.1.1 Basic Network setup:

The two machines used in this configuration contain one network interface each (e.g., eth0).

1. The eth0 interfaces are connected to each other through the external network cloud forming a *public* network. The following static IP addresses were assigned:

Primary node (linux01)
eth0: 9.26.124.30 (255.255.255.0)

Standby node (linux02)
eth0: 9.26.124.31 (255.255.255.0)

2. Make sure that the primary and standby node names are mapped to their corresponding public IP addresses in the /etc/hosts file:

9.26.124.30 linux01 linux01.fullyQualifiedDomain.com
9.26.124.31 linux02 linux02.fullyQualifiedDomain.com

Ensure that the above two lines are present at each /etc/hosts file on each of the two machines.

3. Make sure that the file ~/sqllib/db2nodes.cfg for the instance residing on the machine linux01 has contents as follows:

0 linux01

and the file ~/sqllib/db2nodes.cfg for the instance residing on the machine linux02 has contents as follows:

0 linux02

4. Ensure that the values linux01 and linux02 are used to refer to these two nodes when prompted by the db2haicu setup tool.

5. Ensure that the command 'hostname' when run on the linux01 machine returns the value 'linux01' and the command 'hostname' when run on the linux02 machine returns 'linux02'.

6. Ensure that in the database configuration values HADR_LOCAL_HOST and HADR_REMOTE_HOST the values linux01 and linux02 are used as appropriate, at both the primary and standby HADR databases.

7. The primary and the standby machines should be able to ping each other over both networks. Issue the following commands on both the primary and the standby machines and make sure that they complete successfully:

```
% ping linux01
% ping linux02
```

5.1.2 HADR setup:

We create a primary DB2 instance named 'db2inst1' on the primary node, and a standby instance 'db2inst1' on the standby node. The HADR databases named "HADRDB" are set up between the two instances.

1) The following parameters were used to configure the standby HADR database on the primary instance 'db2inst1'.

```
/home/db2inst1% db2 get db cfg for hadrdb | grep HADR
 HADR database role                                      = PRIMARY
 HADR local host name   (HADR_LOCAL_HOST)                = linux01
 HADR local service name  (HADR_LOCAL_SVC)               = 55555
 HADR remote host name   (HADR_REMOTE_HOST)              = linux02
 HADR remote service name     (HADR_REMOTE_SVC)          = 55555
 HADR instance name of remote server  (HADR_REMOTE_INST) = db2inst1
 HADR timeout value    (HADR_TIMEOUT)                    = 120
 HADR log write synchronization mode     (HADR_SYNCMODE) = SYNC
 HADR peer window duration (seconds)  (HADR_PEER_WINDOW) = 300
```

2) The following parameters were used to configure the standby HADR database on the standby instance 'db2inst1'.

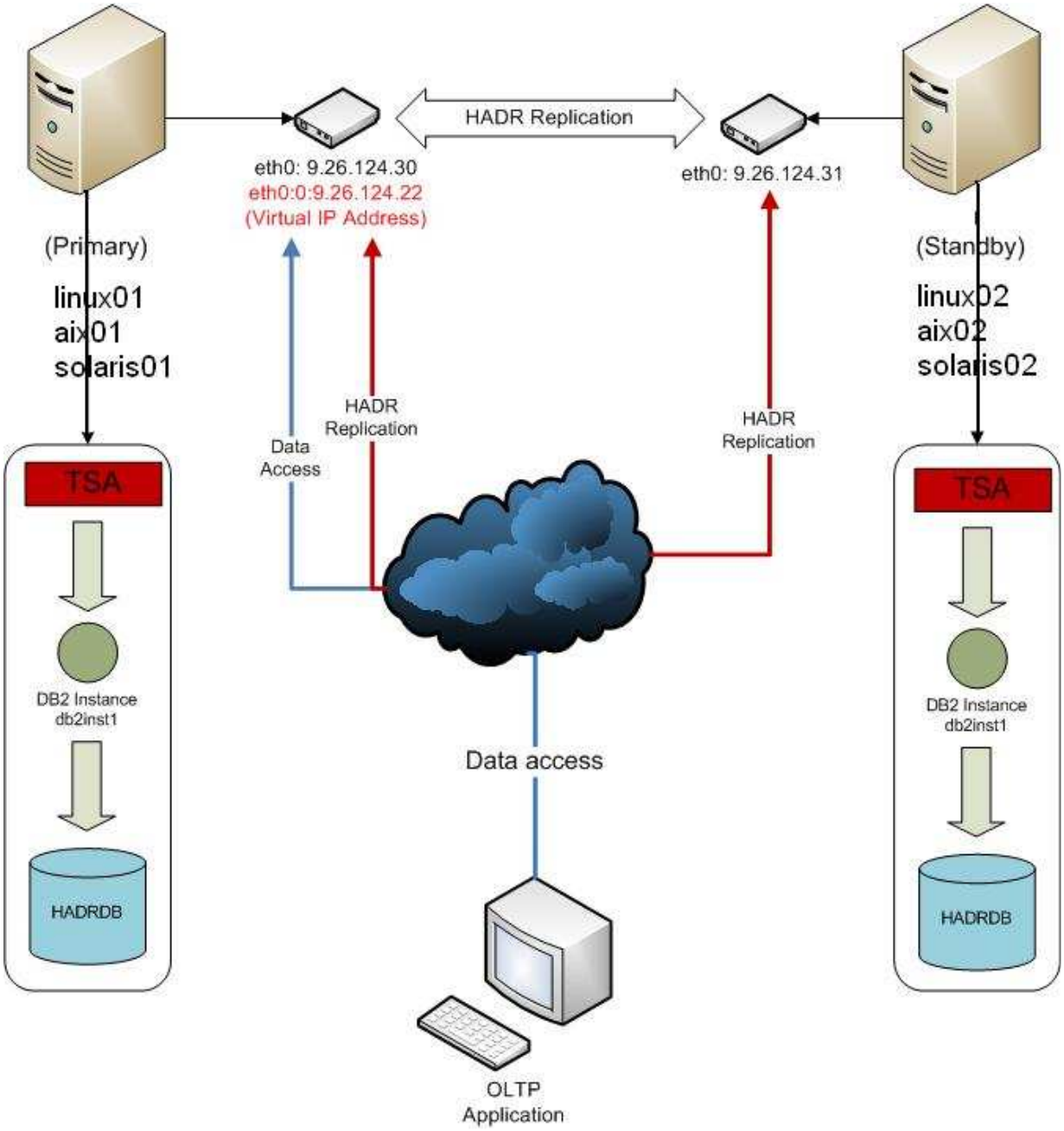
```
/home/db2inst1% db2 get db cfg for hadrdb | grep HADR
 HADR database role                                      = STANDBY
 HADR local host name   (HADR_LOCAL_HOST)                = linux02
 HADR local service name  (HADR_LOCAL_SVC)               = 55555
 HADR remote host name   (HADR_REMOTE_HOST)              = linux01
 HADR remote service name     (HADR_REMOTE_SVC)          = 55555
 HADR instance name of remote server  (HADR_REMOTE_INST) = db2inst1
 HADR timeout value    (HADR_TIMEOUT)                    = 120
 HADR log write synchronization mode     (HADR_SYNCMODE) = SYNC
 HADR peer window duration (seconds)  (HADR_PEER_WINDOW) = 300
```

The configuration values HADR_REMOTE_HOST and HADR_LOCAL_HOST directly refer to the primary and standby machines in question. We must also check that our databases are in "peer" state by issuing the following command:

db2pd –hadr –db database name

Note that steps 4.1.3 – 4.1.5 must also be followed for this configuration.

Fig. 2. Automated Single Network HADR topology



5.2 Utilizing the db2haicu XML File

A sample db2haicu XML file is shown in the output below. It contains all the information that db2haicu needs to know in order to make a DB2 HADR instance highly available. Other sample files can be found in ~/sqllib/samples/ha/xml/.

```
<DB2Cluster xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="db2ha.xsd" clusterManagerName="TSA" version="1.0">
  <ClusterDomain domainName="hadr_linux_domain">
       <Quorum quorumDeviceProtocol="network" quorumDeviceName="9.26.4.5"/>

       <PhysicalNetwork physicalNetworkName="db2_public_network_0"
         physicalNetworkProtocol="ip">
           <Interface interfaceName="eth0" clusterNodeName="linux01">
                <IPAddress baseAddress="9.26.124.30" subnetMask="255.255.255.0"
                 networkName="db2_public_network_0"/>
           </Interface>
           <Interface interfaceName="eth0" clusterNodeName="linux02">
                <IPAddress baseAddress="9.26.124.31" subnetMask="255.255.255.0"
                 networkName="db2_public_network_0"/>
           </Interface>
       </PhysicalNetwork>

      <ClusterNode clusterNodeName="linux01"/>
      <ClusterNode clusterNodeName="linux02"/>
  </ClusterDomain>

  <FailoverPolicy>
      <HADRFailover></HADRFailover>
  </FailoverPolicy>

  <DB2PartitionSet>
      <DB2Partition dbpartitionnum="0" instanceName="db2inst1">
      </DB2Partition>
  </DB2PartitionSet>

  <HADRDBSet>
      <HADRDB databaseName="HADRDB" localInstance="db2inst1"
       remoteInstance="db2inst1" localHost="linux01" remoteHost="linux02" />
      <VirtualIPAddress baseAddress="9.26.124.22" subnetMask="255.255.245.0"
       networkName="db2_public_network_0"/>
  </HADRDBSet>

</DB2Cluster>
```

The existing values in the preceding file can be replaced to reflect your own configuration and environment. Below is a brief description of what the different elements shown in the preceding XML file represent:

- The <*ClusterDomain*> element covers all cluster-wide information. This includes: *quorum information, cluster node information and cluster domain name.*

- The <*PhysicalNetwork*> sub-element of the ClusterDomain element includes all network information. This includes the name of the network and the network interface cards contained in it. We define our single public network using this element.

- The <*FailoverPolicy*> element specifies the failover policy that the cluster manager should use with the cluster domain.

- The <*DB2PartitionSet*> element covers the DB2 instance information. This includes the current DB2 instance *name* and the DB2 partition number.

- Finally, the <*HADRDBSet*> covers the HADR database information. This includes the primary node name, standby node name, primary instance name, standby instance name and the virtual IP address associated with the database.

To configure the HADR system with db2haicu XML mode:

1) Log on to the standby instance.

2) Issue the following command:

    db2haicu –f *path to XML file*

At this point, the XML file will be used to configure the standby instance. If an invalid input is encountered during the process, db2haicu will exit with a non-zero error code.

3) Log on to the primary instance.

4) Issue the following command again:

    db2haicu –f *path to XML file*

At this point, the XML file will be used to configure the primary instance. If an invalid input is encountered during the process, db2haicu will exit with a non-zero error code.

The db2haicu output on the primary and the standby instances is shown below.

Sample output from running db2haicu in XML mode on standby instance

```
/home/db2inst1% db2haicu -f db2ha_sample_HADR.xml
Welcome to the DB2 High Availability Instance Configuration Utility (db2haicu).

You can find detailed diagnostic information in the DB2 server diagnostic log file called
db2diag.log. Also, you can use the utility called db2pd to query the status of the cluster domains
you create.

For more information about configuring your clustered environment using db2haicu, see the topic
called 'DB2 High Availability Instance Configuration Utility (db2haicu)' in the DB2 Information
Center.

db2haicu determined the current DB2 database manager instance is db2inst1. The cluster
configuration that follows will apply to this instance.

db2haicu is collecting information on your current setup. This step may take some time as
db2haicu will need to activate all databases for the instance to discover all paths ...
Creating domain hadr_linux_domain in the cluster ...
Creating domain hadr_linux_domain in the cluster was successful.
Configuring quorum device for domain hadr_linux_domain ...
Configuring quorum device for domain hadr_linux_domain was successful.
Adding network interface card eth0 on cluster node linux01 to the network
db2_public_network_0 ...
Adding network interface card eth0 on cluster node linux01 to the network
db2_public_network_0 was successful.
Adding network interface card eth0 on cluster node linux02 to the network
db2_public_network_0 ...
Adding network interface card eth0 on cluster node linux02 to the network
db2_public_network_0 was successful.
Adding DB2 database partition 0 to the cluster ...
Adding DB2 database partition 0 to the cluster was successful.
The HADR database HADRDB has been determined to be valid for high availability. However, the
database cannot be added to the cluster from this node because db2haicu detected this node is
the standby for the HADR database HADRDB. Run db2haicu on the primary for the HADR
database HADRDB to configure the database for automated failover.
All cluster configurations have been completed successfully. db2haicu exiting ...
```

Sample output from running db2haicu in XML mode on the primary instance

```
/home/db2inst1% db2haicu -f db2ha_sample_HADR.xml
Welcome to the DB2 High Availability Instance Configuration Utility (db2haicu).

You can find detailed diagnostic information in the DB2 server diagnostic log file called
db2diag.log. Also, you can use the utility called db2pd to query the status of the cluster
domains you create.

For more information about configuring your clustered environment using db2haicu, see the
topic called 'DB2 High Availability Instance Configuration Utility (db2haicu)' in the DB2
Information Center.

db2haicu determined the current DB2 database manager instance is db2inst1. The cluster
configuration that follows will apply to this instance.

db2haicu is collecting information on your current setup. This step may take some time as
db2haicu will need to activate all databases for the instance to discover all paths ...
Configuring quorum device for domain hadr_linux_domain ...
Configuring quorum device for domain hadr_linux_domain was successful.
The network adapter eth0 on node linux01 is already defined in network db2_public_network_0
and cannot be added to another network until it is removed from its current network.
The network adapter eth0 on node linux02 is already defined in network db2_public_network_0
and cannot be added to another network until it is removed from its current network.
Adding DB2 database partition 0 to the cluster ...
Adding DB2 database partition 0 to the cluster was successful.
Adding HADR database HADRDB to the domain ...
Adding HADR database HADRDB to the domain was successful.
All cluster configurations have been completed successfully. db2haicu exiting ...
```

Note: The messages regarding the networks (highlighted in red) encountered on the primary machine can be safely ignored. These messages appear because we have already defined the public network to db2haicu through the standby node.

The HADR configuration is completed right after db2haicu runs the XML file on the primary instance. Issue the "lssam" command as *root* to see the resources created during this process:

```
Online IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_linux01_0-rs
                '- Online IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Online IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_linux02_0-rs
                '- Online IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Nominal=Online
        |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs
                |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
                '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
        '- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
                |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
                '- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02

continues …
```

The 'db2pd –ha' command can also be issued from the instance owner ID to display the state of the resources.

```
/home/db2inst1% db2pd -ha
        DB2 HA Status
Instance Information:
Instance Name               = db2inst1
Number Of Domains           = 1
Number Of RGs for instance  = 2

Domain Information:
Domain Name                 = hadr_linux_domain
Cluster Version             = 2.5.3.0
Cluster State               = Online
Number of nodes             = 2

Node Information:
Node Name                   State
--------------------        -------------------
linux02                     Online
linux01                     Online

Resource Group Information:
Resource Group Name             = db2_db2inst1_db2inst1_HADRDB-rg
Resource Group LockState        = Unlocked
Resource Group OpState          = Online
Resource Group Nominal OpState = Online
Number of Group Resources       = 2
Number of Allowed Nodes         = 2
  Allowed Nodes
  -------------
  linux01
  linux02
Member Resource Information:
  Resource Name                 = db2_db2inst1_db2inst1_HADRDB-rs
  Resource State                = Online
  Resource Type                 = HADR
  HADR Primary Instance         = db2inst1
  HADR Secondary Instance       = db2inst1
  HADR DB Name                  = HADRDB
  HADR Primary Node             = linux01
  HADR Secondary Node           = linux02

  Resource Name                 = db2ip_9_26_124_22-rs
  Resource State                = Online
  Resource Type                 = IP

continues …
```

6. Post Configuration testing

Once the db2haicu tool is run on both the standby and primary instances, the setup is complete, and we can take our automated HADR environment for a test run. Issue the "lssam" command, and observe the output displayed to the screen. You will see output similar to this:

```
Online IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_linux01_0-rs
                '- Online IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Online IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_linux02_0-rs
                '- Online IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Nominal=Online
        |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
                |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
                '- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
        '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs
                |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
                '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

Below is a brief description of the resources shown in the preceding sample output and what they represent:

1) Primary DB2 instance resource group:
   db2_db2inst1_linux01_0-rg

Member Resources:

a. db2_db2inst1_linux01_0-rs (primary DB2 instance)

2) Standby DB2 instance resource group:
   db2_db2inst1_linux02_0-rg

Member Resources:

a. db2_db2inst1_linux02_0-rs (standby DB2 instance)

3) HADR database resource group:
   db2_db2inst1_db2inst1_HADRDB-rg

Member Resources:

a. db2_db2inst1_db2inst1_HADRDB-rs (HADR DB)
b. db2ip_9_26_124_22-rs (HADR virtual IP address)

The resource groups mentioned above are created for both the HADR configurations discussed in this paper. However, the created networks are different.

In the case of the single network HADR configuration setup, only the following equivalencies are created by db2haicu:

db2_db2inst1_db2inst1_HADRDB-rg_group-equ
db2_db2inst1_linux01_0-rg_group-equ
db2_db2inst1_linux02_0-rg_group-equ
db2_public_network_0

In the case of the multiple network HADR configuration setup, one extra network is created:

db2_db2inst1_db2inst1_HADRDB-rg_group-equ
db2_db2inst1_linux01_0-rg_group-equ
db2_db2inst1_linux02_0-rg_group-equ
db2_public_network_0
db2_private_network_0

This extra network (db2_private_network_0) represents the private network being used to carry out HADR replication.

In the following steps, we will go through simulating various failure scenarios, and how the preceding system configuration will react to such failures. You can assume that the system reaction to a failure scenario is identical for both HADR configurations, if not mentioned otherwise.

Before continuing with this section, you must note some key points:

1. The resources created by db2haicu during the configuration can be in one of the following states:

Online: The resource has been started and is functioning normally.
Offline: The resource has been successfully stopped
Failed Offline: The resource has malfunctioned.

2. The term "peer" state refers to a state between the primary and standby databases, when HADR replication is synchronized and the standby database is capable of taking over the primary role.

3. The term "reintegration" refers to the process during which a former HADR primary database is started as a standby database following a **takeover by force** command or a cluster-initiated HADR resource group move.

The relationships between the networks and the resource groups are also illustrated in Fig. 3 for the multiple network HADR topology and in Fig. 4 for the single network HADR topology.

Fig. 3. Resource Groups created for a multiple Network HADR topology

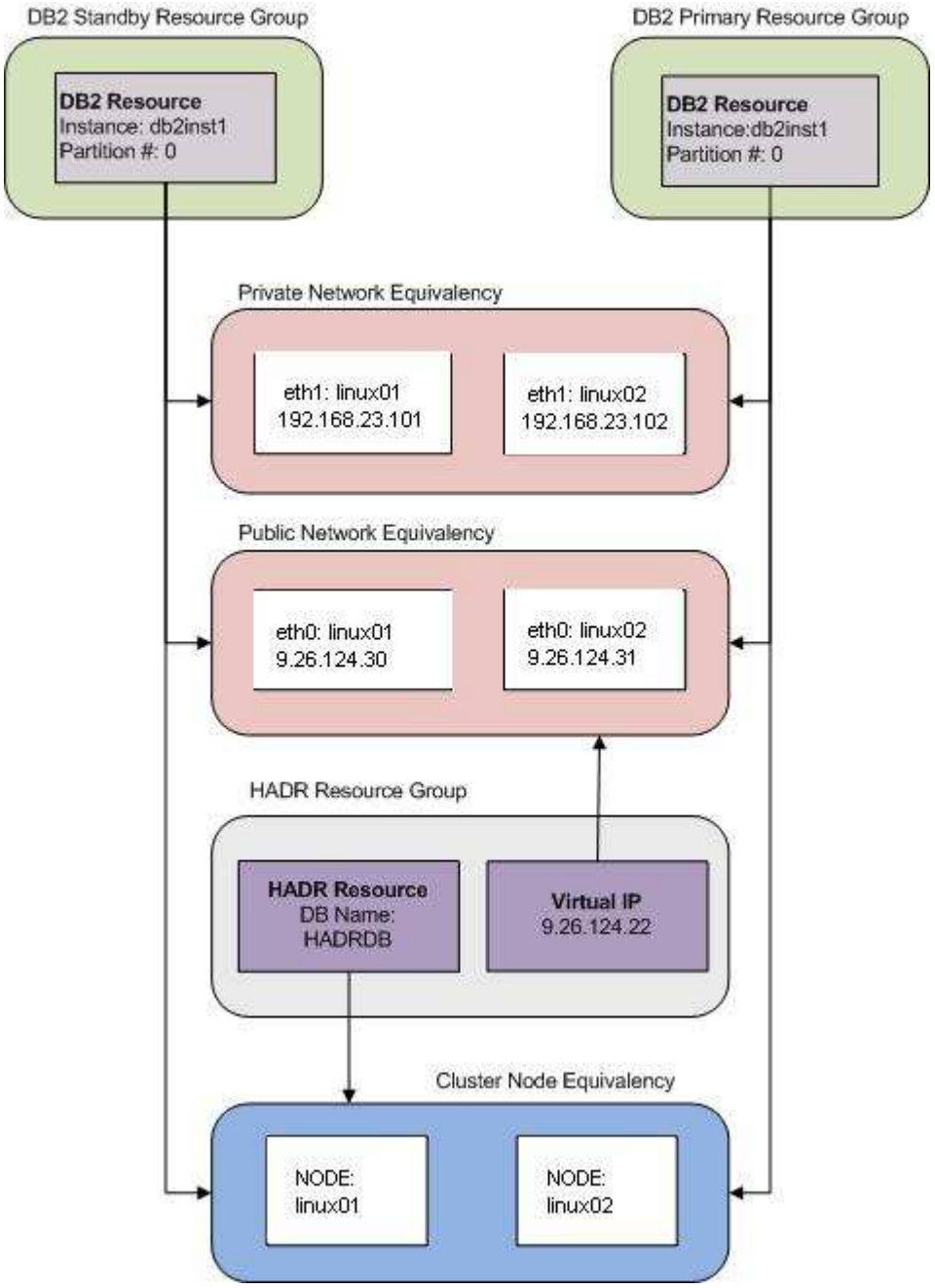
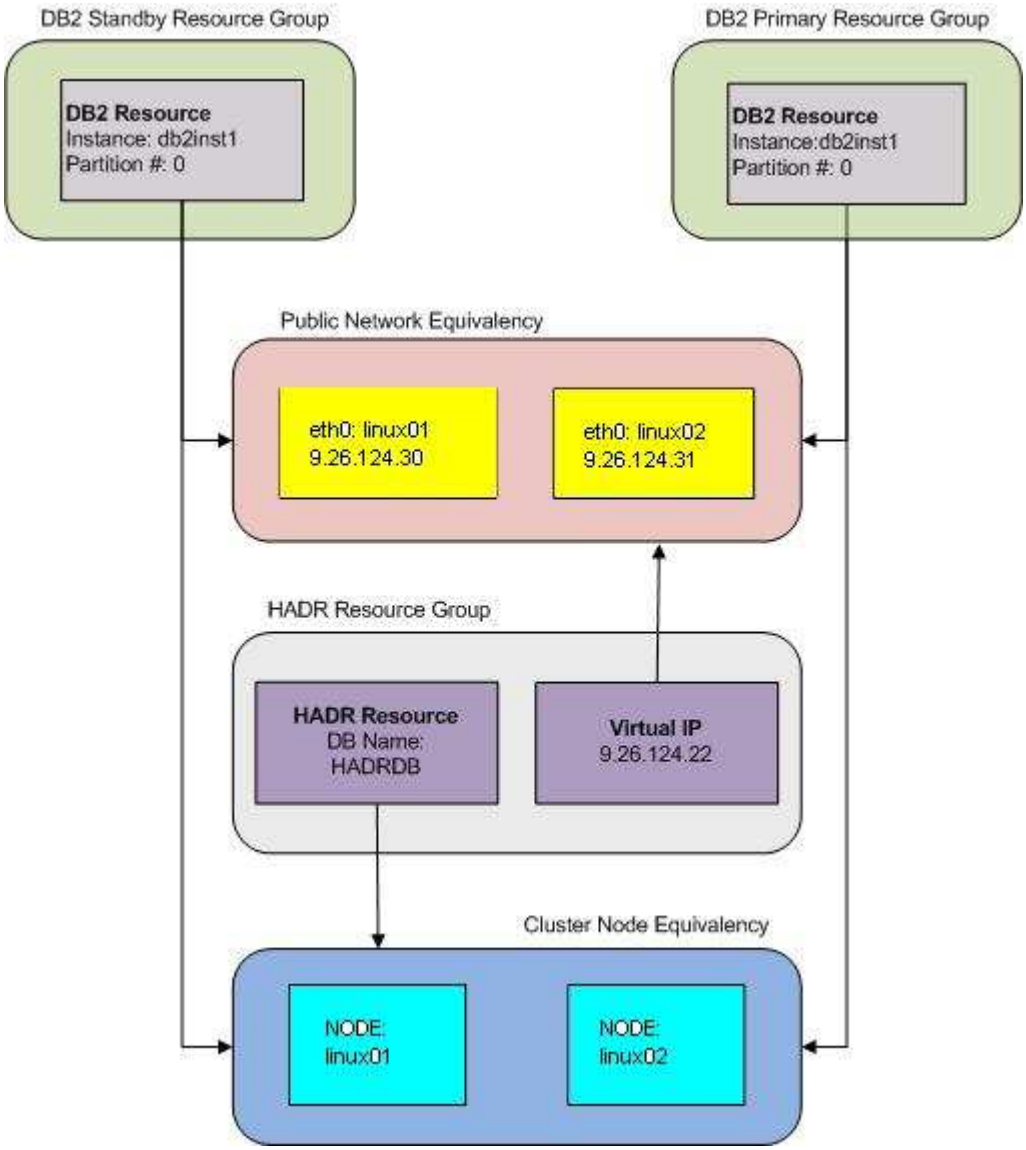


Fig. 4. Resource Groups created for a single Network HADR topology



6.1 The "Power off" test

This test will simulate two failure scenarios: the failure of the standby node, and the failure of the primary node.

A. Standby Node failure:

Follow the instructions below to simulate standby node failure and understand the system reaction.

1) Power off the standby machine (linux02). The easiest way to do this is to simply unplug the power supply.

2) Issue the "lssam" command to observe the state of the resources. You should see output similar to this:

Standby node failure

```
root@linux01:/root# lssam
Online IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_linux01_0-rs
                '- Online IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Failed offline IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
        '- Failed offline IBM.Application:db2_db2inst1_linux02_0-rs
                '- Failed offline IBM.Application:db2_db2inst1_linux02_0-rs:linux02 Node=Offline
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Nominal=Online
        |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
                |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
                '- Failed offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02 Node=Offline
        '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs
                |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
                '- Failed offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02 Node=Offline
```

a. The "Start Inhibited" flag on the standby instance resource group indicates that the standby DB2 instance cannot be started.
b. The "Failed Offline" state of all resources on the standby machine indicates a critical failure.

The primary node will ping and acquire quorum and continue to operate. The clients will connect to the primary database uninterrupted.

3) Plug the standby machine back in.

4) As soon as the machine comes back online, the following series of events will take place:
a. The standby DB2 instance will be started automatically.
b. The standby DB2 HADR database will be activated.
c. HADR data replication will resume and the system will eventually reach "peer" state after all transactions have been replicated.

The preceding steps are carried out to restart HADR replication. After this, the resources will resume the states that they had prior to the failure.

```
Online IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_linux01_0-rs
                '- Online IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Online IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_linux02_0-rs
                '- Online IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Nominal=Online
        |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
                |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
                '- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
        '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs
                |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
                '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

B. Primary Node failure:

Follow the instructions below to simulate a primary node failure:

1) Unplug the power supply for the primary machine (linux01).

2) The clients will not be able to connect to the database. Hence, the Cluster Manager will initiate a failover operation for the HADR resource group.

a. The standby machine (linux02) will ping and acquire quorum.
b. The virtual IP address will be assigned to the eth0 NIC on the standby machine.
c. A takeover operation will allow the standby machine to assume the primary role.

3) Issue the "lssam" or the "db2pd –ha command to examine the state of the resources. After the failover, the resources will settle down to the states shown in the output below:

Primary node failure

```
Failed offline IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Control=StartInhibited Nominal=Online
        '- Failed offline IBM.Application:db2_db2inst1_linux01_0-rs
                '- Failed offline IBM.Application:db2_db2inst1_linux01_0-rs:linux01 Node=Offline
Online IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_linux02_0-rs
                '- Online IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Pending online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Request=Lock Nominal=Online
        |- Pending online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs Control=SuspendedPropagated
                |- Failed offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01 Node=Offline
                '- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
        '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs Control=SuspendedPropagated
                |- Failed offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01 Node=Offline
                '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

a. All resources on the old primary machine (linux01) will assume the "Failed Offline" state.
b. The HADR resource group will be locked.

Direct your attention towards the "Lock" placed on the HADR resource group after the failover. This lock indicates that the HADR databases are no longer in "peer" state. No further actions will be taken on this resource group in case of any more failures.

4) Plug the old primary machine (linux01) back in.

5) As soon as the old primary machine comes back up, we expect **reintegration** to occur:

a. The old primary DB2 instance will be started automatically.
b. The old primary database will now be activated as a "standby".
c. HADR replication will resume.
d. As soon as the HADR system reaches "peer" state, the lock from the HADR resource group will be removed.

6.2 Deactivating the HADR Databases

A. Deactivating the standby Database:

1) Issue the following command on the standby database:
"db2 deactivate db *database name* "

The deactivate command will stop data replication and the HADR databases will not be in "peer" state anymore. This will be reflected by a lock being placed on the HADR resource group.

2) Run "lssam" or "db2pd –ha" to examine the system reaction. You will see output similar to this:

Deactivating standby HADR database

```
root@linux02:/root# lssam
Online IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_linux01_0-rs
                '- Online IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Online IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_linux02_0-rs
                '- Online IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Request=Lock Nominal=Online
        |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs Control=SuspendedPropagated
                |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
                '- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
        '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs Control=SuspendedPropagated
                |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
                '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

Note: If the primary machine is to be powered off in such a state, a failover operation will not be performed because of the lock placed on the HADR resource group. Since HADR was not in "peer" state at the time of node failure, the standby database is not the complete copy of the primary at this point and thus not suitable to take over.

3) Activate the database again to recover from this state:

```
/home/db2inst1% db2 activate database HADRDB
```

B. Deactivating the primary Database:

1) Issue the following command on the primary database:
"db2 deactivate db *database name* "

2) The deactivation of the primary database will cause the HADR resource on the primary node to go offline, and a lock to be placed on the HADR resource group to indicate replication failure.

3) Run the "lssam" or the "db2pd –ha" command. You should see that the state of the resources is as shown below:

```
root@linux01:/root# lssam
Online IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Online
      '- Online IBM.Application:db2_db2inst1_linux01_0-rs
            '- Online IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Online IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
      '- Online IBM.Application:db2_db2inst1_linux02_0-rs
            '- Online IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Pending online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Nominal=Online
      |- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs Control=SuspendedPropagated
            |- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
            '- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
      '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs Control=SuspendedPropagated
            |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
            '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

The HADR resources on the primary and the standby machines will be offline. The database must be activated again to resume replication and to return the HADR resource group to unlock and come online.

/home/db2inst1% db2 activate database HADRDB

6.3 DB2 Failures

This section discusses the HA system configuration reaction to killing or stopping the DB2 daemons on either the primary or standby nodes.

A. Killing the DB2 Instance:

1) Issue the db2_kill command on the primary instance to kill the DB2 daemons.

2) Run the "lssam" or the "db2pd –ha" command to examine the resources. You should see output similar to this:

```
Pending online IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Online
        '- Pending online IBM.Application:db2_db2inst1_linux01_0-rs
                '- Pending online IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Online IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_linux02_0-rs
                '- Online IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Pending online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Nominal=Online
        |- Pending online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
                |- Pending online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
                '- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
        '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs
                |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
                '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

3) The HADR resource and the DB2 resource on the primary node will be in the "Pending Online" state.

4) Repeatedly issue the "lssam" or the "db2pd –ha" command. The cluster manager will automatically start the DB2 instance, and activate the HADR database. This will result in the "Pending Online" state changing to the "Online" state.

We can expect the same system reaction to a db2_kill on the standby instance.

B. Failing the DB2 instance on the standby machine:

1) Log on to the standby instance and rename the db2start executable:

```
/home/db2inst1% mv $HOME/sqllib/adm/db2star2 db2star2.mv
```

2) Issue the db2_kill command on the standby instance.

3) The standby DB2 resource will assume the "Pending Online" state. The cluster manager will try to start the DB2 instance indefinitely, but will fail because of the missing executable.

4) A timeout will occur, and any further start attempts on the DB2 resource will stop. This will be indicated by the "Pending Online" state changing into the "Failed Offline" state, as shown in the output below:

Note: It might take 4-5 minutes for the DB2 resource timeout to occur.

```
Online IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_linux01_0-rs
                '- Online IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Failed offline IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
        '- Failed offline IBM.Application:db2_db2inst1_linux02_0-rs
                '- Failed offline IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Request=Lock Nominal=Online
        |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs Control=SuspendedPropagated
                |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
                '- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
        '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs Control=SuspendedPropagated
                |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
                '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

Also, the lock placed on the HADR resource group will indicate replication failure. To recover from this state:

5) Rename the executable back to its original name:

```
/home/db2inst1% mv $HOME/sqllib/adm/db2star2.mv db2star2
```

6) Issue the following command with root authority on either the primary or the standby nodes:

```
resetrsrc –s "Name = '<Standby DB2 instance resource name>' \
AND NodeNameList = {'<standby node name>'}" IBM.Application
```

In our case, the command will look like:

```
resetrsrc –s "Name = 'db2_db2inst1_linux02_0-rs' \
AND NodeNameList = {'linux02'}" IBM.Application
```

This command will reset the "Failed Offline" flag on the standby instance resource and force the cluster manager to start the instance again. The standby DB2 instance will come online, and the standby database will be activated automatically. Replication will resume and once the databases are in "peer" state, the lock from the HADR resource group will be removed.

C. Failing the DB2 instance on the primary machine:

1) Log on to the primary instance and rename the db2start executable:

```
/home/db2inst1% mv $HOME/sqllib/adm/db2star2 db2star2.mv
```

2) Issue the db2_kill command on the primary instance.

3) The primary HADR and DB2 instance resources will assume the "Pending Online" state.

```
Pending online IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Online
      '- Pending online IBM.Application:db2_db2inst1_linux01_0-rs
            '- Pending online IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Online IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
      '- Online IBM.Application:db2_db2inst1_linux02_0-rs
            '- Online IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Pending online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Nominal=Online
      |- Pending online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
            |- Pending online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
            '- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
      '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs
            |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
            '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

4) The cluster manager will try to start the DB2 instance and activate the HADR database:

a. The database activation will fail because the DB2 instance is not online, and a failover will follow. The virtual IP address will be moved to the standby node, and a takeover operation will cause the standby database to assume the primary role. After the failover, the HADR resource on the old primary node will assume a "Failed Offline" state.

b. The cluster manager will continue to try to bring the DB2 instance resource online on what is now the old primary machine. Eventually, a timeout will occur, and this "Pending Online" state will change into the "Failed Offline" state.

```
Failed offline IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Online
      '- Failed offline IBM.Application:db2_db2inst1_linux01_0-rs
            '- Failed offline IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Online IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
      '- Online IBM.Application:db2_db2inst1_linux02_0-rs
            '- Online IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Nominal=Online
      |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
            |- Failed offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
            '- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
      '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs
            |- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
            '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

To recover from this scenario:

5) Rename the db2start executable to its original name:

/home/db2inst1% mv $HOME/sqllib/adm/db2star2.mv db2star2

6) We must reset the HADR and DB2 resources on the old primary machine to get rid of the "Failed Offline" flag. This is done by issuing the following commands (with root authority) in the given order on either the standby or the primary node:

```
resetrsrc –s "Name = '<DB2 instance resource name on old primary>' \
 AND NodeNameList = {'<old primary instance node name>'}" IBM.Application

resetrsrc –s "Name = '<HADR Resource Name>' \
AND NodeNameList = {'<old primary node name>'}" IBM.Application
```

In our case, the commands will look like:

```
resetrsrc –s "Name = 'db2_db2inst1_linux01_0-rs' \
AND NodeNameList = {'linux01'}" IBM.Application

resetrsrc –s "Name = 'db2_db2inst1_db2inst1_HADRDB-rs'\
AND NodeNameList = {'linux01'}"  IBM.Application
```

It is important to specify the commands exactly as above. Reintegration will occur automatically, and the old primary database will assume the new standby role. After the system has reached "peer" state, the lock from the HADR resource group will be removed.

6.4 Manual Instance Control (db2stop, db2start)

For various reasons, you may want to stop and start either the standby or primary instance.

A. Issuing db2stop and db2stop force commands on the standby instance:

1) Issue the db2stop command on the standby machine. The following error will be encountered and the instance will *not* be stopped:

```
/home/db2inst1% db2stop
05/25/2009 14:31:09     0   0   SQL1025N  The database manager was not
stopped because databases are still active.
```

2) Now issue the *db2stop force* command on the standby instance. The command will go through successfully and the instance will be stopped.

```
/home/db2inst1% db2stop force
05/25/2009 15:57:42     0   0   SQL1064N  DB2STOP processing was successful.
SQL1064N  DB2STOP processing was successful.
```

This will cause HADR replication to halt. To reflect this, we can expect a lock to be placed on the HADR resource group and the standby instance resource group. The sample output below shows the effect of the 'db2stop force' command issued on the standby instance.

db2stop force issued on the standby instance

```
Online IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_linux01_0-rs
                '- Online IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Pending online IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Request=Lock Nominal=Online
        '- Offline IBM.Application:db2_db2inst1_linux02_0-rs Control=SuspendedPropagated
                '- Offline IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Request=Lock Nominal=Online
        |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs Control=SuspendedPropagated
                |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
                '- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
        '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs Control=SuspendedPropagated
                |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
                '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

To recover from this state:

3) Start the standby DB2 instance and activate the database again:

```
/home/db2inst1% db2start; db2 activate database HADRDB
```

B. Issuing db2stop and db2stop force commands on the primary machine:

1) Issue the db2stop command on the primary machine. The following error will be encountered and the instance will *not* be stopped:

```
/home/db2inst1% db2stop
05/25/2009 14:31:09     0   0   SQL1025N  The database manager was not
stopped because databases are still active.
```

2) Now issue the *db2stop force* command on the primary instance. The command will go through successfully and the instance will be stopped.

```
/home/db2inst1% db2stop force
05/25/2009 15:57:42     0   0   SQL1064N  DB2STOP processing was successful.
SQL1064N  DB2STOP processing was successful.
```

This will cause HADR replication to halt. To reflect this, we can expect a lock to be placed on the HADR resource group and the primary instance resource group. The sample output below shows the effect of the 'db2stop force' command issued on the primary instance.

db2stop force issued on the primary instance

```
Pending Online IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Request=Lock Nominal=Online
      '- Offline IBM.Application:db2_db2inst1_linux01_0-rs Control=SuspendedPropagated
            '- Offline IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Online IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
      '- Online IBM.Application:db2_db2inst1_linux02_0-rs
            '- Online IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Pending online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Request=Lock Nominal=Online
      |- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs Control=SuspendedPropagated
            |- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
            '- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
      '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs Control=SuspendedPropagated
            |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
            '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

To recover from this state:

3) Start the DB2 instances and activate the database again:

```
/home/db2inst1% db2start; db2 activate database HADRDB
```

6.5 Manual HADR Control (Takeover, Takeover by force)

Two types of HADR takeover commands can be used on the standby machines, which are **takeover** and **takeover by force**.

A. Issuing takeover commands on the standby Machine:

Issue the takeover command on the standby instance (linux02). The command will complete successfully and the instance will switch database roles with HADR as primary.

```
/home/db2inst1% db2 takeover hadr on db HADRDB
DB20000I  The TAKEOVER HADR ON DATABASE command completed successfully.
```

Takeover issued on the standby instance

```
Online IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Online
      '- Online IBM.Application:db2_db2inst1_linux01_0-rs
            '- Online IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Online IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
      '- Online IBM.Application:db2_db2inst1_linux02_0-rs
            '- Online IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Nominal=Online
      |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
            |- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
            '- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
      '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs
            |- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
            '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

All the reintegration is handled by the takeover command.

B. Issuing takeover by force commands on the standby machine:

Issue the takeover by force command on the standby instance (linux01). The command will complete successfully and the instance will switch role with HADR as primary.

```
/home/db2inst1% db2 takeover hadr on db HADRDB by force
DB20000I  The TAKEOVER HADR ON DATABASE command completed successfully.
```

Takeover by force issued on the standby instance

```
Online IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Online
      '- Online IBM.Application:db2_db2inst1_linux01_0-rs
            '- Online IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Online IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
      '- Online IBM.Application:db2_db2inst1_linux02_0-rs
            '- Online IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Nominal=Online
      |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
            |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
            '- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
      '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs
            |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
            '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

Note that the reintegration of the old primary as the new and current standby will be automatically performed. Furthermore, the non-forced version of the TAKEOVER command should always be attempted before the forced version.

For more Information:
https://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.admin.cmd.doc/doc/r0011553.html

6.6 Network Failures

We discuss network failures in this section by simulating network interface malfunctions on the primary and the standby nodes. The two HADR configurations discussed in this paper will have different reactions to network failures.

A. Network failures in the multiple Network HADR configuration:

*1. Private network interface card failure on the **standby** machine:*

1) Unplug the eth1 cable (private network cable) connected to the standby machine.

2) Issue the "lssam" or the "db2pd –ha" command to examine the state of the resources. The resources should be in the following state:

a. The DB2 instance on the standby machine will be "Failed Offline".
b. A lock will be placed on the HADR resource group to indicate loss of "peer" state.

```
Online IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_linux01_0-rs
                '- Online IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Failed offline IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Binding=Sacrificed Nominal=Online
        '- Offline IBM.Application:db2_db2inst1_linux02_0-rs
                '- Offline IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Nominal=Online
        |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
                |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
                '- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
        '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs
                |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
                '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

To recover from this state:

1) Plug the eth1 cable back in.

2) Repeatedly issue the "lssam" or the "db2pd –ha" command and observe the system resources assume their normal states. Reintegration will occur:

a. The standby DB2 instance will be started automatically.
b. The standby DB2 HADR database will be activated.
c. HADR data replication will resume and the system will eventually reach "peer" state.

*2. Private Network Interface card failure on the **primary** machine:*

1) Unplug the eth1 cable going into the primary machine.

2) Issue the "lssam" or the db2p –ha" command to examine the state of the system resources. You should see output similar to the sample shown below:

a. The primary DB2 resource will be in a "Failed Offline" state.
b. The primary HADR resource will be in a "Pending Online" State

```
Failed offline IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Binding=Sacrificed Nominal=Online
        '- Offline IBM.Application:db2_db2inst1_linux01_0-rs
                '- Offline IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Online IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_linux02_0-rs
                '- Online IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Pending online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Nominal=Online
        |- Pending online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
                |- Pending online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
                '- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
        '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs
                |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
                '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

3) The Cluster Manager will start to execute a failover operation.

4) Repeatedly issue the "lssam" or the "db2pd –ha" command to examine the failover. The system will eventually settle down to the following state:

a. The standby HADR database will assume the primary role.
b. A lock will be placed on the HADR resource group to indicate replication failure.
c. The virtual IP address will come online on the standby machine.

This state is shown in the output below:

```
Offline IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Online
        '- Failed offline IBM.Application:db2_db2inst1_linux01_0-rs
                '- Failed offline IBM.Application:db2_db2inst1_linux01_0-rs:linux01 Node=Offline
Offline IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_linux02_0-rs
                '- Online IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Nominal=Online
        |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
                |- Failed offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01 Node=Offline
                '- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
        '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs
                |- Failed offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01 Node=Offline
                '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

To recover from this state:

1) Plug the eth1 cable back into the old primary machine.

2) Repeatedly issue the "lssam" command to examine the state of the resources.

3) The system will eventually settle down to the following state:

a. The old primary DB2 instance will come online.
b. Reintegration will take place.
c. The old primary HADR database resource state in the cluster, however, will remain "Failed Offline".

This state is shown in the output below:

```
Online IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_linux01_0-rs
                '- Online IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Online IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_linux02_0-rs
                '- Online IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Nominal=Online
        |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
                |- Failed offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
                '- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
        '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs
                |- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
                '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

In this particular case, even when reintegration is successful, the HADR resource on the primary node remains "Failed Offline". We must *reset* this resource to reset the "Failed Offline" state.

Issue the following command on the primary node as *root:*

resetrsrc -s "Name = '*HADR Resource Name*' \
AND NodeNameList = {'*old Primary node name*'}" IBM.Application

In our case, this command will look like:

resetrsrc -s "Name = 'db2_db2inst1_db2inst1_HADRDB-rs' \
AND NodeNameList = {'linux01'}" IBM.Application

3. Public network failure on the standby/primary machines:

In the case where either one of the public network NICs (eth0) fails on the primary or the standby machines, we can expect to see the same system reaction as described in the case of the private NIC failures.

B. Network failures on the single Network HADR configuration:

In the case where only one network is available for data replication and client connections, the reaction of our HA configuration to network failures will be different. Our second HADR configuration setup had 1 NIC per machine and formed one single public network to accommodate client connections as well as data replication.

1. Standby Network Interface failure:

1) Unplug the eth0 cable from the standby machine.

The machine will now be completely isolated from the rest of the network cloud preventing the Cluster Manager from communicating with it. To avoid split brain, this machine will reboot right after the NIC failure is detected.

2) The following series of events will follow:

a. The primary machine will ping and acquire the quorum and will continue to operate.
b. The clients will continue to connect to the primary database.

3) Issue the "lssam" or the "db2pd –ha" command to examine the state of the resources. You will see output similar to this:

```
Online IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_linux01_0-rs
                '- Online IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Failed offline IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Control=StartInhibited Nominal=Online
        '- Failed offline IBM.Application:db2_db2inst1_linux02_0-rs
                '- Failed offline IBM.Application:db2_db2inst1_linux02_0-rs:linux02 Node=Offline
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Nominal=Online
        |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
                |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
                '- Failed offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02 Node=Offline
        '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs
                |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
                '- Failed offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02 Node=Offline
```

Note that this failure scenario will be inferred as a standby node failure rather than a network failure.

4) While the machine is coming back up, plug the network cable back into eth0.

Note: This machine will not come online and will hang in the boot process until the network cable is plugged back in.

5) Issue the "lssam" or the "db2pd –ha" command repeatedly as the standby node comes back online.

2. Primary network interface failure:

1) Unplug the eth0 NIC on the primary machine.

2) Since there is only one NIC available to the machine, the machine will now be completely isolated from the external network cloud. This will cause it to reboot.

3) A failover operation will be executed, and the resources will settle down to the state shown below:

```
Failed offline IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Control=StartInhibited Nominal=Online
        '- Failed offline IBM.Application:db2_db2inst1_linux01_0-rs
                '- Failed offline IBM.Application:db2_db2inst1_linux01_0-rs:linux01 Node=Offline
Online IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_linux02_0-rs
                '- Online IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Pending online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Request=Lock Nominal=Online
        |- Pending online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
                |- Failed offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01 Node=Offline
                '- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
        '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs
                |- Failed offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01 Node=Offline
                '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

Note that this failure will be inferred as a primary node failure as opposed to a network failure.

4) As the old primary machine is booting back up, plug the Ethernet cable back into eth0.

Note: This machine will not come online and may hang in the boot process until the network cable is plugged back in.

5) Repeatedly issue the "lssam" or the "db2pd –ha" commands to examine the state of the resources.

6) As soon as the old primary machine comes back online, we can expect *reintegration* to occur again. The old primary database will assume the new standby role, and HADR data replication will eventually reach "peer" state.

7. Maintenance

7.1 Disabling High Availability

To disable the HA configuration for a particular instance, the *db2haicu –disable* command can be used. After issuing this command, the system will not respond to any failures and all resource groups for the instance will be locked. Any maintenance work can be performed in this state without worrying about cluster manager intervention.

To enable HA, just issue the db2haicu command again, and choose the "Yes" option when prompted to continue.

Disabling High Availability for an HA DB2 instance

```
Welcome to the DB2 High Availability Instance Configuration Utility (db2haicu).

You can find detailed diagnostic information in the DB2 server diagnostic log file called
db2diag.log. Also, you can use the utility called db2pd to query the status of the cluster
domains you create.

For more information about configuring your clustered environment using db2haicu, see the
topic called 'DB2 High Availability Instance Configuration Utility (db2haicu)' in the DB2
Information Center.

db2haicu determined the current DB2 database manager instance is db2inst1. The cluster
configuration that follows will apply to this instance.

db2haicu is collecting information on your current setup. This step may take some time as
db2haicu will need to activate all databases for the instance to discover all paths ...
Are you sure you want to disable high availability (HA) for the database instance db2inst1. This
will lock all the resource groups for the instance and disable the HA configuration parameter.
The instance will not failover if a system outage occurs while the instance is disabled. You will
need to run db2haicu again to enable the instance for HA. Disable HA for the instance
db2inst1? [1]
1. Yes
2. No
1
Disabling high availability for instance db2inst1 ...
Locking the resource group for HADR database HADRDB ...
Locking the resource group for HADR database HADRDB was successful.
Locking the resource group for DB2 database partition 0 ...
Locking the resource group for DB2 database partition 0 was successful.
Locking the resource group for DB2 database partition 0 ...
Locking the resource group for DB2 database partition 0 was successful.
Disabling high availability for instance db2inst1 was successful.
All cluster configurations have been completed successfully. db2haicu exiting ...
```

7.2 Manual takeovers

There might be situations when a DBA wants to perform a manual takeover to switch HADR database roles.

For more Information:
https://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.admin.cmd.doc/doc/r0011553.html

To do this, log on to the standby machine and type in the following command to perform a manual takeover.

db2 takeover hadr on db *database name*

Once the takeover has been completed successfully, the "lssam" or the "db2pd –ha" commands will reflect the changes. The virtual IP address will also be moved to the new primary node as part of the takeover process.

7.3 Db2haicu Maintenance mode

When a system is already configured for High Availability, db2haicu runs in maintenance mode. Typing db2haicu on the primary or standby node will produce the menu shown below. This menu can be used to carry out various maintenance tasks and change any Cluster Manager-specific, DB2-specific or network-specific values configured during the initial setup.

Db2haicu Maintenance mode

```
/home/db2inst1% db2haicu
Welcome to the DB2 High Availability Instance Configuration Utility (db2haicu).

You can find detailed diagnostic information in the DB2 server diagnostic log file called
db2diag.log. Also, you can use the utility called db2pd to query the status of the cluster
domains you create.

For more information about configuring your clustered environment using db2haicu, see the
topic called 'DB2 High Availability Instance Configuration Utility (db2haicu)' in the DB2
Information Center.

db2haicu determined the current DB2 database manager instance is db2inst1. The cluster
configuration that follows will apply to this instance.

db2haicu is collecting information on your current setup. This step may take some time as
db2haicu will need to activate all databases for the instance to discover all paths ...
When you use db2haicu to configure your clustered environment, you create cluster domains.
For more information, see the topic 'Creating a cluster domain with db2haicu' in the DB2
Information Center. db2haicu is searching the current machine for an existing active cluster
domain ...
db2haicu found a cluster domain called hadr_domain on this machine. The cluster configuration
that follows will apply to this domain.

Select an administrative task by number from the list below:
  1. Add or remove cluster nodes.
  2. Add or remove a network interface.
  3. Add or remove HADR databases.
  4. Add or remove an IP address.
  5. Move DB2 database partitions and HADR databases for scheduled maintenance.
  6. Create a new quorum device for the domain.
  7. Destroy the domain.
  8. Exit.
Enter your selection:
```

1) Type '1' and press Enter.

2) The following message will be displayed on the screen:

```
Enter your selection:
1
Do you want to review the status of each cluster node in the domain
before you begin? [1]
1. Yes
2. No
```

3) Type '1' and press Enter to review the status of the current nodes in the cluster:

```
…
1
Domain Name: hadr_domain
Node Name: linux02 --- State: Online
Node Name: linux01 --- State: Online
…
```

4) At the next prompt, type '1' and press Enter to choose the Add option. Proceed with providing the third node name.

```
Do you want to add or remove cluster nodes to or from the domain? [1]
1. Add
2. Remove
1
Enter the host name of a machine to add to the domain:
linux03
Adding node linux03 to the cluster ...
Adding node linux03 to the cluster was successful.
……
```

7.4 Stopping and Starting the Entire Domain

This is the procedure if you wish to stop and start the entire cluster domain.

Before stopping the domain:
%lssam

```
Online IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Online
      '- Online IBM.Application:db2_db2inst1_linux01_0-rs
            '- Online IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Online IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
      '- Online IBM.Application:db2_db2inst1_linux02_0-rs
            '- Online IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Nominal=Online
      |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
            |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
            '- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
      '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs
            |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
            '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

A. Stopping the domain:

1) We must take the resources Offline. This is done by issuing the following commands (with root authority) in the given order on either the standby or the primary nodes:

   /home/root# chrg –o Offline –s "Name like '%'"

2) Wait one or two minutes for all the resources to come Offline as in the output below.

/home/db2inst1% lssam

```
Offline IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Offline
        '- Offline IBM.Application:db2_db2inst1_linux01_0-rs
                '- Offline IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Offline IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Offline
        '- Offline IBM.Application:db2_db2inst1_linux02_0-rs
                '- Offline IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Offline IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Nominal=Offline
        |- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
                |- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
                '- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
        '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs
                |- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
                '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

/home/root% lsrpdomain

Name	OpState	RSCTActiveVersion	MixedVersions	TSPort	GSPort
hadr_linux_domain	Online	2.5.3.0	No	12347	12348

3) To make the domain to Offline run the "stoprpdomain" command.
/home/root/% stoprpdomain hadr_linux_domain

/home/root% lsrpdomain

Name	OpState	RSCTActiveVersion	MixedVersions	TSPort	GSPort
hadr_linux_domain	Offline	2.5.3.0	No	12347	12348

/home/root% lssam
lssam: No resource groups defined or cluster is offline!

B. Starting the domain:

1) Now we are in the process of starting the domain. To start the domain we use the command "startrpdomain".

/home/root# startrpdomain hadr_linux_domain
/home/root% lssam

```
Offline IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Offline
        '- Offline IBM.Application:db2_db2inst1_linux01_0-rs
                '- Offline IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Offline IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Offline
        '- Offline IBM.Application:db2_db2inst1_linux02_0-rs
                '- Offline IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Offline IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Nominal=Offline
        |- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
                |- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
                '- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
        '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs
                |- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
                '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

```
/home/root% lsrpdomain
Name                 OpState RSCTActiveVersion MixedVersions TSPort GSPort
hadr_linux_domain Online     2.5.3.0              No                12347  12348
```

2) We must take the resources Online. This is done by issuing the following commands (with root authority) in the given order on either the standby or the primary nodes:

```
/home/root# chrg –o online –s "Name like '%'"
```

3) Wait one or two minutes for all the resources to come Online as in the output below.

```
/home/root% lssam
```

```
Online IBM.ResourceGroup:db2_db2inst1_linux01_0-rg Nominal=Online
      '- Online IBM.Application:db2_db2inst1_linux01_0-rs
            '- Online IBM.Application:db2_db2inst1_linux01_0-rs:linux01
Online IBM.ResourceGroup:db2_db2inst1_linux02_0-rg Nominal=Online
      '- Online IBM.Application:db2_db2inst1_linux02_0-rs
            '- Online IBM.Application:db2_db2inst1_linux02_0-rs:linux02
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_HADRDB-rg Nominal=Online
      |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs
            |- Online IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux01
            '- Offline IBM.Application:db2_db2inst1_db2inst1_HADRDB-rs:linux02
      '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs
            |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:linux01
            '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:linux02
```

Now we have safely stopped and started the domain using the appropriate commands.

8. Troubleshooting

8.1 Unsuccessful Failover

In the case when a critical failure occurs on the primary cluster node, a failover action is initiated on the HADR resource group, and all HADR resources are moved to the standby machine.

If such a failover operation is unsuccessful, it will be reflected by the fact that all HADR resources residing on both the primary and the standby machines are "Failed Offline". This can be due to the following reasons:

1) The HADR_PEER_WINDOW database configuration parameter is not set to a sufficiently large value.

When moving the HADR resources during a failure, the Cluster Manager issues the following command on the standby database:

db2 takeover hadr on db *database name* by force peer window only

The peer window value is the amount of time that a takeover must be done on the standby database from the time when the primary database failed. If a takeover is not done within this "window", the above-mentioned takeover command will fail, and the standby database will not be able to assume the primary role. Recall from Section 5.1, that a value of 300 seconds is recommended for the HADR_PEER_WINDOW parameter. However, if a takeover fails on your system, you might have to update this parameter to a larger value, and try the failure scenario again. Also, make sure that the Network Time protocol (as stated in Section 4.1) is functional and the dates and times on both the standby and the primary machines are synchronized. If peer window expiration is what caused the takeover to fail, a message indicating this would be logged in the DB2 diagnostic log (see Section 8.3). At this point, you can issue the following command at the standby machine and force HADR takeover on the standby:

db2 takeover hadr on db *database name* by force

However, prior to issuing the above command, you are urged to consult the URL: https://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.admin.cmd.doc/doc/r0011553.html

2) The HADR resource group is in the "Locked" state. A lock on the HADR resource group indicates the fact that no replication is being carried out between the HADR databases and that they are not in "peer" state. In such a state, if a primary node is to fail, a failover will not be initiated. This is because at this point, the standby database cannot be trusted to be the complete copy of the primary, and hence not fit to take over.

3) Ensure that the two machines (hosting the primary and standby HADR databases) are synchronized with respect to time. Ensure NTP is running on each node and the date command, when issued simultaneously at both nodes, will return results that are equal (to within 1 second).

8.2 Db2haicu '-delete' option

Db2haicu can also be run with the "-delete" option. This option removes a system's entire HA configuration and deletes all resources in the cluster for the instance in question. If no other instance is using the domain at the time, the domain is deleted as well.

As a good practice, it is recommended to run db2haicu with the delete option on an instance before it is made highly available. This makes sure that we are starting from scratch and not building on top of leftover resources.

For example, when running db2haicu with an XML file, any invalid attribute in the file will cause db2haicu to exit with a non-zero error code. However, before db2haicu is run again with the corrected XML file, one can run the –delete option to make sure that any temporary resources created during the initial run are cleaned up.

Note that the db2haicu –delete option will leave the instances and the HADR replication unaffected. That is, it will not stop the db2 instances of HADR replications. However, any IP addresses that were highly available are removed and no longer present after the db2haicu –delete command completes.

8.3 The "syslog" and the DB2 server diagnostic log file (db2diag.log)

For debugging and troubleshooting purposes, the necessary data is logged in two files: the syslog, and the DB2 server diagnostic log file (db2diag.log).

Any DB2 instance and database-related errors are logged in the db2diag.log file. The default location of this file is $HOME/sqllib/db2dump/db2diag.log, where $HOME is the DB2 instance home directory. This location can be changed from the following command:

db2 update dbm cfg using DIAGPATH <*new diagnostic log location*>

Also, there are a total of 5 diagnostic levels that can be set to control the amount of data logged. These range from 0-4, where level 0 indicates the logging of only the most critical errors, and level 4 indicates the maximum amount of logging possible. We recommend diagnostic level 3 to be set on both the primary and the standby instances. The command to change the diagnostic level of an instance is:

db2 update dbm cfg using DIAGLEVEL <*diagnostic level number*>

The syslog or the system log is used to log information from the cluster manager. The location and level of the logged messages can be defined in the /etc/syslog.conf file. For example, the following statement in syslog.conf will enable logging of all messages in the /var/log/messages file:

. -/var/log/messages

IBM Canada
8200 Warden Avenue
Markham, ON
L6G 1C7
Canada