

IPWorks Geographic Redundancy

OPERATING INSTRUCTIONS

Copyright

© Ericsson AB 2017, 2018. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	Introduction	1
1.1	Target Groups	1
1.2	Related Information	1
1.3	Scope	1
2	Conceptual Overview	3
2.1	Redundancy with Double Provisioning	3
2.2	Redundancy with Single Provisioning	3
3	Operating Instructions	5
3.1	Install Redundancy with Double Provisioning	5
3.1.1	Install the Standby IPWorks Cluster System	5
3.1.2	Switch the Traffic to the Standby IPWorks Cluster System	8
3.1.3	Recover the Broken IPWorks Cluster System	9
3.2	Install Redundancy with Single Provisioning	9
3.2.1	Configure MySQL Cluster SQL Node	10
3.2.2	Grant Privileges for SQL Nodes for Remote Site	11
3.2.3	Change Master-Host and Setting Binlog	12
3.2.4	Verify the Configuration	13
3.2.5	Switch the Traffic to the Standby IPWorks Cluster System	15
3.3	Remove Geographic Redundancy with Single Provisioning	16
	Reference List	19





1 Introduction

This document provides information on Geographic Redundancy for IPWorks.

1.1 Target Groups

This guide is intended for personnel working with Ericsson IPWorks. The following prior knowledge is required:

- Intermediate Linux and UNIX skills
- Concepts, terminologies, and telecommunication abbreviations, such as TCP/IP and packet data networks.
- Familiar with IPWorks configuration operations

Note: If IPv6 address is needed, first do the configuration according to the section *Configuring IPv6 OAM/Provision Network* in *IPWorks Initial Configuration*, then use IPv6 address in the above configure process instead of IPv4 address.

1.2 Related Information

Trademark information, typographic conventions, and definition and explanation of abbreviations and terminology can be found in the following documents:

- *Trademark Information*
- *Typographic Conventions*
- *Glossary of Terms and Acronyms*

1.3 Scope

This document focuses only on the Operating Instructions for the monolithic deployments, including DNS/ENUM/AAA (see Section 3 on page 5).

In layered deployments, the user data of IPWorks ENUM/AAA/PKI Front End (FE) in both sites are stored by CUDB deployment.

Thus Geographic Redundancy for IPWorks Layered deployments is out of the scope of this document.



Caution!

When Geography Redundancy is enabled, the function IP Allocation of AAA Radius is not available.



2 Conceptual Overview

The Geographic Redundancy solution enables two geographically separated sites to set up an independent and identical IPWorks cluster system respectively (one is the primary, the other is the standby). Once the primary system fails to work, the standby system takes over the services.

IPWorks supports two redundancy scenarios for monolithic deployments:

- Double Provisioning, see Section 2.1 on page 3
- Single Provisioning, see Section 2.2 on page 3

2.1 Redundancy with Double Provisioning

Take Figure 1 as an example, the primary cluster system is set up in Site A, the standby cluster system is set up in Site B with identical data.

- EDA performs double provisioning of AAA/ENUM data to the primary and standby IPWorks cluster systems so that both contain the same user data.

For details about provisioning data, refer to *IPWorks EDA CLI Interface*.

- Configuration data consistency is achieved by manual configuration, it is out of the scope of this document.

For details about the configuration data (such as, DNS or ENUM management) in *IPWorks Configuration Management* and *Configure DNS and ENUM*.

If the primary IPWorks cluster system in Site A fails to work, the standby IPWorks cluster system in Site B is able to take over all DNS/ENUM/AAA services.

2.2 Redundancy with Single Provisioning

This section describes the Geographic Redundancy with single provisioning.

Take Figure 3 as an example, the primary system is set up in Site A and standby system is set up in Site B. If Site A fails to work because of a disaster, Site B takes over all the AAA services..

This solution currently supports AAA user data and ENUM user data, which are shown in Table 1.

*Table 1 AAA/ENUM User Data*

User Data Category	Object
AAA	AAANSUser, AAAUser, AAPPolicy, AAAUserGroup
ENUM	enumzone, enumview, enumzvrel, enumacl, destnode, enumdnrange, enumdnsched

All of objects shown in above table must be configured/provisioned in IPWCLI and they will be replicated to another site automatically when Geographic Redundancy is enabled.

Note: All other objects which are not in the Table 1 will not be replicated automatically. So, they must be configured manually by IPWCLI in both sites.

- EDA provisions AAA data to the primary IPWorks system, and the data is replicated to the standby system so that both systems contain the same user data.

For details about the provisioning data, refer to *IPWorks EDA CLI Interface*.

- Except for the provisioning data, manually configure other configuration data in both sites respectively.

For details about the configuration data, refer to *IPWorks Configuration Management*.



3 Operating Instructions

This section describes how to configure Geographic Redundancy for IPWorks system with following scenarios:

- Install the redundancy with double provisioning, see Section 3.1 on page 5.
- Install the redundancy with single provisioning, see Section 3.2 on page 9.

3.1 Install Redundancy with Double Provisioning

The Figure 1 show the network of Geographic redundancy with double provisioning.

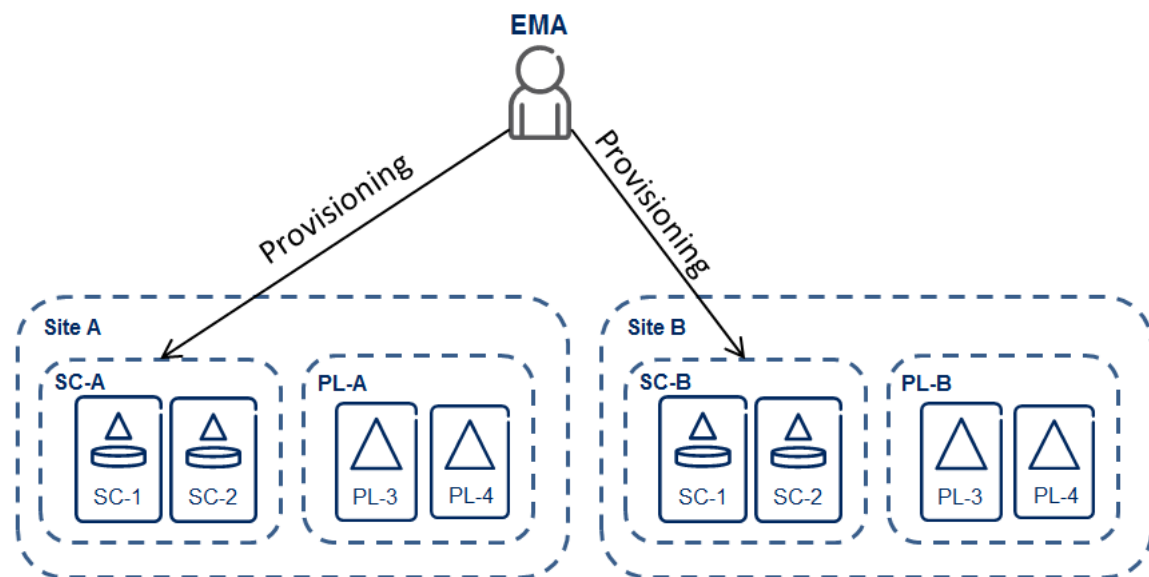


Figure 1 Geographic Redundancy Network

3.1.1 Install the Standby IPWorks Cluster System

This section provides the instructions for configuring a standby IPWorks cluster system in the Geographic Redundancy network.

As Figure 3 shows, one of the IPWorks cluster system (Site A) contains two System Controller nodes (SC-A) and two Payload nodes (PL-A). EDA provisions the user data to SC-A.

The IPWorks Cluster System in Site B works as the standby IPWorks cluster system.



1. Install the IPWorks cluster system on site B.

For more detailed information, refer to IPWorks Deployment Guide.

After the installation, stop "SS process" on SC-B.

2. Stop all provisioning and wait until all provisioning activities are completed.
3. Configure IPWorks cluster with SC-A and SC-B in EDA by using the cluster strategy "Active/Active".

For more detailed information, refer to the "ActiveActive" part in section *Network Element Management* in document **Function Specification Subscriber Activation**, Reference [12].

4. Back up SC-A database as below.

Log on to SC-A and use `mysqldump` command to back up the SC-A database.

```
# /usr/local/mysql/bin/mysqldump -P 3307 -h ipw_sql
--net-buffer-length=100K -c -n -t --single-transaction
ipworks > /export/ipworks_dump_net_100K.sql
```

```
# /usr/local/mysql/bin/mysqldump -P 3307 -h ipw_sql
--net-buffer-length=100K -c -n -t --single-transaction
ipw_prov_aaa > /export/ipw_prov_aaa_dump_net_100K.sql
```

```
# /usr/local/mysql/bin/mysqldump -P 3307 -h ipw_sql
--net-buffer-length=100K -c -n -t --single-transaction
ipw_enum > /export/ipw_enum_dump_net_100K.sql
```

Then, the backup database files `ipworks_dump_net_100K.sql`, `ipw_prov_aaa_dump_net_100K.sql` and `ipw_enum_dump_net_100K.sql` are stored in the `/export` directory in SC-A.

5. Do the following steps to import the backup database on SC-B.

- a. On SC-B, use `mysql` command to delete all records in the `user` table.

```
# /usr/local/mysql/bin/mysql -P 3307 --protocol=TCP
CP -e 'DELETE FROM user' ipworks
```

- b. Copy the database dump file to the `/import` directory on SC-B machine, and use `mysql` command to restore the database from the backup database file.

```
# /usr/local/mysql/bin/mysql -P 3307 --protocol=TCP
CP -f ipworks < /import/ipworks_dump_net_100K.sql
```



```
# /usr/local/mysql/bin/mysql -P 3307 --protocol=
TCP -f ipw_prov_aaa < /import/ipw_prov_aaa_dump
_net_100K.sql

# /usr/local/mysql/bin/mysql -P 3307 --protocol=
TCP -f ipw_enum < /import/ipw_enum_dump_net_100
K.sql
```

6. Start SS process on SC-B.
7. Update data and start the DNS Server Manager to connect with servers on SC-B.

- a. # ipwcli

```
IPWorks> list dnsserver
```

Check the `dnsserver` names in the output, and modify IP address as the same with the PL IP address in Site B one by one.

```
IPWorks> modify dnsserver <dnsserver name> -set
address=169.254.100.3 or 169.254.100.4
```

Note: Where 169.254.100.3 is the IP address of PL3 and 169.254.100.4 is the IP address of PL4.

- b. Start the corresponding DNS Server and DNS Server Manager, and update it through IPWorks CLI.

```
# ipw-ctr start dnssm <PL hostname>
```

```
# ipw-ctr start dns <PL hostname>
```

```
# ipwcli
```

```
IPWorks> select dnsserver <dnsserver name>
```

```
IPWorks> update -rebuild=true
```

8. Update data and start the ENUM server on SC-B.

- a. # ipwcli

```
IPWorks> list enumserver
```

Check the `enumserver` IDs in the output, and modify IP address as the IP address of PS in Site B one by one.

```
IPWorks> modify enumserver <enumservice-ID> -set
address=169.254.100.3 or 169.254.100.4
```

- b. Start the ENUM server.



```
# ipw-ctr start enum <PL hostname>
```

```
start enum ==> success
```

Note: Where 169.254.100.3 is the IP address of PL3 and 169.254.100.4 is the IP address of PL4.

9. Start provisioning. Then EDA spools all commands for SC-B.
10. Set EDA to `on` status, then SC-B runs and EDA double provisions the user data to both sites (SC-A and SC-B) at the same time. This makes sure that the two sites contain the same user data.

3.1.2

Switch the Traffic to the Standby IPWorks Cluster System

If Site A is down because of major disaster, such as, flooding, earthquake, and so on. Site B is able to take over all traffic (see Figure 2).

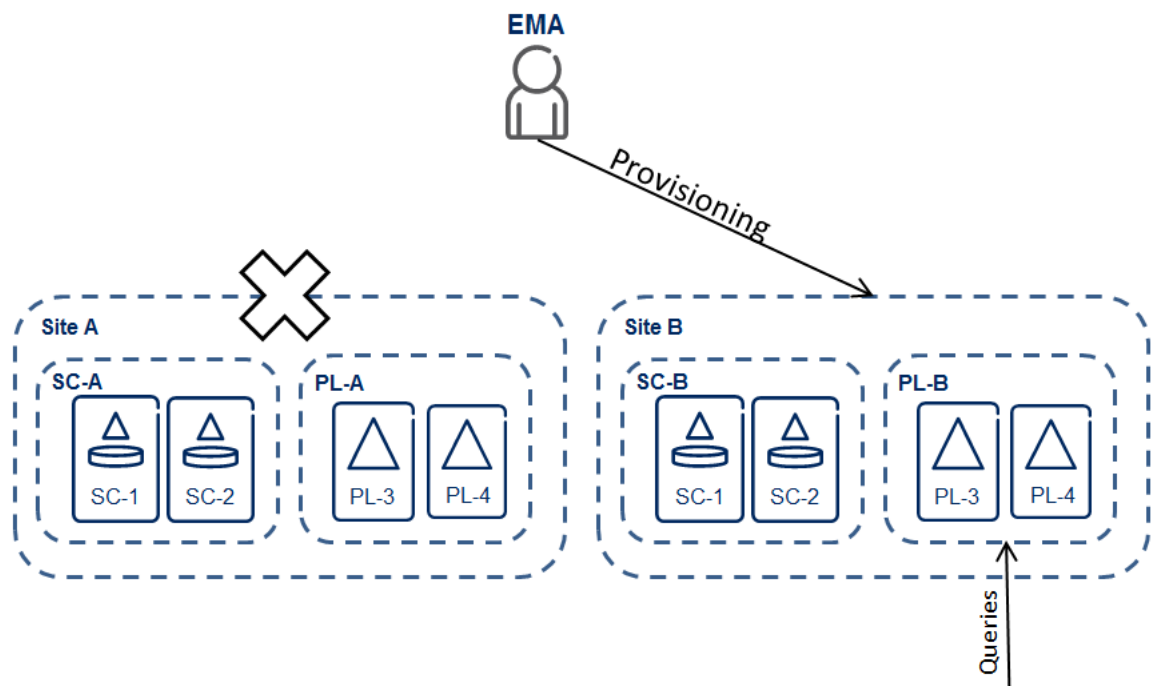


Figure 2 Switch the Traffic to the Standby IPWorks Cluster System

IPWorks clients shall switch to send queries to the standby IPWorks system in Site B after Site A is down.

Additionally, set SC-A to the `Off` status in EDA so that EDA will continue to provision the user data to SC-B. The reason is that EDA returns error when the provisioning fails on any node with the `On` status.



3.1.3 Recover the Broken IPWorks Cluster System

This section gives instructions for recovery of the broken IPWorks cluster system from the redundant one. If SC-A in IPWorks cluster system is broken, SC-B in peer system can be used to recover the broken system.

Follow the steps below to recover the IPWorks system on Site A.

1. Repair the IPWorks cluster system on site A.
2. Dump the database on SC-B, and restore the database file to SC-A, see Section 3.1.1 on page 5.

3.2 Install Redundancy with Single Provisioning

This scenario is applicable when you have two sites deployed as 2 SC nodes + 2 PL nodes. Take below figure as example, they are Site A and Site B. In this scenario, Failover mode must be configured for IPWorks cluster SC-A and SC-B in EDA.

For more detailed information, refer to the “Failover” part in section *Network Element Management* in document **Function Specification Subscriber Activation**, Reference [12].

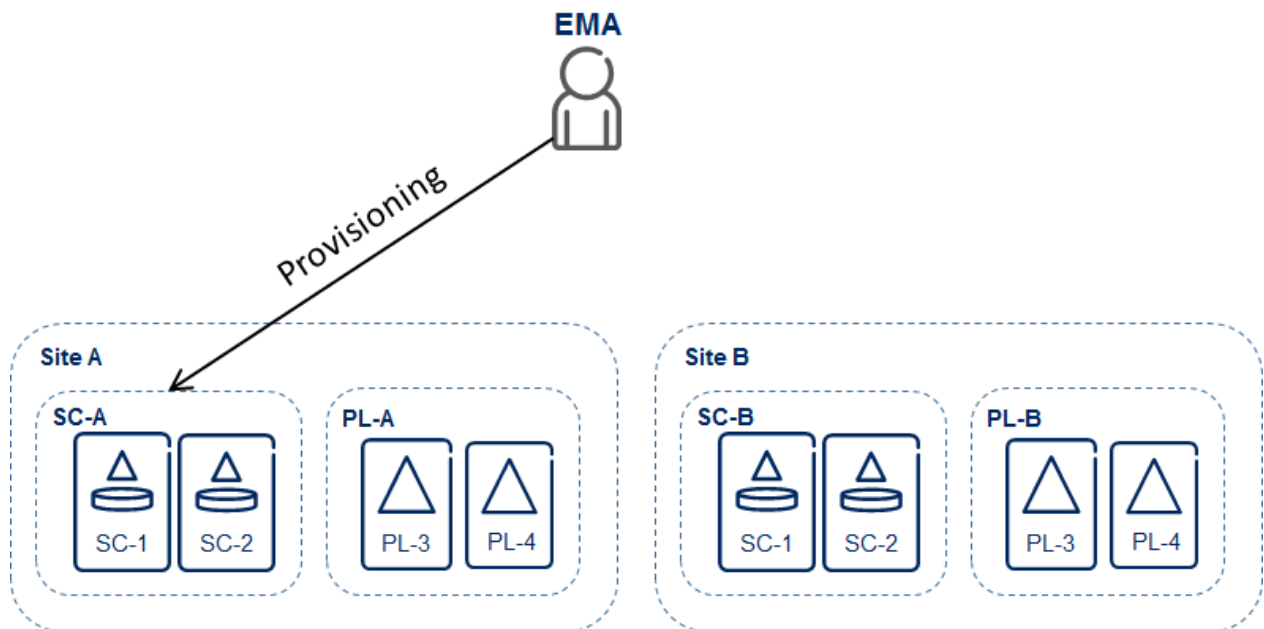


Figure 3 Geographic Redundancy with Single Provisioning

Note: From the graphic, Site A is primary system, Site B is standby system



3.2.1 Configure MySQL Cluster SQL Node

Do the following steps on both Site A and Site B.

1. Stop MySQL Cluster SQL Node on both SC-1 and SC-2.

```
#/etc/init.d/ipworks.mysql stop-sqlnode
```

2. Make Sure that below items exist in the configuration file

```
# vim /etc/ipworks/mysql/confs/ipworks_sqlnode.conf
```

```
server-id=2
slave-skip-errors=1062,1032,1590
replicate-do-db=ipw_prov_aaa
replicate-do-table=ipw_prov_aaa.aaansduser
replicate-do-table=ipw_prov_aaa.aaauser
replicate-do-table=ipw_prov_aaa.aaapolicy
replicate-do-table=ipw_prov_aaa.aaausergroup
replicate-do-table=ipw_prov_aaa.aaauser_policy
replicate-do-table=ipw_prov_aaa.aaauser_groupname
replicate-do-table=ipw_prov_aaa.aaausergroup_policy
replicate-do-db=ipw_enum
replicate-do-table=ipw_enum.ENUMDNRRANGE
replicate-do-table=ipw_enum.ENUMDNSCHED
replicate-do-table=ipw_enum.ENUMZONE
replicate-do-table=ipw_enum.ENUMZVREL
replicate-do-table=ipw_enum.ENUMVIEW
replicate-do-table=ipw_enum.ENUMACL
replicate-do-table=ipw_enum.DESTNODE
log-slave-updates
log-bin=
sync_binlog=1
binlog_format=MIXED
expire_logs_days=3
binlog-do-db=ipw_prov_aaa
binlog-do-db=ipw_enum
slave-net-timeout=10
```

Note:

- The name of table name and database are all case sensitive.
- Make sure all the # from the above parameters are removed.
- `expire_logs_day` is configured as 3. If the data replication between the two sites is down, it must be recovered within 3 days. Otherwise it is possible that some provision operations will be lost on one of the sites.



3. Make sure that the value of item `server-id` in the configuration file `/etc/ipworks/mysql/confs/ipworks_sqlnode.conf` is unique for each site. For example, if `server-id=2` is set for Site A, then `server-id` MUST NOT be set as 2 for Site B.
4. Check the value of `server-uuid` in `/cluster/ipworks/mysql-cluster/sqlnode/auto.cnf` on both Site A and Site B as below:

- a. Execute following command on both Site A and Site B:

```
SC-1:~ # cat /cluster/ipworks/mysql-cluster/sql
node/auto.cnf

[auto]
server-uuid=d90ea29a-8525-11e6-b42d-021020000200
```

- b. If the value of `server-uuid` is same on Site A and Site B, then, follow the instructions below on Site A:

On SC-1, delete the file `/cluster/ipworks/mysql-cluster/sqlnode/auto.cnf`.

5. Start MySQL Cluster SQL node on both SC-1 and SC-2.

```
#/etc/init.d/ipworks.mysql start-sqlnode
```

6. Do step 4 again to check the value of `server-uuid`, make sure that the value is unique for each site.

3.2.2

Grant Privileges for SQL Nodes for Remote Site

1. Run the following command on SC-1 of Site A.

```
# mysql -P 3307 --protocol=tcp -h ipw_sql

mysql> grant all privileges on *.* to 'ipworks'@'<OAM IP
of SC-1 in Site B>' identified by 'ipworks';

mysql> grant all privileges on *.* to 'ipworks'@'<OAM IP
of SC-2 in Site B>' identified by 'ipworks';
```

2. Run the following command on SC-1 of Site B.

```
# mysql -P 3307 --protocol=tcp -h ipw_sql

mysql> grant all privileges on *.* to 'ipworks'@'<OAM IP
of SC-1 in Site A>' identified by 'ipworks';

mysql> grant all privileges on *.* to 'ipworks'@'<OAM IP
of SC-2 in Site A>' identified by 'ipworks';
```



3.2.3 Change Master-Host and Setting Binlog

1. On Site A, record the File and Position as BINLOG_NAME_SITEA and BINLOG_POS_SITEA.

```
mysql> show master status;
```

File	Position	Binlog_Do_DB	Binlog_Ignore_DB	Executed_Gtid_Set
sqlnode-bin.000061	1137	ipw_prov_aaa,ipw_enum		

1 row in set (0.00 sec)

2. On Site B, record the File and Position as BINLOG_NAME_SITEB and BINLOG_POS_SITEB

```
mysql> show master status;
```

File	Position	Binlog_Do_DB	Binlog_Ignore_DB	Executed_Gtid_Set
sqlnode-bin.000027	1744	ipw_prov_aaa,ipw_enum		

1 row in set (0.00 sec)

3. On Site A, change the master-host to moveable IP of MySQL Cluster SQL node on Site B.

```
# mysql -P 3307 --protocol=tcp -h ipw_sql
```

```
mysql> stop slave;
```

```
mysql> change master to master_host='<MIP_PROV_IP
of Site B>', master_log_file='<BINLOG_NAME_SITEB>',
master_log_pos=<BINLOG_POS_SITEB>, master_user='ipworks',
master_password='ipworks', master_port=3307,
master_retry_count=86400, master_connect_retry=5;
```

Where: <MIP_PROV_IP> represents movable IP address for provisioning traffic. For more information, refer to *IPWorks Network Connectivity Overview*.

```
mysql> start slave;
```

```
mysql> exit;
```

4. On Site B, change master-host to moveable IP of MySQL Cluster SQL node on Site A.

```
# mysql -P 3307 --protocol=tcp -h ipw_sql
```

```
mysql> stop slave;
```




```
mysql> change master to master_host='<MIP_PROV_IP
of Site A>', master_log_file='<BINLOG_NAME_SITEA>',
master_log_pos=<BINLOG_POS_SITEA>, master_user='ipworks',
master_password='ipworks', master_port=3307,
master_retry_count=86400, master_connect_retry=5;

mysql> start slave;

mysql> exit;
```

3.2.4 Verify the Configuration

1. On SC-1 of both Site A and Site B, check the status of slave SQL node.

- a. Run the following command.

```
# mysql -P 3307 --protocol=tcp -h ipw_sql

mysql> show master status;
```

Record the File and Position for the local master.

- b. Run the following command.

```
# mysql -P 3307 --protocol=tcp -h ipw_sql

mysql> show slave status\G
```

Confirm that the following statements are correct by checking the command output:

- The `Master_Host` is the moveable IP of MySQL Cluster SQL Node in remote site.
- The value of `Slave_IO_Running` and `Slave_SQL_Running` are both Yes.
- The `Master_Log_File` and `Read_Master_Log_Pos` match File and Position of the remote master from the output fetched by executing the command “show master status” on remote site.

Note: This statement is only available for the scenario without provision.

- c. Exit MySQL.

```
mysql> exit;
```

2. Create one test user on any machine that can connect the SS VIP of Site A.

Take AAA user data as example:



```
# ipwcli -server=<SS VIP of Site A>
```

```
IPWorks> create AAANSUser username001 -set
```

```
password="54654";IMSI="225568997001";MSISDN="137399442
40";apn="MNC007.Mcc460.3gppnetworks.org,Server.alibaba
,mail.com.org";userStatus=enable;certificateid="123456
";certificateissuename="CN=AdminCA1, O=EJBCA Sample,
C=SE"
```

```
1 object(s) created.
```

Note: AAA user data is stored in database `ipw_prov_aaa` and ENUM user data is stored in database `ipw_enum`. So, the mysql commands in this step must be executed in right database.

3. Check this new record in MySQL Cluster on SC-1 of Site B.

```
#mysql -P 3307 --protocol=tcp -h ipw_sql
```

```
mysql> use ipw_prov_aaa;
```

```
mysql> select name, password from aaansduser;
```

```
+-----+-----+
| name          | password |
+-----+-----+
| username001   | 54654    |
+-----+-----+
1 row in set (0.00 sec)
```

4. Create another test user on any machine that can connect the SS VIP of Site B.

```
# ipwcli -server=<SS VIP of Site B>
```

```
IPWorks> create AAANSUser username002 -set
```

```
password="54654";IMSI="225568997001";MSISDN="137399442
40";apn="MNC007.Mcc460.3gppnetworks.org,Server.alibaba
,mail.com.org";userStatus=enable;certificateid="123456
";certificateissuename="CN=AdminCA1, O=EJBCA Sample,
C=SE"
```

```
1 object(s) created.
```

5. Check this new record in MySQL Cluster on SC-1 of Site A.

```
#mysql -P 3307 --protocol=tcp -h ipw_sql
```

```
mysql> use ipw_prov_aaa;
```



```
mysql> select name, password from aaansduser;
```

```
+-----+-----+
| name          | password |
+-----+-----+
| username001   | 54654    |
| username002   | 54654    |
+-----+-----+
2 rows in set (0.00 sec)
```

6. Delete the test users.

- a. Log on to the SC-1 node in Site A, and then execute following command.

```
#ipwcli -server=<SS VIP of Site A>
```

```
IPWorks> delete aaansduser username001
```

```
IPWorks> exit
```

- b. Log on to the SC-1 node in Site B, and then execute following command.

```
#ipwcli -server=<SS VIP of Site B>
```

```
IPWorks> delete aaansduser username002
```

```
IPWorks> exit
```

- c. Double check MySQL Cluster to make sure that there is no record existed in table `aaansduser`, execute the following command either on Site A or Site B.

```
# mysql -P 3307 --protocol=tcp -h ipw_sql
```

```
mysql> use ipw_prov_aaa;
```

```
mysql> select * from aaansduser;
```

```
Empty set (0.00 sec)
```

3.2.5 Switch the Traffic to the Standby IPWorks Cluster System

If Site A is down because of major disaster, such as, flooding, earthquake. Site B is able to take over all traffic (see Figure 4).

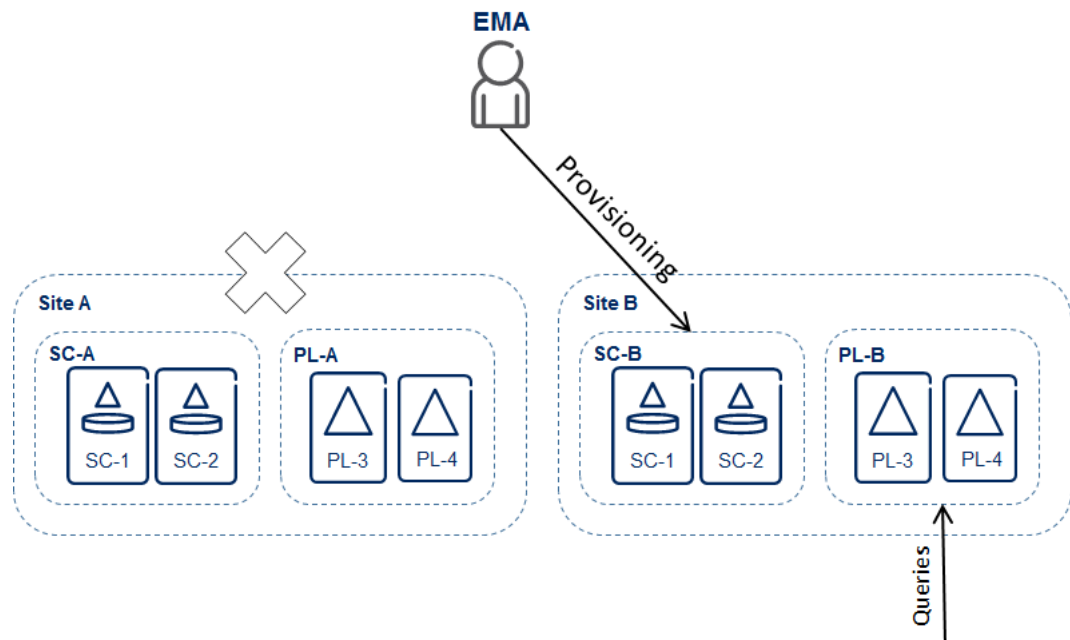


Figure 4 Switch the Traffic to the Standby IPWorks Cluster System

IPWorks clients shall switch to send queries to the standby IPWorks system in Site B after Site A is down.

Additionally, set SC-A to the `Off` status in EDA so that EDA will continue to provision the user data to SC-B. The reason is that EDA returns error when the provisioning fails on any node with the `On` status.

3.3 Remove Geographic Redundancy with Single Provisioning

When you have two sites, Site A and Site B, which are configured as Geographic Redundancy, you can follow this section to remove the configuration. So, the data will not be replicated between Site A and Site B. Stop and reset the slave for MySQL NDB Cluster on both Site A and Site B.

Stop the slave for MySQL NDB Cluster on both Site A and Site B:

1. Log on to the SC node of both sites.
2. Stop the SQL node on both SC-1 and SC-2:


```
#/etc/init.d/ipworks.mysql stop-sqlnode
```
3. Add the symbol `#` before the following lines in `/etc/ipworks/mysql/conf/nfs/ipworks_sqlnode.conf`:


```
#log-bin=
#sync_binlog=1
```



```
#binlog_format=MIXED
#expire_logs_days=3
#binlog-do-db=ipw_prov_aaa
#binlog-do-db=ipw_enum
```

4. Start the SQL node on both SC-1 and SC-2:

```
#!/etc/init.d/ipworks.mysql start-sqlnode
```

5. Stop the slave on both sites.

```
# mysql -P 3307 --protocol=tcp -h ipw_sql
```

```
mysql> stop slave;
```

```
mysql> reset slave all;
```





Reference List

IPWorks Library Document

- [1] *Trademark Information*
- [2] *Typographic Conventions*
- [3] *Glossary of Terms and Acronyms*
- [4] *IPWorks EDA CLI Interface*
- [5] *IPWorks Configuration Management*
- [6] *Configure DNS and ENUM*
- [7] *IPWorks ENUM Front End Function Overview*, 55/155 17-AVA 901 16 Uen
- [8] *IPWorks Deployment Guide*, 21/1553-AVA 901 33/2 Uen
- [9] *IPWorks Deployment Guide*, 21/1553-AVA 901 33/2 Uen
- [10] *IPWorks Network Connectivity Overview*
- [11] *IPWorks Initial Configuration*, 5/1553-AVA 901 33/3
- [12] *Function Specification Subscriber Activation*, 155 17-CRH 109 1438

PCAT and Other Ericsson Document

- [13] *Function Specification Network Element Redundancy Handler*, 6/155 17-CXP 902 0723