

DB2 LUW PURESCALE

CUSTOM DISASTER RECOVERY CONFIGURATION

Marcin Baster

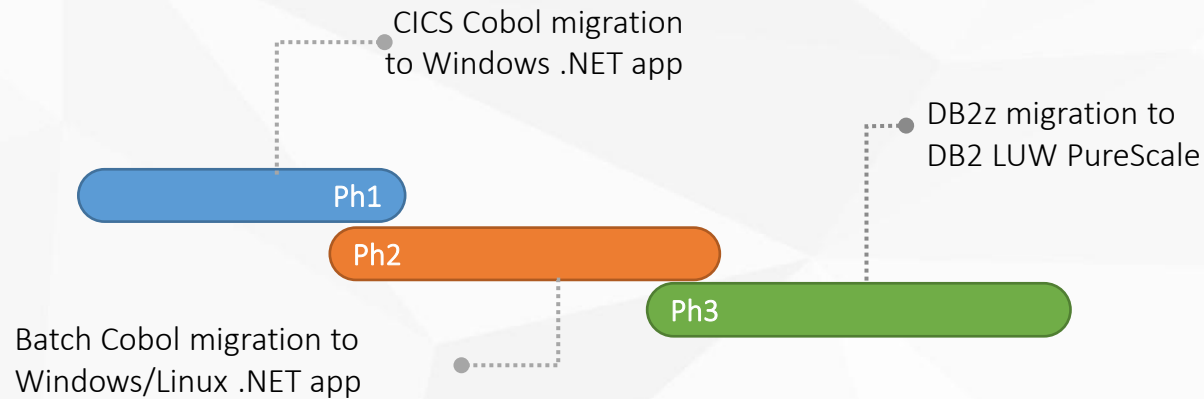
IBM

- » Db2 for zOS Migration Project
- » Db2 LUW PureScale Solution Design
 - » HA strategies for Db2 PureScale
 - » Db2 PureScale / GPFS Replication / PureScale HADR / PureScale GDPC
 - » PKO Solution Design
 - » Infrastructure/ XP8 Solution Diagram / Solution Design Diagrams
 - » Solution Implementation details
 - » overview / disks / filesystems / unmount-pri / syncFSconfig / mount-sec / unmount-sec / mount-pri / switch pri-sec / switch sec-pri
- » Db2 LUW PureScale Solution Testing
 - » Priority testing scope:
 - » stress / topology change / maintenance / backup and restore / recovery testing
 - » Complementary testing scope:
 - » pureScale HADR* / pureScale GDPC*

MIGRATION SCOPE



MAINFRAME MIGRATION - APPROACH



Main assumptions



- » Functional testing
- » Full Volume testing
- » Big bang Migration
 - » Two weeks migration:
 - » 95% of data – first week
 - » 5% of data – migration week
- » Rollback

Db2® pureScale® features four distinct deployment options, each addressing a specific high availability (HA) and/or disaster recovery (DR) requirement with an integrated Db2 solution with no additional dependencies on other software or storage solution.

Deployment models

The following Db2 pureScale deployment models are used for different user environments. Regardless if which model fits your needs, each of the Db2 pureScale deployment models share more than 95% of best practices in terms of system resource allocation (CPU and memory), network (switch, cluster interconnect), storage (disk types, Spectrum Scale configuration), Db2 instance, database, and client configurations.

HA in a single site

This deployment model is for customers who need continuous availability for only a single site. This specific cluster topology consists of multiple members, dual cluster caching facilities (CFs), multiple adapters and switches and a single storage accessible by all members and CFs. It provides constant availability and resiliency against planned and unplanned outages, unlimited capacity and workload balancing.

HA in a single site with redundant storage

This deployment option is for customers looking for storage redundancy options in Db2 pureScale for a single site. This model is to deploy the Db2 pureScale with Spectrum Scale replication for a fully integrated solution that provides transient storage failover capability.

Geographically dispersed pureScale cluster (GDPC)

This deployment model is for users that require an active/active DR solution with zero down time in-site failure. Essentially, this model consists of a split Db2 pureScale cluster stretched over tens of kilometers. It can be deployed in different rooms or floors within the same building, between different buildings in the same city, or even different cities but within public transport range. This distance limitation within tens of kilometers is to ensure that the performance delay, due to the distance, is within an acceptable range for a reasonable workload.

Db2 pureScale with HADR

If active/active DR is not a requirement, then the Db2 pureScale with HADR is another deployment model. This is very similar to HADR support in non-pureScale systems. It is worthwhile to emphasize that Db2 pureScale with HADR is not limited to large, geographic distance use cases, it can also be deployed in single-site scenarios.

DB2 PURESCALE TOPOLOGY

Clients connect anywhere,... ... see single database

Clients connect into any member Automatic load balancing and client reroute may change underlying physical member to which client is connected

DB2 engine runs on several host computers

Co-operate with each other to provide coherent access to the database from any member

Integrated cluster services

Failure detection, recovery automation, cluster file system
In partnership with STG (GPFS,RSCT) and Tivoli (SA MP)
Note TSAMP is for Db2 11.5. In 12.1 we moved to Pacemaker

Low latency, high speed interconnect

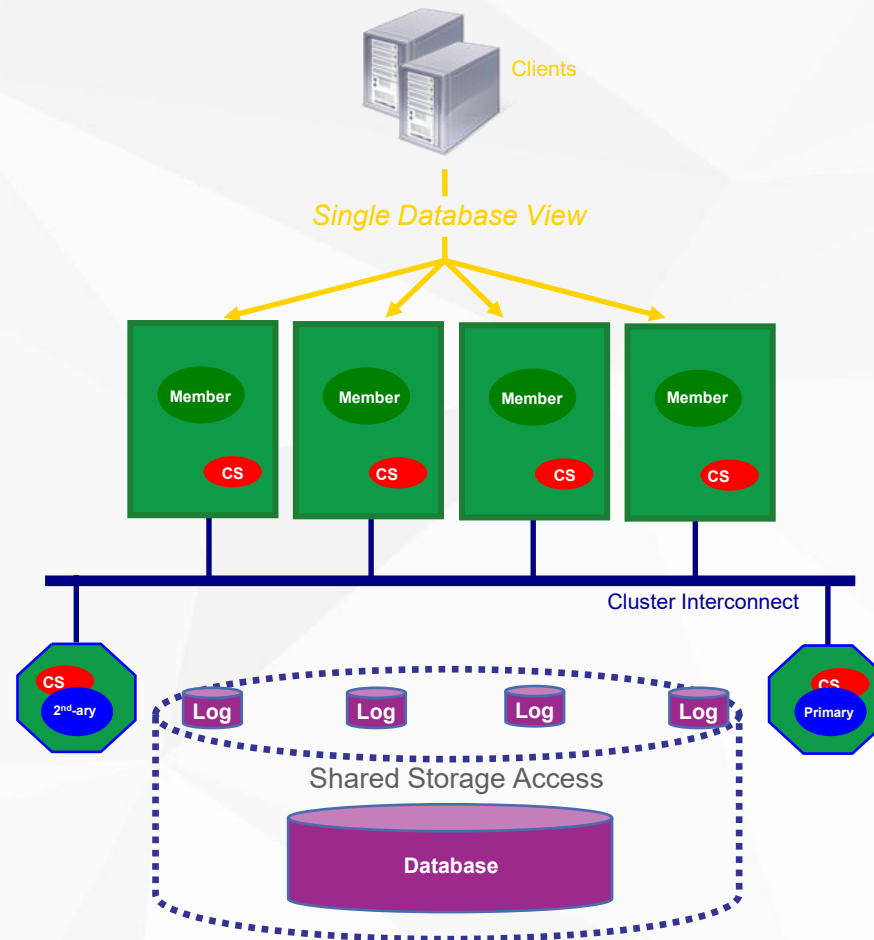
Special optimizations provide significant advantages on RDMA-capable interconnects (eg. InfiniBand)

Cluster Caching Facility (dev'd with STG)

Efficient global locking and buffer management
Synchronous duplexing to secondary ensures availability

Data sharing architecture

Shared access to database. Members write to their own logs
Logs accessible from another host (used during recovery)

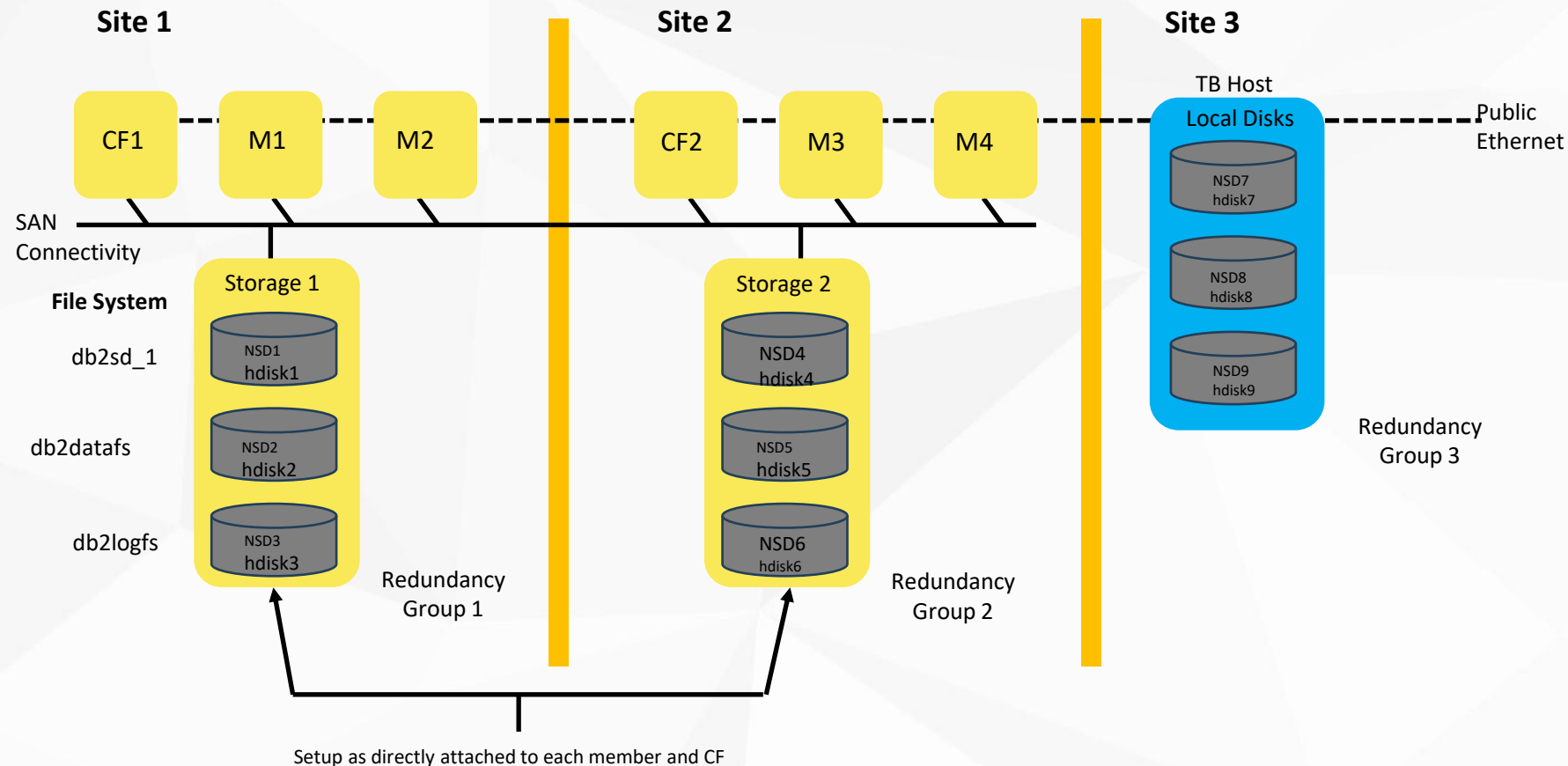


HA in a single site

This deployment model is for customers who need continuous availability for only a single site. This specific cluster topology consists of multiple members, dual cluster caching facilities (CFs), multiple adapters and switches and a single storage accessible by all members and CFs. It provides constant availability and resiliency against planned and unplanned outages, unlimited capacity and workload balancing.

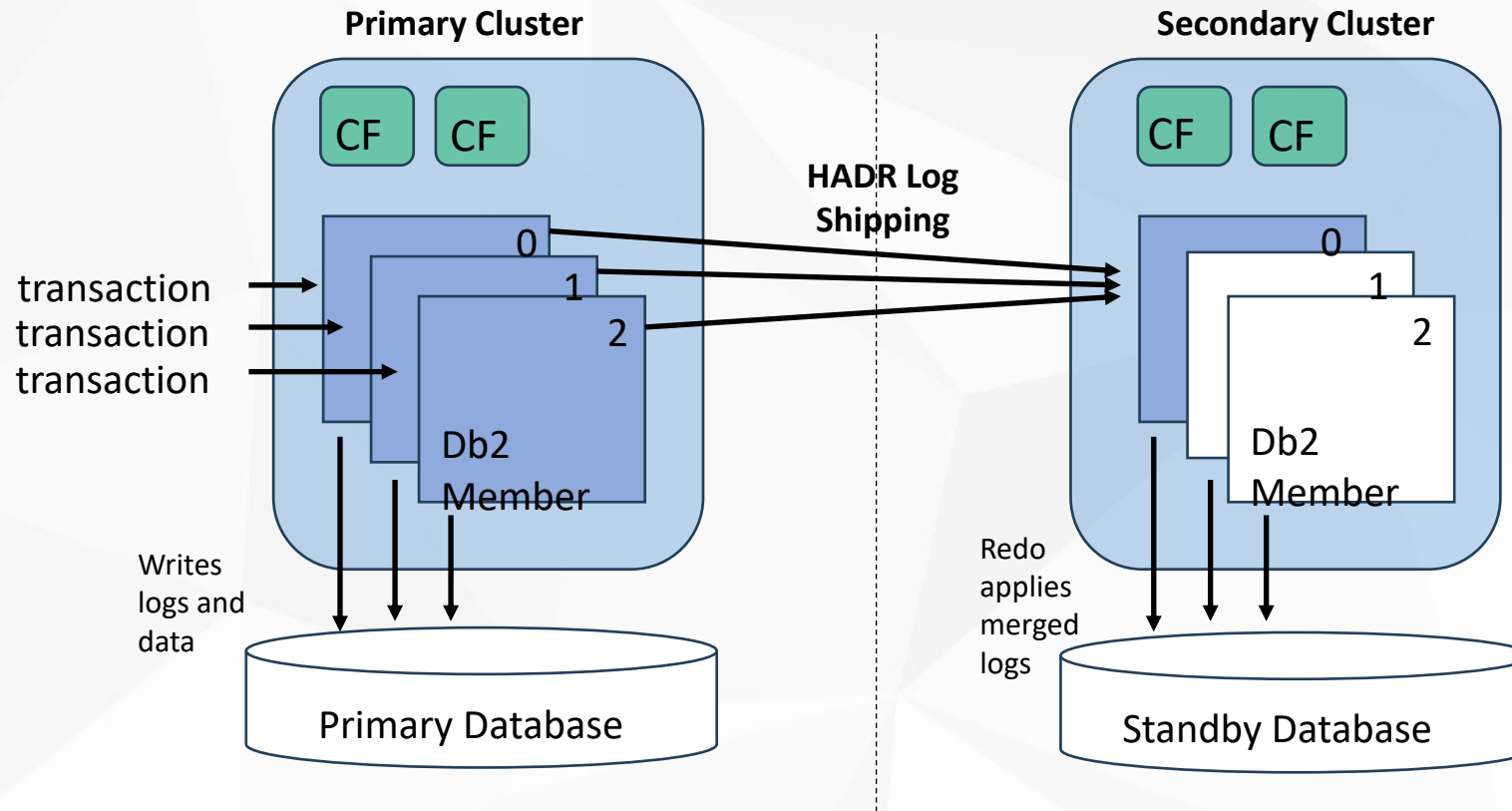
Additional Reference: <https://www.ibm.com/docs/en/db2/12.1.0?topic=editions-introduction-db2-purescale-environment>

DB2 PURESCALE GPFS REPLICATION



HA in a single site with redundant storage
This deployment option is for customers looking for storage redundancy options in Db2 pureScale for a single site. This model is to deploy the Db2 pureScale with Spectrum Scale replication for a fully integrated solution that provides transient storage failover capability.

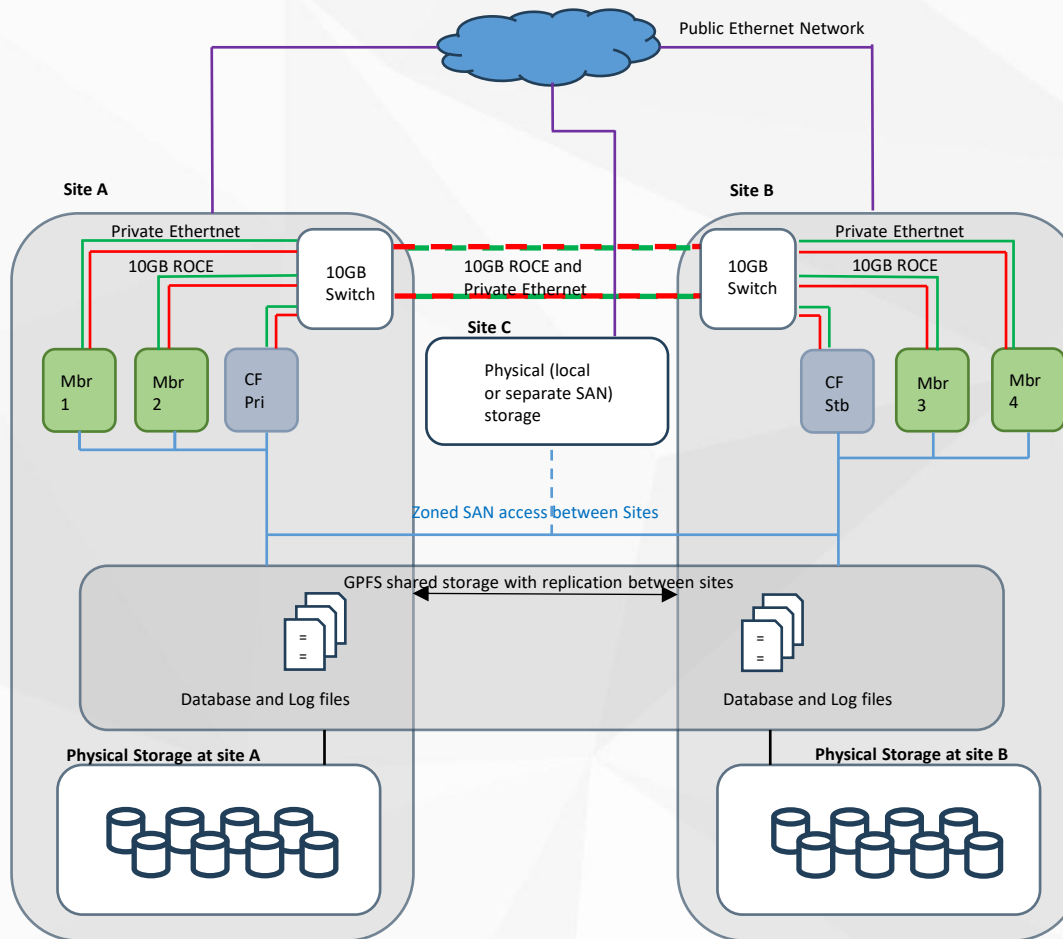
Additional Reference: <https://www.ibm.com/docs/en/db2/12.1.0?topic=environment-configuring-storage-scale-replication>



Db2 pureScale with HADR

If active/active DR is not a requirement, then the Db2 pureScale with HADR is another deployment model. This is very similar to HADR support in non-pureScale systems. It is worthwhile to emphasize that Db2 pureScale with HADR is not limited to large, geographic distance use cases, it can also be deployed in single-site scenarios.

Additional Reference: <https://www.ibm.com/docs/en/db2/12.1.0?topic=idpe-high-availability-disaster-recovery-hadr-in-db2-purescale-environments>

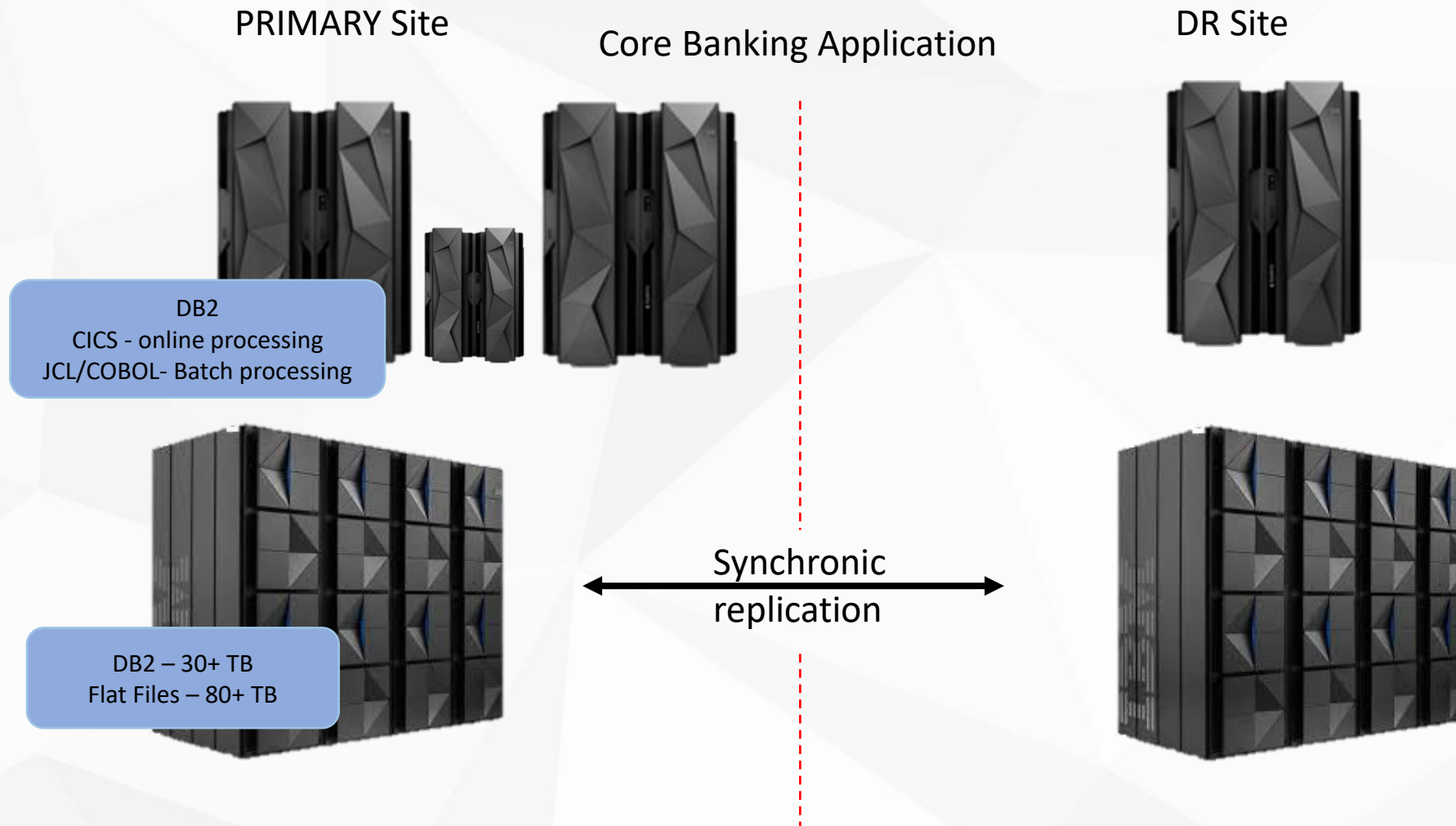


Geographically dispersed pureScale cluster (GDPC)

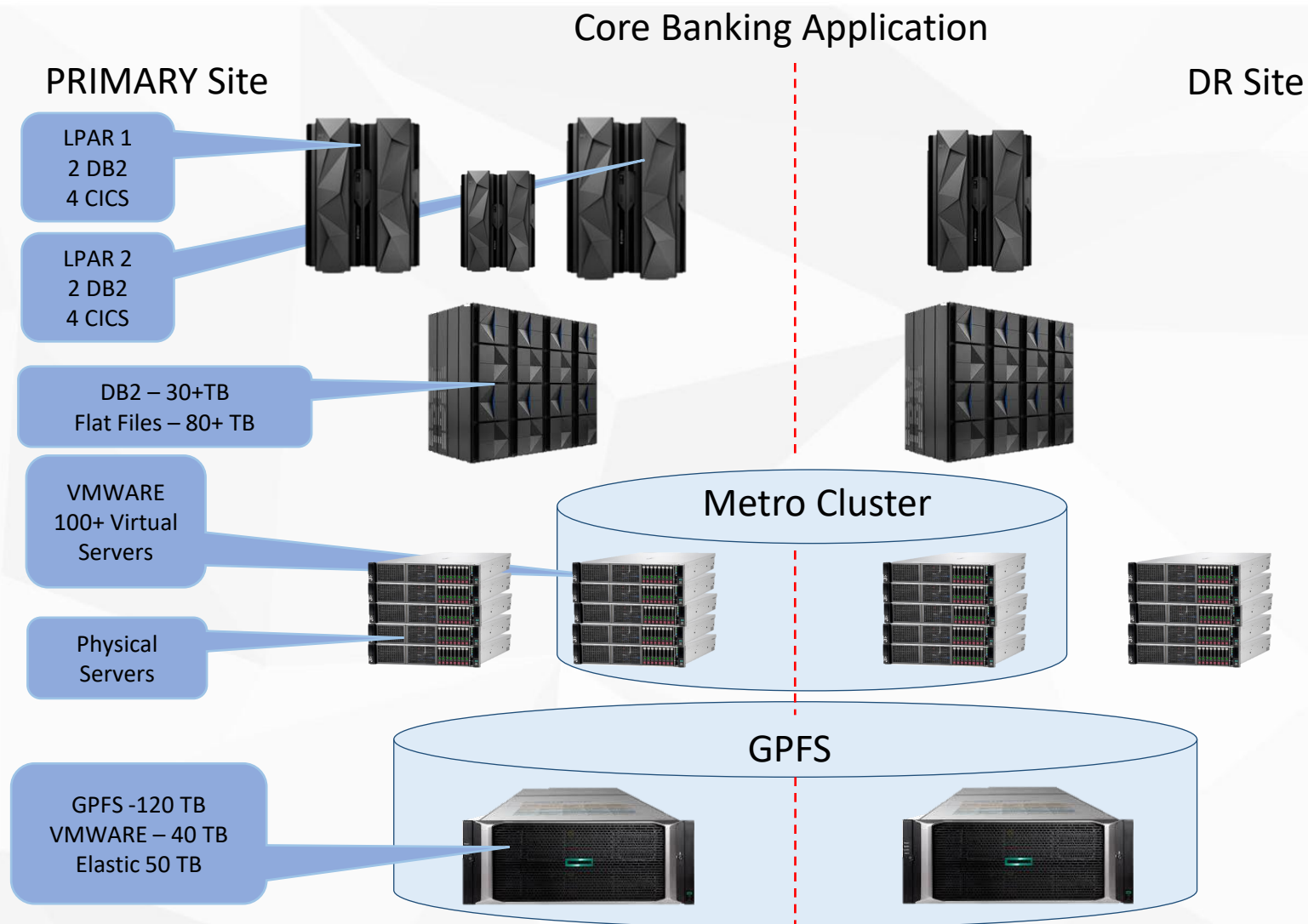
This deployment model is for users that require an active/active DR solution with zero down time in-site failure. Essentially, this model consists of a split Db2 pureScale cluster stretched over tens of kilometers. It can be deployed in different rooms or floors within the same building, between different buildings in the same city, or even different cities but within public transport range. This distance limitation within tens of kilometers is to ensure that the performance delay, due to the distance, is within an acceptable range for a reasonable workload.

Additional Reference: <https://www.ibm.com/docs/en/db2/12.1.0?topic=environment-geographically-dispersed-db2-purescale-cluster-gdpc>

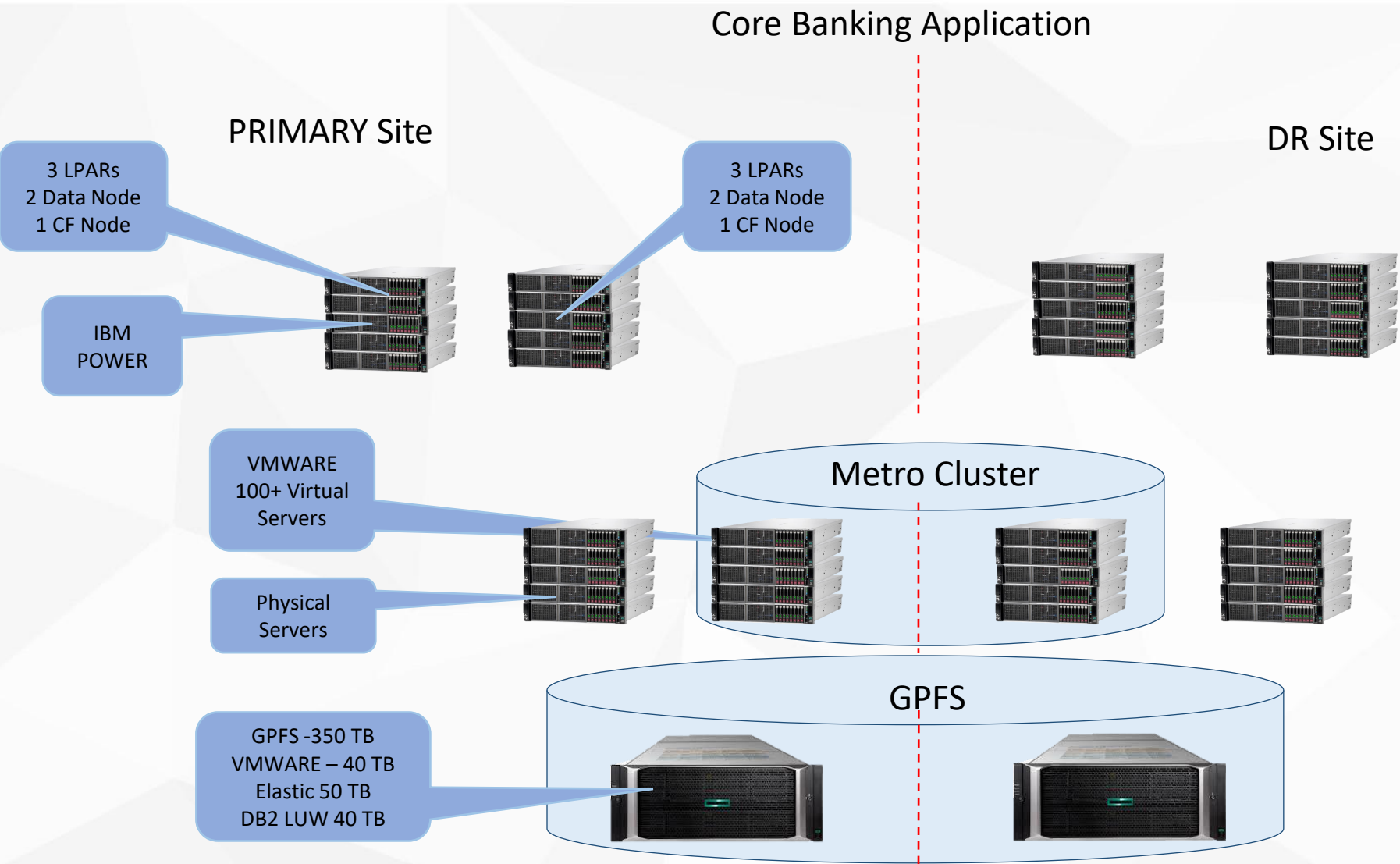
INFRASTRUCTURE – BEFORE PROJECT

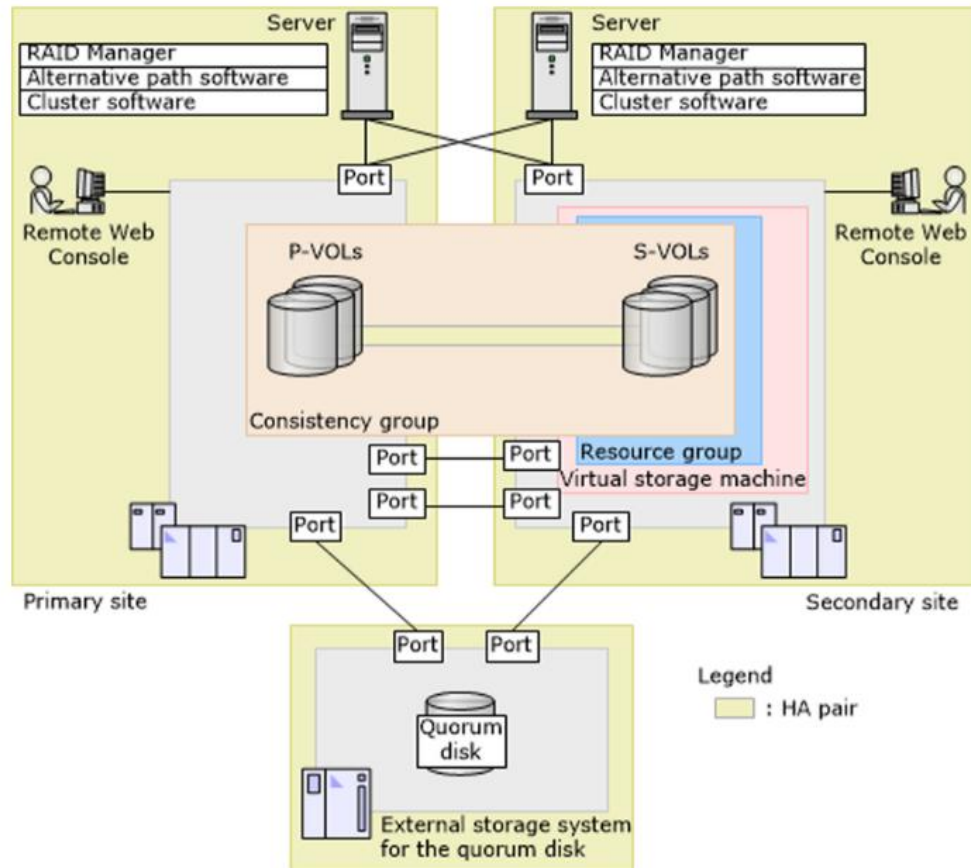


INFRASTRUCTURE – CURRENT



- » One Storage Subsystem for
 - » GPFS
 - » VMWARE
 - » Db2 LUW Purescale
 - » Elasticsearch
 - » Other
- » DB2 LUW & GPFS needs to be in one consistency Group
 - » Synchronic replication
 - » Storage snapshots
- » Performance
 - » Applications always run in one datacenter (no cross-site workload)
 - » No TCPIP replication

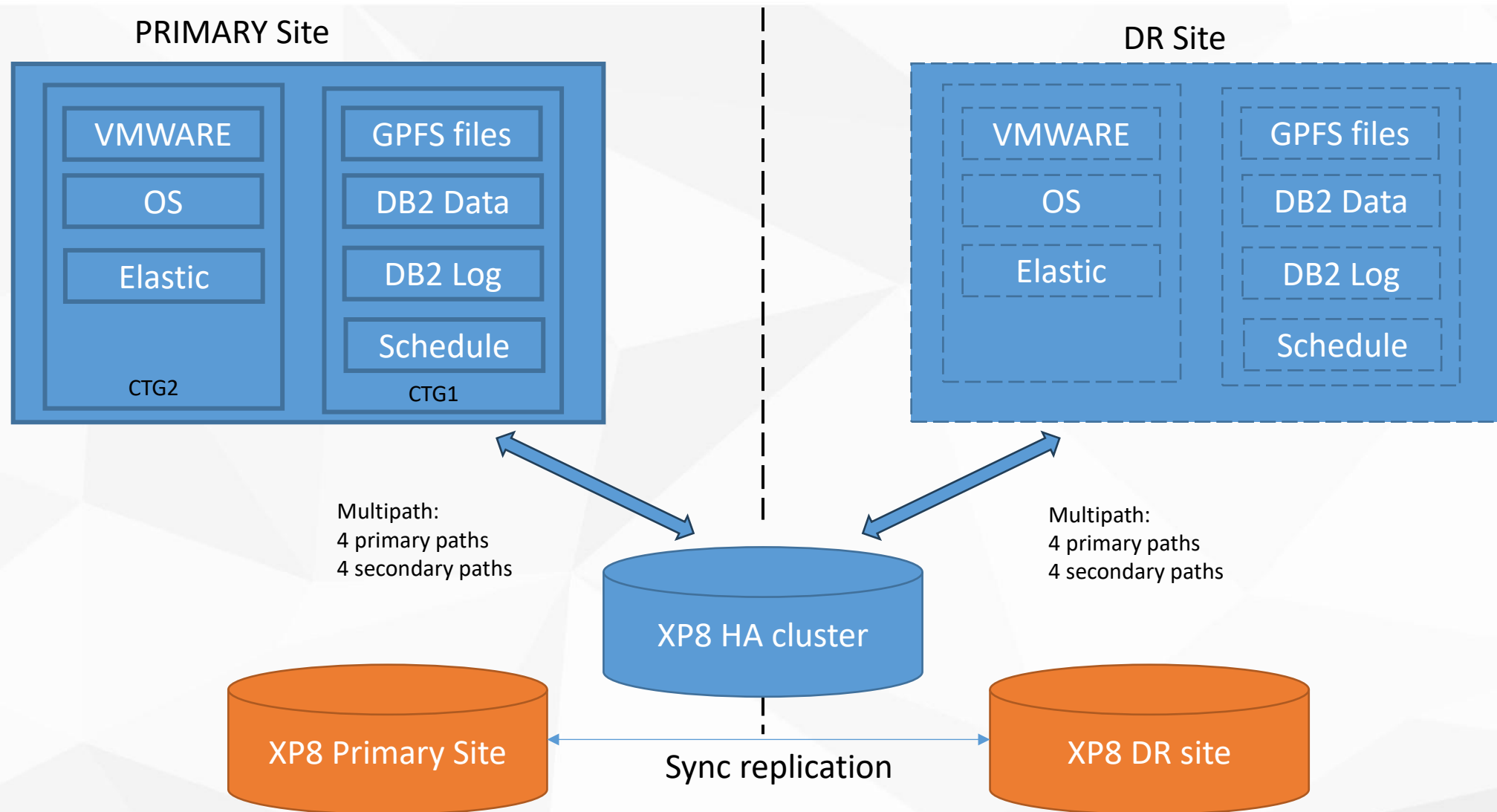


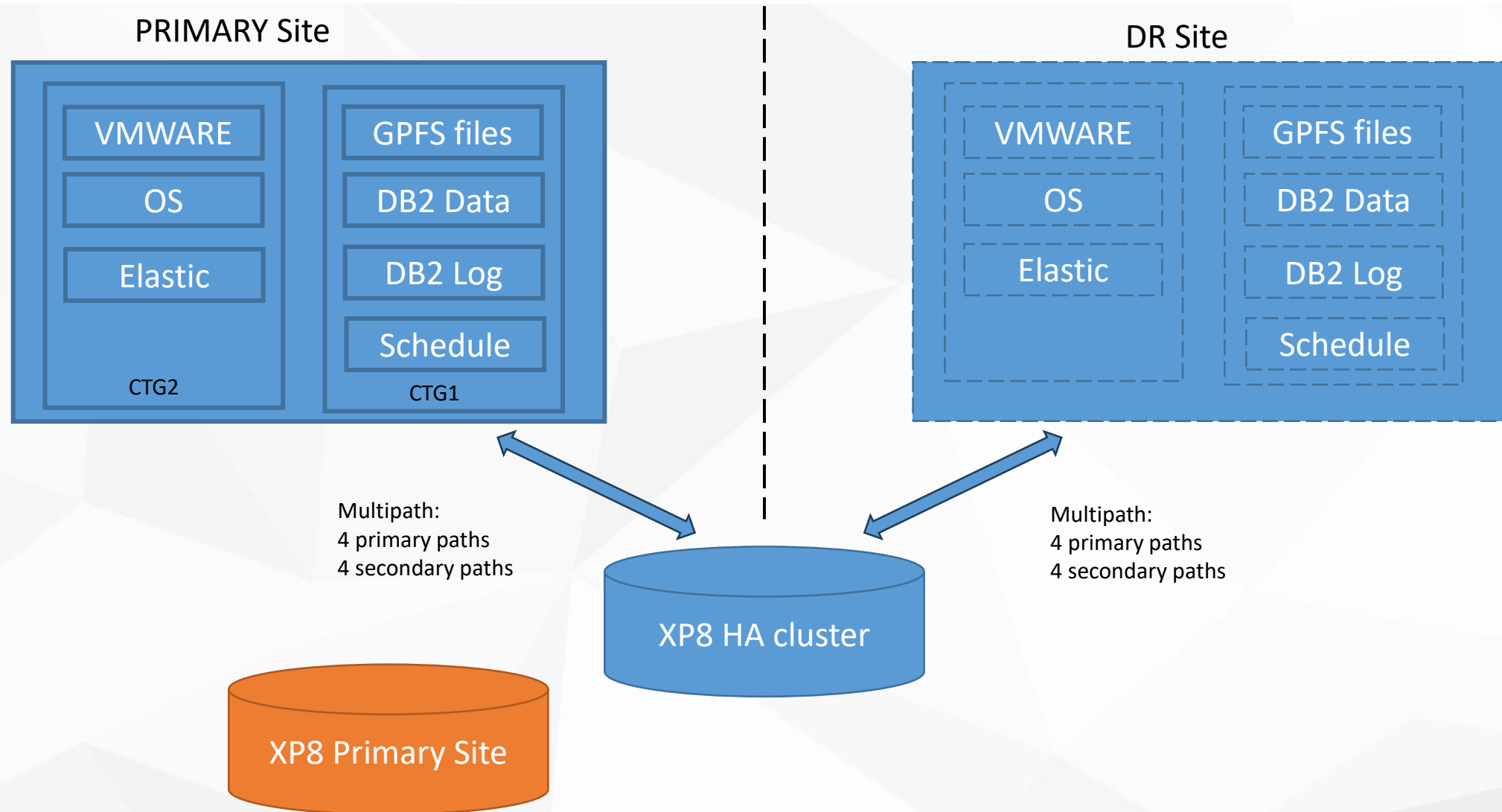


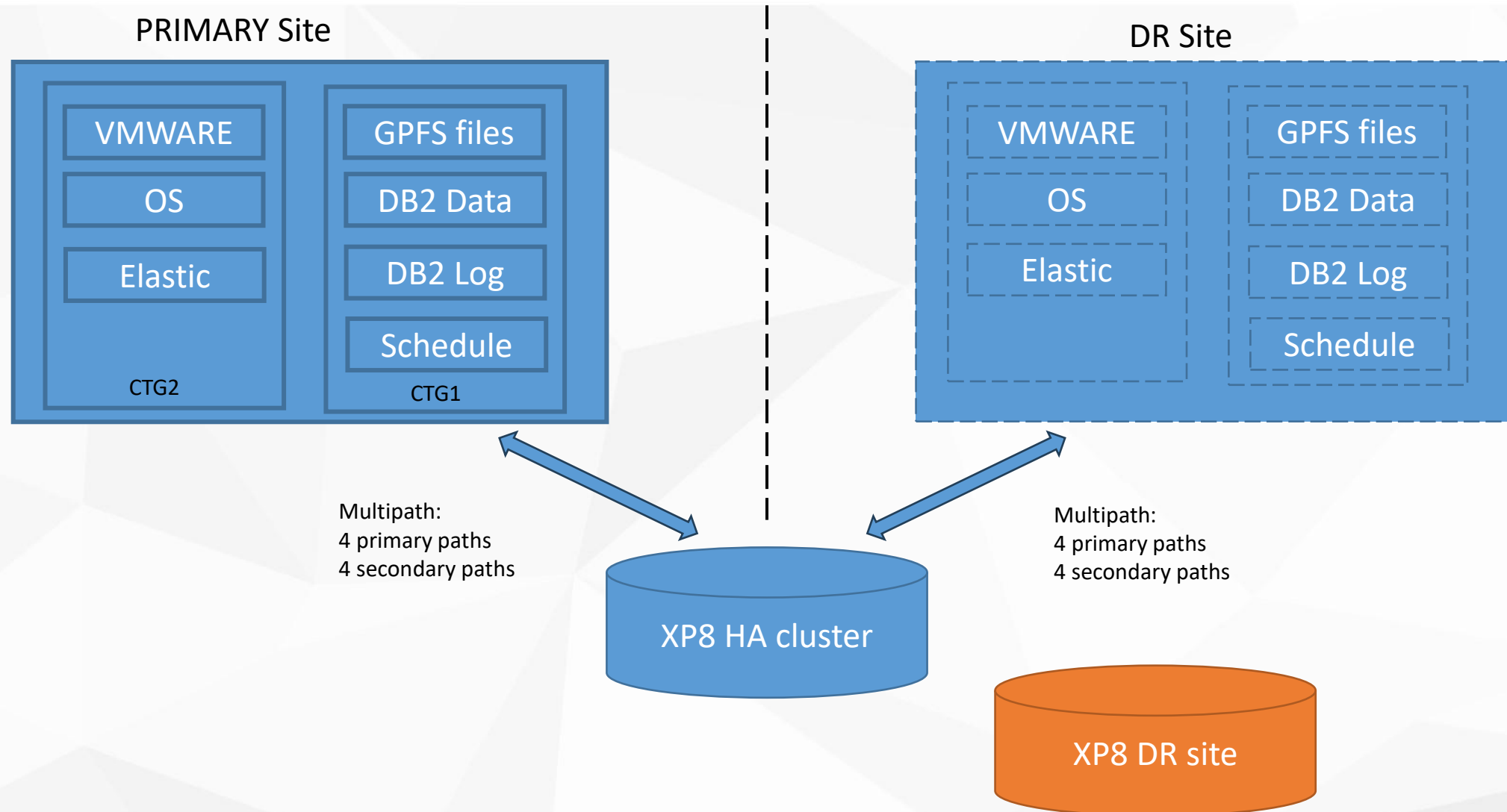
A typical High Availability system consists of:

- storage systems
- paired volumes
- a consistency group
- quorum disk
- a virtual storage machine
- paths
- ports
- alternate path software
- cluster software

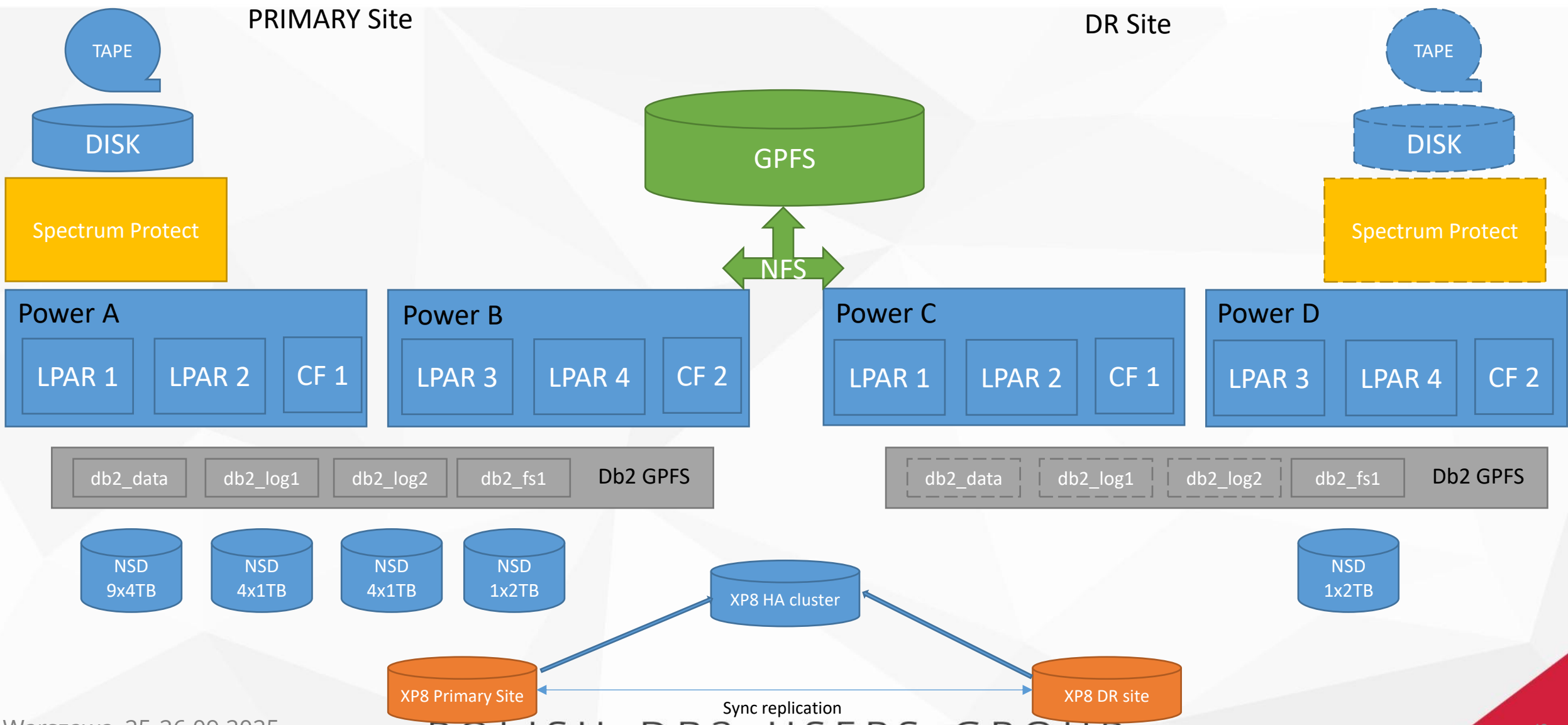
Additional Reference: https://support.hpe.com/hpsc/public/docDisplay?docId=a00119380en_us&docLocale=en_US



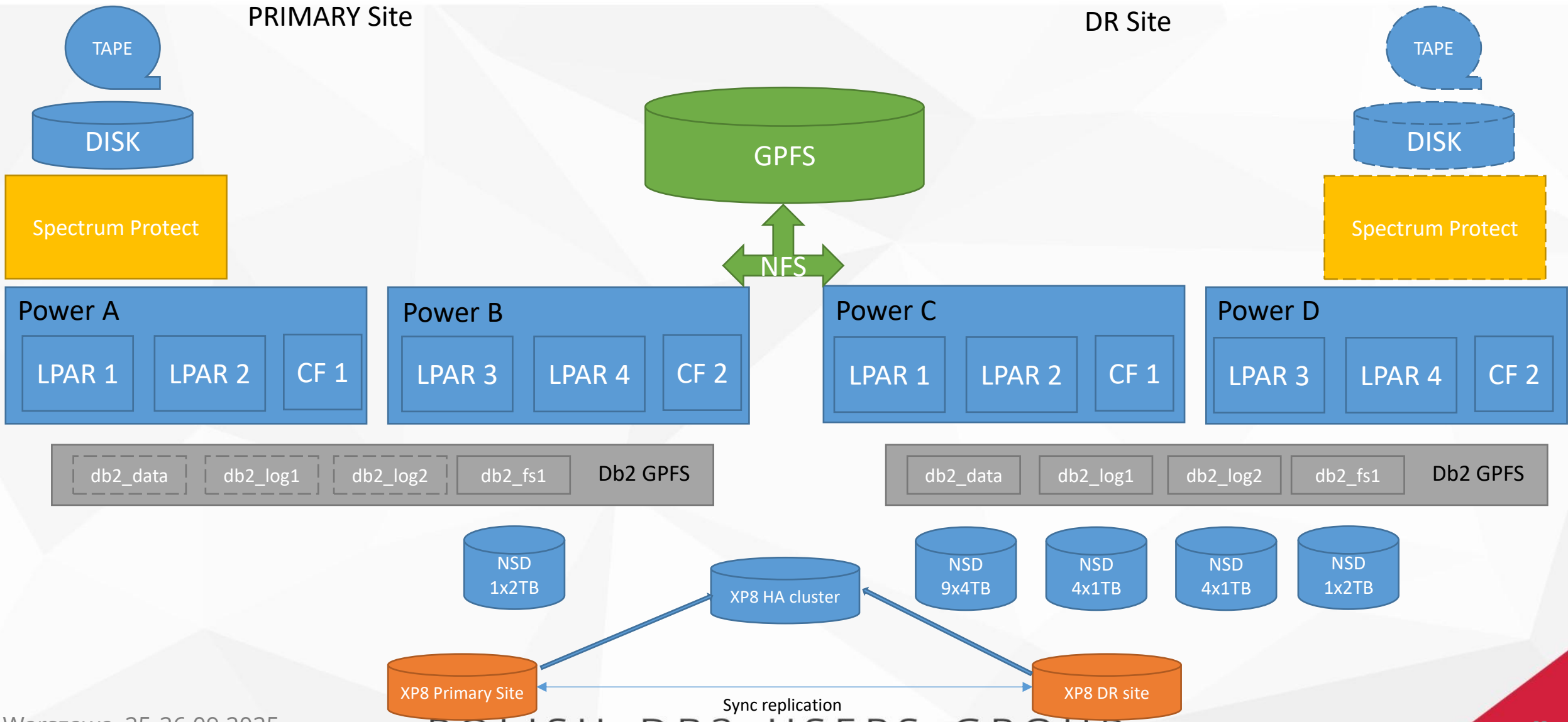




PKO SOLUTION DESIGN - PRIMARY



PKO SOLUTION DESIGN - DR



INSTALLATION #1 - OVERVIEW

Prerequisites + Db2 binaries installation is straightforward.

We do this accordingly to KC:

<https://www.ibm.com/docs/en/db2/12.1.0?topic=environment-installing-db2-purescale-linux>

primary cluster:

```
-----  
passwordless ssh from client machine  
Storage iSCSI (pri-instance/shared-data/shared-logs)  
OS prerequisites + iSCSI multipath  
Db2 12.1 installation + passwordless ssh pri-cluster  
selfmanaged GPFS pri-cluster  
pri-cluster instance + sharedDB
```

secondary cluster:

```
-----  
passwordless ssh from client machine  
Storage iSCSI (sec-instance/shared-data/shared-logs)  
OS prerequisites + iSCSI multipath  
Db2 12.1 installation + passwordless ssh sec-cluster  
selfmanaged GPFS sec-cluster  
sec-cluster instance
```

Takeover setup procedure:

```
-----  
pri-cluster uncatalog  
pri-cluster enter maintenance and unmount  
passwordless ssh pri-cluster / sec-cluster  
mmfsctl syncFSconfig  
sec-cluster mount  
sec-cluster cataloging  
sec-cluster db2cm -add -dbMount (data / logs)  
sec-cluster connect and data manipulation
```

Failback setup procedure:

```
-----  
sec-cluster uncatalog  
sec-cluster enter maintenance and unmounmt  
pri-cluster mount  
pri-cluster exit maintenance  
pri-cluster cataloging  
pri-cluster connect and data manipulation
```

Takeover-Failback testing:

```
-----  
pri-cluster uncatalog and stop  
pri-cluster enter maintenance and unmounmt  
sec-cluster mount and exit maintenance  
sec-cluster cataloging  
sec-cluster connect and data manipulation  
sec-cluster uncatalog and stop  
sec-cluster enter maintenance and unmounmt  
pri-cluster exit maintenance  
pri-cluster cataloging  
pri-cluster connect and data manipulation
```

INSTALLATION #2 - DISKS

For the storage we use 4 disks (Instance1 , Instance2 , SharedDB , Logs). Note that on each cluster we have unique Instance1 and Instance2, that are mounted just once. Other disks (SharedDB , Logs) are shared across clusters (ie floating ones). We can instruct db2icrt to use user-managed GPFS or Db2-managed GPFS.

PRI CLUSTER

Disk name	NSD volume ID	Device
nsddata	0A0B53DE67F19201	/dev/dm-3
nsdinstance	0A0B53DE67F191FE	/dev/dm-2
nsdlogs	0A0B53DE67F19204	/dev/dm-4

***User managed:** /opt/ibm/db2/V12.1/instance/db2icrt -d -m mb-pc-m11 -mnet mb-pc-m11 -cf mb-pc-m11 -cfnet mb-pc-m11 -m mb-pc-m21 -mnet mb-pc-m21 -cf mb-pc-m21 -cfnet mb-pc-m21 -instance_shared_dir /gpfs/pcinstance -u db2fenc1 db2inst1

***Db2 managed:** /opt/ibm/db2/V12.1/instance/db2icrt -d -m mb-pc-m11 -mnet mb-pc-m11 -cf mb-pc-m11 -cfnet mb-pc-m11 -m mb-pc-m21 -mnet mb-pc-m21 -cf mb-pc-m21 -cfnet mb-pc-m21 -instance_shared_dev /dev/dm-2 -instance_shared_mount /db2pricluster -u db2fenc1 db2inst1

SEC CLUSTER

Disk name	NSD volume ID	Device
nsdinstance	0A0B724667937AF9	/dev/dm-2

***User managed:** /opt/ibm/db2/V12.1/instance/db2icrt -d -m mb-sc-m11 -mnet mb-sc-m11 -cf mb-sc-m11 -cfnet mb-sc-m11 -m mb-sc-m21 -mnet mb-sc-m21 -cf mb-sc-m21 -cfnet mb-sc-m21 -instance_shared_dir /gpfs/scinstance -u db2fenc1 db2inst1

***Db2 managed:** /opt/ibm/db2/V12.1/instance/db2icrt -d -m mb-sc-m11 -mnet mb-sc-m11 -cf mb-sc-m11 -cfnet mb-sc-m11 -m mb-sc-m21 -mnet mb-sc-m21 -cf mb-sc-m21 -cfnet mb-sc-m21 -instance_shared_dev /dev/dm-2 -instance_shared_mount /db2seccluster -u db2fenc1 db2inst1

INSTALLATION #3 - FILESYSTEMS

Next, we add new directories to the cluster. We achieve that by creating database in dedicated path, and defining storage group for logs. Note that in user managed version we have mount points prepared upfront, while in Db2 managed we will be using `db2cm` command to create filesystems.

*User managed:

```
db2 "create db PSSHARED on /gpfs/psdata DBPATH ON /gpfs/psdata"
db2 "create stogroup sgrplog on '/gpfs/pslogs'"
db2 "update db cfg for PSSHARED using NEWLOGPATH /gpfs/pslogs"
```

```
db2cm -status | grep pcinstance
db2cm -status | grep psdata
db2cm -status | grep pslogs
```

```
* Clone Set: db2_mount_gpfs_pcinstance-clone [db2_mount_gpfs_pcinstance]:
* Clone Set: db2_mount_gpfs_psdata-clone [db2_mount_gpfs_psdata]:
* Clone Set: db2_mount_gpfs_pslogs-clone [db2_mount_gpfs_pslogs]:
```

*Db2 managed:

```
/opt/ibm/db2/V12.1/bin/db2cm -cfs -create -filesystem db2shareddb -disk /dev/dm-3 -mount /db2shareddb
/opt/ibm/db2/V12.1/bin/db2cm -cfs -create -filesystem db2log -disk /dev/dm-4 -mount /db2log
```

```
db2 "create db PSSHARED on /db2shareddb DBPATH ON /db2shareddb"
db2 "create stogroup sgrplog on '/db2log'"
db2 "update db cfg for PSSHARED using NEWLOGPATH /db2log"
```

```
db2cm -status | grep db2pricluster
db2cm -status | grep db2shareddb
db2cm -status | grep db2log
```

```
* Clone Set: db2_mount_db2pricluster-clone [db2_mount_db2pricluster]:
* Clone Set: db2_mount_db2shareddb-clone [db2_mount_db2shareddb]:
* Clone Set: db2_mount_db2log-clone [db2_mount_db2log]:
```

INSTALLATION #4 – UNMOUNT-PRI

So now we have 3 paths in the cluster (instance + database + logs). Lets prepare move to SEC cluster.
We do here first move procedure, where db2inst1 commands are interleaved with root commands.
We want to do some checking (more for safety, as if we skip those safety checks, procedure will work).

***Db2 managed:**

```
db2 terminate
db2 uncatalog db PSSHARED
db2stop
```

```
lsof +D /db2shareddb/
lsof +D /db2log/
```

```
ssh mb-pc-m11 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -enter -maintenance -mount /db2shareddb"
ssh mb-pc-m21 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -enter -maintenance -mount /db2shareddb"
ssh mb-pc-m11 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -enter -maintenance -mount /db2log"
ssh mb-pc-m21 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -enter -maintenance -mount /db2log"
```

```
Host-level enter maintenance on mount '/db2shareddb' has completed successfully.
Host-level enter maintenance on mount '/db2shareddb' has completed successfully.
Host-level enter maintenance on mount '/db2log' has completed successfully.
Host-level enter maintenance on mount '/db2log' has completed successfully.
```

```
ssh mb-pc-m11 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -status | grep db2priccluster"
ssh mb-pc-m11 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -status | grep db2shareddb"
ssh mb-pc-m11 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -status | grep db2log"
```

*** Clone Set: db2_mount_db2priccluster-clone [db2_mount_db2priccluster]:**

```
/usr/lpp/mmfs/bin/mmlsmount all
```

```
File system db2fs1 is mounted on 2 nodes.
File system db2log is not mounted.
File system db2shareddb is not mounted.
```

INSTALLATION #5 - SYNCFSCONFIG

Now Sync SEC cluster - for this we need to transfer filesystems to SEC cluster.
Once we have passwordless ssh + db2locshh configured across 2 clusters, we can sync GPFS.
We create remote cluster (SEC cluster) definition file, and next sync from PRI cluster.

```
remote_cluster_def
-----
mb-sc-m11.fyre.ibm.com
mb-sc-m21.fyre.ibm.com
-----
```

***User managed:**

```
/usr/lpp/mmfs/bin/mmfsctl psdata syncFSconfig -n remote_cluster_def
/usr/lpp/mmfs/bin/mmfsctl pslogs syncFSconfig -n remote_cluster_def
```

***Db2 managed:**

```
mb-pc-m1: /usr/lpp/mmfs/bin/mmchcluster --use-sudo-wrapper
mb-sc-m1: /usr/lpp/mmfs/bin/mmchcluster --use-sudo-wrapper

/usr/lpp/mmfs/bin/mmfsctl db2shared db2 syncFSconfig -n remote_cluster_def
/usr/lpp/mmfs/bin/mmfsctl db2log      syncFSconfig -n remote_cluster_def
```

INSTALLATION #6 – MOUNT-SEC

Once above is done, we can mount filesystems on SEC cluster. We will do the mount with mmmount command. Note we do this with root on mb-sc-m11 server from SEC cluster. We will also use mmlsmount to observe the changes. Next we use db2cm -add -dbMount to put that in Pacemaker cluster, so that resources are managed.

***Db2 managed:**

```
mmmount db2shareddb -a
mmmount db2log -a
mmlsmount all
```

File system db2fs1 is mounted on 2 nodes.
File system db2log is mounted on 2 nodes.
File system db2shareddb is mounted on 2 nodes.

```
db2start
db2 catalog db PSSHARED on /db2shareddb

/opt/ibm/db2/V12.1/bin/db2cm -status | grep db2seccluster
/opt/ibm/db2/V12.1/bin/db2cm -status | grep db2shareddb
/opt/ibm/db2/V12.1/bin/db2cm -status | grep db2log
```

*** Clone Set: db2_mount_db2seccluster-clone [db2_mount_db2seccluster]:**

```
/opt/ibm/db2/V12.1/bin/db2cm -add -dbMount PSSHARED

/opt/ibm/db2/V12.1/bin/db2cm -status | grep db2seccluster
/opt/ibm/db2/V12.1/bin/db2cm -status | grep db2shareddb
/opt/ibm/db2/V12.1/bin/db2cm -status | grep db2log
```

*** Clone Set: db2_mount_db2seccluster-clone [db2_mount_db2seccluster]:**

*** Clone Set: db2_mount_db2shareddb-clone [db2_mount_db2shareddb]:**

*** Clone Set: db2_mount_db2log-clone [db2_mount_db2log]:**

INSTALLATION #7 – UNMOUNT-SEC

So above is working fine! Now the failback. Note that so far we created SEC cluster by syncing filesystem. This required many steps to sync FS, create DB mounts that resulted in creation of Pacemaker resources. The failback is similar to failover from PRI cluster.

***Db2 managed:**

```
db2 terminate
db2 uncatalog db PSSHARED
db2stop
```

```
lsof +D /db2shareddb/
lsof +D /db2log/
```

```
ssh mb-sc-m11 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -enter -maintenance -mount /db2shareddb"
ssh mb-sc-m21 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -enter -maintenance -mount /db2shareddb"
ssh mb-sc-m11 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -enter -maintenance -mount /db2log"
ssh mb-sc-m21 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -enter -maintenance -mount /db2log"
```

Host-level enter maintenance on mount '/db2shareddb' has completed successfully.

Host-level enter maintenance on mount '/db2shareddb' has completed successfully.

Host-level enter maintenance on mount '/db2log' has completed successfully.

Host-level enter maintenance on mount '/db2log' has completed successfully.

```
ssh mb-sc-m11 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -status | grep db2seccluster"
ssh mb-sc-m11 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -status | grep db2shareddb"
ssh mb-sc-m11 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -status | grep db2log"
```

*** Clone Set: db2_mount_db2priccluster-clone [db2_mount_db2priccluster]:**

```
/usr/lpp/mmfs/bin/mmlsmount all
```

File system db2fs1 is mounted on 2 nodes.

File system db2log is not mounted.

File system db2shareddb is not mounted.

INSTALLATION #8 – MOUNT-PRI

At this point we can switch roles - we can use following procedure to takeover DB on PRI cluster:

***Db2 managed:**

```
mmm mount db2shareddb -a
mmm mount db2log -a
mmlsmount all
```

File system db2fs1 is mounted on 2 nodes.
File system db2log is mounted on 2 nodes.
File system db2shareddb is mounted on 2 nodes.

```
ssh mb-pc-m11 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -exit -maintenance -mount /db2shareddb"
ssh mb-pc-m21 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -exit -maintenance -mount /db2shareddb"
ssh mb-pc-m11 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -exit -maintenance -mount /db2log"
ssh mb-pc-m21 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -exit -maintenance -mount /db2log"
```

Host-level exit maintenance on mount '/db2shareddb' has completed successfully.
Host-level exit maintenance on mount '/db2shareddb' has completed successfully.
Host-level exit maintenance on mount '/db2log' has completed successfully.
Host-level exit maintenance on mount '/db2log' has completed successfully.

```
db2start
db2 catalog db PSSHARED on /db2shareddb
```

```
/opt/ibm/db2/V12.1/bin/db2cm -status | grep db2pricluster
/opt/ibm/db2/V12.1/bin/db2cm -status | grep db2shareddb
/opt/ibm/db2/V12.1/bin/db2cm -status | grep db2log
```

*** Clone Set: db2 mount db2pricluster-clone [db2 mount db2pricluster]:**
*** Clone Set: db2 mount db2shareddb-clone [db2 mount db2shareddb]:**
*** Clone Set: db2 mount db2log-clone [db2 mount db2log]:**

INSTALLATION #9 – SWITCH PRI-SEC

After all above we have setup ready. Now we can do role switch with following:

PRI CLUSTER -> mb-pc-m11

```
db2 terminate
db2 uncatalog db PSSHARED
db2stop
```

```
ssh mb-pc-m11 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -enter -maintenance -mount /db2shareddb"
ssh mb-pc-m21 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -enter -maintenance -mount /db2shareddb"
ssh mb-pc-m11 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -enter -maintenance -mount /db2log"
ssh mb-pc-m21 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -enter -maintenance -mount /db2log"
```

```
/usr/lpp/mmfs/bin/mmlsmount all
```

SEC CLUSTER -> mb-sc-m11

```
mmmount db2shareddb -a
mmmount db2log -a
mmlsmount all
```

```
ssh mb-sc-m11 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -exit -maintenance -mount /db2shareddb"
ssh mb-sc-m21 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -exit -maintenance -mount /db2shareddb"
ssh mb-sc-m11 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -exit -maintenance -mount /db2log"
ssh mb-sc-m21 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -exit -maintenance -mount /db2log"
```

```
db2start
db2 catalog db PSSHARED on /db2shareddb
```

INSTALLATION #10 – SWITCH SEC-PRI

And of course we can takeover back (in reverse order):

SEC CLUSTER -> mb-sc-m11

```
db2 terminate
db2 uncatalog db PSSHARED
db2stop
```

```
ssh mb-sc-m11 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -enter -maintenance -mount /db2shareddb"
ssh mb-sc-m21 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -enter -maintenance -mount /db2shareddb"
ssh mb-sc-m11 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -enter -maintenance -mount /db2log"
ssh mb-sc-m21 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -enter -maintenance -mount
```

```
/db2log"/usr/lpp/mmfs/bin/mmlsmount all
```

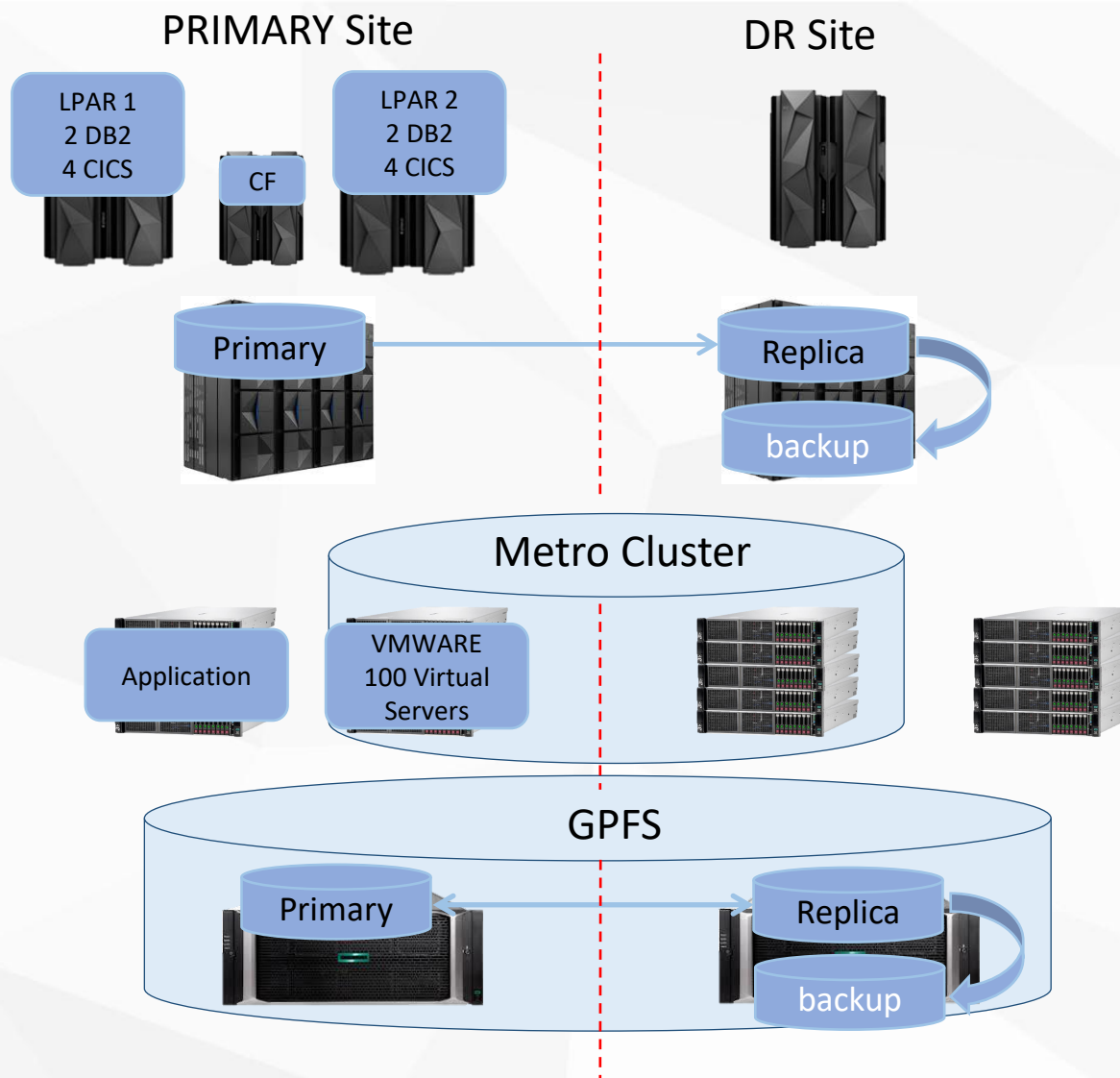
PRI CLUSTER -> mb-pc-m11

```
mmmount db2shareddb -a
mmmount db2log -a
mmlsmount all
```

```
ssh mb-pc-m11 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -exit -maintenance -mount /db2shareddb"
ssh mb-pc-m21 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -exit -maintenance -mount /db2shareddb"
ssh mb-pc-m11 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -exit -maintenance -mount /db2log"
ssh mb-pc-m21 "export DB2INSTANCE=db2inst1;/opt/ibm/db2/V12.1/bin/db2cm -exit -maintenance -mount /db2log"
```

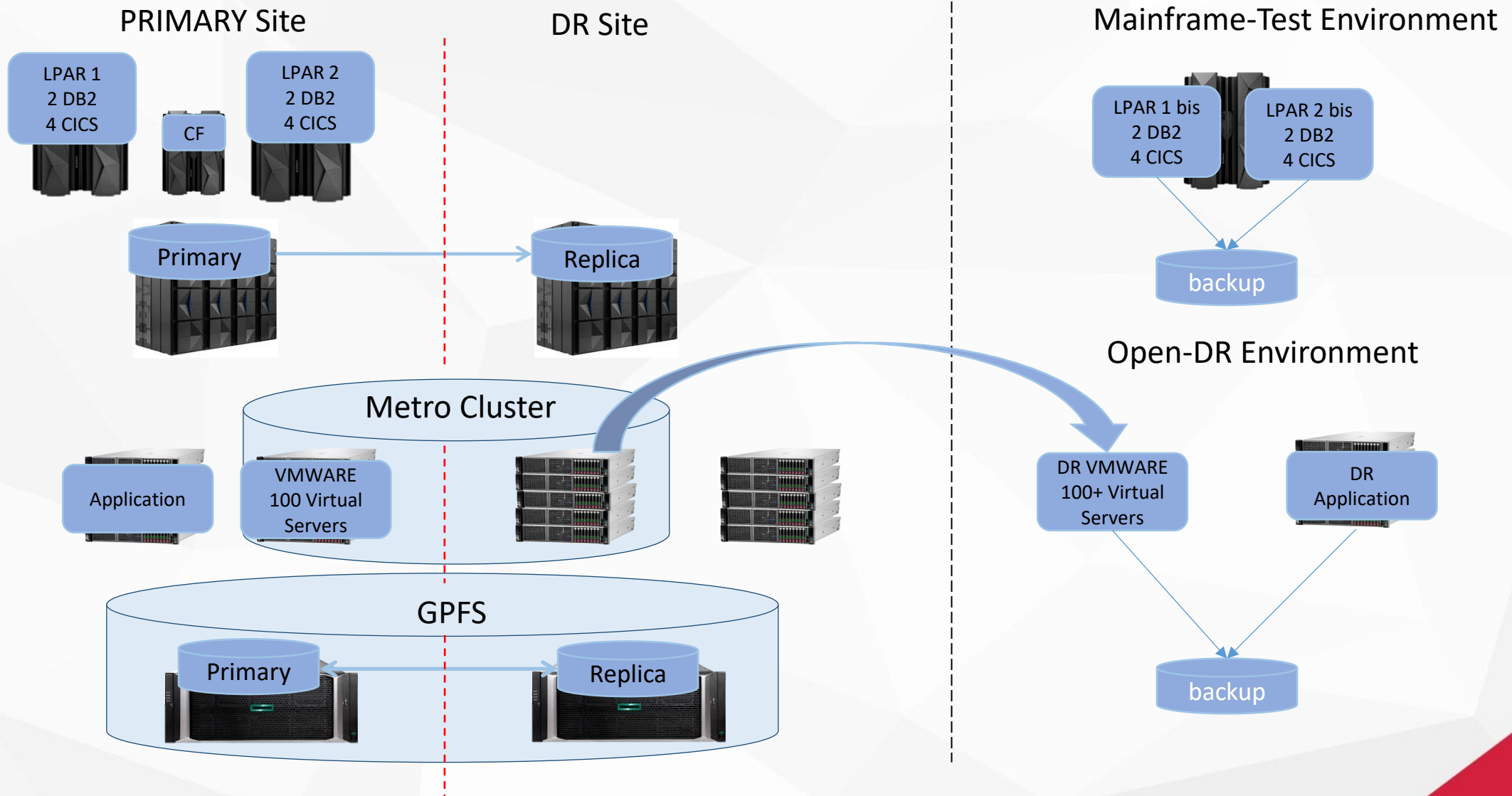
```
db2start
db2 catalog db PSSHARED on /db2shareddb
```

DB2 PURESCALE SOLUTION - TESTING

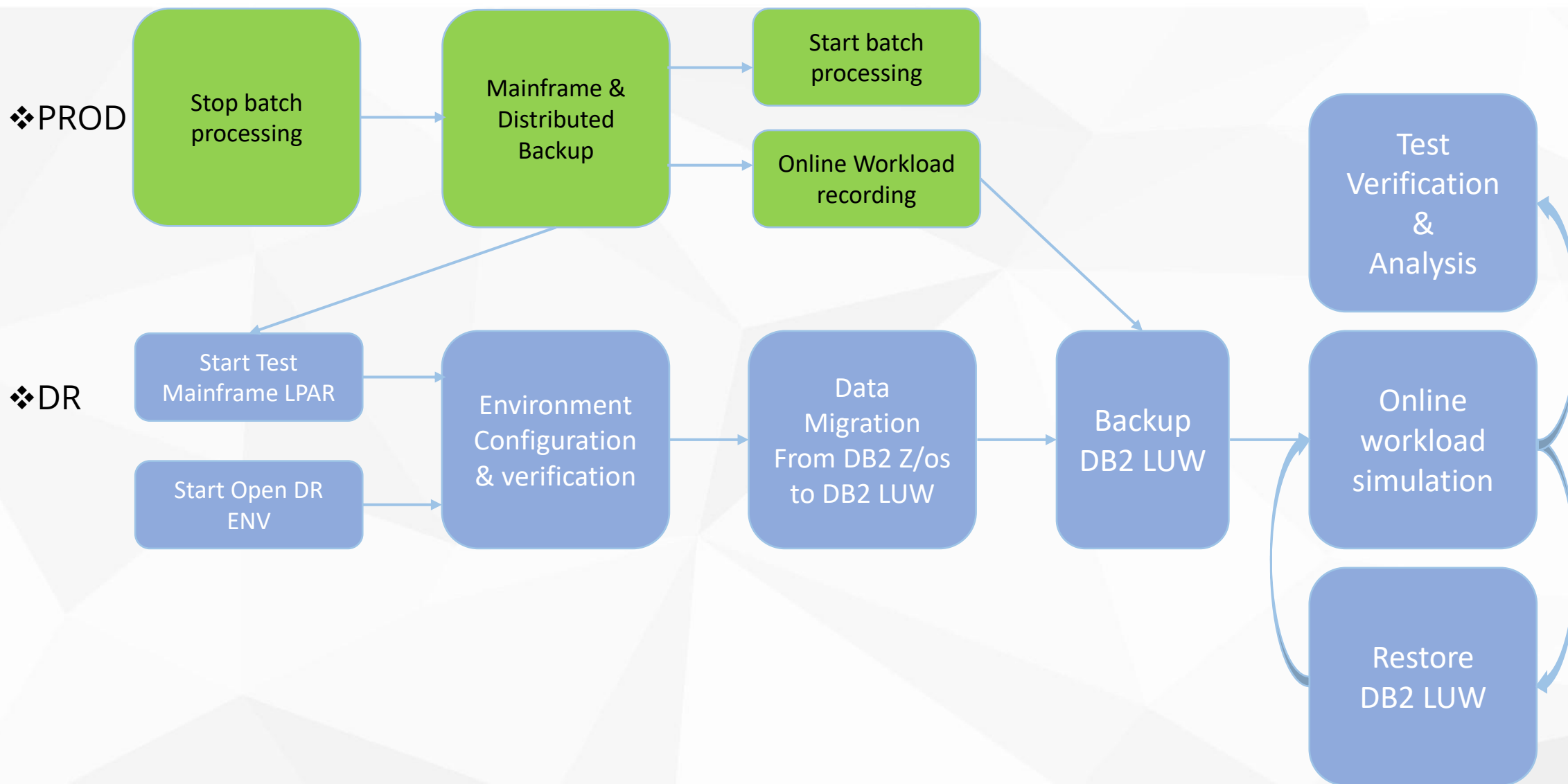


- ❖ Both backups on DR site
- ❖ Mainframe FlashCopy
- ❖ Distributed FullClone
 - ❖ 12 h to finish copy (300 TB)

DB2 PURESCALE SOLUTION - TESTING



DB2 PURESCALE SOLUTION - TESTING



Priority testing scope

Stress

DDLs/DMLs, utilities, reorg, db2readlogapi,
monitor table functions, etc

Topology change

Offline add/drop member
Online add member
Offline add/drop CF
Online add/drop CF

Maintenance

Single node enter/exit maintenance
Multiple node enter/exit maintenance
Upgrade (major version upgrade)
Rolling update (Fixpack/modpack update,
sequential/concurrent modes)

Backup and restore

Database/Tablespace level
Offline/Online
Incremental/delta
Restore with rebuild
Rollforward to eol, eob, pit
Crash during backup/restore and recovery

Priority testing scope

Recovery testing

Kill -9 member
Member drain
Kill -9 primary CF
Kill -9 secondary CF
Reboot member
Reboot primary CF
Reboot secondary CF
Kill -9 both CFs (GCR)
Reboot both CFs (GCR)
Cascading soft kill (member & CF)
Cascading reboot (member & CF)
Total cluster reboot
Database GPFS file system failure
Port down/up ethernet member
Port down/up ethernet primary CF
Port down/up ethernet secondary CF
Port down/up ROCE member
Port down/up ROCE primary CF
Port down/up ROCE secondary CF
ROCE cable pulls
Switch failure
Ethernet cable pulls

Complementary testing scope

pureScale HADR

Stress
Primary planned outage
Primary unplanned outage
Standby planned outage
Standby unplanned outage
Standby replay switch
Graceful takeover
Primary ARCU
ARCU Graceful takeover
HADR two AZ same region (Cloud)
HADR two AZ different region (Cloud)
Monitoring on standby keeping up
with primary

pureScale GDPC

Member failure
CF failure
TB failure
Site failure
Storage failure
Switch failure
ISL failure
Network cable pulls

Additional Reference: <https://www.ibm.com/docs/en/db2/11.1?topic=techniques-troubleshooting-db2-purescale-feature>

APPENDIX 11.5 + TSAMP

INSTALLATION PROCEDURE #1 - OVERVIEW

The order of operations that are logically grouped as follows.

- 1 prerequisites + Db2 binaries installation
- 2 passwordless SSH for 2 clusters + passwordless SSH accross 2 clusters

Prerequisites + Db2 binaries installation is straightforward. We do this accordingly to KC:
<https://www.ibm.com/docs/en/db2/11.5?topic=environment-installing-db2-purescale-linux>

We do above as installation of first PS cluster, on nodes pko-db2-1 and pko-db2-2.
And installation of Second PS cluster, on nodes pko-db2-3 and pko-db2-4.

Next we do steps specific to our procedure – we will explain on next slides:

- 3 storage specification + db2icrt on 2 clusters
- 4 add extra paths on pri cluster and create shareddb

Once steps above were done, remaining steps are following

- 5 prepare to move to sec cluster + gpfs sync
- 6 followed by failover shared db to sec cluster
- 7 followed by failback testing

INSTALLATION PROCEDURE #2 - DISKS

Now the storage – we will use 5 disks (Instance1 , Instance2 , SharedDB , Logs , Tiebreaker).
Note that on each cluster we list 4 disks, as Instance1 and Instance2 are mounted just once.
Other disks (SharedDB , Logs , Tiebreaker) are shared across clusters.

We also instruct db2icrt to use Db2Managed GPFS + gateway tiebreaker device.

PRI CLUSTER

```
sdb      nsd01_db2
sdd      nsd01_db2_inst_pri
sde      DB2_tiebreaker_disk
sdf      nsd01_db2_arch
```

```
/opt/ibm/db2/V11.5/instance/db2icrt -d -m pko-db2-1 -mnet pko-db2-1 -cf pko-db2-1 -
cfnet pko-db2-1 -m pko-db2-2 -mnet pko-db2-2 -cf pko-db2-2 -cfnet pko-db2-2 -
instance shared dev /dev/sdd -instance_shared_mount /db2priclust -tbdev /dev/sde
-u db2fenc1 db2inst1
```

SEC CLUSTER

```
sdb      nsd01_db2
sdd      nsd01_db2_inst_sec
sde      DB2_tiebreaker_disk
sdf      nsd01_db2_arch
```

```
/opt/ibm/db2/V11.5/instance/db2icrt -d -m pko-db2-3 -mnet pko-db2-3 -cf pko-db2-3 -
cfnet pko-db2-3 -m pko-db2-4 -mnet pko-db2-4 -cf pko-db2-4 -cfnet pko-db2-4 -
instance shared dev /dev/sdd -instance_shared_mount /db2secclust -tbdev /dev/sde
-u db2fenc1 db2inst1
```

INSTALLATION PROCEDURE #3 - FILESYSTEMS

Next, we add new directories to the cluster. We will be using TSAMP lssam command to see how it changes the cluster.

```
PRI CLUSTER -> pko-db2-1 -> root
```

```
-----  
lssam |grep db2pricluster
```

```
Online IBM.ResourceGroup:db2mnt-db2pricluster-rg Nominal=Online
```

```
  '- Online IBM.Application:db2mnt-db2pricluster-rs
```

```
    |- Online IBM.Application:db2mnt-db2pricluster-rs:pko-db2-1
```

```
    '- Online IBM.Application:db2mnt-db2pricluster-rs:pko-db2-2
```

```
Online IBM.Equivalency:db2mnt-db2pricluster-rg_group-equ
```

```
/opt/ibm/db2/V11.5/bin/db2cluster -cfs -create -filesystem db2shareddb -disk /dev/sdb -mount  
/db2shareddb
```

```
/opt/ibm/db2/V11.5/bin/db2cluster -cfs -create -filesystem db2log -disk /dev/sdf -mount /db2log  
-----
```

```
PRI CLUSTER -> pko-db2-1 -> db2inst1:
```

```
-----  
db2 create db PSSHARED on /db2shareddb/db DBPATH ON /db2shareddb/db
```

```
db2 connect to PSSHARED
```

```
db2 "create stogroup sgrplog on '/db2log/db2_log_path'"
```

```
lssam |grep db2shareddb
```

```
Online IBM.ResourceGroup:db2mnt-db2shareddb-rg Nominal=Online
```

```
  '- Online IBM.Application:db2mnt-db2shareddb-rs
```

```
    |- Online IBM.Application:db2mnt-db2shareddb-rs:pko-db2-1
```

```
    '- Online IBM.Application:db2mnt-db2shareddb-rs:pko-db2-2
```

```
Online IBM.Equivalency:db2mnt-db2shareddb-rg_group-equ
```

```
lssam |grep db2log
```

```
Online IBM.ResourceGroup:db2mnt-db2log-rg Nominal=Online
```

```
  '- Online IBM.Application:db2mnt-db2log-rs
```

```
    |- Online IBM.Application:db2mnt-db2log-rs:pko-db2-1
```

```
    '- Online IBM.Application:db2mnt-db2log-rs:pko-db2-2
```

```
Online IBM.Equivalency:db2mnt-db2log-rg_group-equ  
-----
```

INSTALLATION PROCEDURE #4 – UNMOUNT-PRI

So now we have 3 paths in the lssam (instance + database + logs). Lets prepare move to SEC cluster.
We do here first move procedure, where db2inst1 commands are interleaved with root commands.
We want to do some checking (more for safety, as if we skip those safety checks, procedure will work).

Here goes the procedure, we start with preparing PRI cluster:

PRI CLUSTER -> pko-db2-1 -> db2inst1 (uncatalog and stop)

```
-----  
db2 terminate  
db2 uncatalog db PSSHARED  
db2stop  
-----
```

PRI CLUSTER -> pko-db2-1 -> root (safety - path must be not used)

```
-----  
fuser -a /db2shared/db  
fuser -a /db2log/db2_log_path  
-----
```

PRI CLUSTER -> pko-db2-1 -> db2inst1 (setup failover to SEC - put locks on lssam):

```
-----  
db2_all "db2cluster -enter -maintenance -mount /db2shared/db"  
db2_all "db2cluster -enter -maintenance -mount /db2log"  
db2cluster -verify -maintenance -mount /db2shared/db  
db2cluster -verify -maintenance -mount /db2log  
db2cluster -unmount -filesystem db2shared/db  
db2cluster -unmount -filesystem db2log  
-----
```

PRI CLUSTER -> pko-db2-1 -> root (safety - filesystem must be not used)

```
-----  
/usr/lpp/mmfs/bin/mmlsmount all  
File system db2fs1 is mounted on 2 nodes.  
File system db2log is not mounted.  
File system db2shared/db is not mounted.  
-----
```

INSTALLATION PROCEDURE #5 - SYNCFSCONFIG

Now Sync SEC cluster - for this we need to transfer filesystems to SEC cluster.
Once we have passwordless ssh + db2locshh configured across 2 clusters, we can sync GPFS.
We create remote cluster (SEC cluster) definition file, and next sync from PRI cluster:

```
PRI CLUSTER -> pko-db2-1 -> root
```

```
-----  
vi remote_cluster_def
```

```
-----  
pko-db2-3.iic.pl.ibm.com
```

```
pko-db2-4.iic.pl.ibm.com  
-----
```

```
/usr/lpp/mmfs/bin/mmfsctl db2shared db syncFSconfig -n remote_cluster_def
```

```
/usr/lpp/mmfs/bin/mmfsctl db2log syncFSconfig -n remote_cluster_def  
-----
```

INSTALLATION PROCEDURE #6 – MOUNT-SEC

Once above is done, we can mount filesystems on SEC cluster. We will do the mount with db2cluster command. Note we do this with root on pko-db2-3 server from SEC cluster. We will also use lssam to observe the changes:

SEC CLUSTER -> pko-db2-3 -> root (mount filesystems)

```
-----  
/opt/ibm/db2/V11.5/bin/db2cluster -mount -filesystem db2shareddb  
/opt/ibm/db2/V11.5/bin/db2cluster -mount -filesystem db2log  
-----
```

Next, we catalog database and create mounts in the cluster. The lssam will show the difference:

SEC CLUSTER -> pko-db2-3 -> db2inst1 (create mounts)

```
-----  
db2 catalog db PSSHARED on /db2shareddb/db  
db2start  
db2cluster -cm -add -database_mounts PSSHARED
```

```
lssam |grep db2seccluster  
Online IBM.ResourceGroup:db2mnt-db2seccluster-rg Nominal=Online  
    '- Online IBM.Application:db2mnt-db2seccluster-rs  
        |- Online IBM.Application:db2mnt-db2seccluster-rs:pko-db2-3  
        '- Online IBM.Application:db2mnt-db2seccluster-rs:pko-db2-4  
Online IBM.Equivalency:db2mnt-db2seccluster-rg_group-equ
```

```
lssam |grep db2shareddb  
Online IBM.ResourceGroup:db2mnt-db2shareddb-rg Nominal=Online  
    '- Online IBM.Application:db2mnt-db2shareddb-rs  
        |- Online IBM.Application:db2mnt-db2shareddb-rs:pko-db2-3  
        '- Online IBM.Application:db2mnt-db2shareddb-rs:pko-db2-4  
Online IBM.Equivalency:db2mnt-db2shareddb-rg_group-equ
```

```
lssam |grep db2log  
Online IBM.ResourceGroup:db2mnt-db2log-rg Nominal=Online  
    '- Online IBM.Application:db2mnt-db2log-rs  
        |- Online IBM.Application:db2mnt-db2log-rs:pko-db2-3  
        '- Online IBM.Application:db2mnt-db2log-rs:pko-db2-4  
Online IBM.Equivalency:db2mnt-db2log-rg_group-equ  
-----
```

Now we create mount counter in the cluster:

SEC CLUSTER -> pko-db2-3 -> db2inst1 (create mounts)

```
-----  
db2 connect to PSSHARED  
db2 "create table T1(C1 INT)"  
db2 "insert into T1 values (1)"  
-----
```

INSTALLATION PROCEDURE #7 – UNMOUNT-SEC

So above is working fine! Now the failback. Note that so far we created SEC cluster by syncing filesystem. This required many steps to sync FS, create DB mounts that resulted in creation of TSAMP resources.

The failback is simpler, as we will leave all TSAMP resources on SEC cluster (similarly to PRI, locked in Issam) Reason is so that we have cluster ready to takeover filesystem easily. Here is the failback, leaving this paths in the cluster):

SEC CLUSTER -> pko-db2-3 -> db2inst1 (prepare to move):

```
-----
db2 terminate
db2 force application all
db2 uncatalog db PSSHARED
db2stop

db2_all "db2cluster -enter -maintenance -mount /db2shareddb"
db2_all "db2cluster -enter -maintenance -mount /db2log"

db2cluster -verify -maintenance -mount /db2shareddb
db2cluster -verify -maintenance -mount /db2log

db2cluster -unmount -filesystem db2shareddb
db2cluster -unmount -filesystem db2log
-----
```

INSTALLATION PROCEDURE #8 – MOUNT-PRI

At this point we can switch roles - after we setup like above (so have those Issam resources on both sites), We can use following procedure to takeover DB on PRI cluster. Those will be following steps:

PRI CLUSTER -> pko-db2-1 -> db2inst1 (takeover DB):

```
db2cluster -mount -filesystem db2shareddb
db2cluster -mount -filesystem db2log

db2_all "db2cluster -exit -maintenance -mount /db2shareddb"
db2_all "db2cluster -exit -maintenance -mount /db2log"

db2cluster -verify -maintenance -mount /db2shareddb
db2cluster -verify -maintenance -mount /db2log

db2start
db2 catalog db PSSHARED on /db2shareddb/db
db2 connect to PSSHARED
db2 "select * from T1"
```

1 record(s) selected. <- so we have changes done on second cluster

```
db2 "insert into T1 values (2)"
```

DB20000I The SQL command completed successfully. <- here we have takeover counter

INSTALLATION PROCEDURE #9 – SWITCH PRI-SEC

After all above we have setup ready. Now we can do role switch with following:

PRI CLUSTER -> pko-db2-1 -> db2inst1 (prepare to move):

```
-----
db2 terminate
db2 force application all
db2 uncatalog db PSSHARED
db2stop

db2_all "db2cluster -enter -maintenance -mount /db2shareddb"
db2_all "db2cluster -enter -maintenance -mount /db2log"

db2cluster -verify -maintenance -mount /db2shareddb
db2cluster -verify -maintenance -mount /db2log

db2cluster -unmount -filesystem db2shareddb
db2cluster -unmount -filesystem db2log
-----
```

SEC CLUSTER -> pko-db2-3 -> db2inst1 (takeover DB):

```
-----
db2cluster -mount -filesystem db2shareddb
db2cluster -mount -filesystem db2log

db2_all "db2cluster -exit -maintenance -mount /db2shareddb"
db2_all "db2cluster -exit -maintenance -mount /db2log"

db2cluster -verify -maintenance -mount /db2shareddb
db2cluster -verify -maintenance -mount /db2log

db2start
db2 catalog db PSSHARED on /db2shareddb/db
db2 connect to PSSHARED
db2 "select * from T1"
```

2 record(s) selected. <- takeover counter before the move

```
db2 "insert into T1 values (3)"
```

DB20000I The SQL command completed successfully. <- takeover counter increase

INSTALLATION PROCEDURE #10 – SWITCH SEC-PRI

And of course we can takeover back (in reverse order):

SEC CLUSTER -> pko-db2-3 -> db2inst1 (prepare to move):

```
-----
db2 terminate
db2 force application all
db2 uncatalog db PSSHARED
db2stop

db2_all "db2cluster -enter -maintenance -mount /db2shareddb"
db2_all "db2cluster -enter -maintenance -mount /db2log"

db2cluster -verify -maintenance -mount /db2shareddb
db2cluster -verify -maintenance -mount /db2log

db2cluster -unmount -filesystem db2shareddb
db2cluster -unmount -filesystem db2log
-----
```

PRI CLUSTER -> pko-db2-1 -> db2inst1 (takeover DB):

```
-----
db2cluster -mount -filesystem db2shareddb
db2cluster -mount -filesystem db2log

db2_all "db2cluster -exit -maintenance -mount /db2shareddb"
db2_all "db2cluster -exit -maintenance -mount /db2log"

db2cluster -verify -maintenance -mount /db2shareddb
db2cluster -verify -maintenance -mount /db2log

db2start
db2 catalog db PSSHARED on /db2shareddb/db
db2 connect to PSSHARED
db2 "select * from T1"
```

3 record(s) selected. <- takeover counter before the move

```
db2 "insert into T1 values (4)"
```

DB20000I The SQL command completed successfully. <- takeover counter increase

THANK YOU

thank you !