Making HA Pace with Pacemaker

- **What is Pacemaker**
- Pacemaker vs TSAMP
- Pacemaker HADR Configurations
- Quorum Devices
- Pacemaker Mutual Failover Configurations
- Test Examples

What is Pacemaker?

DB2 Knowledge Centre:

"*Pacemaker is an open source high-availability cluster resource manager software that runs on a set of nodes. Together with Corosync, an open source group communication system that provides ordered communication delivery, cluster membership, quorum enforcement, and other features among the nodes, it helps detect component failures and orchestrate necessary failover procedures to minimize interruptions to applications..*"

- automates problem detection and failovers
- manages the availability of resources (next slide!)
- provides high availability (HA)

What is Pacemaker - what are Resources?

Resources are services on a host that must be kept highly available.

In Db2®, the following are considered resources:

HADR:

- Db2 instance
- HADR Database
- Ethernet network adapter
- Virtual IP address

- Mutual Failover
  - Db2 instance
  - Shared disk mount points (instance, DB dir., Log, mirrorlog)
  - Ethernet network adapter
  - Virtual IP address

What is Pacemaker - what it does for DB2 resiliency?

The DB2 HA/DR Resiliency Ladder:

- Level #01 – Standalone DB
  - Backups, log archive, disk mirrors
  - RPO=?; RTO>>
  - **potentially DR only**
- Level #02 – HADR Cluster
  - Secondary copy (1-3) of the database
  - RPO=0 sec.; RTO=?;
  - **DR only**
- Level #03 – Pacemaker
  - Provides fully automated failovers
  - RPO=0 sec.; RTO=<seconds>;
  - **HA+DR**

Pacemaker is The Next Level of DB2 resiliency!

Agenda

- What is Pacemaker
- **Pacemaker vs TSAMP**
- Pacemaker HADR Configurations
- Quorum Devices
- Pacemaker Mutual Failover Configurations
- Test Examples

Pacemaker vs TSAMP (01/03)

The "new" vs the "old" technology:

TSAMP: Tivoli System Automation for MultiPlatforms

- Both installed via the DB2 installation:

  db2_install -b /opt/ibm/DB2 -f NOTSAMP -f NOPCMK

- Both configured via command line utilities:
  - TSAMP: db2haicu -interactive or XML input cfg file
  - Pacemaker: db2cm -simpler, easier to automate

db2haicu: https://www.ibm.com/docs/en/db2/11.5?topic=db2haicu-tool
db2cm: https://www.ibm.com/docs/en/db2/11.5?topic=pacemaker-db2cm-db2-cluster-manager-utility

Pacemaker vs TSAMP (02/03)

- Ease of use:
  - Pacemaker - flexible, better performance ("up to 30% faster")
  - TSAMP - anything but...
- Troubleshooting:
  - Pacemaker - IBM Support
  - TSAMP - third party support
- Support and development:
  - Pacemaker - strategic HA component
  - TSAMP - unsupported as of v12.1 (except AIX)
- Cloud-ready?
  - Pacemaker ☑
  - TSAMP ☒

More details: https://www.idug.org/news/the-book-of-db2-pacemaker--chapter-1--red-pill-or-blue-pill

https://www.idug.org/news/the-book-of-db2-pacemaker--chapter-1--red-pill-or-blue-pill

Pacemaker vs TSAMP (03/03)

How to replace TSAMP with Pacemaker?

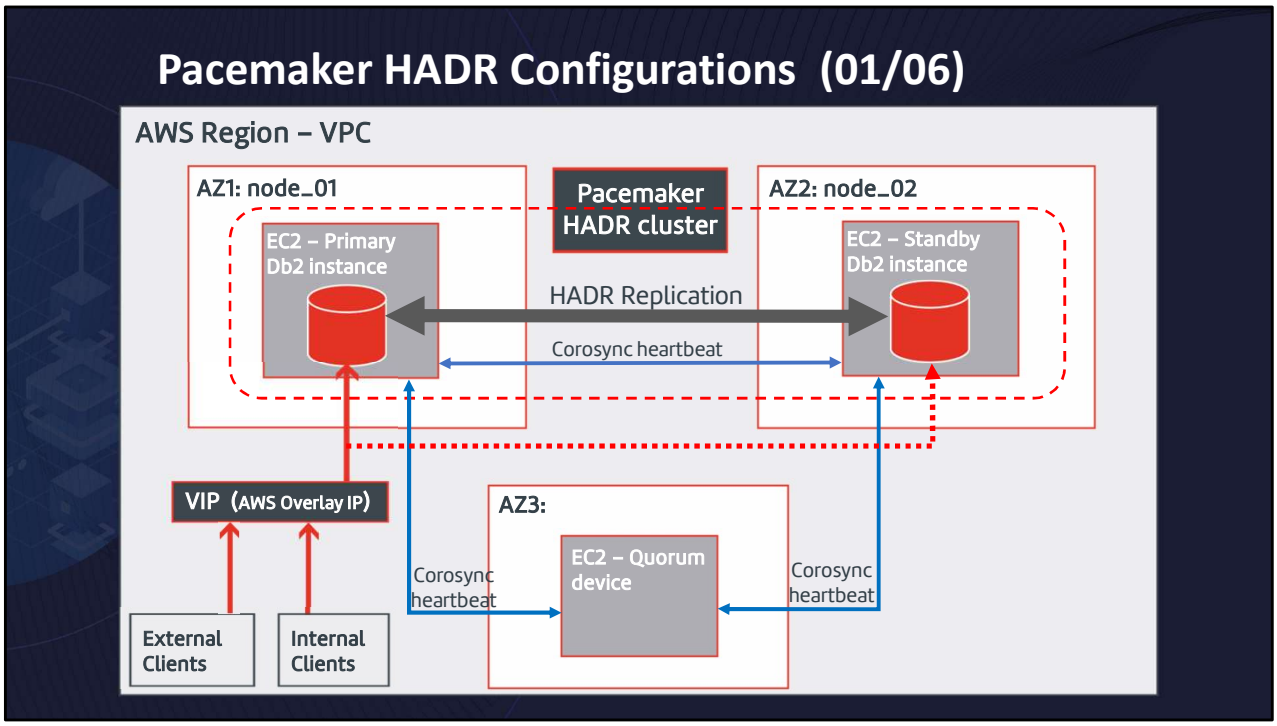
- Backup TSAMP configuration
- Delete TSAMP
- Install Pacemaker
- Configure Pacemaker cluster

NOTE: use reverse procedure to failback to TSAMP from Pacemaker!

How to replace TSAMP with Pacemaker:
https://www.ibm.com/docs/en/db2/11.5?topic=hpo-replacing-existing-tivoli-sa-mp-managed-db2-instance-pacemaker-managed-hadr-db2-instance
And vice-versa:
https://www.ibm.com/docs/en/db2/11.5?topic=hpo-replacing-pacemaker-managed-hadr-db2-instance-tivoli-sa-mp-managed-db2-instance

Agenda

- What is Pacemaker
- Pacemaker vs TSAMP
- **Pacemaker HADR Configurations**
- Quorum Devices
- Pacemaker Mutual Failover Configurations
- Test Examples
- ...



BLOG: https://www.triton.co.uk/configuring-db2-pacemaker-hadr-cluster-with-qdevice-in-aws/

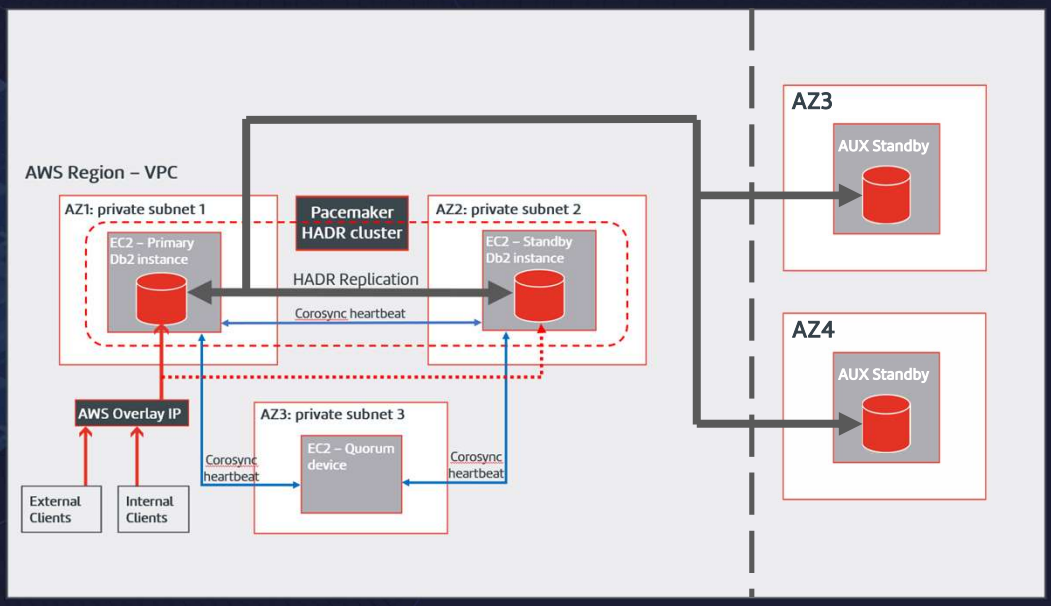
Pacemaker HADR Configurations (02/06)

Type: Shared nothing architecture.

Supported HADR config's:

- **Single Standby**
- **Multiple Standbys**
  - **No suport for automatic failovers to remote AUX Stbys!**

Pacemaker HADR Configurations (02b/06)



https://www.ibm.com/docs/en/db2/11.5?topic=hmsd-scenario-deploying-two-sites-multiple-standby-cluster-same-site-failover-automation

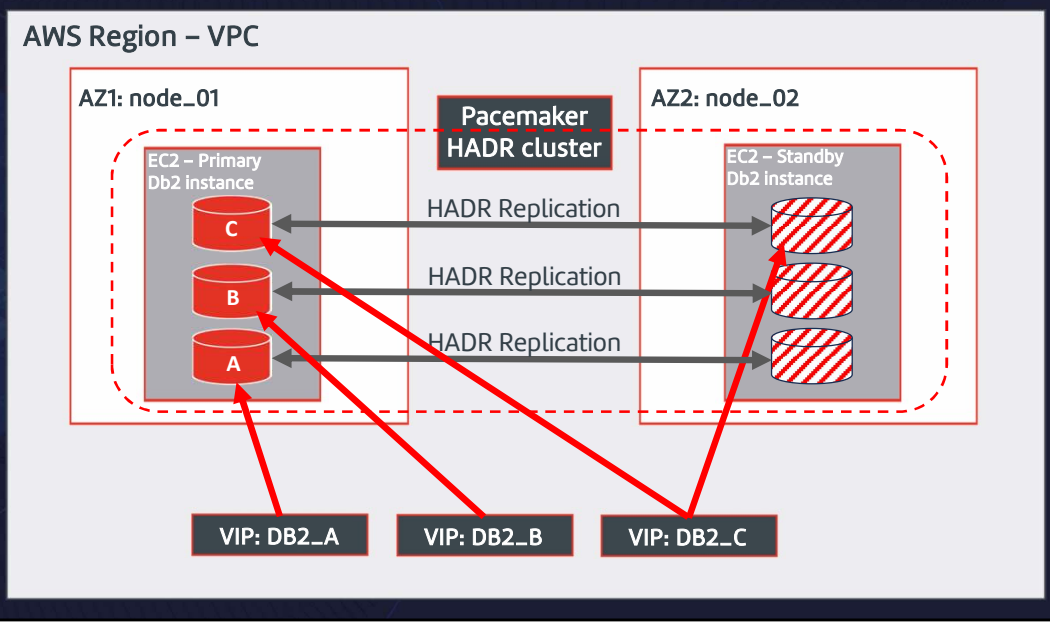
Pacemaker HADR Configurations (03/06)

Type: Shared nothing architecture.

Supported HADR config's:

- Single Standby
- Multiple Standbys
  - No Pacemaker suport for remote AUX Stbys!
- **One VIP address per database**

Pacemaker HADR Configurations (03b/06)



Pacemaker HADR Configurations (04/06)

HADR Restrictions:

- All HADR databases must exist on different systems.
- All HADR databases must be started
  - All Standby databases must be in peer state.
- Supported HADR synchronization modes: SYNC or NEARSYNC.
- Set **HADR_PEER_WINDOW** >= 120 seconds for all HADR databases.
- Disable the Db2 fault monitor.

When you set **hadr_peer_window** to a non-zero time value, then a HADR primary-standby database pair continues to behave as though still in peer state, for the configured amount of time, if the primary database loses connection with the standby database. This helps ensure data consistency.

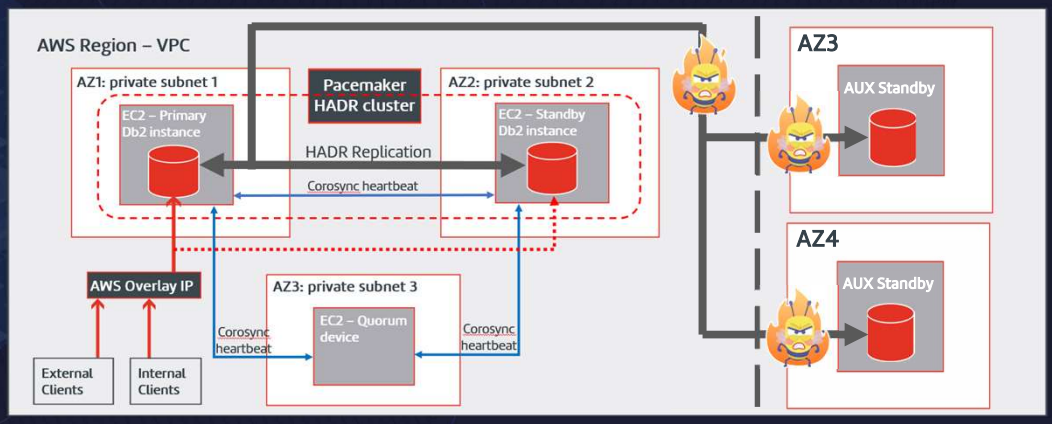
hadr_timeout database configuration parameter
If an HADR database does not receive any communication from its partner database for longer than the length of time that is specified by the **hadr_timeout** database configuration parameter, then the database concludes that the connection with the partner database is lost. If the database is in peer state when the connection is lost, then it moves into disconnected peer state if the **hadr_peer_window** database configuration parameter is greater than zero, or into remote catchup pending state if **hadr_peer_window** is not greater than zero. The state change applies to both primary and standby databases.

hadr_peer_window database configuration parameter
The **hadr_peer_window** configuration parameter does not replace the **hadr_timeout** configuration parameter. The **hadr_timeout** configuration parameter determines how long an HADR database waits before it considers that its connection with the partner database as failed. The **hadr_peer_window** configuration

parameter determines whether the database goes into disconnected peer state after the connection is lost, and how long the database remains in that state. HADR breaks the connection as soon as a network error is detected during send, receive, or poll on the TCP socket. HADR polls the socket every 100 milliseconds. This frequency allows it to respond quickly to network errors detected by the OS. Only in the worst case does HADR wait until the timeout to break a bad connection. In this case, a database application that is running at the time of failure can be blocked for the time equal to the sum of the **hadr_timeout** and **hadr_peer_window** database configuration parameters.

Pacemaker HADR in NAT Environment (05/06)

Network address translation (NAT) is a method of mapping an IP address space into another by modifying network address information in the IP header of packets.



https://www.triton.co.uk/db2-hybrid-hadr-clusters/

Pacemaker HADR Configurations (06/06)

The DB2 Resiliency Ladder - revisited:

- Level #01 – Standalone DB -potential DR
- Level #02 – HADR Cluster -DR only
- Level #03 – Pacemaker -HA+DR
- Level #04 – Hybrid HADR Cluster -HA+DR^2
  - can include Pacemaker (partially)
  - eliminate Cloud providers or on-prem data centres as SPF

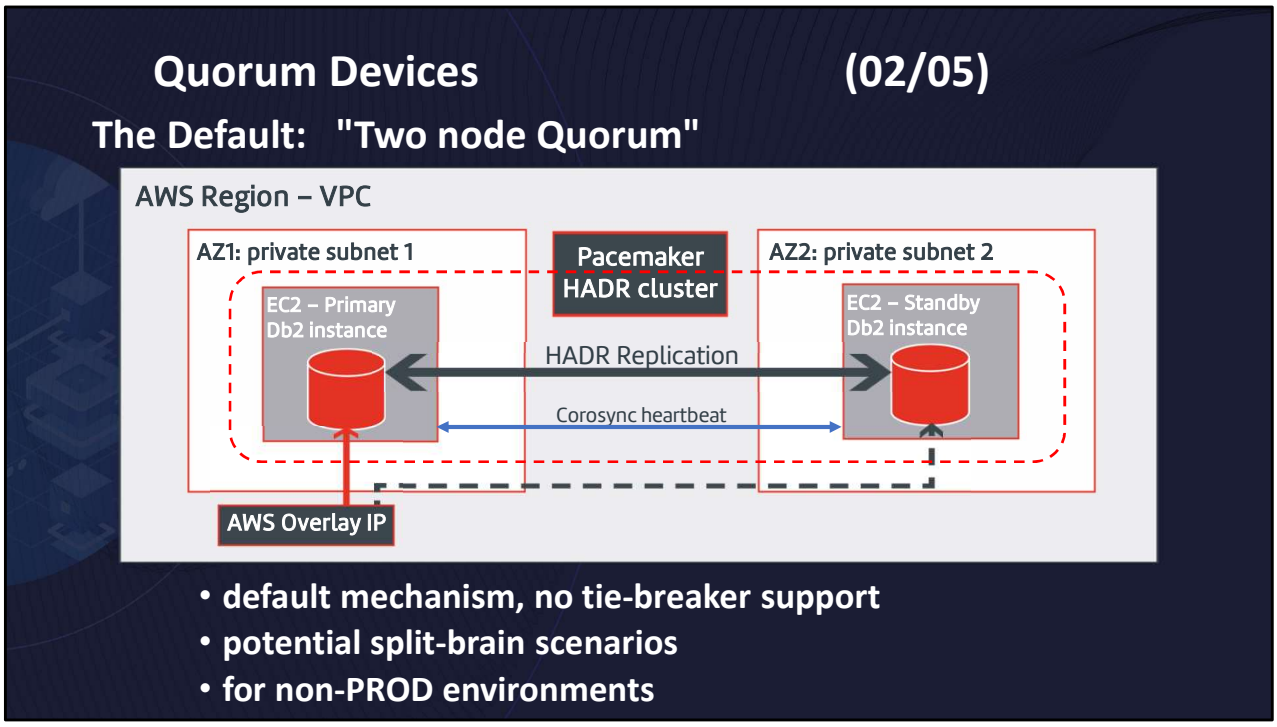
Agenda

- What is Pacemaker
- Pacemaker vs TSAMP
- Pacemaker HADR Configurations
- **Quorum Devices**
- Pacemaker Mutual Failover Configurations
- Test Examples
- ...

Quorum Devices (01/05)

- AKA "Tie-breakers"
- help resolve voting in clusters with even number of nodes
- if you don't create one, one will be created for you

DB2 Knowledge Centre: https://www.ibm.com/docs/en/db2/11.5?topic=component-quorum-devices-support-pacemaker



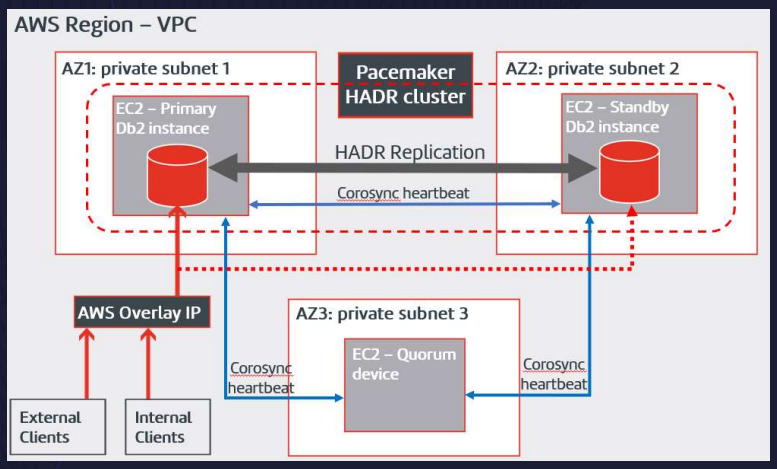
Two-node quorum

The two-node quorum is the default mechanism. Because no tie-breaker mechanism exists, the two-node quorum is prone to the split-brain scenario. It is not intended for production environments.

Quorum Devices (03/05)

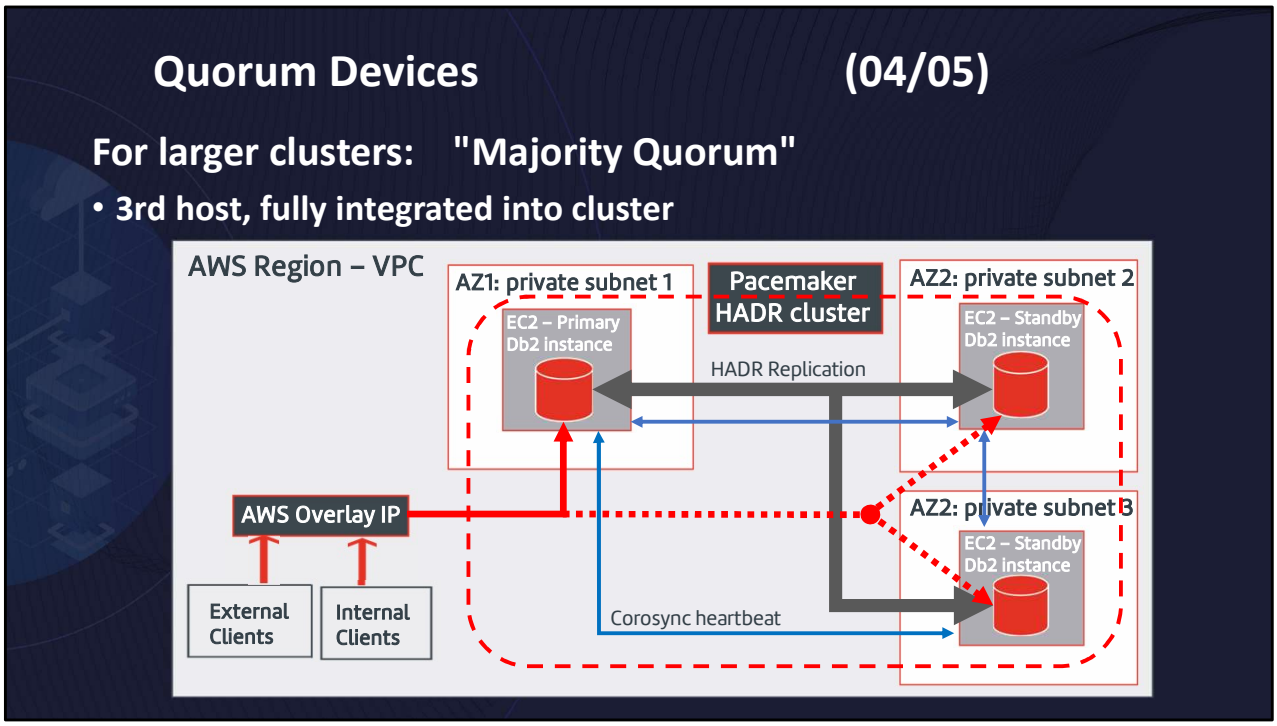
For PROD use: "Qdevice quorum"

- **Separate host, not part of the cluster**
- **more reliable than std. Network IP tiebreakers**
- **can be reused for other clusters!**



QDevice quorum

QDevice is similar to a Network IP tiebreaker in the sense that it requires an external resource accessible by all hosts in the current Pacemaker cluster. But they differ immensely in terms of reliability. QDevice is much more reliable because the quorum decision logic is more robust than a simple TCP/IP ping to the external IP address. QDevice requires that the resource is placed on a separate host, similar to a majority quorum requirement. However, setup is simplified, as the host with the QDevice does not need to be part of the Pacemaker quorum. The QDevice is the best blended quorum solution that combines reliability and simplicity.



Majority quorum
The Majority Quorum avoids split-brain scenarios by adding a third node to the cluster for arbitration. In a split-brain scenario, the side that successfully acquires the third node is the surviving side. The difference between QDevice and Majority Quorum is that the third node is fully integrated into the cluster.

Pacemaker HADR Configurations (05/05)

Quorum types - which is better?

Quorum type	Advantages	Disadvantages
Two Node	•Simplest setup. •No additional hardware or software configuration.	•Potential split-brain scenario, leading to dual primary phenomenon.
QDevice	•More reliable than Two-Node quorum. •No need to include the third host as part of the cluster. •No need to include the full Pacemaker cluster software stack on the third host. Only one Corosync RPM is needed.	•Requires a TCP/IP accessible host from the primary and standby hosts.
Majority	•More reliable than Two-Node quorum.	•Need to include the full Pacemaker cluster stack that is installed and configured on the third host.

Agenda

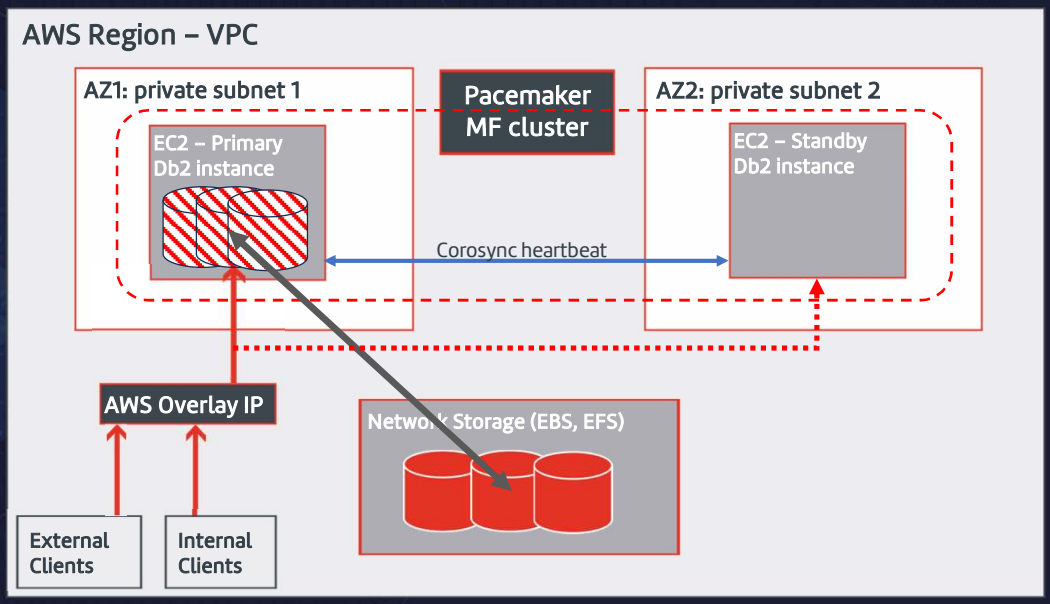
- What is Pacemaker
- Pacemaker vs TSAMP
- Pacemaker HADR Configurations
- Quorum Devices
- **Pacemaker Mutual Failover Configurations**
- Test Examples

Pacemaker MF Configurations (01/08)

The MF Definition:

"Db2 Mutual Failover is a highly available deployment where two hosts have an identical Db2 instance installation and a shared mount, with only one host active at one time."

Pacemaker MF Configurations (02/08)



Pacemaker MF Configurations (03/08)

Type: Disk sharing architecture.

HADR: N/A

- **Support for 2 nodes only**
- **One VIP address per DB2 instance (multiple databases)**

Quorum Devices:

- **Two-node quorum (for non-PROD environments)**
- **[no Majority quorum]**
- **QDevice quorum (3rd host, not part of cluster)**
- **Fencing (*)**

Setting up fencing on AWS:
https://www.ibm.com/support/pages/node/6829815

Pacemaker MF Configurations (05/08)

Associating Mounted Resources with MF Partitions

db2cm -add -dbMount

vs.

db2cm -create -mount

"**add dbMount**" -used for existing databases only:

- Local database directory
- log path
- Mirror log path
- does NOT include DB Data Paths, Logarchive, ...

"**create mount**" - used for *any* mounted FS...

"add dbMount": https://www.ibm.com/docs/en/db2/11.5?topic=domain-associating-mount-resource-partition
vs.
"create mount": https://www.ibm.com/docs/en/db2/11.5?topic=domain-associating-mount-point-remote-file-system

Pacemaker MF Configurations (06/08)

Mutual Failover vs. HADR clusters:

- **Shared Disk Architecture vs. Shared Nothing Architecture**
  - Very Large Databases - Data Warehouses, Lakes, Oceans, …
- **Single Standby vs. Multiple Standbys**
  - MF == HA only...
  - HADR == full HADR 🙃
- **Slower Failovers?**
  - (?detected in Test Envs, but need some more testing here?)
- **Fencing required to prevent data corruption**
  - "SBD Fencing" - configured along with QDevice

SBD Fencing

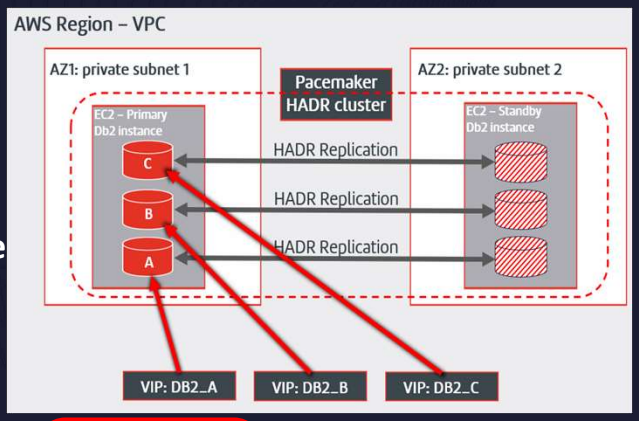
"Mutual Failover utilizes SBD, short for STONITH Block Device, to perform fencing operations in case a node loses quorum. In essence, in the case of communication breakage and split-brain occurs, QDevice Quorum plus diskless SBD will instruct the node without quorum to be self-fenced through a preconfigured software watchdog(/dev/watchdog)."
(from: https://www.idug.org/news/the-book-of-pacemaker---chapter-4-quorumania)

Pacemaker MF Configurations (07/08)

Mutual Failover vs. HADR:

VIP Address(es)

- MF: One VIP per DB2 Instance
- HADR: One VIP per DB2 Database



VIP Address Association:

db2cm -create -primaryVIP <VIP_ADDR> -partition <PID> -instance <INSTID>

vs.

db2cm -create -primaryVIP <VIP_ADDR> -db <DBNAME> -instance <INSTID>

Pacemaker MF Configurations (08/08)

Mutual Failover vs. HADR:

How to initiate the Failover

MF:

db2cm -move -partition <PID> -instance <INST> -host <HOST>

HADR:

db2 takeover HADR on database <DBNAME> [by force]

Agenda

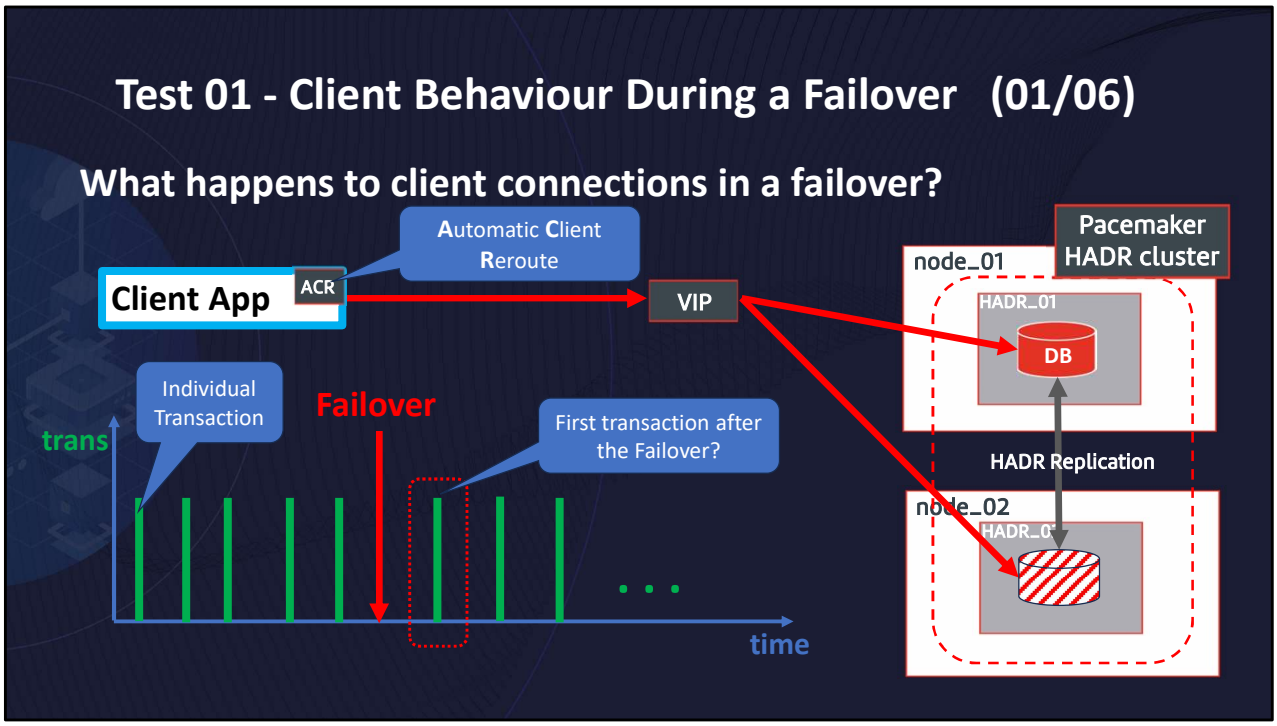
- What is Pacemaker
- Pacemaker vs TSAMP
- Pacemaker HADR Configurations
- Quorum Devices
- Pacemaker Mutual Failover Configurations
- **Test Examples**

Test Case 01

Client Behaviour During a Failover

Test 01 - Client Behaviour During a Failover (01/06)

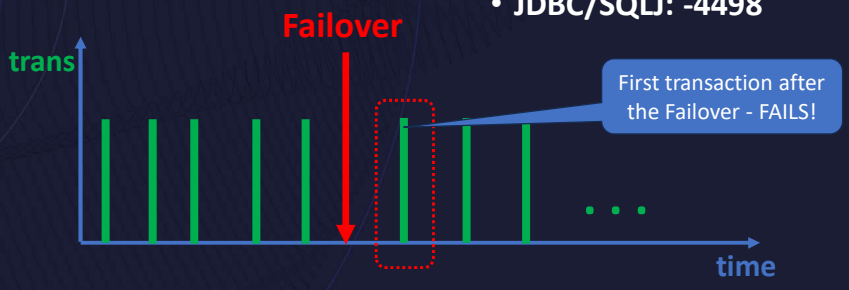
What happens to client connections in a failover?



Test 01 - Client Behaviour During a Failover (02/06)

Case A) The "standard" Failover behaviour

- Client reconnected automatically (ACR), but...
- The first transaction fails and is rolled back!
- Error returned to the client application:
  - CLI (.Net): SQL30108N
  - JDBC/SQLJ: -4498



Test 01 - Client Behaviour During a Failover (03/06)

SQL30108N A connection failed in an automatic client reroute environment. The transaction was rolled back.

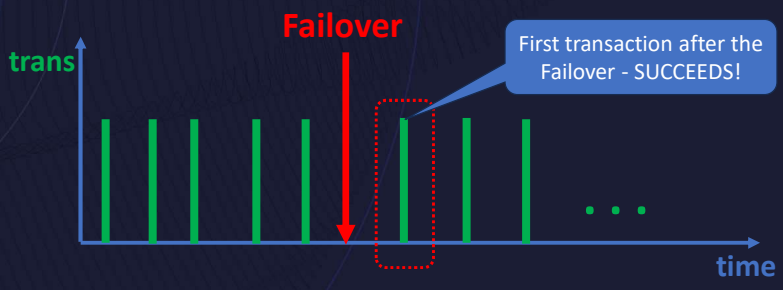
Solution:

- **must make the client aware of this SQL ERRCODE!**
- **resubmit the failed transaction!**

Test 01 - Client Behaviour During a Failover (04/06)

Case B) The "Seamless" Failover

- Client reconnected automatically (ACR)
- First transaction re-executed automatically
- No error returned to the Client!!

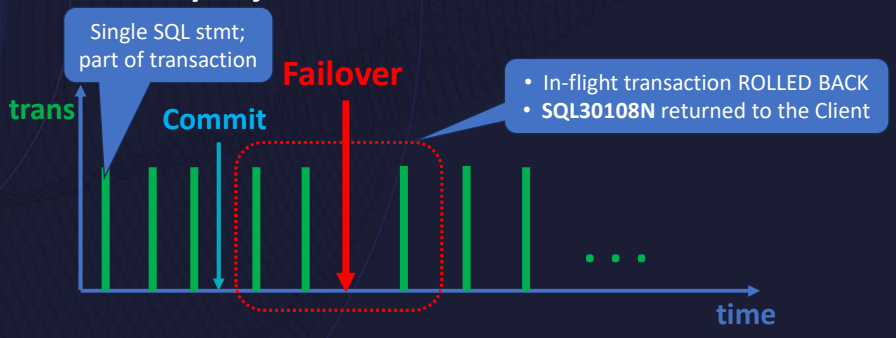


Test 01 - Client Behaviour During a Failover (05/06)

Case B) The "Seamless" Failover, continued…

CAVEAT(S):

1. **Must be the 1st SQL statement in a Transaction!!**
2. **in-memory objects…**



Test 01 - Client Behaviour During a Failover (06/06)

Case B) The "Seamless" Failover, continued…

The Million Dollar question:

- **How to configure the Seamless Failover?**

CLI (.Net):

- **enableAcr = "true"**
- **enableSeamlessAcr = "true"**

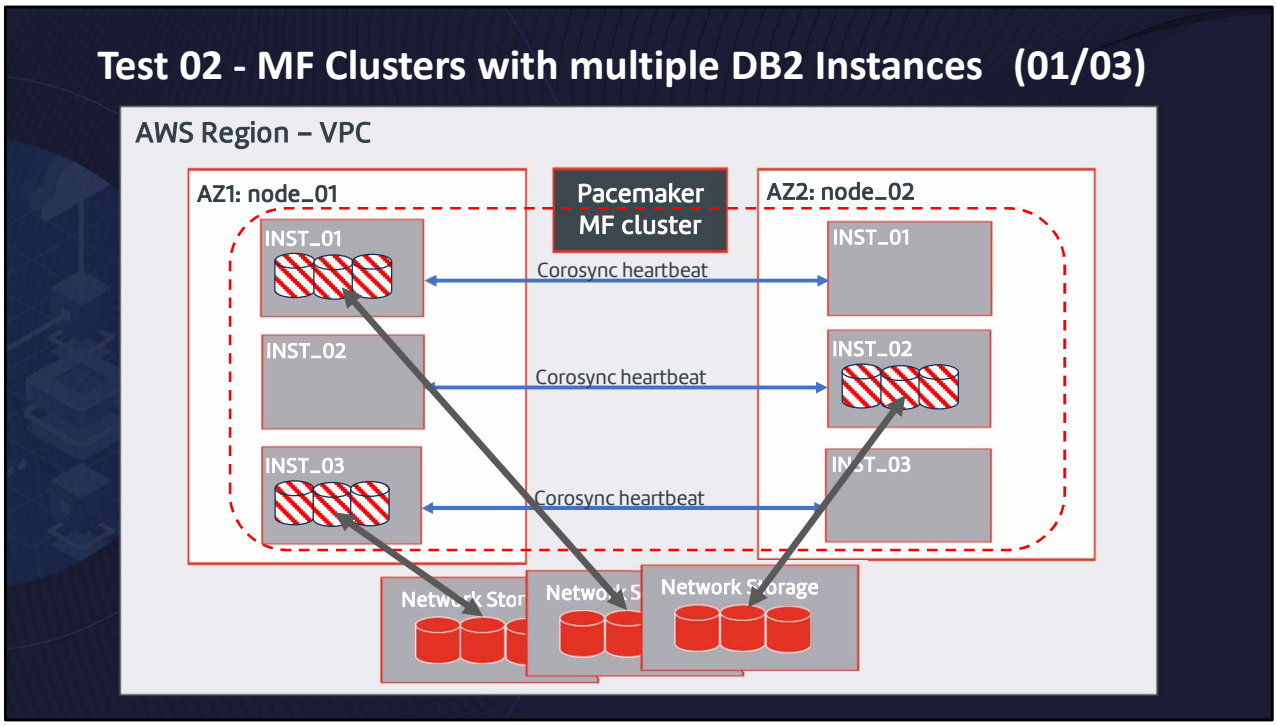
JAVA:

- **enableSeamlessFailover = 1**

Details: https://www.triton.co.uk/db2-pacemaker-hadr-seamless-failover/

Test Case 02

MF Clusters with multiple DB2 Instances



BLOG: https://www.triton.co.uk/configuring-db2-pacemaker-hadr-cluster-with-qdevice-in-aws/

Test 02 - MF Clusters with multiple DB2 Instances (02/03)

Why multiple DB2 instances?

- Several independent application environments
- Not very busy databases (PROD)
- Requirement to independently update each Env.

Problem:

- Failover in one DB2 instance induced failovers in other DB2 instances

Solution:

- Edit the Pacemaker configuration to "fix the DB2 instance stickiness" (IBM Support)

Test 02 - MF Clusters with multiple DB2 Instances (03/03)

How to fix the "DB2 Instance Stickiness"

1. run the command: **crm configure edit**
2. update the **resource-stickiness** to 1000
3. update the parameter for both instance partitions

Note if the resource-stickiness doesn't exist in the crm configure edit output , you would need to add it to the line that starts with "meta", here is an example :

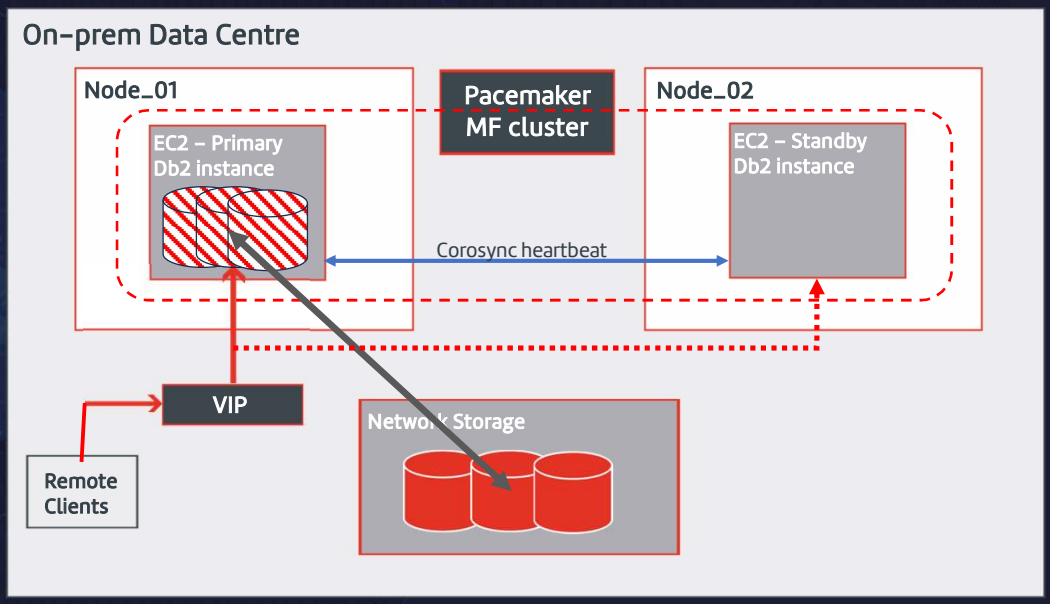
```
primitive db2_db2inst1_0 db2partition \
        params instance=db2inst1 dbpartitionnum=0 \
        op monitor timeout=120s interval=10s on-fail=restart \
        op start interval=0s timeout=900s \
        op stop interval=0s timeout=900s \
        meta is-managed=true resource-stickiness=1000 target-role=Started
```

4. save the configuration and exit

Test Case 03

Disk Corruption in a MF Cluster

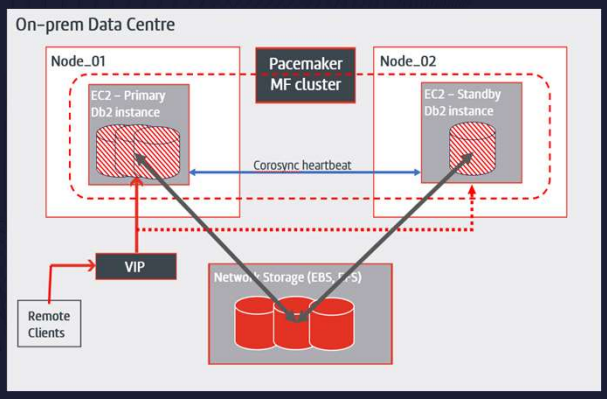
Test 03 - Disk Corruption in a MF Cluster (01/02)



Test 03 - Disk Corruption in a MF Cluster (02/02)

XFS Disk corruption:

- Early days of testing...
- Caused by mounting shared disks on **both nodes** at **the same time**
- Caused by (casual) logging into the **db2inst1** acct on the Standby!
- Problem present in RHEL8(+) due to OS kernel changes.
- Solution (temp workaround): use "OS masking"

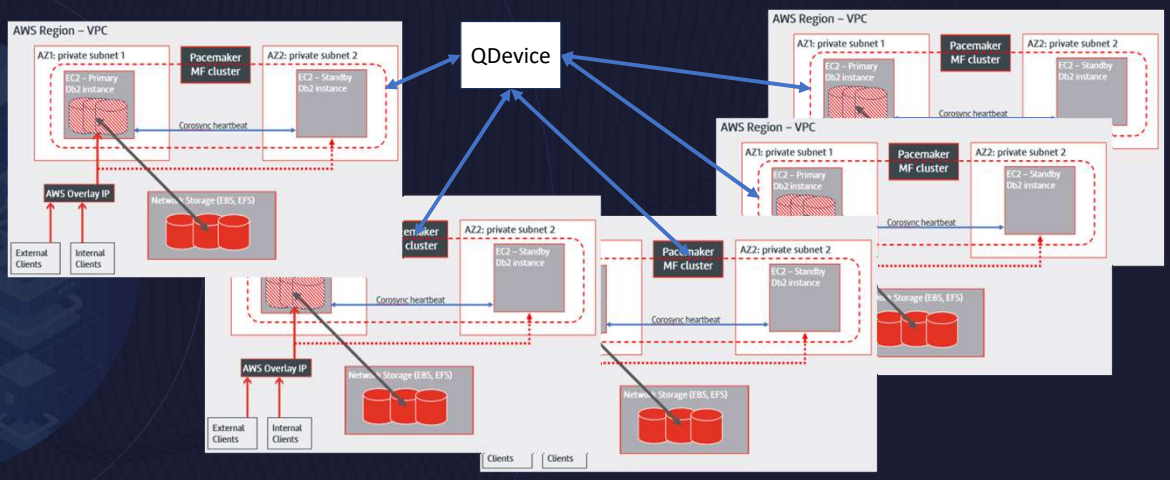


https://www.ibm.com/mysupport/s/defect/aCI3p000000XoPz/dt243737?language=en_US

Test Case 04

Covering Multiple Clusters with a single QDevice

Test 04 - Covering Multiple Clusters with a single QDevice



- Current max: **5 MF clusters** with 7 active DBs, with tendency to grow
- Restriction: all clusters must have different names.

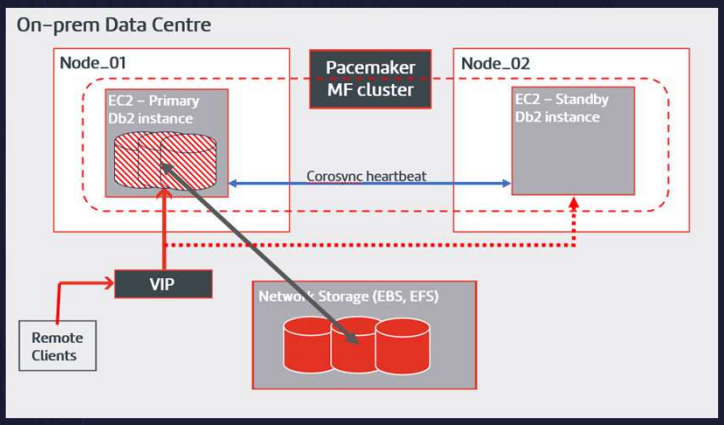
Test Case 05

Takeovers in MF timing out

Test 05 - Takeovers in MF timing out

<u>Initiating the failover by hand:</u>

db2cm -move -partition <PID>
-instance <INST>
-host <HOST>

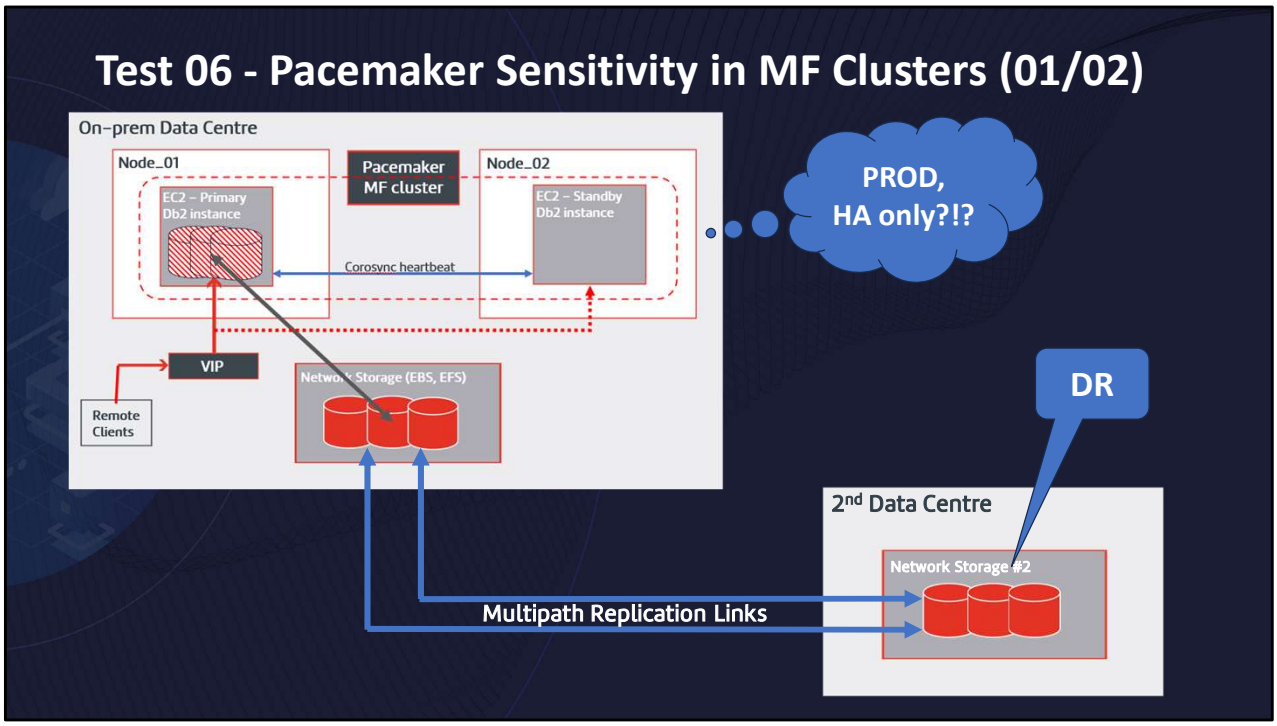


- Root cause found in DB2DIAG.LOG:
  - "communication error" messages - suggesting DB2INST1 has invalid (expired) password
- Solution: reactivate the DB2INST1 acct.
  - set the DB2INST1 acct. to never expire

Test Case 06

Pacemaker sensitivity in MF Clusters

Test 06 - Pacemaker Sensitivity in MF Clusters (01/02)



Test 06 - Pacemaker Sensitivity in MF Clusters (02/02)

The Problem:

- occasional replication link failures (100-200 msec);
- temporary latency spikes in storage replication (typically <=10 sec).
- storage remains 100% available (caching + 120 sec. timeout)!
- DB2 does not report any problems!
- Pacemaker detects the "problem" immediately and initiates a failover
  - doesn't appreciate 120 sec. Storage timeout
  - doesn't use FS caching (uses Direct IO)

Temporary Workaround:

- reduce sensitivity in db2fs (PCMK component that checks disks)

Permanent Fix:

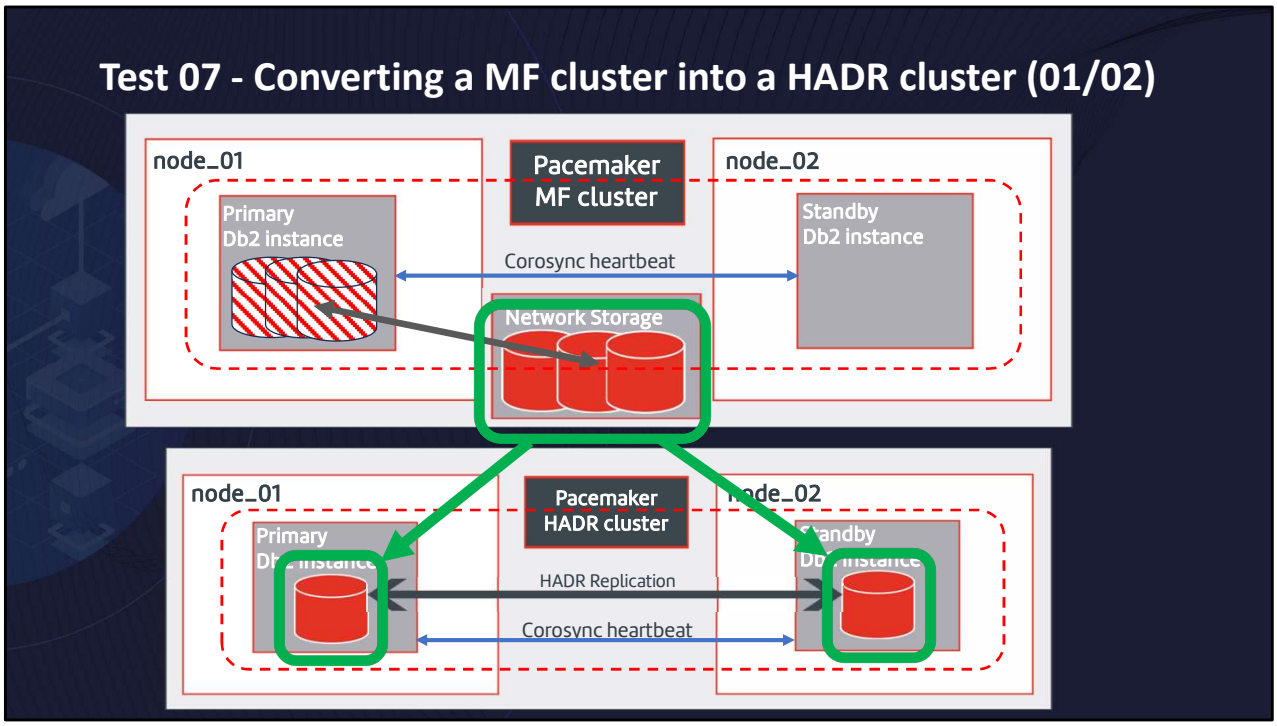
- N/A, but requesting APAR via a DB2 AHA Idea:

[link to DB2 AHA idea]

Test Case 07

Converting a MF cluster into a HADR cluster

Test 07 - Converting a MF cluster into a HADR cluster (01/02)



Test 07 - Converting a MF cluster into a HADR cluster (02/02)

Steps:

- delete MF Pacemaker cluster (config.)
- convert shared disks into dedicated and clone to STBY node
- configure DB2 HADR cluster
- configure HADR Pacemaker cluster
  - "db2cm -create" command fails with:

```
Error: Unable to set the cfg parameter cluster_mgr to PACEMAKER
```

Solution:

- do NOT clone (ex)shared disks
- on Stby: create fresh set of disks + DB2 install, Instance config;
- HADR/Pacemaker cluster config

Test Case XY

Test cases only covering MF?
What about HADR clusters?

→ No issues worth discussing!

Wrap up

Pacemaker Cluster Configurations: HADR vs. MF

- **HADR**
  - **flexible**
  - **mature**
  - **supports DR and HA/DR scenarios**
- **Mutual Failover**
  - **new kid on the block**
  - **very large databases (or numerous smaller ones)**
  - **supports only HA scenarios**

Cases examined:

Mutual Failover	**: 6**
HADR	**: 2**

Further Reading:

https://www.ibm.com/docs/en/db2/11.5?topic=feature-integrated-solution-using-pacemaker
https://www.ibm.com/mysupport/s/defect/aCI3p000000XoPz/dt243737?language=en_US
https://www.ibm.com/docs/en/db2/11.5?topic=hpo-replacing-existing-tivoli-sa-mp-managed-db2-instance-pacemaker-managed-hadr-db2-instance

https://www.triton.co.uk/automating-hadr-failovers-with-pacemaker/
https://www.triton.co.uk/configuring-db2-pacemaker-hadr-cluster-with-qdevice-in-aws/
https://www.triton.co.uk/db2-pacemaker-hadr-seamless-failover/
https://www.triton.co.uk/db2-hybrid-hadr-clusters/

https://www.idug.org/news/the-book-of-db2-pacemaker--chapter-1--red-pill-or-blue-pill
https://www.idug.org/news/the-book-of-db2-pacemaker--chapter-2-pacemaker-cluster--assemble
https://www.idug.org/news/the-book-of-db2-pacemaker--chapter-3pacemaker-resource-model-into-the-pacemaker-verse
https://www.idug.org/news/the-book-of-pacemaker---chapter-4-quorumania
https://www.idug.org/news/the-book-of-db2-pacemaker--chapter-5--pacemaker-configuration-parameters-across-the-db2-verse
https://www.idug.org/news/the-book-of-db2-pacemaker-chapter-6--pacemaker-log-everything-everywhere-all-at-once

https://aws.amazon.com/blogs/architecture/field-notes-set-up-a-highly-available-database-on-aws-with-ibm-db2-pacemaker/

Finally - The End! :-)

Any questions?
Discussion?
(we can take if offline: damir.wilder@triton.co.uk)

Thank you for your attention!