CCDUG

2024 Central Canada Db2 Tech Conference

Intro to Pacemaker, the cluster manager of the future!

Tharmiga Loganathan

IBM Canada Ltd.

Platform: Db2 LUW

Please note

- IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice and at IBM’s sole discretion.
- Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.
- The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.
- The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.
- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

NOTICE AND DISCLAIMER

AGENDA

01 INTRODUCTION

02 PACEMAKER VS. TSA

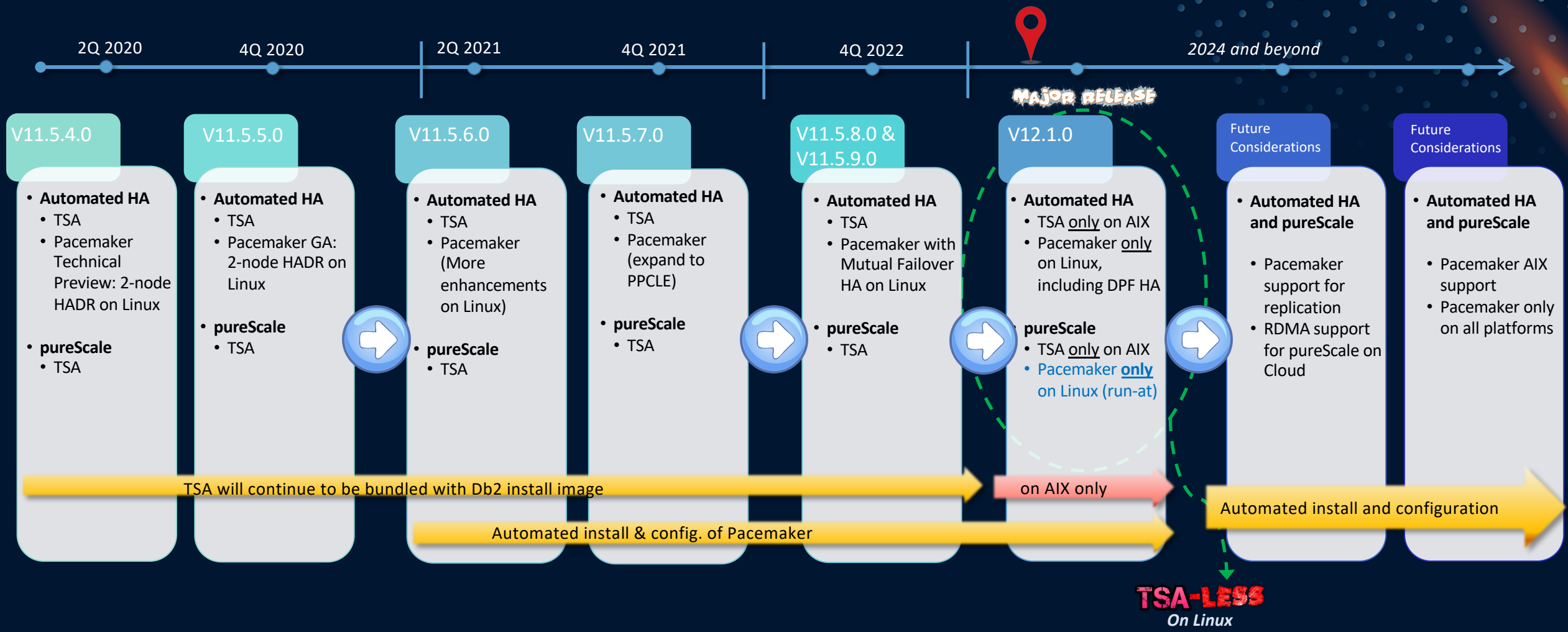
03 ARCHITECTURE

04 FAILOVER BEHAVIOUR

05 UP AND RUNNING

06 DEMO

Db2 Pacemaker Journey

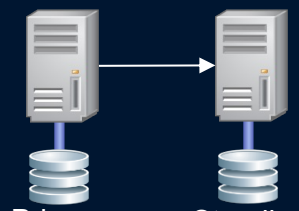


Note: Roadmap subjected to change

Db2 Cloud-Ready *Integrated* HA Topologies with Pacemaker – 10,000' view

V11.5.5.0

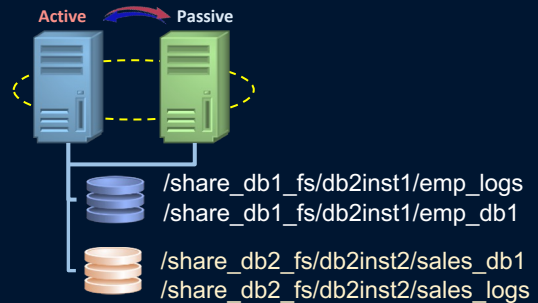
Single DB Partition (EE) with automated HADR



Today's Presentation

V11.5.8.0

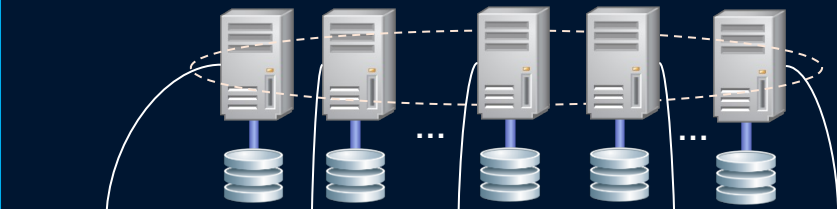
Mutual Failover (MF) (a.k.a. Active/Passive) automated HA with shared storage



Today's Demo

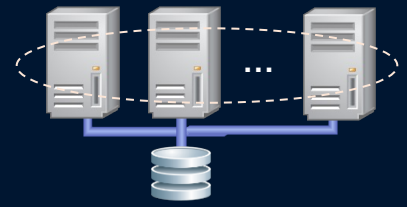
V12.1.0

Database Partitioning Feature (DPF) with automated HA (same site)



V12.1.0

pureScale Online 24x7x365 with automatic failover



Our vision with Pacemaker …

V11.5.8.0 - Announcement of Deprecation of TSA Support on Linux

- Target: TSA will no longer be bundled with Db2 on Linux in v12.1.0

Why Pacemaker over TSA ?

- 18+ years in industry as HA resource cluster manager
- Included by RHEL and SuSE as paid add-on HA package
- Open source: allow for future port to AIX
- Align with IBM Open-Source Strategy

Key driver for change

- Surge of requests for cloud support
- Lack of flexibility with TSA
- Need single solution for all OSes, architectures, form factors

Recovery Performance compared with TSA

HADR

- Dual Reboot - ~45%
- Standby Reboot - ~28%
- Software Failure Primary - ~33%
- Software Failure Standby - ~31%
- User initiated TAKEOVER - ~24%

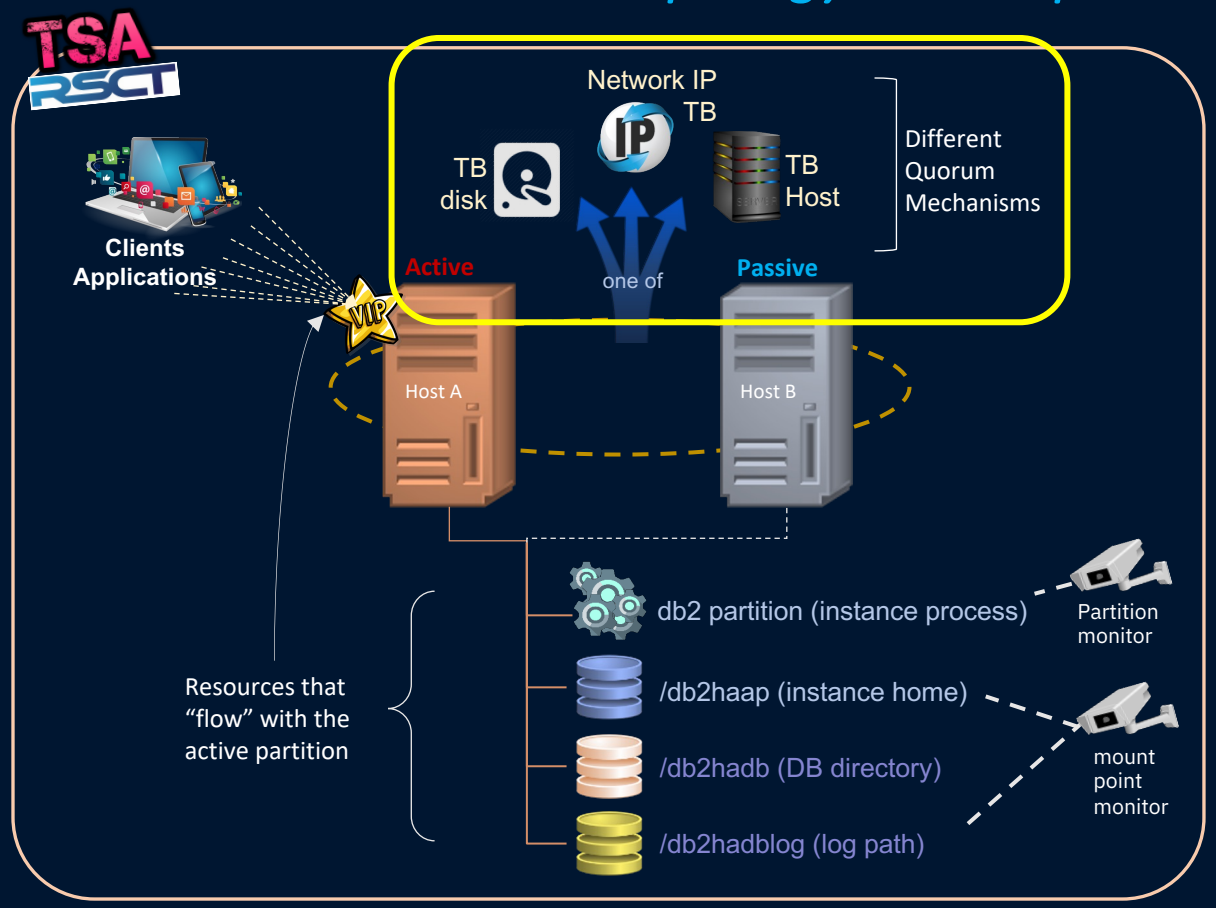
Mutual Failover

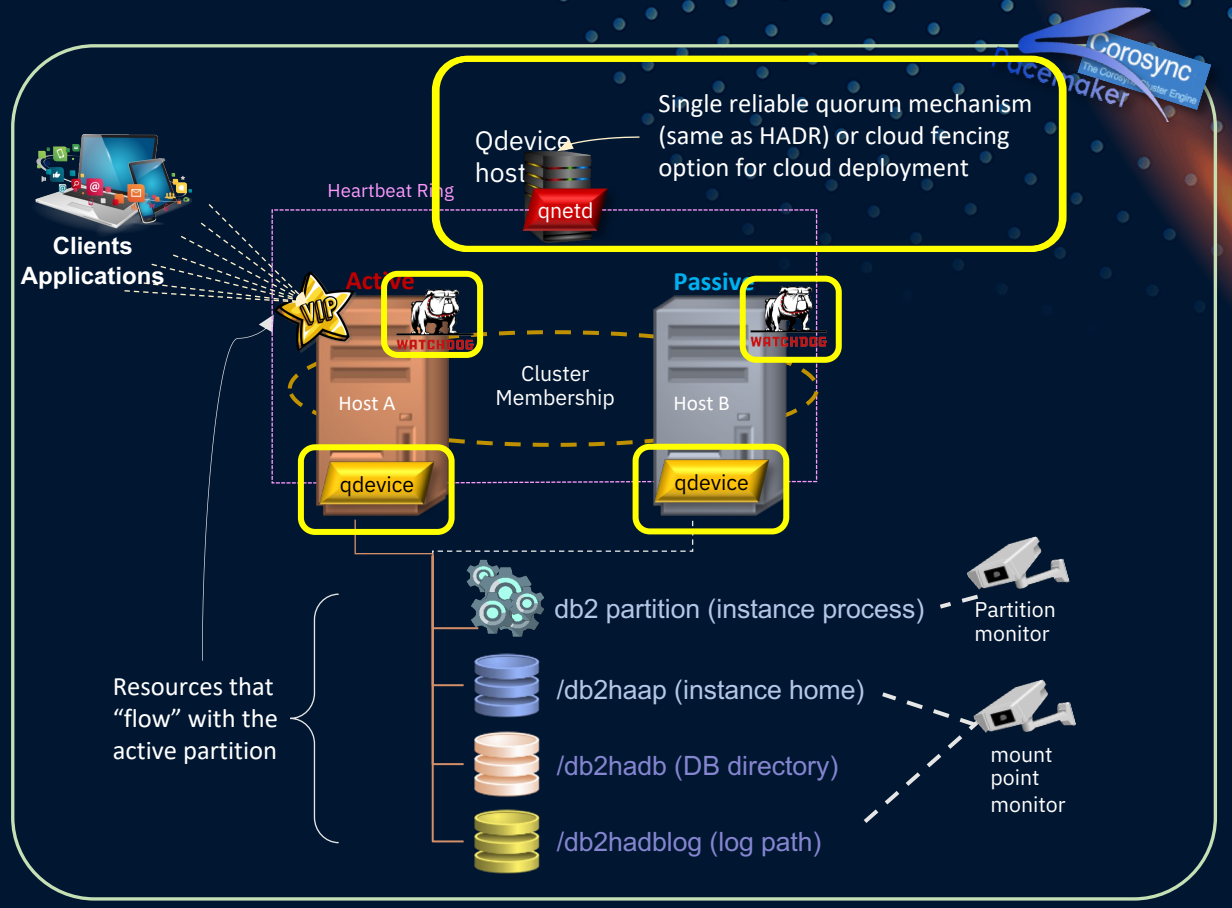
- Reboot - **~155% !!!**
- Software Failure - 29%
- User initiated TAKEOVER - ~50 seconds in Pacemaker, NOT implemented in TSA

Performance result measured from start of test scenario to transaction resumes

Note: *More improvements possible with more experimentation with various config parameters.*

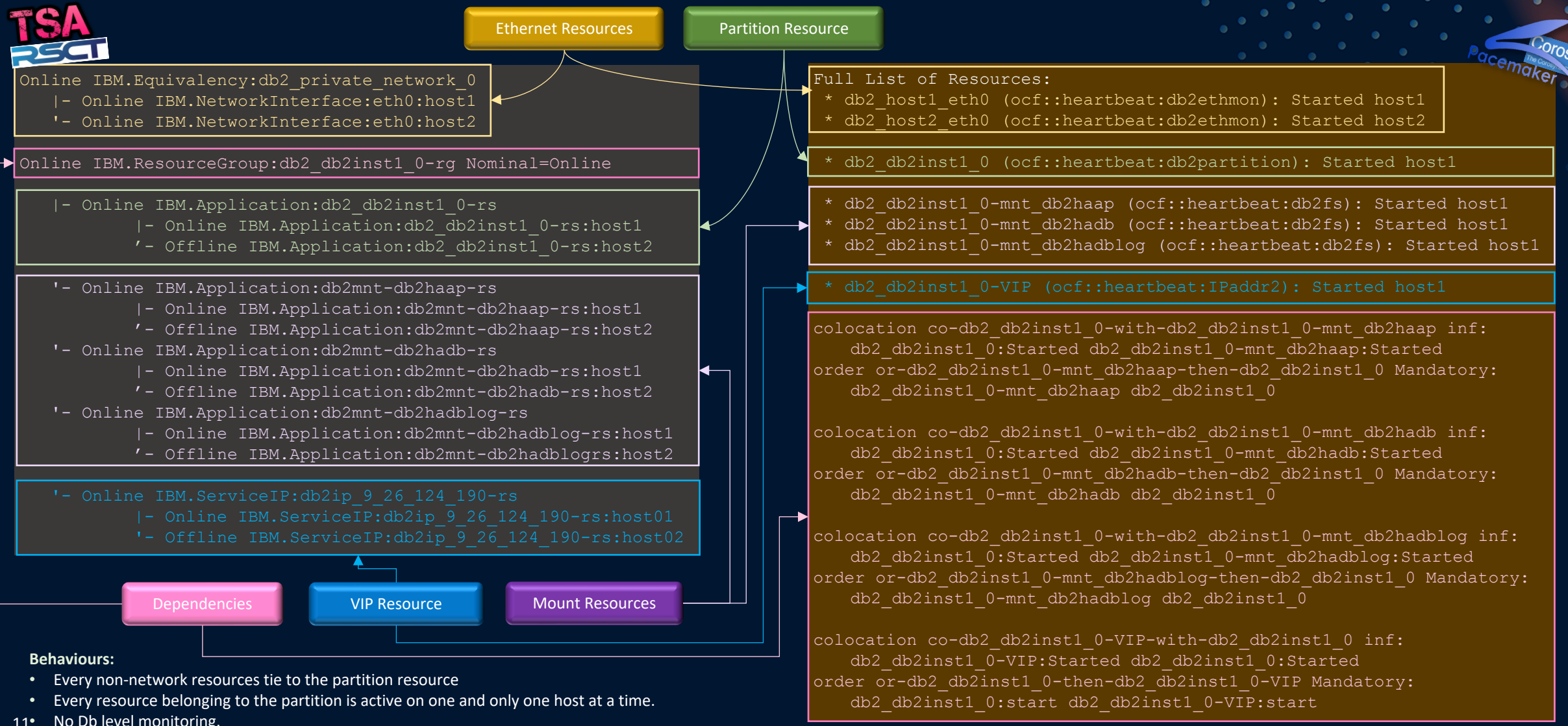
Old Vs New: *Topology & Components*





TB disk: requires SCISI 2/3 (not cloud-friendly), Network IP: not reliable, TB host: heavy handed	Quorum	Qdevice has lightweight non-H/W requirements, reliable, cloud-ready, multi-clusters, cross-arch.
Rely on RSCT Critical Resource Protection to reboot when a resource failed	Fencing	Leverage OS Software Watchdog to reboot host when a fencing action is required
Engine is integrated, but not every Db2 utility is cluster-aware	Engine Integration	Engine and utilities are cluster-aware such as db2relocatedb
No	Cloud-Readiness	Yes
db2haicu - Text-based menu driven utility	Cluster Management	db2cm – option-based command line utility

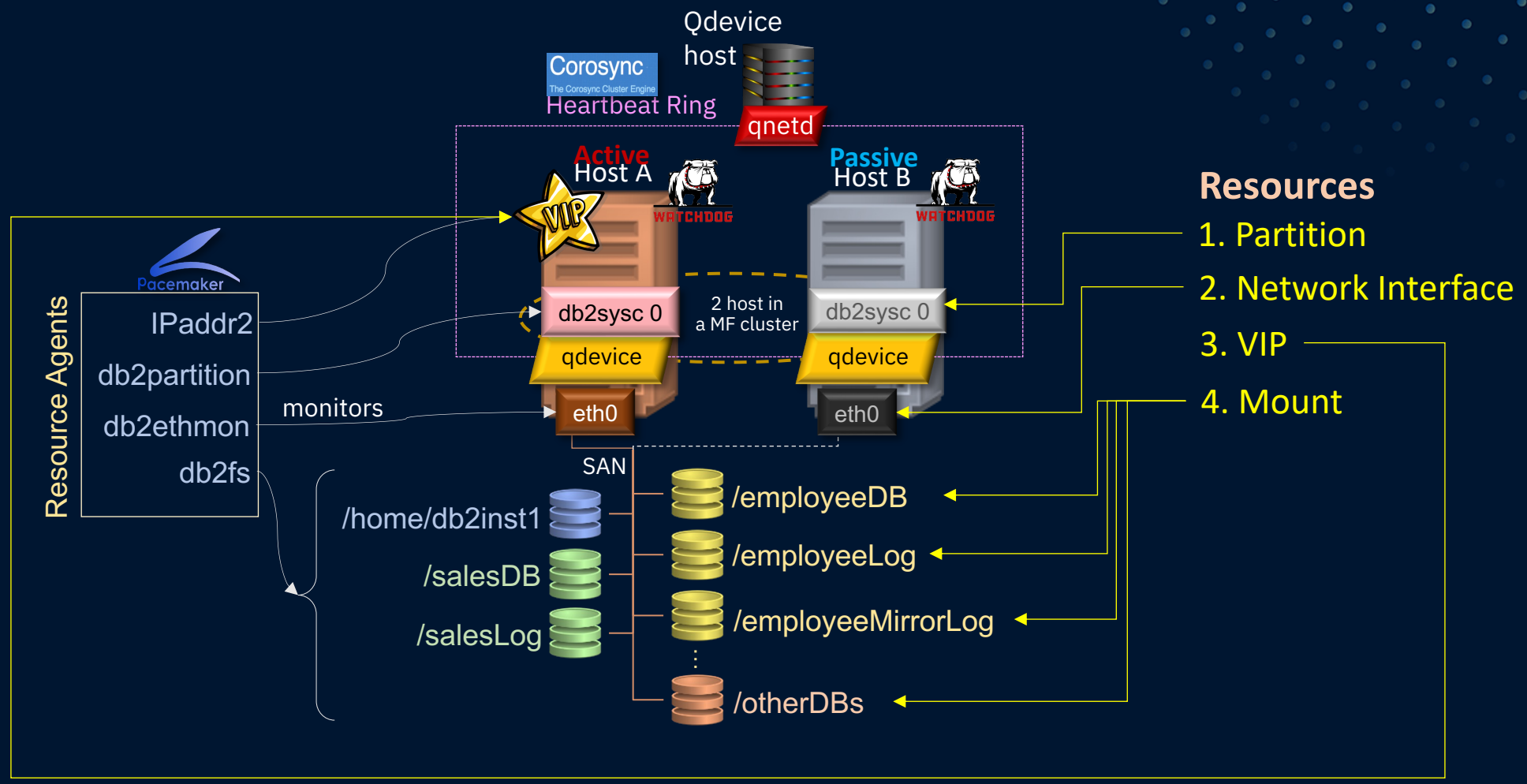
Old Vs New: *Resource Model*



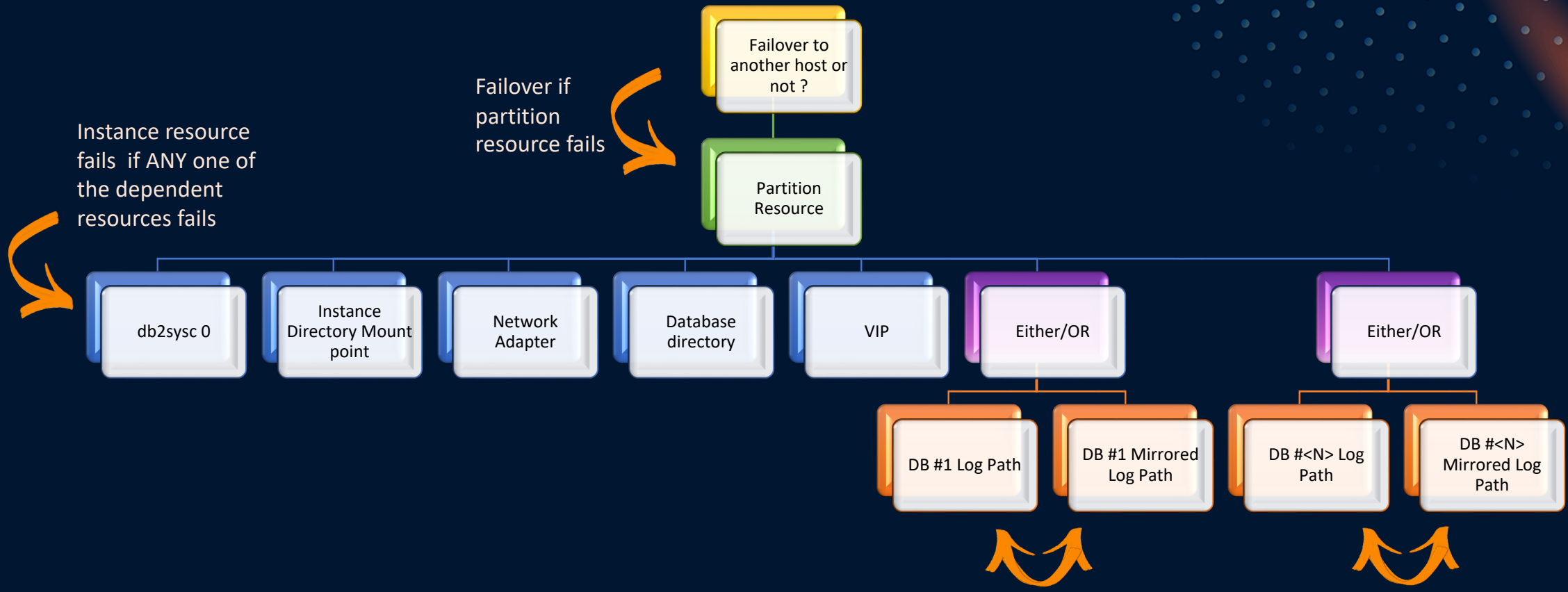
Behaviours:

- Every non-network resources tie to the partition resource
- Every resource belonging to the partition is active on one and only one host at a time.
- No Db level monitoring.

Architecture: Overview of Resource Models Components



Architecture: Resources Dependencies



Note: *A local restart of the resource is always the first action by Pacemaker before a failover will be attempted.*

Architecture: Resource Model: *Resource Agents*

- A set of shell scripts developed and supported by Db2 to perform actions on the resources defined in the resource model.
- A total of four resource agents installed in `/usr/lib/ocf/resource.d/heartbeat/`:

IPaddr2

- Monitor, start, and stop the VIP resource

db2partition

- Monitor, start, and stop a Db2 partition. Verifies the state of the partition (online/offline) and takes the required action to bring up the partition if needed. This is at instance level.

db2ethmon

- Monitor, start, and stop the network adapter resource.
- Same agent as for HADR

db2fs

- Monitor, start, and stop individual mount points.

Architecture: Inferring the cluster topology from "db2cm -list" - *Resource Model*

```
$ ./db2cm -list
      Cluster Status

HA configuration        = Mutual Failover

Domain information:
Domain name             = db2domain
Pacemaker version       = 2.1.2-4.db2pcmk.el8
Corosync version        = 3.1.6
Current domain leader   = lcars-srv-1
Number of nodes         = 2
Number of resources     = 5

Node information:
Name name               State
---------------         --------
lcars-srv-1             Online
lcars-srv-2             Online

Resource Information:

Resource Name           = db2_lcars-srv-1_eth0
  State                      = Online
  Managed                    = true
  Resource Type              = Network Interface
    Node                     = lcars-srv-1
    Interface Name           = eth0

Resource Name           = db2_lcars-srv-2_eth0
  State                      = Online
  Managed                    = true
  Resource Type              = Network Interface
    Node                     = lcars-srv-2
    Interface Name           = eth0

Resource Name           = db2_regress1_0
  State                      = Online
  Managed                    = true
  Resource Type              = Partition
  Instance                   = regress1
  Partition                  = 0
  Current Host               = lcars-srv-1

Resource Name           = db2_regress1_0-instmnt_db2hamf
  State                      = Online
  Managed                    = true
  Resource Type              = File System
  Device                     = "/dev/disk/by-
uuid/f3983a2b-fbf6-4b10-8032-0cecc2e044fb"
  Mount Point                = "/db2hamf"
  File System Type           = ext3
  Mount Options              = "rw,relatime"
  Current Host               = lcars-srv-1

Resource Name           = db2_regress1_0-mnt_logpath
  State                      = Online
  Managed                    = true
  Resource Type              = File System
  Device                     = "/dev/disk/by-
uuid/275f671a-d362-40d1-b4ab-ce16c34599db"
  Mount Point                = "/logpath"
  File System Type           = ext3
  Mount Options              = "acl,user_xattr,noauto"
  Current Host               = lcars-srv-1

Fencing Information:
  Configured
Quorum Information:
  Qdevice

Qdevice information
-------------------
Model:          Net
Node ID:        1
Configured node list:
    0           Node ID = 1
    1           Node ID = 2
Membership node list:   1, 2

Qdevice-net information
----------------------
Cluster name:db2domain
QNetd host:  lcars-disk:5403
Algorithm:   LMS
Tie-breaker: Node with lowest node ID
State:       Connected
```

Cluster node list

Network adapters

Active Partition

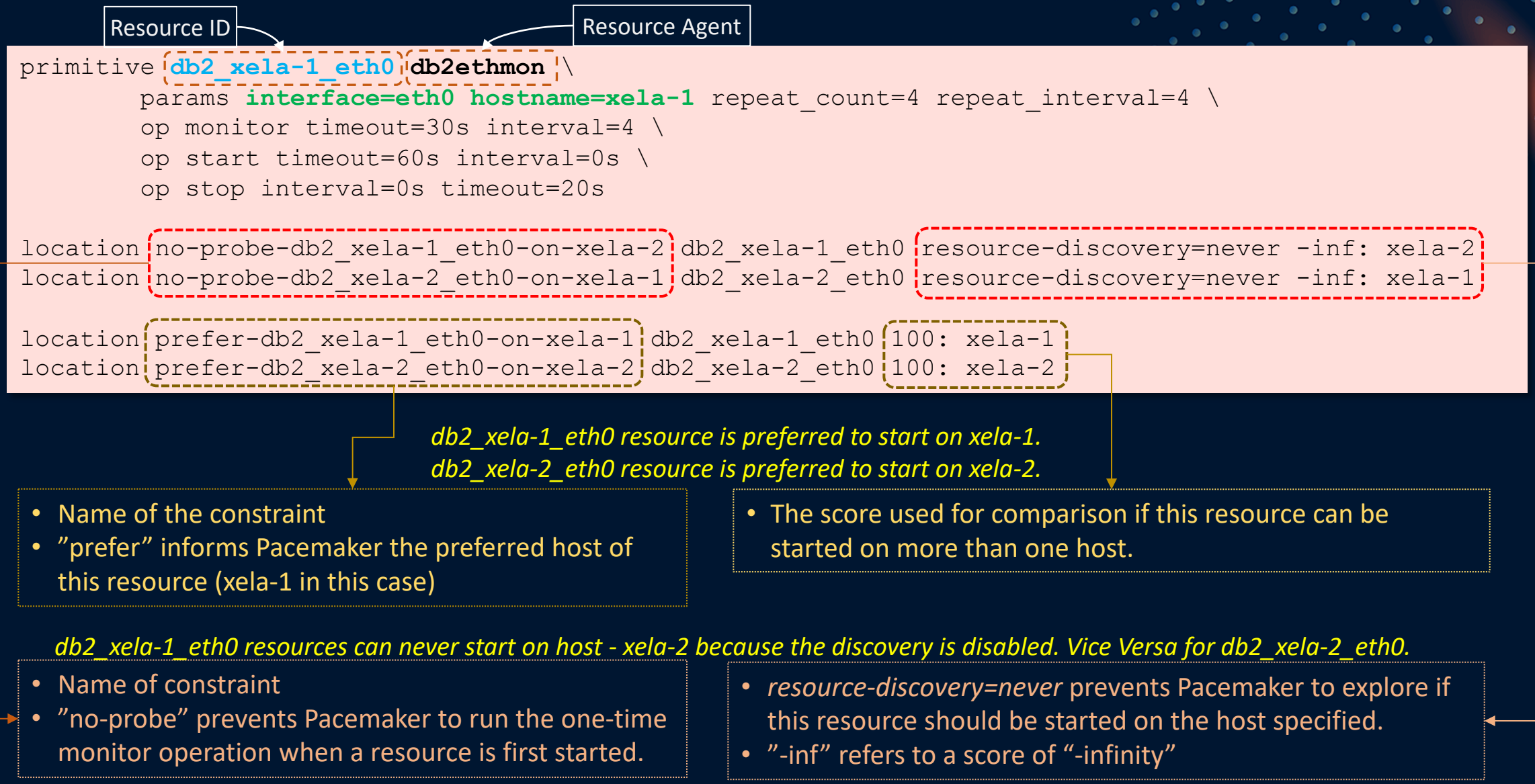
File Systems

Qdevice Quorum

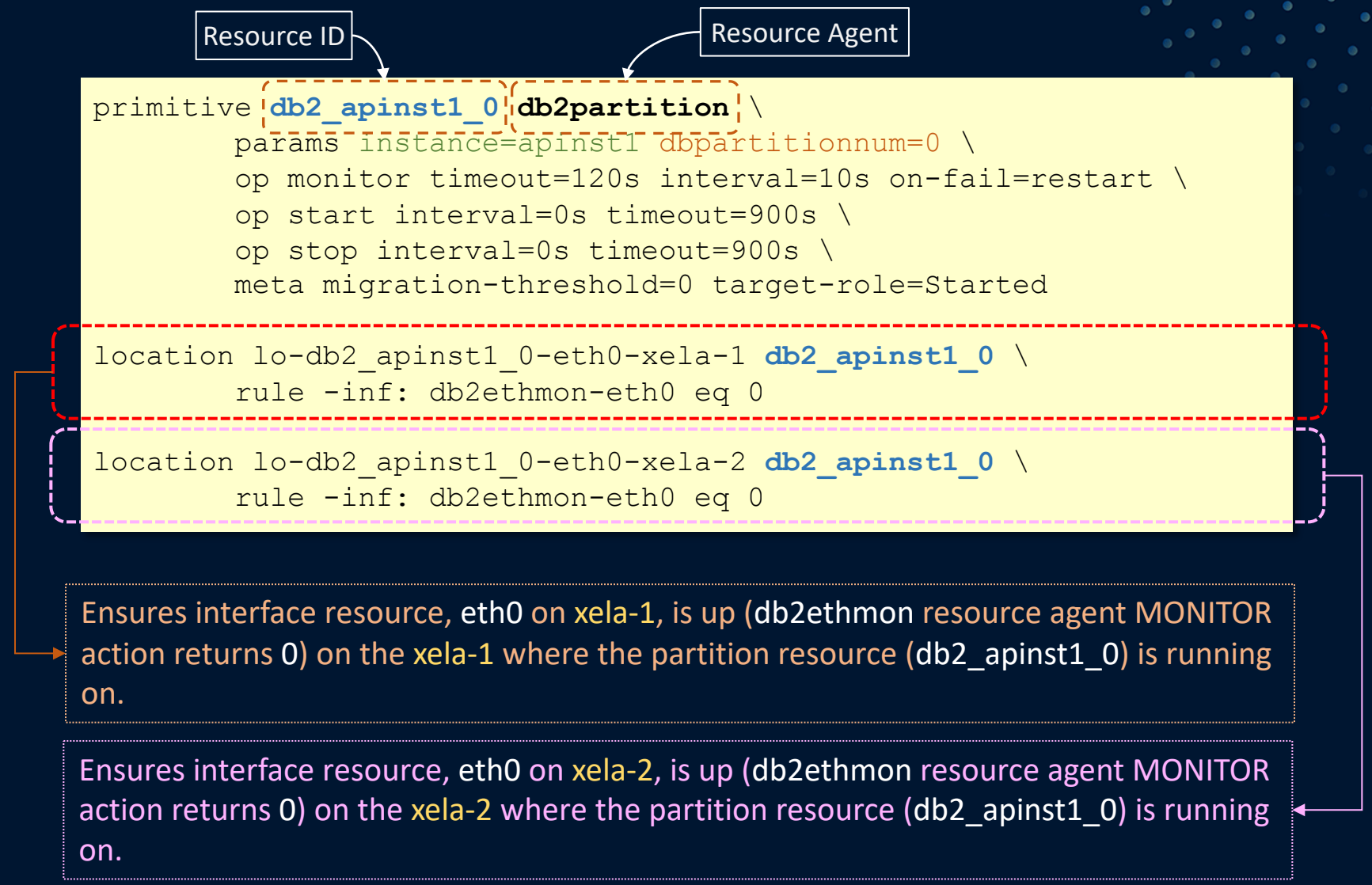
Cluster Topology in this example

- A MF cluster with 2 nodes
- Each node has 1 Ethernet adapter
- Partitions active on lcars-srv-1
- 2 shared FS: one for DB, one for log
- Quorum uses Qdevice with fencing configured

Architecture: Resource Model: *Network resource & constraints*

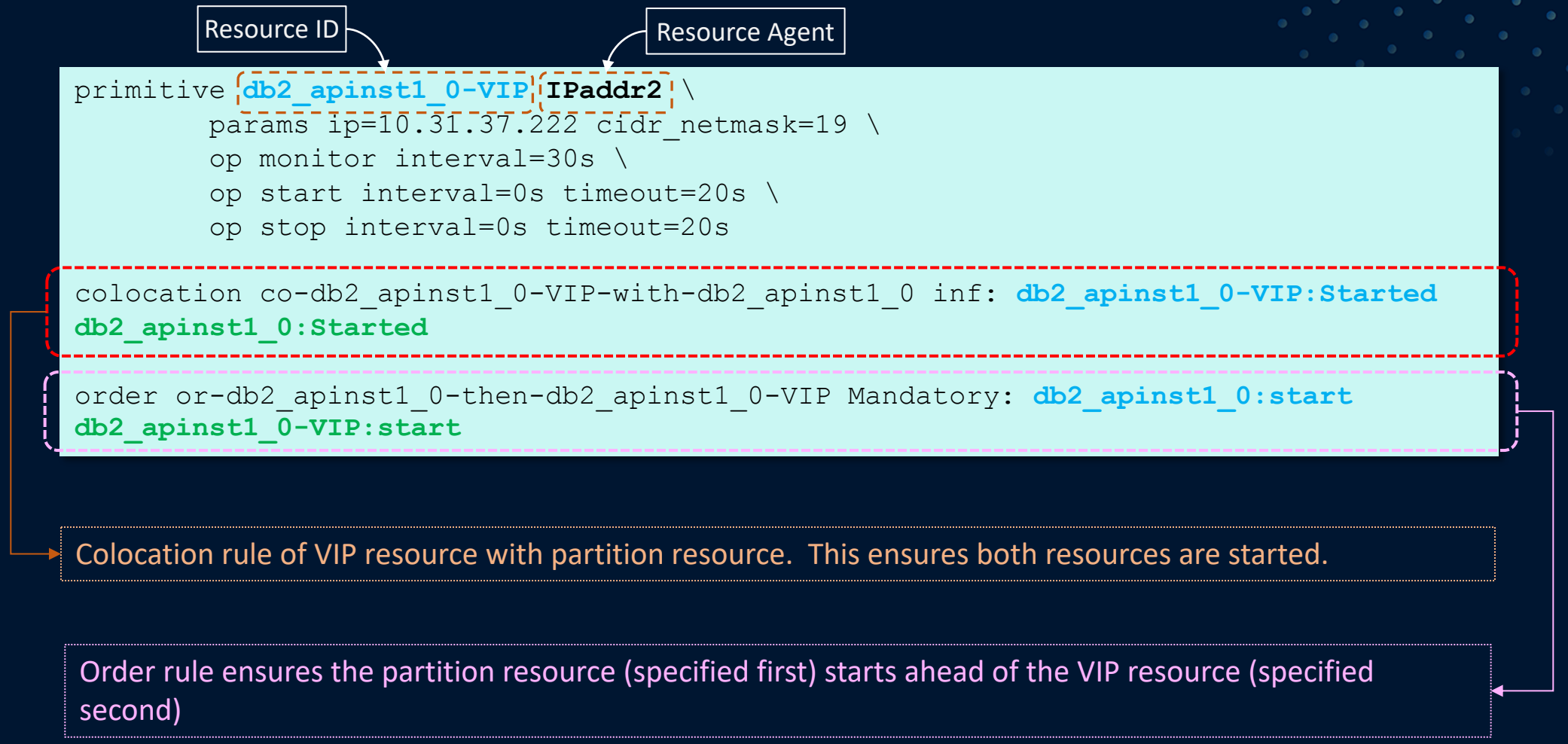


Architecture: Resource Model: *Partition resource & constraints*

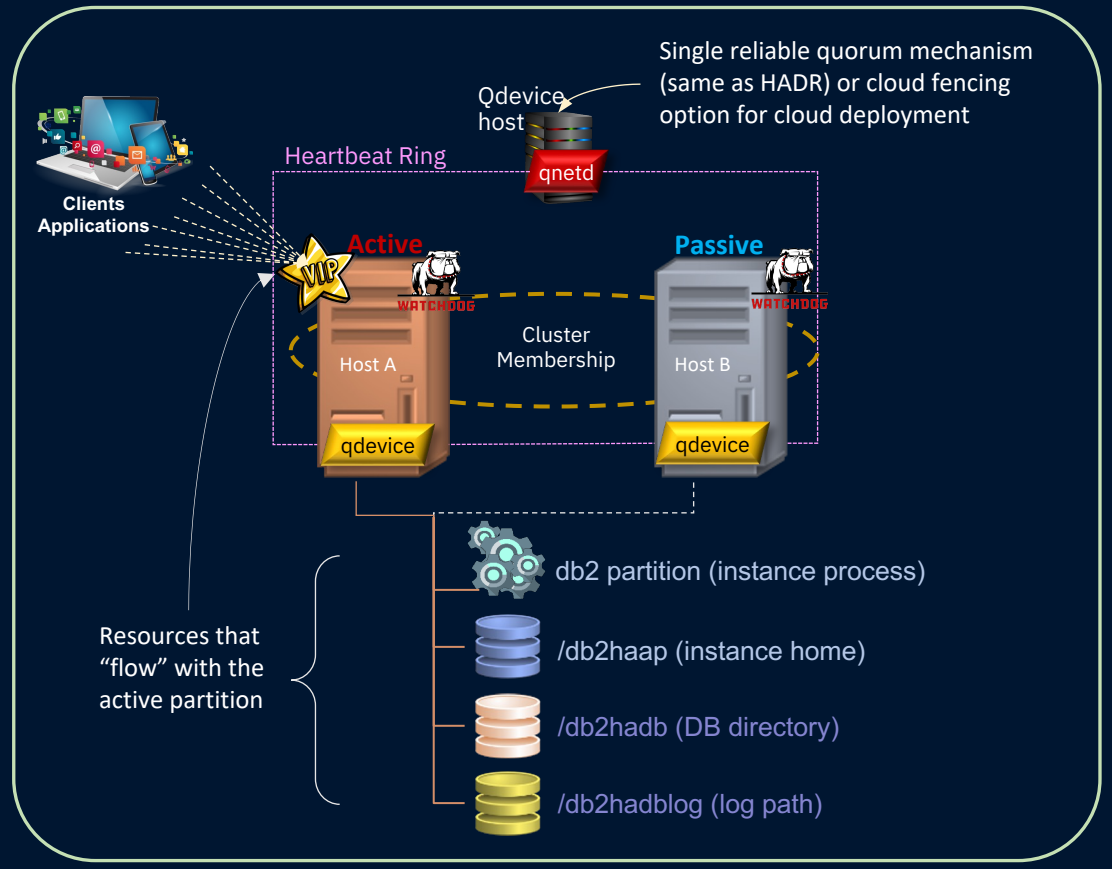


Architecture: Resource Model: *Mount resource & constraints*

Architecture: Resource Model: *Virtual IP resource and constraints*



Architecture: Resource Model: *Fencing with SBD + Qdevice*



The evicted node will be fenced via SBD service where the local watchdog reboots the node and won't be allowed to rejoin cluster until it can gain quorum.

Why is fencing mandatory with MF ?

- Shared storage
- Need to handle potential strayed process(es) in split brain scenario and host failure

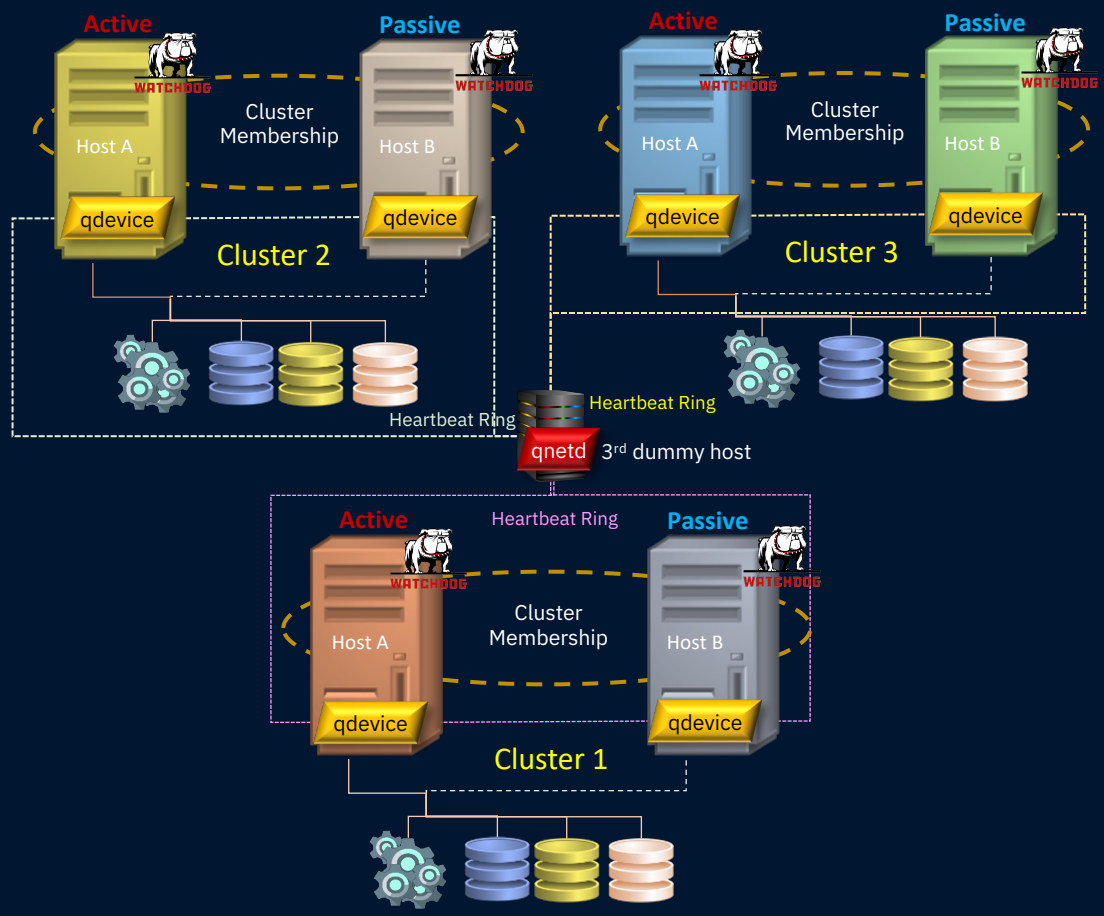
How does it work ?

- Diskless SBD (STONITH Block Device) provides a node fencing mechanism via reboots when quorum is lost without the need of shared disk
- SBD and watchdog (Hardware or Software) must be configured for fencing to work.
- Software watchdog setup is handled by Db2 install if hardware watchdog is not configured.

When is fencing triggered ?

1. Quorum Loss. Communication loss between cluster nodes. Qdevice votes for the node with the lowest node ID.
2. An attempt to stop a mount resource fails

Architecture: Resource Model: *Quorum with Qdevice (3rd host)*



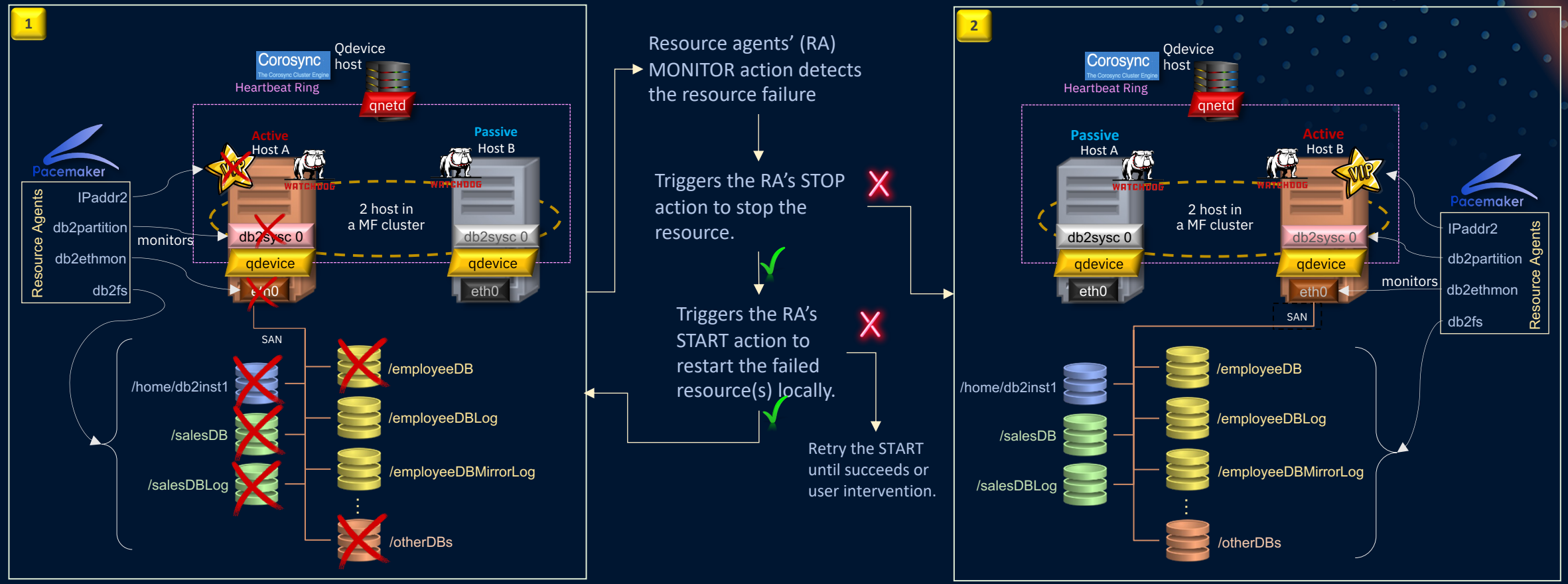
Setup

- ***qdevice*** - separate daemon from Pacemaker running on each cluster nodes
- ***qnetd*** - standalone daemon running on a 3rd host (not in cluster)
- TCP/IP connectivity among the 3 processes

3rd host detail

- Flexible in platform, architecture
- TCP/IP accessible from all hosts
- Possible to share with other Pacemaker clusters
  - e.g. use a RHEL host on Z for clusters nodes on POWER RHEL, x86 SLES, and Z with SLES.
- Small memory, disk footprint
  - Only need to install corosync-qnetd RPM
  - No need to install Db2 or Pacemaker
  - Not part of the Pacemaker cluster

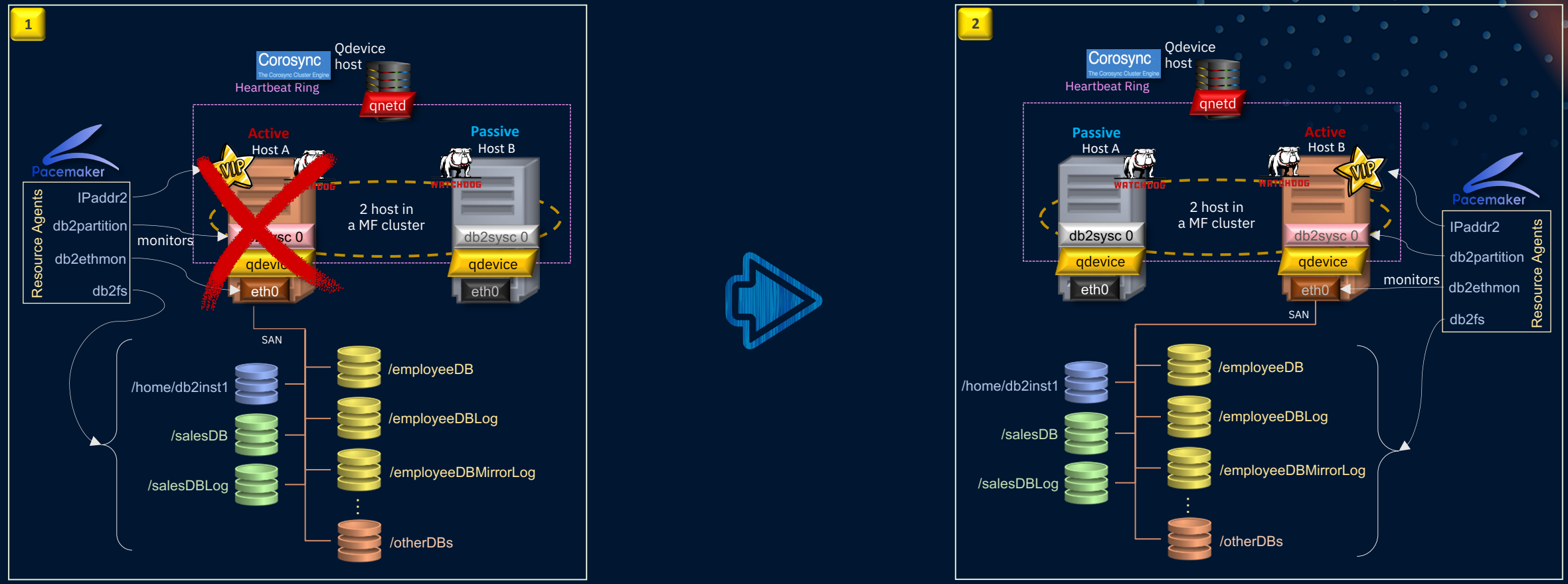
Failure Behaviour: *Resource Failure*



Results

- Resource failure leads to local restart of the resource
- Fencing only occurs if the failed resources failed to be stopped by Pacemaker.

Failure Behaviour: *Host Failure*



Results

- Corosync detects loss of quorum on HostA, notify Pacemaker to restart all resources on the other hosts

Up & Running End-to-End Overview

On-premises
- 2 hosts + 1 for Qdevice

Public Cloud
- 2 hosts
- Optional 3rd host for Qdevice

1. **Hosts Identification**
2. **Shared Storage Setup**
3. **User ID Creation (system level)**
4. **Host1: Db2 Install & Instance Creation**
5. **Unmount & Remount & Cleanup**
6. **Host2: Db2 Install & Instance Creation**
7. **Create Databases**
8. **HA Cluster Creation**

Db2 Instance Users

Cleanup all files on file system after remount

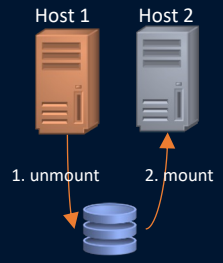
Run db2_install, db2icrt

Setup mirror log path as applicable

Example:
- SAN storage
- Local storage exported from 3rd host

Able to mount on both hosts

Pick any one host, run db2_install, db2icrt



- **db2cm -create -cluster -domain** … -host <host1> -publicEthernet … -host <host2> -publicEthernet …
- **db2cm -create -partition** <#> -instance …
- **db2cm -create -primaryVIP** … -partition <#> -instance …
- **db2cm -create -qdevice** <host>

Same as TSA

Replaces db2haicu in TSA

db2cm options

Up & Running

- db2cm -create -cluster -domain <name> -publicEthernet <NIC> -host <host1>
  -publicEthernet <NIC> -host <host2>
- db2cm -delete -cluster
- db2cm -create -partition <#> -instance <instname>
- db2cm -delete -partition <#> -instance <instname>
- db2cm -create -primaryVIP <IPv4 addr> -partition <#> -instance <instname>
- db2cm -delete -primaryVIP -partition <#> -instance <instname>
- db2cm -create -qdevice <host>
- db2cm -delete -qdevice
- db2cm -export <filename>
- db2cm -import <filename>

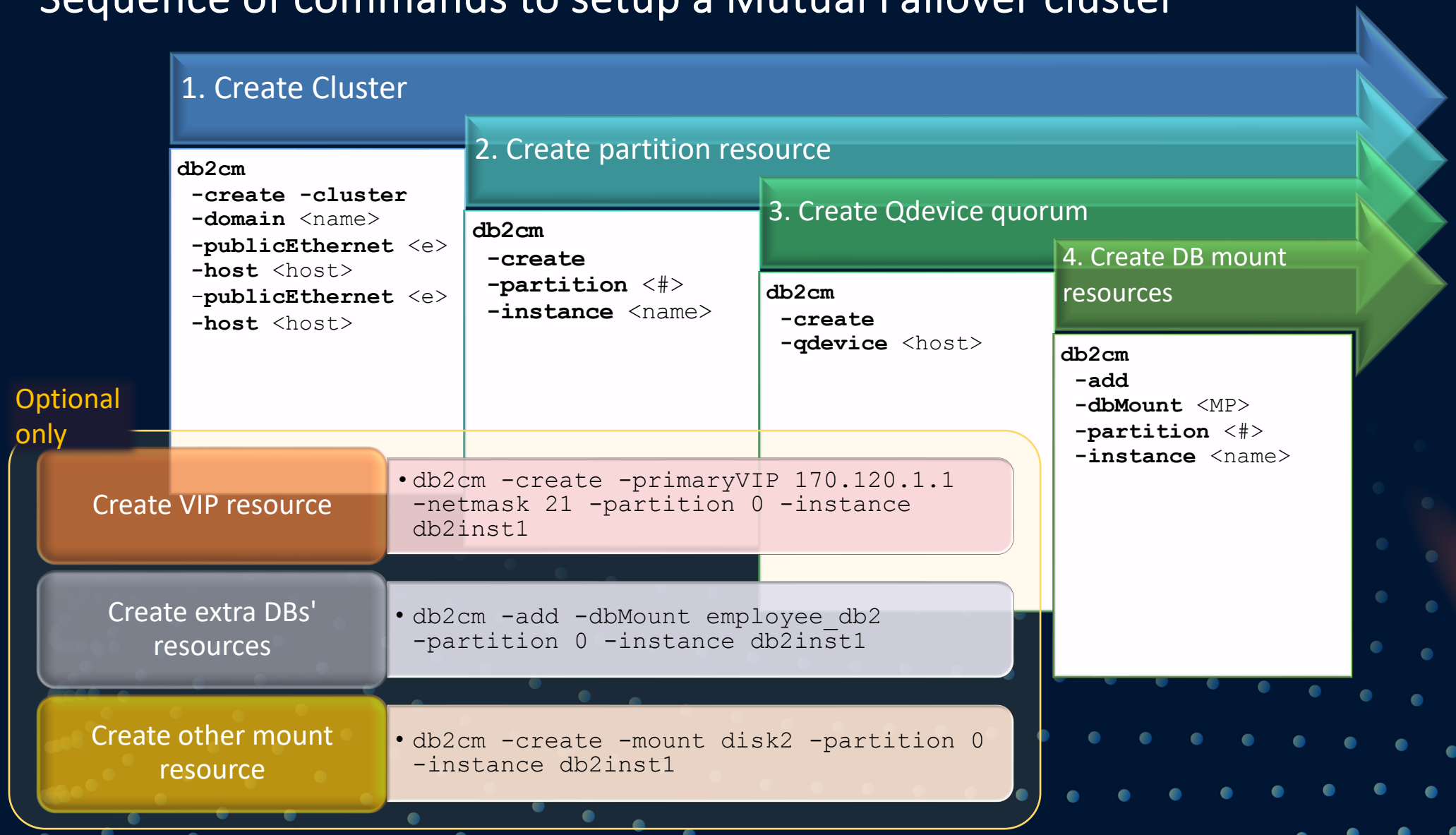
Maintenance

- db2cm -create -mount <mount point> -partition <#> -instance <instname>
- db2cm -delete -mount <mount point> -partition <#> -instance <instname>
- db2cm -add -dbMount <DB Name> -partition <#> -instance <instname>
- db2cm -remove -dbMount <DB Name> -partition <#> -instance <instname>
- db2cm -disable <-all | -partition <#> –instance <instname>>
- db2cm -enable <-all | -partition <#> –instance <instname>>
- db2cm -set -option mountMonitoring <mountPoint> -value <yes|no> -partition <#> -instance <instname>
- db2cm -move -partition <#> -instance <instname> -host <target hostname>

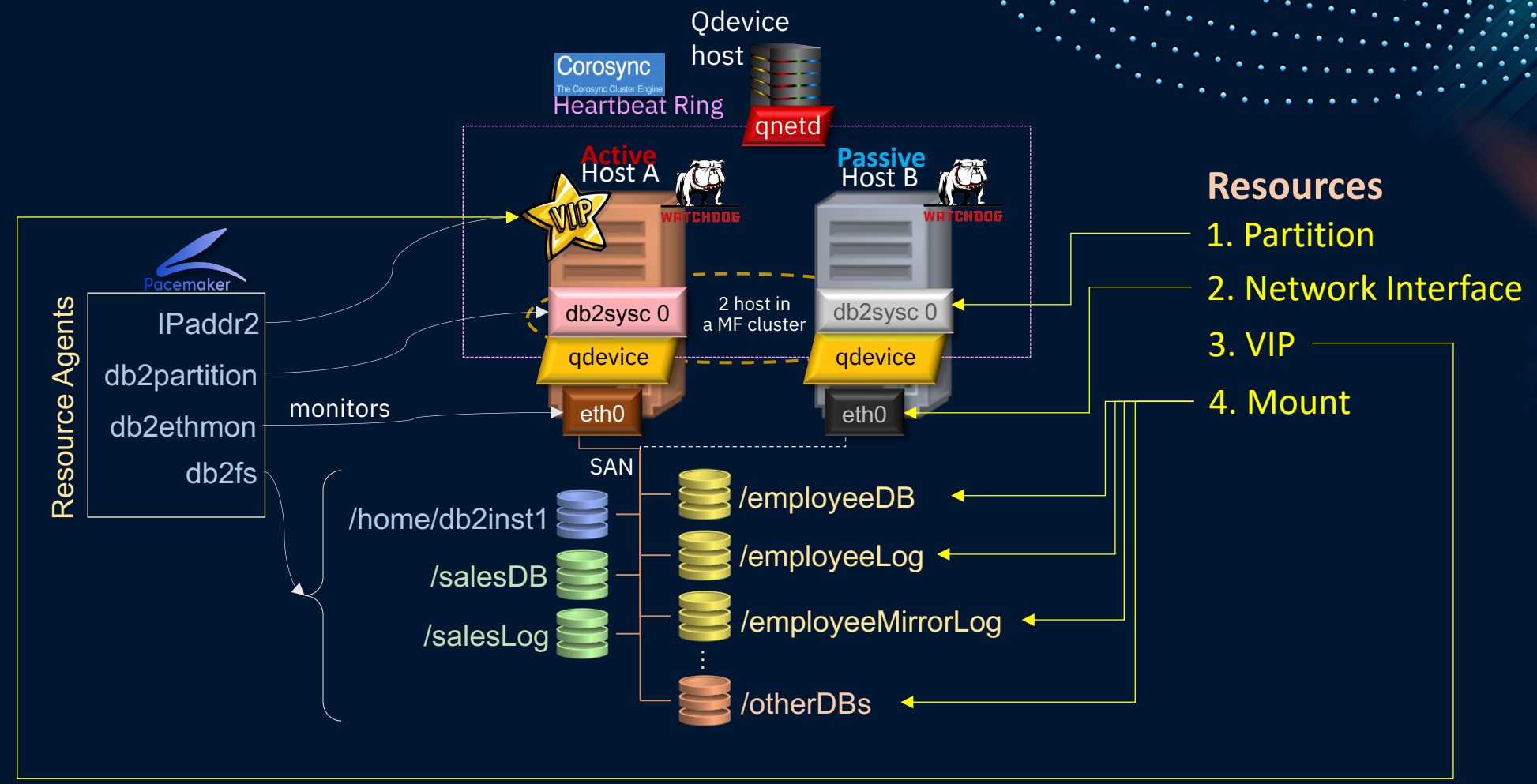
Monitor

- db2cm -list

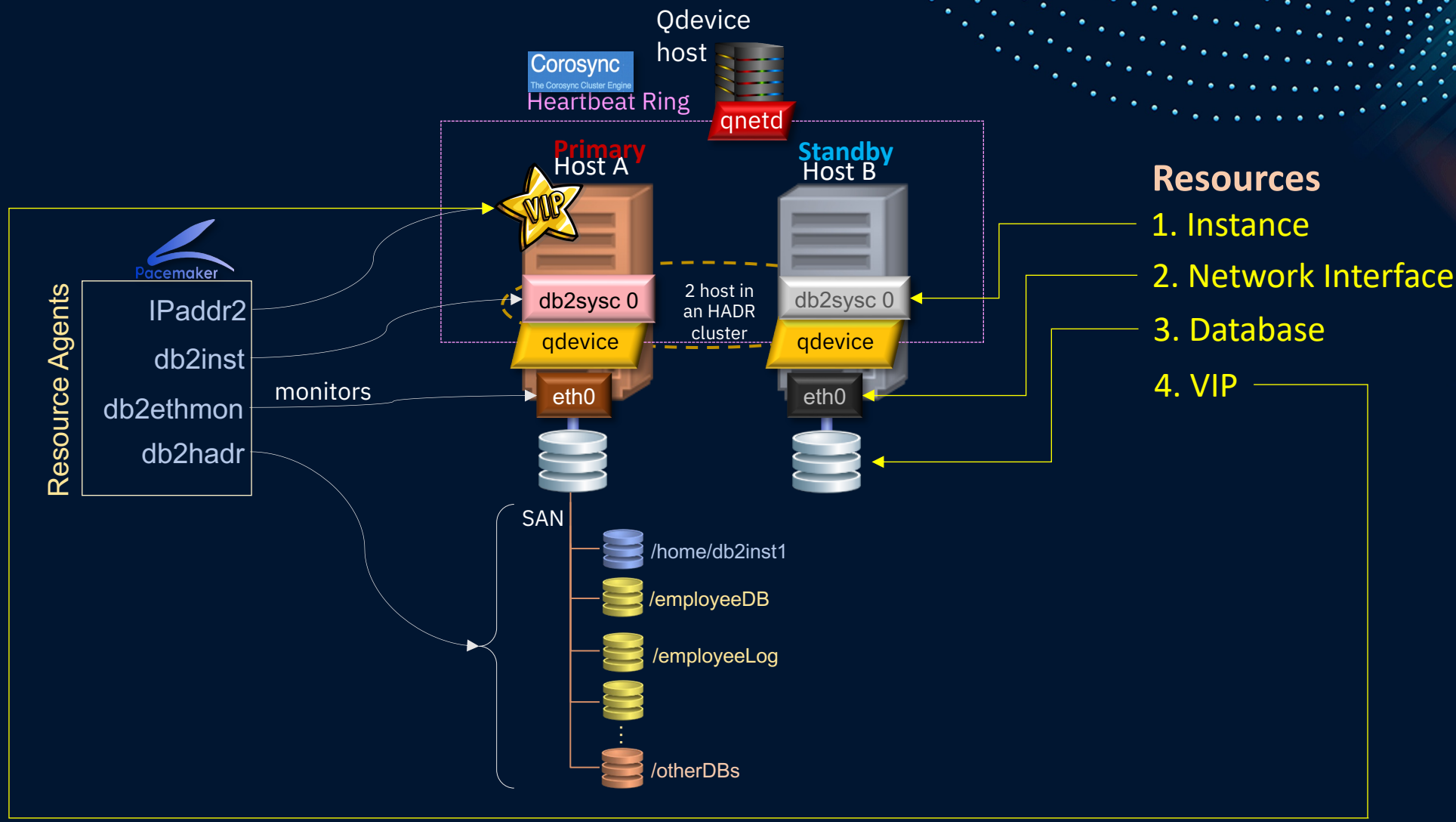
Sequence of commands to setup a Mutual Failover cluster



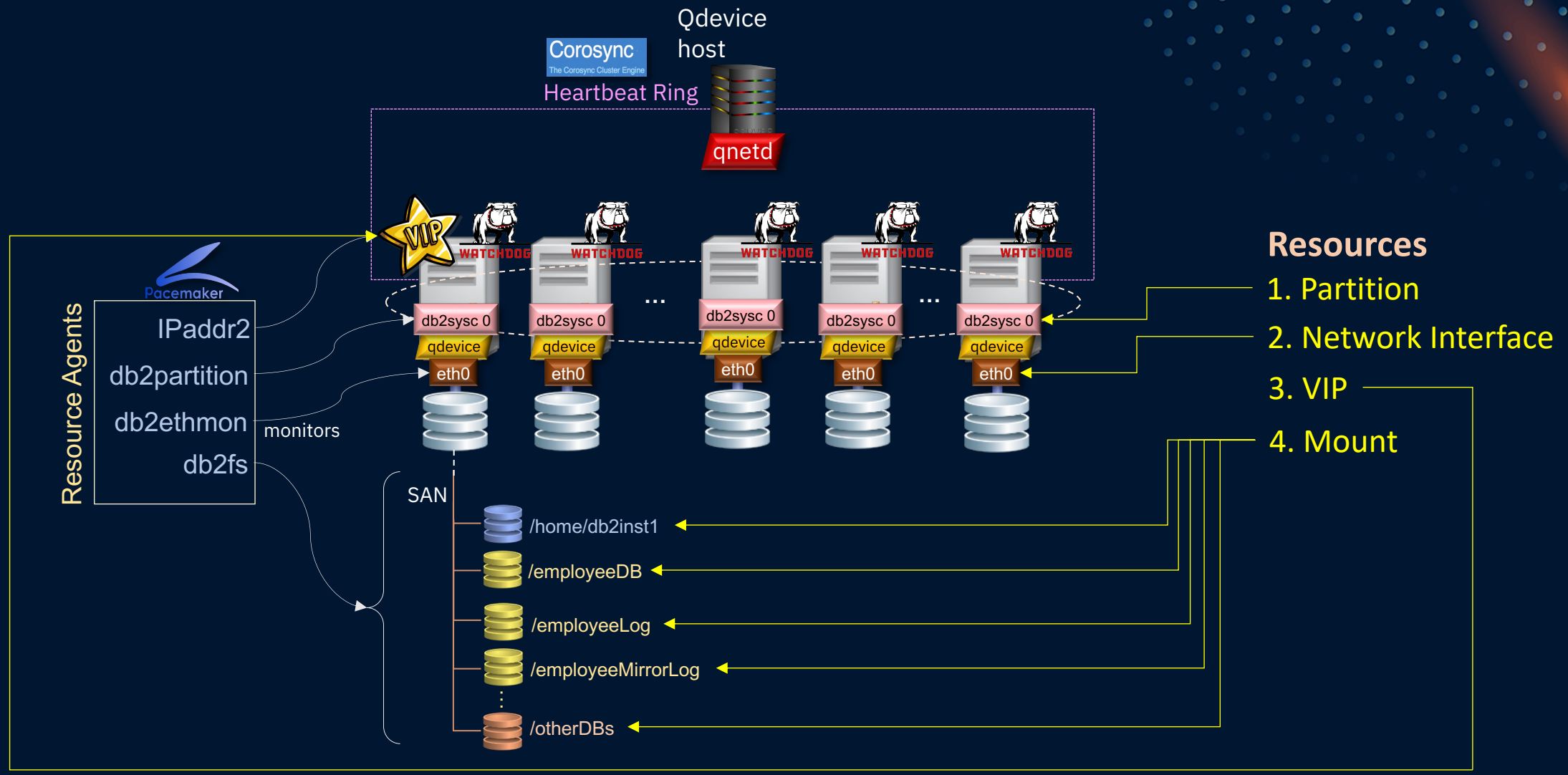
Mutual Failover



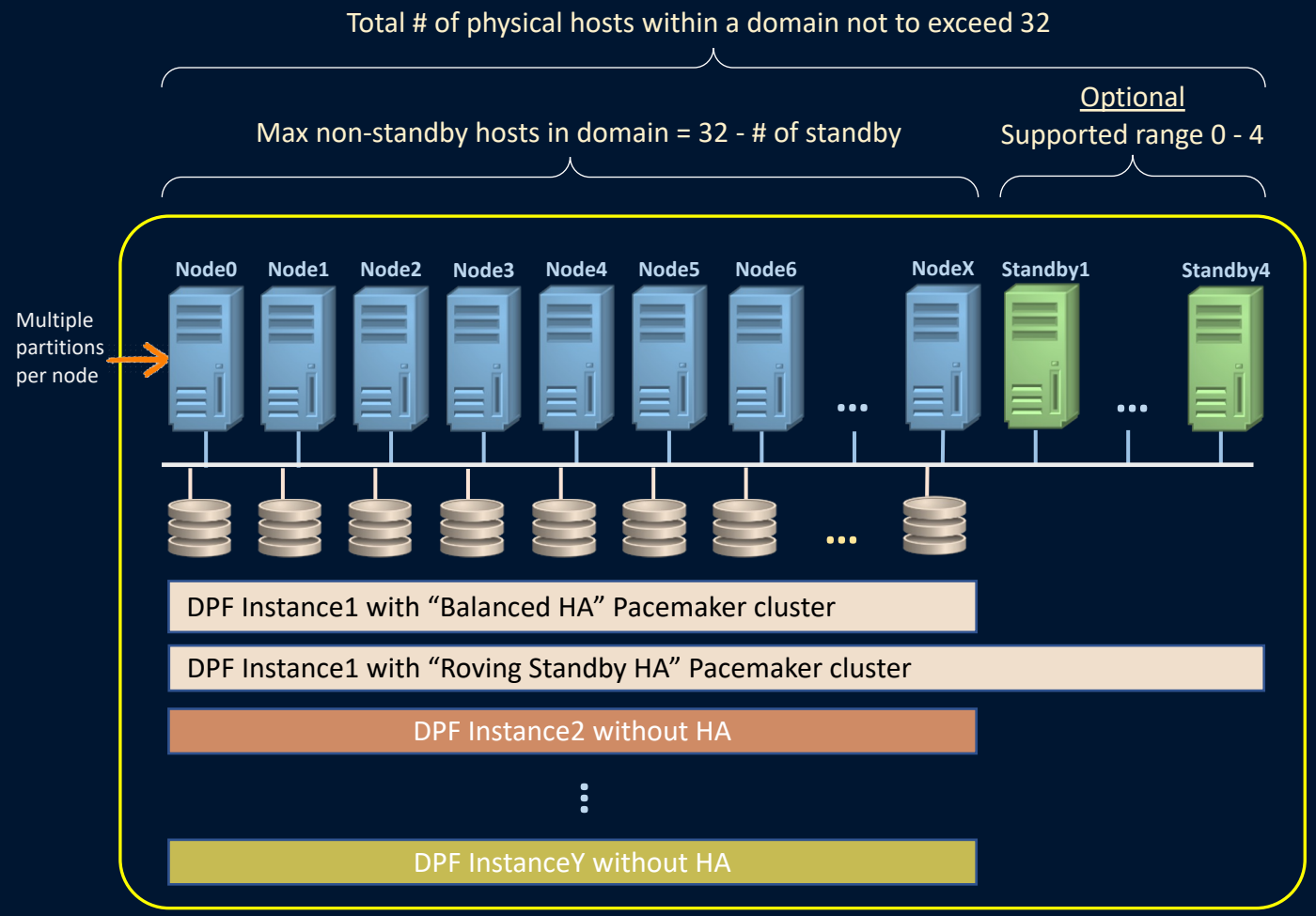
HADR


Qdevice
host
Corosync
The Corosync Cluster Engine
Heartbeat Ring
qnetd
Primary
Host A
Standby
Host B
VIP
Pacemaker
Resource Agents
IPaddr2
db2inst
db2ethmon
db2hadr
monitors
db2sysc 0
qdevice
eth0
2 host in
an HADR
cluster
db2sysc 0
qdevice
eth0
SAN
/home/db2inst1
/employeeDB
/employeeLog
/otherDBs
Resources
1. Instance
2. Network Interface
3. Database
4. VIP

Database Partitioning Feature (DPF)



Sneak Peak at DPF HA topology with Pacemaker



Single Pacemaker domain with one of the following failover policies:

1. Balanced HA – without standby host
2. Roving Standby HA - 1 to 4 standby host(s)
   - Provide up to 4 concurrent host failure

Multiple instances is supported but …

- Only one instance can have HA enabled.
- All instances can span across all hosts, but only the HA enabled instance can use the standbys

Max number partitions supported

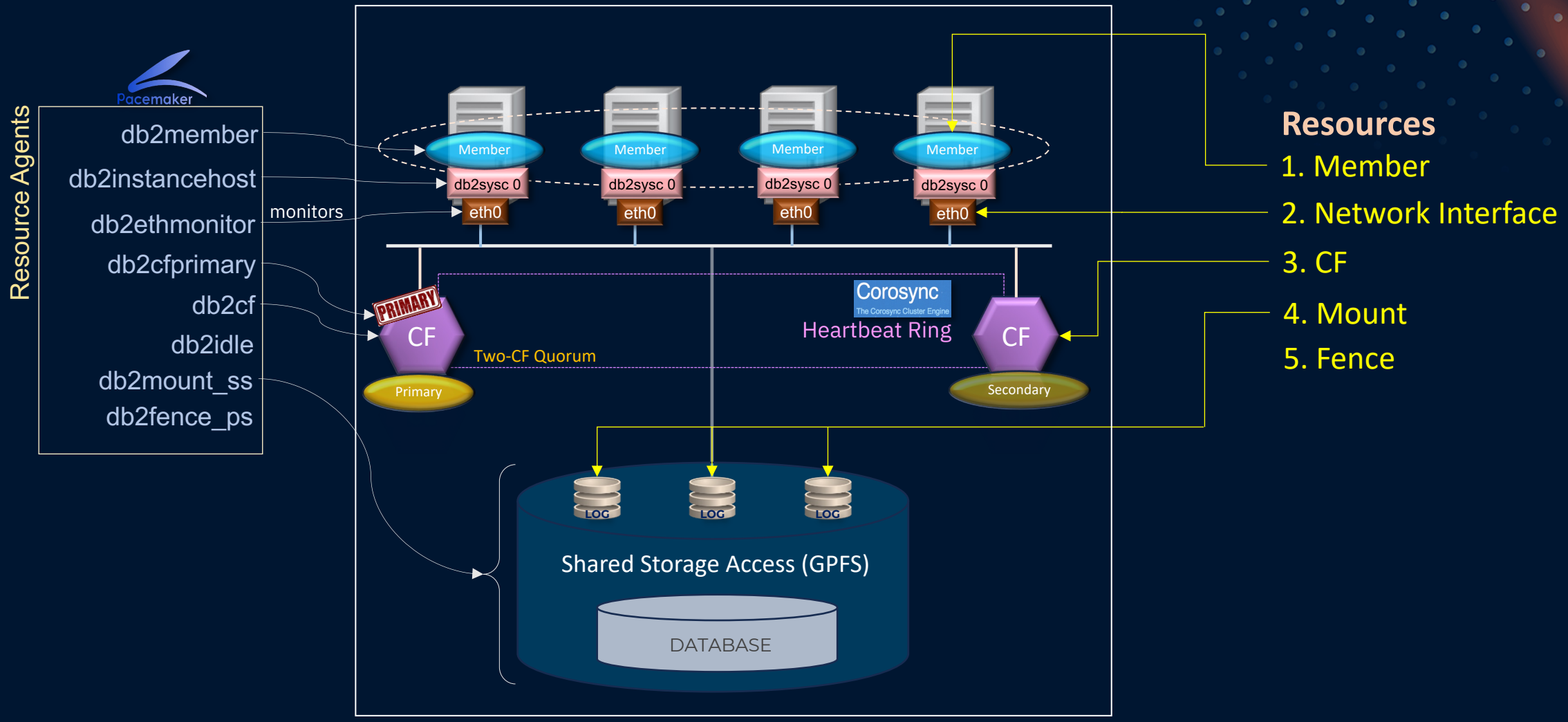
Using rule of thumb of 8 partitions per physical hosts:

- Balanced HA: 8 per host * 32 hosts = **256**
- Roving Standby HA: 8 per host * (32 - 4) hosts = **224**

Note:

- Higher number of partitions can explore deploying more partitions per host than 8 with proper H/W

pureScale


Pacemaker
Resource Agents
db2member
db2instancehost
db2ethmonitor
db2cfprimary
db2cf
db2idle
db2mount_ss
db2fence_ps
monitors
Member
db2sysc 0
eth0
PRIMARY
CF
Primary
Two-CF Quorum
Corosync
The Corosync Cluster Engine
Heartbeat Ring
CF
Secondary
LOG
Shared Storage Access (GPFS)
DATABASE
Resources
1. Member
2. Network Interface
3. CF
4. Mount
5. Fence

V12.1 Highlights with Pacemaker

DPF HA	• *THIRD* HA configuration with Pacemaker!
pureScale HA	• *FOURTH* HA configuration with Pacemaker!
pureScale on AWS	• Tech preview for pureScale on AWS with EFA
Pacemaker Refresh	• Upgrade to latest release – 2.1.8
Support newer OS level	• Validated on RHEL 9.4

DEMO

Showcasing DPF automated failover with Pacemaker

DPF Roving Standby – Failover Behaviour

DPF Balanced HA – Failover Behaviour

THANK YOU