High Availability database configurations

- High Availability database configurations
- Pre-requisites
- TWS DB Set-up
- DWC DB Set-up
- Post Configuration
- Operations and Management



Getting Started

Requirements:

- DB2 must be installed on 2 separate servers and the installations must be owned by same users on both servers and the version and fixpack level should be in sync across both servers.

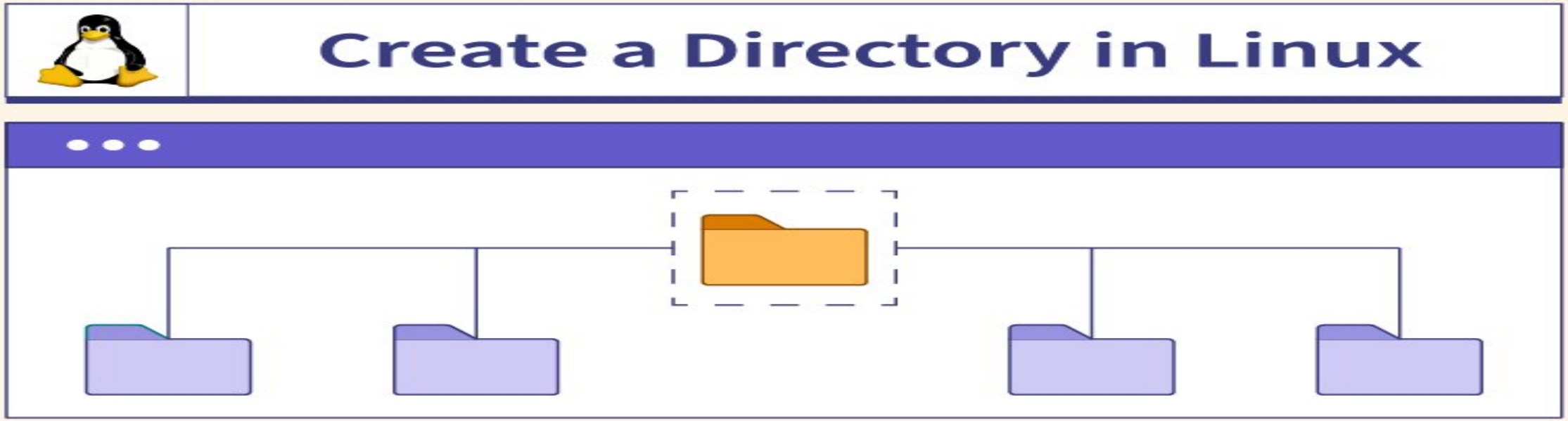
- File /etc/services MUST be edited with 2 new entries as seen below on BOTH servers that will be part of the HADR setup.

  You can use the ports of your choice instead of 50010(TWS)/50110(DWC) and ensure they are not conflicting with any other port/range:

  - DB2_HADR_1 50010/tcp
  - DB2_HADR_2 50110/tcp

TWS DB Set-up(Primary)

The first setup will be focused on the HADR setup for the TWS DB (or whatever you named the DB for the engine)
NOTE: On both servers, all work must be completed as the TWS DB instance owner, in this example the user is db2inst1



On Primary Node:

Create the log and backup directories (example directories listed below are within the db2inst1 home dir):

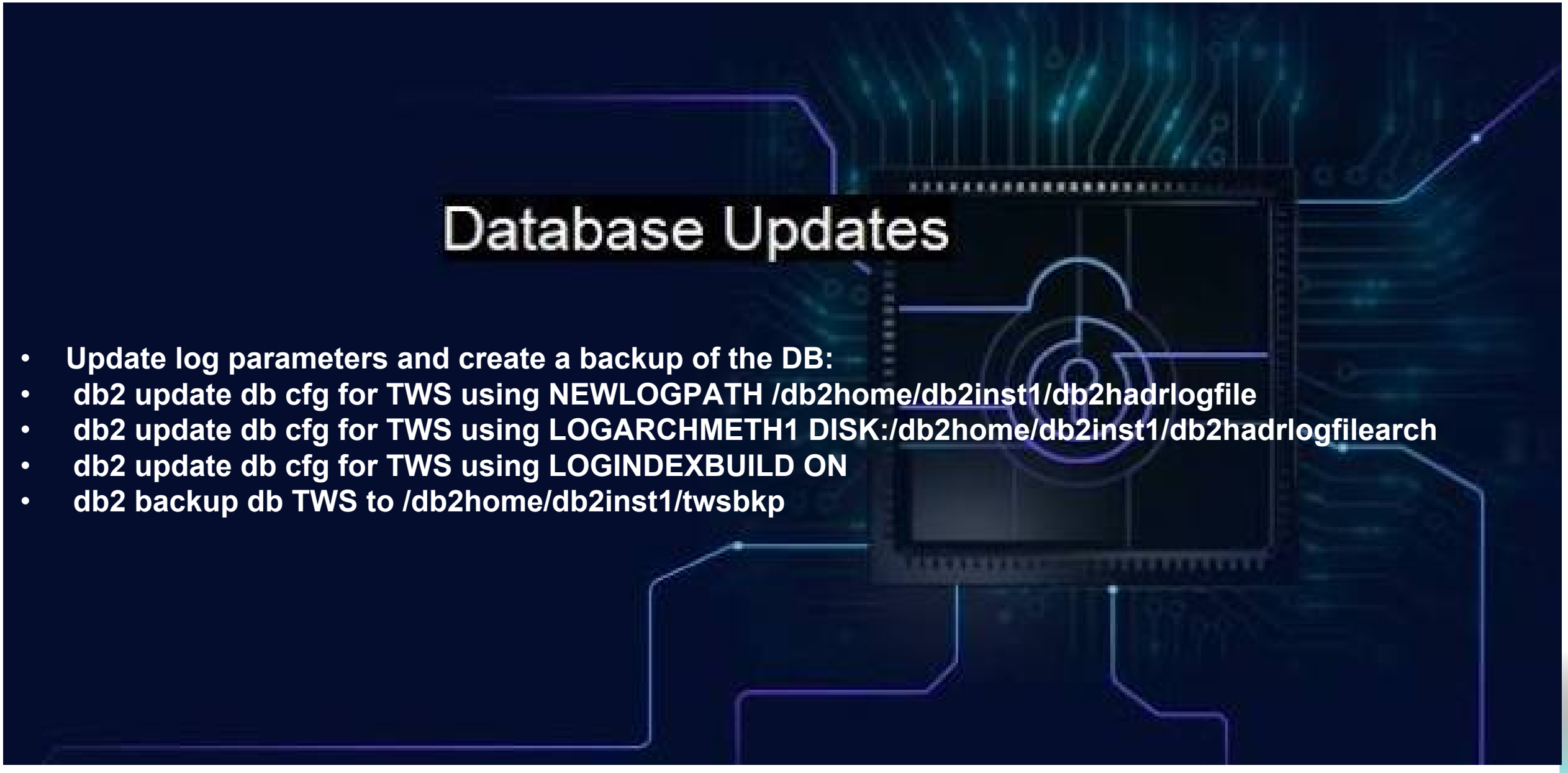
```
mkdir /db2home/db2inst1/db2hadrlogfile
mkdir /db2home/db2inst1/db2hadrlogfilearch
mkdir /db2home/db2inst1/twsbkp
```

TWS DB Set-up



Stop the engines appserver so there are no connections to the DB during the setup process

TWS DB Set-up



- Update log parameters and create a backup of the DB:
- db2 update db cfg for TWS using NEWLOGPATH /db2home/db2inst1/db2hadrlogfile
- db2 update db cfg for TWS using LOGARCHMETH1 DISK:/db2home/db2inst1/db2hadrlogfilearch
- db2 update db cfg for TWS using LOGINDEXBUILD ON
- db2 backup db TWS to /db2home/db2inst1/twsbkp

TWS DB Set-up (Standby)

- On Standby Node:
  - Restore backup, update HADR parameters and restart DB instance:
    - db2 restore db TWS from <PATH_OF_THE_DIR_WHERE_THE_BACKUP_FILE_IS> replace history file
    - db2 update db cfg for TWS using HADR_LOCAL_HOST <this_machine_hostname>



    - db2 update db cfg for TWS using HADR_LOCAL_SVC 50010
    - db2 update db cfg for TWS using HADR_REMOTE_HOST <other_machine_hostname>
    - db2 update db cfg for TWS using HADR_REMOTE_SVC 50010
    - db2 update db cfg for TWS using HADR_REMOTE_INST db2inst1
    - db2 update db cfg for TWS using HADR_SYNCMODE ASYNC
    - db2 update db cfg for TWS using HADR_TIMEOUT 30
    - db2 update db cfg for TWS using HADR_PEER_WINDOW 180
    - db2 UPDATE db cfg for TWS USING HADR_TARGET_LIST NULL
    - db2 update alternate server for db TWS using hostname <other_machine_host_name> port 50000
    - db2 db2stop force
    - db2 db2start

TWS DB Set-up (Primary)

- **On Primary Node:**
  - Update HADR parameters and restart DB instance:
    - db2 update db cfg for TWS using HADR_LOCAL_HOST <this_machine_hostname>
    - db2 update db cfg for TWS using HADR_LOCAL_SVC 50010



    - db2 update db cfg for TWS using HADR_REMOTE_HOST <other_machine_hostname>
    - db2 update db cfg for TWS using HADR_REMOTE_SVC 50010
    - db2 update db cfg for TWS using HADR_REMOTE_INST db2inst1
    - db2 update db cfg for TWS using HADR_SYNCMODE ASYNC
    - db2 update db cfg for TWS using HADR_TIMEOUT 30
    - db2 update db cfg for TWS using HADR_PEER_WINDOW 180
    - db2 UPDATE db cfg for TWS USING HADR_TARGET_LIST NULL
    - db2 update alternate server for db TWS using hostname <other_machine_host_name> port 50000
    - db2 db2stop force
    - db2 db2start

Starting HADR Services TWS DB



On Standby Node:

- db2 start hadr on db TWS as standby
- db2pd -db TWS -hadr (check for the following):
  - Connected

Starting HADR Services TWS DB

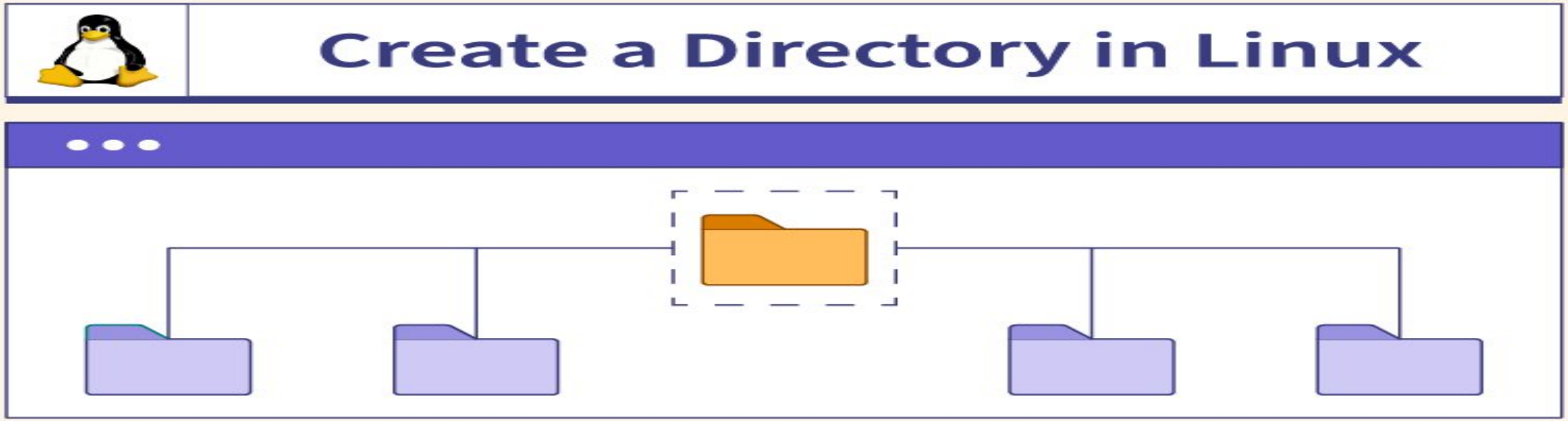


On Primary Node:

- db2 start hadr on db TWS as primary
- db2pd -db TWS -hadr (check for the following):
  - db2 update db cfg for TWS using HADR_LOCAL_HOST <this_machine_hostname>

Setup for DWC DB

The next setup will be focused on the HADR setup for the DWC DB (or whatever you named the DB for the console)
NOTE: On both servers, all work must be completed as the DWC DB instance owner, in this example the user is db2inst2



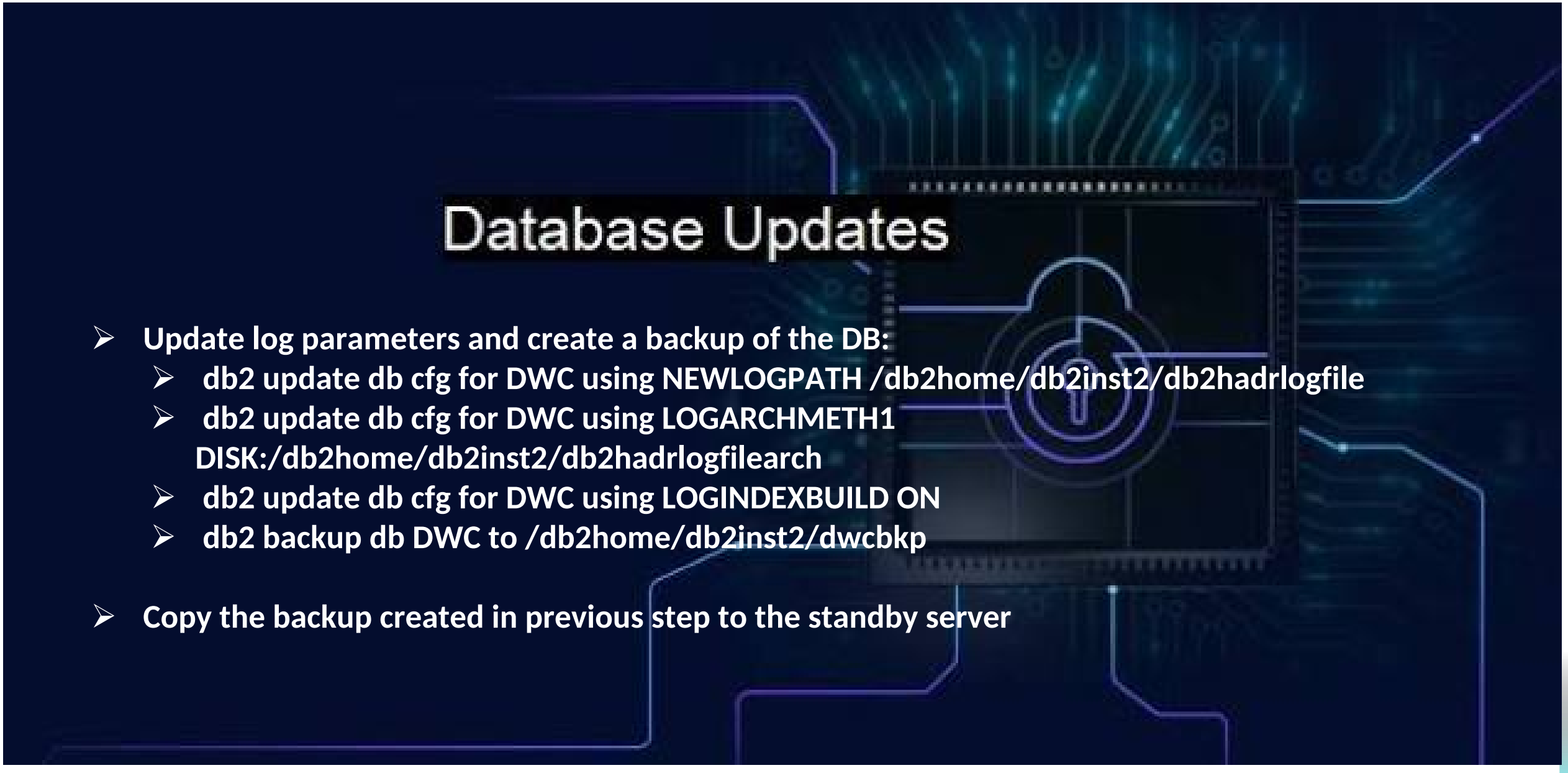
- **<u>On Primary Node</u>:**
  - Create the log and backup directories (example directories listed below are within the db2inst1 home dir):
    - mkdir /db2home/db2inst2/db2hadrlogfile
    - mkdir /db2home/db2inst2/db2hadrlogfilearch
    - mkdir /db2home/db2inst2/dwcbkp

Setup for DWC DB



Stop the consoles appserver so there are no connections to the DB during the setup process

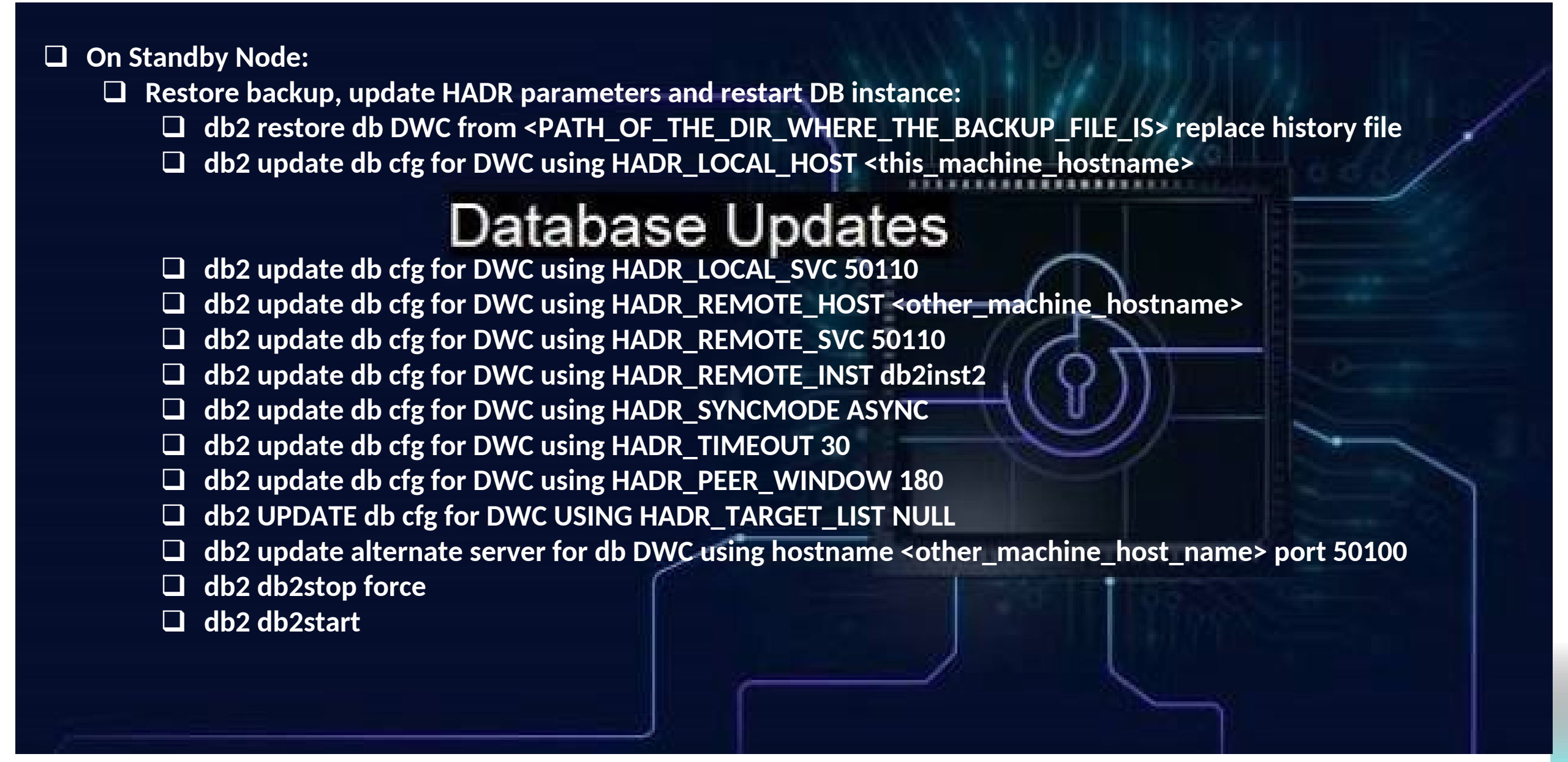
Setup for DWC DB



- Update log parameters and create a backup of the DB:
  - db2 update db cfg for DWC using NEWLOGPATH /db2home/db2inst2/db2hadrlogfile
  - db2 update db cfg for DWC using LOGARCHMETH1 DISK:/db2home/db2inst2/db2hadrlogfilearch
  - db2 update db cfg for DWC using LOGINDEXBUILD ON
  - db2 backup db DWC to /db2home/db2inst2/dwcbkp
- Copy the backup created in previous step to the standby server

Setup for DWC DB (Standby)

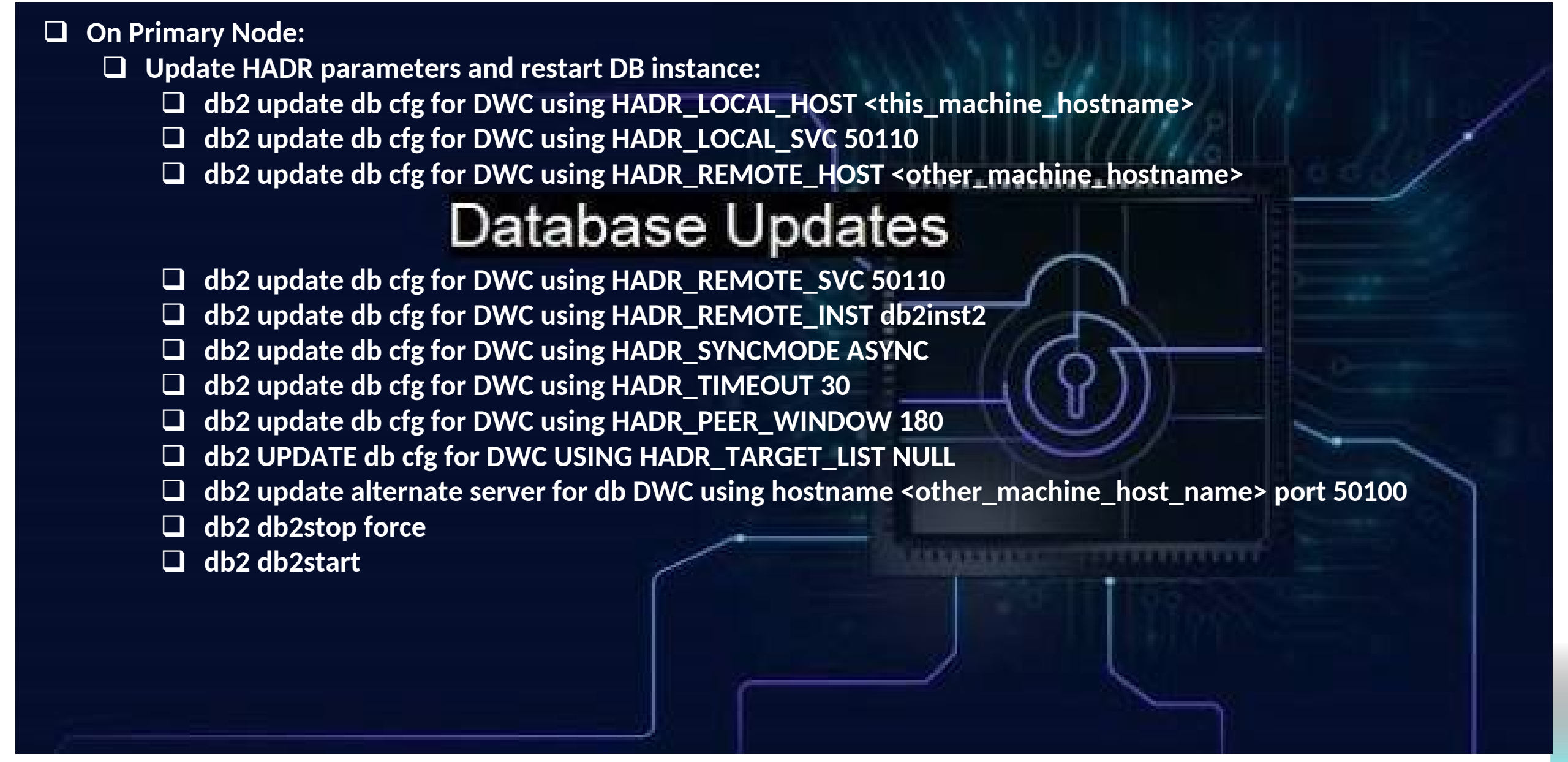
- On Standby Node:
  - Restore backup, update HADR parameters and restart DB instance:
    - db2 restore db DWC from <PATH_OF_THE_DIR_WHERE_THE_BACKUP_FILE_IS> replace history file
    - db2 update db cfg for DWC using HADR_LOCAL_HOST <this_machine_hostname>



    - db2 update db cfg for DWC using HADR_LOCAL_SVC 50110
    - db2 update db cfg for DWC using HADR_REMOTE_HOST <other_machine_hostname>
    - db2 update db cfg for DWC using HADR_REMOTE_SVC 50110
    - db2 update db cfg for DWC using HADR_REMOTE_INST db2inst2
    - db2 update db cfg for DWC using HADR_SYNCMODE ASYNC
    - db2 update db cfg for DWC using HADR_TIMEOUT 30
    - db2 update db cfg for DWC using HADR_PEER_WINDOW 180
    - db2 UPDATE db cfg for DWC USING HADR_TARGET_LIST NULL
    - db2 update alternate server for db DWC using hostname <other_machine_host_name> port 50100
    - db2 db2stop force
    - db2 db2start

Setup for DWC DB (Primary)

- On Primary Node:
  - Update HADR parameters and restart DB instance:
    - db2 update db cfg for DWC using HADR_LOCAL_HOST <this_machine_hostname>
    - db2 update db cfg for DWC using HADR_LOCAL_SVC 50110
    - db2 update db cfg for DWC using HADR_REMOTE_HOST <other_machine_hostname>



    - db2 update db cfg for DWC using HADR_REMOTE_SVC 50110
    - db2 update db cfg for DWC using HADR_REMOTE_INST db2inst2
    - db2 update db cfg for DWC using HADR_SYNCMODE ASYNC
    - db2 update db cfg for DWC using HADR_TIMEOUT 30
    - db2 update db cfg for DWC using HADR_PEER_WINDOW 180
    - db2 UPDATE db cfg for DWC USING HADR_TARGET_LIST NULL
    - db2 update alternate server for db DWC using hostname <other_machine_host_name> port 50100
    - db2 db2stop force
    - db2 db2start

Starting HADR Services for DWC DB



On Standby Node:

- db2 start hadr on db DWC as standby
- db2pd -db DWC -hadr (check for the following):
  - Connected

Starting HADR Services for DWC DB



On Primary Node:

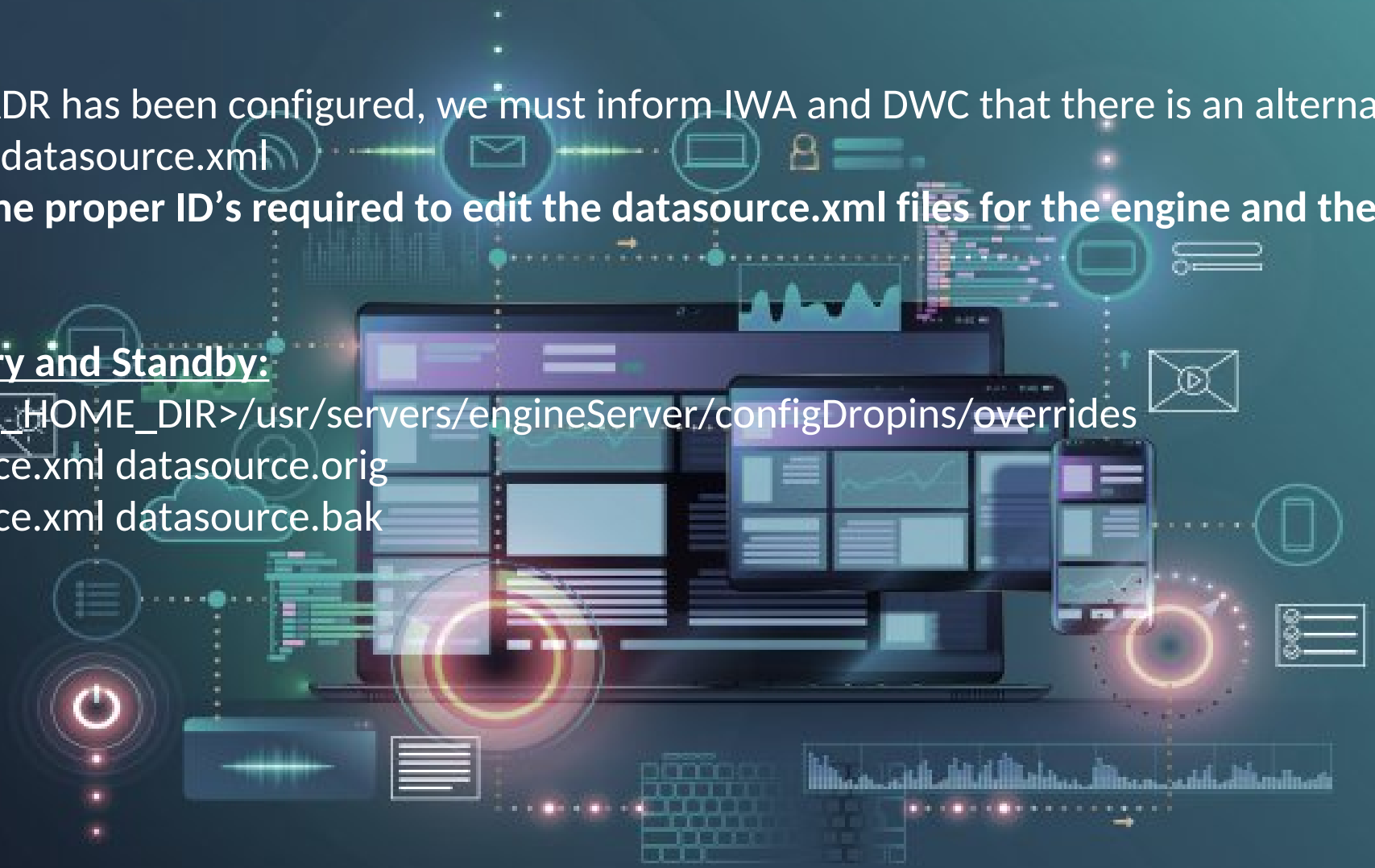
- db2 start hadr on db DWC as primary
- db2pd -db DWC -hadr (check for the following):
  - Connected

Post Configuration Steps

- Now that HADR has been configured, we must inform IWA and DWC that there is an alternate server by editing their datasource.xml
- **NOTE: Use the proper ID's required to edit the datasource.xml files for the engine and the console**

On both Primary and Standby:

- cd to /<TWS_HOME_DIR>/usr/servers/engineServer/configDropins/overrides
- cp datasource.xml datasource.orig
- cp datasource.xml datasource.bak



Post Configuration Steps



vi datasource.bak and make the following changes:

- Edit the following lines from:
  - <variable name="db.clientRerouteAlternateServerName" value="$(dbAlternateServerNames)"/>
  - <variable name="db.clientRerouteAlternatePortNumber" value="$(dbAlternateServerPorts)"/>
- To:
  - <variable name="db.clientRerouteAlternateServerName" value="<primary_host>,<standby_host>"/>
  - <variable name="db.clientRerouteAlternatePortNumber" value="<primary_port>,<standby_port>"/>
- Add the following lines immediately below the above lines in the file:
  - <variable name="db.retryIntervalForClientReroute" value="60"/>
  - <variable name="db.maxRetriesForClientReroute" value="5"/>
  - <variable name="enableSeamlessFailover" value="1"/>

- cp datasource.bak datasource.xml
- Restart the engines appserver

Post Configuration Steps

On both Primary and Standby:

- cd to /<DWC_HOME_DIR>/usr/servers/dwcServer/configDropins/overrides
- cp datasource.xml datasource.orig
- cp datasource.xml datasource.bak



Post Configuration Steps

vi datasource.bak and make the following changes:

- Edit the following lines from:
  - <variable name="db.clientRerouteAlternateServerName" value="$(dbAlternateServerNames)"/>
  - <variable name="db.clientRerouteAlternatePortNumber" value="$(dbAlternateServerPorts)"/>
- To:
  - <variable name="db.clientRerouteAlternateServerName" value="<primary_host>,<standby_host>"/>
  - <variable name="db.clientRerouteAlternatePortNumber" value="<primary_port>,<standby_port>"/>
- Add the following lines immediately below the above lines in the file:
  - <variable name="db.retryIntervalForClientReroute" value="60"/>
  - <variable name="db.maxRetriesForClientReroute" value="5"/>
  - <variable name="enableSeamlessFailover" value="1"/>
- cp datasource.bak datasource.xml
- Restart the engines appserver

Other Operational Steps



- To switch from standby to primary, log in to the server currently acting as the standby server as the DB instance owner and execute the below command
  - db2 takeover hadr on database tws (If primary is down, you will need to add "by force" at the end of the command)

- To stop HADR, you must always stop the HADR service on the standby server first
  - Log in to the standby server as the DB instance owner and execute the following:
    - db2 deactivate db <DB>
    - db2 stop hadr on db <DB>
    - db2stop (if trying to stop DB2)
  - Log in to the primary server as the DB instance owner and execute the following:
    - db2 stop hadr on db <DB>

Other Operational Steps



- To restart HADR, you must always start the HADR service on the standby server first
  - Log in to the standby server as the DB instance owner and execute the following:
    - db2start (ONLY if DB2 has been stopped previously)
    - db2 start hadr on db <DB> as standby
  - Log in to the primary server as the DB instance owner and execute the following:
    - db2 start hadr on db <DB> as primary

- To check HADR status
  - On either standby or primary, execute the following to check the status of each of the HADR nodes:
    - db2pd -db <DB> -hadr

HCLSoftware

hcltechsw.com