Oracle 19c Database Listener Configuration (Single & RAC Database)

1. Basic Steps for Listener Configuration in Oracle 19c

A **listener** is a process that listens for incoming client connection requests and manages traffic between the database and clients. Oracle provides two types of listener configurations:

1. **Static Listener Configuration** (Manually defined services)
2. **Dynamic Listener Configuration** (Services registered automatically)

2. Listener Configuration in a Single Instance Database

2.1 Steps for Configuring a Listener

1. **Check Existing Listener**

```
lsnrctl status
```

If a listener is not running, configure a new one.

2. **Create or Edit Listener.ora File**
   The listener configuration file is located at:

```
$ORACLE_HOME/network/admin/listener.ora
```

Example **listener.ora** (Static and Dynamic Configuration):

```
LISTENER =
  (DESCRIPTION_LIST =
   (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = dbserver)(PORT = 1521))
    )
  )

SID_LIST_LISTENER =
  (SID_LIST =
   (SID_DESC =
      (SID_NAME = orcl)
```

```
    (ORACLE_HOME = /u01/app/oracle/product/19c/dbhome_1)
  )
)
```

3. **Start the Listener**

```
lsnrctl start
```

4. **Check Listener Status**

```
lsnrctl status
```

5. **Register the Listener with the Database**

```
sql

ALTER SYSTEM SET
LOCAL_LISTENER='(ADDRESS=(PROTOCOL=TCP)(HOST=dbserver)(PORT=1521))';
ALTER SYSTEM REGISTER;
```

3. Listener Configuration in Oracle 19c RAC Database

In Oracle RAC, each node has a local listener and a SCAN listener.

3.1 Configuring SCAN Listener

SCAN listener allows clients to connect using a single name, which Oracle RAC resolves dynamically.

1. **Check SCAN Listeners**

```
srvctl status scan_listener
```

2. **Start SCAN Listeners (If Needed)**

```
srvctl start scan_listener
```

3. **Check Local Listeners**

```
srvctl status listener
```

4. **Configure Local Listener**

```
sql

ALTER SYSTEM SET LOCAL_LISTENER='(ADDRESS=(PROTOCOL=TCP)(HOST=node1-
vip)(PORT=1521))' SID='*';
```

5. **Register the Database with the Listener**

```
sql

ALTER SYSTEM REGISTER;
```

4. Static vs Dynamic Listener Configuration

Feature	Static Listener	Dynamic Listener
Configuration	Manually defined in `listener.ora`	Auto-registered by PMON
Service Management	Requires manual entry	Automatically updates if DB restarts
Use Case	Used for Oracle Data Guard, RAC ASM, NOMOUNT stage	Regular database connections
Flexibility	Less flexible	More flexible

5. Complex Oracle DBA Scenario based Questions

Here are some **complex** scenario with **workaround solutions**:

Q1. How do you configure a listener when the database is in NOMOUNT state?

Scenario:

- The database cannot register dynamically because it is in NOMOUNT mode.
- You need to set up a static listener to allow connection.

Workaround:

1. Edit the **listener.ora** file:

```
ini

SID_LIST_LISTENER =
  (SID_LIST =
   (SID_DESC =
      (GLOBAL_DBNAME = orcl)
      (ORACLE_HOME = /u01/app/oracle/product/19c/dbhome_1)
      (SID_NAME = orcl)
    )
  )
```

2. Restart the listener:

```
lsnrctl stop
lsnrctl start
```

3. Connect using **SQL*Plus**:

```
sqlplus sys@orcl as sysdba
```

Q2. How do you troubleshoot "ORA-12514: TNS:listener does not currently know of service requested"?

Scenario:

- A client connection fails because the listener doesn't recognize the requested service.
- This happens if the database hasn't registered with the listener.

Workaround:

1. **Check if the listener recognizes the service:**

```
lsnrctl status
```

If the service is missing, proceed.

2. **Manually Register the Database with the Listener:**

```
sql

ALTER SYSTEM SET
LOCAL_LISTENER='(ADDRESS=(PROTOCOL=TCP)(HOST=dbserver)(PORT=1521))';
ALTER SYSTEM REGISTER;
```

3. **Restart the Listener and Database:**

```
lsnrctl stop
lsnrctl start
```

Q3. How do you configure multiple listeners on a RAC environment?

Scenario:

- In a multi-node RAC, each node requires a local listener and a SCAN listener.

Workaround:

1. **Add a new local listener in `listener.ora`**:

```
ini

LISTENER_NODE1 =
  (DESCRIPTION_LIST =
   (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = node1-vip)(PORT = 1521))
    )
  )
```

2. **Register the Listener in the Database:**

```
sql

ALTER SYSTEM SET LOCAL_LISTENER='(ADDRESS=(PROTOCOL=TCP)(HOST=node1-
vip)(PORT=1521))';
ALTER SYSTEM REGISTER;
```

3. **Restart the listener on both nodes:**

```
srvctl stop listener -n node1
srvctl start listener -n node1
```

Q4. How do you fix a listener hanging issue in RAC after a failover?

Scenario:

- A failover occurs, but the listener on the failed node does not respond.
- The client receives `ORA-12541: TNS:no listener`.

Workaround:

1. **Check the status of the listener:**

```
srvctl status listener -n node1
```

2. **Restart the listener:**

```
srvctl stop listener -n node1
srvctl start listener -n node1
```

3. **Verify registration:**

```
sql

ALTER SYSTEM REGISTER;
```

Q5. How do you configure a dedicated listener for ASM in a RAC setup?

Scenario:

- ASM instances require a separate listener for inter-instance communication.

Workaround:

1. **Edit the `listener.ora` file on each RAC node:**

```
ini

ASM_LISTENER =
  (DESCRIPTION_LIST =
   (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = asm-vip)(PORT = 1522))
    )
  )
```

2. **Start the ASM Listener:**

```
lsnrctl start ASM_LISTENER
```

3. **Register ASM with the Listener:**

```
sql

ALTER SYSTEM SET LOCAL_LISTENER='(ADDRESS=(PROTOCOL=TCP)(HOST=asm-
vip)(PORT=1522))' SID='*';
ALTER SYSTEM REGISTER;
```

Final Thoughts

- **Static Listeners** are crucial for NOMOUNT connections and ASM.
- **Dynamic Listeners** simplify database connectivity in RAC and single instances.
- **Troubleshooting techniques** include checking status, restarting services, and manual registration.
- **RAC requires both local and SCAN listeners** for high availability.

Complex Real-Time Oracle 19c Listener Issues with Workarounds

As an experienced Oracle DBA (15+ years), you may encounter complex listener-related issues in real-world scenarios. Below are some challenging scenarios with detailed **workarounds**.

Scenario 1: ORA-12514 – "TNS: Listener Does Not Currently Know of Service Requested"

Issue:

- The listener is running, but the database service is **not registered** with it.
- Clients get the following error when trying to connect:

```
ORA-12514: TNS:listener does not currently know of service requested
in connect descriptor
```

Root Cause:

- PMON (Process Monitor) failed to auto-register the database with the listener.
- The listener is **not running on the correct port** or is missing in `LOCAL_LISTENER`.
- The database is **not in OPEN mode**.

Workaround:

Step 1: Check the Listener Status

```
lsnrctl status
```

- Verify if the service name is missing.

Step 2: Manually Register the Database with the Listener

Execute the following SQL commands:

```
sql

ALTER SYSTEM SET
LOCAL_LISTENER='(ADDRESS=(PROTOCOL=TCP)(HOST=dbserver)(PORT=1521))';
```

```
ALTER SYSTEM REGISTER;
```

- **Check if the service is now available:**

```
lsnrctl services
```

Step 3: Restart the Listener and Database

```
lsnrctl stop
lsnrctl start
```

Restart the database:

```
sql

shutdown immediate;
startup;
```

Step 4: Verify the Service Registration

```
sql

SELECT NAME, VALUE FROM V$PARAMETER WHERE NAME = 'local_listener';
```

Scenario 2: ORA-12541 – "TNS: No Listener"

Issue:

- The listener is **not running**, but the database is up.
- Clients trying to connect receive:

```
ORA-12541: TNS:no listener
```

Root Cause:

- The listener process crashed.
- Firewall settings are blocking the listener port.
- The listener is not listening on the expected port.

Workaround:

Step 1: Verify if the Listener is Running

```
lsnrctl status
```

- If the listener is **not running**, start it:

```
lsnrctl start
```

Step 2: Check if the Listener Port is Open

```
netstat -tulnp | grep 1521
```

- If the port **1521 is not open**, allow it through the firewall:

```
sudo firewall-cmd --add-port=1521/tcp --permanent
sudo firewall-cmd --reload
```

Step 3: Restart the Listener and Database

```
lsnrctl stop
lsnrctl start

sql

ALTER SYSTEM REGISTER;
```

Scenario 3: ORA-12560 – "TNS: Protocol Adapter Error" (On Windows Server)

Issue:

- Trying to connect to the database using **SQL*Plus** fails with:

```
ORA-12560: TNS:protocol adapter error
```

- This happens **only on Windows**.

Root Cause:

- The Oracle **listener service is not running**.
- The ORACLE_HOME and ORACLE_SID **environment variables are missing**.

Workaround:

Step 1: Check if the Listener is Running

```
cmd

lsnrctl status
```

- If not running, start it:

```
cmd

lsnrctl start
```

Step 2: Restart the Oracle Service on Windows

1. Open **Run → services.msc**
2. Locate **OracleServiceORCL** and **OracleListener**
3. Right-click and select **Start**.

Step 3: Set Environment Variables

```
cmd
```

```
set ORACLE_HOME=C:\app\oracle\product\19.0.0\dbhome_1
set ORACLE_SID=orcl
```

Scenario 4: Listener Hangs After Database Failover in Oracle RAC

Issue:

- A database failover occurs, and the listener on the failed node **stops responding**.
- Clients trying to connect to the SCAN listener receive:

```
ORA-12545: Connect failed because target host or object does not exist
```

Root Cause:

- The **SCAN listener is not re-registering** with the new primary node.
- The **VIP (Virtual IP) did not failover properly**.

Workaround:

Step 1: Check SCAN Listener Status

```
srvctl status scan_listener
```

- If the SCAN listener is **not running**, start it:

```
srvctl start scan_listener
```

Step 2: Restart the Local Listener on the New Primary Node

```
srvctl stop listener -n node1
srvctl start listener -n node1
```

Step 3: Manually Register the Database with the Listener

```
sql

ALTER SYSTEM REGISTER;
```

Step 4: Verify the Remote Listener

```
sql

SHOW PARAMETER REMOTE_LISTENER;
```

If incorrect, reset it:

```
sql

ALTER SYSTEM SET REMOTE_LISTENER='scan-listener.cluster.com:1521'
SCOPE=BOTH;
```

Scenario 5: Connection Slowness Due to Stale Listener Entries

Issue:

- Users complain that connections to the database are **taking longer than usual**.
- The listener log shows delays in processing connection requests.

Root Cause:

- Old, **stale listener entries** remain in the cache.
- **DNS resolution delays** affecting service registration.

Workaround:

Step 1: Check for Stale Connections

```
lsnrctl services
```

- Look for **inactive** or **duplicate** services.

Step 2: Flush Stale Services

```
lsnrctl reload
```

Step 3: Disable DNS Lookup for Faster Resolution

Edit **sqlnet.ora**:

```
ini

NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT)
```

Restart the listener:

```
lsnrctl stop
lsnrctl start
```

Scenario 6: High CPU Usage by Listener Due to Excessive Connections

Issue:

- The `tnslsnr` process is consuming **high CPU**.
- Multiple connection requests cause listener performance issues.

Root Cause:

- Too many **simultaneous connection requests**.
- Misconfigured **SQLNET.EXPIRE_TIME** leading to slow cleanup of dead connections.

Workaround:

Step 1: Set Connection Rate Limits in `listener.ora`

```
ini

RATE_LIMIT = 100
```

Step 2: Enable Dead Connection Detection in `sqlnet.ora`

```
ini

SQLNET.EXPIRE_TIME = 10
```

Step 3: Restart the Listener

```
lsnrctl stop
lsnrctl start
```

Final Thoughts

1. **Monitor listener logs regularly**:

```
tail -f $ORACLE_HOME/diag/tnslsnr/dbserver/listener/alert/log.xml
```

2. **Always verify listener registration** after database restarts:

```
sql

SELECT NAME, VALUE FROM V$PARAMETER WHERE NAME = 'local_listener';
```

3. **Use SCAN listener in RAC environments** for high availability.
4. **Set proper firewall rules** to avoid connection issues.

**********End of Documents Thank You**********