Lab: Adding Multiple PDBs to `tnsnames.ora` and Testing Connections

IMPORTANT: This lab is intended to be both a reference and a lab so it is possible that not all commands will work and that is ok. If a command fails - please continue

Objective:

To add PDB entries for PDBLAB1, PDBLAB2, PDBLAB3, and PDBLAB4 into the `tnsnames.ora` file, test the connections from the command line, and use SQL*Plus commands to show configurations. Additionally, verify the networking configurations in SQL Developer.

Step-by-Step Lab

Step 1: Add PDB Entries to `tnsnames.ora`

1. **Locate the `tnsnames.ora` File:**
   - The `tnsnames.ora` file is typically located in the `$ORACLE_HOME/network/admin` directory.
   - On UNIX-based systems, it might be in `/u01/app/oracle/product/19.3.0/dbhome_1/network/admin/`.
2. **Edit the `tnsnames.ora` File:**
   - Open the `tnsnames.ora` file in a text editor (e.g., `vi` or `nano`).

```
vi /u01/app/oracle/product/19.3.0/dbhome_1/network/admin/tnsnames.ora
```

3. **Add the PDB Entries:**

```
PDBLAB1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = pdblab1)
    )
  )

PDBLAB2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = pdblab2)
    )
  )

PDBLAB3 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = pdblab3)
    )
```

```
  )

PDBLAB4 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = pdblab4)
    )
  )
```

4. **Save and Exit:**
    - Save the changes and exit the text editor.

Step 2: Test Connections from the Command Line

1. **Set Oracle Environment Variables:**

```
. oraenv
ORACLE_SID = [oracle] ? CDBLAB
```

2. **Test Connection to `PDBLAB1`:**

```
sqlplus sys/fenago@PDBLAB1 AS SYSDBA
```

3. **Test Connection to `PDBLAB2`:**

```
sqlplus sys/fenago@PDBLAB2 AS SYSDBA
```

4. **Test Connection to `PDBLAB3`:**

```
sqlplus sys/fenago@PDBLAB3 AS SYSDBA
```

5. **Test Connection to `PDBLAB4`:**

```
sqlplus sys/fenago@PDBLAB4 AS SYSDBA
```

Step 3: Show Configurations with SQL*Plus Commands

1. **Connect to the CDB:**

```
sqlplus / as sysdba
```

2. **Verify the PDBs Status:**

```
SELECT con_id, name, open_mode FROM v$pdbs;
```

3. **Check Listener Status:**

```
lsnrctl status
```

4. **Show PDB Save States:**

```
SELECT con_id, con_name, state FROM DBA_PDB_SAVED_STATES;
```

5. **Exit SQL*Plus:**

```
EXIT;
```

Step 4: Verify Networking Configurations in SQL Developer

1. **Open SQL Developer.**
2. **Create New Connections for PDBLAB1, PDBLAB2, PDBLAB3, and PDBLAB4:**
   - **File -> New -> Database Connection...**
3. **Connection Details for `PDBLAB1`:**
   - **Connection Name**: PDBLAB1
   - **Username**: sys
   - **Password**: fenago
   - **Connection Type**: Basic
   - **Role**: SYSDBA
   - **Hostname**: localhost
   - **Port**: 1521
   - **Service Name**: pdblab1
4. **Test and Save the Connection:**
   - Click the `Test` button to verify the connection details.
   - Click the `Save` button to save the connection.
5. **Repeat for `PDBLAB2`, `PDBLAB3`, and `PDBLAB4`:**
   - **Connection Name**: PDBLAB2
     - **Username**: sys
     - **Password**: fenago
     - **Connection Type**: Basic
     - **Role**: SYSDBA
     - **Hostname**: localhost
     - **Port**: 1521
     - **Service Name**: pdblab2
   - **Connection Name**: PDBLAB3
     - **Username**: sys
     - **Password**: fenago
     - **Connection Type**: Basic
     - **Role**: SYSDBA
     - **Hostname**: localhost
     - **Port**: 1521
     - **Service Name**: pdblab3
   - **Connection Name**: PDBLAB4
     - **Username**: sys
     - **Password**: fenago
     - **Connection Type**: Basic

- **Role**: SYSDBA
- **Hostname**: localhost
- **Port**: 1521
- **Service Name**: pdblab4

6. **Test and Save Each Connection:**
   - Click the `Test` button to verify the connection details.
   - Click the `Save` button to save the connection.

Step 5: Explore PDBs in SQL Developer

1. **Expand Connections:**
   - In the Connections pane, expand `PDBLAB1`, `PDBLAB2`, `PDBLAB3`, and `PDBLAB4` to explore tables, views, and other database objects.
2. **Verify the PDB Connections:**
   - Run a simple query to ensure the connection is working:

```
SELECT name, open_mode FROM v$pdbs;
```

Summary:

By following these steps, you have successfully added PDB entries for PDBLAB1, PDBLAB2, PDBLAB3, and PDBLAB4 to the `tnsnames.ora` file, tested the connections from the command line, and verified the configurations using SQL*Plus commands and SQL Developer. This ensures that the PDBs are correctly configured and accessible from different tools.

Addendum: Using `tnsping` and Additional SQL Commands for CDB and PDB Management

Objective:

To use the `tnsping` utility for network diagnostics and explore additional SQL commands that an Oracle Database Administrator (DBA) should be aware of for managing CDBs and PDBs.

Step-by-Step Guide

Step 1: Using `tnsping` for Network Diagnostics

1. **Run `tnsping` to Test Connectivity:**

```
tnsping PDBLAB1
tnsping PDBLAB2
tnsping PDBLAB3
tnsping PDBLAB4
```

2. **Interpret `tnsping` Results:**
   - The output should show the time it takes to reach the database. A successful response indicates that the network connection to the specified service name is configured correctly.

Step 2: Additional SQL Commands for CDB and PDB Management

General CDB Management Commands

1. **Check the Current Container:**

```
SHOW CON_NAME;
```

2. **List All PDBs and Their Status:**

```
SELECT con_id, name, open_mode FROM v$pdbs;
```

3. **Create a New PDB (if needed):**

```
CREATE PLUGGABLE DATABASE PDBLAB4 ADMIN USER pdb_admin IDENTIFIED BY fenago
FILE_NAME_CONVERT = ('/u01/app/oracle/oradata/CDBLAB/pdbseed/',
'/u01/app/oracle/oradata/CDBLAB/PDBLAB4/');
```

4. **Open a Specific PDB:**

```
ALTER PLUGGABLE DATABASE PDBLAB1 OPEN;
ALTER PLUGGABLE DATABASE PDBLAB2 OPEN;
ALTER PLUGGABLE DATABASE PDBLAB3 OPEN;
ALTER PLUGGABLE DATABASE PDBLAB4 OPEN;
```

5. **Close a Specific PDB:**

```
ALTER PLUGGABLE DATABASE PDBLAB1 CLOSE IMMEDIATE;
```

6. **Drop a PDB:**

```
DROP PLUGGABLE DATABASE PDBLAB4 INCLUDING DATAFILES;
```

7. **Backup a PDB:**

```
BACKUP PLUGGABLE DATABASE PDBLAB1;
```

Checking Database Configuration and Performance

1. **Check Database Version:**

```
SELECT * FROM v$version;
```

2. **Check Initialization Parameters:**

```
SHOW PARAMETERS;
```

3. **Check SGA and PGA Sizes:**

```
SHOW SGA;
SHOW PGA;
```

4. **List All Tablespaces and Their Status:**

```
SELECT tablespace_name, status FROM dba_tablespaces;
```

5. **Check Free Space in Tablespaces:**

```
SELECT tablespace_name, file_id, bytes/1024/1024 AS free_space_mb
FROM dba_free_space;
```

6. **Monitor Active Sessions:**

```
SELECT sid, serial#, status, username, osuser, machine, program
FROM v$session
WHERE status = 'ACTIVE';
```

7. **Check Database Alerts and Logs:**

```
SELECT * FROM v$alert_log;
```

Performance Tuning and Troubleshooting

1. **Check Top SQL by CPU Usage:**

```
SELECT sql_id, sql_text, cpu_time, elapsed_time, executions
FROM v$sql
ORDER BY cpu_time DESC
FETCH FIRST 10 ROWS ONLY;
```

2. **Check Wait Events:**

```
SELECT event, total_waits, time_waited, average_wait
FROM v$system_event
ORDER BY time_waited DESC;
```

3. **Check Long-Running Queries:**

```
SELECT sql_id, sql_text, elapsed_time, executions
FROM v$sql
WHERE elapsed_time > 60000000; -- Adjust time threshold as needed
```

4. **Check Segment Usage:**

```
SELECT segment_name, segment_type, tablespace_name, bytes/1024/1024 AS size_mb
FROM dba_segments
ORDER BY size_mb DESC;
```

Configuration and Network Management

1. **Check Listener Status:**

```
lsnrctl status
```

2. **Reload Listener Configuration:**

```
lsnrctl reload
```

3. **Check TNS Aliases:**

```
lsnrctl services
```

Step 3: SQL Developer Networking Configurations

1. **Open SQL Developer.**
2. **Verify Connection Details:**
    - For each connection (PDBLAB1, PDBLAB2, PDBLAB3, PDBLAB4), right-click and select `Properties`.
    - Ensure the `Connection Type` is set to `TNS` and the `Network Alias` corresponds to the entries in the `tnsnames.ora` file.
3. **Test Connections:**
    - Use the `Test` button in the connection properties window to ensure each connection is properly configured.

Conclusion:

By following these additional steps, you can use `tnsping` for network diagnostics, utilize various SQL commands for comprehensive database management, and verify networking configurations in SQL Developer. These practices ensure robust and efficient management of Oracle CDBs and PDBs.