

Oracle Database 21c

运

维

手

册

目录

一、 前言	3
二、 简单命令的使用	3
三、 实例的启动和关闭.....	5
四、 参数管理.....	7
五、 用户管理.....	7
六、 权限(角色)的赋予和回收.....	8
七、 表空间管理	8
八、 数据库的备份	9
九、 Oracle 诊断体系.....	10
十、 日常检查内容	10

一、前言

Oracle Database 21c 为目前发布的长期支持和稳定版本、该手册主要从日常使用者的角度触发、整理归纳了日常使用的命令和内容。阅读参考者需要具备一定的 Linux/Unix 基础、Oracle Database 基础的数据库运维人员、信息技术人员等。

Oracle Database 的基础学习、进阶学习、高级功能学习请联系作者。

本手册在使用过程中、读者如有疑问、可以联系作者：13764423979(微信同号)

二、简单命令的使用

1、进入 sqlplus

\$sqlplus 用户名/密码 如：

[oracle@host01 ~]\$ sqlplus system/oracle

2、退出 sqlplus

SQL>exit; 或 SQL>quit;

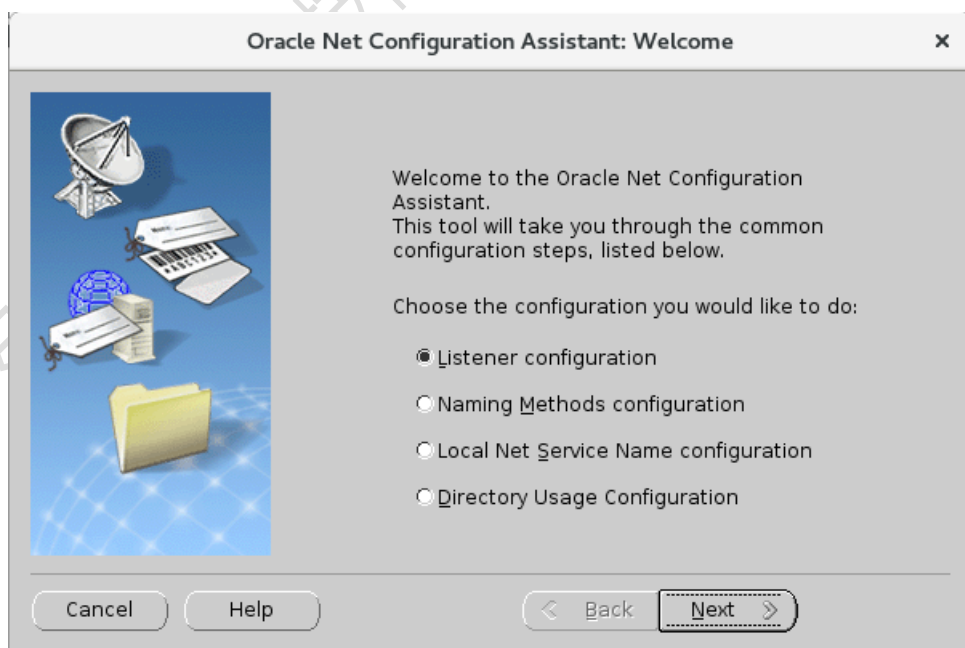
3、使用 sys 用户登录 sqlplus

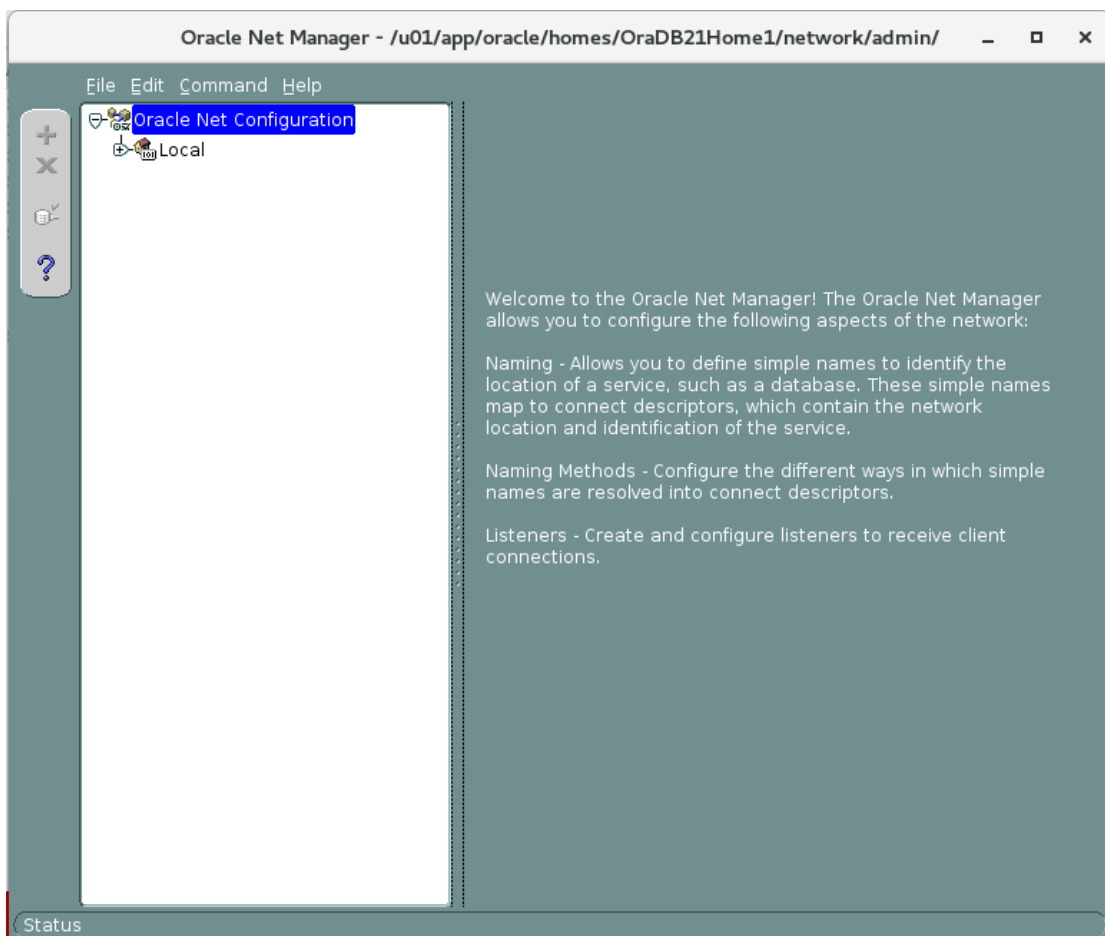
[oracle@host01 ~]\$ sqlplus / as sysdba

4、关于监听

4.1、新建、修改、删除监听器

使用 oracle 用户登录到操作系统，使用 netca 或 netmgr，会弹出图形界面、按照图形界面进行操作。(熟练者也可以选择直接编辑 listener.ora 文件)





4.2、启动监听器

[oracle@host01 ~]\$ lsnrctl start

```
[oracle@host01 ~]$ lsnrctl start

LSNRCTL for Linux: Version 21.0.0.0.0 - Production on 01-APR-2023 01:59:15

Copyright (c) 1991, 2021, Oracle. All rights reserved.

Starting /u01/app/oracle/product/21.3.0.0/db_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 21.0.0.0.0 - Production
System parameter file is /u01/app/oracle/homes/OraDB21Home1/network/admin/listener.ora
Log messages written to /u01/app/oracle/diag/tnslsnr/host01/listener/alert/log.xml
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=host01) (PORT=1521)))
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=EXTPROC1521)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=host01) (PORT=1521)))
STATUS of the LISTENER
-----
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 21.0.0.0.0 - Production
Start Date                01-APR-2023 01:59:16
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   /u01/app/oracle/homes/OraDB21Home1/network/admin/listener.ora
Listener Log File         /u01/app/oracle/diag/tnslsnr/host01/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=host01) (PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=EXTPROC1521)))
The listener supports no services
The command completed successfully
```

4.3、查看监听状态

[oracle@host01 ~]\$ lsnrctl status 或

[oracle@host01 ~]\$ lsnrctl service

4.4、手动注册服务到监听器

SQL> show user

USER is "SYS"

SQL> alter system register;

System altered.

```
[oracle@host01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 21.0.0.0.0 - Production on Sat Apr 1 01:59:41 2023
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> alter system register;

System altered.
```

4.5、关闭监听

[oracle@host01 ~]\$ lsnrctl stop

```
[oracle@host01 ~]$ lsnrctl stop

LSNRCTL for Linux: Version 21.0.0.0.0 - Production on 01-APR-2023 01:58:55

Copyright (c) 1991, 2021, Oracle. All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=host01)(PORT=1521)))
The command completed successfully
```

三、实例的启动和关闭

1、启动实例

[oracle@host01 ~]\$ sqlplus / as sysdba

SQL> startup;

```
SQL> startup
ORACLE instance started.

Total System Global Area 855634368 bytes
Fixed Size 9691584 bytes
Variable Size 406847488 bytes
Database Buffers 436207616 bytes
Redo Buffers 2887680 bytes
Database mounted.
Database opened.
```

注意：如果有设置 pdb 自动打开，此处需要手动打开 pdb;

SQL> alter pluggable database all open;

```
SQL> startup
ORACLE instance started.

Total System Global Area  855634368 bytes
Fixed Size                  9691584 bytes
Variable Size              406847488 bytes
Database Buffers           436207616 bytes
Redo Buffers                2887680 bytes
Database mounted.
Database opened.
SQL> show pdbs;

  CON_ID CON_NAME                                OPEN MODE RESTRICTED
-----
       2 PDB$SEED                                READ ONLY        NO
       3 ORADBPDB1                                MOUNTED
       4 ORADBPDB2                                MOUNTED
SQL> alter pluggable database all open;

Pluggable database altered.
```

2、关闭实例

[oracle@host01 ~]\$ sqlplus / as sysdba

SQL> show con_name

```
CON_NAME
-----
CDB$ROOT
```

SQL> shutdown immediate;

```
[oracle@host01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 21.0.0.0.0 - Production on Sat Apr 1 02:01:11 2023
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
```

注意: 此处必须连接到根容器、在普通 pdb 容器中执行 shutdown immediate 仅为关闭该 PDB。

3、集群环境启动和关闭

在集群环境下可使用单实例一样的方式关闭实例、也可以使用 srvctl 命令关闭和启动数据库实例。

[oracle@host01 ~]\$ srvctl stop database -d oradb -i oradb1

[oracle@host01 ~]\$ srvctl start database -d oradb -i oradb1

四、参数管理

1、查看参数

SQL> show parameter sga_target

NAME	TYPE	VALUE
sga_target	big integer	816M

或:

SQL> select name,value from v\$parameter where name = 'sga_max_size';

NAME	VALUE
sga_max_size	855638016

sga_max_size

855638016

2、修改参数

在内存中修改参数（立即生效、重启无效）

SQL> alter system set shared_pool_size=100m scope=memory;

在参数文件 spfile 中修改参数（当前无效、需重启实例才生效）

SQL> alter system set processes=300 scope=spfile;

在参数文件 spfile 和内存中同时修改（当前生效、重启亦生效）

SQL> alter system set shared_pool_size=100m scope=both;

注意：有些参数只能在参数文件 spfile 中修改、被称为静态参数、静态参数的修改必须重启数据库才能生效。另外一些少量参数可以在 pdb 级别进行修改、修改后仅在所在的 pdb 生效、其他 pdb 仍然继承 cdb 的参数值。

五、用户管理

3、新增用户信息

首先连接到需要新增用户的 pdb

[oracle@host01 ~]\$ sqlplus / as sysdba

SQL> alter session set container=oradbpdb1;

SQL> create user hr identified by hr;

创建用户的同时也可以指定默认表空间、默认临时表空间、概要文件等信息。

SQL> create user hr identified by hr default tablespace users temporary tablespace temp profile default;

如果连接的是根容器：创建的用户为全局用户、默认情况下全局用户需要使用 c##作为前缀。

SQL> create user c##hr identified by oracle;

注意：全局用户可以登录所有容器、而 pdb 用户只能登录用户所在的 pdb。

4、修改用户信息

首先连接到需要用户所在的 **pdb**、多数情况下、修改用户信息的命令和新增用户信息的命令较为相似，将第一个单词由 **create** 换成 **alter**。

```
SQL> alter user hr identified by hr123;
```

```
SQL> alter user hr identified by hr default tablespace users temporary tablespace temp profile default;
```

5、删除用户信息

首先连接到需要用户所在的 **pdb**

```
SQL> drop user hr cascade;
```

注意：用户被删除时、用户拥有的对象(表、视图等)也会被删除。因此需要确认后方可删除、如有必要、可先使用逻辑备份工具备份该用户。

六、权限(角色)的赋予和回收

1、权限的赋予

只有 **DBA** 角色或者具有相应权限的用户才可以赋予权限和回收权限、用户不能赋权给自己、需要首先登录到用户所在的 **pdb**。

```
SQL> grant create table,create view to scott;
```

```
SQL> grant select,update,insert,delete on hr.t1 to scott;
```

2、权限的回收

```
SQL> revoke create table from scott;
```

```
SQL> revoke delete on hr.t1 from scott;
```

3、角色的使用

创建角色

```
SQL> create role r1;
```

注意全局角色和全局用户一样、在命名上有前缀要求

授权给角色

```
SQL> grant create table to r1;
```

回收角色权限

```
SQL> revoke create table from r1;
```

角色授予用户

```
SQL> grant r1 to hr;
```

回收用户角色

```
SQL> revoke r1 from hr;
```

删除角色

```
SQL> drop role r1;
```

七、表空间管理

1、创建表空间

首先需要登录到需要操作的 **pdb**、创建用户表空间与创建回滚表空间、临时表空间的语法略有不同、以下以最为常用的用户表空间为例。

```
SQL>create tablespace users2 datafile '/u01/app/oracle/oradata/ORADB/oradbpdb1/users02.dbf'  
size 100m autoextend on;
```

2、更改表空间

修改数据文件初始化大小

```
SQL>alter database datafile '/u01/app/oracle/oradata/ORADB/oradbpdb1/users02.dbf' resize  
200m;
```

修改数据文件其他属性

```
SQL>alter database datafile
```

```
'/u01/app/oracle/oradata/ORADB/oradbpdb1/users02.dbf' autoextend on next 1m;
```

添加数据文件到表空间

```
SQL> alter tablespace users2 add datafile
```

```
'/u01/app/oracle/oradata/ORADB/oradbpdb1/users02_02.dbf' size 100m autoextend on;
```

3、删除表空间

```
SQL> drop tablespace users2 including contents and datafiles;
```

八、数据库的备份

1、逻辑备份

创建物理目录

```
[oracle@host01 ~]$ mkdir -p /u01/app/oracle/dump1
```

创建逻辑目录对象并授权给指定用户

```
[oracle@host01 ~]$ sqlplus / as sysdba
```

```
SQL> alter session set container=oradbpdb1;
```

```
SQL> create or replace directory dump1 as '/u01/app/oracle/dump1';
```

```
SQL> grant read,write on directory dump1 to hr;
```

expdp 命令的使用

导出表:

```
[oracle@host01 ~]$ expdp hr/hr directory=dump1 dumpfile=t1.dmp tables=t1,t2;
```

导出用户:

```
[oracle@host01 ~]$ expdp hr/hr directory=dump1 dumpfile=hr2.dmp schemas=hr2;
```

导出表空间:

```
[oracle@host01 ~]$ expdp hr/hr directory=dump1 dumpfile=users.dmp tablespaces=users;
```

导出数据库:

```
[oracle@host01 ~]$ expdp hr/hr directory=dump1 dumpfile=db.dmp full=y;
```

expdp 参数的灵活使用:

可使用 content 参数只导出元数据(metadata_only)或者只导出数据部分(data_only) 如:

```
[oracle@host01 ~]$ expdp system/oracle@oradbpdb1 directory=dump1 dumpfile=hr_meta.dmp  
schemas=hr content=data_only;
```

可使用 compress 压缩导出的文件 如:

```
[oracle@host01 ~]$ expdp system/oracle@oradbpdb1 directory=dump1 dumpfile=hr_meta.dmp
```

schemas=hr content=data_only compression=all;
以及其他如 query、flashbackscn 参数等。

2、物理备份

如连接的是 root 容器、则备份整个 cdb、如连接的是 pdb、则是备份该 pdb
RMAN> backup database plus archivelog;

备份某个 pdb

RMAN> backup pluggable database oradbpdb1;

备份某个表空间

RMAN> backup tablespace oradbpdb1:users;

九、Oracle 诊断体系

数据库的警告日志

查看诊断路径

SQL> show parameter diag

NAME	TYPE	VALUE
------	------	-------

diagnostic_dest	string	/u01/app/oracle
-----------------	--------	-----------------

数据库各组件的均警告日志位于/u01/app/oracle/diag 下、如监听器日志为:

[oracle@host01 oracle]\$ cd /u01/app/oracle/diag/tnslsnr/host01/listener/trace

[oracle@host01 trace]\$ ls

listener.log

数据库的实例日志位于:

[oracle@host01 trace]\$ cd /u01/app/oracle/diag/rdbms/oradb/oradb/trace

[oracle@host01 trace]\$ ls alert_oradb.log

alert_oradb.log

其他组件雷同。

十、日常检查内容

1、检查系统状态性能

记录数据库的 cpu 使用、IO、内存等使用情况，使用 vmstat,iostat,sar,top 等命令进行信息收集并检查这些信息，判断资源使用情况。

1.1、CPU 使用情况:

```
top - 00:56:49 up 14:48, 7 users, load average: 0.12, 0.15, 0.26
Tasks: 357 total, 2 running, 355 sleeping, 0 stopped, 0 zombie
%Cpu(s): 6.2 us, 6.2 sy, 0.0 ni, 87.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 2780856 total, 218504 free, 1317548 used, 1244804 buff/cache
KiB Swap: 3145724 total, 1729180 free, 1416544 used. 502636 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
6296	root	20	0	157988	4416	3636	R	6.2	0.2	0:00.01	top
13762	fanchent	20	0	386232	1776	0	S	6.2	0.1	2:47.63	vmtoolsd
1	root	20	0	201904	4932	3088	S	0.0	0.2	0:05.00	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:06.87	ksoftirqd/0
7	root	20	0	0	0	0	S	0.0	0.0	0:02.45	rcu_sched
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
9	root	20	0	0	0	0	R	0.0	0.0	0:02.45	rcuos/0
10	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcuob/0
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
12	root	rt	0	0	0	0	S	0.0	0.0	0:00.23	watchdog/0
13	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	khelper
14	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
15	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	netns
16	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	perf
17	root	20	0	0	0	0	S	0.0	0.0	0:00.04	khungtaskd
18	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	writeback
19	root	25	5	0	0	0	S	0.0	0.0	0:00.00	ksmd

注意上面的红框部分，此部分内容表示系统剩余的 cpu，当其平均值下降至 10%以下的时视为 CPU 使用率异常，需记录下该数值，并将状态记为异常。

1.2、内存使用情况:

```
[root@host01 ~]# free -m
```

	total	used	free	shared	buff/cache	available
Mem:	2715	1272	211	707	1231	489
Swap:	3071	1383	1688			

如上所示，红框部分表示系统总内存，黄框部分表示系统使用的内存，蓝框部分表示系统剩余内存，当剩余内存低于总内存的 10%时视为异常。

1.3、系统 I/O 情况:

```
[root@host01 ~]# iostat -k 1 3
```

Linux 4.1.12-94.3.9.el7uek.x86_64 (host01) 04/01/2023 _x86_64_ (1 CPU)						
avg-cpu:	%user	%nice	%system	%iowait	%steal	%idle
	2.18	0.01	0.58	1.54	0.00	95.69
Device:	tps	kB_read/s	kB_wrtn/s	kB_read	kB_wrtn	
sda	11.64	639.79	439.23	33555398	23036421	
scd0	0.01	1.10	0.00	57650	0	
dm-0	11.47	632.11	403.70	33152744	21173148	
dm-1	10.00	6.07	33.99	318476	1782720	
avg-cpu:	%user	%nice	%system	%iowait	%steal	%idle
	1.01	0.00	0.00	1.01	0.00	97.98

如上所示，第一个红框为设备的使用情况，第二个红框为 cpu IO 等待情况。

1.4、系统负载情况:

```
[root@host01 ~]# uptime
```

```
01:03:07 up 14:54, 7 users, load average: 0.07, 0.13, 0.22
```

如上所示，蓝色字部分表示系统负载，后面的 3 个数值如果有高于 2.5 的时候就表明系统在超负荷运转了，并将此值记录到巡检表，视为异常。

2、检查 Oracle 实例状态

```
SQL> col host_name for a20
```

```
SQL> select instance_name,host_name,startup_time,status,database_status from v$instance;
```

INSTANCE_NAME	HOST_NAME	STARTUP_T	STATUS	DATABASE_STATUS
oradb	host01	31-MAR-23	OPEN	ACTIVE

```
SQL> select instance_name,host_name,startup_time,status,database_status from v$instance;
```

INSTANCE_NAME	HOST_NAME	STARTUP_T	STATUS	DATABASE_STATUS
oradb	host01	31-MAR-23	OPEN	ACTIVE

其中“STATUS”表示 Oracle 当前的实例状态，必须为“OPEN”；“DATABASE_STATUS”表示 Oracle 当前数据库的状态，必须为“ACTIVE”。

```
SQL> select name,log_mode,open_mode from v$database;
```

NAME	LOG_MODE	OPEN_MODE
ORADB	ARCHIVELOG	READ WRITE

```
SQL> select name,log_mode,open_mode from v$database;
```

NAME	LOG_MODE	OPEN_MODE
ORADB	ARCHIVELOG	READ WRITE

其中“LOG_MODE”表示 Oracle 当前的归档方式。“ARCHIVELOG”表示数据库运行在归档模式下，“NOARCHIVELOG”表示数据库运行在非归档模式下。在我们的系统中数据库必须运行在归档方式下。

3、检查 Oracle 监听状态

```
[oracle@host01 ~]$ lsnrctl status
```

```
LSNRCTL for Linux: Version 21.0.0.0.0 - Production on 01-APR-2023 01:08:00
```

```
Copyright (c) 1991, 2021, Oracle. All rights reserved.
```

```
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=host01)(PORT=1521)))
```

```
STATUS of the LISTENER
```

```
-----
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 21.0.0.0.0 - Production
Start Date                01-APR-2023 00:31:41
Uptime                    0 days 0 hr. 36 min. 18 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   /u01/app/oracle/homes/OraDB21Home1/network/admin/listener.ora
Listener Log File         /u01/app/oracle/diag/tnslsnr/host01/listener/alert/log.xml
Listening Endpoints Summary...
```

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=host01)(PORT=1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

Services Summary...

Service "c8209f27c6b16005e053362ee80ae60e" has 1 instance(s).

Instance "oradb", status READY, has 1 handler(s) for this service...

Service "f8348565b41d7658e0531570a8c01365" has 1 instance(s).

Instance "oradb", status READY, has 1 handler(s) for this service...

Service "f8348add9d5777a1e0531570a8c0301b" has 1 instance(s).

Instance "oradb", status READY, has 1 handler(s) for this service...

Service "oradb" has 1 instance(s).

Instance "oradb", status READY, has 1 handler(s) for this service...

Service "oradbXDB" has 1 instance(s).

Instance "oradb", status READY, has 1 handler(s) for this service...

Service "oradbpdb1" has 1 instance(s).

Instance "oradb", status READY, has 1 handler(s) for this service...

Service "oradbpdb2" has 1 instance(s).

Instance "oradb", status READY, has 1 handler(s) for this service...

The command completed successfully

```
[oracle@host01 ~]$ lsnrctl status

LSNRCTL for Linux: Version 21.0.0.0.0 - Production on 01-APR-2023 01:48:28

Copyright (c) 1991, 2021, Oracle. All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=host01)(PORT=1521)))
STATUS of the LISTENER
-----
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 21.0.0.0.0 - Production
Start Date                01-APR-2023 00:31:41
Uptime                    0 days 1 hr. 16 min. 47 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   /u01/app/oracle/homes/OraDB21Home1/network/admin/listener.ora
Listener Log File         /u01/app/oracle/diag/tnslsnr/host01/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=host01)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
Services Summary...
Service "c8209f27c6b16005e053362ee80ae60e" has 1 instance(s).
  Instance "oradb", status READY, has 1 handler(s) for this service...
Service "f8348565b41d7658e0531570a8c01365" has 1 instance(s).
  Instance "oradb", status READY, has 1 handler(s) for this service...
Service "f8348add9d5777a1e0531570a8c0301b" has 1 instance(s).
  Instance "oradb", status READY, has 1 handler(s) for this service...
Service "oradb" has 1 instance(s).
  Instance "oradb", status READY, has 1 handler(s) for this service...
Service "oradbXDB" has 1 instance(s).
  Instance "oradb", status READY, has 1 handler(s) for this service...
Service "oradbpdb1" has 1 instance(s).
  Instance "oradb", status READY, has 1 handler(s) for this service...
Service "oradbpdb2" has 1 instance(s).
  Instance "oradb", status READY, has 1 handler(s) for this service...
The command completed successfully
```

蓝色部分为当前监听器监听到的数据库服务。

检查监听进程是否存在:

```
[oracle@host01 ~]$ ps -ef|grep lsn|grep -v grep
```

```
oracle      4720      1  0 00:31 ? 00:00:00 /u01/app/oracle/product/21.3.0.0/db_1/bin/tnslsnr
```

LISTENER -inherit

```
oracle@host01 ~]$ ps -ef|grep lsn|grep -v grep
oracle 4720 1 0 00:31 ? 00:00:00 /u01/app/oracle/product/21.3.0.0/db_1/bin/tnslsnr LISTENER -inherit
oracle@host01 ~]$
```

4、检查数据库进程数和会话数

当前的系统进程数

```
[oracle@host01 ~]$ ps -ef|grep LOCAL|grep -v grep |wc
```

```
2      18      220
```

```
oracle 4720 1 0 00:31 ? 00:00:00 /u01/app/oracle/product/21.3.0.0/db_1/bin/tnslsnr LISTENER -inherit
[oracle@host01 ~]$ ps -ef|grep LOCAL|grep -v grep |wc
2      18      220
[oracle@host01 ~]$
```

当前数据库支持的进程总量

```
SQL> show parameter processes;
```

```
SQL> select name,value from v$parameter where name ='processes';
```

```
NAME                                VALUE
```

```
processes                          300
```

```
SQL> col name for a30
SQL> col value for a30
SQL> select name,value from v$parameter where name ='processes';

NAME                                VALUE
-----
processes                          300
```

数据库的当前进程数应该要小于数据库支持总量的 75%以下较为安全。

5、查看表空间的状态和大小

查看表空间的状态和大小，看看是否为自动增长，是否为 bigfile 类型，若不是 bigfile 类型则关注单数据文件是否达到 32G；

Oracle 表空间查看：

```
select a.tablespace_name, total, free, total-free as used from
```

```
(select tablespace_name, sum(bytes)/1024/1024 as total from dba_data_files group by tablespace_name) a,
```

```
(select tablespace_name, sum(bytes)/1024/1024 as free from dba_free_space group by tablespace_name) b
```

```
where a.tablespace_name = b.tablespace_name;
```

SQL Output Statistics

```
select a.tablespace_name, total, free, total-free as used from
(select tablespace_name, sum(bytes)/1024/1024 as total from dba_data_files group by tablespace_name) a,
(select tablespace_name, sum(bytes)/1024/1024 as free from dba_free_space group by tablespace_name) b
where a.tablespace_name = b.tablespace_name;
```

	TABLESPACE_NAME	TOTAL	FREE	USED
1	SYSTEM	1330	9.0625	1320.9375
2	SYSAUX	590	31.9375	558.0625
3	UNDOTBS1	120	7.8125	112.1875
4	USERS	5	2.3125	2.6875

查看表空间是否自动扩展状况:

SQL> select TABLESPACE_NAME,AUTOEXTENSIBLE from dba_data_files;

select TABLESPACE_NAME,AUT ... x

SQL Output Statistics

```
select TABLESPACE_NAME,AUTOEXTENSIBLE from dba_data_files;
```

	TABLESPACE_NAME	AUTOEXTENSIBLE
1	SYSTEM	YES
2	SYSAUX	YES
3	UNDOTBS1	YES
4	USERS	YES

6、检查 oracle 日志文件

在本节主要检查相关的日志文件，包含：检查操作系统的日志文件，检查 Oracle 日志文件，检查 Oracle 核心转储目录，检查 Root 用户和 Oracle 用户的 email，总共四个部分。

检查操作系统日志文件

```
[root@host01 ~]# cat /var/log/messages |grep failed
```

查看是否有与 Oracle 用户相关的出错信息。

检查 oracle 日志文件

```
cat /u01/app/oracle/diag/rdbms/oradb/oradb/trace/alert_oradb.log | grep ora-
```

```
cat /u01/app/oracle/diag/rdbms/oradb/oradb/trace/alert_oradb.log | grep err
```

at /u01/app/oracle/diag/rdbms/oradb/oradb/trace/alert_oradb.log | grep failOracle 在运行过程中，会在警告日志文件(alert_SID.log)中记录数据库的一些运行情况：数据库的启动、关闭，启动时的非缺省参数；数据库的重做日志切换情况，记录每次切换的时间，及如果因为检查点(checkpoint)操作没有执行完成造成不能切换，会记录不能切换的原因；对数据库进行的某些操作，如创建或删除表空间、增加数据文件；数据库发生的错误，如表空间不够、出现坏块、数据库内部错误(ORA-600)等。定期检查日志文件，根据日志中发现的问题及时进行处理：

问题	处理
启动参数不对	检查初始化参数文件
因为检查点操作或归档操作没有完成造成重做日志不能切换	如果经常发生这样的情况，可以考虑增加重做日志文件组；想办法提高检查点或归档操作的效率；
有人未经授权删除了表空间	检查数据库的安全问题，是否密码太简单；如有必要，撤消某些用户的系统权限
出现坏块	检查是否是硬件问题(如磁盘本生有坏块)，如果不是，检查是哪个数据库对象出现了坏块，对这个对象进行重建
表空间不够	增加数据文件到相应的表空间
出现 ORA-600	根据日志文件的内容查看相应的 TRC 文件，如果是 Oracle 的 bug，要及时打上相应的补丁

7、检查备份状态

通过 RMAN TARGET / 进入 RMAN 命令行，然后通过 list backup 查看备份情况：

```
[oracle@host01 ~]$ rman target /
```

```
Recovery Manager: Release 21.0.0.0.0 - Production on Sat Apr 1 01:29:10 2023
Version 21.3.0.0.0
```

```
Copyright (c) 1982, 2021, Oracle and/or its affiliates. All rights reserved.
```

```
connected to target database: ORADB (DBID=2886223984)
```

```
RMAN> list backup;
```

确认数据库备份的有效性。

8、检查 RAC 服务状态

如果数据库为 RAC 结构，则进入 grid 用户，通过 crs_stat -t -v 命令查看 RAC 服务是否为 online 状态，确定每个节点都处于 open 状态。

```
su - grid
crs_stat -t -v
```

9、DG 的同步检查

如果数据库有搭建 DG、需要检查 DG 的同步情况

```
SQL> select name,value,unit,time_computed from v$dataguard_stats where name in ('transport lag','apply lag');
```