



Monitoring Oracle

Oracle: Database PowerPack version 106

Table of Contents

Introduction	5
What is Oracle Database?	5
What Does the Oracle: Database PowerPack Monitor?	6
What Operating Systems Does the Oracle: Database PowerPack Monitor?	6
Installing the Oracle: Database PowerPack	7
Configuring Oracle Monitoring	8
Prerequisites for Monitoring Oracle Database Instances	8
Required Privileges	10
Monitoring Pluggable Databases (PDBs)	11
Creating Users and Assigning Privileges	11
Prerequisites to Monitor Using Transport Layer Security (TLS/TCPS)	12
Monitoring TCPS Use Case	13
Configuring Oracle Credentials	15
Suggested Timeout Configuration	15
Creating an SSH/Key Credential	16
Creating an SSH/Key Credential in the Classic SL1 User Interface	17
Creating a PowerShell Credential (Windows Users)	18
Creating a PowerShell Credential (Windows Users) in the Classic SL1 User Interface	19
Creating a Database Credential	21
Creating a Database Credential in the Classic SL1 User Interface	23
Creating a SOAP/XML Credential	25
HTTP Headers	26
Headers for PDBs	26
Headers for RAC	29
Headers for Non-CDB	31
Headers for ASM	32
Headers for TCPS	33
HTTP Headers Special Keywords	34
TNS_ADMIN	34
TNS_ADMIN Within a SID Header	35
GRID_PATH	36

TYPE_OF_OS	36
Deprecated HTTP Configuration Headers	38
Creating a SOAP/XML Credential in the Classic SL1 User Interface	38
HTTP Headers	40
Headers for PDBs	40
Headers for RAC	43
Headers for Non-CDB	45
Headers for ASM	46
Headers for TCPS	47
HTTP Headers Special Keywords	48
TNS_ADMIN	48
TNS_ADMIN Within an SIDE Header	48
GRID_PATH	49
TYPE_OF_OS	50
Deprecated HTTP Configuration Headers	52
Enabling PEM on a Linux Machine	52
Discovering Oracle Database Instances	53
Discovering Oracle Database Instances in the SL1 Classic User Interface	56
Verifying Discovery and Dynamic Application Alignment	58
Snippet and Snippet Argument Configuration for New Oracle Dynamic Applications	62
Running SQL Queries from Snippet Arguments	62
Running Raw SQL Queries from the Snippet	63
Indexed Raw SQL Query	64
Running Combined Raw SQL Queries and Snippet Arguments	65
Running SSH Commands from a Snippet	65
Viewing Oracle Component Devices	66
Viewing Oracle Component Devices in the Classic SL1 User Interface	66
Oracle: Database Dashboards	67
Device Dashboard	67
Oracle Database: Instance	67
Oracle: Database Minimum Permissions Needed	69
Folder path through lsnrctl, tnsnames.ora	69

Execute permission to run lsnrctl 70

Read permission to read libclntsh.so.21.1, libclntshcore.so.21.1, libnnz21.so 70

Folder path to read tnsnames.ora and mesg folder 70

Read permission for oratab 71

Chapter

1

Introduction

Overview

This manual describes how to configure SL1 to monitor Oracle Database instances.

This chapter covers the following topics:

<i>What is Oracle Database?</i>	5
<i>What Does the Oracle: Database PowerPack Monitor?</i>	6
<i>What Operating Systems Does the Oracle: Database PowerPack Monitor?</i>	6
<i>Installing the Oracle: Database PowerPack</i>	7

NOTE: ScienceLogic provides this documentation for the convenience of ScienceLogic customers. Some of the configuration information contained herein pertains to third-party vendor software that is subject to change without notice to ScienceLogic. ScienceLogic makes every attempt to maintain accurate technical information and cannot be held responsible for defects or changes in third-party vendor software. There is no written or implied guarantee that information contained herein will work for all third-party variants. See the End User License Agreement (EULA) for more information.

What is Oracle Database?

Oracle Database is a multi-model database management system used for running online transaction processing, data warehousing, and mixed database workloads. Oracle Database supports relational, document, graph, and key-value data models, which improves storage efficiency and can retrieve structured and unstructured data. The system is designed for high performance, scalability, and reliability, and offers features such as ACID-compliant transactions, data security, and high availability. Oracle Database is suitable for a wide range of applications, from small-scale installations to large enterprise environments.

What Does the Oracle: Database PowerPack Monitor?

The "Oracle: Database" PowerPack includes Dynamic Applications that can monitor performance metrics and collect configuration data for Oracle databases and their instances.

- Supported "Oracle: Database" PowerPack versions: 18x, 19x, 21x
- Supported connection types: SSH, PowerShell, TCP, TCPS

In addition to Dynamic Applications, the PowerPack includes the following features:

- Dynamic Applications to discover, model, and monitor performance metrics and collect configuration data for Oracle Database instances
- Event Policies and corresponding alerts that are triggered when Oracle devices meet certain status criteria
- Device Classes for each of the Oracle devices monitored
- Sample Credentials for discovering Oracle devices
- A Device Dashboard to display summary information about an Oracle Database instance
- Discovery of Container Databases (CDBs) and Pluggable Databases (PDBs)
- Discovery of Active-Active RAC nodes individually
- Monitoring of Automatic Storage Management (ASM) instances

What Operating Systems Does the Oracle: Database PowerPack Monitor?

The *Oracle: Database* PowerPack supports operating system discovery for a variety of operating systems. The following operating systems can be monitored by the *Oracle: Database* PowerPack:

- AIX
- CentOS
- HP-UX
- Oracle Linux
- Red Hat Enterprise Linux
- Solaris
- SUSE Linux
- Ubuntu
- Windows

Installing the Oracle: Database PowerPack

Before completing the steps in this manual, you must import and install the latest version of the *Oracle: Database PowerPack*.

If you have the *SLPS: Oracle DB PowerPack* or the *Oracle DB PowerPack* installed, you must remove them from your SL1 system.

You must also remove any pre-existing discovered "Oracle: Database" PowerPack device trees and all Oracle device classes before installing this PowerPack. If you are using an SL1 version that is not at least 12.2.0, you must delete the "Oracle: Database" PowerPack example PowerShell credentials.

NOTE: If you are upgrading from an earlier version of the PowerPack, see the [Release Notes](#) for the version you are installing for upgrade instructions.

TIP: By default, installing a new version of a PowerPack overwrites all content from a previous version of that PowerPack that has already been installed on the target system. You can use the **Enable Selective PowerPack Field Protection** setting in the **Behavior Settings** page (System > Settings > Behavior) to prevent new PowerPacks from overwriting local changes for some commonly customized fields. For more information, see the section on [Global Settings](#).

To download and install the PowerPack:

1. Search for and download the PowerPack from the **PowerPacks** page (Product Downloads > PowerPacks & SyncPacks) at the [ScienceLogic Support Site](#).
2. In SL1, go to the **PowerPacks** page (System > Manage > PowerPacks).
3. Click the **[Actions]** button and choose *Import PowerPack*. The **Import PowerPack** dialog box appears.
4. Click **[Browse]** and navigate to the PowerPack file from step 1.
5. Select the PowerPack file and click **[Import]**. The **PowerPack Installer** modal displays a list of the PowerPack contents.
6. Click **[Install]**. The PowerPack is added to the **PowerPacks** page.

NOTE: If you exit the **PowerPack Installer** modal without installing the imported PowerPack, the imported PowerPack will not appear in the **PowerPacks** page. However, the imported PowerPack will appear in the **Imported PowerPacks** modal. This page appears when you click the **[Actions]** menu and select *Install PowerPack*.

Chapter

2

Configuring Oracle Monitoring

Overview

The following sections describe how to configure your Oracle Database instances for monitoring by SL1 using the *Oracle: Database PowerPack*:

This chapter covers the following topics:

<i>Prerequisites for Monitoring Oracle Database Instances</i>	8
<i>Configuring Oracle Credentials</i>	15
<i>Enabling PEM on a Linux Machine</i>	52
<i>Discovering Oracle Database Instances</i>	53
<i>Verifying Discovery and Dynamic Application Alignment</i>	58
<i>Snippet and Snippet Argument Configuration for New Oracle Dynamic Applications</i>	62
<i>Viewing Oracle Component Devices</i>	66

Prerequisites for Monitoring Oracle Database Instances

To configure the SL1 system to monitor Oracle Database instances using the *Oracle: Database PowerPack*, you must have a minimum of two users:

1. SSH user to access instance related data (pmon, smon, listener status, tnsnames.ora, etc.) from the server hosting the Oracle DB instance.
2. Oracle Database user to access the instance.

NOTE: On a multi-tenant instance like a CDB, there can be multiple users to access CDBs and PDBs

- Minimal permissions for SSH user access:
 - Permissions to access \$ORACLE_HOME directory:
 - Directories: --x
 - Binaries (\$ORACLE_HOME/bin): --x
 - Libraries (\$ORACLE_HOME/lib): r-x
 - File "listener.ora": r--
 - File "tnsnames.ora": r--
 - ORACLE_HOME Directory: --x
 - Everything else that's not executable in \$ORACLE_HOME: r--
 - Permissions to execute commands. For example, \$ORACLE_HOME/lsnrctl status.
 - Permissions to read files. For example, listeners.ora/tnsnames.ora (\$ORACLE_HOME or \$TNS_ADMIN folder).

NOTE: For more information about the minimum permissions needed, and why they are required, see the Oracle: Database Minimum Permissions Needed Appendix

NOTE: r = read; w = write; x = execute; - = denied. All files and directories can be owned by another non-credential user in SL1. However, the permissions must be given to the credential user's group, or to everyone else (Other).

TIP: If you do not want to configure permissions, you can move the SSH user to the group used by the Oracle installer.

The Oracle database user must have access to the following tables:

- dba_data_files
- dba_free_space
- dba_registry
- dba_scheduler_jobs
- dba_tablespaces
- dba_temp_files
- gv\$sort_segment
- sys.dba_ind_partitions
- sys.dba_ind_subpartitions
- sys.dba_indexes
- sys.dba_objects

- sys.v_\$database_block_corruption
- sys.v_\$lock
- v_\$archive_dest
- v_\$archived_log
- v_\$block_change_tracking
- v_\$controlfile
- v_\$database
- v_\$datafile
- v_\$datafile_header
- v_\$diag_alert_text
- v_\$dispatcher
- v_\$latch
- v_\$librarycache
- v_\$log
- v_\$log_history
- v_\$logfile
- v_\$open_cursor
- v_\$parameter
- v_\$resource_limit
- v_\$rman_backup_job_details
- v_\$rollstat
- v_\$rowcache
- v_\$session
- v_\$sesstat
- v_\$statname
- v_\$sysstat
- v_\$tablespace
- v_\$tempfile
- v_\$version
- v_\$asm_client (required for ASM Dynamic Applications)
- v_\$asm_disk (required for ASM Dynamic Applications)
- v_\$asm_diskgroup (required for ASM Dynamic Applications)
- v_\$recovery_file_dest(required for ASM/RAC Dynamic Applications)

Required Privileges

All Oracle database users must have the following privileges:

- System Privileges: CREATE SESSION
- Role Privileges: SELECT_CATALOG_ROLE
- Table Privileges:
 - SELECT ON SYS.V_\$DIAG_ALERT_EXT
 - SELECT ON SYS.TS\$
- Other: Permission to alter sessions to alter sessions.

Monitoring Pluggable Databases (PDBs)

To monitor pluggable databases (PDBs), you must have permissions to view `DBA_PDBS` and `V_\$PDBS`.

NOTE: You can create multiple users to access a CDB and all PDBs by creating a minimum of two Database Credentials, one for the CDB, and one for each PDB.

Creating Users and Assigning Privileges

If you want to monitor a container database (CDB) and PDB, log in to the CDB and create a user and grant access to containers using the following permissions:

- Create a user: CREATE USER C##[USERNAME] IDENTIFIED BY [PASSWORD] CONTAINER=ALL;
- Grant access to query tables: GRANT CREATE SESSION TO C##[USERNAME] CONTAINER=ALL;
- Grant access to all common tables and views on a CDB: GRANT SELECT_CATALOG_ROLE TO C##[USERNAME] CONTAINER=ALL;
- Grant access for "Oracle: DB Tablespace Temp Stats" Dynamic Application: GRANT SELECT ON SYS.TS\$ TO C##[USERNAME] CONTAINER=ALL;
- Grant access for "Oracle: DB Log Alerts Config" Dynamic Application: GRANT SELECT ON SYS.V_\$DIAG_ALERT_EXT TO C##[USERNAME] CONTAINER=ALL;
- Show PDBs:
 - ALTER USER C##[USERNAME] SET CONTAINER_DATA=(CDB\$ROOT,PDB1,PDB2) FOR DBA_PDBS CONTAINER=CURRENT;
 - ALTER USER C##[USERNAME] SET CONTAINER_DATA=(CDB\$ROOT,PDB1,PDB2) FOR V_\$PDBS CONTAINER=CURRENT;

For Non-Container Databases (Non-CDBs):

- Create a user: CREATE USER [USERNAME] IDENTIFIED BY [PASSWORD];
- Grant access to query tables: GRANT CREATE SESSION TO [USERNAME];
- Grant access to all common tables and views on a CDB: GRANT SELECT_CATALOG_ROLE TO [USERNAME];

- Grant access for "Oracle: DB Tablespace Temp Stats" Dynamic Application: GRANT SELECT ON SYS.TS\$ TO [USERNAME];
- Grant access for "Oracle: DB Log Alerts Config" Dynamic Application: GRANT SELECT ON SYS.V_\$DIAG_ALERT_EXT TO [USERNAME];

NOTE: If you are monitoring RAC and ASM instances, no additional permissions are required.

NOTE: Each Oracle database user will need a corresponding SL1 database credential for database access.

Prerequisites to Monitor Using Transport Layer Security (TLS/TCPS)

To monitor using TCPS you must configure the collector with Oracle Instant Client, Oracle Database libraries, wallet and network files.

NOTE: You need to configure every collector that the PowerPack will use. Alternatively, you can change the Collector Affinity of every Dynamic Application to use the "root device's collector" to use only one single collector.

For the workflow below, assume the following example:

	Oracle Database	SL1 Collector
Version	Oracle Database 21c	Oracle Instant Client 21.15
ORACLE_HOME	/u01/app/oracle/product/21.0.0/dbhome_1	/opt/oracle/instant_client_21_15

1. Download the Oracle Instant Client that matches your Oracle Database version.
2. Install the Oracle Instant Client on your SL1 Collector, typically at `/opt/oracle`.
3. Using SSH, connect to your Oracle Database and copy the following folders:
 - `$ORACLE_HOME/jdk`
 - `$ORACLE_HOME/jlib`
 - `$ORACLE_HOME/bin/orapki`
4. Transfer the folders to your SL1 Collector:
 - Copy `jdk` to `/opt/oracle/jdk`
 - Copy `jlib` to `/opt/oracle/jlib`
 - Copy `orapki` to `/opt/oracle/instant_client_21_15/orapki`
5. Confirm proper installation by running the following commands on your collector:

```
which orapki
```

```
which sqlplus
```

6. Create a configuration file called "sqlnet.ora" at the location `/opt/oracle/instant_client_21_15/network/admin/sqlnet.ora`.
7. Create a network file called "tnsnames.ora" at the location `/opt/oracle/instant_client_21_15/network/admin/tnsnames.ora`. The contents of this file will control your connections from SL1 to Oracle: Database.
8. On your collector, create an Oracle wallet for SL1 using the following command:

```
orapki wallet create -wallet /opt/oracle/instant_client_21_15/wallet -pwd  
YOUR_PASSWORD -auto_login
```

IMPORTANT: Use the `-auto_login` option so Python can access the wallet. Do not use `-auto_login_local`, as this will cause issues.

9. Create or add your SSL certificate to the SL1 wallet. The certificate must have a key size of 2048 bits or greater.
10. Exchange certificates between the Oracle Database wallet and the SL1 wallet if needed.
11. Change the collector's Instant Client owner and permissions for monitoring:
 - `sudo chown -R 'em7admin:s-em7-core' /opt/oracle`
 - `chmod -v 640 /opt/oracle/instant_client_21_15/wallet/*`

Monitoring TCPS Use Case

Assume this Oracle DB:

Instance	Host	Port	Protocol
cdb1 (SID)	192.0.2.110	2484	TCPS
sil0_pdb (SERVICE_NAME)	192.0.2.110	2484	TCPS

You must configure the collector's `tnsnames.ora` file to point to these instances. The `tnsnames.ora` file should have a format similar to the following:

```
AN_ALIAS_YOU_LIKE=
```

```
(DESCRIPTION=
```

```
(ADDRESS=
```

```
(PROTOCOL=TCPS)
```

```
(HOST=YOUR_ORACLE_ADDRESS)
```

```
(PORT=YOUR_ORACLE_PORT)
)
(CONNECT_DATA=(SERVER=dedicated)
(SID=YOUR_ORACLE_SID)
)
)
```

Modifying the collector's SL1 tnsnames.ora file with the example Oracle Database above, the tnsnames.ora file should look like this:

```
cdb1_ssl=
(DESCRIPTION=
(AADDRESS=
(PROTOCOL=TCPS)
(HOST=192.0.2.110)
(PORT=2484)
)
(CONNECT_DATA=(SERVER=dedicated)
(SID=cdb1)
)
)
silo_pdb_tcps=
(DESCRIPTION=
(AADDRESS=
(PROTOCOL=TCPS)
(HOST=192.0.2.110)
(PORT=2484)
```

```
)
(CONNECT_DATA=(SERVER=dedicated)

(SERVICE_NAME=silo_pdb)

)

)
```

After configuring the tnsnames.ora file, fill the SL1 database credential with the following:

1. In the **Oracle Connect Type** field, enter "Oracle Real Application Clusters (SERVICE_NAME)"
2. In the **Oracle Database SID** field, enter the tnsnames.ora alias you created above. For example, "cdb1_ ssl" or "silo_pdb_tcps".
3. Complete the other fields of the Database Credential as described in the [Creating a Database Credential](#) section.

Configuring Oracle Credentials

To monitor Oracle Database instances using SL1, you must create at least three credentials. The types of credentials that are required for monitoring depend on the type of server that is hosting the Oracle Database:

- Linux and Unix users must use an [SSH/Key credential](#) for access
- Windows users must use a [PowerShell credential](#).
- You must also use a [Database credential](#) for each CDB, PDB, Non-CDB, or ASM
- All users must use a [SOAP/XML credential](#) to link all credentials

NOTE: The SOAP/XML credential is the only credential that is used for discovery.

Suggested Timeout Configuration

There are current platform limitations to implementing a timeout for Database Server sessions.

To prevent issues and perform the suggested configuration:

1. Create a new profile for the DB user in the Database.
2. Configure the profile with these queries:
 - ```
ALTER PROFILE <profile> LIMIT IDLE_TIME 3
```
  - ```
ALTER PROFILE <profile> LIMIT CONNECT_TIME 3
```

With this configuration implemented, a three-minute timeout session is established. This allows the user time to stop the collection if it prevents SL1 from creating SIGTERMs.

Creating an SSH/Key Credential

Linux and Unix users must create an SSH/Key credential.

To create an SSH/Key credential:

1. Go to the **Credentials** page (Manage > Credentials).
2. Locate the "Oracle: DB Example SSH" sample credential, then click its **[Actions]** icon (⋮) and select **Duplicate**. A copy of the credential, called **Oracle: DB Example SSH copy** appears.

3. Supply values in the following fields:
 - **Name**. Type a new name for the credential.
 - **All Organizations**. Toggle on (blue) to align the credential to all organizations, or toggle off (gray) and then select one or more specific organizations from the **What organization manages this service?** drop-down field to align the credential with those specific organizations.
 - **Hostname/IP**. Type "%D" or the IP address of the server that is hosting the Oracle Database.
 - **Port**. Type "22".
 - **Username**. Type the username for the Linux server that is hosting the Oracle Database.
 - **Password**. Type the password for the Linux server that is hosting the Oracle Database.
 - **Private Key (PEM Format)**. Optional. Use if required for SSH authentication. For information on gathering a private key, see the section on [Enabling PEM on a Linux Machine](#).


NOTE: The private key can have a maximum of 64 characters per line. Therefore, you cannot use keys in the OpenSSH format, because that format uses 70 characters per line. When you attempt to save the credential, SL1 will validate that the private key entered is in the correct format. You will be able to save the credential only if the private key is correctly formatted.

4. Click **[Save & Close]**.

NOTE: The credential ID will appear in the ID column of the Credentials page after it has been saved. Take note of the ID as you will need it when creating the SOAP/XML credential.

Creating an SSH/Key Credential in the Classic SL1 User Interface

To create an SSH/Key credential :

1. Go to the **Credential Management** page (System > Manage > Credentials).
2. Click the wrench icon () for the "Oracle: DB Example SSH" credential. The **Credential Editor** modal page appears.
3. Supply values in the following fields:
 - **Credential Name.** Type a new name for the credential.
 - **Hostname/IP.** Type "%D" or the IP address of the server that is hosting the Oracle Database.
 - **Port.** Type the default port for SSH (22).
 - **Username.** Type the username for the server that is hosting the Oracle Database.
 - **Password.** Type the password for the server that is hosting the Oracle Database.
 - **Private Key (PEM Format).** Optional. Use if required for SSH authentication. For information on gathering a private key, see the section on [Enabling PEM on a Linux Machine](#).

NOTE: The private key can have a maximum of 64 characters per line. Therefore, you cannot use keys in the OpenSSH format, because that format uses 70 characters per line. When you attempt to save the credential, SL1 will validate that the private key entered is in the correct format. You will be able to save the credential only if the private key is correctly formatted.

4. Click the **[Save As]** button.
5. When the confirmation message appears, click **[OK]**.

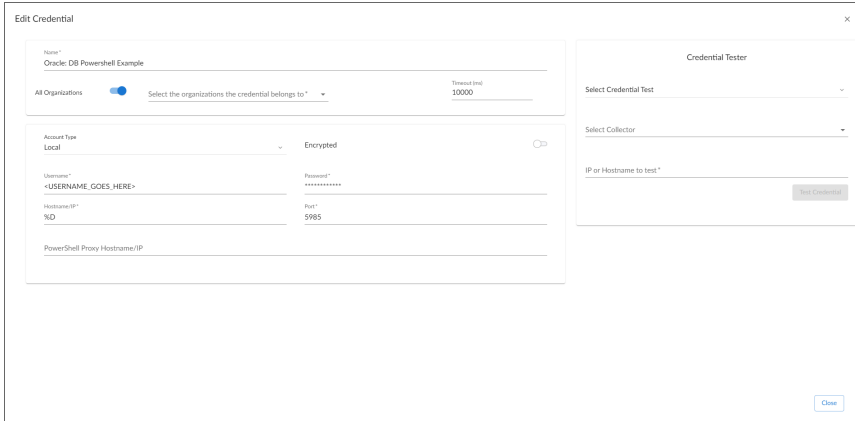
NOTE: The credential ID will appear at the top of the window after it has been saved. Take note of the ID as you will need it when creating the SOAP/XML credential.

Creating a PowerShell Credential (Windows Users)

Windows users must create a PowerShell credential.

To create a PowerShell credential:

1. Go to the **Credentials** page (Manage > Credentials).
2. Locate the "Oracle: DB Example PSH" sample credential, then click its **[Actions]** icon (⋮) and select **Duplicate**. A copy of the credential, called **Oracle: DB Example PSH copy** appears.



3. Supply values in the following fields:

- **Name**. Type a new name for the credential.
- **All Organizations**. Toggle on (blue) to align the credential to all organizations, or toggle off (gray) and then select one or more specific organizations from the **What organization manages this service?** drop-down field to align the credential with those specific organizations.
- **Timeout (ms)**. Time, in milliseconds, after which SLI will stop trying to communicate with the authenticating server. For collection to be successful, SLI must connect to the authenticating server, execute the PowerShell command, and receive a response within the amount of time specified in this field.
- **Account Type**. Select *Local*. However, if you plan to host an Oracle Database in a server that is part of an Active Directory, select *Active Directory* and configure a user without admin permissions.

NOTE: For ease of configuration, ScienceLogic recommends using an Active Directory account that is a member of the local Administrators group.

- **Hostname/IP**. Hostname/IP of the AD Server, not the server's IP that is part of the AD.
- **Username**. Type the username for the Windows server that is hosting the Oracle Database.
- **Password**. Type the password for the Windows server that is hosting the Oracle Database.

- **Encrypted.** Select whether SL1 will communicate with the device using an encrypted connection:
 - Toggle on (blue) if SL1 will communicate with the device using an encrypted connection. If the connection is encrypted, when communicating with the Windows server, SL1 will use a local user account with authentication of type "Basic Auth". You must then use HTTPS and can use a Microsoft Certificate or a self signed certificate.
 - Toggle off (gray) if the connection is not encrypted. If the connection is not encrypted, when communicating with the Windows server, SL1 will not encrypt the connection.
- **Port.** Type "5985" (http) or "5986" (https).
- **PowerShell Proxy Hostname/IP.** Leave this field blank.
- **Active Directory Host/IP.** If you selected Active Directory in the **Account Type** field, type the hostname or IP address of the Active Directory server that will authenticate the credential.
- **Active Directory Domain.** If you selected Active Directory in the **Account Type** field, type the domain where the monitored Windows device resides.

4. Click **[Save & Close]**.

NOTE: The credential ID will appear in the ID column of the Credentials page after it has been saved. Take note of the ID as you will need it when creating the SOAP/XML credential.

If you do not have Local Administrator access to the servers that you want to monitor with PowerShell or WinRM, or if the monitored Windows server is a Domain Controller that will not be in the local Administrators group, then you must first create a domain user account or create a local user account on the Windows Server.


To configure Windows Servers to allow access by your non-administrator user account:

1. See **Option 3: Creating a Non-Administrator User Account** in the **Configuring Windows Servers for Monitoring with PowerShell** manual, and follow Option 3's steps.
2. Configure a Server Authentication Certificate. See **Step 2: Configuring a Server Authentication Certificate** in the same manual to follow steps if needed.
3. Configure a Windows Remote Management. See **Step 3: Configuring Windows Remote Management** and follow the "Option 1: Using a Script to Configure Windows Remote Management" instructions.

Creating a PowerShell Credential (Windows Users) in the Classic SL1 User Interface

Windows users must create a PowerShell credential.

To create a PowerShell credential:

1. Go to the **Credential Management** page (System > Manage > Credentials).
2. Click the wrench icon () for the "Oracle: DB Example PSH" credential. The **Credential Editor** modal page appears.

3. Supply values in the following fields:

- **Profile Name.** Type a new name for the credential.
- **Account Type.** Select *Local*. However, if you plan to host an Oracle Database in a server that is part of an Active Directory, select Active Directory and configure a user without admin permissions.

NOTE:For ease of configuration, ScienceLogic recommends using an Active Directory account that is a member of the local Administrators group.

- **Active Directory Settings.** Hostname/IP of the AD Server, not the server's IP that is part of the AD.
- **Domain.** Domain the user is in.
- **Hostname/IP.** Type "%D" or the IP address of the server that is hosting the Oracle Database.
- **Timeout (ms).** Type the time, in milliseconds, after which SL1 will stop trying to collect data from the authenticating server. For collection to be successful, SL1 must connect to the authenticating server, execute the PowerShell command, and receive a response within the amount of time specified in this field.
- **Username.** Type the username for the Windows server that is hosting the Oracle Database.
- **Password.** Type the password for the Windows server that is hosting the Oracle Database.
- **Encrypted.** Select whether SL1 will communicate with the device using an encrypted connection. Choices are:
 - *yes.* When communicating with the Windows server, SL1 will use a local user account with authentication of type "Basic Auth". You must then use HTTPS and can use a Microsoft Certificate or a self-signed certificate.
 - *no.* When communicating with the Windows server, SL1 will not encrypt the connection.
- **Port.** Type "5985" (http) or "5986" (https).
- **PowerShell Proxy Hostname/IP.** Leave this field blank.

4. Click the **[Save As]** button.

NOTE: The credential ID will appear at the top of the window after it has been saved. Take note of the ID as you will need it when creating the SOAP/XML credential.

If you do not have Local Administrator access to the servers that you want to monitor with PowerShell or WinRM, or if the monitored Windows server is a Domain Controller that will not be in the local Administrators group, then you must first create a domain user account or create a local user account on the Windows Server.

To configure Windows Servers to allow access by your non-administrator user account:

1. See **Option 3: Creating a Non-Administrator User Account** in the *Configuring Windows Servers for Monitoring with PowerShell* manual, and follow Option 3's steps.
2. Configure a Server Authentication Certificate. See **Step 2: Configuring a Server Authentication Certificate** in the same manual to follow steps if needed.
3. Configure a Windows Remote Management. See **Step 3: Configuring Windows Remote Management** and follow the "Option 1: Using a Script to Configure Windows Remote Management" instructions.

Creating a Database Credential

To monitor multiple CDB or PDB instances, you must create a database credential.

To create a database credential:

1. Go to the **Credentials** page (Manage > Credentials).
2. Click the **[Create New]** button and then select *Create Database Credential*. The **Create Credential** modal page appears:

The screenshot shows a 'Create Credential' modal window. On the left, there are input fields for 'Name', 'All Organizations' (a toggle), 'What organization manages this service?' (a dropdown), 'Timeout (ms)' (1500), 'Database Type' (Oracle & *SQLNet), 'Database Name', 'Database User', 'Password', 'Hostname/IP', and 'Oracle System Identifier (SID)'. On the right, there is a 'Credential Tester' section with 'Select Credential Test' and 'Select Credential' dropdowns, and a 'Test Credential' button. At the bottom right, there is a 'Save & Close' button.

3. Supply values in the following fields:
 - **Name**. Type a name for the credential.
 - **All Organizations**. Toggle on (blue) to align the credential to all organizations, or toggle off (gray) and then select one or more specific organizations from the **What organization manages this service?** drop-down field to align the credential with those specific organizations.
 - **Timeout (ms)**. Type a time, in milliseconds, after which SL1 will stop trying to communicate with the database.
 - **Database Type**. Select *Oracle & *SQLNet*.

- **Database Name.** Type the name of the database parent. If the database does not have a parent, type the name of the database to access.
 - To discover CDBs, type the CDB's SID.
 - To discover PDBs, type the SID of the parent CDB.
 - To discover ASMs, type the SID of the ASM instance
- **DB User.** Type the username for the Oracle Database account.
 - To discover CDBs, type the instance username.
 - To discover PDBs, type the PDB instance username.
 - To discover ASMs, use the recommended "ASMSNMP" username.
- **Password.** Type the password for the Oracle Database account.
 - To discover CDBs, type the instance password.
 - To discover PDBs, type the PDB instance password.
 - To discover ASMs, type the ASM instance password.
- **Hostname/IP.** Type "%D" or the IP address where the database resides.
- **Port.** Type the port number associated with the database you want to access with this credential. For the *Oracle and *SQLNet* database type, the default value is 1521.
 - To discover CDBs, type the listener port value.
 - To discover PDBs, type the PDB listener port value.
 - To discover ASMs, type the ASM listener port value.

Oracle Settings

- **Oracle Connect Type.** Select the method SL1 should use to connect to the Oracle database, depending on your Oracle database setup. The choices supported by this PowerPack are:
 - *Oracle System Identifier (SID).* Select this option if you want to discover a CDB or Non-CDB instance. PDBs do not typically have a SID.
 - *Oracle Real Application Clusters (SERVICE_NAME).* Select this option if you want to discover a PDB instance, or if your setup uses SERVICE_NAME. Ensure that SERVICE_NAME points to a single instance for more precise monitoring.

NOTE: ScienceLogic recommends selecting "Oracle System Identifier (SID) for better instance control, or "Oracle Real Application Clusters (SERVICE_NAME) if mapping to a single instance. The "Oracle Real Application Clusters (SERVICE_NAME)" field represents SERVICE_NAME, regardless of whether the database is RAC or not.

- **Oracle Database SID.** Type the SID or SERVICE_NAME, depending on the Oracle Connect Type selected in the **Oracle Connect Type** field.
 - To discover CDBs, type the instance SID.
 - To discover PDBs, type the PDB instance SERVICE_NAME.
 - To discover ASMs, type the ASM instance SID
 - To discover instances using TCPS, type the alias in your SL1 tnsnames.ora file for the corresponding instance. Be sure to check the prerequisites for monitoring TCPS before choosing this option.

NOTE: Any PDB instance SERVICE_NAME or SID that you entered in the database credential should match the PDB alias in your Oracle DB tnsnames.ora file.

4. Click **[Save & Close]**.

NOTE: The credential ID will appear in the ID column of the Credentials page after it has been saved. Take note of the ID as you will need it when creating the SOAP/XML credential.

Creating a Database Credential in the Classic SL1 User Interface

You must create a database credential if you want to monitor multiple CDB or PDB instances .

To create a database credential:

1. Go to the **Credential Management** page (System > Manage > Credentials).
2. In the **Credential Management** page, click the **[Actions]** menu. Select **Create Database Credential**.
3. The **Credential Editor** modal page appears. In this page, you can define the new database credential. To define the new credential, supply values in the following fields:

Basic Settings

- **Profile Name.** Type a name for the credential.
- **DB Type.** Select *Oracle & *SQLNet*.
- **Database Name.** Type the name of the database parent. If the database does not have a parent, type the name of the database to access.
 - To discover CDBs, type the CDB's SID.
 - To discover PDBs, type the SID of the parent CDB.
 - To discover ASMs, type the SID of the ASM instance
- **DB User.** Type the username for the Oracle Database account.
 - To discover CDBs, type the instance username.
 - To discover PDBs, type the PDB instance username.

- To discover ASMs, use the recommended "ASMSNMP" username.
- **Password.** Type the password for the Oracle Database account.
 - To discover CDBs, type the instance password.
 - To discover PDBs, type the PDB instance password.
 - To discover ASMs, type the ASM instance password.
- **Hostname/IP.** Type "%D" or the IP address where the database resides.
- **Port.** Type the port number associated with the database you want to access with this credential. For the *Oracle and *SQLNet* database type, the default value is 1521.
 - To discover CDBs, type the listener port value.
 - To discover PDBs, type the PDB listener port value.
 - To discover ASMs, type the ASM listener port value.

Oracle Settings

- **Oracle Connect Type.** Select the method SL1 should use to connect to the Oracle database, depending on your Oracle database setup. The choices supported by this PowerPack are:
 - *Oracle System Identifier (SID).* Select this option if you want to discover a CDB or Non-CDB instance. PDBs do not typically have a SID.
 - *Oracle Real Application Clusters (SERVICE_NAME).* Select this option if you want to discover a PDB instance, or if your setup uses SERVICE_NAME. Ensure that SERVICE_NAME points to a single instance for more precise monitoring.

NOTE: ScienceLogic recommends selecting "Oracle System Identifier (SID) for better instance control, or "Oracle Real Application Clusters (SERVICE_NAME) if mapping to a single instance. The "Oracle Real Application Clusters (SERVICE_NAME)" field represents SERVICE_NAME, regardless of whether the database is RAC or not.

- **Oracle Database SID.** Type the SID or SERVICE_NAME, depending on the Oracle Connect Type selected in the **Oracle Connect Type** field.
 - To discover CDBs, type the instance SID.
 - To discover PDBs, type the PDB instance SERVICE_NAME.
 - To discover ASMs, type the ASM instance SID
 - To discover instances using TCPS, type the alias in your SL1 tnsnames.ora file for the corresponding instance. Be sure to check the prerequisites for monitoring TCPS before choosing this option.

NOTE: Any PDB instance SERVICE_NAME or SID that you entered in the database credential should match the PDB alias in your Oracle DB tnsnames.ora file.

NOTE: Any PDB instance SID that you entered in the database credential should match the SID in your tnsnames.ora file. Your SOAP/XML credential can have the alias of the PDB instance SID.

4. Click **[Save]**.

NOTE: The credential ID will appear at the top of the window after it has been saved. Take note of the ID as you will need it when creating the SOAP/XML credential.

Creating a SOAP/XML Credential

To create a SOAP/XML credential:

1. Go to the **Credentials** page (Manage > Credentials).
2. Locate the appropriate sample credential for your use case, then click its **[Actions]** icon (⋮) and select **Duplicate**. A copy of the credential with its name and copy as a suffix appears.

The screenshot shows the 'Edit Credential' dialog box. The main form contains the following fields and options:

- Name:** Oracle DB Example
- All Organizations:** Toggle on (blue)
- Timeout (ms):** 2000
- Content Encoding:** text/xml
- Method:** GET
- HTTP Version:** HTTP/1.1
- URL:** http://%D
- HTTP Auth User:** oracle@db_user_name
- HTTP Auth Password:** *****
- Proxy Hostname/IP:** optional
- Proxy Port:** 0
- Proxy User:** optional
- Proxy Password:** *****
- Embedded Password (EPS):** *****
- Embed Value [%1]:**
- Embed Value [%2]:**

The 'Credential Tester' section on the right includes:

- Select Credential Test:** dropdown menu
- Select Collector:** dropdown menu
- IP or Hostname to test:** text input field
- Test Credential:** button

A 'Close' button is located at the bottom right of the dialog.

3. Supply values in the following fields:

- **Name.** Type a new name for the credential.
- **All Organizations.** Toggle on (blue) to align the credential to all organizations, or toggle off (gray) and then select one or more specific organizations from the **What organization manages this service?** drop-down field to align the credential with those specific organizations.
- **URL.** Leave the default value of "https://%D".

- **HTTP Auth User.** This field is not required, but cannot be left blank. Enter any value here.
- **HTTP Auth Password.** This field is not required, but cannot be left blank. Enter any value here.
- **HTTP Headers.** Add the following headers by clicking **+ Add a header** and see the following workflow that describes header formats based on the Oracle devices that you are monitoring.

HTTP Headers

HTTP Headers allow you to register the Oracle Database servers you want to discover. Additionally, there are special keywords that can be used, described in the [HTTP Headers Special Keywords](#) section.

NOTE: A header should be added for each Oracle Database instance that you are monitoring.

NOTE: Only the SIDs listed in the credential will be discovered.

CAUTION: Do not end the HTTP header with a backslash.

Headers for Oracle DB Server OS:

- **SSH.** <cred_id> for the SSH/Key credential you created.

or

- **PSH.** <cred_id> for the PowerShell credential you created.

NOTE: Only one OS type is supported per credential.

Headers for PDBs

- **CDB SID.** The SID of the CDB where the PDB is plugged in.
- **PDB NAME.** Must always be in upper case. This can be retrieved with the following commands:

```
show pdbs
```

```
SELECT PDB_ID, DBID, GUID, PDB_NAME, STATUS FROM DBA_PDBS;
```

- **DB_CID.** The ID of the Database credentials for the CDB and PDB

For the SID, the CDB value will always be either "True" or "False". For a PDB, the CDB value will be the container SID.

Example Use Case 1, assuming this Oracle DB:

Instance	DB Credential ID	Parent CDB
cdb1	122	N/A
PDB1	123	cdb1
PDB2	23456	cdb1

```
SSH: 77
```

```
SID: cdb1:DB_CID:122:CDB:True
```

```
PDB: PDB1:DB_CID:123:CDB: cdb1
```

```
PDB: PDB2:DB_CID:23456:CDB: cdb1
```

Example Use Case 2, assuming this Oracle DB:

Instance	DB Credential ID	Parent CDB
cdb2	332	N/A
PDB3	333	cdb2

```
PSH: 78
```

```
SID: cdb2:DB_CID:332:CDB:true
```

```
PDB: PDB3:DB_CID:333:CDB: cdb2
```

Example Use Case 3, assuming this Oracle DB:

Instance	DB Credential ID	Parent CDB
silodb	1241	N/A
SILO_PDB55	1242	silodb

```
SSH: 55
```

```
SID: silodb:DB_CID:1241:CDB:true
```

```
PDB: SILO_PDB55:DB_CID:1242:CDB: silodb
```

Example Use Case 4, assuming this Oracle DB:

Instance	DB Credential ID	Parent CDB
cdb1	56	N/A
PDB1	111	cdb1

Instance	DB Credential ID	Parent CDB
PDB2	222	cdb1
PDB3	333	cdb1
cdb2	77	N/A
PDB1	90	cdb2

```
SSH:100
```

```
SID:cdb1:DB_CID:56:CDB:true
```

```
PDB:PDB1:DB_CID:111:CDB:cdb1
```

```
PDB:PDB2:DB_CID:222:CDB:cdb1
```

```
PDB:PDB3:DB_CID:333:CDB:cdb1
```

```
SID:cdb2:DB_CID:77:CDB:true
```

```
PDB:PDB1:DB_CID:90:CDB:cdb2
```

Example Use Case 5, assuming this Oracle DB:

Instance	DB Credential ID	Parent CDB
oraclecdb	56	N/A
PDB1	111	oraclecdb
PDB2	222	oraclecdb

```
PSH:99
```

```
SID:oraclecdb:DB_CID:56:CDB:true
```

```
PDB:PDB1:DB_CID:111:CDB:oraclecdb
```

```
PDB:PDB2:DB_CID:222:CDB:oraclecdb
```

IMPORTANT: ScienceLogic strongly recommends NOT including "TNS_ADMIN" in PDB headers.

CAUTION: Be sure to put `CDB:true` as the last element in the header. If you do not, the PowerPack may report incorrectly that the SOAP credential is incorrect.

NOTE: PDB headers should always be followed by their parent CDB.

Headers for RAC

This version of the Oracle: Database PowerPack is designed to monitor Oracle Real Application Clusters (RAC) environments. While comprehensive cluster-wide monitoring is limited, you can effectively monitor each RAC node individually by discovering and configuring them separately.

NOTE: This section includes instructions for discovering Container Databases (CDBs). If your cluster includes Pluggable Databases (PDBs), please refer to the [Monitoring Pluggable Databases \(PDBs\)](#) section after completing this section. This section is also applicable to non-CDB environments.

- **SID.** The node SID.
- **DB_CID.** The ID of the database credential you created.

Example Use Case 1, assuming this Oracle DB:

Instance	DB Credential ID	Parent CDB
cdbrac1	51	N/A
cdbrac2	52	N/A

IMPORTANT: OS information will only be gathered from one node.

```
SSH: 50
```

```
SID:cdbrac1:DB_CID:51
```

```
SID:cdbrac2:DB_CID:52
```

Example Use Case 2, assuming this Oracle DB:

Instance	DB Credential ID	Parent CDB
cdbrac1	51	N/A
PDB1	53	cdbrac1
cdbrac2	52	N/A
PDB1	54	cdbrac2

```
SSH: 50
```

SID:cdbrac1:DB_CID:51:CDB:true

PDB:PDB1:DB_CID:53:CDB:cdbrac1

SID:cdbrac2:DB_CID:52:CDB:true

PDB:PDB1:DB_CID:54:CDB:cdbrac2

Example Use Case 3, assuming this Oracle DB:

Instance	DB Credential ID	Parent CDB
cdbrac1	51	N/A
PDB1	53	cdbrac1
cdbrac2	52	N/A
PDB1	54	cdbrac2
cdbrac3	55	N/A
+ASM1	56	N/A
+ASM2	57	N/A

SSH:50

SID:cdbrac1:DB_CID:51:CDB:true

PDB:PDB1:DB_CID:53:CDB:cdbrac1

SID:cdbrac2:DB_CID:52:CDB:true

PDB:PDB1:DB_CID:54:CDB:cdbrac2

SID:cdbrac3:DB_CID:55

ASM:56

ASM:57

Example Use Case 4, assuming this Oracle DB and multiple SOAP/XML credentials to gather OS information from multiple nodes:

Instance	DB Credential ID	Parent CDB
cdbrac1	51	N/A
PDB1	53	cdbrac1
cdbrac2	52	N/A
PDB1	54	cdbrac2
cdbrac3	55	N/A

SOAP/XML Credential 1:

SSH:50

SID:cdbrac1:DB_CID:51:CDB:true

PDB:PDB1:DB_CID:53:CDB:cdbrac1

SOAP/XML Credential 2:

SSH:56

SID:cdbrac2:DB_CID:52:CDB:true

PDB:PDB1:DB_CID:54:CDB:cdbrac2

SOAP/XML Credential 3:

SSH:57

SID:cdbrac3:DB_CID:55

CAUTION: This configuration will likely cause duplicate collection and event generation.

Headers for Non-CDB

Headers for non-CDBs are handled like the headers for CDBs, but the PDB header is not included.

Example Use Case 1, assuming this Oracle DB:

Instance	DB Credential ID
rac1	51
rac2	52

SSH:50

SID:rac1:DB_CID:51

SID:rac2:DB_CID:52

Example Use Case 2, assuming this Oracle DB:

Instance	DB Credential ID
orcl	52

```
PSH:51
```

```
SID:orcl:DB_CID:52
```

Example Use Case 3, assuming this Oracle DB:

Instance	DB Credential ID
devdb	52
testdb	54
prod1	56
staging	59
qaenv	57

```
PSH:51
```

```
SID:devdb:DB_CID:52
```

```
SID:testdb:DB_CID:54
```

```
SID:prod1:DB_CID:56
```

```
SID:staging:DB_CID:59
```

```
SID:qaenv:DB_CID:57
```

Headers for ASM

Headers for ASMs only need to include the credential ID.

Example Use Case 1, assuming this Oracle DB:

Instance	DB Credential ID
cdbrac1	51
cdbrac2	52
+ASM1	53
+ASM2	54

```
SSH:50
```

```
SID:cdbrac1:DB_CID:51
```

```
SID:cdbrac2:DB_CID:52
```

```
ASM:53
```


ASM: 54

Example Use Case 2, assuming this Oracle DB:

Instance	DB Credential ID
orcl	52
+ASM1	53

PSH: 51

SID:orcl:DB_CID:52

ASM: 53

Headers for TCPS

To force the PowerPack use TCPS, you must add a header with "TCPS:/path/to/instant_client". By adding this header, every instance listed in the headers will reference the SL1 tnsnames.ora file, so it is important to ensure that all required instances are included in the tnsnames.ora file.

Example Use Case 1, assuming this Oracle DB:

- SL1 Collector's Oracle Instant Client path: /opt/oracle/instant_client_21_15

Instance	DB Credential ID	Parent CDB
cdbrac1	51	N/A
PDB1	53	cdbrac1

SSH: 50

SID:cdbrac1:DB_CID:51:CDB:true

PDB:PDB1:DB_CID:53:CDB:cdbrac1

TCPS:/opt/oracle/instant_client_21_15

Example Use Case 2, assuming this Oracle DB:

- SL1 Collector's Oracle Instant Client path: /opt/oracle/instant_client_19_24

Instance	DB Credential ID	Parent CDB
orcl	52	N/A
PDBPROD	54	orcl
+ASM1	53	N/A

PSH: 51

```
SID:orcl:DB_CID:52:CDB:true
```

```
PDB:PDBPROD:DB_CID:54:CDB:orcl
```

```
ASM:53
```

```
TCPS:/opt/oracle/instant_client_19_24
```

HTTP Headers Special Keywords

This section contains special keywords you can use to modify or enhance the discovery and data collection process, giving you greater control over how your Oracle Database instances are managed.

TNS_ADMIN

The `TNS_ADMIN` parameter is used to specify a custom path for the `tnsnames.ora` file, which overrides the default location (`$Oracle_HOME/network/admin`). This is useful if your configuration files are stored in a non-standard directory. By including a standalone `TNS_ADMIN` header, you instruct the PowerPack to apply the specified custom path to all SIDs listed under the corresponding SOAP/XML credential.

IMPORTANT: Do not end the path specified in `TNS_ADMIN` with a backslash (`/`), as this can cause configuration errors.

NOTE: `TNS_ADMIN` and `GRID_PATH` are not supported for Windows devices.

Example Use Case 1:

```
SSH:50
```

```
SID:cdbrac1:DB_CID:51
```

```
SID:cdbrac2:DB_CID:52
```

```
TNS_ADMIN:/u01/app/oracle/network
```

Example Use Case 2:

```
SSH:51
```

```
SID:orcl:DB_CID:52
```

```
TNS_ADMIN:/home/oracle
```

TNS_ADMIN Within a SID Header

Including the TNS_ADMIN parameter within a SID header directs the PowerPack to use a custom path for the tnsnames.ora file specifically for that SID. This allows precise control over which SIDs use a non-default location for their network configuration files.

IMPORTANT: If the CDB header requires `CDB:true`, ensure this is always the last part of the header.

IMPORTANT: The TNS_ADMIN parameter is not supported for PDB headers.

Example Use Case 1:

- In this case, only "cdbrac1" uses the custom path `/u01/app/oracle/network`, while "cdbrac2" uses the default location.

```
SSH: 50
```

```
SID:cdbrac1:DB_CID:51:TNS_ADMIN:/u01/app/oracle/network
```

```
SID:cdbrac2:DB_CID:52
```

Example Use Case 2:

- In this case, "orcl" and "silo_cdb" use custom paths, while "mycdb" uses the default location.

```
SSH: 51
```

```
SID:orcl:DB_CID:52:TNS_ADMIN:/home/slmon
```

```
SID:mycdb:DB_CID:53
```

```
SID:silo_cdb:DB_CID:54:TNS_ADMIN:/home/oracle/network/admin
```

Example Use Case 3:

- In this case, "orclcdb" is configured with a custom TNS_ADMIN path and marked as a CDB (`CDB:true`). The associated PDBs, PDB1 and PDB2, are linked to "orclcdb" but do not utilize the TNS_ADMIN parameter.

```
SSH: 99
```

```
SID:orclcdb:DB_CID:56:TNS_ADMIN:/slmon/oracle:CDB:true
```

```
PDB:PDB1:DB_CID:111:CDB:orclcdb
```

```
PDB:PDB2:DB_CID:222:CDB:orclcdb
```

GRID_PATH

The GRID_PATH header is used to specify the location of the GRID_HOME directory and locate Oracle listeners, particularly for RAC (Real Application Clusters) or ASM (Automatic Storage Management) environments. Additionally, if the PowerPack is unable to execute the lsnrctl status command using the default listener path, it will attempt to use the grid_path specified in the HTTP header as an alternative.

NOTE: TNS_ADMIN and GRID_PATH are not supported for Windows devices.

Example Use Case 1:

- In this case, if the default listener path fails, the PowerPack will attempt to locate the listeners using the GRID_PATH provided.

```
SSH: 50
```

```
SID: cdbrac1:DB_CID:51:TNS_ADMIN:/u01/app/oracle/network
```

```
SID: cdbrac2:DB_CID:52
```

```
GRID_PATH:/u01/app/grid
```

Example Use Case 2:

- If the initial attempt to use default listener path is unsuccessful, the PowerPack will fall back to use the specified GRID_PATH.

NOTE: Ensure that the GRID_PATH provided accurately reflects the location of the Grid Infrastructure on your system to facilitate correct listener management for RAC or ASM environments.

```
SSH: 50
```

```
SID: orcl_rac_01:DB_CID:51
```

```
SID: orcl_rac_02:DB_CID:52
```

```
GRID_PATH:/u01/app/grid
```

TYPE_OF_OS

The TYPE_OF_OS header is used to address compatibility issues when the PowerPack encounters difficulties determining the operating system, or when an OS is part of a supported family but is not explicitly recognized. This header helps ensure that the PowerPack operates correctly by specifying the OS type manually.

The following values are supported:

- linux
- solaris
- aix
- windows
- ubuntu

Example Use Case 1:

- In this case, the TYPE_OF_OS is specified as linux, ensuring proper compatibility with the PowerPack.

```
SSH:50
```

```
SID:cdbrac1:DB_CID:51:TNS_ADMIN:/u01/app/oracle/network
```

```
SID:cdbrac2:DB_CID:52
```

```
TYPE_OF_OS:linux
```

Example Use Case 2:

- In this case, the TYPE_OF_OS is set to windows, which helps the PowerPack handle any compatibility issues with the Windows operating system.

```
PSH:51
```

```
SID:orcl:DB_CID:52
```

```
SID:mycdb:DB_CID:53
```

```
SID:silo_cdb:DB_CID:54
```

```
TYPE_OF_OS:windows
```

Example Use Case 3:

- In this case, the TYPE_OF_OS is set to windows, and is applied to the CDB and its associated PDBs to ensure compatibility.

```
PSH:99
```

```
TYPE_OF_OS:windows
```

```
SID:orclcdb:DB_CID:56:CDB:true
```

```
PDB:PDB1:DB_CID:111:CDB:orclcdb
```

```
PDB:PDB2:DB_CID:222:CDB:orclcdb
```

Example Use Case 4:

- In this case, `TYPE_OF_OS` is set to `ubuntu`, which is used to address any issues related to the Ubuntu OS.

```
TYPE_OF_OS:ubuntu
```

```
SSH:50
```

```
SID:cdbrac1:DB_CID:51
```

```
SID:cdbrac2:DB_CID:52
```

Deprecated HTTP Configuration Headers


The following headers have been deprecated and will no longer function:

- **HOST**. This header was previously used to specify a default host for all SIDs listed under a SOAP/XML credential. This was useful if your Oracle DB was hosted on a different machine than the OS server.
- **PORT**. This header was used to specify the port number for database instances in the SOAP/XML credential. This method is retained for compatibility but is now outdated. ScienceLogic recommends using `DB_CID` for new configurations.
- **SERVICE_NAME**. This header was used to specify the service name of the Oracle database. This header was part of an older configuration method and is now obsolete with the introduction of `DB_CID`. ScienceLogic does not recommend using `SERVICE_NAME` with `DB_CID`.

4. Click **[Save & Close]**.

Creating a SOAP/XML Credential in the Classic SL1 User Interface

To create the SOAP/XML credential:

1. Go to the **Credential Management** page (System > Manage > Credentials).
2. Click the wrench icon () for either the chosen sample credential for Windows users. The **Credential Editor** modal page appears.
3. Update the values in the following fields:

Basic Settings

- **Profile Name**. Type a new name for the credential.
- **URL**. Leave the default value of `https://%D`.
- **HTTP Auth User**. This field is not required, but cannot be left blank. Enter any value here.
- **HTTP Auth Password**. This field is not required, but cannot be left blank. Enter any value here.

NOTE: Discovering multiple instances on a single database server is supported, but all instances must share the same credentials entered in the SOAP/XML credential's *HTTP Auth User* and *HTTP Auth Password* fields.

- **HTTP Headers.** Add the following headers by clicking **+ Add a header** and see the following workflow that describes header formats based on the Oracle devices that you are monitoring:

HTTP Headers

HTTP Headers allow you to register the Oracle Database servers you want to discover. Additionally, there are special keywords that can be used, described in the [HTTP Headers Special Keywords](#) section.

NOTE: A header should be added for each Oracle Database instance that you are monitoring.

NOTE: Only the SIDs listed in the credential will be discovered.

CAUTION: Do not end the HTTP header with a backslash.

Headers for Oracle DB Server OS:

- **SSH.** <cred_id> for the SSH/Key credential you created.
- or
- **PSH.** <cred_id> for the PowerShell credential you created.

NOTE: Only one OS type is supported per credential.

Headers for PDBs

- **CDB SID.** The SID of the CDB where the PDB is plugged in.
- **PDB NAME.** Must always be in upper case. This can be retrieved with the following commands:

```
show pdbs
```

```
SELECT PDB_ID, DBID, GUID, PDB_NAME, STATUS FROM DBA_PDBS;
```

- **DB_CID.** The ID of the Database credentials for the CDB and PDB

For the SID, the CDB value will always be either "True" or "False". For a PDB, the CDB value will be the container SID.

Example Use Case 1, assuming this Oracle DB:

Instance	DB Credential ID	Parent CDB
cdb1	122	N/A
PDB1	123	cdb1
PDB2	23456	cdb1

```
SSH: 77
```

```
SID: cdb1:DB_CID:122:CDB:True
```

```
PDB: PDB1:DB_CID:123:CDB: cdb1
```

```
PDB: PDB2:DB_CID:23456:CDB: cdb1
```

Example Use Case 2, assuming this Oracle DB:

Instance	DB Credential ID	Parent CDB
cdb2	332	N/A
PDB3	333	cdb2

```
PSH: 78
```

```
SID: cdb2:DB_CID:332:CDB:true
```

```
PDB: PDB3:DB_CID:333:CDB: cdb2
```

Example Use Case 3, assuming this Oracle DB:

Instance	DB Credential ID	Parent CDB
silodb	1241	N/A
SILO_PDB55	1242	silodb

```
SSH: 55
```

```
SID: silodb:DB_CID:1241:CDB:true
```

```
PDB: SILO_PDB55:DB_CID:1242:CDB: silodb
```

Example Use Case 4, assuming this Oracle DB:

Instance	DB Credential ID	Parent CDB
cdb1	56	N/A
PDB1	111	cdb1
PDB2	222	cdb1

Instance	DB Credential ID	Parent CDB
PDB3	333	cdb1
cdb2	77	N/A
PDB1	90	cdb2

```
SSH:100
```

```
SID:cdb1:DB_CID:56:CDB:true
```

```
PDB:PDB1:DB_CID:111:CDB:cdb1
```

```
PDB:PDB2:DB_CID:222:CDB:cdb1
```

```
PDB:PDB3:DB_CID:333:CDB:cdb1
```

```
SID:cdb2:DB_CID:77:CDB:true
```

```
PDB:PDB1:DB_CID:90:CDB:cdb2
```

Example Use Case 5, assuming this Oracle DB:

Instance	DB Credential ID	Parent CDB
oracledb	56	N/A
PDB1	111	oracledb
PDB2	222	oracledb

```
PSH:99
```

```
SID:oracledb:DB_CID:56:CDB:true
```

```
PDB:PDB1:DB_CID:111:CDB:oracledb
```

```
PDB:PDB2:DB_CID:222:CDB:oracledb
```

IMPORTANT: ScienceLogic strongly recommends NOT including "TNS_ADMIN" in PDB headers.

CAUTION: Be sure to put `CDB:true` as the last element in the header. If you do not, the PowerPack may report incorrectly that the SOAP credential is incorrect.

NOTE: PDB headers must always be followed by their parent CDB.

Headers for RAC

This version of the Oracle: DatabasePowerPack is designed to monitor Oracle Real Application Clusters (RAC) environments. While comprehensive cluster-wide monitoring is limited, you can effectively monitor each RAC node individually by discovering and configuring them separately.

NOTE: This section includes instructions for discovering Container Databases (CDBs). If your cluster includes Pluggable Databases (PDBs), please refer to the [Monitoring Pluggable Databases \(PDBs\)](#) section after completing this section. This section is also applicable to non-CDB environments.

- **SID.** The node SID.
- **DB_CID.** The ID of the database credential you created.

Example Use Case 1, assuming this Oracle DB:

Instance	DB Credential ID	Parent CDB
cdbrac1	51	N/A
cdbrac2	52	N/A

IMPORTANT: OS information will only be gathered from one node.

```
SSH:50
```

```
SID:cdbrac1:DB_CID:51
```

```
SID:cdbrac2:DB_CID:52
```

Example Use Case 2, assuming this Oracle DB:

Instance	DB Credential ID	Parent CDB
cdbrac1	51	N/A
PDB1	53	cdbrac1
cdbrac2	52	N/A
PDB1	54	cdbrac2

```
SSH:50
```

```
SID:cdbrac1:DB_CID:51:CDB:true
```

```
PDB:PDB1:DB_CID:53:CDB:cdbrac1
```

```
SID:cdbrac2:DB_CID:52:CDB:true
```

```
PDB:PDB1:DB_CID:54:CDB:cdbrac2
```

Example Use Case 3, assuming this Oracle DB:

Instance	DB Credential ID	Parent CDB
cdbrac1	51	N/A
PDB1	53	cdbrac1
cdbrac2	52	N/A
PDB1	54	cdbrac2
cdbrac3	55	N/A
+ASM1	56	N/A
+ASM2	57	N/A

```
SSH:50
```

```
SID:cdbrac1:DB_CID:51:CDB:true
```

```
PDB:PDB1:DB_CID:53:CDB:cdbrac1
```

```
SID:cdbrac2:DB_CID:52:CDB:true
```

```
PDB:PDB1:DB_CID:54:CDB:cdbrac2
```

```
SID:cdbrac3:DB_CID:55
```

```
ASM:56
```

```
ASM:57
```

Example Use Case 4, assuming this Oracle DB and multiple SOAP/XML credentials to gather OS information from multiple nodes:

Instance	DB Credential ID	Parent CDB
cdbrac1	51	N/A
PDB1	53	cdbrac1
cdbrac2	52	N/A
PDB1	54	cdbrac2
cdbrac3	55	N/A

SOAP/XML Credential 1:

```
SSH:50
```

```
SID:cdbrac1:DB_CID:51:CDB:true
```

```
PDB:PDB1:DB_CID:53:CDB:cdbrac1
```

SOAP/XML Credential 2:

```
SSH:56
```

```
SID:cdbrac2:DB_CID:52:CDB:true
```

```
PDB:PDB1:DB_CID:54:CDB:cdbrac2
```

SOAP/XML Credential 3:

```
SSH:57
```

```
SID:cdbrac3:DB_CID:55
```

CAUTION: This configuration will likely cause duplicate collection and event generation.

Headers for Non-CDB

Headers for non-CDBs are handled like the headers for CDBs, but the PDB header is not included.

Example Use Case 1, assuming this Oracle DB:

Instance	DB Credential ID
rac1	51
rac2	52

```
SSH:50
```

```
SID:rac1:DB_CID:51
```

```
SID:rac2:DB_CID:52
```

Example Use Case 2, assuming this Oracle DB:

Instance	DB Credential ID
orcl	52

```
PSH:51
```

```
SID:orcl:DB_CID:52
```

Example Use Case 3, assuming this Oracle DB:

Instance	DB Credential ID
devdb	52
testdb	54
prod1	56
staging	59
qaenv	57

PSH: 51

SID:devdb:DB_CID:52

SID:testdb:DB_CID:54

SID:prod1:DB_CID:56

SID:staging:DB_CID:59

SID:qaenv:DB_CID:57

Headers for ASM

Headers for ASMs only need to include the credential ID.

Example Use Case 1, assuming this Oracle DB:

Instance	DB Credential ID
cdbrac1	51
cdbrac2	52
+ASM1	53
+ASM2	54

SSH: 50

SID:cdbrac1:DB_CID:51

SID:cdbrac2:DB_CID:52

ASM: 53

ASM: 54

Example Use Case 2, assuming this Oracle DB:

Instance	DB Credential ID
orcl	52
+ASM1	53

```
PSH:51
```

```
SID:orcl:DB_CID:52
```

```
ASM:53
```

Headers for TCPS

To force the PowerPack use TCPS, you must add a header with "TCPS:/path/to/instant_client". By adding this header, every instance listed in the headers will reference the SL1 tnsnames.ora file, so it is important to ensure that all required instances are included in the tnsnames.ora file.

Example Use Case 1, assuming this Oracle DB:

- SL1 Collector's Oracle Instant Client path: `/opt/oracle/instant_client_21_15`

Instance	DB Credential ID	Parent CDB
cdbrac1	51	N/A
PDB1	53	cdbrac1

```
SSH:50
```

```
SID:cdbrac1:DB_CID:51:CDB:true
```

```
PDB:PDB1:DB_CID:53:CDB:cdbrac1
```

```
TCPS:/opt/oracle/instant_client_21_15
```

Example Use Case 2, assuming this Oracle DB:

- SL1 Collector's Oracle Instant Client path: `/opt/oracle/instant_client_19_24`

Instance	DB Credential ID	Parent CDB
orcl	52	N/A
PDBPROD	54	orcl
+ASM1	53	N/A

```
PSH:51
```

```
SID:orcl:DB_CID:52:CDB:true
```

```
PDB:PDBPROD:DB_CID:54:CDB:orcl
```

```
ASM:53
```

```
TCPS:/opt/oracle/instant_client_19_24
```

HTTP Headers Special Keywords

This section contains special keywords you can use to modify or enhance the discovery and data collection process, giving you greater control over how your Oracle Database instances are managed.

TNS_ADMIN

The TNS_ADMIN parameter is used to specify a custom path for the tnsnames.ora file, which overrides the default location (\$ORACLE_HOME/network/admin). This is useful if your configuration files are stored in a non-standard directory. By including a standalone TNS_ADMIN header, you instruct the PowerPack to apply the specified custom path to all SIDs listed under the corresponding SOAP/XML credential.

IMPORTANT: Do not end the path specified in TNS_ADMIN with a backslash (/), as this can cause configuration errors.

NOTE: TNS_ADMIN and GRID_PATH are not supported for Windows devices.

Example Use Case 1:

```
SSH:50
```

```
SID:cdbrac1:DB_CID:51
```

```
SID:cdbrac2:DB_CID:52
```

```
TNS_ADMIN:/u01/app/oracle/network
```

Example Use Case 2:

```
SSH:51
```

```
SID:orcl:DB_CID:52
```

```
TNS_ADMIN:D:/home/oracle
```

TNS_ADMIN Within an SIDE Header

Including the TNS_ADMIN parameter within a SID header directs the PowerPack to use a custom path for the tnsnames.ora file specifically for that SID. This allows precise control over which SIDs use a non-default location for their network configuration files.

IMPORTANT: If the CDB header requires `CDB: true`, ensure this is always the last part of the header.

IMPORTANT: The `TNS_ADMIN` parameter is not supported for PDB headers.

Example Use Case 1:

- In this case, only "cdbrac1" uses the custom path `/u01/app/oracle/network`, while "cdbrac2" uses the default location.

```
SSH:50
```

```
SID:cdbrac1:DB_CID:51:TNS_ADMIN:/u01/app/oracle/network
```

```
SID:cdbrac2:DB_CID:52
```

Example Use Case 2:

- In this case, "orcl" and "silo_cdb" use custom paths, while "mycdb" uses the default location.

```
SSH:51
```

```
SID:orcl:DB_CID:52:TNS_ADMIN:/home/slmon
```

```
SID:mycdb:DB_CID:53
```

```
SID:silo_cdb:DB_CID:54:TNS_ADMIN:/home/oracle/network/admin
```

Example Use Case 3:

- In this case, "orclcdb" is configured with a custom `TNS_ADMIN` path and marked as a CDB (`CDB:true`). The associated PDBs, PDB1 and PDB2, are linked to "orclcdb" but do not utilize the `TNS_ADMIN` parameter.

```
SSH:99
```

```
SID:orclcdb:DB_CID:56:TNS_ADMIN:/slmon/oracle:CDB:true
```

```
PDB:PDB1:DB_CID:111:CDB:orclcdb
```

```
PDB:PDB2:DB_CID:222:CDB:orclcdb
```

GRID_PATH

The `GRID_PATH` header is used to specify the location of the `GRID_HOME` directory and locate Oracle listeners, particularly for RAC (Real Application Clusters) or ASM (Automatic Storage Management) environments. Additionally, if the PowerPack is unable to execute the `lsnrctl` status command using the default listener path, it

will attempt to use the `grid_path` specified in the HTTP header as an alternative.

NOTE: `TNS_ADMIN` and `GRID_PATH` are not supported for Windows devices.

Example Use Case 1:

- In this case, if the default listener path fails, the PowerPack will attempt to locate the listeners using the `GRID_PATH` provided.

```
SSH:50
```

```
SID:cdbrac1:DB_CID:51:TNS_ADMIN:/u01/app/oracle/network
```

```
SID:cdbrac2:DB_CID:52
```

```
GRID_PATH:/u01/app/grid
```

Example Use Case 2:

- If the initial attempt to use default listener path is unsuccessful, the PowerPack will fall back to use the specified `GRID_PATH`.

NOTE: Ensure that the `GRID_PATH` provided accurately reflects the location of the Grid Infrastructure on your system to facilitate correct listener management for RAC or ASM environments.

```
SSH:50
```

```
SID:orcl_rac_01:DB_CID:51
```

```
SID:orcl_rac_02:DB_CID:52
```

```
GRID_PATH:/u01/app/grid
```

`TYPE_OF_OS`

The `TYPE_OF_OS` header is used to address compatibility issues when the PowerPack encounters difficulties determining the operating system, or when an OS is part of a supported family but is not explicitly recognized. This header helps ensure that the PowerPack operates correctly by specifying the OS type manually.

The following values are supported:

- `linux`
- `solaris`

- aix
- windows
- ubuntu

Example Use Case 1:

- In this case, the TYPE_OF_OS is specified as linux, ensuring proper compatibility with the PowerPack.

```
SSH:50
```

```
SID:cdbrac1:DB_CID:51:TNS_ADMIN:/u01/app/oracle/network
```

```
SID:cdbrac2:DB_CID:52
```

```
TYPE_OF_OS:linux
```

Example Use Case 2:

- In this case, the TYPE_OF_OS is set to windows, which helps the PowerPack handle any compatibility issues with the Windows operating system.

```
PSH:51
```

```
SID:orcl:DB_CID:52
```

```
SID:mycdb:DB_CID:53
```

```
SID:silo_cdb:DB_CID:54
```

```
TYPE_OF_OS:windows
```

Example Use Case 3:

- In this case, the TYPE_OF_OS is set to windows, and is applied to the CDB and its associated PDBs to ensure compatibility.

```
PSH:99
```

```
TYPE_OF_OS:windows
```

```
SID:orclcdb:DB_CID:56:CDB:true
```

```
PDB:PDB1:DB_CID:111:CDB:orclcdb
```

```
PDB:PDB2:DB_CID:222:CDB:orclcdb
```

Example Use Case 4:

- In this case, TYPE_OF_OS is set to ubuntu, which is used to address any issues related to the Ubuntu OS.

```
TYPE_OF_OS:ubuntu
```

```
SSH:50
```

```
SID:cdbrac1:DB_CID:51
```

```
SID:cdbrac2:DB_CID:52
```

Deprecated HTTP Configuration Headers

The following headers have been deprecated and will no longer function:

- **HOST.** This header was previously used to specify a default host for all SIDs listed under a SOAP/XML credential. This was useful if your Oracle DB was hosted on a different machine than the OS server.
- **PORT.** This header was used to specify the port number for database instances in the SOAP/XML credential. This method is retained for compatibility but is now outdated. ScienceLogic recommends using DB_CID for new configurations.
- **SERVICE_NAME.** This header was used to specify the service name of the Oracle database. This header was part of an older configuration method and is now obsolete with the introduction of DB_CID. ScienceLogic does not recommend using SERVICE_NAME with DB_CID.

4. Click the **[Save As]** button.

Enabling PEM on a Linux Machine

Linux and Unix users can create an SSH/Key credential in order to monitor Oracle Database instances in SL1. The **Private Key (PEM Format)** field may be filled when [creating an SSH/Key credential](#). To enable PEM on a Linux machine, perform the following steps:

1. Create a PEM folder to place the identity keys by running the following command:

```
mkdir -p ~/ssh
```

NOTE: ScienceLogic suggests that you create a PEM folder inside the .ssh folder of the user that will use the PEM authentication.

2. Run the following command on your Linux machine to create the SSH key. This command will create public and private keys:

```
ssh-keygen -b 2048 -t rsa -f ~/.ssh/identity
```

NOTE: When prompted for a password, you can press **[Enter]** to leave it blank and move forward. You will be prompted for a password every time the system runs a Dynamic Application.

NOTE: The value "identity" in the command above will be the name of the file that is generated. This value can be replaced with any file name.

3. The private key generated from this command is the .pem file needed for the SSH/Key credential. Copy the contents of the file to input into the SL1 credential.
4. Add the generated public key to the `authorized_keys` file that is found in `~/.ssh/authorized_keys` manually or by using the following command:

```
cat ~/.ssh/identity.pub >> ~/.ssh/authorized_keys
```

5. Restart the SSH service by running the following command:

```
sudo service ssh restart
```

After completing the steps above, you can create an SSH/Key credential in SL1 by entering your Linux server username, Linux server password, and private key. If you would like to create an SSH/Key credential by entering only your Linux server username and private key, perform the following steps on your Linux machine:

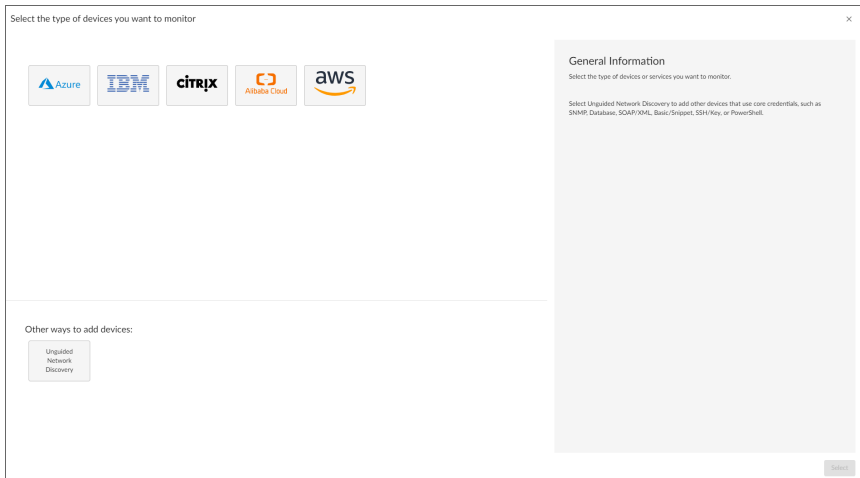
1. Find the `sshd_config` file.
2. Find the `PasswordAuthentication` command line, delete `yes`, and input `no`.
3. Restart the SSH service by running the following command:

```
sudo service ssh restart
```

Discovering Oracle Database Instances

To create and run a discovery session that will discover an Oracle instance, perform the following steps:

1. On the **Devices** page (🔍) or the **Discovery Sessions** page (Devices > Discovery Sessions), click the **[Add Devices]** button. The **Select** page appears:



2. Click the **[Unguided Network Discovery]** button. Additional information about the requirements for discovery appears in the **General Information** pane to the right.
3. Click **[Select]**. The **Add Devices** page appears:

Add Devices

Name
OracleInstance

Description (Optional)

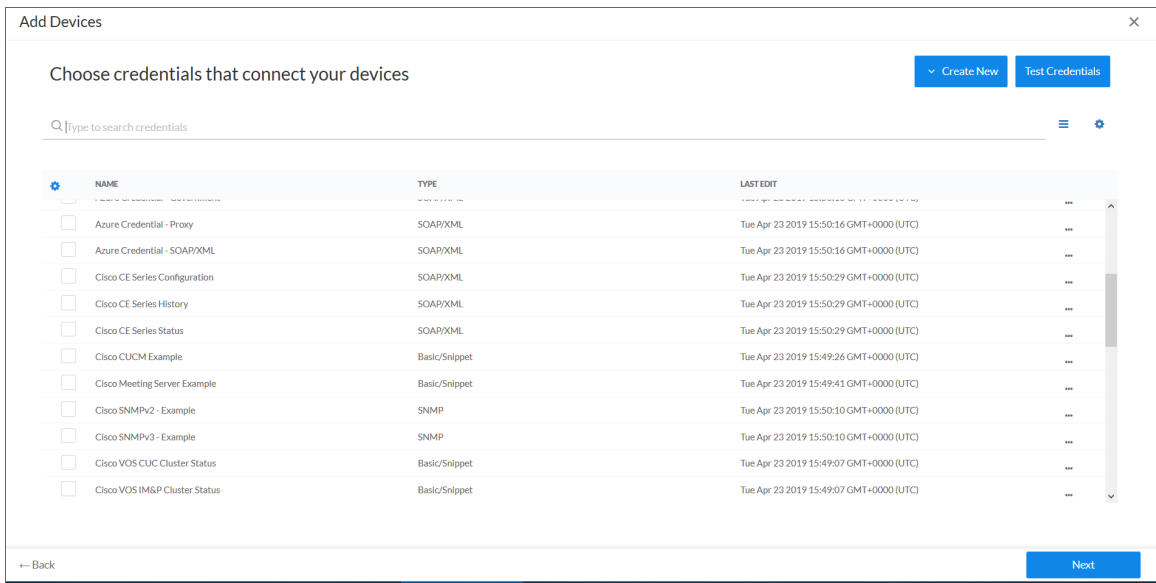
Select the organization to add discovered devices to
System

← Back Next

4. Complete the following fields:

- **Name.** Type a unique name for this discovery session. This name is displayed in the list of discovery sessions on the **[Discovery Sessions]** tab.
- **Description.** Optional. Type a short description of the discovery session. You can use the text in this description to search for the discovery session on the **[Discovery Sessions]** tab.
- **Select the organization to add discovered devices to.** Select the name of the organization to which you want to add the discovered devices.

5. Click **[Next]**. The **Credentials** page of the **Add Devices** wizard appears:




6. On the **Credentials** page, select the **SOAP/XML credential** you created.

7. Click **[Next]**. The **Discovery Session Details** page of the **Add Devices** wizard appears:

8. Complete the following fields:

- **List of IPs/Hostnames.** Type the IP address for the server that is hosting your Oracle Database.
- **Which collector will monitor these devices?** Select an existing collector to monitor the discovered devices. Required.
- **Run after save.** Select this option to run this discovery session as soon as you click [**Save and Close**].

In the **Advanced options** section, click the down arrow icon () to complete the following fields:

- **Discover Non-SNMP.** Enable this setting.
- **Model Devices.** Enable this setting.



9. Click [**Save and Close**] to save the discovery session. The **Discovery Sessions** page (Devices > Discovery Sessions) displays the new discovery session.

10. If you selected the **Run after save** option on this page, the discovery session runs, and the **Discovery Logs** page displays any relevant log messages. If the discovery session locates and adds any devices, the **Discovery Logs** page includes a link to the **Device Investigator** page for the discovered device.

Discovering Oracle Database Instances in the SL1 Classic User Interface

To model and monitor your Oracle Database instances, you must run a discovery session. To create and run a discovery session that will discover your Oracle Database instances, perform the following steps:

1. Go to the **Discovery Control Panel** page (System > Manage > Classic Discovery).

2. Click the **[Create]** button to create a new discovery session. The **Discovery Session Editor** window appears:
3. Enter values in the following fields:
 - **IP Address Discovery List**. Type the IP address for the server that is hosting your Oracle Database. One discovery session per server is supported.
 - **Other Credentials**. Select the SOAP/XML credential that you configured in the previous section.
 - **Discover Non-SNMP**. Select this checkbox.
 - **Model Devices**. Select this checkbox.
4. You can enter values in the other fields on this page, but are not required to and can simply accept the default values. For more information about the other fields on this page, see the **Discovery & Credentials** manual.
5. Click the **[Save]** button and then close the **Discovery Session Editor** window.
6. The discovery session you created will appear at the top of the **Discovery Control Panel** page. Click its lightning-bolt icon () to run the discovery session.
7. The **Discovery Session** window will be displayed.
8. When the server that is hosting the Oracle Database is discovered, click its device icon () to view the **Device Properties** page for that device.
9. After the server hosting the Oracle Database is discovered, the "Oracle: DB Instance Discovery" Dynamic Application will automatically be aligned. This Dynamic Application will discover the Oracle Database instances which will appear in the **Device Manager** page.

NOTE: If you are on a Windows system and are having issues with discovery, please see the **Monitoring Windows Systems with PowerShell** manual section.

Verifying Discovery and Dynamic Application Alignment

During discovery, SL1 will first identify the root device, followed by the associated Oracle Database instance, whether it's a CDB, PDB, non-CDB, or RAC node, depending on the configuration in the SOAP/XML credential header. All relevant Dynamic Applications will then be automatically aligned to the discovered components, ensuring accurate monitoring and configuration.

NOTE: If you discovered a non-CDB in a Windows instance, the "Oracle: DB Archived File System Stats" and "Oracle: DB Non-Archived File System Stats" Dynamic Applications will not be aligned.

IMPORTANT: In version 106 of this PowerPack, the "Oracle: DB ASM Diskgroup Config (moved)" and "Oracle DB ASM Instance Config (moved)" Dynamic Applications are disabled by default and were duplicated since they are now aligned to a different device. The old version of these Dynamic Applications are still in the PowerPack with the names "Oracle: DB ASM Diskgroup Config (moved)" and "Oracle: DB ASM Instance Config (moved)", but now with some collection objects removed. These Dynamic Applications will be removed completely in a future release of this PowerPack.

To verify alignment of the Oracle Database Dynamic Applications:

1. After discovery has completed, click the device icon for the Oracle device (). From the **Device Properties** page for the Oracle device, click the **[Collections]** tab. The **Dynamic Application Collections** page appears.

NOTE: It can take two to three polling cycles after the discovery session has completed for Dynamic Applications to appear in the **Dynamic Application Collections** page.

2. All applicable Dynamic Applications are automatically aligned to the root device and component devices during discovery:

You should see the following Dynamic Applications aligned to the root device:

- Oracle: DB Instance Discovery
- Oracle: DB Server Config
- Oracle: DB ASM Discovery
- Oracle: DB ASM Diskgroup Config
- Oracle: DB ASM Diskgroup Stats
- Oracle: DB ASM Disks Config
- Oracle: DB ASM Disks Stats

Close	Properties	Thresholds	Collections	Monitors	Schedule		
Logs	Toolbox	Interfaces	Relationships	Tickets	Redirects	Notes	Attributes
Device Name	10.2.5.104			Managed Type	Physical Device		
IP Address / ID	10.2.5.104 79			Category	Servers		
Class	Linux			Sub-Class	Generic		
Organization	Oracle_test_TCPS ASM			Uptime	0 days, 00:00:00		
Collection Mode	Active			Collection Time	2024-09-23 18:47:00		
Description				Group / Collector	CUG KNT-DIST-CU-78		
Device Hostname							

Dynamic Application™ Collections							Expand	Actions	Reset	Guide
	Dynamic Application	ID	Poll Frequency	Type	Credential	Collector				
+	Oracle: DB ASM Diskgroup Stats	829	5 mins	Snippet Performance	Oracle: SOAP 10.2.5.104 - TCPS	KNT-DIST-CU-78				
+	Oracle: DB ASM Disks Stats	830	5 mins	Snippet Performance	Oracle: SOAP 10.2.5.104 - TCPS	KNT-DIST-CU-78				
+	Oracle: DB ASM Discovery	828	15 mins	Snippet Configuration	Oracle: SOAP 10.2.5.104 - TCPS	KNT-DIST-CU-78				
+	Oracle: DB ASM Diskgroup Config	826	15 mins	Snippet Configuration	Oracle: SOAP 10.2.5.104 - TCPS	KNT-DIST-CU-78				
+	Oracle: DB ASM Disks Config	831	15 mins	Snippet Configuration	Oracle: SOAP 10.2.5.104 - TCPS	KNT-DIST-CU-78				
+	Oracle: DB Instance Discovery	800	10 mins	Snippet Configuration	Oracle: SOAP 10.2.5.104 - TCPS	KNT-DIST-CU-78				
+	Oracle: DB Server Config	799	15 mins	Snippet Configuration	Oracle: SOAP 10.2.5.104 - TCPS	KNT-DIST-CU-78				

[Select Action]

Copyright © 2003 - 2024 ScienceLogic, Inc. All rights reserved.

Using an Oracle PP on an existing Linux device will not interfere with the historical data on the device. Instead, the Oracle will align at the root with the other Linux Dynamic Applications.

Close	Properties	Thresholds	Collections	Monitors	Schedule			
Logs	Toolbox	Interfaces	Relationships	Tickets	Redirects	Notes	Attributes	
Device Name	bkserv1dbo101.burgerking.com.br			Managed Type	Physical Device			
IP Address / D	172.22.4.23 2561			Category	Servers			
Class	Linux			Sub-Class	Oracle Linux Server 7			
Organization	BKB-BURGERKING-DAL			Uptime	4 days, 04:55:43			
Collection Mode	Active			Collection Time	2021-03-31 14:40:00			
Description	--			Group / Collector	RMI-MCMS-DAL-COLLECTOR-1 sldalsldc04			
Device Hostname								

Dynamic Application™ Collections				Expand	Actions	Reset	Guide
+ Linux: Zombie Process	1555	5 mins	Snippet Performance	BKB-BKB-DAL-LINUX-SSH	--		
+ SLCOE: Linux Inodes Monitoring	1837	5 mins	Snippet Performance	BKB-BKB-DAL-LINUX-SSH	sldalsldc04		
+ Linux: Configuration Discovery	1557	15 mins	Snippet Configuration	BKB-BKB-DAL-LINUX-SSH	sldalsldc04		
+ Linux: CPU Configuration	1554	1440 mins	Snippet Configuration	BKB-BKB-DAL-LINUX-SSH	sldalsldc04		
+ Linux: Hardware Configuration	1551	1440 mins	Snippet Configuration	BKB-BKB-DAL-LINUX-SSH	sldalsldc04		
+ Linux: ICDA Cache	2135	15 mins	Snippet Configuration	BKB-BKB-DAL-LINUX-SSH	sldalsldc04		
+ Linux: ICDA Interface Cache	2145	5 mins	Snippet Configuration	BKB-BKB-DAL-LINUX-SSH	sldalsldc04		
+ Linux: Network Configuration	1547	15 mins	Snippet Configuration	BKB-BKB-DAL-LINUX-SSH	--		
+ Linux: Route Table Configuration	1556	15 mins	Snippet Configuration	BKB-BKB-DAL-LINUX-SSH	--		
+ Linux: System Configuration	1546	1440 mins	Snippet Configuration	BKB-BKB-DAL-LINUX-SSH	sldalsldc04		
+ Linux: TCP Services Configuration	1544	15 mins	Snippet Configuration	BKB-BKB-DAL-LINUX-SSH	--		
+ Oracle: DB Instance Discovery	2074	15 mins	Snippet Configuration	BKB-ORACLE-231	sldalsldc04		
+ Oracle: DB Server Config	2073	15 mins	Snippet Configuration	BKB-ORACLE-231	sldalsldc04		
+ SLCOE: Log File Monitoring (SSH)	1800	15 mins	Snippet Configuration	BKB-BKB-DAL-LINUX-SSH	sldalsldc04		
+ Linux: IC Filesystem Inventory	2137	0 mins	Internal Collection Inventory	BKB-BKB-DAL-LINUX-SSH	--		
+ Linux: IC Interface Inventory	2140	0 mins	Internal Collection Inventory	BKB-BKB-DAL-LINUX-SSH	--		
+ Linux: IC Process Inventory	2143	0 mins	Internal Collection Inventory	BKB-BKB-DAL-LINUX-SSH	--		
+ Linux: Process Inventory (ICDA)	1799	0 mins	Internal Collection Inventory	BKB-BKB-DAL-LINUX-SSH	--		
+ Linux: IC Availability	2136	0 mins	Internal Collection Performance	BKB-BKB-DAL-LINUX-SSH	--		
+ Linux: IC Detail	2139	0 mins	Internal Collection Performance	BKB-BKB-DAL-LINUX-SSH	--		
+ Linux: IC Filesystem Performance	2138	0 mins	Internal Collection Performance	BKB-BKB-DAL-LINUX-SSH	--		
+ Linux: IC Interface Performance	2141	0 mins	Internal Collection Performance	BKB-BKB-DAL-LINUX-SSH	--		
+ Linux: IC Port Performance	2144	0 mins	Internal Collection Performance	BKB-BKB-DAL-LINUX-SSH	--		
+ Linux: IC Process Performance	2142	0 mins	Internal Collection Performance	BKB-BKB-DAL-LINUX-SSH	--		
+ Linux: Process Perf (ICDA)	1801	0 mins	Internal Collection Performance	BKB-BKB-DAL-LINUX-SSH	--		

[Select Action] [Go]

Save

You should see the following Dynamic Applications aligned to **ASM**:

- Oracle: DB ASM Instance Config

You should see the following Dynamic Applications aligned to **CDB**:

- Oracle: DB Archived File System Stats
- Oracle: DB Non-Archived File System Stats
- Oracle: DB Instance Config
- Oracle: DB PDB Discovery
- Oracle: DB RMAN Backup Status Config

NOTE: The Oracle RAC Dynamic Applications will only be aligned on RAC systems.

If your CDB is part of a RAC (Real Application Cluster), the following additional Dynamic Applications will also be aligned:

- Oracle: DB RAC Disk Group Space Stats
- Oracle: DB RAC Flash Recovery Stats
- Oracle: DB RAC Global Cache Stats

You should see the following Dynamic Applications aligned to **PDB Oracle Database instances**:

- Oracle: DB Archived File System Stats
- Oracle: DB Blocking Session
- Oracle: DB Non-Archived File System Stats
- Oracle: DB Instance Config
- Oracle: DB Chained Row Stats
- Oracle: DB Data Guard Gap Stats
- Oracle: DB Database Size Stats
- Oracle: DB Instance Invalid Object Stats
- Oracle: DB Integrity Metrics Stats
- Oracle: DB Logswitch Rate Stats
- Oracle: DB Long Running Session
- Oracle: DB Open Cursors per Session Stats
- Oracle: DB Performance Stats
- Oracle: DB Resource Stats
- Oracle: DB Session Stats
- Oracle: DB Tablespace Stats
- Oracle: DB Tablespace Temp Stats
- Oracle: DB Components Status Config
- Oracle: DB Log Alerts Config
- Oracle: DB Tablespaces and Datafiles Status Config

You should see the following Dynamic Applications aligned to **non-CDB Oracle Database instances**:

- Oracle: DB Archived File System Stats
- Oracle: DB Blocking Session
- Oracle: DB Non-Archived File System Stats
- Oracle: DB Instance Config
- Oracle: DB Chained Row Stats
- Oracle: DB Data Guard Gap Stats
- Oracle: DB Database Size Stats
- Oracle: DB Instance Invalid Object Stats
- Oracle: DB Integrity Metrics Stats
- Oracle: DB Logswitch Rate Stats
- Oracle: DB Long Running Session
- Oracle: DB Open Cursors per Session Stats
- Oracle: DB Performance Stats
- Oracle: DB Resource Stats

- Oracle: DB Session Stats
- Oracle: DB Tablespace Stats
- Oracle: DB Tablespace Temp Stats
- Oracle: DB Components Status Config
- Oracle: DB Log Alerts Config
- Oracle: DB Tablespaces and Datafiles Status Config

If your CDB is part of a RAC (Real Application Cluster), the following additional Dynamic Applications will also be aligned:

NOTE: The Oracle RAC Dynamic Applications will only be aligned on RAC systems.

- Oracle: DB RAC Disk Group Space Stats
- Oracle: DB RAC Flash Recovery Stats
- Oracle: DB RAC Global Cache Stats

Snippet and Snippet Argument Configuration for New Oracle Dynamic Applications

You can configure snippets and snippet arguments in *Oracle: Database Dynamic Applications* to run SQL queries.

Snippet arguments can be used for simple queries consisting of only SELECT and WHERE. Complex queries must be defined in the snippet.

Running SQL Queries from Snippet Arguments

In the **[Collections]** tab of a Dynamic Application, you can select a collection object in the **Collection Object Registry** and edit the argument in the **Snippet Arguments** field.

For example, using the collection objects from the "Oracle: DB Performance Stats" Dynamic Application:

Snippet argument without a filter:

```
oracle://&siilo_args=column=<column_name>&table=<table_name>
```

Snippet argument with a filter:

```
oracle://&siilo_args=column=<column_name>&table=<table_name>&filter=<where clause>
```

Example:

```
oracle://&silos_args=column=name&table=my_table&filter=name LIKE '%abc%'
AND id != 0
```

NOTE: Spaces can be used in any of the arguments if necessary. Column should name a single column only.

Queries are consolidated into a single SQL query for each table. If you want to separate the queries, use the following format:

```
oracle://&silos_args=column=name&table=my_table a_names&filter=name LIKE
'a%'
```

```
oracle://&silos_args=column=name&table=my_table b_names&filter=name LIKE
'b%'
```

Running Raw SQL Queries from the Snippet

You can run raw SQL queries in the snippets in Dynamic Applications by going to the **[Snippets]** tab and selecting the snippet from the **Snippet Registry**.

In the "Oracle: DB Chained Rows Stats" Dynamic Application, you can edit the snippet to include a raw SQL query in the following way:

```
from silo.oracle_db.OracleDB_sql_collector import OracleDBSQLCollector

from silo.apps.errors import ErrorManager

import logging

query = """SELECT X, Y FROM table"""

with ErrorManager(self):

    collector = OracleDBSQLCollector(self)

    results = collector.collect_raw(query)

    if results:
```

```
query_order = ["X", "Y"] #  
  
collector.handle_raw_results(self.oids, query_order, results)
```

X and Y are collection object OIDs, in the order corresponding to the query column names.

Indexed Raw SQL Query

You can run indexed SQL queries in the snippets in Dynamic Applications by going to the **[Snippets]** tab and selecting the snippet from the **Snippet Registry**.

For example, in the "Oracle: DB Tablespaces and Datafiles Status Config" Dynamic Application, you can edit the snippet to include an indexed SQL query in the following way:

```
from silo.oracle_db.OracleDB_sql_collector import OracleDBSQLCollector  
  
from silo.apps.errors import ErrorManager  
  
import logging  
  
  
query = """SELECT x, x, y FROM table""" #<-The first column is the  
index; it is ok to repeat a column  
  
  
with ErrorManager(self):  
  
    collector = OracleDBSQLCollector(self)  
  
    results = collector.collect_raw(query)  
  
    if results:  
  
        query_order = ["X", "Y"]  
  
        collector.handle_raw_results(self.oids, query_order, results,  
indexed=True)
```

X and Y are collection object OIDs, in the order corresponding to the query column names, not including the index column.

Running Combined Raw SQL Queries and Snippet Arguments

If snippet arguments and raw queries are combined in a single Dynamic Application, the snippet argument code must be executed first. The following example was added to the snippet in the "Oracle: DB Session Stats" Dynamic Application:

```
with ErrorManager(self):  
  
    collector = OracleDBSQLCollector(self)  
  
    # Collect snippet arg results  
  
    collector.collect()  
  
    collector.handle_results()  
  
    # Collect raw query results  
  
    results = collector.collect_raw(query)  
  
    if results:  
  
        collector.handle_raw_results(self.oids, query_order, results)
```

Running SSH Commands from a Snippet

You can run SSH commands the snippets in Dynamic Applications by going to the **[Snippets]** tab and selecting the snippet from the **Snippet Registry**.

For example, in the "Oracle: DB Non-Archived File System Stats" Dynamic Application, you can edit the snippet to include an SSH command in the following way:

```
ssh.append("your_ssh_command")  
  
ssh_results = ssh_collector.run_commands(ssh)
```


Viewing Oracle Component Devices

In addition to the **Devices** page, you can view the Oracle Database and all associated component devices in the following places in the user interface:

- The **Device Investigator** Map page (click **Map** in the **Device Investigator** page) displays a map of a particular device and all of the devices with which it has parent-child relationships. Double-clicking any of the listed devices reloads the page to make the selected device the primary device.
- The **Device Components** page (Devices > Device Components) displays a list of all root devices and component devices discovered by SL1. The **Device Components** page displays all root devices and component devices in an indented view, so you can easily view the hierarchy and relationships between child devices, parent devices, and root devices. To view the component devices associated with an Oracle Database, find the Oracle Database and click its plus icon (+).
- The **Component Map** page (Classic Maps > Device Maps > Components) allows you to view devices by root node and view the relationships between root nodes, parent components, and child components in a map. This makes it easy to visualize and manage root nodes and their components. SL1 automatically updates the **Component Map** as new component devices are discovered. The platform also updates each map with the latest status and event information. To view the map for an Oracle Database, go to the **Component Map** page and select the map from the list in the left NavBar. To learn more about the **Component Map** page, see the **Views** manual.

Viewing Oracle Component Devices in the Classic SL1 User Interface

In addition to the **Device Manager** page (Registry > Devices > Device Manager), you can view the Oracle Database instances and all associated component devices in the following places in the user interface:

- The **Device View** modal page (click the bar-graph icon  for a device, then click the **Topology** tab) displays a map of a particular device and all of the devices with which it has parent-child relationships. Double-clicking any of the devices listed reloads the page to make the selected device the primary device.
- The **Device Components** page (Registry > Devices > Device Components) displays a list of all root devices and component devices discovered by SL1 in an indented view, so you can easily view the hierarchy and relationships between child devices, parent devices, and root devices. To view the component devices associated with an Oracle Database instance, find the Oracle device and click its plus icon (+).
- The **Component Map** page (Classic Maps > Device Maps > Components) allows you to view devices by root node and view the relationships between root nodes, parent components, and child components in a map. This makes it easy to visualize and manage root nodes and their components. SL1 automatically updates the **Component Map** as new component devices are discovered. The platform also updates each map with the latest status and event information. To view the map for an Oracle Database instance, go to the **Component Map** page and select the map from the list in the left NavBar. To learn more about the **Component Map** page, see the **Views** manual.

Oracle: Database Dashboards

Overview

The following section describes the device dashboard that is included in the *Oracle: Database PowerPack*:

This chapter covers the following topics:

<i>Device Dashboard</i>	67
<i>Oracle: Database Minimum Permissions Needed</i>	69

Device Dashboard

The *Oracle: Database PowerPack* includes a device dashboard that provides summary information for an Oracle Database instance.

NOTE: If you want to align the device dashboard as the default device dashboard for the Oracle Database instance, you must do so manually.

Oracle Database: Instance

The Oracle Database: Instance device dashboard displays the following information:

- Six gauges that display the following metrics:
 - Chained Rows
 - Data Size
 - Tablespace Max Files

- Login Count
- Logswitch Counter
- Session Locks
- A bar graph that displays the Top 5 Oracle: DB Tablespace Stats Space Percent Used
- Four line graphs that display the following information:
 - Oracle: DB Tablespace Stats | Space Percent Used | UNDOTBS (%)
 - Performance Stats
 - Utilization Percent
 - Active Users

Appendix

3

Overview

This appendix describes the minimum user permissions for Oracle: Database and why they are needed.

If your user is "oracle", the default Oracle OS user, you should already have all the required permissions.

Oracle: Database Minimum Permissions Needed

At a minimum, SL1 needs the following:

- To be able to retrieve `lsnrctl status` output: This is used to check instance status, to determine if the instance is up or down. The output is also used to model child devices.
- To be able to retrieve `tnsnames.ora` info: SL1 matches any provided credentials with the contents of `tnsnames.ora` to verify whether the credentials are correct. The file information is also used to monitor PDBs.

To get that information, the Oracle: Database user permissions listed below are needed.

Folder path through `lsnrctl`, `tnsnames.ora`

This permission is needed to access `lsnrctl` and `tnsnames.ora`. Every folder to reach those files must have "read" and "execute" permissions. For example:

If `ORACLE_HOME` is `/u01/app/oracle/product/21.0.0/dbhome_1`

- give "read" & "execute" to `/u01`
- give "read" & "execute" to `app`
- give "read" & "execute" to `oracle`
- give "read" & "execute" to `product`
- give "read" & "execute" to `21.0.0`
- give "read" & "execute" to `dbhome_1`

NOTE: If you are not using “oracle” in your SSH credential, and want to give permissions to a user outside the “orainstall” group, the commands should use `o=` where `o=` stands for “other” and `rx` stands for “read & execute”.

```
chmod o=rx /u01
```

```
chmod o=rx /u01/app
```

```
chmod o=rx /u01/app/oracle
```

```
chmod o=rx /u01/app/oracle/product
```

```
chmod o=rx /u01/app/oracle/product/21.0.0
```

```
chmod o=rx $ORACLE_HOME
```

To give permissions to bin and lib folders:

```
chmod o=rx $ORACLE_HOME/bin
```

```
chmod o=rx $ORACLE_HOME/lib
```

Execute permission to run `lsnrctl`

This permission is needed to run the `lsnrctl` command to check instance status.

```
chmod o=x $ORACLE_HOME/bin/lsnrctl
```

Read permission to read `libclntsh.so.21.1`, `libclntshcore.so.21.1`, `libnnz21.so`

This permission is needed because `lsnrctl` depends on them.

```
chmod o=r $ORACLE_HOME/lib/libclntsh.so.21.1
```

```
chmod o=r $ORACLE_HOME/lib/libclntshcore.so.21.1
```

```
chmod o=r $ORACLE_HOME/lib/libnnz21.so
```

Folder path to read `tnsnames.ora` and `mesg` folder

This permission is needed because SLI reads `tnsnames.ora` to verify if provided credentials match with `tnsnames.ora`, and because `lsnrctl` depends on `mesg` files.

```
chmod o=rx $ORACLE_HOME/network
```

```
chmod o=rx $ORACLE_HOME/network/admin
```

```
chmod o=rx $ORACLE_HOME/network/mesg
```

```
chmod o=r $ORACLE_HOME/network/admin/tnsnames.ora
```

```
chmod -R o=r $ORACLE_HOME/network/mesg/
```

Read permission for oratab

This permission is needed because SLI gets `ORACLE_HOME` from `oratab`.

```
chmod o=r /etc/oratab
```

`ORACLE_HOME` is needed to properly run `lsnrctl` and read `tnsnames.ora`.

© 2003 - 2024, ScienceLogic, Inc.

All rights reserved.

LIMITATION OF LIABILITY AND GENERAL DISCLAIMER

ALL INFORMATION AVAILABLE IN THIS GUIDE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. SCIENCELOGIC™ AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT.

Although ScienceLogic™ has attempted to provide accurate information on this Site, information on this Site may contain inadvertent technical inaccuracies or typographical errors, and ScienceLogic™ assumes no responsibility for the accuracy of the information. Information may be changed or updated without notice. ScienceLogic™ may also make improvements and / or changes in the products or services described in this Site at any time without notice.

Copyrights and Trademarks

ScienceLogic, the ScienceLogic logo, and EM7 are trademarks of ScienceLogic, Inc. in the United States, other countries, or both.

Below is a list of trademarks and service marks that should be credited to ScienceLogic, Inc. The ® and ™ symbols reflect the trademark registration status in the U.S. Patent and Trademark Office and may not be appropriate for materials to be distributed outside the United States.

- ScienceLogic™
- EM7™ and em7™
- Simplify IT™
- Dynamic Application™
- Relational Infrastructure Management™

The absence of a product or service name, slogan or logo from this list does not constitute a waiver of ScienceLogic's trademark or other intellectual property rights concerning that name, slogan, or logo.

Please note that laws concerning use of trademarks or product names vary by country. Always consult a local attorney for additional guidance.

Other

If any provision of this agreement shall be unlawful, void, or for any reason unenforceable, then that provision shall be deemed severable from this agreement and shall not affect the validity and enforceability of any remaining provisions. This is the entire agreement between the parties relating to the matters contained herein.

In the U.S. and other jurisdictions, trademark owners have a duty to police the use of their marks. Therefore, if you become aware of any improper use of ScienceLogic Trademarks, including infringement or counterfeiting by third parties, report them to Science Logic's legal department immediately. Report as much detail as possible about the misuse, including the name of the party, contact information, and copies or photographs of the potential misuse to: legal@sciencelogic.com. For more information, see <https://sciencelogic.com/company/legal>.

ScienceLogic

800-SCI-LOGIC (1-800-724-5644)

International: +1-703-354-1010