Monitoring Oracle

Oracle: Database PowerPack version 105

Table of Contents

Chapter 1

Introduction

Overview

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

This chapter covers the following topics:

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.

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.

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

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 device trees and all Oracle device classes before installing.

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 ***System Administration*** manual.)

IMPORTANT: The minimum required MySQL version is 5.6.0.

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:

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, 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_path/bin): `--x`
        - Libraries (oracle_path/lib): ` r-x`
        - File "listeners.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, $ORACLE_HOME/lsnrctl status or tnsping.
    - Permissions to read files. For example, listeners.ora/tnsnames.ora ($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_ext
- 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$asm_client (required for ASM/RAC Dynamic Applications)
  - v$asm _disk (required for ASM/RAC Dynamic Applications)
  - v$asm_diskgroup (required for ASM/RAC Dynamic Applications)
  - v$recovery_file_dest(required for ASM/RAC Dynamic Applications)
- All Oracle database users must have the following privileges:
  - sys_privileges: CREATE SESSION
  - role_privileges: SELECT_CATALOG_ROLE
  - tan_privileges: SELECT ON SYS.V_$DIAG_ALERT_EXT, SELECT ON SYS.TS$
  - Permission to alter sessions.
  - To monitor pluggable databases (PDBs), permissions to view DBA_PDBS and V_$PDBS

Examples for 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 the "Oracle: DB Tablespace Temp Stats" Dynamic Application: GRANT SELECT on SYS.TS$ to C##*[USERNAME]* CONTAINER=ALL;
  - Grant access for the "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;
- If you want to monitor a non-container database (non-CDB), you can create a user and grant access to the user using the following permissions:
  - 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 non-CDB: GRANT SELECT_CATALOG_ROLE to *[USERNAME]*;

    - Grant access for the "Oracle: DB Tablespace Temp Stats" Dynamic Application: GRANT SELECT on SYS.TS$ to *[USERNAME]*;

    - Grant access for the "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 needed.

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

NOTE: You can create a single user to access a CDB and all PDBs but you will need to create a minimum of 2 database credentials, one credential for the CDB and one credential for all PDBs.

Configuring Oracle Credentials

To monitor Oracle Database instances using SL1, you must create at least two 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*** and Windows users must use a ***PowerShell credential***
- If you are using version 104 and above of the PowerPack, you must also use a ***Database credential*** for each CDB, PDB, or Non-CDB
- 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 implement 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 minute timeout session is established. This allows the user time to stop the collection if it prevents SL1 to create SIGTERMs.

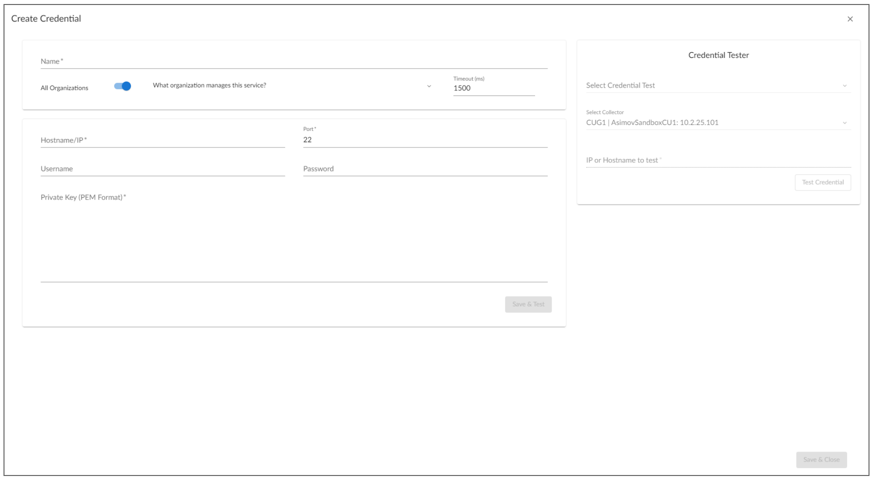
Creating an SSH/Key Credential (Linux Users)

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

NOTE: If you are on an SL1 system prior to version 11.1.0, you will not be able to duplicate the sample credential. It is recommended that you create your new credentials using ***the SL1 classic user interface*** so you do not overwrite the sample credential(s).

To create an SSH/Key credential:

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



Create Credential

Timeout (ms): 1500

Port*: 22

Select Collector: CUG1 | AsimovSandboxCU1: 10.2.25.101

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 (Linux Users) in the Classic SL1 User Interface

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

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 SSH Server Example" 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 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 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.

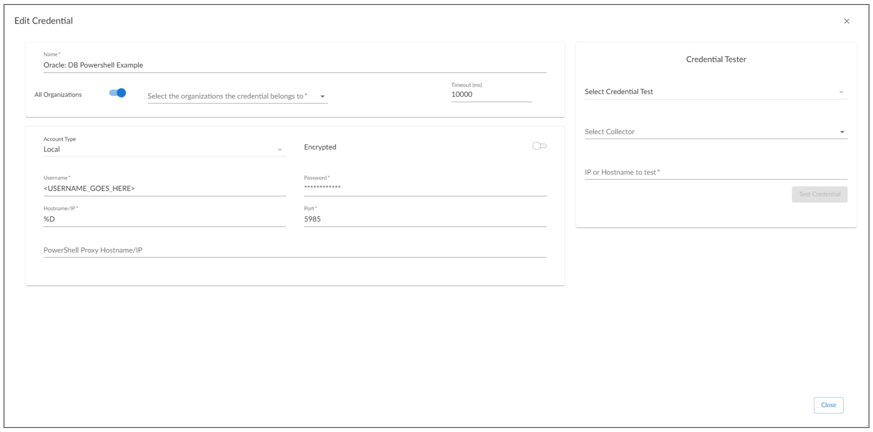
Creating a PowerShell Credential (Windows Users)

Windows users must create a PowerShell credential.

> **NOTE**: If you are on an SL1 system prior to version 11.1.0, you will not be able to duplicate the sample credential. It is recommended that you create your new credentials using ***the SL1 classic user interface*** so you do not overwrite the sample credential(s).

To create a PowerShell credential:

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



Edit Credential

Name *: Oracle: DB Powershell Example

Timeout (ms): 10000

Account Type: Local

Username *: <USERNAME_GOES_HERE>

Password *: ************

Hostname/IP *: %D

Port *: 5985

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 SL1 will stop trying to communicate with 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.

- ***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 Powershell Example" 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.

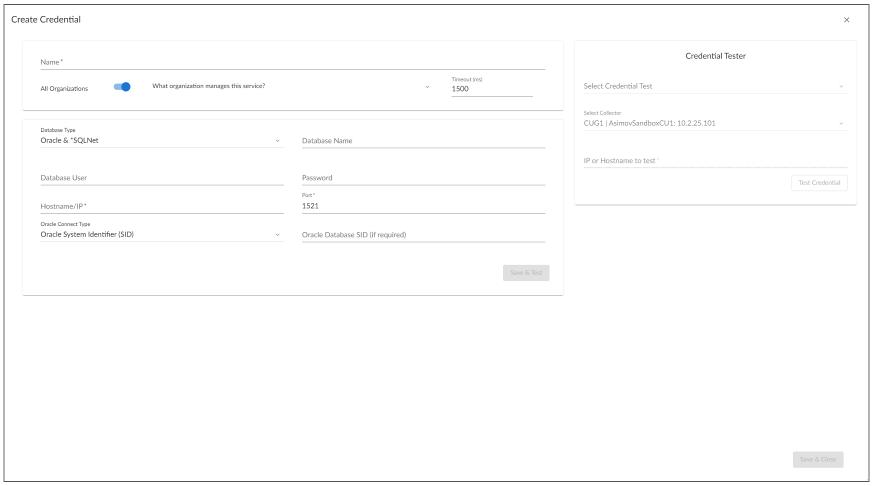
Creating a Database Credential

Linux or Unix users that want to monitor multiple CDB or PDB instances must create a database credential.

> **NOTE:** CDBs and PDBs each need a separate database credential but if you have PDBs that share the same username and password, they can use the same database credential and the SID name field can be left blank.

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:



Timeout (ms)
1500

Database Type
Oracle & *SQLNet

Port *
1521

Oracle Connect Type
Oracle System Identifier (SID)

Select Collector
CUG1 | AsimovSandboxCU1: 10.2.25.101

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.
   - ***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.
   - ***Password.*** Type the password for the Oracle Database account.

  - To discover CDBs, type the instance password.
  - To discover PDBs, type the PDB 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.

<u>Oracle Settings</u>

- ***Oracle Connect Type***. Select the method SL1 should use to connect to the Oracle database. 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.
  - *Oracle Real Application Clusters (SERVICE)*. Select this option if you want to discover a PDB instance.

> NOTE: This field is used to populate the 'SERVICE_NAME' field identified in the SOAP/XML credential. This method must be selected even if the server is not a RAC system.

- ***Oracle Database SID (if required)***. Type the value for the Oracle Connect Type (either Oracle SID, Oracle RAC, or Oracle Server) selected in the ***Oracle Connect Type*** field.

  - To discover CDBs, type the instance SID.
  - To discover PDBs, type the PDB instance SID.
  - If you want multiple instances to share the same credential, leave the field blank.

> 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. Oracle Database is case-sensitive, so your PDB name must match the case shown either in your tnsnames.ora or in the ***PDB_NAME*** column of the **DBA_PDBS** table.

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

Linux or Unix users that want to monitor multiple CDB or PDB instances must create a database credential.

NOTE: CDBs and PDBs each need a separate database credential but if you have PDBs that share the same username and password, they can use the same database credential and the SID name field can be left blank.

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:

   <u>Basic Settings</u>

   - ***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.
   - ***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.
   - ***Password***. Type the password for the Oracle Database account.
     - To discover CDBs, type the instance password.
     - To discover PDBs, type the PDB 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.

Oracle Settings

- ***Oracle Connect Type***. Select the method SL1 should use to connect to the Oracle database. 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.
    - *Oracle Real Application Clusters (SERVICE)*. Select this option if you want to discover a PDB instance.

> **NOTE:** This field is used to populate the 'SERVICE_NAME' field identified in the SOAP/XML credential. This method must be selected even if the server is not a RAC system.

- ***SID (if required)***. Enter the value for the Oracle Connect Type (either Oracle SID, Oracle RAC, or Oracle Server) selected in the ***Oracle Connect Type*** field.
    - To discover CDBs, type the instance SID.
    - To discover PDBs, type the PDB instance SID.
    - If you want multiple instances to share the same credential, leave the field blank.

> **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]**.

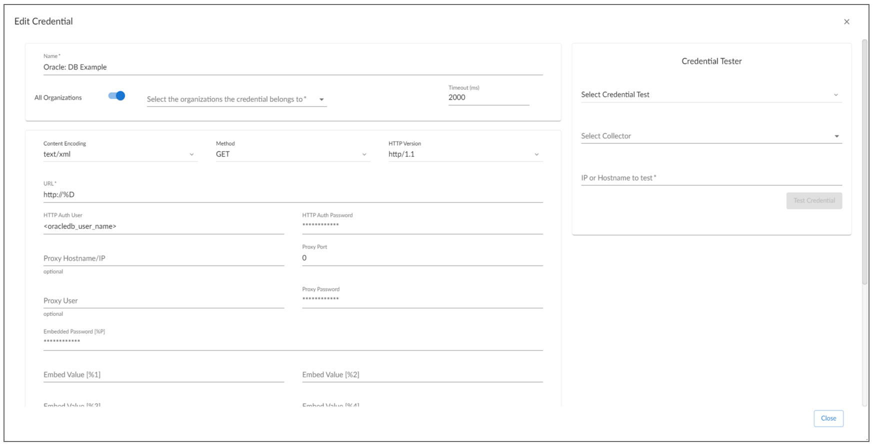
> **NOTE:** The credential ID will appear at the top of the window after it has been saved. Take note of the ID as you will need it when creating the SOAP/XML credential.

Creating a SOAP/XML Credential

> **NOTE**: If you are on an SL1 system prior to version 11.1.0, you will not be able to duplicate the sample credential. It is recommended that you create your new credentials using ***the SL1 classic user interface*** so you do not overwrite the sample credential(s).

To create a SOAP/XML credential:

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



Edit Credential

Name *: Oracle: DB Example

All Organizations

Select the organizations the credential belongs to *

Timeout (ms): 2000

Content Encoding: text/xml

Method: GET

HTTP Version: http/1.1

URL *: http://%D

HTTP Auth User: <oracledb_user_name>

HTTP Auth Password: ************

Proxy Hostname/IP (optional)

Proxy Port: 0

Proxy User (optional)

Proxy Password: ************

Embedded Password [%P]: ************

Embed Value [%1]

Embed Value [%2]

Embed Value [%3]

Embed Value [%4]

Credential Tester

Select Credential Test

Select Collector

IP or Hostname to test *

3. Supply values in the following fields:

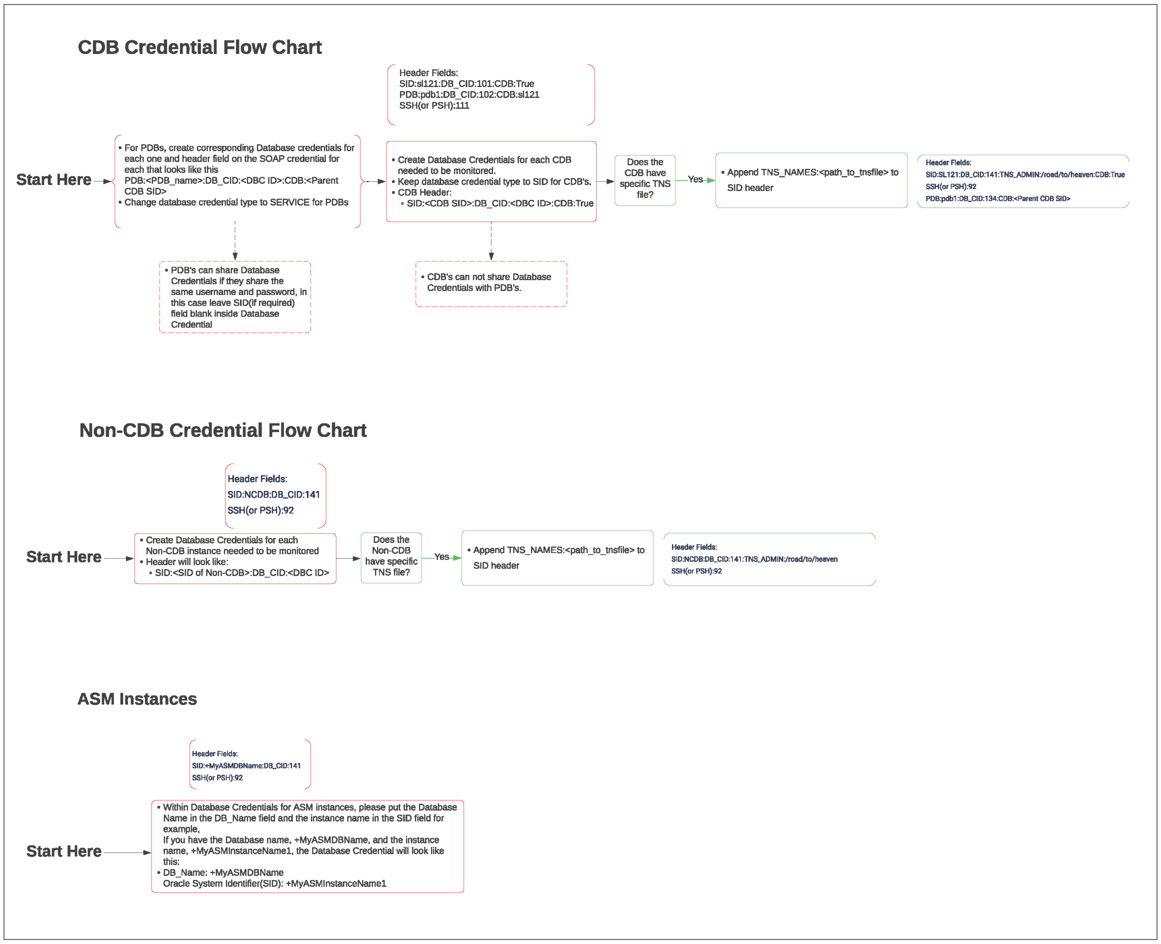
    - ***Name***. Type a new name for the credential.
    - ***All Organizations***. Toggle on (blue) to align the credential to all organizations, or toggle off (gray) and then select one or more specific organizations from the ***What organization manages this service?*** drop-down field to align the credential with those specific organizations.
    - ***URL***. Leave the default value of "https://%D".
    - ***HTTP Auth User***. Type the username for the Oracle Database account. If you created a database credential, this field is not required.
    - ***HTTP Auth Password***. Type the password for the Oracle Database account. If you created a database credential, this field is not required.

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

- ***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:

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


CDB Credential Flow Chart
Start Here
Does the CDB have specific TNS file?
Yes
Non-CDB Credential Flow Chart
Start Here
Does the Non-CDB have specific TNS file?
Yes
ASM Instances
Start Here

Header Fields:
SID:sl121:DB_CID:101:CDB:True
PDB:pdb1:DB_CID:102:CDB:sl121
SSH(or PSH):111

- For PDBs, create corresponding Database credentials for each one and header field on the SOAP credential for each that looks like this PDB:<PDB_name>:DB_CID:<DBC ID>:CDB:<Parent CDB SID>
- Change database credential type to SERVICE for PDBs

- PDB's can share Database Credentials if they share the same username and password, in this case leave SID(if required) field blank inside Database Credential

- Create Database Credentials for each CDB needed to be monitored.
- Keep database credential type to SID for CDB's.
- CDB Header:
  - SID:<CDB SID>:DB_CID:<DBC ID>:CDB:True

- CDB's can not share Database Credentials with PDB's.

- Append TNS_NAMES:<path_to_tnsfile> to SID header

Header Fields:
SID:SL121:DB_CID:141:TNS_ADMIN:/road/to/heaven:CDB:True
SSH(or PSH):92
PDB:pdb1:DB_CID:134:CDB:<Parent CDB SID>

Header Fields:
SID:NCDB:DB_CID:141
SSH(or PSH):92

- Create Database Credentials for each Non-CDB instance needed to be monitored
- Header will look like:
  - SID:<SID of Non-CDB>:DB_CID:<DBC ID>

- Append TNS_NAMES:<path_to_tnsfile> to SID header

Header Fields:
SID:NCDB:DB_CID:141:TNS_ADMIN:/road/to/heaven
SSH(or PSH):92

Header Fields:
SID:+MyASMDBName:DB_CID:141
SSH(or PSH):92

- Within Database Credentials for ASM instances, please put the Database Name in the DB_Name field and the instance name in the SID field for example,
  If you have the Database name, +MyASMDBName, and the instance name, +MyASMInstanceName1, the Database Credential will look like this:
- DB_Name: +MyASMDBName
  Oracle System Identifier(SID): +MyASMInstanceName1

- If you are using version 103 of the PowerPack, you can format the header as follows, `SID: <Oracle Instance SID>:PORT<Oracle Instance Port that is listening for DB requests>`. For example:

  ```
  SID:SL121:PORT:22
  ```

- If you are using version 103 of the PowerPack and discovering ASM instances, the entire ASM name must be included in the header field. For example, if an ASM instance is named "+ASM1" and your credential ID is 112, you should enter the following headers:

  ```
  SSH:112
  ```

  ```
  SID:+ASM1:PORT:1521
  ```

If you are using version 104 of the PowerPack and discovering an ASM instances, the entire ASM name must be include in the header field. For example, if an ASM instance is named "+ASM1" and your credential ID is 123, you should enter the following header:

```
SID:+ASM1:DB_CID:123
```

- `SERVICE_NAME: <Service Name acts as an alias to SID>:PORT:<Oracle Instance Port that is listening for DB requests>`. For example:

```
SERVICE_NAME:SL121:PORT:1521
```

NOTE: If the **SERVICE_NAME** is different than the **SID**, discovery will not work.

- If a host, or IP address, for the Oracle: Database is assigned to a different IP address from the server, you must apend an additional field to the SOAP credential. For example, `HOST`.

If each Oracle database instance is registered to a unique IP address, add `HOST` in addition to the SID and port. The PowerPack will connect the host IP addresses to the respective Oracle database instance. For example:

```
Ex - sid:<>:port<>:host:<>
```

If all of your Oracle database instances are registered to a single IP address that is not the same IP address as the server, add a new HTTP header with only the host information. This header will force the PowerPack to use the host IP address for all SIDs. For example:

```
HOST:<>
```

If no host is provided in the header, the PowerPack will use the root IP address or the server IP address to connect to all Oracle database instances.

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

- `<OS_TYPE>:<CRED_ID>`. The OS type and ID of the SSH/Key credential or PowerShell credential you created. For OS type, enter SSH for Linux or PSH for Windows. For example:

```
SSH:152
```

```

or

PSH:153
```

NOTE: Only one OS type per credential is supported.

- If you are monitoring CDB instances, you should have selected the *Oracle System Identifier (SID)* connect type in your database credential. Type `SID:<Oracle Instance SID>:DB_CID:<Credential ID>:CDB:True`. For example:

  ```
  SID:SL121:DB_CID:122:CDB:True
  ```

- If you are monitoring PDB instances, you should have selected the *SERVICE* connect type in your database credential. Type `PDB:<PDB Service Name>:DB_CID:<Credential ID>:CDB:<Parent Oracle Instance SID>`. For example:

  ```
  PDB:PDB121:DB_CID:123:CDB:SL121
  ```

- If you are monitoring non-CDB instances, you should have selected the *Oracle System Identifier (SID)* connect type in your database credential. Type `SID:<Oracle Instance SID>:DB_CID:<Credential ID>`. For example:

  ```
  SID:NCDB121:DB_CID:124
  ```

CAUTION: The **HOST** keyword and support for using a secondary IP to connect to an instance is not applicable on a multi tenant environment that doesn't share the same credential.

NOTE: For example HTTP headers when discovering PDBs in multiple CDBs, see the ***Example HTTP Headers to Discover PDBs in Multiple CDBs*** section.

- If your `tnsnames.ora` file is located in a custom path, enter the path in a header after a `DB_CID` or `PORT` entry in a header. For example:

```
SID:SL121:DB_CID:141:TNS_ADMIN:/example/file/path

or

SID:SL121:DB_CID:141:TNS_ADMIN:/example/file/path:CDB:True
```

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

NOTE: If your `tnsnames.ora` file is located in a custom path, you must setup the environment variable on the oracle server by using the following command, replacing the command with your `tnsname.ora` file path: `export TNS_ADMIN=INSERT_FILE_PATH`

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. Oracle Database is case-sensitive, so your PDB name must match the case shown either in your tnsnames.ora or in the ***PDB_NAME*** column of the **DBA_PDBS** table.

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 "Oracle: DB Example" 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.

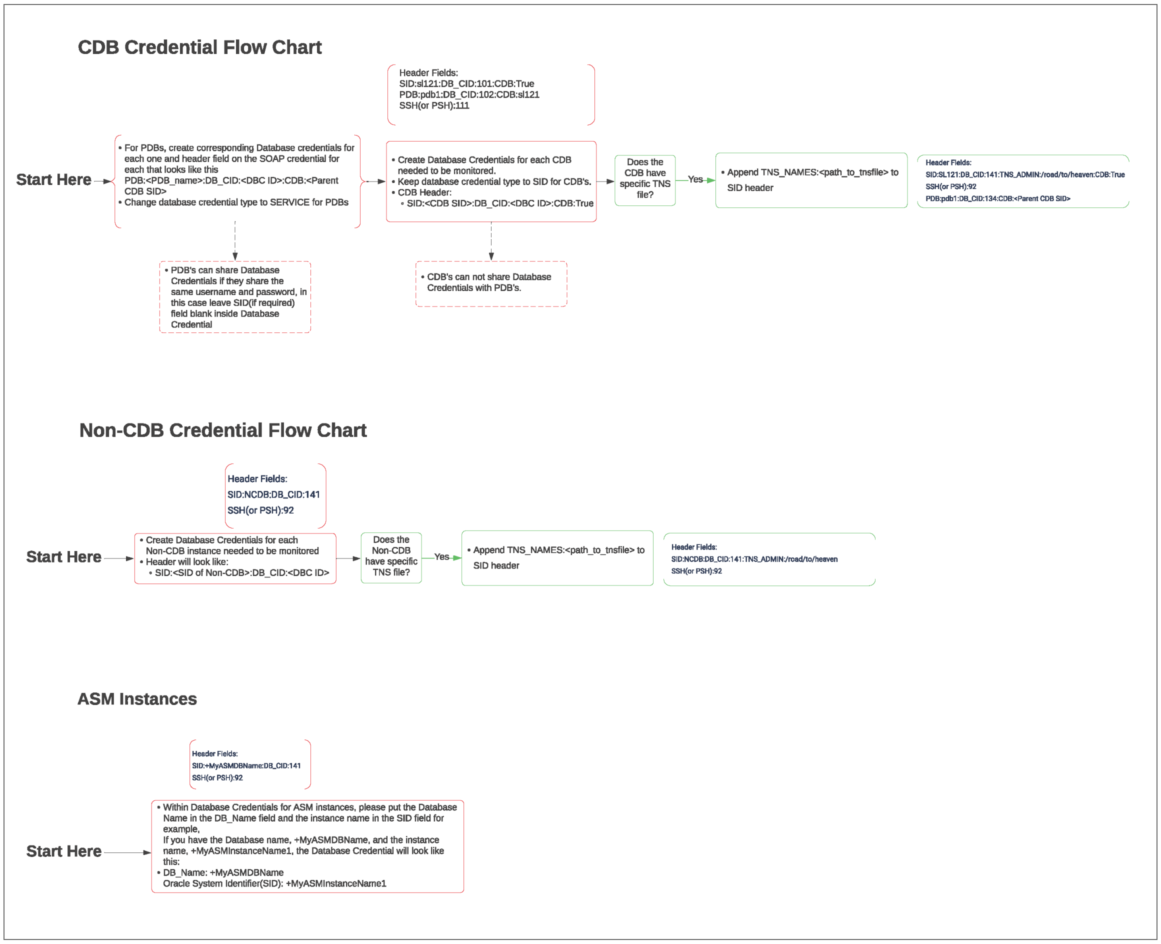
- ***URL***. Leave the default value of https://%D.
- ***HTTP Auth User***. Type the username for the Oracle Database account. If you created a database credential, this field is not required.
- ***HTTP Auth Password***. Type the password for the Oracle Database account. If you created a database credential, this field is not required.

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

- ***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:

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



CDB Credential Flow Chart

Start Here

- For PDBs, create corresponding Database credentials for each one and header field on the SOAP credential for each that looks like this PDB:<PDB_name>:DB_CID:<DBC ID>:CDB:<Parent CDB SID>
- Change database credential type to SERVICE for PDBs

- PDB's can share Database Credentials if they share the same username and password, in this case leave SID(if required) field blank inside Database Credential

Header Fields:
SID:sl121:DB_CID:101:CDB:True
PDB:pdb1:DB_CID:102:CDB:sl121
SSH(or PSH):111

- Create Database Credentials for each CDB needed to be monitored.
- Keep database credential type to SID for CDB's.
- CDB Header:
  - SID:<CDB SID>:DB_CID:<DBC ID>:CDB:True

- CDB's can not share Database Credentials with PDB's.

Does the CDB have specific TNS file?

Yes

- Append TNS_NAMES:<path_to_tnsfile> to SID header

Header Fields:
SID:SL121:DB_CID:141:TNS_ADMIN:/road/to/heaven:CDB:True
SSH(or PSH):92
PDB:pdb1:DB_CID:134:CDB:<Parent CDB SID>

Non-CDB Credential Flow Chart

Header Fields:
SID:NCDB:DB_CID:141
SSH(or PSH):92

Start Here

- Create Database Credentials for each Non-CDB instance needed to be monitored
- Header will look like:
  - SID:<SID of Non-CDB>:DB_CID:<DBC ID>

Does the Non-CDB have specific TNS file?

Yes

- Append TNS_NAMES:<path_to_tnsfile> to SID header

Header Fields:
SID:NCDB:DB_CID:141:TNS_ADMIN:/road/to/heaven
SSH(or PSH):92

ASM Instances

Header Fields:
SID:+MyASMDBName:DB_CID:141
SSH(or PSH):92

Start Here

- Within Database Credentials for ASM instances, please put the Database Name in the DB_Name field and the instance name in the SID field for example,
  If you have the Database name, +MyASMDBName, and the instance name, +MyASMInstanceName1, the Database Credential will look like this:
- DB_Name: +MyASMDBName
  Oracle System Identifier(SID): +MyASMInstanceName1

  - If you are using version 103 of the PowerPack, you can format the header as follows, `SID: <Oracle Instance SID>:PORT<Oracle Instance Port that is listening for DB requests>`. For example:

```
SID:SL121:PORT:22
```

- If you are using version 103 of the PowerPack and discovering ASM instances, the entire ASM name must be included in the header field. For example, if an ASM instance is named "+ASM1" and your credential ID is 112, you should enter the following headers:

  ```
  SSH:112

  SID:+ASM1:PORT:1521
  ```

  If you are using version 104 of the PowerPack and discovering an ASM instances, the entire ASM name must be include in the header field. For example, if an ASM instance is named "+ASM1" and your credential ID is 123, you should enter the following header:

  ```
  SID:+ASM1:DB_CID:123
  ```

- `SERVICE_NAME: <Service Name acts as an alias to SID>:PORT:<Oracle Instance Port that is listening for DB requests>`. For example:

  ```
  SERVICE_NAME:SL121:PORT:1521
  ```

NOTE: If the **SERVICE_NAME** is different than the **SID**, discovery will not work.

- If a host, or IP address, for the Oracle: Database is assigned to a different IP address from the server, you must add an additional header to the SOAP credential.

  If each Oracle database instance is registered to an IP address, add a header for host in addition to SID and port. The PowerPack will connect the host IP addresses to the respective Oracle database instance. For example:

  ```
  Ex - sid:<>:port<>:host:<>
  ```

  If all Oracle database instances are registered to a single IP address that is different from the server, add a header with only the host information. This header will force the PowerPack to use the host IP address to connect to all Oracle database instances. For example:

  ```
  HOST:<>
  ```

  If no host is provided in the header, the PowerPack will use the root IP address or the server IP address to connect to all Oracle database instances.

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

- `<OS_TYPE>:<CRED_ID>`. The OS type and ID of the SSH/Key credential or PowerShell credential you created. For OS type, enter SSH for Linux or PSH for Windows. For example:

  ```
  SSH:152

  or

  PSH:153
  ```

NOTE: Only one OS type per credential is supported.

- If you are monitoring CDB instances, you should have selected the *Oracle System Identifier (SID)* connect type in your database credential. Type `SID:<Oracle Instance SID>:DB_CID:<Credential ID>:CDB:True`. For example:

  ```
  SID:SL121:DB_CID:122:CDB:True
  ```

- If you are monitoring PDB instances, you should have selected the *SERVICE* connect type in your database credential. Type `PDB:<PDB Service Name>:DB_CID:<Credential ID>:CDB:<Parent Oracle Instance SID>`. For example:

  ```
  PDB:PDB121:DB_CID:123:CDB:SL121
  ```

- If you are monitoring non-CDB instances, you should have selected the *Oracle System Identifier (SID)* connect type in your database credential. Type `SID:<Oracle Instance SID>:DB_CID:<Credential ID>`. For example:

  ```
  SID:NCDB121:DB_CID:124
  ```

CAUTION: The **HOST** keyword and support for using a secondary IP to connect to an instance is not applicable on a multi tenant environment that doesn't share the same credential.

NOTE: For example HTTP headers when discovering PDBs in multiple CDBs, see the *Example HTTP Headers to Discover PDBs in Multiple CDBs* section.

- If your `tnsnames.ora` file is located in a custom path, enter the path in a header after a `DB_CID` or `PORT` entry in a header. For example:

```
SID:SL121:DB_CID:141:TNS_ADMIN:/example/file/path

or

SID:SL121:DB_CID:141:TNS_ADMIN:/example/file/path:CDB:True
```

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

NOTE: If your `tnsnames.ora` file is located in a custom path, you must setup the environment variable on the oracle server by using the following command, replacing the command with your `tnsname.ora` file path: `export TNS_ADMIN=INSERT_FILE_PATH`

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

Example HTTP Headers to Discover PDBs in Multiple CDBs

This example describes the HTTP headers that you can use in a SOAP/XML credential to discover PDBs in multiple CDBs. The following credential and database values are used for this example:

Credential or Database	Example Value
SL1 Database Credential ID	`24`
Linux Server Address	`192.0.2.96`

Credential or Database	Example Value
Non-CDB Instance	`NONCDB96`
First CDB Instance	`CDB96`
PDB Instances in First CDB Instance	`PDB1` `PDB2`
Second CDB Instance	`CDB96_SL`
PDB Instances in Second CDB Instance	`PDB1` `PDB2SL`

To monitor the Oracle database example values listed above, the following HTTP headers would be entered in the SOAP/XML credential:

- `SID:NONCDB96:DB_CID:24`
- `SID:CDB96:DB_CID:24:CDB:True`
- `PDB:PDB1:DB_CID:24:CDB:CDB96`
- `SID:CDB96_SL:DB_CID:24:CDB:True`
- `PDB:PDB1:DB_CID:24:CDB:CDB96_SL`
- `PDB:PDB2SL:DB_CID:24:CDB:CDB96_SL`

NOTE: For more information on creating a SOAP/XML credential, see the *Creating a SOAP/XML Credential* section.

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.

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 -f identity - t rsa
```

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 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:

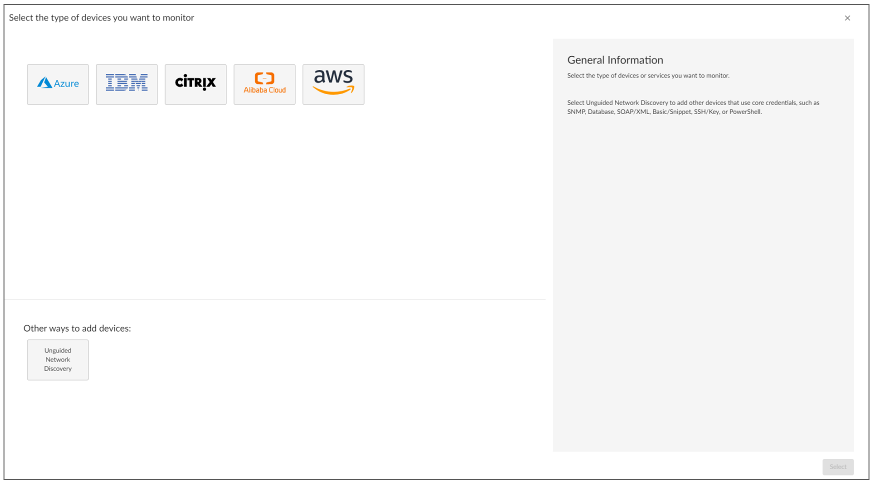
1. Find the `sshd_config` file.
2. Find the `PasswordAuthentication` command line, delete `yes`, and input `no`.
3. Restart the SSH service by running the following command:

    ```
    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:



Select the type of devices you want to monitor

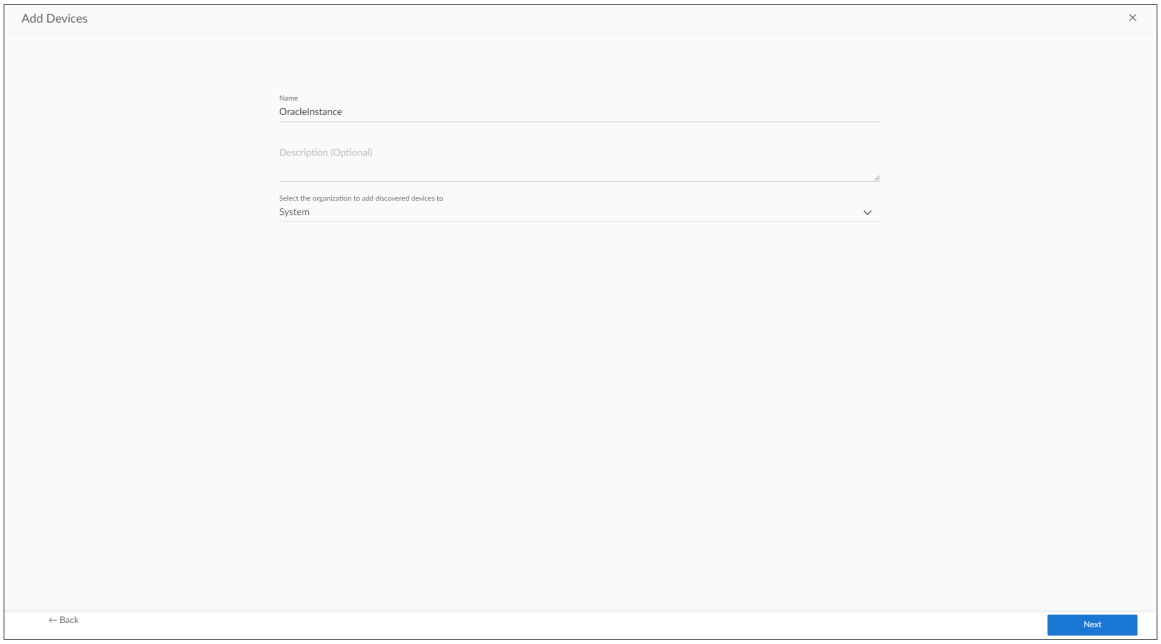
General Information

Select the type of devices or services you want to monitor.

Select Unguided Network Discovery to add other devices that use core credentials, such as SNMP, Database, SOAP/XML, Basic/Snippet, SSH/Key, or PowerShell.

Other ways to add devices:

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

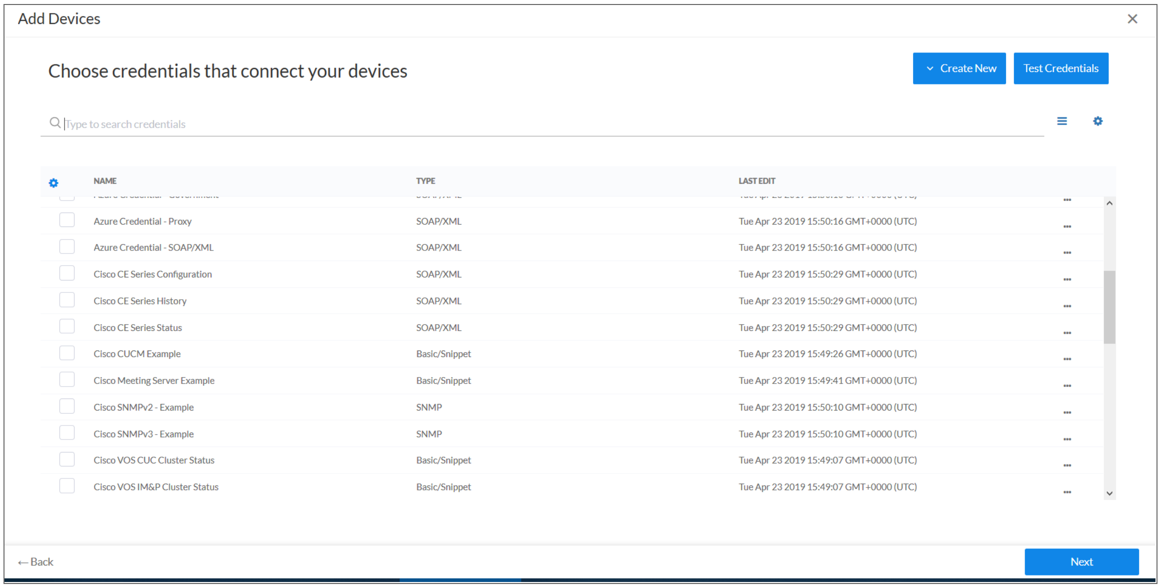
Description (Optional)

Select the organization to add discovered devices to: System

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:



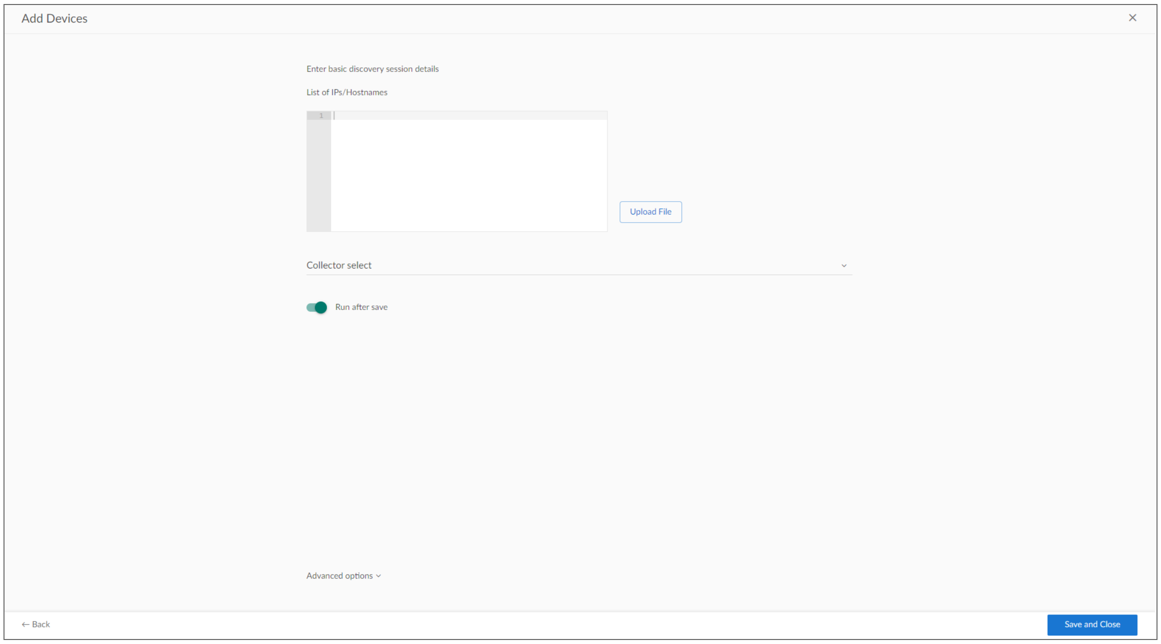
Choose credentials that connect your devices

Type to search credentials

	NAME	TYPE	LAST EDIT	
□	Azure Credential - Proxy	SOAP/XML	Tue Apr 23 2019 15:50:16 GMT+0000 (UTC)	...
□	Azure Credential - SOAP/XML	SOAP/XML	Tue Apr 23 2019 15:50:16 GMT+0000 (UTC)	...
□	Cisco CE Series Configuration	SOAP/XML	Tue Apr 23 2019 15:50:29 GMT+0000 (UTC)	...
□	Cisco CE Series History	SOAP/XML	Tue Apr 23 2019 15:50:29 GMT+0000 (UTC)	...
□	Cisco CE Series Status	SOAP/XML	Tue Apr 23 2019 15:50:29 GMT+0000 (UTC)	...
□	Cisco CUCM Example	Basic/Snippet	Tue Apr 23 2019 15:49:26 GMT+0000 (UTC)	...
□	Cisco Meeting Server Example	Basic/Snippet	Tue Apr 23 2019 15:49:41 GMT+0000 (UTC)	...
□	Cisco SNMPv2 - Example	SNMP	Tue Apr 23 2019 15:50:10 GMT+0000 (UTC)	...
□	Cisco SNMPv3 - Example	SNMP	Tue Apr 23 2019 15:50:10 GMT+0000 (UTC)	...
□	Cisco VOS CUC Cluster Status	Basic/Snippet	Tue Apr 23 2019 15:49:07 GMT+0000 (UTC)	...
□	Cisco VOS IM&P Cluster Status	Basic/Snippet	Tue Apr 23 2019 15:49:07 GMT+0000 (UTC)	...

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. The IP address can be assigned to a different IP address than the server.
   - ***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 discover the root device then the Database instance. All applicable Dynamic Applications will be aligned to the component.

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

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.

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 Chained Rows Stats
- Oracle: DB Components Status Config
- Oracle: DB Data Guard Gap Stats
- Oracle: DB Database Size Stats
- Oracle: DB Instance Config
- Oracle: DB Instance Invalid Object Stats
- Oracle: DB Integrity Metrics Stats

NOTE: The Oracle: DB Integrity Metrics Stats Dynamic Application uses the prior() function which requires SL1 version 8.14.2 or newer.

- Oracle: DB Log Alerts Config

NOTE: The Oracle: DB Log Alerts Config Dynamic Application requires that the database administrator grant SELECT privileges in the v$diag_alert_text to the monitoring user.

- Oracle: DB Logswitch Rate Stats
- Oracle: DB Long Running Session
- Oracle: DB Non-Archived File System Stats
- Oracle: DB Open Cursors per Session Stats
- Oracle: DB Performance Stats

- Oracle: DB Resource Stats
- Oracle: DB RMAN Backup Status Config
- Oracle: DB Session Stats
- Oracle: DB Tablespace Stats
- Oracle: DB Tablespace Temp Stats
- Oracle: DB Tablespaces and Datafiles Status Config

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.

You should see the following Dynamic Applications aligned to **Non-CDB RAC Oracle Database instances on Linux and Solaris Systems**:

- Oracle: DB Archived File System Stats
- Oracle: DB Chained Rows Stats
- Oracle: DB Components Status Config
- Oracle: DB Data Guard Gap Stats
- Oracle: DB Database Size Stats
- Oracle: DB Instance Invalid Object Stats
- Oracle: DB Instance Config
- Oracle: DB Integrity Metrics Stats

NOTE: The Oracle: DB Integrity Metrics Stats Dynamic Application uses the prior() function which requires SL1 version 8.14.2 or newer.

- Oracle: DB Log Alerts Config

NOTE: The Oracle: DB Log Alerts Config Dynamic Application requires that the database administrator grant SELECT privileges in the v$diag_alert_text to the monitoring user.

- Oracle: DB Logswitch Rate Stats
- Oracle: DB Non-Archived File System Stats
- Oracle: DB Open Cursors per Session Stats
- Oracle: DB Performance Stats
- Oracle: DB RAC Disk Group Space Stats
- Oracle: DB RAC Flash Recovery Stats
- Oracle: DB RAC Global Cache Stats

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

- Oracle: DB Resource Stats
- Oracle: DB RMAN Backup Status Config
- Oracle: DB Session Stats
- Oracle: DB Tablespace Stats
- Oracle: DB Tablespace Temp Stats
- Oracle: DB Tablespaces and Datafiles Status Config

You should see the following Dynamic Applications aligned to **Non-CDB RAC Oracle Database instances on Windows Systems**:

- Oracle: DB Chained Rows Stats
- Oracle: DB Components Status Config
- Oracle: DB Data Guard Gap Stats
- Oracle: DB Database Size Stats
- Oracle: DB Instance Config
- Oracle: DB Instance Invalid Object Stats
- Oracle: DB Integrity Metrics Stats

NOTE: The Oracle: DB Integrity Metrics Stats Dynamic Application uses the prior() function which requires SL1 version 8.14.2 or newer.

- Oracle: DB Log Alerts Config

NOTE: The Oracle: DB Log Alerts Config Dynamic Application requires that the database administrator grant SELECT privileges in the v$diag_alert_text to the monitoring user.

- Oracle: DB Logswitch Rate Stats
- Oracle: DB Open Cursors per Session Stats
- Oracle: DB Performance Stats
- Oracle: DB RAC Disk Group Space Stats
- Oracle: DB RAC Flash Recovery Stats
- Oracle: DB RAC Global Cache Stats

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

- Oracle: DB Resource Stats
- Oracle: DB RMAN Backup Status Config
- Oracle: DB Session Stats
- Oracle: DB Tablespace Stats
- Oracle: DB Tablespace Temp Stats
- Oracle: DB Tablespaces and Datafiles Status Config

You should see the following Dynamic Applications aligned to **CDB root device (non-RAC) instances**:

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

You should see thee following Dynamic Applications aligned to **CDB with RAC instances**:

- Oracle: DB Archived File System Stats
- Oracle: DB Non-Archived File System Stats
- Oracle: DB Instance Config
- Oracle: DB RAC Disk Group Space Stats
- Oracle: DB RAC Flash Recovery Stats
- Oracle: DB RAC Global Cache Stats
- Oracle: DB RMAN Backup Status Config
- Oracle: DB PDB Discovery

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

- Oracle: DB ASM Diskgroup Config
- Oracle: DB ASM Instance Config
- Oracle: DB Instance Config
- Oracle: DB RMAN Backup Status Config

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

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 the 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://&silo_args=column=<column_name>&table=<table_name>
```

Snippet argument with a filter:

```
oracle://&silo_args=column=<column_name>&table=<table_
name>&filter=<where clause>
```

Example:

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

oracle://&silo_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.

Chapter

3

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:

Device Dashboard

The *Oracle: Database* PowerPack includes a device dashboard that provides summary information for an Oracle Database instance. The device dashboard is aligned as the default device dashboard for the Oracle Database instance.

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 SL1 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 SL1 gets `ORACLE_HOME` from oratb.

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

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

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